A SYSTEMS ANALYSIS OF HUMANS AND MACHINES IN SPACE ACTIVITIES

vol. I

by

DAVID GORDON STUART
B.S., Massachusetts Institute of Technology (1975)
S.M., Massachusetts Institute of Technology (1979)

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

DOCTOR OF SCIENCE

at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 1986

Signature of Author
Department of Aeronautics and Astronautics May 12, 1986

Certified by
Professor Rene H. Miller
Thesis Supervisor
H.N. Slater Professor of Flight Transportation

Certified by
Professor David L. Akin
Rockwell International Assistant Professor
of Aeronautics and Astronautics

Certified by
Professor Amedeo R. Odoni
Professor of Aeronautics and Astronautics
and Civil Engineering

Certified by
Professor Thomas B. Sheridan
Professor of Engineering and Applied Psychology

Accepted by
Professor Harold Y. Wachman
Chairman, Department Graduate Committee

A SYSTEMS ANALYSIS OF HUMANS AND MACHINES IN SPACE ACTIVITIES

by
DAVID GORDON STUART

Submitted to the Department of Aeronautics and Astronautics
on April 11, 1986, in partial fulfillment of the
requirements for the Degree of Doctor of Science in
Aeronautics and Astronautics

ABSTRACT

A study is performed on the economic tradeoffs of humans and machines in space activities, including the development of a method to assign parts of a job to humans and machines, while minimizing total program cost.

The job is modeled as a set of concurrent tasks, performed at a permanent space facility by a variety of human-machine options (e.g. humans, teleoperators, or automated equipment). A detailed line-item cost model is developed, showing the effects of the human-machine options on total program costs. From this model, a general cost equation is extracted. Because of the effects of the task completion times, this equation is nonlinear.

A selection of existing optimization techniques (e.g. fractional functionals, quadratic programming, linear approximation programming) is applied to the general cost equation. Because of its nonlinearity, these methods are either unsuitable or applicable only at heavy computational cost. However, the combinatorial character of the problem is recognized: potential solutions are combinations of assignments of human-machine options to tasks. Also, a theorem is proved: under the study's assumptions, for minimum cost, all the tasks must have equal completion times.

These observations lead to the development of the Staged Optimization Method (SOM). This iterative method alternates the selection of human-machine options for individual tasks with the minimization of the complete cost equation. Because it takes advantage of the mathematical character of the problem, the SOM is a polynomial algorithm, which reaches the optimal solution without requiring large amounts of computation. The robustness of the SOM and the effects of the study's assumptions are discussed, and future extensions of the method are suggested.

The Staged Optimization Method is illustrated by applying it to a case example: the in-orbit construction of a Microwave Radiometer. This example is developed in some detail, including task descriptions, preliminary designs of human-machine options, and estimates of parameters from past data. The method is demonstrated on several versions of the case example. Some of these examples include learning curves in procurement, to display the SOM's ability to handle some nonlinear cost functions.

DEDICATION

To the memory of
Frederick H. Osborn Jr.

The stars his destination.

ACKNOWLEDGEMENTS

I have been at MIT for fifteen years, which is a long stay by anyone's reckoning. During that time I have learned more than I thought I could, worked on an astronomy satellite, learned a martial art, helped to design a space habitat and a space factory, rebuilt a vacuum chamber for vapor deposition, studied space applications of humans and machines, played soccer and ice hockey, and got married. It's been a wild ride.

Fortunately, many people helped. Most of all, I thank my wife, Dale, without whom I would have lost all my marbles long ago. True, she wasn't here for all of those fifteen years, but somehow I can't remember what went on before she appeared...

Thanks also to my long-suffering doctoral committee, Professors Rene Miller, David Akin, Amedeo Odoni, and Thomas Sheridan. They are my teachers and my friends, and I am proud to have their names on my title page.

Directly and indirectly, many people (and machines) helped with this thesis. In particular, I give special thanks to:

Teresa Villarreal and Ali Azar, who searched NASA's catacombs with me, and hauled away treasure;

Joe Oliveira, a man for all seasons (in any combination);

Werner Gruhl of NASA Headquarters, for the wisdom of his experience;

Shirley LaCroix and Eugene Hill of NASA Headquarters, who led us to gold;

Wayne Zimmerman of JPL - he and I tilt at the same windmills;

Georg von Tiesenhausen of NASA Marshall, whose ARAMIS contract released this thesis topic from Pandora's box;

Professors Robert Simpson and Bruce Walker, and Dr. Dennis Mathaisel, who actually read this volume;

Georges Whittinghill of NASA JSC, and Jack Kelly, Chris Roberts, and Marc Vaucher of the CSP, who gave me good data when I needed it most;

The Corvus Concept and Bob Lucas, an unbeatable team;

All the wonderful people at Kendall Press, who work a lot faster than I do;

And all my colleagues in the SSL - they are the best.

Lastly but not leastly, I want to thank my family for all of their love and encouragement: my sister Lesley, whose graduation was the example that I needed; Mathilde Schwartz, whose love crosses the Atlantic disguised as knitted sweaters and chocolate; my grandmother, Betty Winkler, for her kindness and support to Dale and to me. And most of all, I thank my parents, Ralph and Lilian Stuart Smith, who have given me everything, including the ability to reach for the universe - they are truly the very finest people that I know.

OBFUSCATION

... The Graduate Student pushed his eyeglasses up his nose, buttoned his jacket, and paced in front of Alice. He spoke in a harried voice:

"I have this theory, you see, that a thesis is like... a brontosaurus!"

The Graduate Student struck an academic pose, with his shoulders back and one finger pointed at the sky. "You see, like a thesis, a brontosaurus starts out harmless enough, with a little reptilian head, with beady brown eyes and a smile on its scaly face... But as you move along it," he added, striding along an imaginary dinosaur, "it grows and <u>grows</u> and <u>GROWS!</u>"

He flung his arms wide to measure the metaphorical reptile, and his sleeves retreated up to his elbows. "And it has four <u>huge</u> legs, solid as Truth, sunk to the knees in the Universal Swamp!" His arms stretched even further, now measuring the universe. "And inside, there are <u>miles</u> and <u>miles</u> of convoluted intestinal explanations, and eleven appendices!"

Overwhelmed by the appendices, the Graduate Student fell silent. His arms dropped and his glasses slid down his nose. His eyes saddened, and more wrinkles appeared in his clothing. In a tired voice, he continued:

"And like the back end of the brontosaurus, the thesis tails off and tails off, and somehow you can never reach the tip, the very end of it..."

"Oh," said Alice, "... what's a brontosaurus?"

from
The Adventures of Alice at MIT
Lewis Nuttenbolt

TABLE OF CONTENTS

Chapter 4: Case Example: Construction of a Microwave Radiometer

CHAPTER 1:

INTRODUCTION AND GENERAL APPROACH

1.1 Introduction:

At this time the U.S. space program faces some important questions on future directions in space utilization, specifically on the design and operation of permanent space facilities such as the Space Station. Some of these essential questions involve the future roles of humans and machines in space activities.

An important aspect of the Space Shuttle and of permanent space facilities is that they make human presence in space less costly and potentially more productive than in the past. Continuous human involvement in space research, technology development, and space operations will soon be possible. As astronauts learn to use new pressure suits, manned maneuvering units, and specialized tools, space walks will become routine and valuable.

However, this is also the decade of automation and robotics. Current advances in manipulators, remote sensors, microprocessor chips, and machine intelligence hold the potential for significantly more accurate, versatile, and reliable machinery to perform tasks in space. In the near-term, satellites in geosynchronous orbit will require teleoperation and automation, since human presence is not yet available there. And U.S. space stations and factories are expected to be substantially automated, especially those sections requiring near-total zero-gravity.

As in any engineering decision, the relative costs of these alternatives will be a strong element in the choices between options. However, the currently available cost models are not adequate to describe the costs of human and machine options for permanent space facilities. And the current costing techniques are not capable of minimizing the cost of a large-scale job in space by assigning tasks to humans and machines.

Part of this insufficiency is due to lack of data: a number of cost-driving variables have uncertain values. In some cases, this is because some new capabilities have not yet been tested. As examples, the productivities of humans and machines in space construction have not yet been assessed, and the development cost of a versatile teleoperator has not yet been accurately predicted. In other cases, certain important pricing policies have not yet been elaborated. For example, the price of cargo launch on the Shuttle is still under debate.

Therefore final decisions on the merits of human and machine options must wait in part until design studies, ground and space simulations, and policy decisions produce realistic values for the uncertain parameters. To help this process, the cost-driving parameters should first be identified by analyses of the economics of humans and machines in space. These important parameters can then be studied in detail.

Another inadequacy is that costing techniques for space activities model traditional space flights. In these models, almost all of the costs are incurred before launch, in research and development, and the operations costs consist entirely of ground support. Future activities at permanent space facilities require new models of space operations, which may include multiple launches of hardware, humans, and consumables into space, and possibly development and procurement of improved hardware late in the program. It is likely that the costs of future programs will be dominated by operations costs incurred after the initial launch.

In past space programs, decisions on the use of humans or machines were based largely on availability of people or technology. If humans were present, then they performed any tasks not possible by machines (e.g. the Skylab repair operations). If humans were not able to perform a task, machines were developed to handle them (e.g. the Viking and Voyager probes to other planets).

However, this situation is changing. With the advent of the Shuttle and of space facilities, many space activities can be performed either by humans <u>or</u> machines, within the bounds of available technology. For example, certain Shuttle contingency operations can be performed either by humans in extravehicular activity or by the Remote Manipulator System. With the development of teleoperator systems for the Space Station, many more such choices will become available.

In that case, the deciding factors between human and machine options become economic rather than technological. The appropriate roles for humans and machines are dictated by their relative development, procurement, and operations costs, in the context of the job to be done. There is therefore a need for quantitative analyses of the economic tradeoffs between humans and machines in space. The cost elements of each option must be identified, and the impact of these elements on overall program cost must be modeled.

Current costing methods can calculate the cost of doing the entire job with humans, or the entire job with machines, and they can compare these total costs. But this process imposes a single option on all the tasks to be done, which may not be the true least-cost solution.

In general, the issue is not human versus machine, but rather the most appropriate mixture of the two. It is likely that future jobs in space will be performed by partnerships of humans and machines, with each option assigned to tasks in such a way that the overall program cost is minimized.

Therefore there is a need for a cost optimization method which considers individual tasks within a job and computes specific numbers of humans or machines for each task, while minimizing the total cost. As will be shown later in the study, each task cannot be optimized just by itself: to find the true minimum, the individual tasks must be considered in the context of the complete job.

1.2 Objectives of this Study:

This study has three interrelated primary objectives:

1) This study will develop an optimization method that will assign a space job's individual tasks to humans or machines, minimizing overall program cost. This method will calculate the optimal numbers of humans or machines to be applied to each task, and therefore the optimal completion time for each task. It will be applicable to a wide variety of space activities, and to various human and machine options.

2) A line-item cost model will be developed, relating the physical parameters of space activities to overall program cost. This model will include detailed descriptions of operations costs, and be applicable to a variety of space activities. The line-item cost model will generate a credible overall cost equation, and will explain the physical sources of cost elements of that equation.

3) A detailed case example will be developed, to generate realistic numbers for examples used in this study's theoretical development. The case example will also demonstrate the process required to apply this study's methods to actual scenarios, and will illustrate the validity of assumptions in the study method. Also, sensitivity analyses on the case example will identify its cost-driving parameters.

1.3 General Approach to this Study:

The major steps in this study are flowcharted in Figure 1.1. The chronology of the research runs from top to bottom in the figure. The items in double boxes cover the study's first objective, the theoretical development of the Staged Optimization Method. The single-boxed items deal with the study's other objectives, the line-item cost model and the case example. The chapters and appendices describing each step are indicated next to each box.

The chapters in this report concentrate on the double-boxed items, which include this study's theoretical contribution. Thus Chapter 2 presents the general human-machine cost equation, and describes how its nonlinearities makes traditional optimization techniques unsuitable. Chapter 3 presents this study's answer to the problem, the Staged Optimization Method, and explores its operation. Chapter 4 illustrates this method with a case example of space construction, presenting both simple and complex cases.

The study's other objectives (the line-item cost model and the details of the case example) are developed mostly in appendices. Their results are used throughout the report, particularly in the examples in Chapter 4.

To achieve its objectives, the study method includes four consecutive efforts. These are described in the following sections.

1.3.1 Development of Line-Item Cost Model and Case Example:

Since this study's principal objectives involve the economic tradeoffs between humans and machines in space, the first step is to examine how humans and machines are related to overall program costs. This is done by building a detailed line-item cost model.

The level of detail of this model is set by the goal of this study: an optimization method which assigns human and machine options to individual tasks within a job in space. Therefore the cost model must include those physical details of human and machine options which give them different cost behaviors (e.g. hardware masses, consumables requirements, and productivities). The model must also differentiate between individual tasks, so that the costs for each task can be separately calculated.

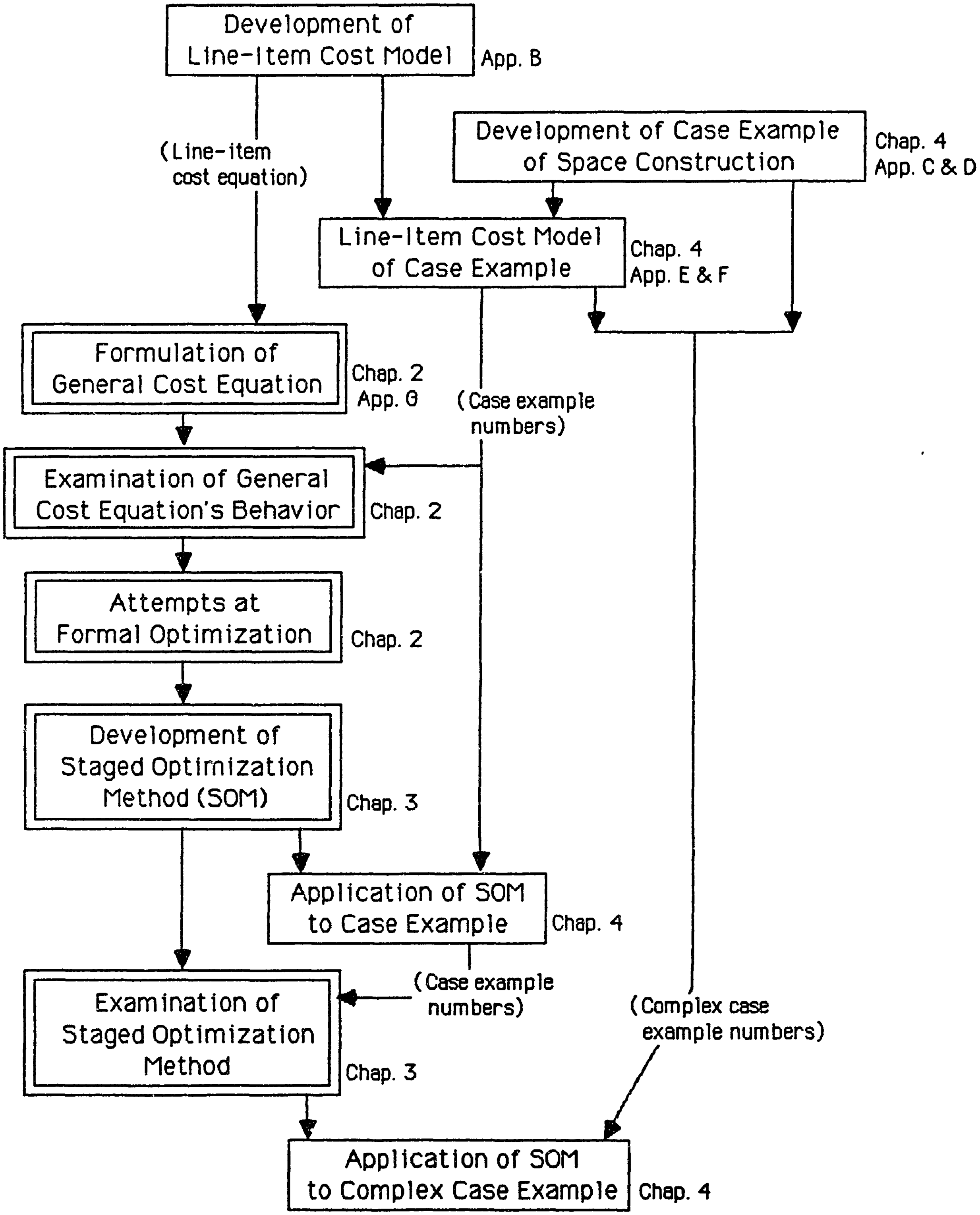

Development of Line-Item Cost Model
App. B
(Line-item cost equation)
Development of Case Example of Space Construction
Chap. 4
App. C & D
Line-Item Cost Model of Case Example
Chap. 4
App. E & F
Formulation of General Cost Equation
Chap. 2
App. G
(Case example numbers)
Examination of General Cost Equation's Behavior
Chap. 2
Attempts at Formal Optimization
Chap. 2
Development of Staged Optimization Method (SOM)
Chap. 3
Application of SOM to Case Example
Chap. 4
(Case example numbers)
(Complex case example numbers)
Examination of Staged Optimization Method
Chap. 3
Application of SOM to Complex Case Example
Chap. 4

FIGURE 1.1 FLOWCHART OF MAJOR STEPS IN THIS STUDY

This model should also be applicable to a range of space activities, and to a variety of human-machine options. Therefore the equations in the cost model are expressed in variables, which take on specific values only in particular applications. For example, the section which models teleoperator costs expresses hardware masses, consumables, maintenance factors, and support requirements as variables, so that it can model a variety of teleoperator designs. For a particular design, these variables' values are set by numerical inputs to the model.

The line-item cost model has two purposes:

1) To generate a credible detailed cost equation for humans and machines in space activities;

2) To develop a physical understanding of the problem, i.e. a knowledge of the physical meanings of individual terms in the cost equation.

The detailed cost equation will be the starting point for the theoretical development of this study. The physical understanding of the problem is necessary to judge the validity of modeling assumptions and optimization methods.

To develop that physical understanding further, a detailed case example is presented. The scenario selected is the in-orbit construction of a 140-ton Microwave Radiometer, using humans in extravehicular activity, free-flying teleoperators, and automated beam-builders. The development of this case example includes the definition of construction tasks and the preliminary design of human support and machine hardware.

The cost model is then applied to the case example to generate example numbers (e.g. spares and consumables requirements, completion times for tasks) and line-item costs for the construction scenario. Sensitivity analyses are also performed to examine the effect of various parameters (e.g. productivities, consumables requirements) on overall program costs.

The numbers developed for this example will be used throughout the study to illustrate theoretical concepts with credible numbers. For instance, this study's Staged Optimization Method will be demonstrated on this case example.

1.3.2 Theoretical Examination of General Cost Equation:

The detailed cost equation provided by the line-item model is not in a form suitable for examination and attempts at optimization. Therefore its terms are expanded, rearranged, and recombined into a general cost equation. This equation expresses the total program cost (ProgCost) in terms of a set of "decision variables":

a) the numbers of humans, teleoperators, and units of automated equipment assigned to each task;

b) the completion times for each task.

The study then examines the behavior of the general cost equation, to determine what type of optimization method will be required. To display the cost behavior, a "restricted" case example is formulated, applying only two options (humans and teleoperators) to a single task.

The total cost ProgCost for this one-task two-option case is a function of the numbers of humans (Nhum) and the numbers of teleoperators (Ntel) assigned to the task. ProgCost forms a two-dimensional surface over the Nhum-Ntel plane, and can be displayed in contour plots. Several contour plots are used to illustrate the saddle-shaped behavior of the total cost, and the effects of parameter values (e.g. hardware masses, productivities) on this surface.

Based on this understanding of the cost behavior, several "traditional" optimization techniques are then attempted on the restricted case. The intention is to extend any successful technique to the more complex multi-task, multi-option reality. The nonlinear character of the cost equation suggests either a problem transformation or nonlinear programming.

The varying successes of both approaches are described in this study. The most promising technique is a piecewise-linear approximation of the basic nonlinearity in the cost equation. However, even this method involves large amounts of computation just for the restricted case. Its application to the more complicated reality would lead to a computational explosion.

Despite these difficulties, useful insights into the problem are gained from these optimization attempts. These insights suggest that the overall problem is fundamentally combinatorial: the possible solutions are permutations of choices of humans and machines for individual tasks. This in turn suggests a reevaluation of the optimization approach.

1.3.3 Development of Staged Optimization Method:

This reevaluation leads to the development of this study's Staged Optimization Method (SOM). The first step is a theorem which proves that, under the study's assumptions, for minimum total cost all of the job's tasks should end at the same time, i.e. the individual task completion times and the overall job completion time have the same value.

The effect of this theorem is to reduce the complexity of the general cost equation, making ProgCost a function of the numbers of humans and machines assigned to each task and of the total job time (Jobtime). And for each combinatorial option (i.e. each permutation of choices of humans and machines) the cost equation can be recast in terms of the job time alone.

The optimization can now be performed iteratively in stages. Starting with an initial combination of choices, the total cost ProgCost is minimized with respect to Jobtime. This identifies the optimum job completion time for that permutation.

Next, this optimum job time is used to select a more optimal combination of human and machine choices. The cost equation is recast for this permutation, and once again optimized with respect to Jobtime. This process repeats, alternately selecting a human-machine permutation and optimizing total job time, until it converges on a minimum-cost solution.

To illustrate the Staged Optimization Method, it is applied to this study's case example of space construction. This is a four-task example with three human-machine options, and the method computes the specific numbers of humans, teleoperators, and units of automated equipment to be assigned to each task for minimum total program cost.

1.3.4 Examination of Staged Optimization Method:

Using numbers from the case example, the mechanics of the SOM are described, including graphical representations of the method's path through the design space. Although this is partially a combinatorial method, it does not have to consider every permutation of human-machine choices to reach a solution.

In fact, its combination-selection method lets it slice rapidly through the design space, reducing the computational load by 95% in the case example (this reduction would be even higher in more complex cases). Part of this speed is due to an efficient method to select a likely initial combination to start the optimization process.

While the method searches for a global cost minimum, it is possible for it to become trapped in a local minimum in its design space. This issue is discussed, including a description of what physical circumstances in a space scenario might lead to such an ambiguity. This problem can be avoided by some straightforward enhancements to the SOM, which are described in this study.

The modeling and procedural assumptions made during the theoretical development of the SOM are reviewed. Their effects on the method's application to actual space scenarios is discussed. This establishes the types of space activities for which this method is useful, and points out further theoretical developments to extend the method beyond its current assumptions.

The advantages of the Staged Optimization Method are explored. The SOM's computational efficiency is particularly useful, because it allows the method to be applied to large scenarios, i.e. with many tasks and many human-machine options.

Also important is the SOM's ability to optimize nonlinear cost functions, including cost terms with fractional exponents (e.g. learning curves in procurement). This is possible because the optimization is performed in stages: at each stage, the function to be optimized is not the multi-variable general cost equation but a specific single-variable case of the equation. Therefore a variety of nonlinear optimization techniques can be used (e.g. gradient-search).

To illustrate these advantages, the SOM is applied to a more complex case example, with ten construction tasks and learning curves.

1.4 Summary of Previous Work:

The remainder of this chapter reviews relevant previous work in the disciplines of space mission definition, space equipment design, parametric cost modeling, and formal optimization.

1.4.1 Parametric Analysis and Optimization:

Parametric analysis, one of the tools of operations research, has been in use for at least two decades. The optimization methods applied to the parameters, however, have been steadily improved, including some recent advances.

The most widely used optimization method is Linear Programming (LP), which minimizes a linear "objective function" subject to linear constraints (Ref. 1-1). The principal drawback to LP is that the required linear cost model may not be a sufficiently accurate representation of reality. Many real-life situations exhibit nonlinear cost behavior, or nonlinear constraints.

Therefore several variations of LP have been developed to expand its range of application (Ref. 1-1). Integer Programming (IP) deals with problems in which the values of certain variables must be integral (e.g. when hardwa must be purchased by the unit). Dynamic Programming tackles problems which include succesive stages, in which early decisions affect the environment of later decisions.

Another variation on LP is Diagonal Ascent Linear Programming (DALP), developed by Akin (Ref. 1-2). This technique tackles cost functions which include a nonrecurring 'existence' cost, independent of the decision variables. For example, the nonrecurring cost of a factory in space will be incurred first, regardless of the operational level of output of the factory.

There is also Nonlinear Programming, a collection of techniques to deal with nonlinear objective functions or constraints (Ref. 1-1). One such method is Separable Programming, which uses substitutions to isolate the decision variables, and several techniques to optimize piecewise-linear approximations of the equations. Another technique is Quadratic Programming, which deals with quadratic objective functions with linear constraints.

As will be shown in Chapter 2, the general cost equation in this study is fundamentally nonlinear, and it is therefore the nonlinear programming methods that are of greatest interest. Attempts to apply these techniques will be described. The difficulties encountered will lead to a reevaluation of the problem, and to the development of this study's Staged Optimization Method.

1.4.2 Parametric Cost Modeling and Optimization of Space Scenarios:

There are very few instances in the literature of applications of operations research techniques to space utilization scenarios. There have been a number of applications of parametric cost modeling to small, specific areas, such as choices between types of power systems for satellites, or optimal orbit-change strategies. But large-scale models of space activities have seldom been constructed.

In 1978 and 1979, Miller and Akin published several papers on parametric analyses of earth-to-orbit launch vehicles for support of large-scale construction in space (Refs. 1-3, 1-4). These studies adapted cost modeling techniques from air transportation analyses to the particular physical and economic constraints of space activities. They identified two major cost-driving parameters in that problem: launch cost (in dollars/kilogram) from earth to orbit, and human productivity (amount done/person-hour) in orbit.

Building on this work, Akin developed a cost model of large construction projects in space, using the manufacture of Solar Power Satellites (SPS) as a test example (Ref. 1-2). This study specifically addressed two options for SPS construction: raw materials from earth, and raw materials from the moon. Using DALP, an optimum transition schedule from earth materials to lunar materials was computed. However, the models in that study concentrated on transportation (vehicles and orbital choices), and therefore did not include details of human and machine options for specific tasks.

In a NASA-sponsored study of lunar resources utilization, Bock et al also modeled costs of SPS construction from earth and lunar materials (Ref. 1-5). In a related study, the MIT Space Systems Lab designed a space manufacturing facility to produce SPS components from lunar materials (Ref. 1-6). Both studies concluded that manufacture of SPS components was potentially cheaper in space, in part due to intensive automation of the space factory. However, detailed tradeoffs of humans and machines were not performed in either study.

1.4.3 Human Applications in Space:

Since 1975, the MIT Space Systems Lab (SSL) has explored human productivity in space construction. Zero-g structural assembly has been simulated in underwater tests, both at MIT and at the NASA Marshall Space Flight Center Neutral Buoyancy Facility (Refs. 1-7, 1-8, 1-9). Besides simulating very high productivities, this research has also focused on the usefulness of various assembly aids, such as the Shuttle RMS and an underwater equivalent to the Manned Maneuvering Unit.

These neutral buoyancy simulations have led to a Shuttle flight experiment, called Experimental Assembly of Structures in EVA (EASE). Although its final results are not yet available, the preliminary results validate the underwater simulations of assembly by humans.

These simulations provide the best current estimates of human productivity in space construction. This data, and the experience developed in these projects, are used in the case examples in this study.

1.4.4 Machine Applications in Space:

In recent years NASA has devoted some R&D to the potential uses of machines in space. Some theoretical and experimental work on automatic docking and satellite repair techniques has been done by NASA in-house, particularly at the Marshall Space Flight Center. Recently, some of this research has led to development of the Teleoperator Maneuvering System (TMS) concept, expected to be flown from the Shuttle by 1991.

The TMS (recently renamed the Orbital Maneuvering Vehicle, or OMV) is a free-flying multipurpose satellite tender (Ref. 1-10). Eventually, one or more OMV units would be parked at the Space Station to handle satellite retrievals, inspections, and repair jobs near the facility.

To perform maintenance and repair tasks, the OMV might carry a Remote Orbital Servicing System (ROSS), consisting of a pair of manipulator arms, stereo cameras, and support equipment (Ref. 1-11). The designs of the OMV and ROSS are used in the design of an example space teleoperator for this study.

The MIT Man-Machine Systems Lab has performed extensive research on teleoperation and its applications. Some of these studies, particularly on human and supervisory control of teleoperators (Refs. 1-12, 1-13), contribute to the performance prediction of space teleoperators in this study's case example.

The MIT Space Systems Lab has extended its underwater simulations to machines by building the Beam Assembly Teleoperator (BAT), which attempts the same neutral buoyancy assemblies done by humans (Ref. 1-14). In addition, the SSL is currently developing the underwater Multimode Proximity Operations Device (MPOD) which will be used to study human factors in the control of free-flying teleoperators. This experience also contributes to the design of teleoperators in this study.

NASA has also commissioned the design of some automated machinery for space use. One such device is the Beam-Builder (Ref. 1-15) which produces truss beams for space construction. A version of this machine is included in this study's case example.

1.4.5 Human-Machine Tradeoffs:

Most previous human-machine tradeoff studies (e.g. Ref. 1-16) have concentrated on defining the ranges of capability of humans (e.g. endurance, sensory capacity), and on defining factors requiring the use of machines (e.g. radiation hazards to humans). The proposed tradeoff methods are qualitative, listing relative qualities of humans and machines and attempting to match these qualities to task descriptions.

Some systematic methods have also been developed. Price (Ref. 1-17) has formulated an iterative process for functional allocation of humans and machines. This method mixes qualitative evaluations and quantitative checks on proposed designs. The overall method is stated in general terms, however, leaving the development of figures of merit and their numerical evaluation to the user.

An extensive study by the MIT Space Systems Lab reviewed the entire field of Automation, Robotics, and Machine Intelligence Systems (ARAMIS). In Phase I (Ref. 1-18), potential applications of ARAMIS to space tasks were identified. Candidate options for each task were described, ranging from fully-human to fully-machine. These options' relative merits were evaluated using a set of qualitative scales. A systematic method was used to identify promising human-machine options for NASA to develop. However, the study's methods consider the space tasks individually, not in the context of a space project.

Phase II of the ARAMIS study (Ref. 1-19) concentrated on the use of telepresence in space activities, and defined a proposed near-term telepresence unit for NASA to develop. In both phases, however, detailed economic tradeoffs between humans and machines were outside the scope of the ARAMIS study.

A study entitled The Humans Role in Space (THURIS) examined the application of the human-machine spectrum (from fully-manual through teleoperated to independent-machine) to space tasks (Ref. 1-20). This study modified and extended the ARAMIS method, developing cost functions for various space tasks. So the THURIS method to select human-machine options includes relative cost estimates.

However, the THURIS cost functions are for individual tasks, and do not depend on the completion times for the tasks. The THURIS method is designed to identify the single most promising option for one task (or for a group of tasks) during the preliminary design process. It does not examine all the costs of a space scenario, and it is not intended to calculate the number of units of each option to assign to each task.

A study by the Jet Propulsion Laboratory has also developed a method for human-machine tradeoff analysis (Ref. 1-21). This method uses decision analysis techniques to develop multi-attribute figures of merit (e.g. combinations of cost, productivity, and safety). It also includes a tradeoff analysis between humans and various levels of automation at a Space Station, formulating a linear program to select optimal alternatives.

The JPL study resembles this study in some respects. Both have the overall objective of investigating the human-machine tradeoffs at permanent space facilities. And both model space tasks and human-machine alternatives to formulate the tradeoffs as an optimization problem.

However, the two studies are solving different problems. The JPL method considers activities at a well-defined near-term Space Station, and therefore assumes a fixed-size human crew. It develops a value for person-hours spent in Station activities, and maximizes "crew hours saved" by automating functions while staying under a cost target. The assumption of a fixed crew size and the use of decision analysis to quantify attributes of the human-machine alternatives allows the formulation of a linear objective function subject to linear constraints.

By comparison, this study considers future activities in space, and assumes a variable-size crew of humans and machines. Its method uses overall cost as the only figure of merit, but includes human and machine productivities directly in its cost equations. The objective to be minimized is the total cost of a particular job, and this objective function turns out to be fundamentally nonlinear.

CHAPTER 2:

THE GENERAL COST EQUATION

This chapter extracts the overall cost equation from the line-item cost model (presented in Appendix B) and reduces this cost equation to a general form. The behavior of the cost equation is examined, specifically the effect of the human and machine variables. The equation is nonlinear, and that nonlinearity makes the application of traditional optimization techniques difficult: several unsuccessful attempts are described. This leads to a reevaluation of the problem, recognizing the combinatorial aspects of the cost equation.

2.1 The Line-Item Cost Model:

The purpose of this study is to explore the economic tradeoffs between humans and machines in space activities. Therefore the study's first major task is to model the cost contributions of the human and machine options to the overall program cost. As described in Chapter 1, this is achieved through the formulation of a line-item cost model. This model has two purposes:

1) To generate a credible detailed cost equation for humans and machines in space activities;

2) To develop a physical understanding of the problem.

The line-item cost model is presented in detail in Appendix B, and briefly reviewed here.

2.1.1 Scope of Model:

The cost model focuses on activities at permanent space facilities in low-earth orbit, such as the planned Space Station. The human-machine tradeoffs in such facilities are significant design issues, currently under study throughout the U.S. space program.

The model concentrates specifically on in-orbit construction of structures and spacecraft. This takes advantage of the MIT Space Systems Lab's extensive experience in simulation of space construction by humans and teleoperators.

However, the model's form is sufficiently general that it could be adapted to a variety of space activities (e.g. operation of research and production facilities, or spacecraft servicing and repair). Most of the cost items in the model are not affected by the character of the tasks being performed. And the optimization techniques developed later in this study do not depend on the type of space activity being modeled.

2.1.2 Structure of the Cost Model:

The line-item cost model consists of a set of equations relating the physical parameters of the space activities (e.g. hardware masses, consumables requirements) to line-item costs; these cost items are then summed to obtain the overall program cost. The model itself is expressed entirely with variables; specific values for these parameters are provided to the model as numerical inputs.

The cost model considers a particular job to be done in space. For example, the in-orbit construction of a Microwave Radiometer is a job; this will be used as an illustrative case example in Chapter 4.

The job to be done is modeled as a set of tasks. For example, in the construction of a Radiometer, the assembly of truss structure might be "Task 1". Each task may be performed once or (more likely) many times. Therefore the size of each task is quantified by the number of repetitions of that task. In the assembly of truss structure, the assembly of one structural cell might be a "repetition" of the task.

2.1.3 Cost Model Inputs:

To describe the job to be done, the cost model receives task-related inputs, including the number of different tasks to be done, and the number of repetitions of each task. The number of repetitions of each task is listed in the array NREPS[j], where j is the index identifying the task.

The line-item cost model includes three basic human-machine options for space construction: humans (HUM), teleoperators (TEL), and automated equipment (AEQ). These options are separate because their cost behaviors are different. The physical characteristics of these human-machine options are also provided as model inputs.

These human and machine inputs include the numbers of humans, teleoperators, and units of automated equipment assigned to each task. They are expressed as the arrays Nhum[j], Ntel[j], and Naeq[j] respectively, where j indicates the task they are assigned to. The human and machine inputs also include equipment masses, consumables and maintenance needs, and ground support requirements.

The cost model also receives situation inputs, which describe parameters of the problem not tied to the choice of humans or machines for the tasks. Examples include cost factors for R&D, procurement, and launch, and requirements for general work-site support.

2.1.4 Productivities:

Some of the task-related and human-machine inputs overlap. These are the productivities of humans and machines in performing the various tasks. These productivities relate three physical factors with important effects on overall cost:

1) the size of the job to be done;
2) the numbers of humans and machines working on each task;
3) the amount of time required for each task.

The size of the job drives many of the research and development, procurement, and launch costs - for example, the costs of the space structure to be assembled in a construction scenario. The numbers of humans and machines at the worksite determine their procurement and launch costs. And the times to complete tasks define certain program expenses such as consumables, maintenance, and salary costs.

In the cost model, the triangular relationship between these factors is expressed as a productivity:

$$\frac{\text{(repetitions of the task)}}{\text{(number of humans or machines)} * \text{(time)}}$$

For example, if Task j is entirely performed by humans, the human productivity for Task j is:

$$\text{ODhum}[j] = \frac{\text{NREPS}[j]}{\text{Nhum}[j] * \text{Ttime}[j]}$$

where
ODhum[j] = Human productivity in Task j (reps/person-day)
NREPS[j] = Number of repetitons of Task j (reps)
Nhum[j] = Number of humans assigned to Task j (persons)
Ttime[j] = Time to complete Task j (days)
(For convenience, these definitions are repeated in Appendix A.)

In this study, it is assumed that the productivity relationship of a human or machine performing a task depends only on:

a) the design of the human-machine option;

b) the physical description of the task.

For example, if a machine has a productivity of 15 repetitions/machine-day in doing Task 1, this figure is determined uniquely by the design of the machine and by the physical description of Task 1.

Since the cost model is applied to specified jobs in space, the physical descriptions of the tasks are fixed. The model also receives specific human and machine designs, described in its human and machine inputs. Therefore the productivities of each option for each task are also provided as fixed inputs. Their values can be varied parametrically from case to case, but within each application of the cost model the productivities are constants.

2.1.5 Task Completion Times:

If a task is performed by a mix of humans, teleoperators, and automated equipment, the cost model calculates a "crew productivity" (in reps/day):

$$(ODhum[j] * Nhum[j]) + (ODtel[j] * Ntel[j]) + (ODaeq[j] * Naeq[j])$$

where ODtel[j] and ODaeq[j] are the teleoperator and automated equipment productivities in doing Task j. Then the time to complete Task j is (in days):

$$Ttime[j] = \frac{NREPS[j]}{(ODhum[j]*Nhum[j]) + (ODtel[j]*Ntel[j]) + (ODaeq[j]*Naeq[j])}$$

Therefore, if NREPS[j] is a fixed input to the model, and the productivities are fixed by the task descriptions and the human-machine designs, the task completion time Ttime[j] is function of the human and machine assignments Nhum[j], Ntel[j], and Naeq[j].

The cost model receives the numbers of task repetitions, the productivities, and the values of Nhum[j], Ntel[j], and Naeq[j] as explicit inputs. It can therefore use the relationship above to compute the completion times for each task, and the associated time-dependent costs.

The line-item cost model assumes that all space tasks start at the same time, and that the tasks are performed concurrently until individual tasks end. This is shown schematically in Figure 2.1. This "concurrent-tasks" assumption is important in this study, and its effect on the study's models and methods is discussed in Chapter 3.

Since the completion time for each task is a function of its human-machine assignments, which are entered as independent model inputs, the task completion times are also independent of each other. Therefore the tasks are finished at various times, as shown in Figure 2.1.

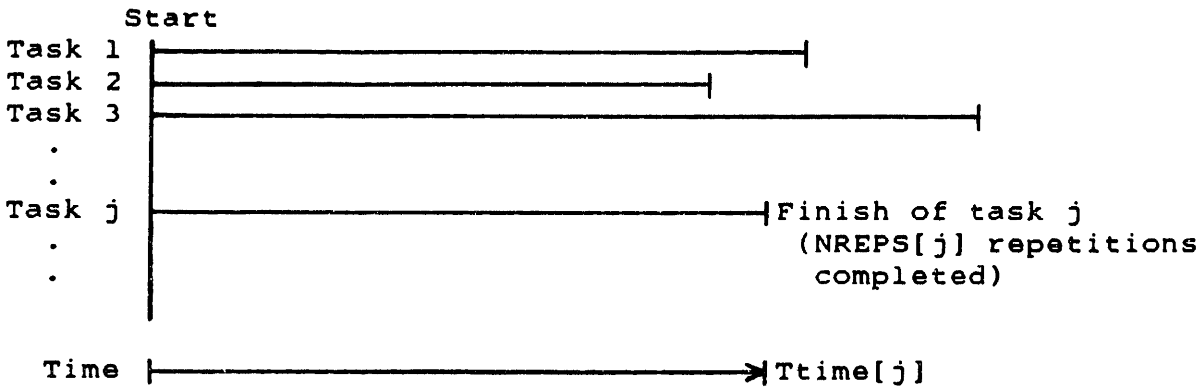


FIGURE 2.1: TASK TIMES IN THE LINE-ITEM COST MODEL

Therefore some of the crew (humans and/or machines) will run out of work as tasks are finished. For example, the humans assigned to Task 2 in the figure will be idle after Ttime[2] days. It is beyond the scope of this model to reassign idle humans and machines to other not-yet-completed tasks. Instead, the model assumes that idle crew members are immediately sent back to the ground and discharged (or reassigned to a different program at someone else's cost).

This assumption is discussed in Appendix B, section B.5.2. As will be shown in Chapter 3, for the minimum-total-cost solution, all the tasks should have the same completion time. So in the optimal case the issue of idle crew members is not relevant: everyone finishes their tasks at the same time, and the entire crew is reassigned or returns to the ground. Therefore this assumption is acceptable.

The whole-job completion time (Jobtime) is the completion time of the longest task (e.g. Ttime[3] in Figure 2.1). Some of the program costs depend on Jobtime, such as costs for station-keeping propellant and general worksite support.

2.1.6 Cost Elements:

The line item cost model breaks the program cost into four general categories: research and development (rd), procurement (pr), mission preparation and launch (la), and ground support (gs). These categories are further broken down into line-item cost elements. Table 2.1 lists these cost elements, including each element's variable name. A lexicon of abbreviations used in the variable names appears in Appendix A.

TABLE 2.1: COST ELEMENTS IN LINE-ITEM COST MODEL

Research & Development (CTrd):	
CTrdSST	R&D of space structure to be assembled
CTrdHAB	R&D of human habitation modules
CTrdHSE	R&D of human support equipment (e.g. EMU, MMU, and misc. tools)
CTrdTEL	R&D of teleoperators and their support facilities (e.g. racks, power systems, storage facilities for spares and consumables)
CTrdAEQ	R&D of automated equipment and support facilities (e.g. power systems, storage facilities for spares)
CTrdCSE	R&D of construction support equipment (low-tech equipment, e.g. construction jigs, work platforms)
CTrdTSK	R&D of task procedures (e.g. simulations)
CTrdGSE	R&D of ground support equipment
Procurement (CTpr):	
CTprSST	Proc. of space structure to be assembled
CTprHAB	Proc. of human habitation modules (including spares)
CTprHSE	Proc. of human support equipment (including spares)
CTprTEL	Proc. of teleoperators (including spares)
CTprAEQ	Proc. of automated equipment (including spares)
CTprCSE	Proc. of construction support equipment (including spares)
CTprHUMt	Cost of training humans in space crews
CTprHUMb	Salary and overhead for humans in space crews
CTprGSE	Proc. of ground support equipment (including spares)
Mission Preparation and Launch (CTla):	
CTlaSST	Mission prep. and launch of space structure to be assembled
CTlaHUM	Launch of humans in space crews
CTlaHAB	Mission prep. and launch of human habitation modules (including spares and consumables)
CTlaHSE	Mission prep. and launch of human support equipment (including spares and consumables)
CTlaTEL	Mission prep. and launch of teleoperators (including spares and consumables)
CTlaAEQ	Mission prep. and launch of automated equipment (including spares)
CTlaCSE	Mission prep. and launch of construction support equipment
CTlaCSP	Launch of construction site propellant
CTlaINS	Launch insurance
Ground Support (CTgs):	
CTgsGSCt	Ground support crew training
CTgsGSCb	Salary and overhead for ground support crew
CTgsTDR	User fees for tracking and data relay

2.1.7 Line-Item Cost Equation:

The line-item cost model is essentially an equation which relates its model inputs to the overall program cost. This complete cost equation, as implemented in the cost model, is presented in Appendix G, section G.1. It is a long equation (8 pages), part of which is reprinted in Table 2.2.

As this partial example shows, the cost equation is the sum of many line items. (Each item is identified in brackets in the right-hand column.) Each term in the equation is an algebraic combination of model inputs and (sometimes) intermediate results.

The model inputs are physical parameters of the space scenario. For example, DKrdH is the the R&D cost factor (in dollars/kilogram) for high-technology hardware; and KGunSSTh is the mass of one unit of high-tech space structure to be assembled (in kg/unit).

TABLE 2.2: PARTIAL EXAMPLE OF LINE-ITEM COST EQUATION

Total program cost ProgCost =

$$
\begin{aligned}
&\mathrm{DKrdH} * \mathrm{KGunSSTh} && \{\mathrm{CTrdSST}\}\\
&+ (\mathrm{DKrdH} * \mathrm{KGunSSTh} * \mathrm{NunSSTh} * 0.035)\\
&+ \mathrm{DKrdL} * \mathrm{KGunSSTl}\\
&+ (\mathrm{DKrdL} * \mathrm{KGunSSTl} * \mathrm{NunSSTl} * 0.035)\\
&+ ((\mathrm{DKrdH} * \mathrm{KGunHABh}) + (\mathrm{DKrdL} * \mathrm{KGunHABl})) * \mathrm{YSNOhum} && \{\mathrm{CTrdHAB}\}\\
&\quad\vdots\\
&+ (\mathrm{NtotHUMi} + (\mathrm{NtotPD} * \mathrm{HDhum} * \mathrm{WFmmu})) * \mathrm{KGunMMU} * \mathrm{DKprH}\\
& && \{\mathrm{CTprMMU}\}\\
&\quad\vdots
\end{aligned}
$$

$$\text{where } \mathrm{NtotHUMi} = \sum_{j=1}^{\mathrm{NTSK}} \mathrm{Nhum}[j]$$

$$\text{and where } \mathrm{NtotPD} = \sum_{j=1}^{\mathrm{NTSK}} \mathrm{Nhum}[j] * \mathrm{Ttime}[j]$$

Other quantities in the equation terms are intermediate results, calculated by the cost model from its inputs. For example, NtotHUMi is the total initial number of humans at the worksite, calculated by summing the humans assigned to each task (Nhum[j]) over all tasks. Similarly, NtotPD is the total number of person-days of work by humans, calculated by computing the person-days for each task and summing them over all tasks.

2.2 General Form of the Cost Equation:

The line-item cost equation includes the detailed contributions of the human and machine options to the overall program cost. However, this form of the equation is too complex to show clearly the relationships between individual variables and total cost.

The equation is therefore unsuitable for theoretical examination and for development of optimization methods. This section (2.2) presents the reorganization and compression of this line-item equation into a more useful "general cost equation".

2.2.1 Objective Function and Decision Variables:

The principal objective of this study is the development of an optimization method to be applied to the general cost equation. Therefore this equation should be in a form as suitable as possible for optimization. This suggests some useful directions in the rearrangement of the line-item cost equation.

In setting up a problem for optimization, the first step is to identify the objective, i.e. the quantity to be maximized or minimized. In this study, the objective is to minimize the overall program cost ProgCost. Therefore the cost equation is already set up to become the objective function.

The next step is to identify the decision variables, i.e. those variables whose values will be calculated to minimize the program cost. For optimization, the cost equation will be formulated with the decision variables as the only variables; the remaining quantities will be treated as constants. These "constants" can still be varied, but each change in their values will require a new optimization of the problem.

Since the primary objectives of the study involve the economic tradeoffs between humans and machines, the variables of interest are those which describe the human and machine options and their applications to the various tasks. Therefore the likely set of decision variables is:

1) Nhum[j], the number of humans assigned to each task j

2) Ntel[j], the number of teleoperators assigned to each task j

3) Naeq[j], the number of units of automated equipment assigned to each task j

Of course, other decision variables are possible, e.g. the productivities of the human-machine options. Each possible selection of decision variables would focus the study on a different fundamental problem; for example, productivities might be used as variables to optimize the physical design of human-machine options by finding optimum productivity levels. For the objectives of this study, however, the variables listed above are the most appropriate.

This study assumes that fractional values of humans or machines can be assigned to tasks (e.g. 3.7 humans or 4.1 teleoperators). These represent humans or machines who share their work time between two or more tasks. (This assumption is further discussed in Appendix B, section B.4.4). Mathematically, the effect of this assumption is that the decision variables are not constrained to integral values, which simplifies the optimization.

With the decision variables listed above, the optimization method will calculate the values of Nhum[j], Ntel[j], and Naeq[j] which minimize ProgCost. These values will indicate the appropriate human or machine options for each task: for example, if humans are appropriate for Task 1 and teleoperators are not, the optimization will drive Nhum[1] to a nonzero number and Ntel[1] to zero.

Since there are decision variables for each task j, this formulation allows the calculation of an optimum mix of humans and machines for the entire job, assigning some tasks to humans, some tasks to machines, and possibly some tasks to humans and machines together.

To formulate the problem with these decision variables, the cost equation must first be rewritten in terms of these variables, and all the terms in each decision variable collected together. For optimization, the ideal form of the resulting equation would be:

```
ProgCost = (constant 0)
           + (constant 1) * (decision variable 1)
           + ...
           + (constant n) * (decision variable n)
           + ...
```

In this case, the equation would be linear, with no coupling between the decision variables, and formal optimization methods (e.g. Linear Programming) could be used to minimize ProgCost.

Some of the terms in the cost equation can be rearranged into this form. For example, NtotHUMi in the equation fragment above can be expanded to:

$$Nhum[1] + Nhum[2] + \ldots + Nhum[j] + \ldots$$

and therefore products including NtotHUMi can be separated into functions of individual decision variables.

Some terms, however, are not so easily separated. For example, NtotPD in the equation fragment expands to:

$$\begin{aligned}&(Nhum[1] * Ttime[1])\\ &+ (Nhum[2] * Ttime[2])\\ &+ \ldots\\ &+ (Nhum[j] * Ttime[j])\\ &+ \ldots\end{aligned}$$

As described in section 2.1.5, the time required to complete each task, Ttime[j], is equal to:

$$\frac{NREPS[j]}{(ODhum[j]*Nhum[j]) + (ODtel[j]*Ntel[j]) + (ODaeq[j]*Naeq[j])}$$

Therefore each Ttime[j] term couples the decision variables Nhum[j], Ntel[j], and Naeq[j] together in its denominator. This creates nonlinear terms which cannot be broken apart cleanly.

As will be seen later in this chapter, this will lead to nonlinearites in the problem which will make optimization by traditional means impossible. This in turn will lead to the development of the study's Staged Optimization Method (in Chapter 3).

For the moment, however, this problem can be sidestepped by making the Ttime[j] terms themselves decision variables, so that they can be left unexpanded in the cost equation. The equation is still nonlinear, since it includes products of the decision variables. But its algebraic form is much clearer.

In summary, the <u>decision variables</u> chosen for the general cost equation are:

1) Nhum[j], the number of humans assigned to each task j
2) Ntel[j], the number of teleoperators assigned to each task j
3) Naeq[j], the number of units of automated equipment assigned to each task j
4) Ttime[j], the time required to complete each task j

2.2.2 Reduction of the Cost Equation:

The next step is the manipulation of the line-item cost equation to isolate and combine terms in the individual decision variables. This process is carried out in Appendix G. Section G.2 expands and rearranges the original line items. Section G.3 combines some of the terms within each category, compressing the overall cost equation down to 6 pages.

At this stage of the reduction, the equation terms are not yet in terms of the desired decision variables Nhum[j], Ntel[j], Naeq[j], and Ttime[j]. For conciseness, the equation keeps certain summation terms, such as:

NtotHUMi = summation of Nhum[j] over all j
NtotPD = summation of (Nhum[j] * Ttime[j]) over all j

Similar terms are used for teleoperators and automated equipment.

The resulting reduced form of the cost equation is presented in Table 2.3. This form displays some of the fundamental cost behavior of humans and machines in space activities. The program costs consist of four general types of terms:

1) Some constant terms, covering the R&D costs for whole-job hardware (e.g the structure to be assembled), for human-related equipment (e.g. habitation modules), for teleoperators, and for automated equipment. Also included are some whole-job procurement and launch costs (e.g. for the structure to be assembled). For the moment, it is assumed that all human-machine options (and therefore their R&D costs) are present; section 3.5.3 will discuss the effects of removing one or more options from the scenario.

2) For each human or machine option, some terms dependent on the total number of units of the option at the worksite, e.g. terms dependent on the number of humans NtotHUMi. In general, these terms cover the procurement and launch of initial space hardware, the procurement of initial ground support equipment, and the training of space and ground support crews.

3) For each human or machine option, some terms dependent on the total usage of the option at the worksite, e.g. terms dependent on the number of person-days NtotPD. These terms cover the procurement and launch of spares and consumables for hardware and crew, the crew rotation launches, the procurement of ground support equipment spares, the salaries of humans in space and on the ground, and the cost of communications to support the humans and machines in space.

4) Some terms dependent on the whole-job completion time Jobtime. These terms cover the procurement of spares for the general ground support equipment, the launch of construction site propellant, the salary of the general support ground crew, and the cost of the associated communications.

TABLE 2.3: REDUCED FORM OF THE COST EQUATION

ProgCost =

(Rw + Rh + Rt + Ra + Cw)	Constants
+ (learning curve terms dependent on NtotHUMi) + (constant 1) * (NtotHUMi)	Number-of- humans
+ (learning curve terms dependent on NtotPD) + (constant 2) * (NtotPD)	Human usage
+ (learning curve terms dependent on NtotTELi) + (constant 3) * (NtotTELi)	Number-of- teleop.
+ (learning curve terms dependent on NtotTD) + (constant 4) * (NtotTD)	Teleop. usage
+ (learning curve terms dependent on NtotAEQi) + (constant 5) * (NtotAEQi)	Number-of- auto. eq.
+ (learning curve terms dependent on NtotAD) + (constant 6) * (NtotAD)	Auto. eq. usage
+ (learning curve term dependent on Jobtime) + (constant 7) * (Jobtime)	Whole-job timecosts

This compressed cost equation (presented in section G.3) includes all the terms of the original cost equation, and therefore its behavior matches exactly the behavior of the line-item cost model. However, this equation is still complex, because it contains two types of nonlinearity: the task completion times, and learning curves, both discussed below.

2.2.3 Effect of Task Completion Times:

The task completion times Ttime[j] appear in the cost equation's usage terms, within the variables NtotPD, NtotTD, and NtotAD. In addition, the whole-job completion time Jobtime is equal to the longest Ttime[j]. As described in section 2.2.1, the Ttime[j] terms couple the decision variables Nhum[j], Ntel[j], and Naeq[j] together in their numerators, so that they cannot be cleanly separated into functions of single decision variables.

However, these terms cannot be ignored, because the time-dependent costs are an important part of the overall program costs. Thus the nonlinearity of the Ttime[j] and Jobtime terms is a fundamental part of the behavior of the cost equation. Any attempt to replace these terms with more tractable expressions runs the risk of losing the realism of the cost model. For the moment, therefore, these terms are kept unchanged in the formulation. (Section 2.4.4 will present an attempt at piecewise linearization of Ttime[j].)

2.2.4 Temporary Removal of Learning Curves:

The other source of nonlinearity is the learning curves in some of the procurement terms. These terms couple decision variables in expressions with fractional exponents. For example, the "learning curve terms dependent on NtotHUMi" in Table 2.3 have the form:

$$\text{(a constant)} * (1^{-q} + 2^{-q} + \ldots + \text{trunc}(\#\text{units})^{-q} + (\#\text{units} - \text{trunc}(\#\text{units})) * (\#\text{units})^{-q})$$

$$\text{where } (\#\text{units}) = \text{(another constant)} * (\text{NtotHUMi}) = \text{number of units to be procured}$$

When expanded, these learning curve series will include the term:

$$\text{NtotHUMi}^{-q} = (\text{Nhum}[1] + \text{Nhum}[2] + \ldots + \text{Nhum}[j] + \ldots)^{-q}$$

where $-q = -0.32193$ for an 80% learning curve. This type of nonlinearity is very difficult to deal with in formal optimization, and usually requires numerical methods. (A detailed discussion of learning curves appears in section B.7.2.)

However, if the learning curve parameter is set at 100% (no learning) then $-q = 0$, and the general learning curve form above reduces to:

$$\text{(a constant)} * (\#\text{units})$$

$$\text{where } (\#\text{units}) = \text{(another constant)} * (\text{NtotHUMi})$$

This is the case when each item is bought at a fixed price, with no economies of mass production.

Fortunately, as described in section G.4, these learning curves do not have major effects on the overall program costs in this study's examples. If the effect of Ttime[j] is a "first-order" nonlinearity, the learning curves could be called "second-order" nonlinearities.

Therefore, at this stage of the study, the learning curves in procurement are set at 100%, which makes their cost terms separable by decision variables. This will considerably simplify the examination of the cost equation in the remainder of this Chapter, and the theoretical development of optimization methods in Chapter 3.

This removal of the learning curves is temporary. As will be seen in Chapter 3, this study's Staged Optimization Method will be able to handle nonlinear cost functions, including those with learning curves. Some of this study's examples (in Chapter 4) will include such curves in the procurement terms.

In Appendix G, section G.4, the learning curves have been set to 100%, and their simplified terms have been combined with other terms where possible. This produces a cost equation of the form:

$$\begin{aligned}
\text{ProgCost} = {} & \\
& (\text{Rw} + \text{Rh} + \text{Rt} + \text{Ra} + \text{Cw}) \\
& + \text{Ch} * (\text{NtotHUMi}) \\
& + \text{Hh} * (\text{NtotPD}) \\
& + \text{Ct} * (\text{NtotTELi}) \\
& + \text{Ht} * (\text{NtotTD}) \\
& + \text{Ca} * (\text{NtotAEQi}) \\
& + \text{Ha} * (\text{NtotAD}) \\
& + \text{Hw} * (\text{Jobtime})
\end{aligned}$$

where:
Rw = Whole-job R&D costs (e.g. R&D of structure to be assembled)
Rh = R&D cost for human-related equipment (e.g. hab. modules)
Rt = R&D cost for teleoperators
Ra = R&D cost for automated equipment
Cw = Whole-job general costs (e.g. procurement of structure to be assembled)
Ch = Number-of-humans cost factor ($/person)
Hh = Human usage cost factor ($/person-day)
Ct = Number-of-teleoperators cost factor ($/teleoperator)
Ht = Teleoperator usage cost factor ($/tel-day)
Ca = Number-of-automated-equipment-units cost factor ($/aeq)
Ha = Automated equipment usage cost factor ($/aeq-day)
Hw = Whole-job timecost factor ($/day)

These terms are further defined in Appendix A. Expressions for each of these constants appear in Appendix G, section G.4.

2.2.5 General Cost Equation:

The resulting cost equation can be expressed in terms of the decision variables Nhum[j], Ntel[j], Naeq[j], and Ttime[j], by expanding the summation terms and clustering the terms by decision variables. This leads to the general cost equation presented in Table 2.4. It is this equation that the remainder of this chapter will examine and attempt to optimize.

2.3 Nonlinear Character of the General Cost Equation:

This section examines the fundamental behavior of the general cost equation. A simplified case is formulated, and its saddle-shaped cost surface is displayed in a contour plot. The effects of various parameters are presented in other plots, to show that this saddle-shape is a robust behavior, which must be dealt with in the optimization.

2.3.1 Development of a One-Task Two-Option Case:

The first step in the examination of Table 2.4's general cost equation is to formulate the simplest case of the equation that still yields useful insights. Therefore a restricted case is formulated, in which:

a) there is only one task to be performed in space;

b) there are only two options to perform the task: humans and teleoperators.

These restrictions have several effects on the general cost equation. The R&D, number-of-units, and usage terms for automated equipment are removed, since automated equipment is not an option. The cost terms for tasks 2,...,j,... do not appear, because there is only one task to be done. And the term Jobtime can be replaced by Ttime[1], since

$$\text{Jobtime} = \text{MAX}(\text{Ttime}[1]) = \text{Ttime}[1].$$

The resulting "restricted case cost equation" is shown in Table 2.5. For simplicity, the task suffix "[1]" has been omitted from Nhum[1], Ntel[1], and Ttime[1], since there is only one task to be costed.

This equation for ProgCost has three variables, Nhum, Ntel, and Ttime. As described in section 2.1.5, the task completion time is:

$$\text{Ttime} = \frac{\text{NREPS}}{(\text{ODhum} * \text{Nhum}) + (\text{ODtel} * \text{Ntel})}$$

TABLE 2.4: GENERAL FORM OF THE COST EQUATION (WITHOUT LEARNING CURVES)

Total program cost
ProgCost =

Rw	Whole-job R&D
+ Rh	Human-related R&D
+ Rt	Teleoperator-related R&D
+ Ra	Automated-equipment-related R&D
+ Cw	Whole-job general costs
+ Ch * Nhum[1]	Task 1 costs
+ Hh * (Nhum[1] * Ttime[1])	
+ Ct * Ntel[1]	
+ Ht * (Ntel[1] * Ttime[1])	
+ Ca * Naeq[1]	
+ Ha * (Naeq[1] * Ttime[1])	
+ Ch * Nhum[2]	Task 2 costs
+ Hh * (Nhum[2] * Ttime[2])	
+ Ct * Ntel[2]	
+ Ht * (Ntel[2] * Ttime[2])	
+ Ca * Naeq[2]	
+ Ha * (Naeq[2] * Ttime[2])	
.	
.	
.	
+ Ch * Nhum[j]	Task j costs
+ Hh * (Nhum[j] * Ttime[j])	
+ Ct * Ntel[j]	
+ Ht * (Ntel[j] * Ttime[j])	
+ Ca * Naeq[j]	
+ Ha * (Naeq[j] * Ttime[j])	
.	
.	
.	
+ Hw * Jobtime	Whole-job timecosts

where:
Rw, Rh, Rt, Ra, Cw, Ch, Hh, Ct, Ht, Ca, Ha, Hw = constants
Jobtime = MAX(Ttime[1], Ttime[2],..., Ttime[j],...)
(All of the equation's components are defined in Appendix A.)

TABLE 2.5: COST EQUATION FOR RESTRICTED CASE
(ONE TASK TO PERFORM, HUMANS AND TELEOPERATORS ONLY)

Total program cost ProgCost =

Rw	Whole-job R&D
+ Rh	Human-related R&D
+ Rt	Teleoperator-related R&D
+ Cw	Whole-job general costs
+ Ch * Nhum	Task costs
+ Hh * (Nhum * Ttime)	
+ Ct * Ntel	
+ Ht * (Ntel * Ttime)	
+ Hw * Ttime	Whole-job timecosts

where Rw, Rh, Rt, Cw, Ch, Hh, Ct, Ht, Hw = constants

Substituting this expression for Ttime produces an expression for ProgCost in terms of Nhum and Ntel only, shown in Table 2.6. The advantage to this restricted case cost equation is that ProgCost forms a two-dimensional surface over the Nhum-Ntel plane. This surface is displayed in the next section.

TABLE 2.6: COST EQUATION FOR RESTRICTED CASE, REARRANGED

$$\begin{aligned} \text{ProgCost} = \; & (Rw + Rh + Rt + Cw) \\ & + Ch * Nhum \\ & + Hh * NREPS * \frac{Nhum}{(ODhum * Nhum) + (ODtel * Ntel)} \\ & + Ct * Ntel \\ & + Ht * NREPS * \frac{Ntel}{(ODhum * Nhum) + (ODtel * Ntel)} \\ & + Hw * NREPS * \frac{1}{(ODhum * Nhum) + (ODtel * Ntel)} \end{aligned}$$

where:
Rw, Rh, Rt, Cw, Ch, Hh, Ct, Ht, Hw = constants
NREPS = Number of repetitions of the task (constant)
ODhum = Daily productivity of humans in doing the task (constant)
ODtel = Daily productivity of teleoperators in doing the task (constant)
Nhum = Number of humans assigned to the task
Ntel = Number of teleoperators assigned to the task

2.3.2 Behavior of the Restricted Case Cost Equation:

In the equation in Table 2.6, the constant term (Rw + Rh + Rt + Cw) covers whole-job R&D, human-related R&D, teleoperator-related R&D, and whole-job general costs. These do not include Nhum or Ntel, and therefore do not affect the tradeoff between the two variables.

The term (Ch * Nhum) covers number-of-humans costs, e.g. procurement and launch of habitation modules and human support equipment. The larger the human workforce, the larger is this cost. Similarly, the term (Ct * Ntel) covers number-of-teleoperators costs.

The term $$Hh * NREPS * \frac{Nhum}{(ODhum * Nhum) + (ODtel * Ntel)}$$

covers human usage costs, i.e. costs proportional to the number of person-days spent in space. These costs include consumables, salaries, crew rotation launches, hardware spares, and ground support for the humans.

This term's denominator ((ODhum * Nhum) + (ODtel * Ntel)) is the number of repetitions of the task performed each day by the entire crew of humans and teleoperators. How much of the task is done by humans and how much by teleoperators depends on their productivities ODhum and ODtel, and on the number of humans and teleoperators at the worksite. Therefore Ntel has an effect on the human usage cost, because it affects the quantity of the task to be done by humans: the more repetitions done by teleoperators, the fewer to be done by humans.

The term $$Ht * NREPS * \frac{Ntel}{(ODhum * Nhum) + (ODtel * Ntel)}$$

covers teleoperator usage costs (proportional to teleoperator-days in space), including teleoperator spares and consumables, and ground control of the teleoperators during construction. Similarly to the human usage term, the teleoperator usage cost is affected by Nhum.

The term $$Hw * NREPS * \frac{1}{(ODhum * Nhum) + (ODtel * Ntel)}$$

covers whole-job timecosts, such as worksite propellant and general ground support. The job completion time is inversely proportional to Nhum and Ntel: the more humans or teleoperators are assigned to the task, the sooner it is completed.

As mentioned in the previous section, ProgCost forms a surface over the Nhum-Ntel plane. The behavior of this cost surface can be displayed by a set of contour plots. Within each contour plot, the quantities Rw, Rh, Rt, Cw, Ch, Hh, Ct, Ht, Hw, NREPS, ODhum, and ODtel are held constant. From plot to plot, however, some of these quantities are varied parametrically, to examine their effects on the equation's overall behavior.

For the first contour plot, the quantities Rw, Rh, Rt, Cw, Ch, Hh, Ct, Ht, and Hw are given numerical values calculated from the expressions in section G.4, using model inputs from this study's case example (to be presented in Chapter 4). The quantities NREPS, ODhum, and ODtel are given the values of NREPS[1], ODhum[1], and ODtel[1] from the case example's Task 1.

The resulting "baseline" contour plot is shown in Figure 2.2. The axes of the plot (Nhum and Ntel) run from -2 to 7. The "feasible region" in this problem is the positive quadrant; there is no physical significance to negative humans or teleoperators. However, parts of the nonpositive quadrants are included, to show more clearly the mathematical behavior of the cost equation. It is assumed that both options (and their R&D costs) are present throughout the contour plot, i.e. there are no discontinuities as the contours cross the axes; the effects of completely removing an option from the scenario will be discussed with the study's optimization method (section 3.5.3).

The ProgCost surface is indicated by contour lines with values labeled along the edges of the graph. Each contour line is associated with a specific value of ProgCost, and therefore indicates those combinations of values of Nhum and Ntel which yield that particular total program cost.

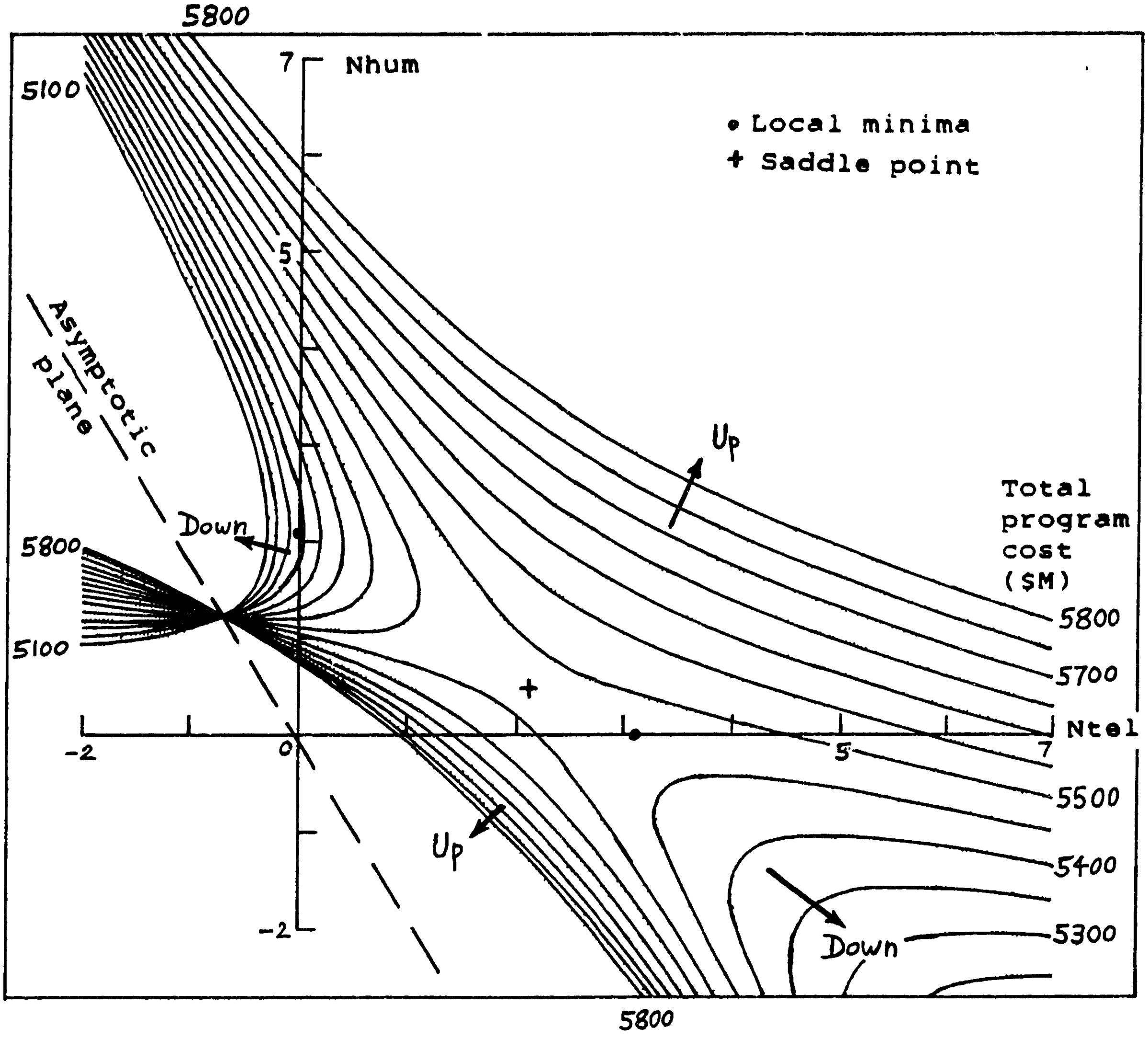
It is important to note that each line does <u>not</u> indicate a minimum cost for the program. If the values of Nhum and Ntel are well chosen, ProgCost will be on a low contour, corresponding to a low program cost; other choices for Nhum and Ntel may put ProgCost on a much higher contour.

Figure 2.2 shows that the overall cost surface is a <u>saddle shape</u>, with two upward slopes and two downward troughs, and a saddle point at roughly (Ntel=2.1, Nhum=0.6). Several details of the figure give useful insights on the behavior of the cost equation.

The first observation is that each contour value corresponds to <u>two curves</u>. For example, the 5800-$million contour consists of two lines: one that curves around near the origin, and another which crosses the positive quadrant further away. This indicates that there are two ways to make ProgCost equal to 5800 $million:

a) Keep Nhum and Ntel small and take a long time to complete the job. Then the total cost is dominated by whole-job timecosts. This produces the curve near the origin.

b) Make Nhum and Ntel large and take a short time to complete the job. Then ProgCost is dominated by initial hardware costs. This produces the curve further away from the origin.



Rw = 437.403 $M	Ch = 237.800 $M/pers	Hw = 0.569 $M/day
Rh = 1574.313 $M	Hh = 0.439 $M/pers-day	NREPS = 16,688 reps
Rt = 168.430 $M	Ct = 60.190 $M/tel	ODhum = 9.1 reps/pers-day
Cw = 1256.669 $M	Ht = 1.592 $M/tel-day	ODtel = 16.0 reps/tel-day

FIGURE 2.2: CONTOUR PLOT OF ProgCost FOR RESTRICTED CASE
(COST EQUATION CONSTANTS AT BASELINE VALUES)

As the contour values get lower (e.g. 5500 $million), the two curves approach each other toward the saddle point. Below the saddle point value (roughly 5480 $million), each contour still consists of two lines, but now one curve loops near the Nhum axis, and the other loops near the Ntel axis. For those contours, each value of Progcost can be achieved either with a large Nhum and small Ntel, or with a small Nhum and large Ntel.

A second observation concerns the locations and directions of the two upward slopes. The first slope is near the origin: as the values of Nhum and Ntel approach zero, the contours climb steeply upward. Physically, this corresponds to very small workcrews, and therefore very long times for completion. Therefore ProgCost is dominated by the whole-job timecost term:

$$Hw * NREPS * \frac{1}{(ODhum * Nhum) + (ODtel * Ntel)}$$

whose denominator goes to zero as Nhum and Ntel approach the origin. Mathematically, the contour lines climb up an asymptotic plane which is normal to the Nhum-Ntel plane along the line:

$$(ODhum * Nhum) + (ODtel * Ntel) = 0$$

as shown by the dashed line in Figure 2.2. Physically, this plane represents workcrews (humans and teleoperators) with daily productivities of zero. The contours cross through this plane when this zero denominator is neutralized by zero numerators, i.e. when:

$$(Hh * NREPS * Nhum) + (Ht * NREPS * Ntel) + (Hw * NREPS) = 0$$

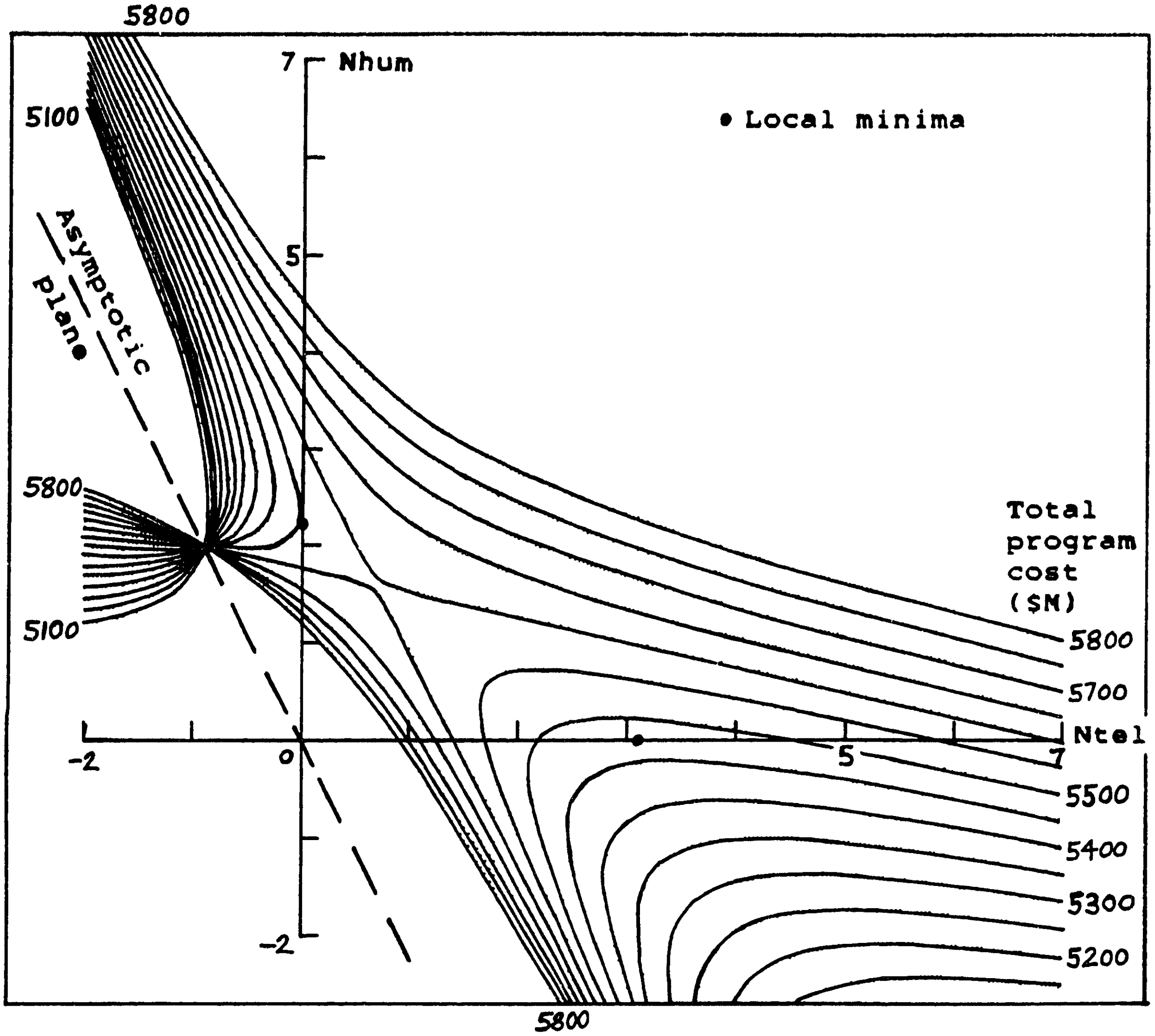
Since NREPS, Hh, Ht, Hw > 0, then either Nhum or Ntel must be negative (and the other positive) at this cross-through point. Therefore this point must lie outside the feasible region.

The other upward slope occurs in the positive quadrant, away from the origin (i.e. for Nhum > 2, Ntel > 3). In this region the size of the workcrew is large, and therefore the completion time is short. So ProgCost is dominated by initial hardware costs in its number-of-humans term (Ch * Nhum) and its number-of-teleoperators term (Ct * Ntel). Therefore, as Nhum and Ntel become large, the cost surface approaches a planar upward slope.

A third observation is that between the two upward slopes there are two downward troughs, each extending down from the saddle point. In Figure 2.2, the saddle point is inside the positive quadrant, and therefore part of each trough is in the feasible region. Therefore the minimum feasible value of ProgCost can occur down either one of the troughs.

In this baseline case, the contours indicate that the minimum feasible value of ProgCost is just under 5250 $million, at (Nhum = 2.1, Ntel = 0). The other trough's minimum is at (Nhum = 0, Ntel = 3.1) and has value 5475 $M, 225 $million higher.

The shape of the cost surface in Figure 2.2 is affected by variations in the parameters of the cost equation. Figure 2.3 shows the effect of reducing the humans' daily productivity by 20% to 7.3 reps/pers-day (still well within the range estimated in section C.4.2). Other parameters are left at their baseline values.



Rw = 437.403 $M	Ch = 237.800 $M/pers	Hw = 0.569 $M/day
Rh = 1574.313 $M	Hh = 0.439 $M/pers-day	NREPS = 16,688 reps
Rt = 168.430 $M	Ct = 60.190 $M/tel	ODhum = 7.3 reps/pers-day
Cw = 1256.669 $M	Ht = 1.592 $M/tel-day	ODtel = 16.0 reps/tel-day

FIGURE 2.3: CONTOUR PLOT OF ProgCost (LOWER HUMAN PRODUCTIVITY)

The resulting cost surface is still saddle-shaped, but the saddle point has been shifted upward and to the left. All of the contours for (Nhum > 0) have shifted, and the lines for 5500 and 5550 $M have gone through the saddle point to become part of the trough contours. The saddle point value has climbed near 5600 $M (from the baseline near 5480 $M). The asymptotic plane has also tilted slightly, because its angle is set by the ratio of ODhum and ODtel.

Physically, the change in ODhum has made humans relatively more expensive: (Nhum = 2.1, Ntel = 0) now costs roughly 5560 $million, roughly 310 $million more than the baseline case. Graphically, the cost surface has been pushed up proportionally to Nhum, compressing the contours toward and into the Nhum trough.

Along the Ntel axis, however, the change in ODhum has no effect, and therefore the contours cross the Ntel axis at the same values as in Figure 2.2. This can be checked by superimposing the two figures.

Because the human option is more expensive, the Nhum trough in Figure 2.3 has a minimum feasible ProgCost of roughly 5555 $M, at (Nhum = 2.4). The Ntel trough still has its baseline minimum of 5475 $million at (Ntel = 3.1), which is now the overall feasible minimum.

Figure 2.3 has shown the result of a decrease in human productivity ODhum. A similar decrease in the teleoperator productivity ODtel would have symmetric effects on the contour plot, moving the saddle point downward and to the right, and increasing the values of the Ntel trough contours.

Thus the principal effect of variations in productivities is to shift the cost surface's saddle point, roughly along the (-1) slope line. This changes the depths of the troughs in the feasible region, which can shift the overall minimum from one trough to the other. A larger change in productivity can move the saddle point entirely out of the positive quadrant, leaving only one trough in the feasible region.

Appendix J presents three more contour plots, showing the effects of variations in the number-of-humans parameter Ch, the human usage parameter Hh, and the whole-job timecost parameter Hw. Similarly to the productivities, variations in the teleoperator parameters Ct and Ht would have symmetric results in the contour plots.

To summarize all the contour plots, the cost surface is a saddle shape, with two local minima. Which of these minima is the global minimum depends on the values of the parameters ODhum, Ch, Hh, ODtel, Ct, Ht, and Hw. The strongest effect on the selection of the overall minimum is from variations in productivity: in the example above, a 20% decrease in ODhum (relative to baseline) switches the global minimum from the Nhum trough to the Ntel trough.

The human usage factor Hh requires a 50% increase in its value to achieve a similar result. The number-of-humans factor Ch has a lesser effect: a 50% increase makes the two trough minima roughly equal. The weakest effect is from the whole-job timecost factor Hw: a 100% increase makes the two minima roughly equal; however, that increase has a strong effect on overall cost, pushing up the overall minimum nearly 400 $million over baseline.

2.3.3 Comments on Cost Equation Behavior:

The contour plots also show that the cost equation's saddle shape is a robust behavior: the variations in parameters move the cost surface over the Nhum-Ntel plane, but do not change its basic saddle shape. Therefore any optimization method to find the feasible minimum on this surface must deal with that nonlinear shape. In particular, it must be able to select between the two potential minima in the two downward troughs.

The overall minimum is easily identifiable in the restricted example's contour plots. However, if automated equipment were added as another option, ProgCost would become a multi-dimensional surface in the Nhum-Ntel-Naeq space, with three lower-cost troughs. If more tasks were included, each additional task would then add three more dimensions to the cost domain. Therefore contour plots are not useful for mapping costs in realistic multi-task multi-option jobs.

The minima in the contour plots above are constraint minima rather than functional minima. In other words, they occur at the boundaries of the feasible region, and the slope of the cost surface is nonzero at the minimum. In fact, the saddle shape has no zero-slope minimum anywhere, even outside the positive quadrant; the only stationary point is the saddle point. As will be shown in the following sections, this complicates the optimization.

Physically, since the contour plots' trough minima are constraint minima, the overall minimum is either an all-human option or an all-teleoperator option. This is a logical result for the one-task restricted case: either humans or teleoperators are the cheapest option to perform the task. This characteristic will be important in the later reevaluation of the problem, and in the development of the Staged Optimization Method.

2.4 Attempts at Formal Optimization:

The study's principal objective is the development of an optimization method to minimize the overall cost of a space job by calculating the optimal number of humans and machines assigned to each task. The selected approach was to optimize the one-task two-option case first; the technique developed for this case would then be expanded to more realistic multi-task multi-option cases. This section describes attempts to find or adapt existing formal optimization methods to the one-task two-option case.

Therefore, at this stage of the study, the equation to be optimized is the restricted case cost equation of Table 2.6, repeated for convenience in Table 2.7. Its two decision variables are Nhum and Ntel. The equation includes terms proportional to the decision variables, terms with decision variables in both numerator and denominator, and one term with decision variables in the denominator only.

TABLE 2.7: BASIC OPTIMIZATION PROBLEM

Minimize: ProgCost

$$\begin{aligned}
&= (Rw + Rh + Rt + Cw)\\
&\quad + Ch * Nhum\\
&\quad + Hh * NREPS * \frac{Nhum}{(ODhum * Nhum) + (ODtel * Ntel)}\\
&\quad + Ct * Ntel\\
&\quad + Ht * NREPS * \frac{Ntel}{(ODhum * Nhum) + (ODtel * Ntel)}\\
&\quad + Hw * NREPS * \frac{1}{(ODhum * Nhum) + (ODtel * Ntel)}
\end{aligned}$$

subject to:

$$Nhum \geq 0 \quad \text{and} \quad Ntel \geq 0$$

where:
Rw, Rh, Rt, Cw, Ch, Hh, Ct, Ht, Hw = constants
NREPS = Number of repetitions of the task (constant)
ODhum = Daily productivity of humans in doing the task (constant)
ODtel = Daily productivity of teleoperators in doing the task (constant)
Nhum = Number of humans assigned to the task
Ntel = Number of teleoperators assigned to the task

As the first step, some literature search and consultation was done, to check whether this particular form of equation had already been formally optimized. The search yielded several review papers on nonlinear optimization techniques (e.g. Ref. 2-1), including general-purpose methods such as linear decomposition (Ref. 2-2) and gradient-search techniques (Refs. 2-3 and 2-4). Also reviewed were sensitivity analysis methods (Ref. 2-5) and optimal scheduling techniques (Ref. 2-6). Some references within these papers were checked as well.

Although useful in describing some of the state-of-the-art in nonlinear optimization and scheduling, these sources did not include any methods specifically applicable to the equation in Table 2.7. The first consultations had similar results: the experts were unaware of any existing technique for the formal optimization of this equation (Refs. 2-7 and 2-8).

2.4.1 Method of Fractional Functionals:

However, another consultation (Ref. 2-9) suggested a textbook which includes a problem on "linear fractional programming" (Ref. 2-10, Chapter 13, exercise 27). That exercise's objective function resembles the fractional terms in Table 2.7, and the chapter's acknowledgments includes two associated references on fractional programming (Refs. 2-11 and 2-12).

Charnes and Cooper (Ref. 2-11) deal with "linear fractional functionals". Their method considers objective functions of the form:

$$\text{Maximize} \quad \frac{c_1 x_1 + c_2 x_2 + \ldots + c_n x_n + r}{d_1 x_1 + d_2 x_2 + \ldots + d_n x_n + v}$$

subject to m constraints:

$$a_{11} x_1 + a_{12} x_2 + \ldots + a_{1n} x_n \leq b_1$$

$$a_{21} x_1 + a_{22} x_2 + \ldots + a_{2n} x_n \leq b_2$$

$$\vdots$$

$$a_{m1} x_1 + a_{m2} x_2 + \ldots + a_{mn} x_n \leq b_m$$

and to $x_k \geq 0 \qquad (k=1,2,\ldots,n)$.

where x's are the decision variables, and c_k, d_k, r, v, and a_{hk} are constants.

Using the substitution $\quad x_k = \dfrac{y_k}{y_0}$

where $y_0 \geq 0$ is chosen such that:

$$d_1 y_1 + d_2 y_2 + \ldots + d_n y_n + v y_0 = G$$

(where G = a specified nonzero number)

and where $y_k \geq 0$ $(k=0,1,2,\ldots,n)$,

the problem transforms into:

Maximize $c_1 y_1 + c_2 y_2 + \ldots + c_n y_n + r y_0$

subject to m constraints:

$$a_{11} y_1 + a_{12} y_2 + \ldots + a_{1n} y_n - b_1 y_0 \leq 0$$

$$a_{21} y_1 + a_{22} y_2 + \ldots + a_{2n} y_n - b_2 y_0 \leq 0$$

$$\vdots$$

$$a_{m1} y_1 + a_{m2} y_2 + \ldots + a_{mn} y_n - b_m y_0 \leq 0$$

and to:

$$d_1 y_1 + d_2 y_2 + \ldots + d_n y_n + v y_0 = G$$

and to $y_k \geq 0$ $(k=0,1,2,\ldots,n)$.

This new problem can be handled by Linear Programming, and Ref. 2-11 proves that the optimal y_k values can be transformed back into the optimal x_k by reversing the substitutions.

This method was attempted on the optimization problem in Table 2.7. First, the problem was turned into a maximization problem by multiplying the objective function by (-1). Next, the fractional terms in the cost equation were combined, the terms were rearranged, and simpler constants were substituted for products of constants. The resulting problem is shown in Table 2.8.

TABLE 2.8: BASIC OPTIMIZATION PROBLEM, REARRANGED

Maximize (-ProgCost)

$$= \frac{(C1 * Nhum) + (C2 * Ntel) + R}{(ODhum * Nhum) + (ODtel * Ntel)}$$

$$+ Cg$$

$$- (Ch * Nhum)$$

$$- (Ct * Ntel)$$

subject to:

$$Nhum \geq 0 \quad \text{and} \quad Ntel \geq 0$$

where

$Cg = -(Rw + Rh + Rt + Cw)$

$C1 = -(Hh * NREPS)$

$C2 = -(Ht * NREPS)$

$R = -(Hw * NREPS)$

The next step is to apply the substitutions:

$$Nhum = \frac{Yhum}{Yzero} \quad \text{and} \quad Ntel = \frac{Ytel}{Yzero}$$

where Yzero $\geq$ 0 is chosen such that

$$(ODhum * Yhum) + (ODtel * Ytel) = G$$

where G = a specified nonzero number.

The resulting optimization problem is:

Maximize (-ProgCost)

$$= (\frac{C1}{G} * Yhum) + (\frac{C2}{G} * Ytel) + (\frac{R}{G} * Yzero)$$

$$+ Cg$$

$$- (Ch * \frac{Yhum}{Yzero})$$

$$- (Ct * \frac{Ytel}{Yzero})$$

subject to:

$$(ODhum * Yhum) + (ODtel * Ytel) = G$$

$$Yhum, Ytel, Yzero \geq 0$$

The substitutions have indeed turned the fractional part of the objective function into a linear expression, at the cost of an extra decision variable (Yzero) and an additional linear constraint. However, the terms in the equation that were proportional to the decision variables have now become fractional, so the new problem cannot be solved by Linear Programming. If the method is applied again, the now-fractional terms will become linear, but the linear terms will once again become fractional.

Therefore Charnes' and Cooper's method cannot be applied to objective functions which include both fractional and proportional terms. However, Bradley and Frey have extended this method to more general homogeneous functions (Ref. 2-12). They consider problems of the form:

$$\text{Maximize} \quad \frac{c(x) + r}{d(x) + v}$$

$$\text{subject to:} \quad A(x) \geq 0$$

$$s(x) = 1$$

where $x = (x_1, x_2, \ldots, x_n)$ and c(x), d(x), s(x), and A(x) are continuous functions. In addition, c, d, s, and A must be homogeneous of degree one, e.g.:

$$c(Q * x) = Q * c(x) \qquad (\text{for } Q > 0)$$

This method also transforms the original problem into a related Linear Programming problem, but the substitutions involve the s function in their definitions.

To apply Bradley's and Frey's method to the problem in Table 2.8, all of the terms in the objective function are first combined into one fractional term. Substituting simpler constants for products of constants, the resulting problem is shown in Table 2.9.

The needed s(x) = 1 condition can be provided by adding a decision variable Nadd = 1. The objective function's denominator (ODhum * Nhum) + (ODtel * Ntel) is continuous and homogeneous. However, the c(x) part of the numerator is continuous but not homogeneous:

$$c(Q * x) =$$

$$Q^2 * ((K1 * Nhum^2) + (K3 * (Nhum * Ntel)) + (K5 * Ntel^2))$$

$$+ Q * ((K2 * Nhum) + (K4 * Ntel)$$

$$\neq Q * c(x)$$

TABLE 2.9: FRACTIONAL OPTIMIZATION PROBLEM

Maximize (-ProgCost) =

$$\frac{(K1 * Nhum^{2}) + (K2 * Nhum) + (K3 * (Nhum * Ntel)) + (K4 * Ntel) + (K5 * Ntel^{2}) + R}{(ODhum * Nhum) + (ODtel * Ntel)}$$

subject to:

$$Nhum \geq 0 \quad \text{and} \quad Ntel \geq 0$$

where
K1 = -(Ch * ODhum)
K2 = -(Hh * NREPS) - ((Rw + Rh + Rt + Cw) * ODhum)
K3 = -(Ch * ODtel + Ct * ODhum)
K4 = -(Ht * NREPS) - ((Rw + Rh + Rt + Cw) * ODtel)
K5 = -(Ct * ODtel)
R = -(Hw * NREPS)

Unfortunately, the homogeneity of c(x) is required to prove that solving the transformed problem also solves the original problem. Therefore Bradley's and Frey's method cannot be applied to this study's problem because its objective function includes a quadratic numerator.

2.4.2 Stationary Points in the Cost Surface:

Since a direct formal optimization technique was not available, the study then tried to adapt existing methods to the problem in Table 2.7. The next technique to be considered was the identification of stationary points in the cost surface (Ref. 2-13).

Stationary points include saddle points and local functional minima and maxima. If the cost surface ProgCost is a function of two independent variables Nhum and Ntel, then such stationary points occur when the partial derivatives of ProgCost with respect to Nhum and Ntel are equal to zero:

$$\frac{d\ (ProgCost)}{d\ (Nhum)} = 0 \qquad \text{and} \qquad \frac{d\ (ProgCost)}{d\ (Ntel)} = 0$$

Since the cost equation in Table 2.7 is continuously differentiable over the feasible region, these partial derivatives provide two simultaneous equations which can be solved to locate the stationary points.

If the restricted case is expanded to include more tasks and more options, ProgCost will be a function of more variables. If these variables are independent, the partial derivatives can be set equal to zero to identify stationary points. If the variables are not independent (e.g. some of them are related through constraint equations), then the method of Lagrange Multipliers (Ref. 2-13) can be used to find the stationary points.

Unfortunately, finding the stationary points is not sufficient to minimize ProgCost. As shown in the contour plots in section 2.3.2, the restricted case's feasible minima are constraint minima rather than functional minima. Therefore they are not stationary points: at the minima, one or more of the partial derivatives have nonzero values.

This can be partially remedied by adding penalty functions to the cost equation. For example, in the restricted case, the critical constraints ($Nhum \geq 0$) and ($Ntel \geq 0$) can be represented by adding the terms $\frac{R1}{Nhum}$ and $\frac{R2}{Ntel}$ to the equation for ProgCost.

If R1 and R2 are small (e.g. R1 = 0.001), these terms add steep upward slopes as the cost surface approaches either the Nhum or Ntel axis. Then the two troughs in the contour plots no longer extend downward out of the feasible region; instead each trough becomes bathtub-shaped, with a local minimum at the bottom. These local minima are stationary points, and they are near the original constraint minima: the smaller the value of R1 or R2, the closer the stationary point is to the true minimum.

Therefore this technique could be used to find approximate coordinates for the feasible minima. However, there are three problems. First, there must be a rational method to choose the values of R1 and R2: low enough to make the stationary points close to the original minima, but high enough to avoid computational problems. In the restricted case example, suitable values can be selected by examining the contour plots. In more complex cases, however, the actual shape of the cost surface will not be displayable.

Second, this technique will identify three stationary points in the restricted case's cost surface: the saddle point and two local minima. Further checks will be required to identify the minima, and to decide which of them is the global minimum. In multi-task multi-option cases, there will be many more stationary points requiring such evaluations.

Third, even after the inclusion of the proper penalty functions and the identification of the global minimum, the result is still only an approximation to the true minimum, which is on a nearby constraint line. To find accurate coordinates for that minimum, additional analysis is required.

Therefore the use of stationary points to find an approximate minimum involves some serious problems in implementation, requiring considerable additional computation to produce useful results. And if learning curves are included in the cost equation, this technique is not suitable, because the learning curve terms are not differentiable. Therefore this study considers alternative methods to minimize the cost equation.

2.4.3 Separation of Variables:

An optimization problem is called separable if its objective function and constraints are sums of functions of individual decision variables:

$$\text{Max (or Min)} \quad f_1(x_1) + f_2(x_2) + \ldots + f_n(x_n)$$

subject to:

$$g_{11}(x_1) + g_{12}(x_2) + \ldots + g_{1n}(x_n) \leq b_1$$

$$g_{21}(x_1) + g_{22}(x_2) + \ldots + g_{2n}(x_n) \leq b_2$$

$$\vdots$$

$$g_{m1}(x_1) + g_{m2}(x_2) + \ldots + g_{mn}(x_n) \leq b_m$$

The decision variables appear separately, one in each function f_k in the objective function, and one in each function g_{ik} in the constraints. These functions f_k and g_{ik} can be nonlinear.

If a problem can be cast in this form, its nonlinear functions can be replaced by linear approximations, and the resulting problem can be solved with Linear Programming. There are a variety of techniques for piecewise-linearization of the functions, such as "inner" linearization by the Lambda Method or by Generalized Programming, or "outer" linearization by the Frank-Wolfe Algorithm or by the Method of Approximation Programming (Ref. 2-10).

The optimization problem in Table 2.7 can be put into the form above through a series of variable substitutions. Each substitution adds a constraint to the problem. First, the denominator of the fractional terms is replaced by:

Y1 = (ODhum * Nhum) + (ODtel * Ntel)

Since ODhum, Nhum, ODtel, Ntel ≥ 0, Y1 is also ≥ 0. In fact, if Y1 = 0, ProgCost becomes infinite, which clearly cannot be a solution to the minimization problem. So Y1 must be > 0 for a feasible solution. Also, the productivities ODhum and ODtel and the numbers Nhum and Ntel must have finite values because of physical restrictions on the problem. Therefore Y1 < infinity.

The optimization problem now becomes:

Minimize: ProgCost

$$= (Rw + Rh + Rt + Cw) + Ch * Nhum + Hh * NREPS * \frac{Nhum}{Y1} + Ct * Ntel + Ht * NREPS * \frac{Ntel}{Y1} + Hw * NREPS * \frac{1}{Y1}$$

subject to:

$$Y1 - (ODhum * Nhum) - (ODtel * Ntel) = 0$$

$$0 < Y1 < (infinity)$$

$$Nhum, Ntel \geq 0$$

The second substitution replaces the fractional terms in the objective function by multiplicative terms, at the cost of a fractional constraint. The variable substitution is:

$$Y2 = \frac{1}{Y1}$$

where 0 < Y2 < (infinity). Physically, Y2 is the time for the humans and teleoperators to perform one repetition of the task, so it is logical to expect a nonzero finite value for Y2 in a feasible solution. The problem becomes:

Minimize: ProgCost

$$= (Rw + Rh + Rt + Cw) + Ch * Nhum$$

(continued)

$$+ Hh * NREPS * Nhum * Y2$$

$$+ Ct * Ntel$$

$$+ Ht * NREPS * Ntel * Y2$$

$$+ Hw * NREPS * Y2$$

subject to:

$$Y1 - (ODhum * Nhum) - (ODtel * Ntel) = 0$$

$$Y2 - \frac{1}{Y1} = 0$$

$$Y1, Y2 > 0$$

$$Nhum, Ntel \geq 0$$

To separate the products (Nhum * Y2) and (Ntel * Y2), four more variables are introduced:

$$Z1 = (0.5) * (Nhum + Y2)$$

$$Z2 = (0.5) * (Nhum - Y2)$$

$$Z3 = (0.5) * (Ntel + Y2)$$

$$Z4 = (0.5) * (Ntel - Y2)$$

So $(Nhum * Y2) = (Z1^2 - Z2^2)$ and $(Ntel * Y2) = (Z3^2 - Z4^2)$. Also substituting simpler constants for products of constants, the resulting problem is shown in Table 2.10.

The problem is now in the desired form, with the variables separated into individual terms within the objective function and constraint equations. Some of these terms are nonlinear: the Z-variable terms in the objective, and the $\frac{1}{Y1}$ term in one constraint. To solve the problem in this form, two general approaches are possible.

The first alternative is Quadratic Programming (Ref. 2-10). This technique can be applied to optimization problems with quadratic objective functions but with linear constraints. The advantage to this technique is that it could be applied to the problem in Table 2.10 without further work on the already quadratic objective function.

TABLE 2.10: OPTIMIZATION PROBLEM WITH SEPARATED VARIABLES

Minimize:

$$\begin{aligned} \text{ProgCost} = \; & Kg \\ & + Ch * Nhum \quad + Kh * Z1^{2} \quad - Kh * Z2^{2} \\ & + Ct * Ntel \quad + Kt * Z3^{2} \quad - Kt * Z4^{2} \\ & + Kw * Y2 \end{aligned}$$

subject to:

$$Y1 - (ODhum * Nhum) - (ODtel * Ntel) = 0$$

$$Z1 - (0.5 * Nhum) - (0.5 * Y2) = 0$$

$$Z2 - (0.5 * Nhum) + (0.5 * Y2) = 0$$

$$Z3 - (0.5 * Ntel) - (0.5 * Y2) = 0$$

$$Z4 - (0.5 * Ntel) + (0.5 * Y2) = 0$$

$$Y2 - \frac{1}{Y1} = 0$$

$$Y1, Y2 > 0$$

$$Nhum, Ntel \geq 0$$

where

$Kg = (Rw + Rh + Rt + Cw)$ $\qquad$ $Kt = (Ht * NREPS)$

$Kh = (Hh * NREPS)$ $\qquad$ $Kw = (Hw * NREPS)$

However, quadratic programming also has some disadvantages. First, it requires linear constraints, and therefore the equation

$$Y2 - \frac{1}{Y1} = 0$$

would have to be replaced by a linear approximation. This will be discussed in detail in the next section.

Second, quadratic programming cannot differentiate between a local or global maximum, and may therefore be trapped into a less-than-optimal answer. The exception is for strictly convex objective functions, which always lead to global minima. Therefore the application of quadratic programming to this study's problem would require, for each case, either a proof that ProgCost is strictly convex, or some additional computations to locate and compare all the local minima.

The second possible approach is linear approximation of all the nonlinear terms in the optimization problem, leading to a solution by Linear Programming (LP). In effect, this method builds a piecewise-linear model of the original problem's nonlinear objective function and feasible region, then solves for the optimum using linear techniques.

The advantage to this method is that it is robust, i.e. it will find the global minimum in its domain, using efficient LP techniques. However, because this domain is only a linear approximation to the original problem's domain, the result will be an approximation to the true minimum. How close the two will be depends on how accurately the linear approximation models the original problem, and that accuracy can cost heavily in computation.

Both the quadratic and linear programming approaches mentioned above require the linear approximation of the nonlinear constraint in Table 2.10. The following section presents this approximation, and discusses its impact on these formal optimization techniques.

2.4.4 Piecewise Linear Approximation:

The piecewise-linear approximation method presented in this section is the Lambda Method (Ref. 2-10). This technique is used to replace the constraint

$$Y2 - \frac{1}{Y1} = 0$$

by a set of straight-line segments. This process is shown schematically in Figure 2.4.

The first step is the selection of a number of "break points" on the curve to be linearized. The number and location of these points depends on the geometry of the curve, and on the need for accuracy in the approximation. The break points are connected by straight-line segments. It is the equations for these segments that will replace the nonlinear equation in the problem.

The total number of break points is called Npts, and i is the index identifying each point. Each break point has a Y1-coordinate YONEi and a Y2-coordinate YTWOi. For example, in Figure 2.4, Npts = 4, and two break points of interest (with index numbers 2 and 3) have coordinates (YONE2, YTWO2) and (YONE3, YTWO3), respectively.

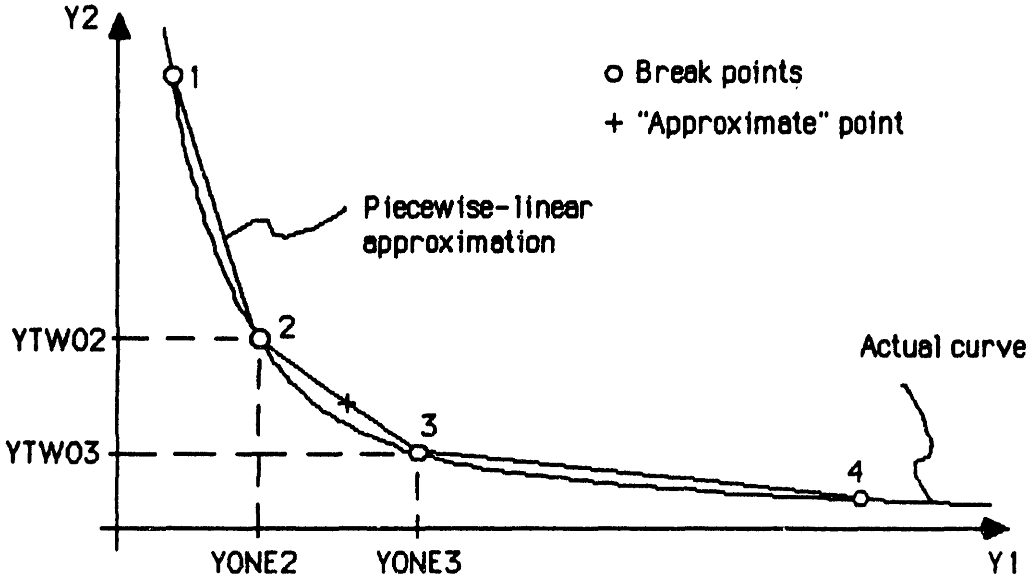


FIGURE 2.4: SCHEMATIC OF PIECEWISE LINEARIZATION

Associated with each break point i is a weighting factor Li ("Lambda-i"). These factors are used to compute the coordinates of "approximate points" on the line segments between break points, through the relations:

$$YONE = \sum_{i=1}^{Npts} (Li * YONEi)$$

$$YTWO = \sum_{i=1}^{Npts} (Li * YTWOi)$$

and the Li values are constrained by

$$\sum_{i=1}^{Npts} Li = 1$$

For an arbitrary "approximate point" on one of the line segments, only the break points at the ends of that segment have $Li \geq 0$, and the other break points have $Li = 0$.

For example, for the "approximate point" in Figure 2.4, break point 2 has weighting factor $L2 \geq 0$, break point 3 has $L3 \geq 0$, and the other points have $Li = 0$. Therefore the equations for the coordinates of this approximate point are:

$$
\begin{aligned}
YONE &= (L1 * YONE1) + (L2 * YONE2) + (L3 * YONE3) + (L4 * YONE4)\\
&= (L2 * YONE2) + (L3 * YONE3)
\end{aligned}
$$

$$
\begin{aligned}
YTWO &= (L1 * YTWO1) + (L2 * YTWO2) + (L3 * YTWO3) + (L4 * YTWO4)\\
&= (L2 * YTWO2) + (L3 * YTWO3)
\end{aligned}
$$

with $L2 + L3 = 1$

Thus the coordinates of an arbitrary point on a line segment are weighted averages of the coordinates of the break points at the ends of that segment.

The requirement that only the break points at the end of the segment can have nonzero weighting factors is called the "Adjacency Condition". In other words, only two break points can have $Li \geq 0$, and these two points must be adjacent, i.e. their i-values must be successive numbers.

Any choice of weighting factors which meets the Adjacency Condition and the requirement that

$$\sum_{i=1}^{Npts} Li = 1$$

corresponds to a single point on the piecewise-linear approximation. For example, in Figure 2.4, ($L1 = 0$, $L2 = 0.4$, $L3 = 0.6$, $L4 = 0$) corresponds to a point on the line segment between break points 2 and 3. This point's coordinates YONE and YTWO can be calculated from the equations above, and their values will approximately fit the constraint equation:

$$YTWO - \frac{1}{YONE} = 0$$

The accuracy of that fit depends on the shape of the curve and the number of break points used. At the break points themselves, the YONE and YTWO values fit this equation exactly.

To incorporate this piecewise-linear approximation into the optimization problem of Table 2.10, the variables Y1 and Y2 are replaced by the expressions for YONE and YTWO. The nonlinear constraint is removed, but a sum-of-Li constraint is added. As an example, if the piecewise-linearization uses 4 break points, the resulting optimization problem is shown in Table 2.11.

TABLE 2.11: OPTIMIZATION PROBLEM WITH PIECEWISE-LINEAR CONSTRAINT

Minimize: ProgCost

$$
\begin{aligned}
&= Kg \\
&\quad + Ch * Nhum \quad + Kh * Z1^{2} \quad - Kh * Z2^{2} \\
&\quad + Ct * Ntel \quad + Kt * Z3^{2} \quad - Kt * Z4^{2} \\
&\quad + Kw * ((L1 * YTWO1) + (L2 * YTWO2) \\
&\qquad\qquad + (L3 * YTWO3) + (L4 * YTWO4))
\end{aligned}
$$

subject to:

$$
\begin{aligned}
&((L1 * YONE1) + (L2 * YONE2) \\
&\quad + (L3 * YONE3) + (L4 * YONE4)) \\
&- (ODhum * Nhum) - (ODtel * Ntel) = 0
\end{aligned}
$$

$$
\begin{aligned}
&Z1 - (0.5 * Nhum) \\
&- (0.5 * ((L1 * YTWO1) + (L2 * YTWO2) \\
&\qquad\qquad + (L3 * YTWO3) + (L4 * YTWO4))) = 0
\end{aligned}
$$

$$
\begin{aligned}
&Z2 - (0.5 * Nhum) \\
&+ (0.5 * ((L1 * YTWO1) + (L2 * YTWO2) \\
&\qquad\qquad + (L3 * YTWO3) + (L4 * YTWO4))) = 0
\end{aligned}
$$

$$
\begin{aligned}
&Z3 - (0.5 * Ntel) \\
&- (0.5 * ((L1 * YTWO1) + (L2 * YTWO2) \\
&\qquad\qquad + (L3 * YTWO3) + (L4 * YTWO4))) = 0
\end{aligned}
$$

$$
\begin{aligned}
&Z4 - (0.5 * Ntel) \\
&+ (0.5 * ((L1 * YTWO1) + (L2 * YTWO2) \\
&\qquad\qquad + (L3 * YTWO3) + (L4 * YTWO4))) = 0
\end{aligned}
$$

$$L1 + L2 + L3 + L4 = 1$$

$$L1, L2, L3, L4 \geq 0$$

$$Nhum, Ntel \geq 0$$

As shown in the table, these substitutions have eliminated the decision variables Y1 and Y2 from the problem, replacing them with new decision variables L1, L2, L3, L4. The quantities YONEi and YTWOi are constants, i.e. the coordinates of the break points selected for the approximation. Therefore the problem now has linear constraints.

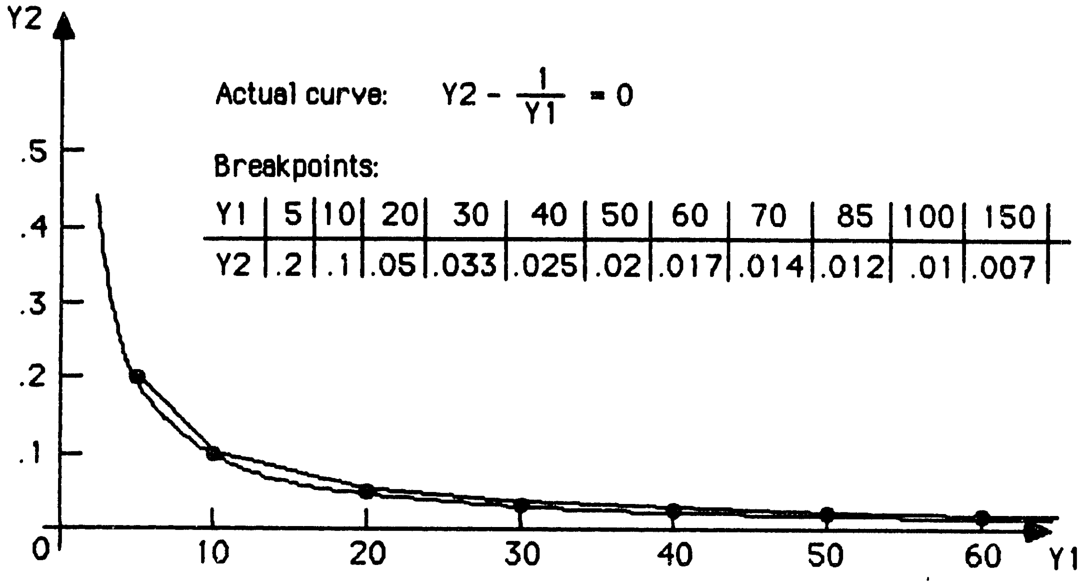
If the problem requires a linearized objective function as well, piecewise-linear approximations for the squared Z-terms can be similarly developed. The resulting linear expressions for the Z variables and their squares would then be substituted into the objective function and constraints.

Whether the problem is tackled by Quadratic Programming or further linearized for Linear Programming, the Adjacency Condition must be maintained during the actual optimization. To be valid, any solution to the problem must have only two nonzero L-values, corresponding to two adjacent break points. This is handled by including a "restricted-entry criterion" in the optimization technique, which limits the investigated solutions to those which satisfy the Condition.

A crucial question in the application of linear approximations is whether the linearized problem is a sufficiently accurate model of the original problem. For this study's restricted case, this can be checked by generating a contour plot of the linear approximation, similar to the contour plots of section 2.3.2. The actual and approximated behaviors of ProgCost can then be compared.

The first step is the selection of break points for the piecewise-linear approximation of Y1 and Y2. After several attempts, a useful example was generated by using 11 break points ranging from Y1 = 5 to Y1 = 150. Physically, Y1 is the crew productivity, i.e. the number of repetitions of the task performed by the humans and teleoperators in one day. Therefore a range of 5 to 150 for Y1 is possible; the actual value would depend on the numbers of humans and teleoperators employed and their productivities.

Figure 2.5 presents the Y1-Y2 plane, showing the original curve from the nonlinear constraint, and part of the piecewise-linear approximation. The two lines appear quite close to each other. However, the Y2-values along the curves are low enough that a small distance between curves can lead to a significant ratio between their values. For example, for (Y1 = 15) the actual curve has (Y2 = 0.067); the linear approximation has (Y2 = 0.075), 12.5% higher.



Actual curve: $Y2 - \frac{1}{Y1} = 0$

Breakpoints:

Y1	5	10	20	30	40	50	60	70	85	100	150
Y2	.2	.1	.05	.033	.025	.02	.017	.014	.012	.01	.007

FIGURE 2.5: PIECEWISE-LINEARIZATION OF CONSTRAINT EQUATION

Figure 2.6 presents the resulting contour plot of ProgCost. The axes, contour values, and numerical constants are the same as in the baseline case in Figure 2.2, so that the approximate and actual cost surfaces can be compared. Each break point has a fixed value of. Y1, and therefore corresponds to a straight line in the Nhum-Ntel plane: these are shown as dashed lines in Figure 2.6.

At the break points, the values of the approximate ProgCost are equal to those of the actual ProgCost. Therefore, along the dashed lines, the contours in Figure 2.6 match exactly the contours in Figure 2.2. This can be checked by superimposing the figures.

In between dashed lines, however, the approximate cost surface deviates from the baseline surface. Away from the origin in the positive quadrant, i.e. for (Nhum > 2.5) or (Ntel > 3), these deviations are not large, so those contours in Figure 2.6 are close to those in Figure 2.2. Nearer the origin, however, the cost surface is very sensitive to inaccuracies in Y2, and the contours bear little resemblance to each other. For example, the 5500 $M contour is a trough contour in Figure 2.6, but not in Figure 2.2.

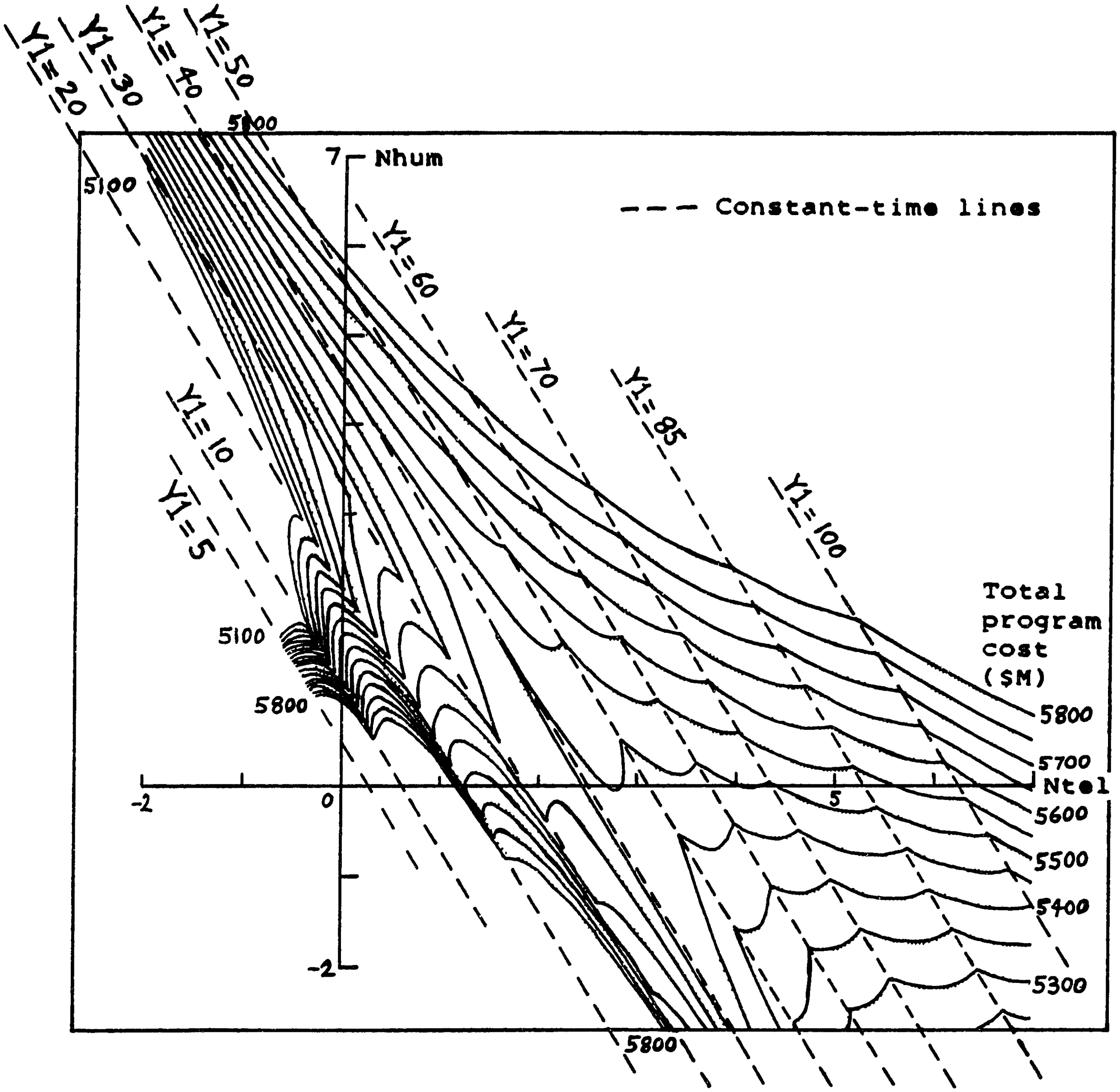


FIGURE 2.6: CONTOUR PLOT OF ProgCost FOR RESTRICTED CASE WITH PIECEWISE-LINEAR APPROXIMATION

These discrepancies can be remedied by adding more break points in useful locations. In Figure 2.7, three break points have been added, with Y1 values of 15, 25, and 35. The resulting cost surface is closer to the baseline. Although distorted, a saddle point is now evident, and the geometry of the two troughs is closer to the baseline shape.

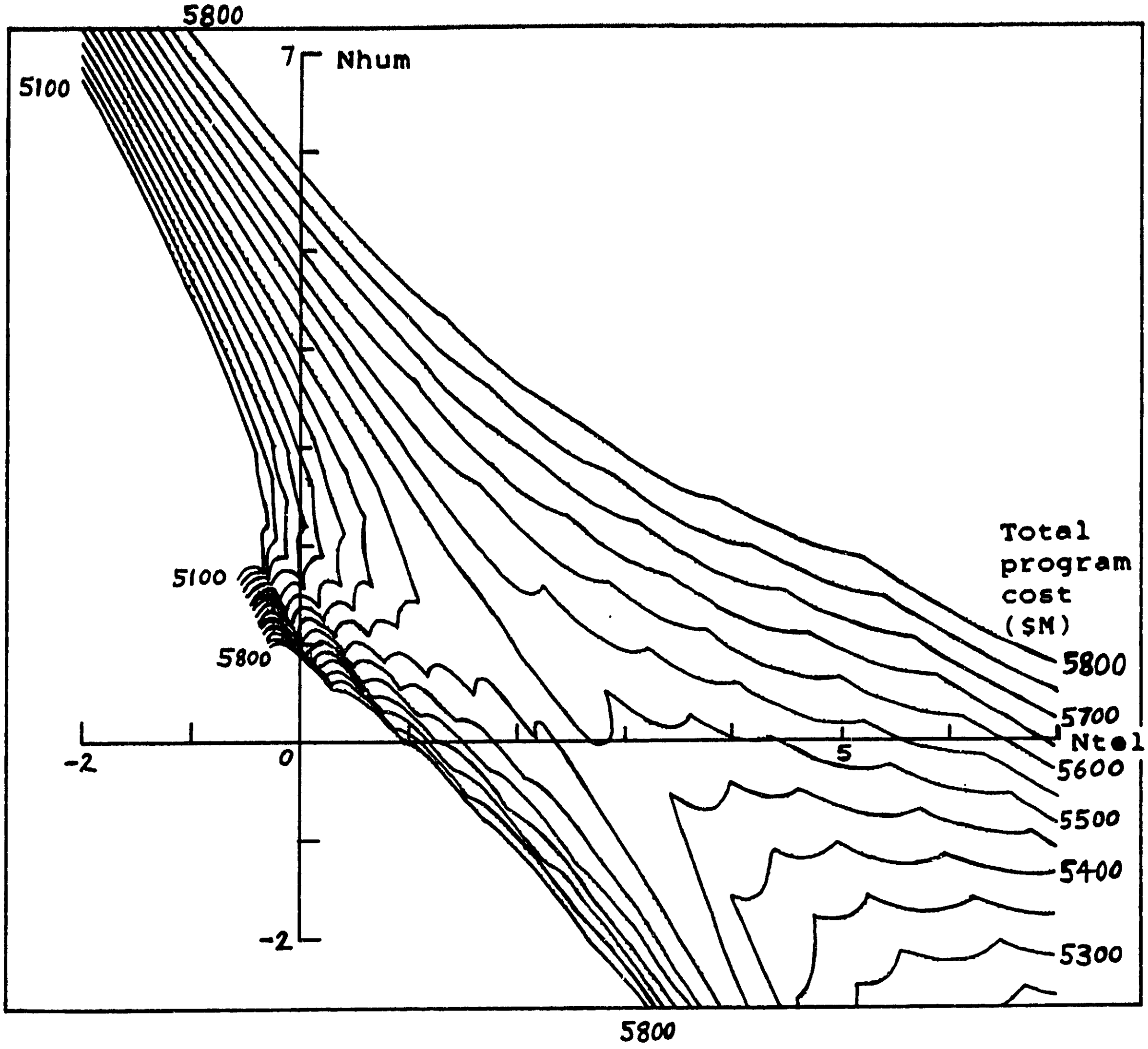


FIGURE 2.7: CONTOUR PLOT OF ProgCost WITH PIECEWISE-LINEAR APPROXIMATION (MORE BREAK POINTS)

However, the actual and approximate cost surfaces are not congruent enough to get a reliable minimum-ProgCost solution from the piecewise-linearization. The accuracy of the approximation can be improved with more break points, but the optimization's computational load is growing: Figure 2.7 already replaces the two decision variables Y1 and Y2 with fourteen L-variables for the break points.

This problem is compounded in multi-task cases. Each additional task would add terms dependent on its completion time, leading to another nonlinear constraint. So each task would require a piecewise-linearization effort similar to the example above.

Another problem involves the selection process for the number and locations of the break points. In the example above, the points were defined by trial and error, using contour plots to identify useful Y1-values. This visual display of the cost surface was particularly important in the selection of additional break points to improve the accuracy of the linearization.

However, in a multi-task or multi-option case, the cost surface would be multi-dimensional, and contour plots would not be available to help this process. Therefore additional logic would be required to define adequate break points, increasing the computational load of the problem.

Thus this study's basic optimization problem is not easily solved by linear approximation, because its complex behavior is difficult to model with linear expressions. If a linear model is formulated, its accuracy is difficult to verify. An adequate piecewise-linear model would require many break points (hence many new decision variables) and would impose the Adjacency Condition on the problem's optimization technique. The resulting computational load makes this approach unlikely for realistic multi-task multi-option cases.

2.5 General Comments on Formal Optimization:

In summary, the preceding sections have examined the behavior of this study's general cost equation, and reviewed a spectrum of existing optimization techniques to tackle the problem. The cost equation includes a basic "first-order" nonlinearity due to its time-dependent terms, and some "second-order" nonlinearities (e.g. learning curves) in some of its terms.

Concentrating on the time-related nonlinearity, the contour plots in section 2.3.2 have shown that the cost surface is fundamentally saddle-shaped (in two or more dimensions). This precludes direct application of Linear Programming or its close variations.

Because the optimization problem includes fractions with decision variables in the numerators and denominators, the Method of Fractional Functionals was attempted. However, because the objective function includes proportional terms as well as fractional terms, this method is unsuitable for this problem.

Also considered were methods to identify stationary points in the cost surface, such as differentiation or Lagrange Multipliers. Unfortunately, the problem's feasible minima are constraint minima, and therefore not stationary points. This can be remedied by adding penalty functions to the objective function, but the resulting minima are only approximations to the true minima. And the problem would have many stationary points, requiring additional logic and computation to identify the global minimum.

The nonlinear objective function can be made separable by a series of variable substitutions. The resulting problem has a quadratic objective function and a set of mostly-linear constraint equations. However, one or more of the constraint equations must include the original problem's nonlinearity. This problem can be tackled by Quadratic Programming after linear approximation of the constraints, or by Linear Programming after linearization of the entire problem.

However, the piecewise-linearization of the nonlinear constraints runs into difficulties, because the complex behavior of the cost equations requires a very detailed approximation. The resulting computational load is excessive, and there are problems in verifying the accuracy of the procedure.

In general, therefore, this study's optimization problem is difficult to tackle with existing techniques. Considerable logical and computational work would be required to guarantee accurate results from any of the reviewed methods. This suggests that a new technique is required, which will take advantage of some special features of this problem to overcome these difficulties.

Chapter 3 will present the development of such a technique, the Staged Optimization Method, showing how this study's problem can be recast in a partly combinatorial form. An example of this method will appear in Chapter 4. As will be discussed in Chapter 5, the Staged Optimization Method has the added advantage that it can also handle "second-order" nonlinearities such as learning curves.

CHAPTER 3:

THE STAGED OPTIMIZATION METHOD

This Chapter presents a new approach to the study's optimization problem: the Staged Optimization Method, or SOM. In section 3.2, the combinatorial character of the problem is recognized: feasible solutions to the problem are combinations of individual human-machine assignments to the tasks. Section 3.3 proves a theorem: under this study's assumptions, for minimum overall cost all the tasks must have equal completion times.

These observations make possible the theoretical development of the SOM (section 3.4). Section 3.5 presents the Staged Optimization Method's iterative procedure, which moves through the combinatorial design space to find the optimal assignments of humans and machines to tasks. In the process, it also optimizes the numbers of humans and machines for each task.

Next, the operation of the SOM is examined, including discussions of local and global minima (section 3.6) and of the effects of the study's assumptions (section 3.7). Section 3.8 describes the advantages of the SOM, including a description of its computational complexity. Finally, section 3.9 examines possible future extensions of the SOM, to tackle some more complex problems.

3.1 General Approach:

Chapter 2 developed and examined this study's general cost equation, which was presented in Table 2.4. For convenience, this equation is repeated here, in Table 3.1.

Chapter 2 also attempted to minimize this cost equation with existing optimization techniques, e.g. fractional functionals, stationary points, separation of variables, Quadratic Programming, and linear approximation programming. As was shown in that chapter, these existing methods are not suitable for this problem, because of the nonlinear character of the cost equation, i.e. the saddle-shaped behavior shown in the contour plots of section 2.3.2.

However, there are particular aspects of the problem which make a specialized optimization technique possible: the combinatorial character of the problem, and the equality of the task completion times in the minimum-cost solution. These concepts will be presented in this chapter.

TABLE 3.1: GENERAL FORM OF THE COST EQUATION (WITHOUT LEARNING CURVES)

Total program cost
ProgCost =

$$(\mathrm{Rw} + \mathrm{Rh} + \mathrm{Rt} + \mathrm{Ra} + \mathrm{Cw}) \qquad \text{Constant costs}$$

$$\left.\begin{array}{l} + \mathrm{Ch} * \mathrm{Nhum}[1] \\ + \mathrm{Hh} * (\mathrm{Nhum}[1] * \mathrm{Ttime}[1]) \\ + \mathrm{Ct} * \mathrm{Ntel}[1] \\ + \mathrm{Ht} * (\mathrm{Ntel}[1] * \mathrm{Ttime}[1]) \\ + \mathrm{Ca} * \mathrm{Naeq}[1] \\ + \mathrm{Ha} * (\mathrm{Naeq}[1] * \mathrm{Ttime}[1]) \end{array}\right] \qquad \text{Task 1 costs}$$

$$\vdots$$

$$\left.\begin{array}{l} + \mathrm{Ch} * \mathrm{Nhum}[j] \\ + \mathrm{Hh} * (\mathrm{Nhum}[j] * \mathrm{Ttime}[j]) \\ + \mathrm{Ct} * \mathrm{Ntel}[j] \\ + \mathrm{Ht} * (\mathrm{Ntel}[j] * \mathrm{Ttime}[j]) \\ + \mathrm{Ca} * \mathrm{Naeq}[j] \\ + \mathrm{Ha} * (\mathrm{Naeq}[j] * \mathrm{Ttime}[j]) \end{array}\right] \qquad \text{Task j costs}$$

$$\vdots$$

$$+ \mathrm{Hw} * \mathrm{Jobtime} \qquad \text{Whole-job timecosts}$$

where:
Rw, Rh, Rt, Ra, Cw, Ch, Hh, Ct, Ht, Ca, Ha, Hw = constants

Rw = Whole-job R&D costs (e.g. R&D of structure to be assembled)
Rh = R&D cost for human-related equipment (e.g. hab. modules)
Rt = R&D cost for teleoperators
Ra = R&D cost for automated equipment
Cw = Whole-job general costs (e.g. procurement of structure to be assembled)
Ch = Number-of-humans cost factor ($/person)
Hh = Human usage cost factor ($/person-day)
Ct = Number-of-teleoperators cost factor ($/teleoperator)
Ht = Teleoperator usage cost factor ($/tel-day)
Ca = Number-of-automated-equipment-units cost factor ($/aeq)
Ha = Automated equipment usage cost factor ($/aeq-day)
Hw = Whole-job timecost factor ($/day)

$$\mathrm{Ttime}[j] = \frac{\mathrm{NREPS}[j]}{(\mathrm{ODhum}[j]*\mathrm{Nhum}[j]) + (\mathrm{ODtel}[j]*\mathrm{Ntel}[j]) + (\mathrm{ODaeq}[j]*\mathrm{Naeq}[j])}$$

$$\mathrm{Jobtime} = \mathrm{MAX}(\mathrm{Ttime}[1], \mathrm{Ttime}[2], \ldots, \mathrm{Ttime}[j], \ldots)$$

(All of the equation's components are also defined in Appendix A.)

These characteristics of the problem do not remove the fundamental nonlinearity of the cost equation. Instead, they make a solution possible by reducing the number of decision variables in the cost equation until it can be minimized with straightforward techniques, despite its nonlinearity.

The general cost equation in Table 3.1 puts the total program cost ProgCost in the form:

```
ProgCost = f(Nhum[1], Ntel[1], Naeq[1], Ttime[1],
             ..., Nhum[j], Ntel[j], Naeq[j], Ttime[j],
             ..., Jobtime)
```

If there are NTSK tasks to be done, ProgCost is therefore a function of ((NTSK * 4) + 1) variables. Since Jobtime must be equal to one of the Ttime terms, and since the Ttime terms can be replaced by expressions using the other variables, this can be reduced to:

```
ProgCost = f(Nhum[1], Ntel[1], Naeq[1],
             ..., Nhum[j], Ntel[j], Naeq[j], ...)
```

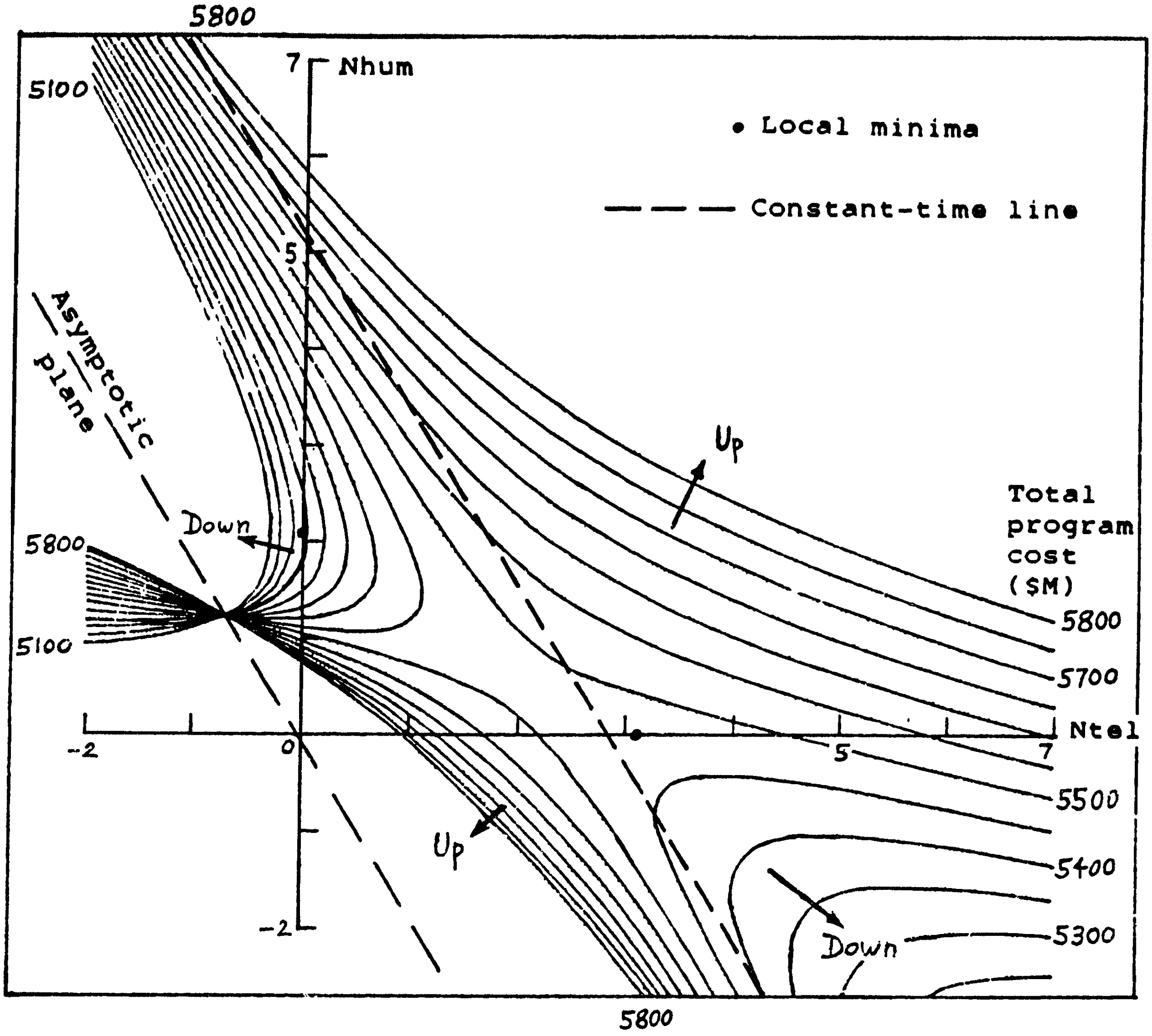
in which ProgCost is a function of (NTSK * 3) variables. For example, a four-task job would make ProgCost a nonlinear function of twelve variables, which is too complex for straightforward optimization.

3.2 Combinatorial Character of the Problem:

3.2.1 Single-Option Candidate Minima:

For the one-task two-option case described in section 2.3, contour plots showed a saddle-shaped behavior, with two downward troughs in the cost surface. This shape is robust, i.e. variations in cost parameters displace the cost surface but do not alter its saddle-shaped character. For convenience, the baseline contour plot in Figure 2.2 is repeated here in Figure 3.1, with the addition of a constant-time line which will be useful later.

In this figure, the ProgCost surface has two local minima, each a candidate to be the overall feasible minimum. These minima occur in the troughs, at the points where the floors of those troughs meet the edges of the feasible region. This leads to the important observation that because of the orientation of the troughs, these minima must lie on the Nhum and Ntel axes.



Rw = 437.403 $M	Ch = 237.800 $M/pers	Hw = 0.569 $M/day
Rh = 1574.313 $M	Hh = 0.439 $M/pers-day	NREPS = 16,688 reps
Rt = 168.430 $M	Ct = 60.190 $M/tel	ODhum = 9.1 reps/pers-day
Cw = 1256.669 $M	Ht = 1.592 $M/tel-day	ODtel = 16.0 reps/tel-day

FIGURE 3.1: CONTOUR PLOT OF ProgCost FOR RESTRICTED CASE (COST EQUATION CONSTANTS AT BASELINE VALUES)

In other words, although the feasible region includes the entire positive quadrant of the Nhum-Ntel plane, the potential cost minima are constrained to be either human-only or teleoperator-only options. Therefore the candidate minima can be located by searching for the minimum value of ProgCost along each axis, i.e. first with (Ntel = 0), then with (Nhum = 0).

If more options were available (e.g. automated equipment as a third alternative) this effect would still be present. This is because there is no coupling of the options in the cost terms, i.e. no cost term includes variables from more than one option. Therefore the cost behavior of any two options is independent of the other options.

For example, in a three-option case, ProgCost forms a multidimensional surface in Nhum-Ntel-Naeq space. Considering any point (Nhum, Ntel, Naeq), ProgCost can be calculated from:

$$
\begin{array}{ll}
\text{ProgCost} = & \\
\quad (Rw + Rh + Rt + Ra + Cw) & \text{Constant costs} \\
\quad \left.\begin{array}{l} + Ch * Nhum \\ + Hh * (Nhum * Ttime) \\ + Ct * Ntel \\ + Ht * (Ntel * Ttime) \\ + Ca * Naeq \\ + Ha * (Naeq * Ttime) \end{array}\right] & \text{Task costs} \\
\quad + Hw * Ttime & \text{Whole-job timecosts}
\end{array}
$$

If the Naeq value of this point is fixed at any value NaeqX, the cost terms dependent on Naeq become either constant or dependent on Ttime only. They can therefore be combined with other terms in the equation:

$$
\begin{array}{ll}
\text{ProgCost} = & \\
\quad \left.\begin{array}{l} (Rw + Rh + Rt + Ra + Cw) \\ + Ca * NaeqX \end{array}\right] & \text{Constant costs} \\
\quad \left.\begin{array}{l} + Ch * Nhum \\ + Hh * (Nhum * Ttime) \\ + Ct * Ntel \\ + Ht * (Ntel * Ttime) \end{array}\right] & \text{Task costs} \\
\quad \left.\begin{array}{l} + Hw * Ttime \\ + Ha * (NaeqX * Ttime) \end{array}\right] & \text{Whole-job timecosts}
\end{array}
$$

This equation has the same behavior as the equation for the Nhum-Ntel case, with higher constant costs and a higher whole-job timecost factor. Therefore the Nhum-Ntel surface for the fixed value of NaeqX will show a saddle-shape (as in Figure 3.1), with local minima on the Nhum and Ntel axes.

Therefore, for any proposed minimum (NhumX, NtelX, NaeqX), which is not on any of the axes (i.e. NhumX, NtelX, NaeqX > 0), the value of Naeq can be held constant at NaeqX, and lower minima will be found on the Nhum and Ntel axes. Therefore the point (NhumX, NtelX, NaeqX) cannot be a true minimum.

This argument can be applied several times, fixing the values NhumX, NtelX, NaeqX in turn, to show that the local minima must always occur on the Nhum, Ntel, and Naeq axes. Therefore the candidate minima in a multi-option problem can be found by minimizing ProgCost along each axis.

Physically, this means that a single task is most cheaply done either by humans alone, by teleoperators alone, or by automated equipment alone, but not by combinations of options. Under the assumptions of the model, this is a logical result: given several options to perform a single task, one option will be inherently cheaper than the others, and that option by itself will be the optimum choice.

3.2.2 Task Definitions to Decouple Productivities:

One of this study's assumptions which contributes to this "one-option-alone" effect is the mutual independence of the productivities in the cost model: for any given task, the productivity of a human-machine option is not affected by the assignment of any other options to the task. For example, humans have the same productivity for Task 1, whether Task 1 is done by humans alone or humans and teleoperators together.

Physically, an exception to this effect could occur if a partnership of humans and machines had a higher productivity than either option alone. To illustrate, for a hypothetical task "Position and Attach Component":

A human alone might take:
a) 11 minutes to align the component;
b) 4 minutes to fasten it;
for a total of 15 minutes per repetition of the task, i.e. a productivity of 4 reps/pers-hr.

A teleoperator alone might take:
a) 8 minutes to align the component;
b) 12 minutes to fasten it;
for a total of 20 minutes and a productivity of 3 reps/tel-hr.

If the productivities are independent, then a partnership of one human and two teleoperators will have crew productivity:

4 + (2 * 3) = 10 reps/crew-hr.

However, if the teleoperators concentrate only on aligning the components, and the human only on fastening them, each component will take:

a) 4 minutes to align;
b) 4 minutes to fasten.

In one hour, the teleoperators will align 15 components, and the human will fasten them, for a higher crew productivity of 15 reps/crew-hour.

This increased crew productivity might lead to a minimum ProgCost below the minimum costs for humans alone or teleoperators alone. This would apparently contradict the "one-option-alone" effect stated above.

However, the reason for the improvement in crew productivity is that each human-machine option specializes in a particular section of the task: the teleoperators align and the human fastens. Therefore the task can be conceptually split into two subtasks:

1) "Align Component", with human productivity 5.45 reps/pers-hr, and teleoperator productivity 7.5 reps/tel-hr;

2) "Fasten Component", with human productivity 15 reps/pers-hr, and teleoperator productivity 5 reps/tel-hr.

This redefinition into separate tasks removes the contradiction between the situation and this study's formulation. Independent productivities can now be applied to the two tasks, and all the relevant behaviors are properly modeled:

a) If one human spends (0.733) of his time in aligning components and (0.267) time in fastening them, in one hour he will align 4 components and fasten 4 components, as in the original definition.

b) Similarly, a teleoperator spending (0.4) time in aligning and (0.6) time in fastening will align and fasten 3 components in one hour.

c) But two teleoperators will align 15 components per hour, and one human will fasten 15 components per hour, as in the increased-productivity definition.

Therefore this study's formulation can handle tasks in which human-machine partnerships would lead to increased productivities: each such task is split into "smaller" tasks, individually more suited to each option. The optimization method will then select minimum-cost options for these smaller tasks, possibly splitting the original task between different options.

Original tasks which are more productively handled by partnerships of options are identified by recognizing that they consist of subtasks which are more efficiently performed by different options. Therefore the logic which identifies such original tasks leads directly to the definition of the smaller tasks.

Besides the productivities, it is possible to have other parameters coupling the options. For example, a teleoperator's procurement cost might be reduced if humans were also assigned to the same task, because the teleoperator might not require as much dexterity. This situation can be handled with two modifications to the original problem.

First, this situation is in fact a variation on the productivity case: the teleoperator handles the non-dextrous part of the task, and the human handles the dextrous part. So the task is split into two subtasks, one non-dextrous and one dextrous. Second, a lower-dexterity lower-cost teleoperator is added as another option to the problem, with a low productivity for the dextrous task. Then the optimization can assign the dextrous task to the human and the non-dextrous task to the cheaper teleoperator.

However, some dependences may bridge not only options but tasks also. For example, if Task 1 is performed by automated equipment and Task 2 by humans, and both tasks are in close physical proximity, there may be a cost penalty on the automated equipment to guarantee physical safety for the humans nearby. This can be partly handled by adding a more-expensive "safe" automated equipment option to the problem. But then some logic must also be added to the optimization algorithms, to include the restriction: "if Task 2 is assigned to humans, Task 1 cannot be handled by the regular automated equipment".

In general, most inter-option dependences can be handled by this study's techniques, using appropriate definitions of tasks and human-machine options. In each case, the physical and mathematical character of the dependence will provide clues on how to remodel the situation. If some interactions cannot be eliminated within this study's modeling techniques, then the optimization technique developed in this chapter may require modifications to solve the problem.

3.2.3 Combinations of Assignments to Tasks:

The recognition that the costs for each task are minimized by one option alone makes the problem fundamentally combinatorial. In the general cost equation in Table 3.1, the costs related to Task 1 will be minimized by using either humans alone, teleoperators alone, or automated equipment alone; similarly for Tasks 2, Therefore, for the complete problem, the optimal solution will be associated with a particular set of assignments of options to tasks. For example, in a four-task case the minimum-cost solution might occur when the set of assignments is:

Task 1:	Humans (HUM)
Task 2:	Teleoperators (TEL)
Task 3:	Humans (HUM)
Task 4:	Automated equipment (AEQ)

Thus the possible solutions to the problem include the set of all combinations of assignments of options to tasks, i.e.:

Task 1:	HUM	HUM	HUM	HUM	HUM	HUM		AEQ
Task 2:	HUM	HUM	HUM	HUM	HUM	HUM	. . .	AEQ
Task 3:	HUM	HUM	HUM	TEL	TEL	TEL		AEQ
Task 4:	HUM	TEL	AEQ	HUM	TEL	AEQ		AEQ

For NTSK tasks to be done, and z options to perform each task, the total number of possible combinations is (z^{NTSK}), e.g. 81 combinations for four tasks and three options.

The assignment of a specific option to each task considerably simplifies the overall cost equation. As an example, for a four-task case and the arbitrary assignments:

Task 1:	HUM
Task 2:	TEL
Task 3:	HUM
Task 4:	AEQ

the cost equation for that combination is shown in Table 3.2. For each task, only those terms related to the selected option are kept. If the Ttime terms are replaced by their simplified expressions, the resulting equation shows ProgCost as a function of five variables: Nhum[1], Ntel[2], Nhum[3], Naeq[4], and Jobtime.

This reduction from twelve to five variables in the cost equation bodes well for the optimization. However, this simpler equation applies only to one combination; the full four-task three-option case includes 81 such equations, any of which might contain the overall optimum solution.

Thus the overall optimization problem can now be separated into two major tasks:

1) Selecting the optimal combination of assignments of options to tasks;

2) Optimizing within that combination for the minimum-cost values of the decision variables.

For the four-task three-option case, the 81 optimizations are affordable. However, a more realistic ten-task three-option case would require 59,049 optimizations, an excessive computation load. Fortunately, the Staged Optimization Method (SOM) will not need to consider all the possible combinations. As will be seen in section 3.8, the direct approach (optimizing all combinations and comparing the results) is an exponential algorithm, while the SOM is a polynomial algorithm.

TABLE 3.2: COST EQUATION FOR ONE COMBINATION OF ASSIGNMENTS (ARBITRARY EXAMPLE COMBINATION)

Total program cost
ProgCost =

(Rw + Rh + Rt + Ra + Cw)	Constant costs
+ Ch * Nhum[1] + Hh * (Nhum[1] * Ttime[1])	Task 1 costs (HUM)
+ Ct * Ntel[2] + Ht * (Ntel[2] * Ttime[2])	Task 2 costs (TEL)
+ Ch * Nhum[3] + Hh * (Nhum[3] * Ttime[3])	Task 3 costs (HUM)
+ Ca * Naeq[4] + Ha * (Naeq[4] * Ttime[4])	Task 4 costs (AEQ)
+ Hw * Jobtime	Whole-job timecosts

where:
Rw, Rh, Rt, Ra, Cw, Ch, Hh, Ct, Ht, Ca, Ha, Hw = constants

$$\text{Ttime}[1] = \frac{\text{NREPS}[1]}{\text{ODhum}[1] * \text{Nhum}[1]}$$

$$\text{Ttime}[2] = \frac{\text{NREPS}[2]}{\text{ODtel}[2] * \text{Ntel}[2]}$$

$$\text{Ttime}[3] = \frac{\text{NREPS}[3]}{\text{ODhum}[3] * \text{Nhum}[3]}$$

$$\text{Ttime}[4] = \frac{\text{NREPS}[4]}{\text{ODaeq}[4] * \text{Naeq}[4]}$$

Jobtime = MAX(Ttime[1], Ttime[2], Ttime[3], Ttime[4])

3.3 Theorem: Equal Task Times for Minimum Cost:

Although simpler than the general cost equation, the equation in Table 3.2 is still not easy to optimize, because of the definition of Jobtime. However, this complication can be removed, leading to a very tractable form of the cost equation.

Section 3.3.1 performs this simplification by proving that in the minimum-cost solution, all of the tasks must have the same completion times. Section 3.3.2 then presents a graphical illustration of this equal-times theorem.

For generality, the theorem is formulated for an arbitrary number of tasks, with more than one option to perform each task. For clarity, the number of options is kept at two (humans and teleoperators); the mathematical proof can be easily extended to cases with more human-machine options.

3.3.1 Theorem and Mathematical Proof:

Hypothesis: For the minimum value of ProgCost from the equation:

$$\begin{array}{ll}
\text{ProgCost} = & \\
\quad (Rw + Rh + Rt + Cw) & \text{Constant costs} \\
\quad \left.\begin{array}{l} + Ch * Nhum[1] \\ + Hh * (Nhum[1] * Ttime[1]) \\ + Ct * Ntel[1] \\ + Ht * (Ntel[1] * Ttime[1]) \end{array}\right] & \text{Task 1 costs} \\
\quad \quad \vdots & \\
\quad \left.\begin{array}{l} + Ch * Nhum[j] \\ + Hh * (Nhum[j] * Ttime[j]) \\ + Ct * Ntel[j] \\ + Ht * (Ntel[j] * Ttime[j]) \end{array}\right] & \text{Task j costs} \\
\quad \quad \vdots & \\
\quad + Hw * Jobtime & \text{Whole-job timecosts}
\end{array}$$

where:

$Rw, Rh, Rt, Cw, Ch, Hh, Ct, Ht, Hw = \text{constants} \geq 0$
(see Table 3.1 for definitions)

$$Ttime[j] = \frac{NREPS[j]}{(ODhum[j]*Nhum[j]) + (ODtel[j]*Ntel[j])}$$

$$Jobtime = MAX(Ttime[1], \ldots, Ttime[j], \ldots)$$

the individual task times and the overall job time must all have the same value, i.e.

```
     Ttime[1] = Ttime[2] = ... = Ttime[j] = ... = Jobtime
```

__Proof:__ Consider the ProgCost equation in the form:

```
ProgCost =

     (Rw + Rh + Rt + Cw)                        Constant costs
       .
       .
       .
                                               _
     + Ch * Nhum[j]                             |
     + Hh * (Nhum[j] * Ttime[j])                |  Task j costs
     + Ct * Ntel[j]                             |
     + Ht * (Ntel[j] * Ttime[j])               _|
       .
       .
       .
                                               _
     + Ch * Nhum[k]                             |
     + Hh * (Nhum[k] * Ttime[k])                |  Task k costs
     + Ct * Ntel[k]                             |
     + Ht * (Ntel[k] * Ttime[k])               _|
       .
       .
       .

     + Hw * Jobtime                             Whole-job timecosts
```

where Task k is the longest task, i.e. Jobtime = Ttime[k], and Task j is an arbitrary task other than Task k. Therefore:

```
     Ttime[j] < Ttime[k]
```

```
Define:
     the Task j costs =
         TaskCost[j] = Ch * Nhum[j]
                     + Hh * (Nhum[j] * Ttime[j])
                     + Ct * Ntel[j]
                     + Ht * (Ntel[j] * Ttime[j])

     the completion time ratio =
         r[j] = Ttime[j]   < 1
                -------
                Jobtime

     a reduced human workforce =
         NhumR[j] = r[j] * Nhum[j]       (< Nhum[j])
```

and a reduced teleoperator workforce =

$$\text{NtelR}[j] = r[j] * \text{Ntel}[j] \qquad (< \text{Ntel}[j])$$

If Task j's workforce is decreased by replacing Nhum[j] by the smaller NhumR[j], and Ntel[j] by the smaller NtelR[j], then the task completion time will become:

$$\text{TtimeR}[j] = \frac{\text{NREPS}[j]}{(\text{ODhum}[j] * \text{NhumR}[j]) + (\text{ODtel}[j] * \text{NtelR}[j])}$$

$$= \frac{\text{NREPS}[j]}{(\text{ODhum}[j] * r[j] * \text{Nhum}[j]) + (\text{ODtel}[j] * r[j] * \text{Ntel}[j])}$$

$$= \frac{\text{Ttime}[j]}{r[j]} \quad > \text{Ttime}[j]$$

And the Task j costs then become:

$$\begin{aligned}
\text{TaskCostR}[j] = {} & \text{Ch} * \text{NhumR}[j] \\
& + \text{Hh} * (\text{NhumR}[j] * \text{TtimeR}[j]) \\
& + \text{Ct} * \text{NtelR}[j] \\
& + \text{Ht} * (\text{NtelR}[j] * \text{TtimeR}[j]) \\[1ex]
= {} & \text{Ch} * r[j] * \text{Nhum}[j] \\
& + \text{Hh} * (r[j] * \text{Nhum}[j] * \frac{\text{Ttime}[j]}{r[j]}) \\
& + \text{Ct} * r[j] * \text{Ntel}[j] \\
& + \text{Ht} * (r[j] * \text{Ntel}[j] * \frac{\text{Ttime}[j]}{r[j]}) \\[1ex]
= {} & r[j] * \text{Ch} * \text{Nhum}[j] \\
& + \text{Hh} * (\text{Nhum}[j] * \text{Ttime}[j]) \\
& + r[j] * \text{Ct} * \text{Ntel}[j] \\
& + \text{Ht} * (\text{Ntel}[j] * \text{Ttime}[j])
\end{aligned}$$

The second and fourth terms in TaskCostR[j] are identical to the second and fourth terms in the original TaskCost[j]. However, since (r[j] < 1), the first and third terms in TaskCostR[j] are smaller than the first and third terms in TaskCost[j]. Therefore:

$$\text{TaskCostR}[j] < \text{TaskCost}[j]$$

So the Task j costs can be lowered by reducing that task's workforce and therefore lengthening its completion time Ttime[j]. As long as Ttime[j] is smaller than Jobtime (i.e. r[j] < 1), such cost reductions are possible. When Ttime[j] has the same value as the longest completion time Ttime[k], then (Ttime[j] = Jobtime) and (r[j] = 1), and the process above yields no cost reduction.

Therefore, for any candidate minimum-cost solution to the optimization problem, if any task has a shorter completion time than any other task, another lower-cost solution can be generated by lengthening the short task's completion time, until it matches the longer task's. Therefore the true minimum solution cannot have any tasks shorter than the longest task, i.e.

Ttime[1] = Ttime[2] = ... = Ttime[j] = ... = Jobtime

(QED)

3.3.2 Notes on the Mathematical Theorem:

First note:

This theorem does not indicate anything about the value of Jobtime for the minimum-cost solution. It only states that for any longest task, the overall cost can be reduced by lengthening the other tasks to match. For example, in the theorem formulation above, Jobtime is equal to Ttime[k], and the theorem proves that the overall cost will be reduced when all other tasks are lengthened to match Ttime[k].

However, this reduced cost might not yet have its true minimum value. If the truly optimal value of Jobtime is different than the value of Ttime[k], then there is still room for improvement. After all the task times have been lengthened to match Time[j], ProgCost can be reduced even further by adjusting the task times again (this time all together) toward the truly optimal Jobtime.

The point is that the truly optimal value of Jobtime is a separate problem, outside the bounds of this theorem: that optimal value must be calculated by an overall optimization of the cost equation. What the theorem of equal task times proves is that, whatever the optimal value of Jobtime turns out to be, in the true minimum-cost solution all of the task completion times must be equal to that Jobtime.

Second note:

Later in this chapter, this theorem and the combinatorial character of the problem (from section 3.2) will be used to develop the Staged Optimization Method. It is therefore important to show that there is no contradiction between the two. In section 3.2.1, a contour plot of ProgCost was used to show that for a single task the local cost minima occur on the number-of- units axes (e.g. the Nhum and Ntel axes in Figure 3.1). However, these local minima correspond to different task completion times Ttime.

But the theorem of equal task times states that in a multitask job the minimum cost occurs when all the tasks have the same completion time Jobtime. Using this result, one step in the optimization method will require that each task be completed in a particular Jobtime. The question is: if a task is given a fixed value of Jobtime for its completion, will its candidate minima still be on the axes, or will the Jobtime restriction allow a point inside the feasible region to be the minimum?

In a one-task two-option case, the task completion time is:

$$\text{Ttime} = \frac{\text{NREPS}}{(\text{ODhum} * \text{Nhum}) + (\text{ODtel} * \text{Ntel})}$$

Therefore any fixed value of Ttime corresponds to a straight line in the ProgCost contour plot, as shown in Figure 3.1. This constant-time line is parallel to the asymptotic plane in the Figure, since they both have slopes $\left(-\frac{\text{ODtel}}{\text{ODhum}}\right)$.

Along any (Ttime = constant) line,

$$\text{Nhum} = \frac{\text{NREPS}}{\text{ODhum} * \text{Ttime}} - \left(\frac{\text{ODtel}}{\text{ODhum}} * \text{Ntel}\right)$$

Substituting this into the cost equation and rearranging terms,

$$\begin{aligned}
\text{ProgCost} = {} & \left.\begin{aligned}
&(\text{Rw} + \text{Rh} + \text{Rt} + \text{Cw})\\
&+ \text{Ch} * \frac{\text{NREPS}}{\text{ODhum} * \text{Ttime}}\\
&+ \text{Hh} * \frac{\text{NREPS}}{\text{ODhum}}\\
&+ \text{Hw} * \text{Ttime}
\end{aligned}\right] \text{Constants}\\
& \left.\begin{aligned}
&- \text{Ch} * \left(\frac{\text{ODtel}}{\text{ODhum}} * \text{Ntel}\right)\\
&- \text{Hh} * \text{Ttime} * \left(\frac{\text{ODtel}}{\text{ODhum}} * \text{Ntel}\right)\\
&+ \text{Ct} * \text{Ntel}\\
&+ \text{Ht} * \text{Ttime} * \text{Ntel}
\end{aligned}\right] \begin{array}{l}\text{Terms proportional}\\ \text{to Ntel}\end{array}
\end{aligned}$$

This linear behavior guarantees that the candidate minima must be at the edge of the feasible region, i.e. along the axes. In one case, when Ttime has a value such that:

$$\text{Ct} + (\text{Ht} * \text{Ttime}) - \left(\text{Ch} * \frac{\text{ODtel}}{\text{ODhum}}\right) - \left(\text{Hh} * \text{Ttime} * \frac{\text{ODtel}}{\text{ODhum}}\right) = 0$$

then the slope of ProgCost is zero, so that all the points along the line are candidate minima. In that instance, using the axes-intersects as candidate minima is acceptable, since ProgCost has the same value along the entire line. Practically, amid an infinite number of possible values for Ttime, that specific case has zero probability of occurrence.

Therefore the candidate minima occur on the number-of-units axes even when the task completion time takes on a fixed value. So the use of the theorem of equal task times does not affect the combinatorial character of the problem.

3.3.3 Physical Interpretation of the Theorem:

The theorem above is illustrated graphically in Figure 3.2. The figure shows the cost elements of a four-task example, drawn as bar graphs. The horizontal axis measures time from the start of work in space. As in the theorem formulation, the options for each task are assumed to be humans and teleoperators; the example can be easily extended to more options.

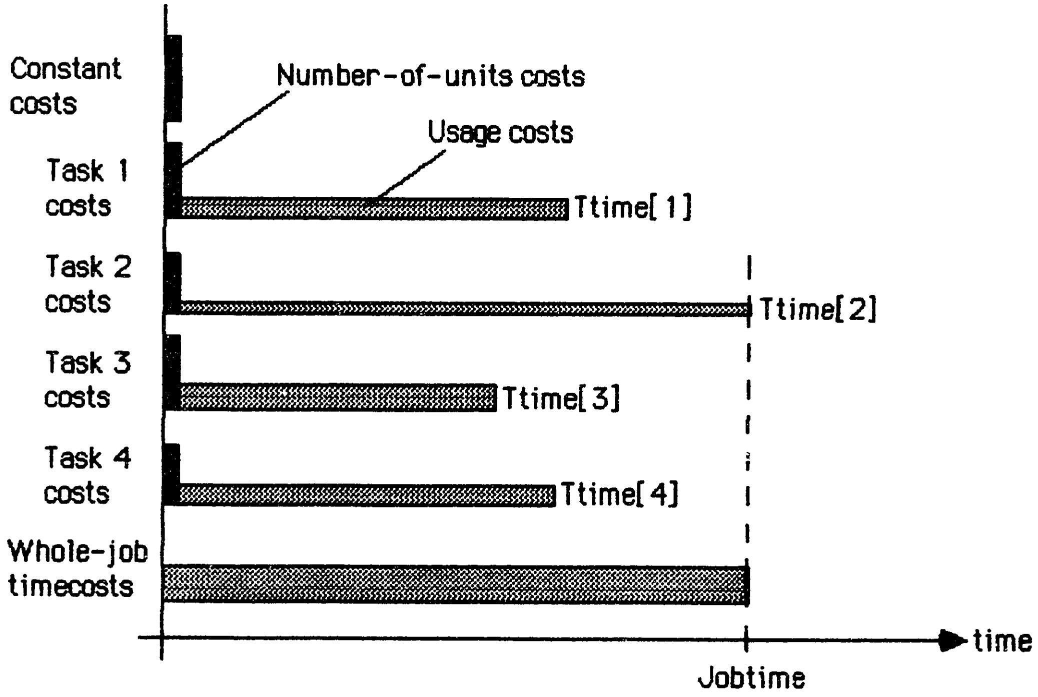


FIGURE 3.2: SCHEMATIC OF COST ELEMENTS IN FOUR-TASK EXAMPLE

At the top of the figure, the constant costs (Rw + Rh + Rt + Ra + Cw) are displayed. Because they are incurred before the start of work in space, they are represented by an impulse at (time = 0).

Below the constant costs are the task costs. For each task, these consist of two components (labeled for Task 1 in the figure):

1) a "number-of-units" cost element;

2) a "usage" cost element;

Because there are two human-machine options for each task, each "number-of-units" cost element includes a number-of-humans term and a number-of-teleoperators term. For example, the number-of-units element for Task 2 =

$$\begin{aligned}&\mathrm{Ch} * \mathrm{Nhum}[2]\\&+ \mathrm{Ct} * \mathrm{Ntel}[2]\end{aligned}$$

These costs cover the procurement and launch of human support equipment and teleoperators, before the start of actual work in space. Therefore this element is represented as an impulse at (time = 0).

The "usage" cost element includes the human and teleoperator usage terms, e.g. for Task 2:

$$\begin{aligned}&\mathrm{Hh} * (\mathrm{Nhum}[2] * \mathrm{Ttime}[2])\\&+ \mathrm{Ht} * (\mathrm{Ntel}[2] * \mathrm{Ttime}[2])\end{aligned}$$

This element extends from (time = 0) to (time = Ttime[2]). Its height is (Hh*Nhum[2] + Ht*Ntel[2]), indicating a constant spending rate, and making its area equal to the usage costs. Such costs include procurement and launch of spares and consumables, and ground support during space tasks.

A single element covers the whole-job timecosts:

$$\mathrm{Hw} * \mathrm{Jobtime}$$

This element extends from (time = 0) to (time = Jobtime). Its height is the constant spending rate Hw, so its area is equal to the whole-job timecosts. Launch of worksite propellant and general worksite ground support are examples of these costs. This example's longest task is Task 2, so Jobtime = Ttime[2].

The Task 1 usage element consists of the terms:

$$\begin{aligned}&\mathrm{Hh} * (\mathrm{Nhum}[1] * \mathrm{Ttime}[1])\\&+ \mathrm{Ht} * (\mathrm{Ntel}[1] * \mathrm{Ttime}[1])\end{aligned}$$

$$\begin{aligned}= &\frac{\mathrm{Hh} * \mathrm{Nhum}[1] * \mathrm{NREPS}[1]}{(\mathrm{ODhum}[1] * \mathrm{Nhum}[1]) + (\mathrm{ODtel}[1] * \mathrm{Ntel}[1])}\\ + &\frac{\mathrm{Ht} * \mathrm{Ntel}[1] * \mathrm{NREPS}[1]}{(\mathrm{ODhum}[1] * \mathrm{Nhum}[1]) + (\mathrm{ODtel}[1] * \mathrm{Ntel}[1])}\end{aligned}$$

The completion time Ttime[1] can be lengthened by reducing the workforces Nhum[1] and Ntel[1]. If the relative sizes of Nhum[1] and Ntel[1] are kept the same (i.e. by multiplying both by the same reduction factor r[1] < 1) then the value of the usage element will also remain the same. Graphically, the area of the usage element remains constant: any lengthening along the time axis is accompanied by a proportional reduction in height.

Physically, usage costs are proportional to the numbers of person-hours and teleoperator-hours spent on the task. For a constant number of repetitions of the task, constant productivities, and a constant ratio between humans and teleoperators, the task will require fixed numbers of person-hours and teleoperator-hours for completion. A decrease in Nhum[1] and Ntel[1] will correspond to a proportional increase in Ttime[1].

Therefore the usage costs will stay constant while the workforce size and the completion time are traded off against each other. On the other hand, Task 1's number-of-units cost element:

Ch * Nhum[1]
+ Ct * Ntel[1]

will be reduced by a decrease in Nhum[1] and Ntel[1], since it is proportional to these quantities. Therefore the overall Task 1 costs can be lowered by reducing the task's workforce and lengthening its completion time, up to the value of Ttime[2].

If Ttime[1] is lengthened beyond Ttime[2], then Task 1 becomes the longest task, and therefore takes over and increases the value of Jobtime. Graphically, this lengthens the whole-job timecosts element. Since that element depends only on Jobtime (i.e. its height is fixed), its area also increases. Therefore the overall effect of lengthening Ttime[1] beyond Ttime[2] is uncertain: the Task 1 costs will be lowered, but the whole-job timecosts will increase, so the overall program costs may decrease or increase.

By the same arguments, the overall costs of the other tasks can also be reduced by shrinking their workforces and lengthening their completion times, up to Ttime[2]. The resulting graph is shown in Figure 3.3.

3.3.4 More Notes on the Theorem:

Third note:

This theorem assumes that fractional workforces and task completion times are acceptable (e.g. 4.3 humans completing a task in 162.73 days). The fractional workforce aasumption is stated in section 2.2.1 and discussed in section B.4.4. Since task times are defined as functions of the workforces (section 2.1.5), they can be fractional as well.

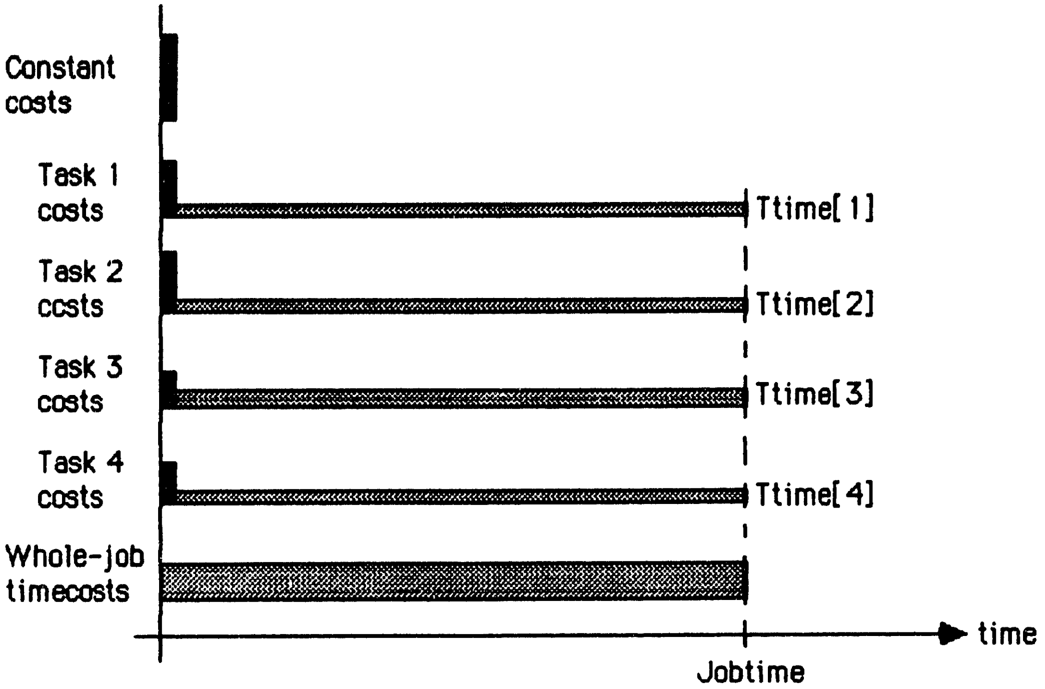


FIGURE 3.3: IMPROVED COST ELEMENTS (EQUAL TASK TIMES)

An implicit corollary of this assumption is that the sizes of workforces and task times do not alter the parameters of space scenarios. As examples, for any given task, 0.7 humans have the same productivity (in reps/pers-hour) as 2.6 humans; and a task requires the same number of person-hours whether it is done in 0.3 days or 9.6 days.

Fourth note:

This theorem can be applied to other cost functions, provided that these functions include certain features. First, the costs must be separable into constant costs, individual task costs, and whole-job timecosts. (The constant costs can be equal to zero.)

Second, each task's costs must separate into a number-of-units element and a usage element. The number-of-units cost must be monotonic with the workforce sizes, i.e. a decrease in workforce size must cause a decrease in this cost element. The usage cost must either remain constant or decrease with increasing completion time. Under these restrictions, a reduction of workforce will lead to a decrease ın the task costs.

If the above conditions are met, lengthening the shorter tasks to match the longest task will reduce overall costs, and the theorem will apply. To keep the overall problem feasible, a third condition is that the whole-job timecosts must monotonically increase with Jobtime, so that a longer job completion time increases these costs. Otherwise, the minimum-cost solution to the problem will be that all tasks should take infinite time. Examples of costs which increase with Jobtime are general work-site support (e.g. high-level management), stationkeeping propellant, and interest on the early R&D costs (if included in the model).

Fifth note:

The cost behavior shown in the theorem is somewhat different from the cost behavior of typical jobs on earth. This is due to the relative importances of the number-of-units costs in space and earth jobs. In space work, the procurement and launch of initial space hardware is likely to be a substantial part of the overall costs. In those cases, lengthening space tasks by reducing their workforces is worth doing, because the reductions in associated hardware can significantly lower the program costs.

By comparison, an earth job incurs relatively less up-front costs. For example, an earth construction job would typically involve less procurement of new machinery: the hardware would come from the company's standing inventory (which is often bought with loans, paid back as timecosts). There is also typically no requirement to provide onsite housing for the workforce; they are expected to provide for themselves from their salaries. So an earth job is dominated more by usage costs, such as the salaries and support of workers, and the maintenance costs for machinery.

As in space, work on earth does include an incentive to finish the work as soon as possible, to provide a completed product and to free tha workforce for other projects. This leads to a desire to keep the longest task short. However, once an earth job's longest task is scheduled, there is less advantage in lengthening the other tasks to match, because their usage costs would remain constant, and their number-of-units costs are relatively less significant. Instead, there is often a desire to finish individual tasks early, so that their workers can be moved to another project.

If a space job includes sequential tasks, i.e. some tasks are completed before some others are started, then workforces could be reassigned, thus performing more than one task. Then their number-of-units costs would be spread over several tasks, and therefore less essential to minimize. In that situation, the theorem of equal task times would have to be reconsidered.

However, this study tackles concurrent rather than sequential tasks, and its Staged Optimization Method is developed accordingly. This concurrent-tasks assumption (mentioned in section 2.1.5) will be discussed in more detail in section 3.7. Section 3.9 will suggest some future extensions to the SOM to enable it to handle the more complex sequential-task problem.

3.4 Development of the Staged Optimization Method:

3.4.1 Optimization of One Combination of Assignments:

The combinatorial formulation and the theorem of equal task times can now be applied to simplify the optimization problem. The cost equation for one example combination of assignments was presented in Table 3.2. That table also contains equations for the task times, which can be rearranged into equations for the other variables:

$$\text{Nhum}[1] = \frac{\text{NREPS}[1]}{\text{ODhum}[1] * \text{Ttime}[1]}$$

$$\text{Ntel}[2] = \frac{\text{NREPS}[2]}{\text{ODtel}[2] * \text{Ttime}[2]}$$

$$\text{Nhum}[3] = \frac{\text{NREPS}[3]}{\text{ODhum}[3] * \text{Ttime}[3]}$$

$$\text{Naeq}[4] = \frac{\text{NREPS}[4]}{\text{ODaeq}[4] * \text{Ttime}[4]}$$

These equations are substituted into the cost equation to make ProgCost a function of the task times only:

$$\begin{aligned}
\text{ProgCost} = {} & (\text{Rw} + \text{Rh} + \text{Rt} + \text{Ra} + \text{Cw}) \\
& + \text{Ch} * \frac{\text{NREPS}[1]}{\text{ODhum}[1] * \text{Ttime}[1]} \\
& + \text{Hh} * \frac{\text{NREPS}[1]}{\text{ODhum}[1]} \\
& + \text{Ct} * \frac{\text{NREPS}[2]}{\text{ODtel}[2] * \text{Ttime}[2]} \\
& + \text{Ht} * \frac{\text{NREPS}[2]}{\text{ODtel}[2]} \\
& + \text{Ch} * \frac{\text{NREPS}[3]}{\text{ODhum}[3] * \text{Ttime}[3]} \\
& + \text{Hh} * \frac{\text{NREPS}[3]}{\text{ODhum}[3]} \\
& + \text{Ca} * \frac{\text{NREPS}[4]}{\text{ODaeq}[4] * \text{Ttime}[4]} \\
& + \text{Ha} * \frac{\text{NREPS}[4]}{\text{ODaeq}[4]} \\
& + \text{Hw} * \text{Jobtime}
\end{aligned}$$

But, by the theorem proved above, the minimum-cost solution must have:

Ttime[1] = Ttime[2] = Ttime[3] = Ttime[4] = Jobtime

So Jobtime can be substituted for the task completion times in the equation above. Rearranging the terms, the resulting "equivalent cost equation" is shown in Table 3.3. Therefore the minimum-cost solution to the equivalent cost equation in Table 3.3 will also be the minimum-cost solution to the original cost equation in Table 3.2.

It should be noted that both of these equations are associated with the specific example combination selected for Table 3.2:

Task 1:	HUM
Task 2:	TEL
Task 3:	HUM
Task 4:	AEQ

Each possible combination of assignments has a specific cost equation (similar to Table 3.2), and an equivalent cost equation (similar to Table 3.3).

The equation in Table 3.3 shows ProgCost as a function of the single variable Jobtime. In this form, ProgCost can be minimized with respect to Jobtime, using a variety of existing single-variable optimization techniques. As will be shown in section 3.5, the Staged Optimization Method expresses ProgCost in this form, and minimizes it by setting its derivative equal to zero.

In effect, the theorem of equal task times is used to formulate ProgCost as a function of Jobtime, so that it can be minimized by a one-parameter optimization. This can be applied to more complex cost functions, provided that the study's assumptions are not violated, and that the theorem of equal task times still applies (section 3.3.4, fourth note). If the resulting equation for ProgCost includes nondifferentiable terms, then the minimization can be performed with gradient-search tcchniques.

This versatility will make the Staged Optimization Method applicable to a wide variety of cost functions. For example, Chapter 4 will present some examples including nondifferentiable learning curves.

Because all of the constants in Table 3.3's equation are positive, the typical behavior of ProgCost can be predicted. Figure 3.4 graphically displays the general shape of ProgCost as a function of Jobtime.

TABLE 3.3: EQUIVALENT COST EQUATION FOR ONE COMBINATION OF ASSIGNMENTS

ProgCost =

$$\left.\begin{array}{l} (Rw + Rh + Rt + Ra + Cw) \\[1em] + \dfrac{Hh * NREPS[1]}{ODhum[1]} \quad + \dfrac{Ht * NREPS[2]}{ODtel[2]} \\[1em] + \dfrac{Hh * NREPS[3]}{ODhum[3]} \quad + \dfrac{Ha * NREPS[4]}{ODaeq[4]} \end{array}\right] \text{Constants}$$

$$+ \frac{Ch * NREPS[1]}{ODhum[1]} * \frac{1}{Jobtime}$$

$$+ \frac{Ct * NREPS[2]}{ODtel[2]} * \frac{1}{Jobtime}$$

$$+ \frac{Ch * NREPS[3]}{ODhum[3]} * \frac{1}{Jobtime}$$

$$+ \frac{Ca * NREPS[4]}{ODaeq[4]} * \frac{1}{Jobtime}$$

$$+ Hw * Jobtime$$

where:

Rw, Rh, Rt, Ra, Cw, Ch, Hh, Ct, Ht, Ca, Ha, Hw = constants > 0
NREPS[j], ODhum[j], ODtel[j], ODaeq[j] = constants > 0, for all j

and from which:

$$Nhum[1] = \frac{NREPS[1]}{ODhum[1]} * \frac{1}{Jobtime}$$

$$Ntel[2] = \frac{NREPS[2]}{ODtel[2]} * \frac{1}{Jobtime}$$

$$Nhum[3] = \frac{NREPS[3]}{ODhum[3]} * \frac{1}{Jobtime}$$

$$Naeq[4] = \frac{NREPS[4]}{ODaeq[4]} * \frac{1}{Jobtime}$$

(The components of these equations are defined in Appendix A.)

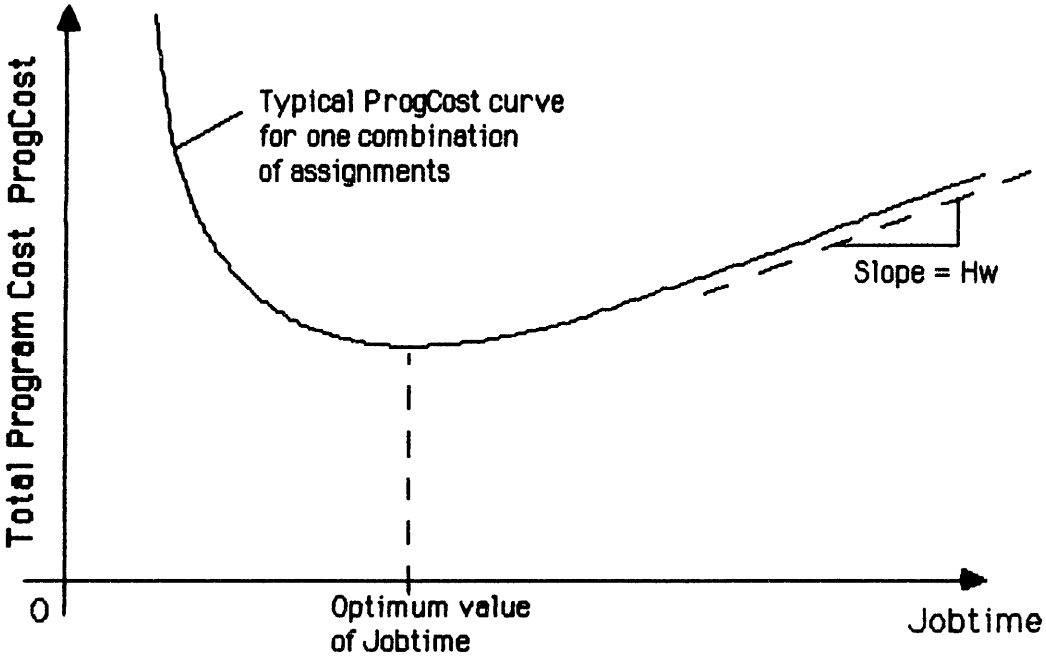


FIGURE 3.4: GENERAL SHAPE OF PROGCOST AS A FUNCTION OF JOBTIME FOR ONE COMBINATION OF ASSIGNMENTS

In this figure, it should be noted that:

a) ProgCost is the total program cost for one combination of assignments of options to tasks. Each combination generates a different ProgCost line, but all have the same general shape.

b) The horizontal axis measures Jobtime, the total completion time for the job - not elapsed time. Each Jobtime value corresponds to a different case of the program, i.e. a case in which the numbers of humans and machines are selected to produce that specific completion time.

When Jobtime is small (i.e. the workforces are large, and complete the job quickly) ProgCost is dominated by the $\frac{1}{\text{Jobtime}}$ terms, which climb to infinity as Jobtime approaches zero. So the left-hand end of the ProgCost line climbs sharply upward.

When Jobtime is large (i.e. small workforces), the $\frac{1}{\text{Jobtime}}$ terms are not significant, and the behavior of ProgCost is dominated by the whole-job timecost term (Hw * Jobtime). Therefore as Jobtime climbs towards infinity (and the workforces drop to zero), the ProgCost curve asymptotically approaches a straight line with slope Hw.

Between these two upward climbs, ProgCost will have a minimum, balancing the $\frac{1}{\text{Jobtime}}$ against the Jobtime term. Physically, this minimum balances the number-of-units costs against the whole-job timecosts.

3.4.2 Selection of Minimum-Cost Assignments:

The previous section showed that for any particular combination of assignments there is a specific optimum value of Jobtime. This section will show that, conversely, a particular value of Jobtime leads to a specific combination of minimum-cost assignments. The next section will combine these two effects into the Staged Optimization Method.

When Jobtime takes on any fixed value, an optimum combination of task assignments can be selected by considering each task individually and selecting the minimum-cost option for that task. Recalling the general cost equation (from Table 3.1), the total program cost is:

```
ProgCost =
     (Rw + Rh + Rt + Ra + Cw)             Constant costs
          .
          .
          .

     + Ch * Nhum[j]                     -|
     + Hh * (Nhum[j] * Ttime[j])         |
     + Ct * Ntel[j]                      |   Task j costs
     + Ht * (Ntel[j] * Ttime[j])         |
     + Ca * Naeq[j]                      |
     + Ha * (Naeq[j] * Ttime[j])        _|
          .
          .
          .

     + Hw * Jobtime                      Whole-job timecosts
```

where j = 1, 2, ..., NTSK.

In this general case, ProgCost cannot yet be minimized task-by-task, because variations in any task's Ttime[j] might also change the value of Jobtime (and therefore the whole-job timecosts); and it is the completion times of the other tasks which determine whether a task will have this effect.

Applying the theorem of equal task times, the equivalent cost equation for minimization is:

$$
\begin{array}{ll}
\text{ProgCost} = & \\
\quad (Rw + Rh + Rt + Ra + Cw) & \text{Constant costs} \\
\quad\quad \vdots & \\
\left.\begin{array}{l}
\quad + Ch * Nhum[j] \\
\quad + Hh * (Nhum[j] * Jobtime) \\
\quad + Ct * Ntel[j] \\
\quad + Ht * (Ntel[j] * Jobtime) \\
\quad + Ca * Naeq[j] \\
\quad + Ha * (Naeq[j] * Jobtime)
\end{array}\right] & \text{Task j costs} \\
\quad\quad \vdots & \\
\quad + Hw * Jobtime & \text{Whole-job timecosts}
\end{array}
$$

ProgCost still cannot be minimized task-by-task, because any variation in one task's completion time now requires a similar variation in all the task times.

However, if Jobtime is constrained to any <u>fixed value</u>, the situation is different, for two reasons. First, the term for whole-job timecosts is now a constant, and therefore does not affect the minimization of ProgCost.

Second, each task's completion time is now a constant, and therefore ProgCost is now a function of the workforce assignments only (i.e. Nhum[j], Ntel[j], and Naeq[j], for all j). ProgCost can therefore be minimized through variations in these quantities.

This has the important effect of <u>decoupling</u> the tasks from each other: because Jobtime has a fixed value, variations in one task's variables no longer have any effect on any other task's variables. For example, the minimum-cost values for Nhum[j], Ntel[j], and Naeq[j] have no effect on the optimal values of Nhum[k], Ntel[k], and Naeq[k].

Therefore, when Jobtime has a fixed value, ProgCost can be optimized by considering each task j <u>individually</u>, minimizing that task's costs by finding appropriate values of Nhum[j], Ntel[j], and Naeq[j]. Within that task, these decision variables are coupled by the equation (from Table 3.1):

$$
\text{Jobtime} = \frac{NREPS[j]}{(ODhum[j] * Nhum[j]) + (ODtel[j] * Ntel[j]) + (ODaeq[j] * Naeq[j])}
$$

However, the combinatorial formulation of the problem states that for any Task j, the minimum cost solution will assign either humans alone, teleoperators alone, or automated equipment alone to the task. And section 3.3.2 (second note) showed that the theorem of equal task times does not contradict the combinatorial character of the problem. Therefore, if humans are selected, the Task j costs will be:

$$\begin{aligned}&Ch * Nhum[j]\\&+ Hh * (Nhum[j] * Jobtime)\end{aligned}$$

subject to:

$$Jobtime = \frac{NREPS[j]}{(ODhum[j] * Nhum[j])}$$

Similarly, if teleoperators are selected, the Task j costs will be:

$$\begin{aligned}&Ct * Ntel[j]\\&+ Ht * (Ntel[j] * Jobtime)\end{aligned}$$

subject to:

$$Jobtime = \frac{NREPS[j]}{(ODtel[j] * Ntel[j])}$$

Similar equations apply if automated equipment is selected.

In each of these cases, the equation for Jobtime provides an explicit solution for Nhum[j], Ntel[j], or Naeq[j] in terms of known quantities. For example, if humans are selected, the value of Nhum[j] must be:

$$Nhum[j] = \frac{NREPS[j]}{(ODhum[j] * Jobtime)}$$

and the Task j costs must be:

$$\begin{aligned}&Ch * \frac{NREPS[j]}{ODhum[j]} * \frac{1}{Jobtime}\\&+ Hh * \frac{NREPS[j]}{ODhum[j]}\end{aligned}$$

The teleoperators-only and automated-equipment-only cases produce similar equations.

Thus the minimization of the Task j costs reduces to the explicit calculations of these costs for each human-machine option. Then the task costs for each option are compared, and the lowest-cost option is selected for Task j.

To minimize the overall ProgCost, this process is applied to each task in turn, generating a minimum-cost assignment for each task. Taken together, these choices of options form the optimal combination of assignments for that value of Jobtime.

This selection process for the minimum-cost combination is illustrated in Figure 3.5. ProgCost is plotted as a function of Jobtime, for a few combinations from a four-task three-option example. (The full family of combinations would have 81 curves.) For Jobtime = c, task-by-task minimization might identify the minimum-cost task assignments:

Task 1: AEQ
Task 2: HUM
Task 3: TEL
Task 4: TEL

which correspond to curve A. Graphically, the selection process considers the vertical line for a particular value of Jobtime, and finds the ProgCost curve with the lowest intersection.

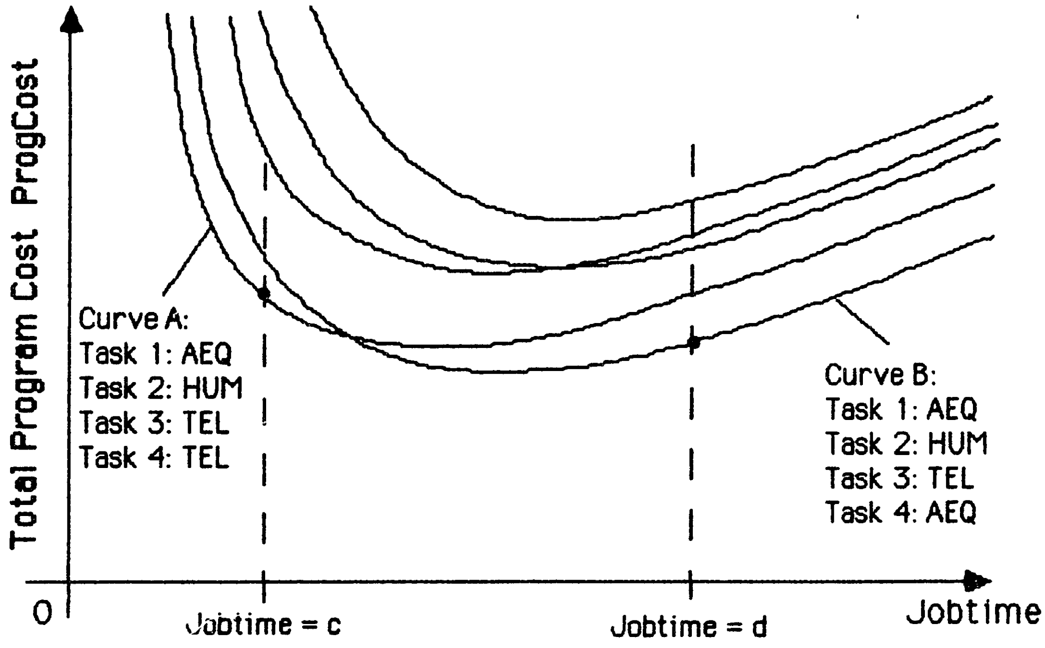


FIGURE 3.5: SELECTION OF MINIMUM-COST COMBINATIONS OF ASSIGNMENTS FOR FIXED VALUES OF Jobtime

This technique can only be applied with fixed values of Jobtime, because the relative merits of the human-machine options vary with Jobtime. For example, for Jobtime = d, the selection process might find that Task 4 is done more cheaply by automated equipment than teleoperators, so that the minimum-cost combination is curve B for that completion time.

Curves plotted for this study's case example confirm this "crossing" behavior between the lines. Physically, some human-machine options have large number-ofunits costs and small usage costs; these are favored for a long Jobtime, because this keeps their workforces small. Other options have low up-front costs and high usage costs; they are preferred for a short Jobtime, because they can afford large workforces.

It is important to note that the selection process does not have to compute ProgCost values for all possible combinations of assignments. This is because it takas advantage of the fact that these combinations are generated from relatively small numbers of tasks and human-machine options. This has important implications for the Staged Optimization Method's computational complexity, which will be discussed in section 3.8.

3.5 The Staged Optimization Method (SOM):

3.5.1 Procedure for the SOM:

The Staged Optimization Method (SOM) minimizes ProgCost by converting the general problem (as shown in Table 3.1) into a series of optimization "stages". Each stage includes two processes:

a) the selection of a candidate combination of human-machine assignments to tasks;

b) for that combination of assignments, the minimization of ProgCost with respect to Jobtim^.

Thus the SOM replaces the original nonlinear optimization problem with iterative applications of a combinatorial selection and a one-parameter optimization. The steps in the method are summarized in Table 3.4, and described below.

Step 1: For each individual task to be done, one human-machine option is selected (e.g. humans, teleoperators, or automated equipment). These selected options form the starting combination of assignments for the method. For example, in a four-task job, the starting combination might be:

Task 1: HUM
Task 2: TEL
Task 3: HUM
Task 4: AEQ

A technique for selecting a likely starting combination of assignments will be described in the next section.

TABLE 3.4: MAJOR STEPS IN THE STAGED OPTIMIZATION METHOD

1) Select a starting combination of assignments of human-machine options to perform the job's individual tasks;

2) For that combination, combine all the task costs into an equation for total program cost ProgCost, expressed entirely in terms of Jobtime;

3) Minimize ProgCost with respect to Jobtime, identifying the minimum-cost Jobtime for that combination of assignments;

4) Using that fixed value of Jobtime, decompose problem into individual tasks, and calculate costs of each human-machine option for each task;

5) From these option costs, select a better combination of assignments of options to tasks;

6) Repeat steps 2 through 5, until step 5 selects the same combination as the previous iteration.

Step 2: For the selected combination of assignments, the total program cost ProgCost can be expressed entirely in terms of the job completion time Jobtime, by using the theorem of equal task times. For the four-task example, this equivalent cost equation was presented in Table 3.3. This form of the equation can be generalized to any number of tasks and any number of human-machine options:

ProgCost =

$$\left.\begin{aligned}&(Rw + Rh + Rt + Ra + Cw)\\&+ \sum_{j=1}^{NTSK} \frac{Hhmo[j] * NREPS[j]}{ODhmo[j]}\end{aligned}\right]\text{Constants}$$

$$+ \sum_{j=1}^{NTSK} \frac{Chmo[j] * NREPS[j]}{ODhmo[1]} * \frac{1}{Jobtime}$$

$$+ Hw * Jobtime$$

where hmo[j] indicates "human-machine option selected for task j",

$$\text{e.g. Hhmo[j]} = \begin{cases} \text{Hh} & \text{if humans are selected} \\ \text{Ht} & \text{if teleoperators are selected} \\ \text{Ha} & \text{if automated equipment is selected} \end{cases}$$

$$\text{ODhmo[j]} = \begin{cases} \text{ODhum[j]} & \text{if humans are selected} \\ \text{ODtel[j]} & \text{if teleoperators are selected} \\ \text{ODaeq[j]} & \text{if automated equipment is selected} \end{cases}$$

$$\text{Chmo[j]} = \begin{cases} \text{Ch} & \text{if humans are selected} \\ \text{Ct} & \text{if teleoperators are selected} \\ \text{Ca} & \text{if automated equipment is selected} \end{cases}$$

<u>Step 3</u>: ProgCost is minimized with respect to Jobtime, using appropriate one-parameter optimization techniques. In this case, the above expression for ProgCost is differentiable, and Figure 3.4 showed that ProgCost has a functional minimum. Therefore the derivative of the expression can be set equal to zero, producing an equation of the form:

$$((-1) * \text{Constant} * \text{Jobtime}^{-2}) + \text{Hw} = 0$$

$$\text{where Constant} = \sum_{j=1}^{\text{NTSK}} \frac{\text{Chmo[j]} * \text{NREPS[j]}}{\text{ODhmo[j]}}$$

$$\text{so the minimum-cost value of Jobtime} = \text{sqrt}\left(\frac{\text{Constant}}{\text{Hw}}\right)$$

<u>Step 4</u>: Using the technique developed in section 3.4.2, Jobtime is fixed at the value computed above, and individual task costs are calculated. For each task j:

$$\text{Task j costs} = \text{Chmo[j]} * \frac{\text{NREPS[j]}}{\text{ODhmo[j]}} * \frac{1}{\text{Jobtime}}$$

$$+ \text{Hhmo[j]} * \frac{\text{NREPS[j]}}{\text{ODhmo[j]}}$$

Within each task, these costs are calculated for each human-machine option, e.g. first for humans (Chmo[j] = Ch; ODhmo[j] = ODhum[j]; Hhmo[j] = Hh), then for teleoperators (Chmo[j] = Ct; ODhmo[j] = ODtel[j]; Hhmo[j] = Ht), etc. The results are the costs of each task, for each human-machine option, if all the tasks completion times have the value of Jobtime calculated in step 3.

Step 5: For each task, the task costs for each human-machine option are compared, and the least expensive option is identified. Collectively, these option selections for each task form the combination of assignments with the lowest ProgCost, for the particular value of Jobtime calculated in step 3.

For example, in the four-task case, the costs calculated in step 4 might indicate that automated equipment is the least expensive option for Task 1, humans for Task 2, teleoperators for Task 3, and automated equipment for Task 4. The resulting improved combination of assignments would be:

Task 1: AEQ
Task 2: HUM
Task 3: TEL
Task 4: AEQ

(Note: Section 3.8.1 will describe an alternative method for steps 4 and 5, providing some useful insights into the computational complexity of the SOM.)

Step 6: The new combination of assignments is taken back to step 2, and the optimization process in steps 2 through 5 is repeated. The method iterates until a pass through step 5 yields the same combination of assignments as the previous iteration.

At that point, the method has identified the overall problem's optimum Jobtime, and the optimal combination of options for the tasks. The numbers of each option to be assigned to each task are calculated from the generalized form of equations in Table 3.3:

$$\text{Nhmo}[j] = \frac{\text{NREPS}[j]}{\text{ODhmo}[j] * \text{Jobtime}}$$

and the minimum value of ProgCost is computed from the equation shown under "Step 2" above.

Graphically, the path of the Staged Optimization Method through the design space is illustrated in Figure 3.6. A few ProgCost curves are shown, from the 81 curves of a four-task three-option case example. Step 1 selects a starting combination of assignments, which corresponds to one of the curves (but not to any particular point on the curve).

Step 2 writes that curve's equation for ProgCost, in terms of Jobtime only. Step 3 slides down that curve to minimize ProgCost, identifying the minimum point and the associated value of Jobtime.

Step 4 fixes that value of Jobtime, and computes the costs of each human-machine option, for each task. Step 5 compares these costs for each task, and drops down the vertical Jobtime line to select the lowest-cost combination of assignments.

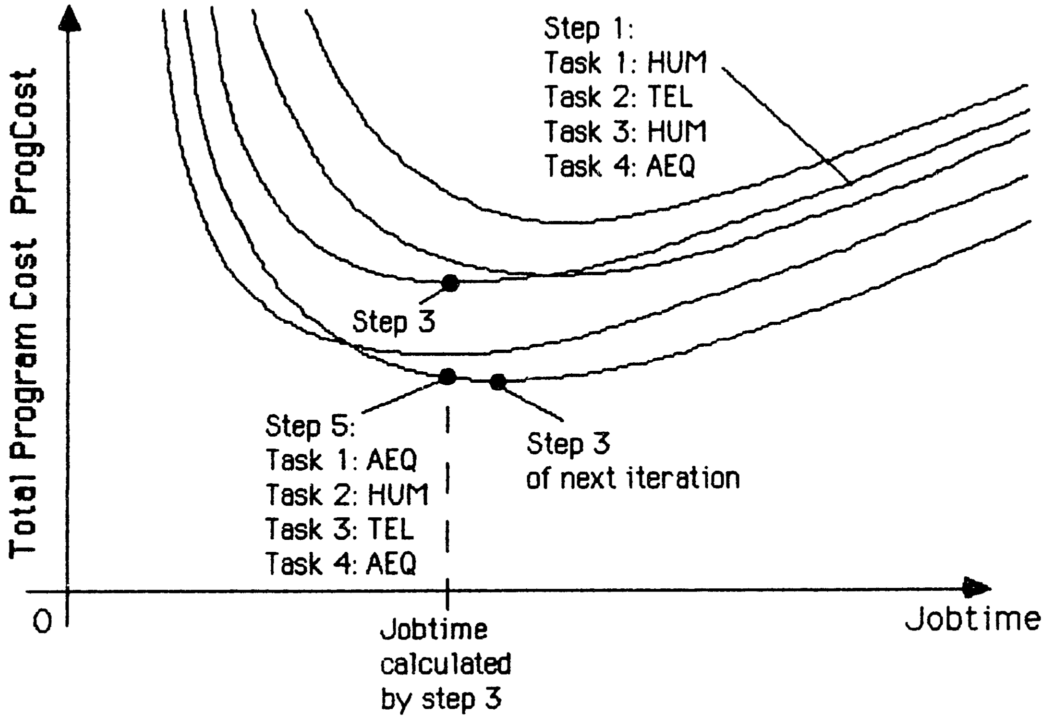


FIGURE 3.6: PATH OF THE STAGED OPTIMIZATION METHOD THROUGH AN EXAMPLE DESIGN SPACE

This new combination is iterated back to step 2, which sets up the ProgCost equation for that curve. Then step 3 of this iteration minimizes ProgCost on the curve, reaching the overall minimum in the figure. Steps 4 and 5 then selects the lowest-cost combination of assignments, which turns out to be the same as the last iteration's. The process stops, and the optimal values of the job completion time, the human-machine assignments, and the total program cost are displayed as outputs.

Thus the Staged Optimization Method deals with this study's nonlinear optimization problem by alternating one-parameter optimizations and combinatorial selections. The one-parameter optimizations are possible because they are applied to individual combinations of assignments, i.e. each happens along a single curve in the figure. The selections of combinations are possible because they are done for fixed values of Jobtime, i.e. each happens along a vertical line in the figure.

By alternately sliding along a curve to its minimum, and dropping vertically to the lowest curve, the SOM travels through the design space to the overall minimum. This is done without requiring calculations for each possible combination of assignments.

3.5.2 Starting Point for Method:

The Staged Optimization Method can be started with any arbitrary combination of assignments, starting at step 1 in the description above. Or the SOM can start with an arbitrary value of Jobtime, beginning at step 4, but it is difficult to predict a starting value of Jobtime near the eventual optimum. However, there is a technique to choose a likely starting combination of assignments, close enough to the optimum to reduce the computation load.

This starting technique resembles the SOM's selection process for assignments (steps 4 and 5), in that it considers each task individually. However, while step 4 of the SOM starts with a value for Jobtime, in this case there is no likely value of Jobtime available.

Therefore the starting technique conceptually separates the job to be done into its individual tasks, and makes each task a job by itself. So the costs of any task j are given by:

$$
\begin{aligned}
\mathrm{TaskCost}[j] = \; & \\
& \mathrm{Ch} * \mathrm{Nhum}[j] \\
& + \mathrm{Hh} * (\mathrm{Nhum}[j] * \mathrm{Ttime}[j]) \\
& + \mathrm{Ct} * \mathrm{Ntel}[j] \\
& + \mathrm{Ht} * (\mathrm{Ntel}[j] * \mathrm{Ttime}[j]) \\
& + \mathrm{Ca} * \mathrm{Naeq}[j] \\
& + \mathrm{Ha} * (\mathrm{Naeq}[j] * \mathrm{Ttime}[j]) \\
& \\
& + \mathrm{Hw} * \mathrm{Ttime}[j]
\end{aligned}
$$

where Ttime[j] =

$$\frac{\mathrm{NREPS}[j]}{(\mathrm{ODhum}[j] * \mathrm{Nhum}[j]) + (\mathrm{ODtel}[j] * \mathrm{Ntel}[j]) + (\mathrm{ODaeq}[j] * \mathrm{Naeq}[j])}$$

Constant costs have been omitted, because they have no effect on the calculations which follow.

Each task is then optimized by itself, trading off its workforce size against its timecost. This is the reason for modeling each task as a separate job: if the Hw * Ttime[j] term were not included, minimizing the costs would reduce the workforces to zero.

As discussed in section 3.2, the combinatorial character of this problem implies that for mimimum costs the task must be performed by humans alone, teleoperators alone, or automated equipment alone. If humans are selected, the costs for Task j can be expressed in terms of Ttime[j]:

TaskCost[j] =

$$Ch * \frac{NREPS[j]}{ODhum[j]} * \frac{1}{Ttime[j]}$$

$$+ Hh * \frac{NREPS[j]}{ODhum[j]}$$

$$+ Hw * Ttime[j]$$

Then TaskCost[j] is minimized relative to Ttime[j], identifying the optimum completion time HumTime[j] and the minimum task costs HumCost[j]. Then the explicit relationship between Ttime[j] and Nhum[j]:

$$Nhum[j] = \frac{NREPS[j]}{ODhum[j]} * \frac{1}{Ttime[j]}$$

is used to compute the optimum number of humans HumNumber[j].

Similar calculations for the teleoperator-only and automated-equipment-only cases produce their minimum costs TelCost[j] and AeqCost[j]. The minimum costs for each human-machine option are then compared, to find the lowest-cost option for Task j. As an example, Task 1 might have its lowest costs when:

HumNumber[1] = 3.2 humans

and Task 2 might be optimized by:

TelNumber[2] = 4.6 teleoperators.

Thus this process generates a set of assignments which optimize each task individually. This is a step in the right direction, and therefore the resulting combination of assignments is a useful choice to start the Staged Optimization Method.

In some cases, this technique may even generate the optimum combination for the entire problem (e.g. directly identifying the lowest line in Figure 3.6). But this is not guaranteed, because the different values of Jobtime may change the relative merits of options for some tasks. For example, Task 1 might be individually optimized by:

HumNumber[1] = 3.2 humans
HumTime[1] = 124 days.

But when the tasks are reassembled into the original job, the the truly optimal Jobtime might turn out to be 240 days, and for this longer Jobtime another option might be cheaper for Task 1. As will be seen in Chapter 4, in this study's case example this technique produces a good starting combination, but not an optimal one.

If the starting technique does identify the optimal assignments, the SOM still has work to do. It must reassemble the multi-task job and minimize its cost, calculating the optimal Jobtime for all the tasks, and the associated optimal workforce sizes for each task.

3.5.3 Reduced-Option Cases:

When the Staged Optimization Method is applied to a space scenario which includes humans, teleoperators, and automated equipment, ProgCost includes the R&D costs of each of these human-machine options in the constant terms Rh (for human support hardware), Rt (for teleoperators), and Ra (for automated equipment).

In the description above, the SOM assumes that all three options are present in the optimal solution, and therefore formulates the problem with all three R&D terms in ProgCost. However, the optimization process may lead to optimal assignments which omit an option altogether. For example, automated equipment might not be selected for any task. In that case, the total program cost can be further reduced by eliminating that option altogether, i.e. by removing its R&D cost term.

Even when all the options appear in the solution, the removal of an option may lower the total cost. If an option's R&D cost is high enough, the savings from its elimination may be greater than the expense of switching some tasks to other human-machine options. In that case, the "reduced-option" scenario will optimize to a lower ProgCost.

Therefore it is not sufficient to apply the Staged Optimization Method to the three-option case. It must also be used on reduced-option cases, e.g.:

Humans and teleoperators
Humans and automated equipment
Teleoperators and automated equipment
Humans alone
Teleoperators alone
Automated equipment alone

Fortunately, the SOM's algorithms can also handle these cases, and in fact the calculations are simpler because there are fewer options to consider. (The computational complexity of the reduced-option cases is discussed in section 3.8.) Once the SOM has been applied to the three-option, two-option, and one-option cases, all of the minimum-cost solutions are compared to find the true optimal solution to the problem.

3.6 Local and Global Minima in Method's Solution:

Under certain circumstances, the Staged Optimization Method can reach a solution which is a local minimum in the design space rather than the desired global minimum. This section describes the situations which can lead to this result, and suggests a simple extension of the SOM to avoid this problem.

The Staged Optimization Method's path through the design space was illustrated in section 3.5.1's Figure 3.6, using plots of several curves for ProgCost as a function of Jobtime. Such graphs are the clearest way to describe the behavior of the SOM, and they are therefore also used in this section.

3.6.1 Behavior of the ProgCost-Jobtime Curves:

Each ProgCost-Jobtime curve corresponds to one combination of assignments of options to tasks, and has an equation of the form:

ProgCost =

$$\left.\begin{array}{l} (Rw + Rh + Rt + Ra + Cw) \\ \displaystyle + \sum_{j=1}^{NTSK} \frac{Hhmo[j] * NREPS[j]}{ODhmo[j]} \end{array}\right\} \text{Constants}$$

$$+ \sum_{j=1}^{NTSK} \frac{Chmo[j] * NREPS[j]}{ODhmo[1]} * \frac{1}{Jobtime}$$

$$+ Hw * Jobtime$$

where hmo[j] indicates "human-machine option selected for task j",

$$\text{e.g. } Hhmo[j] = \begin{cases} Hh & \text{if humans are selected} \\ Ht & \text{if teleoperators are selected} \\ Ha & \text{if automated equipment is selected} \end{cases}$$

For a given job to be done, Rw, Rh, Rt, Ra, Cw, NREPS[j], and Hw have fixed values. The values of Hhmo[j], Chmo[j], and ODhmo[j] depend on which human-machine options are selected for each task; it is those three factors which affect the shapes of the ProgCost-versus-Jobtime curves. Their effects are displayed in the following two figures.

If one hypothetical human-machine option (HMO) is applied to an arbitrary set of four tasks, with the nominal parameters:

Chmo = 250 $M/hmo

Hhmo = 2.0 $M/hmo-day

ODhmo[1] = 70 reps/hmo-day
ODhmo[2] = 20 reps/hmo-day
ODhmo[3] = 2 reps/hmo-day
ODhmo[4] = 40 reps/hmo-day

and using the four-task case example values (from Chapter 4) for Rw, Rh, Rt, Ra, Cw, NREPS[j], and Hw, the resulting ProgCost-Jobtime curve is the top line in each graph in Figure 3.7. For low values of Jobtime, this curve is dominated by the $\frac{1}{\text{Jobtime}}$ terms in the equation above, which climb to infinity as Jobtime goes to zero. For high values of Jobtime, the curve is dominated by the (Hw * Jobtime) term, and therefore approaches a straight line with slope Hw.

In the upper graph, the other lines correspond to reduced values of the number-of-units cost factor Chmo. Since this factor appears only in the $\frac{1}{\text{Jobtime}}$ terms, its effect is limited to the left side of the curve. A lower Chmo pulls down that end of the curve, moving the minimum to the left. On the right-hand side, the curves still approach the same straight line with slope Hw.

The lower graph in Figure 3.7 shows the effect of the usage cost factor Hhmo on the same curve. Hhmo appears only in constant terms in the equation above, and therefore a lower Hhmo moves the curve downward, without changing its shape or the Jobtime value of its minimum. The right-hand ends of the curves approach parallel lines, each with slope Hw.

Figure 3.8 shows the effects of productivities on the same ProgCost-Jobtime curve. The nominal productivities listed above are all multiplied by constant factors to produce the lower curves in the figure. As the productivities increase, both the constant terms and the $\frac{1}{\text{Jobtime}}$ terms in the equation are reduced. Therefore the curve moves downward, but its left-hand end moves down further. As in the Hhmo curves, the right-hand ends approach parallel lines with slope Hw.

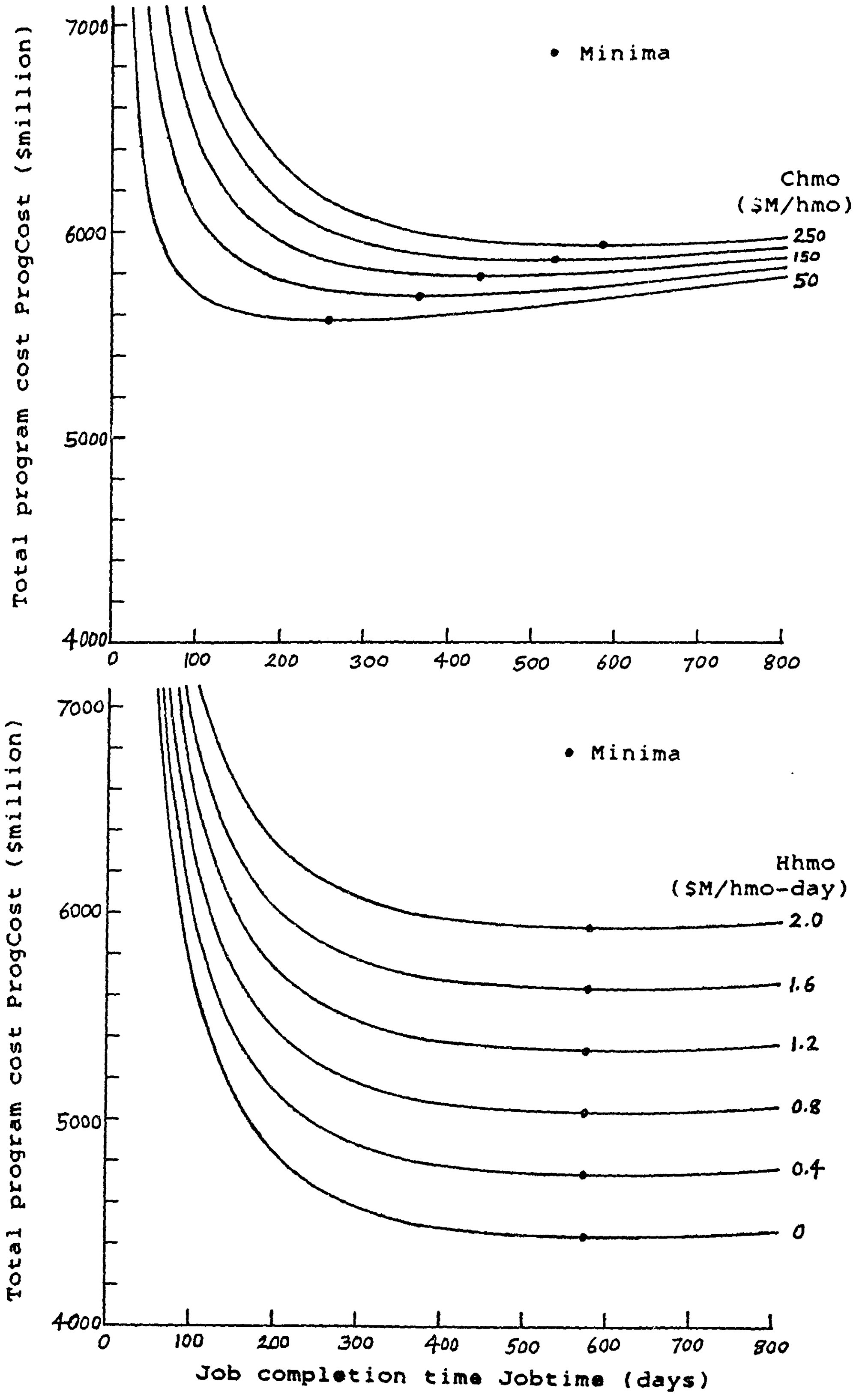


FIGURE 3.7: EFFECTS OF NUMBER-OF-UNITS COST FACTOR Chmo AND USAGE COST FACTOR Hhmo ON EXAMPLE ProgCost-Jobtime CURVE

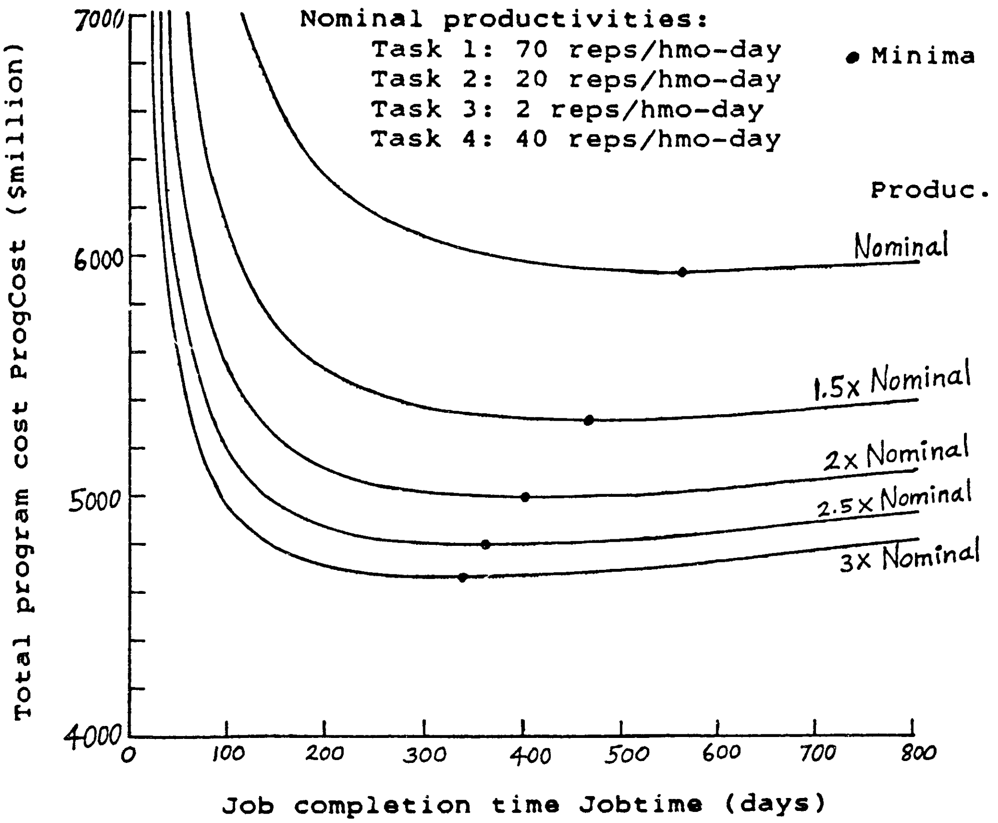


FIGURE 3.8: EFFECT OF PRODUCTIVITIES ON EXAMPLE ProgCost-Jobtime CURVE

3.6.2 Possibility of Multiple Local Minima:

The effects displayed above can lead to separate local minima in the design space. If two human-machine options (OPN1 and OPN2) have the parameters:

Copn1 = 250 $M/unit
Hopn1 = 0.9 $M/unit-day

and

Copn2 = 50 $M/unit
Hopn2 = 1.4 $M/unit-day

with the nominal productivities above for both options, the two resulting ProgCost-Jobtime curves are shown in Figure 3.9. If these curves are at the bottom of the design space, they offer two local minima as candidates for the overall solution.

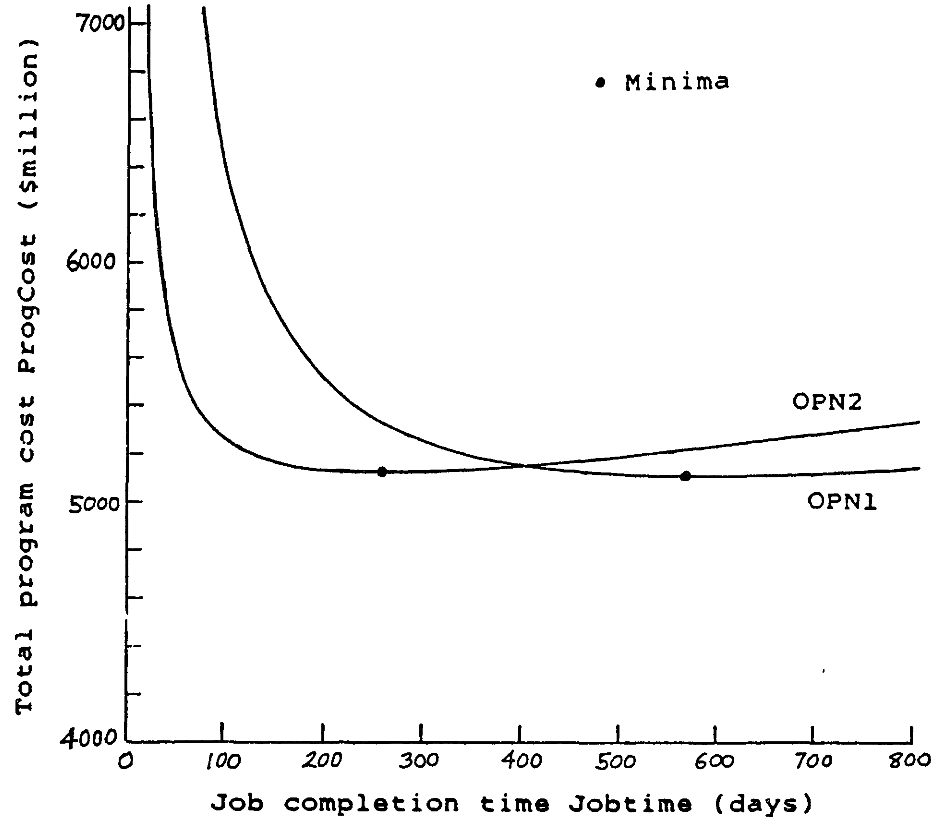


FIGURE 3.9: EXAMPLE OF CROSSING ProgCost-Jobtime CURVES SHOWING TWO LOCAL MINIMA

The curves' crossing behavior is due to the inverse effects of the number-of-units and usage factors in the two options. The relatively high Copn1 raises the left side of OPN1's curve; the high Hopn2 raises the right side of OPN2. By comparison, variations in productivities would not cause such shapes, because they would affect both ends of the curves similarly.

If the SOM's path through the design space comes down the right side of the figure, it will reach OPN1's minimum. Seeing no curves below this point, it will conclude that this is the global minimum. Similarly, a leftward path may trap the SOM in OPN2's minimum, leaving the method unaware of the other potential solution.

Fortunately, this situation is unlikely to occur in a real scenario, for several reasons. First, the curves in the example above are for single options, e.g. OPN1 is assigned to all four tasks. In those instances, the SOM will consider each curve separately in its reduced-option "OPN1-only" and "OPN2-only" cases, and the two minima will be compared for the final solution.

However, the crossing behavior shown in Figure 3.9 can also occur in curves associated with combinations of assignments. As in the single-option cases, if a combination omits one or more options, it will be separately optimized in a reduced-option case. Therefore the SOM can only be trapped by combinations which include the same list of options, e.g. two combinations of humans, teleoperators, and automated equipment, but with different assignments of these options to tasks.

In those cases, the shape-driving parameter Chmo is replaced by a weighted average of the parameters Chmo[j], from the terms:

$$\sum_{j=1}^{NTSK} \frac{Chmo[j] * NREPS[j]}{ODhmo[1]} * \frac{1}{Jobtime}$$

and the other shape-driving parameter Hhmo is also replaced by a weighted average from:

$$\sum_{j=1}^{NTSK} \frac{Hhmo[j] * NREPS[j]}{ODhmo[j]}$$

Therefore, for a combination of assignments including all the human-machine options, its shape-driving parameters will be "averaged", moderating the effects of the extreme values for Chmo and Hhmo. Thus two arbitrary combinations of assignments are likely to end up with similar shapes for their ProgCost-Jobtime curves.

Despite this moderating effect, the following situation can still lead to a two-minimum behavior: two combinations of assignments are selected, each including all of the options available. But each combination is dominated by a different option, because this option is assigned to most of the largest tasks. Then, if the dominant options in each combination have inverted values for Chmo and Hhmo (as in OPN1 and OPN2 above), the shapes of their curves will approach those in Figure 3.9 (though moderated slightly by the other options in each combination).

Thus the range of situations which can cause two minima is narrow. As a further constraint, the relative values of the Chmo and Hhmo parameters are also limited. In Figure 3.9, because the right-hand ends of the curves are constrained to approach the slope Hw, a small variation in one of the Hhmo values will eliminate the crossing behavior.

For example, if Hopnl (0.9 in the figure) rises above 1.1 or falls below 0.6, one of the minima rises above the other curve, eliminating the two-minimum situation. One effect of this restriction is that the ProgCost values of the two minima have to be relatively close to each other, although their Jobtime values can be quite different.

Furthermore, to cause a problem for the SOM, both curves must be at the bottom of the design space, i.e. their minima must be valid candidates for the global minimum. To do this, both combinations must have relatively similar productivities for each task; otherwise another combination will have a curve below the crossing curves. For example, if combination 1 assigns OPN1 to Task 1 with a high daily productivity, and combination 2 assigns OPN2 to Task 1 with a low productivity, then a hybrid combination, similar to combination 2 but with OPN1 doing Task 1, will probably be cheaper than either combination 1 or 2.

Physically, this means that although they may cross as in Figure 3.9, a combination dominated by humans and one dominated by automated equipment are not likely to be at the bottom of the design space. Since humans will be more productive in some tasks, and automated equipment more productive in other tasks, a hybrid combination of humans and automated equipment will probably be cheaper.

A more likely problem situation would involve one combination dominated by humans (Ch = 238, Hh = 0.4 in Chapter 4's case example) and a combination dominated by teleoperators (Ct = 60, Ht = 1.6). The teleoperators might have hourly productivities one-half those of the humans, but they might work twice as long each day to make the daily productivities roughly equal. As it turns out, this situation does not occur in the study's four-task case example: plots of the ProgCost-Jobtime curves show a number of crossings within the design space, but there is only one local (and global) minimum.

3.6.3 Enhancement of SOM to Handle Local Minima:

Since the plotting of 59,049 curves for a ten-task three-option case would be excessive, the Staged Optimization Method can be protected against multiple local minima by applying the method more than once, with different starting points. First, values of Jobtime are selected to bracket the likely range of minima (e.g. Jobtime = 10 and 800 days). Then the SOM is applied, starting at step 4 (described in section 3.5.1), with the first value of Jobtime. Steps 4 and 5 find the lowest curve at that Jobtime, and the SOM continues on to the minimum. This is repeated for each starting value of Jobtime in turn, and the resulting minima are compared to select the lowest.

For example, in Figure 3.9 two runs of the SOM would be sufficient: the first, from Jobtime = 10, would find the OPN2 minimum; the second, from Jobtime ≈ 800, would find the OPN1 minimum. Since the two minima are separate, the method could be applied a third time with a Jobtime value between the minima, to check for a possible third minimum in the middle.

Thus it is not difficult to find all the local minima, because the SOM operates in a design space of one-parameter equations for ProgCost, whose minima are relatively easy to locate. The constrained shapes of the ProgCost-Jobtime curves are also an advantage, because they limit the locations where minima can occur.

In summary, it is possible to have more than one local minimum in the SOM's design space, which might trap the SOM in a non-optimal solution. However, the range of physical situations leading to this problem is narrow: it requires combinations of all the human-machine options, with similar daily productivities for each task, yet with "averaged" number-of-units and usage cost factors sufficiently different to cause a crossing behavior in their ProgCost-Jobtime curves. The values of those parameters must position the crossing curves with separate minima, and these must be minima for the entire design space.

If such a situation does occur, it can be dealt with easily by applying the SOM several times from different starting points. This identifies the local minima, which are then compared to select the overall solution. Some logic can be added to the SOM to decide where more starting points should be added to guarantee finding all the local minima.

3.7 Effect of Study Assumptions on the SOM:

3.7.1 Summary of Major Assumptions:

The following descriptions summarize the major assumptions in this study, i.e. those which may affect the application of the SOM to actual space scenarios.

Assumption 1:
Future space scenarios are modeled as jobs consisting of individual tasks. All of a job's tasks start simultaneously, and run concurrently until individual tasks end.

This assumption is mentioned in section 2.1, particularly 2.1.5. The tasks are scaled by making each task consist of "repetitions" of the task; for example, the task "Install a component" might consist of the installation of 600 identical components, with each component's installation being one repetition of the task.

The concurrency of the tasks is an important assumption in this study. It affects the formulation of the general cost equation and the development of the SOM (particularly the theorem of equal task times). As a result, it limits the application of the SOM to a certain class of problems. These limitations are described in section 3.7.2. Some suggestions for extensions of the SOM beyond this assumption are presented in section 3.9.

Assumption 2:
Each task is performed by one or more human-machine options.

This study examines human-machine tradeoffs by modeling specific human-machine options (e.g. humans, teleoperators, and automated equipment), and then selecting the optimal assignments of these options for each task. The options are scaled as units (e.g. one human is a unit), and modeled by descriptive inputs (e.g. mass, productivity, consumables requirement).

Therefore, to apply the SOM to a space scenario, human-machine options for that job's tasks must be defined and modeled. This is not restrictive, because any number of options can be handled, and they include minor variations of a design (e.g. several types of teleoperators, with slightly different characteristics).

Assumption 3:
The available numbers of each option are unconstrained.

This study tackles a design problem: for a future space scenario, what are the minimum-cost assignments of humans and machines to tasks? Therefore there are no limits on workforce sizes. Whatever the optimal numbers of humans and machines are, those numbers will be put into orbit to perform the job at minimum cost.

As a result, the present SOM cannot be applied to a job with limited availability of humans or machines. Such scenarios pose a fundamentally different theoretical problem. However, as will be discussed in section 3.9, the extension of the SOM to include such limits is a likely future step in development.

Assumption 4:

Fractional workforces can be assigned to tasks.

This assumption is mentioned in section 2.2.1 and discussed in section B.4.4. For any individual task, fractional numbers of human-machine options can be assigned (e.g. 3.2 humans). These represent units of options which share their work time between more than one task. Because the task completion times are calculated from the workforce sizes, fractional task times are also acceptable.

This assumption allows fractional values for optimal results. This considerably simplifies the optimization process, by avoiding integer requirements. If integral results are required, an extension to the SOM can take the fractional results and round them, or examine nearby integral solutions to determine the true minimum. This rounding process assumes that the fractional solution is close to the integral one; physically, this is more likely to be true in large jobs (e.g. with option numbers around ten or more), where the percentage difference between fractional and integral results is small.

Assumption 5:

The intrinsic characteristics of each human-machine option are <u>independent</u> of the parameters of other options, and of the assignment of other options to tasks.

This assumption is discussed in detail in section 3.2.2. It states that the values of the parameters describing any human-machine option (e.g. productivity, consumables requirement, procurement cost) depend only on the design of the option and (in the case of productivity) on the task being performed. They are therefore unaffected by the characteristics of the other options. In particular, for a given task, one option's productivity is fixed, whether or not other options are assigned to the task.

In general, conflicts with this assumption are unlikely, because the parameters in the study's models are intrinsic to the individual options (for example, the mass of a teleoperator is unlikely to be affected by the designs of other options). The exception is productivity, which may be enhanced by partnerships of options. However, as described in section 3.2.2, productivity dependences can be resolved by careful definitions of the tasks, possibly splitting some tasks into subtasks.

Other dependences can be handled by the same technique, or by adding some internal restrictions to the SOM, preventing the method from violating the interactions. In some cases the SOM can tolerate some coupling of parameters, if they only affect some steps of the method. Learning curves in procurement are examples of this, as will be shown in Chapter 4. Only in rare instances would a scenario's interactions require a major modification to the SOM.

In addition to the major assumptions above, a number of minor assumptions are made during the estimation of numerical values of model inputs for the study's case example. These are described in Appendices C, D, and E.

3.7.2 The Concurrent-Tasks Assumption:

This study's cost model (and therefore its general cost equation) assumes that all the tasks begin at the same time and run concurrently until individual tasks end. At face value, this assumption appears to limit the application of the SOM to jobs which can be modeled as concurrent tasks. Practically, however, there is some leeway within this restriction.

First of all, the assumption does not require that all tasks be performed all the time. Within a working day, a task may be worked on intermittently, provided that the inputs to the model take these interruptions into account. For example, a human fastening components at the rate of one every 20 minutes (3 reps/pers-hr) might have to stop for 10 minutes between each component (possibly to wait for another component to be positioned). Then the human's true productivity is 2 reps/pers-hr. As long as the cost model uses that value, the idle times are not a violation of the assumption.

On a larger scale, the assumption permits shifting of some workforces between tasks, possibly to complete one task before another. This can be shown graphically, using some hypothetical examples in Figures 3.10 and 3.11.

Figure 3.10 is based on Figure 3.3, showing the cost elements for tasks which all begin together. By the theorem of equal task times (in section 3.3), in the optimal solution these tasks all end at the same Jobtime. In this example, it is assumed that:

Jobtime = 10 days

Task 1 consists of 100 repetitions
Task 2 consists of 200 repetitions

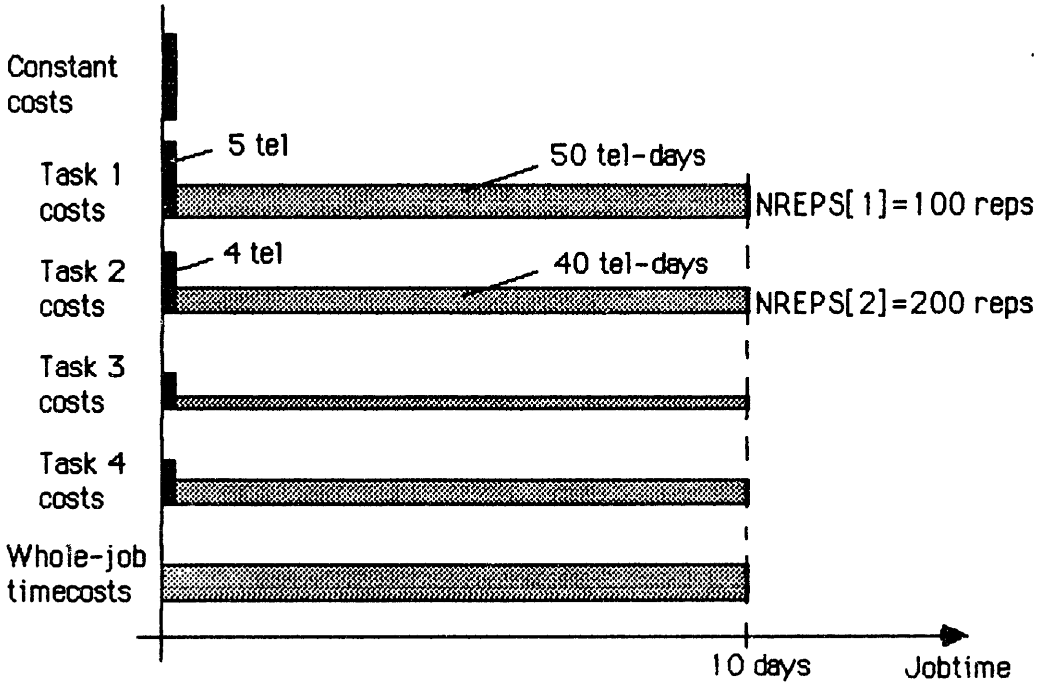


FIGURE 3.10: SCHEMATIC OF OPTIMAL COST ELEMENTS IN HYPOTHETICAL FOUR-TASK CASE

It is also assumed that Tasks 1 and 2 are optimally performed by teleoperators, with daily productivities:

ODtel[1] = 2 reps/tel-day
ODtel[2] = 5 reps/tel-day

Therefore Task 1 requires 5 teleoperators and 50 tel-days to complete, while Task 2 needs 4 teleoperators and 40 tel-days.

Figure 3.11 shows the effect of transferring Task 1's five teleoperators to Task 2 at the beginning of the job. The enlarged crew of 9 tels has a production rate of 45 reps/day, and therefore finishes all of Task 2 in 4.44 days. Then Task 1's five tels are returned, and Task 2's four tels are loaned to Task 1. The 9 tels now have a production rate of 18 reps/day, and complete Task 1 in the remaining 5.56 days.

The workforce sizes, tel-hours for each task, and total costs are the same in both figures. Therefore the SOM's optimal human-machine assignments in Figure 3.10 are also the solution to Figure 3.11, in which Task 2 is completed before Task 1 begins.

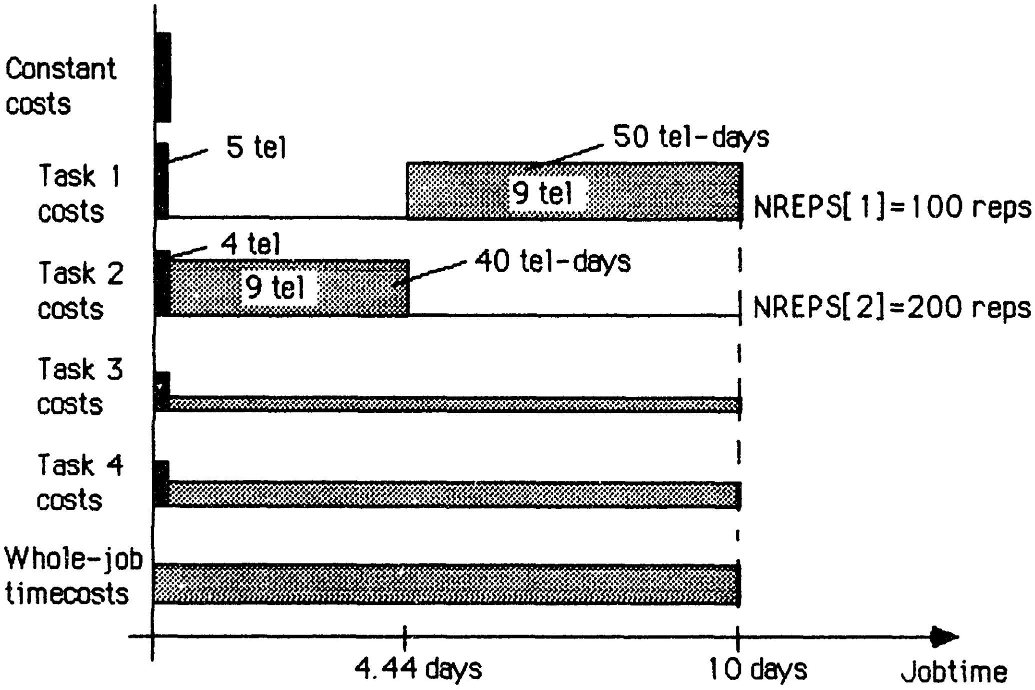


FIGURE 3.11: COST ELEMENTS WITH TWO SEQUENTIAL TASKS

Thus if a task must precede another, under certain circumstances both tasks can be modeled as concurrent tasks and optimized by the SOM. Then their workforces can be shifted around without compromising the solution. The special circumstances are that both tasks are optimized by the same human-machine option. Therefore both workforces have the same productivities for each task, and can therefore "repay" any workforce loans with the same effectiveness.

On the other hand, if the two tasks were performed by different options, this give-and-take might not work. For example, if humans from Task 3 are shifted to Task 2, Task 2 will be completed earlier; but if Task 2's teleoperators are unable to do Task 3, they will not be able to repay the humans' help, and Task 3 will not be completed in Jobtime.

Despite this leeway, the concurrent-tasks assumption is the strongest restriction in the application of the SOM to space scenarios, because it prevents the current SOM from handling most sequential-task problems. Section 3.9 will discuss possible extensions to the SOM to handle such situations.

Fortunately, the concurrent-tasks assumption covers a number of scenarios of interest. For instance, this study includes a case example of the in-orbit construction of a Microwave Radiometer (in Chapter 4). Many parts of the radiometer can be built separately, and some tasks can be started very shortly after others (for example, the alignment of feed horns can start after one horn has been installed). Therefore this job can be accurately modeled by concurrent tasks.

Similarly, the simultaneous operation of many experiments aboard a research module might be modeled as concurrent tasks. And parts of the maintenance schedule for an unmanned platform might fit this assumption, e.g. the simultaneous replacement of worn and faulty hardware, substitution of payloads, and replenishment of propellant, while the platform is docked to the Space Station. In general, any group of simultaneous tasks with the same starting time can be isolated and optimized by the SOM.

3.8 Advantages of the Staged Optimization Method:

This section describes the two principal advantages of the Staged Optimization Method:

1) its computational efficiency;
2) its ability to handle nonlinear cost functions.

Sections 3.8.1, 3.8.2, and 3.8.3 discuss the computational complexities of the original problem and of the SOM. In general, this complexity is driven by two parameters:

a) the number of human-machine options, z
b) the number of tasks to be done, NTSK

Section 3.8.4 discusses nonlinear cost functions.

3.8.1 Computational Complexity of the Original Problem:

As discussed in Chapter 2, this study's basic optimization problem involves a multi-parameter objective function. Its cost surface is not well behaved, having a multi-dimensional saddle-shape with many local minima. Therefore the problem resists traditional methods of optimization. Approximation methods appear possible, but they would require considerable computation to model the actual behavior accurately, and to identify the global minimum from the local minima.

The recognition of the combinational character of the problem (section 3.2) reduces that complexity, by reformulating the problem into a "design space" of potential solutions. Each of these potential solutions corresponds to one combination of assignments of human-machine options to the tasks. And each combination can be minimized by a one-parameter optimization.

For each combination of assignments, the equation to be minimized is a function of the single variable Jobtime. Since each combination has a fixed set of options, its complexity is not affected by the number of different options available. And the number of tasks to be done only adds terms to the equation, not extra variables. Therefore the computational complexity of the one-parameter optimizations is essentially independent of the numbers of options and tasks.

However, if z human-machine options are available for NTSK tasks, the total number of possible combinations of assignments is (z^{NTSK}), e.g. 81 combinations for a four-task three-option case, and 59,049 for a ten-task three-option case. Therefore the direct approach to the problem (minimizing each of the possible combinations and comparing the minima) would be a polynomial algorithm in z, but also an exponential algorithm in NTSK, leading to unacceptable computation loads for realistic scenarios.

3.8.2 Computational Complexity of the SOM:

As described in section 3.5, the Staged Optimization Method does not have to optimize all possible combinations. The SOM alternates selecting combinations of assignments and minimizing one-parameter equations. As described above, the one-parameter optimizations are not very complex, and not significantly affected by z or NTSK. Therefore the computational complexity of the SOM is driven by its selection process for combinations.

This selection process takes place in steps 4 and 5 of the SOM, as described in section 3.5.1. In that formulation, its complexity is not easy to evaluate. However, an alternative selection procedure has been suggested (Ref. 3-1), which yields useful insights into this complexity. For some scenarios, this procedure may also be more efficient than the one in section 3.5.

This alternative selection procedure considers each task separately, and makes use of the relationships between the individual task costs and the human-machine options. For example, if any task j is performed by humans, its task costs are (from section 3.4.2):

$$Ch * \frac{NREPS[j]}{ODhum[j]} * \frac{1}{Jobtime}$$

$$+ Hh * \frac{NREPS[j]}{ODhum[j]}$$

If the task is done by teleoperators, its task costs are:

$$Ct * \frac{NREPS[j]}{ODtel[j]} * \frac{1}{Jobtime}$$

$$+ Ht * \frac{NREPS[j]}{ODtel[j]}$$

For certain values of Jobtime, humans will be cheaper than teleoperators, and vice-versa for the other values. To find the "crossover" point between these two conditions, the two expressions are set equal to each other. Rearranging the terms:

$$Jobtime = \frac{(Ct * ODhum[j]) - (Ch * ODtel[j])}{(Hh * ODtel[j]) - (Ht * ODhum[j])}$$

Similar expressions identify the crossover points between humans and automated equipment, and between teleoperators and automated equipment. Taken together, these points separate Jobtime into regions: in each region, one of the human-machine options is cheapest for Task j.

This result is shown graphically in Figure 3.12. The Task j line is split into 3 regions by the HUM-TEL and TEL-AEQ crossover points. The HUM-AEQ point is inside the TEL region, and therefore redundant. In general, z options will lead to:

$$\binom{z}{2} = \frac{z!}{2! * (z-2)!} = (0.5) * (z^2 - z)$$

crossover points between options, but Jobtime will be split into at most z regions by (z - 1) of those points.

Therefore this process requires an algorithm to calculate the crossover points, and to decide which are region boundaries. A clever algorithm may be able to avoid calculating some of the points, based on the data it has already developed. In addition, the expression for Jobtime above can take on negative or (less likely) infinite values, pushing some options entirely out of the feasible range of Jobtime (e.g. Task 2 in the figure). In the worst case, however, the algorithm will have to calculate $(0.5) * (z^2 - z)$ crossover points for each of NTSK tasks. Therefore this is a polynomial algorithm in both z and NTSK.

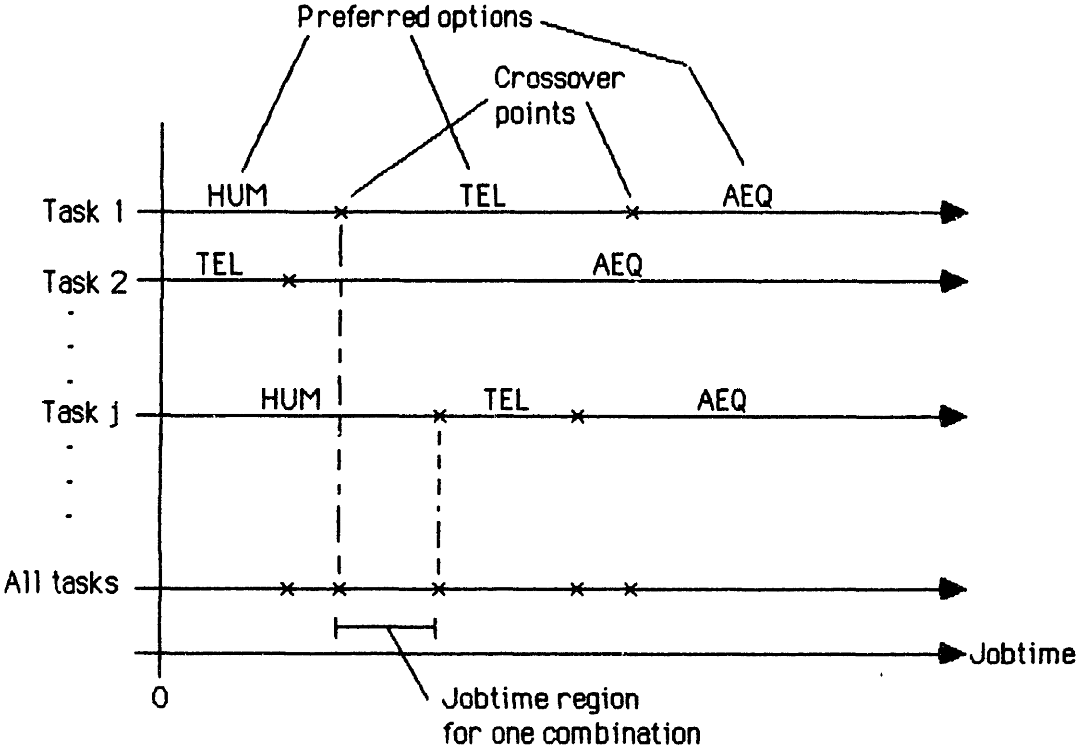


FIGURE 3.12: PREFERRED-OPTION REGIONS ALONG Jobtime LINES

For the selection procedure, the information in Figure 3.12 is stored in a lookup table. Collectively, the preferred-option regions for each task combine to form "preferred-combination" regions along Jobtime. This is shown graphically in Figure 3.12 by dropping the tasks' crossover points down to the "All tasks" line. The indicated "Jobtime region" along this line corresponds to one preferred combination of assignments (Task 1: TEL; Task 2: AEQ; ... Task j: HUM; ...).

Therefore the original SOM's steps 4 and 5 can be replaced by a simpler calculation. For a fixed value of Jobtime (provided by step 3), the lookup table is consulted to identify the preferred combination of assignments. That combination is then iterated back to step 2 by the method. In fact, both selection procedures are quite similar, but the alternative presented here does almost all of its calculations in step 1, beforo the iterative part of the SOM.

In both procedures, the SOM finds a preferred combination of assignments from a fixed value of Jobtime, and stops when this combination repeats one found earlier. Therefore, in the worst case, the SOM will pick one Jobtime value from each of the preferred-combination regions before finding the optimal solution. The maximum number of these regions is the sum of all the tasks' useful crossover points, plus one. For z options and NTSK tasks, this number is:

$$((z - 1) * \mathrm{NTSK}) + 1$$

Therefore the SOM is a polynomial algorithm in z and NTSK, a considerable improvement over the original problem.

This computational efficiency bodes well for the application of the SOM to actual space scenarios. The method is quick and cheap enough that it could be nested within parametric simulations. For example, the effect of teleoperator productivity on a construction scenario could be studied, by incrementing its value across a likely range, and applying the SOM at each increment. The results would show how the value of that productivity affects the optimal solution.

In the future, extended versions of this method may be implemented in computers at the worksite, allowing onsite personnel to respond to situation changes (e.g. changes in task definitions, or better data on productivities) by reoptimizing human-machine assignments in real time. Also, the cost consequences of proposed changes in hardware could be identified, helping decisions on their implementation.

3.8.3 Computational Complexity of the Reduced-Option Cases:

As described in section 3.5.3, the complete solution of the optimization problem requires applying the SOM to reduced-option cases. The number of these cases is a function of the number of options, z, and is given by:

$$\binom{z}{z} + \binom{z}{z-1} + \binom{z}{z-2} + \ldots + \binom{z}{2} + \binom{z}{1}$$

$$= 2^{z} - \binom{z}{0} \qquad \text{(Ref. 3-2)}$$

$$= (2^{z} - 1)$$

Therefore the reduced-option cases are exponential in z. However, two features of the SOM can significantly reduce this complexity.

First, the method can use data generated during the first application of the SOM to avoid having to consider all of the options. When the SOM is applied to the all-options case, its step 1 separates the individual tasks and calculates the minimum cost of each task for each of the options. If any Task j has one option OPN which is much cheaper than the others, then that option should not be removed in the reduced-option cases. Mathematically, the advantage of removing OPN is the elimination of its R&D cost Ropn. If the other options for Task j (optimally applied) lead to costs that are higher by more than Ropn, the reduced-option case cannot be an improvement.

Second, for those cases which must be considered, much of the data from the first SOM application can be carried over. The single-task optimizations to start the method are the same, with one or more options omitted. If the alternative selection technique described in section 3.8.2 is used, the new preferred-combination lookup tables are formed by subsets of the original crossover points. Therefore a substantial amount of calculation can be avoided in the reduced-option cases.

Another factor mitigating this exponential complexity is that although realistic scenarios are likely to include many tasks, their number of human-machine options are likely to be limited. For example, a scenario might consider two human options (extra-vehicular and intra-vehicular), two types of free-flying teleoperators, three types of fixed-base teleoperators, and four types of automated equipment, for a total of 11 options. In the worst case, this would lead to 2047 cases to be considered; if three of the options were so advantageous that they should not be removed, that number would drop to 128.

It should also be noted that the reduced-option cases make the original problem (section 3.8.1) not only exponential in NTSK but exponential in z as well. Therefore the SOM is a substantial improvement, since it is a polynomial algorithm in NTSK and a mitigated exponential algorithm in z.

3.8.4 Nonlinear Cost Functions:

Because a variety of techniques are available to minimize one-parameter equations, the Staged Optimization Method can be applied to cost functions which include nonlinear terms. By comparison, traditional optimization techniques are usually restricted to linear functions or specific types of nonlinear expressions (e.g. quadratic cost functions).

Of course, the nonlinearities must not violate the SOM's fundamental assumptions, described in section 3.7.1. And the theorem of equal task times must still apply, as explained in section 3.3.4, fourth note.

For the SOM, the easiest nonlinearities are those which are contained within individual tasks. For example, the productivity of humans in performing Task 1 might be a nonlinear function of the number of humans assigned to this task:

OHhum[1] = f(Nhum[1])

Then the cost terms for Task 1 would include nonlinear expressions in Nhum[1], but this nonlinearity would not affect the costs of any other task. Also, if humans were selected for Task 1, there would still an explicit relationship between the Task 1 completion time and the number of humans assigned to the task.

In this case, the SOM would be applied by including Task 1's nonlinear terms when ProgCost is defined in terms of Jobtime (step 2). To find the optimum Jobtime (step 3), an appropriate minimization technique would be selected: if the nonlinear equation is differentiable, setting its derivative equal to zero is sufficient; if the nonlinearity is more complex, a gradient-search method can be used.

When the problem is decomposed into individual tasks (step 4), the Task 1 costs include the nonlinear terms for the human option. Because the effect of these terms is limited to Task 1, the separated tasks are still an accurate representation of the problem. Therefore the SOM proceeds to the optimal solution, with its accuracy unaffected by the nonlinearity.

Nonlinearities which apply across more than one task can be more difficult. For example, a learning curve in the procurement of teleoperators has the form:

$$(\text{First-unit cost}) * (1^{-q} + 2^{-q} + \ldots + \text{NtotTEL}^{-q})$$

where

NtotTEL = (Ntel[1] + ... + Ntel[j] + ...)   over all j.

Therefore the cost per teleoperator is a function of the numbers of teleoperators assigned to all the tasks.

Steps 2 and 3 of the SOM are not affected: a nonlinear equation for ProgCost is formed, and ProgCost is minimized with respect to Jobtime by a gradient-search method (differentiation will not handle the learning curve series). In step 4, however, when the problem is decomposed into individual tasks, the teleoperator procurement cost for each task must be modeled without reference to variables from other tasks. If the learning curve is applied within each task, e.g. for Task j:

$$(\text{First-unit cost}) * (1^{-q} + 2^{-q} + \ldots + \text{Ntel}[j]^{-q})$$

then these cost terms will probably be high, because the learning curves are applied to subsets of the whole production run. So during steps 4 and 5 the SOM's model of the problem is not entirely accurate, and some attention must be paid to the method's behavior.

Fortunately the method is iterative, and therefore tends to converge to the optimal solution, even if some of its steps rely on approximations of the problem's costs. In the case of procurement learning curves, the discrepancies between applying them to individual tasks and to the entire job are not large, and therefore the separate tasks in steps 4 and 5 are close enough to the actual cost behavior. In Chapter 4, the SOM will be applied to examples which include procurement learning curves.

3.9 Possible Future Extensions of the SOM:

It may be possible to extend the Staged Optimization Method to cover the sequential-task problem, i.e. scenarios in which certain tasks must be completed before others can begin. This might be done by modeling the problem as successive SOM problems, as illustrated in Figure 3.13.

The figure shows a job in which Task 1 must precede Tasks 3 and 4, but Task 2 is continuously performed throughout. Conceptually, the situation can be modeled as two successive parts, the first ending at Jobtime1, the second running from Jobtime1 to Jobtime2. The durations of each part are independent of each other, so this example's total program cost would be a function of the two independent variables Jobtime1 and Jobtime2.

Therefore the extended SOM would have to include multiple-parameter optimizations, whose difficulty would depend on the character of the ProgCost equation. For example, if this equation turned out to be linear with respect to the Jobtimes, the optimization step in the SOM could use Linear Programming.

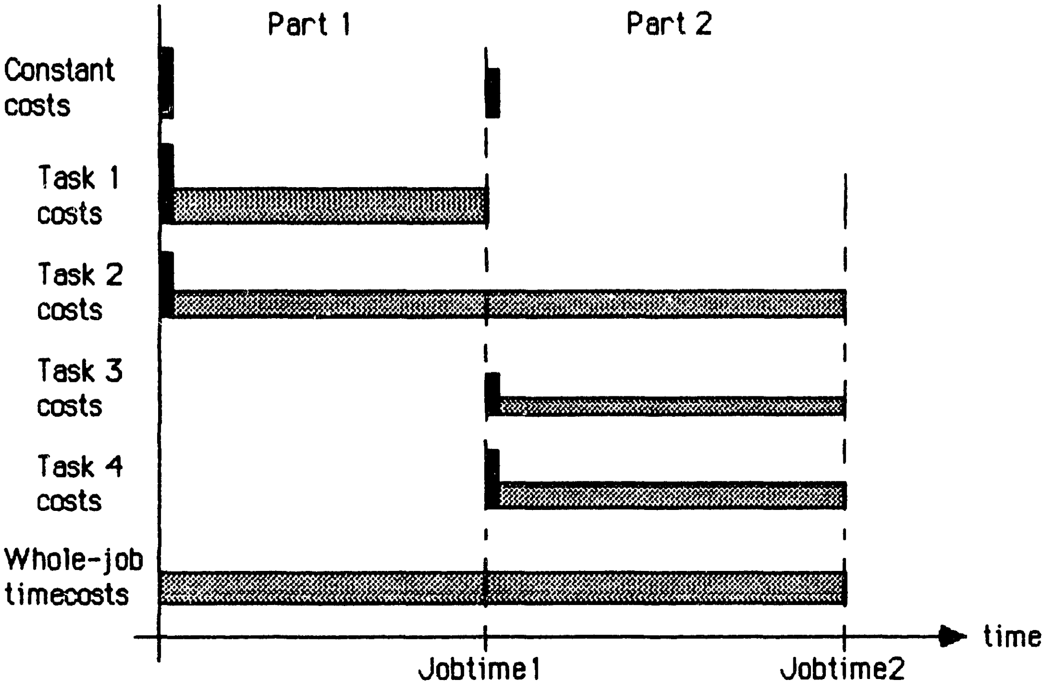


FIGURE 3.13: COST ELEMENTS IN SEQUENTIAL-TASK PROBLEM MODELED AS CONSECUTIVE SOM PROBLEMS

Within each of the successive parts, certain tasks can be optimized by the current SOM's techniques (e.g. Task 1 in the first part, Tasks 3 and 4 in the second). The whole-job timecosts are not a problem, because their cost rate Hw is unaffected by Jobtimel or Jobtime2. The constant costs are also not a problem, because they do not affect the optimization.

However, tasks which bridge both parts of the job (e.g. Task 2) require special handling. One difficulty is deciding how many of the task's repetitions are performed in part 1 and how many in part 2. If Task 2's workforce remains unchanged for both parts (a possible optimum situation, if the same option is cheapest in both parts), then the numbers of repetitions are direct functions of Jobtimel and Jobtime2. Therefore the Task 2 usage costs could be expressed in terms of those variables.

The most difficult aspect of the problem is likely to be the potential reassignment of workforces from part 1 to part 2. In the example above, returning the Task 1 workforce to the ground at Jobtimel is a waste of resources. Instead, those units can be reassigned to Tasks 3 or 4, eliminating some of the number-of-units costs in part 2.

Therefore a complete optimization of the problem requires making the idle workforces (at the end of part 1) available to the tasks in part 2. This reassignment probably requires a combinatorial optimization nested within the overall combinatorial problem.

It may also be possible to tackle this reassignment by defining the idle workforces as different human-machine options, with no change in their usage cost factors, but with their numbers-of-units costs set to zero (since they are already procured and onsite). Then the SOM techniques might be applied to part 2 with the extended set of human-machine options.

This approach would still require an overall objective function including both parts of the job. Also, the new options for part 2 would have limited availability, i.e. their numbers would be constrained by the sizes of some workforces in part 1. Therefore the SOM would have to be modified to deal with limited resources in some of its options.

In general, the extension of the SOM to include limited resources is a likely next step. Besides contributing to the development of the sequential-tasks optimization method, this would also adapt the SOM to operations at a fixed-size space facility, such as a Space Station with limited crew size (e.g. only 6 humans).

Conceptually, this extended SOM would replace the option selections in steps 4 and 5 with an Assignment Problem, in which limited resources would be optimally assigned to tasks, with constant costs for each assignment path. (These costs are constant because of the fixed value of Jobtime at that point in the iteration.) The resulting assignments would include more than one option for some tasks, which would in turn require changing the algorithms in steps 2 and 3 of the method.

In summary, the Staged Optimization Method offers techniques which may be extendable to more complex problems, such as scenarios including sequential tasks. However, these extensions to the SOM would require a significant amount of further theoretical development, beyond the scope of this study. A likely next step would be the extension of the SOM to include resources with limited availability.

CHAPTER 4:

CASE EXAMPLE: CONSTRUCTION OF A MICROWAVE RADIOMETER

This chapter demonstrates the study's line-item cost model and Staged Optimization Method by applying them to a case example: the on-orbit construction of a Microwave Radiometer.

Section 4.1 presents an abbreviated description of the Microwave Radiometer and its construction tasks, referring to Appendix C for details. Four-task and ten-task examples are formulated from this scenario. Section 4.2 describes the human-machine options to perform the construction, referring to several appendices for detailed designs. The productivities of these options are estimated.

The line-item cost model is applied to the four-task case example in section 4.3, showing the individual cost elements. In this line-item costing, the numbers of humans and machines assigned to each task are arbitrary, and therefore the overall program cost is not at its minimum value.

To minimize that program cost, the Staged Optimization Method (SOM) is applied to the same four-task example in section 4.4. The optimization process is shown step-by-step, illustrating the iterative selection of options for the tasks and the calculation of the minimumcost solution.

Finally, in section 4.5 the SOM is applied to examples including procurement learning curves, to illustrate the method's application to nonlinear cost functions. Both four-task and ten-task examples are presented.

4.1 Microwave Radiometer and Construction Tasks:

4.1.1 Description of Radiometer:

The Microwave Radiometer in this study's case example was defined by the NASA Langley Research Center in 1981, as "a potential future space mission requiring large space structures technology" (Ref. 4-1). The purpose of the Microwave Radiometer is to map the earth's soil moisture content from space, to aid in global crop forecasting.

The operating orbit of the radiometer is between 650 and 1000 km in altitude, and at least 60 degrees in inclination. However, the assembly of the radiometer is expected to happen in an orbit easily reachable by the Shuttle (e.g. 370-km altitude, 28.5-degree inclination).

The chosen baseline configuration (from Refs. 4-1 and 4-2) is sketched in Figure 4.1. The radiometer consists of four main components:

1) a 725-m diameter collecting antenna;
2) a stiffener ring surrounding the antenna;
3) a linear array of 600 feed horn assemblies;
4) two 580-meter support columns extending from the stiffener ring to the feed horn array.

Appendix C presents more detailed descriptions of these components, including a mass breakdown of the entire structure. From these descriptions, the Radiometer is modeled physically as 3,804 units of low-technology hardware and 600 units of high-technology hardware. Appendix C estimates the masses of these units, their packing densities during launch, and the learning curve parameter in their procurement. These quantities are inputs required by the line-item cost model.

4.1.2 Construction Tasks:

Appendix C also defines ten construction tasks to assemble the Microwave Radiometer, based on the descriptions in Refs. 4-1 and 4-2. For this chapter's illustrative case example, four of the ten tasks are selected; they are presented in Table 4.1, and described below. (All ten tasks will be used in the more complex example in Chapter 5.)

TABLE 4.1: FOUR-TASK CASE EXAMPLE

Task:		Number of repetitions
1:	Manufacture of truss beams	16,688
2:	Assembly of tetrahedral truss	3,728
3:	Assembly of membrane and surface controls	600
4:	Installation of feed horn assemblies	600

Task 1 is the manufacture of graphite/epoxy truss beams from as-delivered stock (e.g. rolls of pre-preg or stacked components). This task nominally produces 18-meter truss beams, but with size variations as needed. Each beam produced is a repetition of the task.

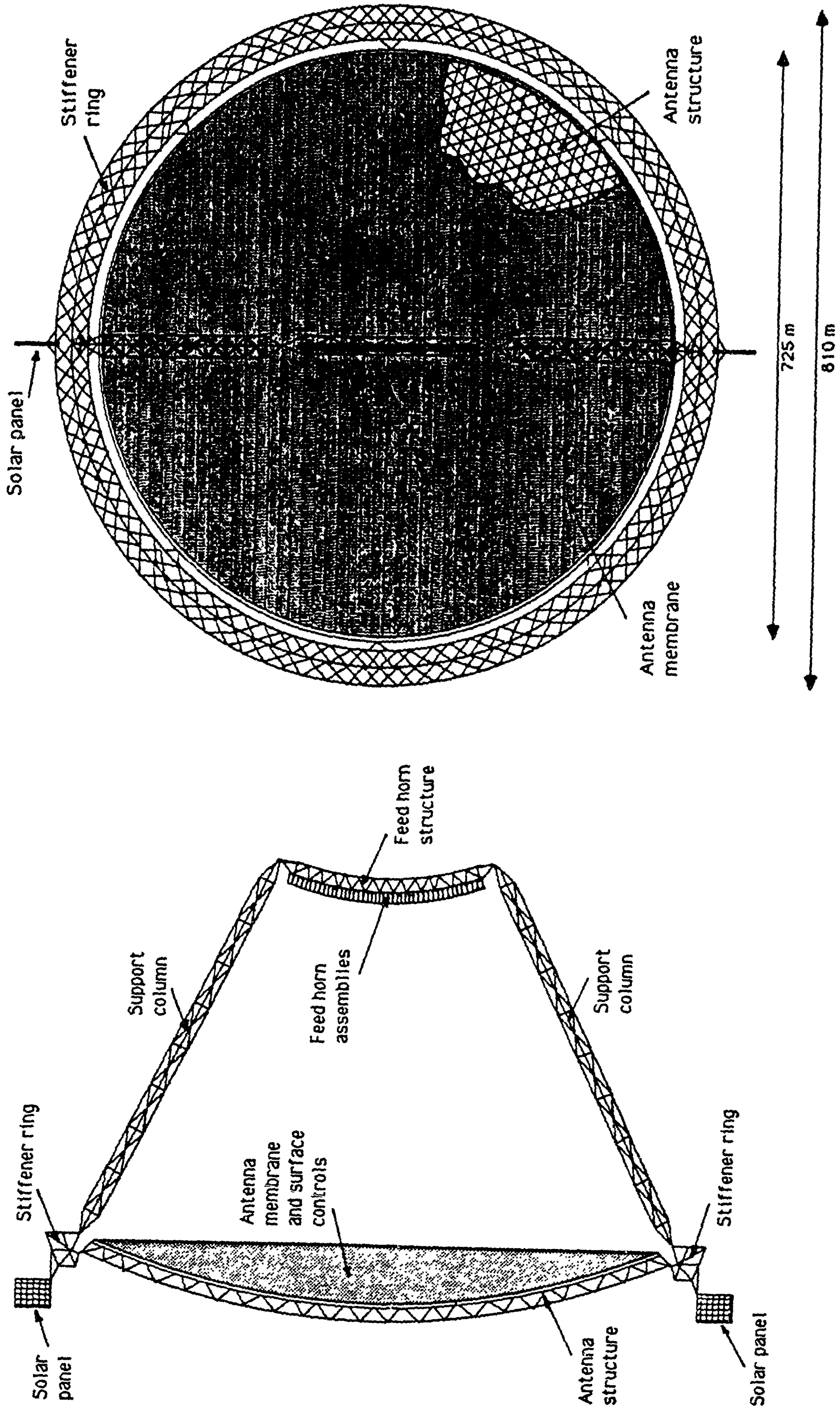


FIGURE 4.1: MICROWAVE RADIOMETER TO BE ASSEMBLED

Task 2 is the assembly of tetrahedral trusses for the antenna, the stiffener ring, and the feed horn structure. These trusses are made up of truss beams and node fittings, and one repetition of the task is defined as the attachment of beams to one node fitting. Therefore there are as many repetitions as node fittings (as listed in Appendix C, Table C.2).

Task 3 is the assembly of the antenna membrane and associated surface controls to the antenna truss structure. For the physical model of the radiometer (in Appendix C), it is assumed that the antenna membrane consists of 600 units. This assumption is carried over to this task definition. The installation of one such unit is a repetition of the task.

Task 4 is the installation of the feed horn assemblies into the feed horn truss structure. This involves the structural connection of the horns through their attachment fittings, and the electrical connection of power and data cables. The installation of each assembly is a repetition of this task.

4.1.3 Human-Machine Productivities:

The case example includes three human and machine options to perform the construction tasks:

1) humans in extravehicular activity;
2) free-flying teleoperators;
3) automated beam-builders.

These options handle the tasks listed in Table 4.1 with various levels of productivity. Appendix C presents estimates of these productivities, based on the task definitions, on past simulations of construction activities, and on the equipment designs in Appendix D.

Since the estimation of productivity for future hardware is uncertain, Appendix C shows most of these productivities as a range of values straddling a likely "nominal" value. These nominal values are collected in Table 4.2.

As discussed in Appendix C, the (0.001) productivities indicate that automated equipment is not suitable for some tasks. These values are not set to zero, to avoid divide-by-zero problems in later calculations.

TABLE 4.2: ESTIMATED "NOMINAL" PRODUCTIVITIES FOR FOUR-TASK CASE EXAMPLE

Task:	Human productivity (reps/pers-hr)	Teleoperator productivity (reps/tel-hr)	Auto. Equip. productivity (reps/aeq-hr)
1	1.3	1.0	3.7
2	3.8	2.7	0.001
3	0.33	0.13	0.001
4	5.6	4.8	0.001

4.2 Construction Equipment Designs:

4.2.1 Description of Equipment:

In this study's case example, the human option consists of humans in extravehicular activity, using extravehicular mobility units (pressure suits) and manned maneuvering units. The humans at the worksite also require habitation modules. All of this hardware includes support equipment (e.g. power systems), and requires spare parts and consumables.

The teleoperator option consists of a free-flying construction teleoperator and a teleoperator support station. The support station includes power systems and storage for spare parts and consumables.

The automated equipment option is represented by an automatic beam-builder, which is capable of producing a variety of truss beam designs. This beam-builder includes a power generation system, and it requires spare parts but no consumables.

Preliminary designs of these pieces of equipment are presented in Appendix D. These designs include estimated mass breakdowns, duty cycles, consumables and spare parts requirements, and (in some cases) R&D cost estimates. Some of these parameters are expressed as ranges of values, to account for uncertainties in predictions of future technology.

Appendix D also includes an analysis of the atmospheric drag on the orbital construction site, and a preliminary design of the worksite propulsion system. These are combined to size the stationkeeping propellant requirement.

4.2.2 R&D and Procurement Cost Factors:

The line-item cost model calculates R&D and procurement costs for most of the space equipment by using dollar/kilogram cost factors for high-technology and low-technology hardware. Since R&D and procurement costs are a substantial fraction of overall program costs, these cost factors are important parameters in the study.

Therefore Appendix D presents the results of an extensive review of past NASA hardware, including communications, astronomy, earth resources, and planetary spacecraft, and the Shuttle Orbiter and Spacelab. The program costs of these spacecraft are adjusted into R&D and procurement costs, and these are plotted as functions of spacecraft mass. Because of the considerable scatter in this data, the dollar/kilogram factors are estimated as ranges of values. Regression analyses are used to identify the likely "nominal" values:

Nominal R&D cost factors:
High-tech hardware: 130,000 $/kg (in 1984$)
Low-tech hardware: 13,000 $/kg (in 1984$)

Nominal procurement cost factors:
High-tech hardware: 40,000 $/kg (in 1984$)
Low-tech hardware: 4000 $/kg (in 1984$)

4.3 Line-Item Cost Model of Four-Task Case:

To develop a physical understanding of the human-machine tradeoffs in space activities, the line-item cost model is applied to the four-task case example (from Table 4.1). This section reviews this example's inputs, intermediate outputs, and line-item costs. Section 4.4 will aply the Staged Optimization Method to the same example.

4.3.1 Inputs to the Cost Model:

The values of the inputs to the line-item cost model are presented in Appendix E, including some input values defined in Appendices C and D. Some of these values are single numbers, and some are estimated ranges. Almost all of the input values are defined or estimated from the construction scenario to be performed, i.e. from task descriptions, equipment designs, and analogous past space activities.

The exception is the numbers of humans, teleoperators, and units of automated equipment assigned to each task. In the line-item cost model, these values are arbitrary, i.e. they would be assigned by the Program Manager in a real space project. For this case example, values for these inputs were selected to be realistic possible assignments, but also to illustrate some features of the cost modal.

These values for Nhum[j], Ntel[j], and Naeq[j] are presented in Table 4.3. Some of the numbers assigned reflect the options' nominal productivities. For example, in Task 1 the automated equipment has a higher productivity than the humans or the teleoperators; therefore more units of automated equipment are assigned to the task.

TABLE 4.3: NUMBERS OF HUMANS AND MACHINES ASSIGNED TO EACH TASK

Task number [j]	Nhum[j] (pers)	Ntel[j] (tel)	Naeq[j] (aeq)
1	0.6	0.4	4.3
2	0.5	0.4	0.3
3	1.2	1.0	0.2
4	0.4	0.3	0.0

As mentioned in section 4.1.3, the automated equipment is unsuitable for tasks 2, 3, and 4. To show the cost model's response to unproductive use of hardware, some low numbers of automated equipment units are assigned to these tasks. For task 4, the automated equipment assignment is (0.0), to show how the cost model responds to zero values in such assignments. (Such values are acceptable; it is zero values for productivities which lead to problems.)

4.3.2 Intermediate Outputs of the Cost Model:

When the line-item cost model is applied to these inputs, it first calculates the daily productivities shown in Table 4.4. For example, the daily productivities of humans (in reps/pers-day) are the products of their hourly productivities and their daily work time (7 hrs/day).

TABLE 4.4: DAILY PRODUCTIVITIES AND TASK TIMES

Task	Daily Productivities				Task
#	Humans	Teleop.	Auto. Equip.	Crew	time
1	9.10	16.00	74.00	330.06	50.56
2	26.60	43.20	0.02	30.59	121.89
3	2.31	2.08	0.02	4.86	123.56
4	39.20	76.80	0.02	38.72	15.50

Longest task time Jobtime = 123.56 days

These values and the assignments in Table 4.3 are combined into the daily crew productivities (reps/crew-day) in Table 4.4. Dividing these into the total number of repetitions of each task yields the task completion times (days). The overall job time is the completion time for the longest task, i.e. for Task 3 in this example.

In the process of calculating line-item costs, the model computes a number of intermediate results of interest, presented in Table 4.5. The initial number of humans at the worksite is the sum of the human assignments in Table 4.3. Its value of 2.70 persons could be interpreted as 3 humans spreading their time between this example's tasks, but with one of the humans spending (0.3) of his time working on some other program. The number of habitation modules is one-third of the number of humans, with a similar interpretation.

The number of person-hours in space is for actual construction time in EVA. A related quantity is the total number of person-days in space, which is used to calculate the spares requirement for the habitation modules: spares equivalent to (0.016) of a habitation module are required during the program. The numbers of person-hours of work and person-days in space are also used to calculate the total consumables requirement (including MMU propellant) for the humans in space.

Similar calculations lead to the teleoperator values. Although there are fewer teleoperators at the worksite, they work longer hours (16 hrs/day) than the humans, and therefore turn in a larger number of tel-hours. Because the teleoperators have a relatively high consumption of propellant, their total consumables requirement is substantially higher than for the humans.

TABLE 4.5: SOME INTERMEDIATE RESULTS OF THE COST MODEL

Humans:
Initial number of humans = 2.70 pers
Total number of person-hours of space work = 1720.23 pers-hrs
Total number of person-days in space = 245.75 pers-days
Initial number of habitation modules = 0.900 habs
Total habitation module spares (in equiv. habs) = 0.016 habs
Total human-related consumables = 23,599 kg

Teleoperators:
Initial number of teleoperators = 2.10 tels
Total number of teleoperator-hours of space work = 3154.97 tel-hrs
Total teleoperator spares (in equiv. tels) = 0.095 tels
Total teleoperator consumables = 77,612 kg

Automated equipment:
Initial number of automated equipment units = 4.8 aeqs
Total number of auto.-equip.-hours of space work = 5573.75 aeq-hrs
Total automated equipment spares (in equiv. aeqs) = 0.167 aeqs

Total requirement for construction site propellant = 13,468 kg

Ground support:
Total number of people in ground support crew = 83.33 gsc
Total number of person-hours on the ground = 71,856 gsc-hrs
(during construction)

The automated equipment has a nominal daily work time of 20 hrs/day, and therefore turns in a large total of aeq-hrs. Its spares requirement is also pushed up by its high duty cycle. Automated equipment requires no consumables.

The requirement for construction site propellant is 13,468 kg, to offset the orbital drag for 123.56 days. However, the daily requirement could range up to five times higher during maximum sunspot activity; such sensitivities will be examined in section 4.3.4.

During the actual construction, the ground support crew numbers 83.33, and they spend 71,856 person-hours in monitoring the worksite, humans, teleoperators, and automated equipment.

4.3.3 Final Output of the Cost Model:

From the "nominal" values of inputs in Appendix E, and from the illustrative human and machine assignments in Table 4.3, this study's cost model computes line-item costs for the four-task case example. These costs are presented in Table 4.6. In subsequent discussions, these results are referred to as the "baseline" case.

TABLE 4.6: LINE-ITEM COSTS FOR FOUR-TASK CASE EXAMPLE
(Costs in $Million (1984$))

Total R&D cost (CTrd):	2525.476	
Space structure to be assembled (CTrdSST):		415.843
Human habitation modules (CTrdHAB):		1513.213
Human support equipment (CTrdHSE):		59.100
Teleoperators (CTrdTEL):		163.930
Automated equipment (CTrdAEQ):		343.330
Construction support equipment (CTrdCSE):		17.290
Task procedures (CTrdTSK):		1.270
Ground support equipment (CTrdGSE):		11.500
Total procurement cost (CTpr):	1776.461	
Space structure to be assembled (CTprSST):		633.671
Human habitation modules (CTprHAB):		439.087
Human support equipment (CTprHSE):		117.705
Teleoperators (CTprTEL):		107.174
Automated equipment (CTprAEQ):		465.444
Construction support equipment (CTprCSE):		5.320
Training of humans in space crews (CTprHUMt):		2.700
Salary and overhead for humans in space (CTprHUMb):		1.419
Ground support equipment (CTprGSE):		3.942
Total mission prep. and launch cost (CTla):	1229.678	
Space structure to be assembled (CTlaSST):		538.544
Launch of humans in space crews (CTlaHUM):		.851
Human habitation modules (CTlaHAB):		70.011
Human support equipment (CTlaHSE):		90.342
Teleoperators (CTlaTEL):		282.713
Automated equipment (CTlaAEQ):		52.474
Construction support equipment (CTlaCSE):		4.881
Construction site propellant (CTlaCSP):		47.138
Launch insurance (CTlaINS):		142.725
Total ground support cost (CTgs):	79.803	
Training of ground support crew (CTgsGSCt):		3.541
Salary & overhead for ground sup. crew (CTgsGSCb):		19.547
User fees for tracking and data relay (CTgsTDR):		56.715

Total program cost (ProgCost):	5611.418	

The estimated total program cost to perform the example's four tasks is 5611 $million in 1984$. Of that estimate, almost half is for R&D costs (2525 $M), and that fraction is dominated by the R&D costs of the human habitation modules (1513 $M). This large constant cost attached to the human option will play an important role in the final decisions of the optimization process (as will be shown in section 4.4).

In the total procurement cost of 1776 $million, the largest item (634 $M) covers the sizable quantities of space structure to be manufactured. Also expensive are the 4.967 units of automated equipment (465 $M) and the 0.916 habitation modules (439 $M).

Much of the total mission preparation and launch cost (1230 $million) is for the launch of the space structure to be assembled (539 $M). The launch cost for teleoperators (283 $M) is high because it includes the launch of their 77,612 kg of consumables (mostly propellant). The launch insurance figure (143 $M) applies to all of the hardware and consumables launched.

The ground support cost (80 $million) covers training and salaries for the ground support crew, and communications costs during construction. It should be noted that the other categories also include some ground-support-related costs, e.g. the R&D and procurement of ground support equipment, and the mission preparation of hardware at the launch site.

The baseline inputs and results presented above are not the minimum-cost solution to the four-task case example. This is because the numbers of humans, teleoperators, and automated equipment units in Table 4.3 are arbitrary examples, not the optimal values. It is possible to search for an optimum (or near-optimum) solution by running many line-item cases, systematically varying human-machine assignments and identifying cost trends.

For this four-task three-option example, twelve variables would have to be optimized during this process. In fact, in this simple case some of these assignments are obvious: for example, the unsuitability of the automated equipment for tasks 2, 3, and 4 suggests that no automated equipment should be assigned to these tasks.

However, in a more realistic construction scenario, the problem is considerably more difficult. For example, the full radiometer construction model involves ten tasks, and the options might include humans, fixed-base and free-flying teleoperators, and perhaps three types of automated equipment; in that case the number of variables to optimize would be sixty, and many of the choices between options might not be obvious.

4.3.4 Sensitivity Analyses of the Line-Item Costs:

To develop further the physical understanding of the problem, the cost-driving parameters of the case example were identified. This was done through sensitivity analyses: some of the cost model inputs were systematically varied across ranges of values to identify their impact on overall program cost. The effects of these inputs are graphed in Appendix F; one of these graphs is repeated here as an example.

Figure 4.2 presents the total program cost ProgCost as a function of the dollar/kilogram factor for procurement of high-technology hardware. To produce this graph, only the procurement cost factor is varied, through the range estimated in section D.2.1; the other model inputs have the baseline values listed in Appendix E. The procurement cost factor is normalized by dividing it by its baseline value (40,000 $/kg). At that value, the total program cost has the baseline case's value of 5611 $million.

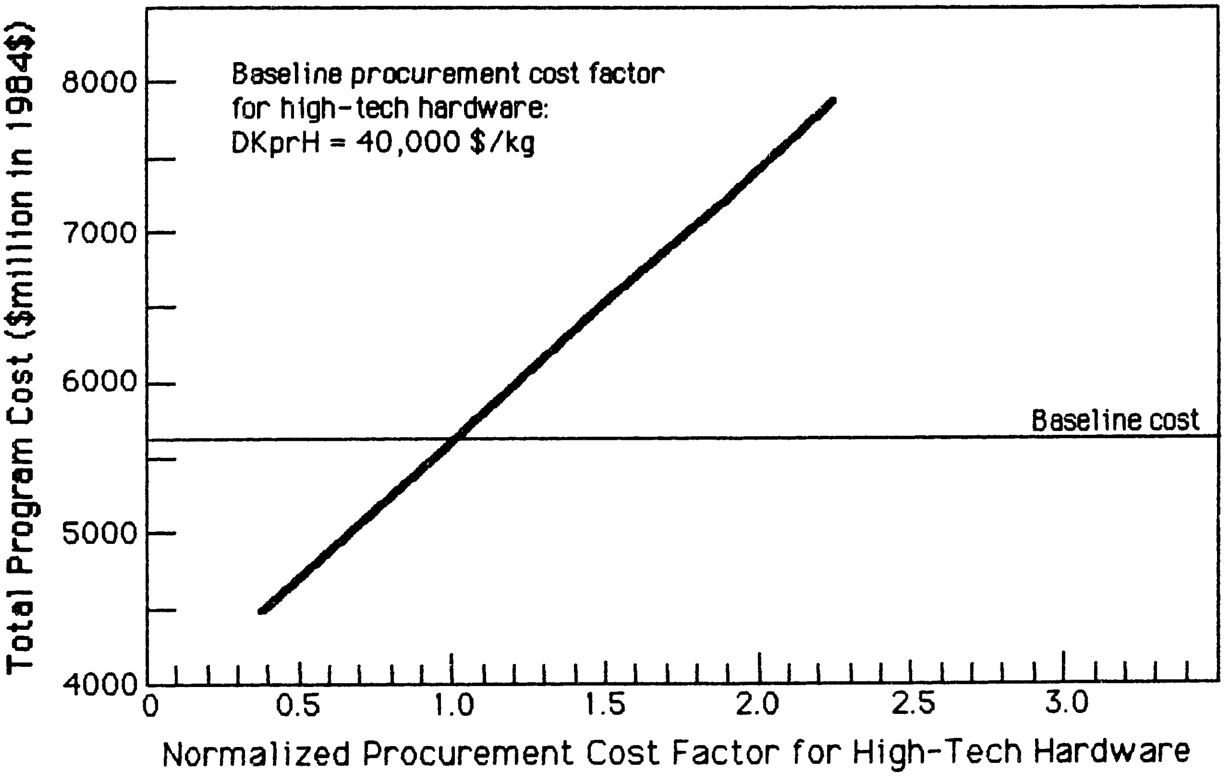


FIGURE 4.2: EFFECT OF HIGH-TECH PROCUREMENT COST FACTOR ON CASE EXAMPLE'S OVERALL PROGRAM COST

As the figure shows, this procurement cost factor has a strong impact on total program cost. This is because a substantial fraction of the overall cost is for procurement of high-tech hardware (e.g. parts of the radiometer and habitation module, the teleoperator, and the automated equipment), and because the range of uncertainty in the cost factor is large (the high estimate is 5 times the low estimate).

The parameters with strong effects on ProgCost (i.e. those causing 10% to 60% variations) are:

1) the R&D cost factor for high-tech hardware;
2) the procurement cost factor for high-tech hardware;
3) the per-kilogram launch cost on the Shuttle;
4) the mass of the habitation module.

In addition, the launch packing density for space structure can have a very strong effect, but only if it falls below 92.5 kilograms/cubic meter.

Other less important parameters (i.e. those causing 4% to 10% variations in ProgCost) include:

5) the R&D cost factor for low-tech hardware;
6) the mass of the teleoperator;
7) the consumables requirement for the teleoperator;
8) the mass of the automated equipment;
9) the requirement for construction site propellant.

The effects of productivities are more complex, because they are coupled to other parameters. Higher productivities in a task always lower the costs, but the extent of the decrease depends on the importance of the task in the overall job, the numbers of humans and machines assigned to it, and whether the task is the longest task in the job (e.g. Task 3 in the example). The effects of these and other parameters are displayed and discussed in Appendix F.

4.4 Application of the Staged Optimization Method to the Four-Task Case Example:

This section presents a step-by-step example of the use of the Staged Optimization Method. The SOM is applied to this study's four-task three-option case example of space construction. The computer programs used to implement the SOM are listed in Appendix I.

This example is illustrative: its purpose is to demonstrate the operation of the Staged Optimization Method, not to decide on the optimum techniques to assemble a Microwave Radiometer. As discussed in Appendices C, D, and E, a complete analysis of Radiometer construction would require firmer values for many of the model inputs.

4.4.1 Inputs to the Staged Optimization Method:

The Staged Optimization Method operates on the total program cost equation shown in Table 3.1. In that equation, the physical characteristics of the construction scenario are represented by the parameters Rw, Rh, Rt, Ra, Cw, Ch, Hh, Ct, Ht, Ca, Ha, Hw, NREPS[j], ODhum[j], ODtel[j], and ODaeq[j]. The values of these parameters are obtained from the equations in Appendix G, section G.4. With two exceptions, the inputs to those equations are the same as for the line-item cost model example, i.e. those listed in Appendix E.

The first exception is the input lcparam, the learning curve parameter for the procurement of hardware other than space structure: as discussed in section 2.2.4, the equations in section G.4 include the assumption that (lcparam = 100%), rather than the 90% used in the line-item model.

The second exception is that the parameters Nhum[j], Ntel[j], and Naeq[j] are undefined in the SOM example, since they are the decision variables to be calculated. The values for these quantities listed in Appendix E are therefore ignored.

The input values and the equations in section G.4 produce the parameter values shown in Table 4.7. One physical observation from these numbers is that the number-of-humans cost factor Ch is relatively high (238 $million/pers, compared to 60 $M/tel for teleoperators). This is due to the high cost for procurement and launch of the habitation modules. On the other hand, the teleoperator usage cost factor Hh is high (1.6 $M/tel-day, compared to 0.4 $M/pers-day for humans), because it includes launch costs for a sizable propellant requirement.

4.4.2 Starting Combination of Assignments:

Step 1 of the Staged Optimization Method is the selection of a starting combination of assignments of options to tasks. As described in section 3.5.2, this combination is selected by considering each individual task as a complete job.

For each task, three cases are formulated: humans alone, teleoperators alone, and automated equipment alone. Each of these cases is optimized, and the minimum-cost values of the relevant variables are displayed. For example, for the humans-only case, the optimal number of humans (HumNumber), the optimal completion time (HumTime), and the minimum cost (HumCost) are calculated.

TABLE 4.7: PARAMETER VALUES IN THE GENERAL COST EQUATION (FOR FOUR-TASK THREE-OPTION CASE EXAMPLE)

Whole-job R&D cost ($M)	Rw = 437.403
Human-related R&D cost ($M)	Rh = 1574.313
Teleoperator-related R&D cost ($M)	Rt = 168.430
Automated-equipment-related R&D cost ($M)	Ra = 345.330
Whole-job general costs ($M)	Cw = 1256.669
Number-of-humans cost factor ($M/pers)	Ch = 237.800
Human usage cost factor ($M/pers-day)	Hh = 0.439
Number-of-teleoperators cost factor ($M/tel)	Ct = 60.190
Teleoperator usage cost factor ($M/tel-day)	Ht = 1.592
Number-of-automated-equipment-units cost factor ($M/aeq)	Ca = 122.731
Automated equipment usage cost factor ($M/aeq-day)	Ha = 0.116
Whole-job timecosts ($M/day)	Hw = 0.569

Task	Number of reps NREPS	Human productivity ODhum (reps/pers-day)	Teleoperator productivity ODtel (reps/tel-day)	Auto. equip. productivity ODaeq (reps/aeq-day)
1	16,688	9.10	16.00	74.00
2	3,728	26.60	43.20	0.02
3	600	2.31	2.08	0.02
4	600	39.20	76.80	0.02

The results of this process are shown in Table 4.8. For example, the Task 1 numbers indicate that if that task is done by humans alone, its minimum cost (1801 $M) occurs when 2.1 humans perform the task in 875 days. The cheapest option for Task 1 is automated equipment, which can perform the task for 277 $M with 1.02 units in 220 days.

Comparing HumCost, TelCost, and AeqCost for each task leads to the following starting combination of assignments:

Task 1: AEQ
Task 2: TEL
Task 3: HUM
Task 4: TEL

TABLE 4.8: CALCULATED DATA FOR SELECTION OF STARTING COMBINATION OF ASSIGNMENTS

	If done by HUM		If done by TEL		If done by AEQ	
Task	HumNumber (HumTime)	HumCost	TelNumber (TelTime)	TelCost	AeqNumber (AeqTime)	AeqCost
1	2.10 (875.08)	1801.116	3.14 (332.02)	2038.586	1.02 (220.46)	277.264
2	0.58 (241.91)	337.012	0.90 (95.50)	246.156	29.41 (6338.10)	28850.370
3	0.79 (329.33)	489.037	1.65 (174.61)	658.096	11.80 (2542.71)	6377.534
4	0.19 (79.95)	97.771	0.27 (28.74)	45.166	11.80 (2542.71)	6377.534

Notes:
1) Costs are in $million.
2) Task times are in days.
3) Each task is modeled as an individual job, so each task's costs include (Hw * task time) costs.
4) Constant costs (e.g. R&D) are not included in the table.

As a check on the process, the line-item cost model was applied to the case example with the selected combination of assignments. The values of AeqNumber[1], TelNumber[2], HumNumber[3], and TelNumber[4] were entered as values for the inputs Naeq[1], Ntel[2], Nhum[3], and Ntel[4], respectively. The other numbers of units were set to zero, and lcparam was set to 100%. The line-item calculations confirmed Table 4.8's task times for the selected options. For the individually optimized tasks, the computed total program cost was 4643 $million.

By comparison, section 4.3.3's baseline line-item example had a ProgCost value of 5611 $million. Therefore Table 4.8's optimal numbers for HUM, TEL, and AEQ are clearly an improvement over the arbitrary selections in the earlier example. However, Table 4.8's numbers are not yet optimal, because each task has a different completion time. The SOM can reduce the program cost further.

4.4.3 Iterative Optimization:

In step 2 of the SOM (described in section 3.5.1), an equivalent cost equation is defined for the starting combination of assignments, by setting all of the task times equal to Jobtime:

$$\text{ProgCost} =$$

$$\left.\begin{aligned} &(\text{Rw} + \text{Rh} + \text{Rt} + \text{Ra} + \text{Cw})\\ &+ \frac{\text{Ha} * \text{NREPS}[1]}{\text{ODaeq}[1]} + \frac{\text{Ht} * \text{NREPS}[2]}{\text{ODtel}[2]}\\ &+ \frac{\text{Hh} * \text{NREPS}[3]}{\text{ODhum}[3]} + \frac{\text{Ht} * \text{NREPS}[4]}{\text{ODtel}[4]} \end{aligned}\right] \text{Constants}$$

$$\begin{aligned} &+ \frac{\text{Ca} * \text{NREPS}[1]}{\text{ODaeq}[1]} * \frac{1}{\text{Jobtime}}\\ &+ \frac{\text{Ct} * \text{NREPS}[2]}{\text{ODtel}[2]} * \frac{1}{\text{Jobtime}}\\ &+ \frac{\text{Ch} * \text{NREPS}[3]}{\text{ODhum}[3]} * \frac{1}{\text{Jobtime}}\\ &+ \frac{\text{Ct} * \text{NREPS}[4]}{\text{ODtel}[4]} * \frac{1}{\text{Jobtime}}\\ &+ \text{Hw} * \text{Jobtime} \end{aligned}$$

Step 3 is the minimization of ProgCost with respect to Jobtime. In this case example, this can be done by setting the derivative of ProgCost equal to zero:

$$((-1) * \text{Constant} * \text{Jobtime}^{-2}) + \text{Hw} = 0$$

where

$$\begin{aligned} \text{Constant} = &\frac{\text{Ca} * \text{NREPS}[1]}{\text{ODaeq}[1]} + \frac{\text{Ct} * \text{NREPS}[2]}{\text{ODtel}[2]}\\ &+ \frac{\text{Ch} * \text{NREPS}[3]}{\text{ODhum}[3]} + \frac{\text{Ct} * \text{NREPS}[4]}{\text{ODtel}[4]} \end{aligned}$$

So, for the starting combination of assignments, the minimum-cost value of Jobtime is:

$$\text{OptJobTime} = \text{sqrt}\left(\frac{\text{Constant}}{\text{Hw}}\right) = 408.66 \text{ days.}$$

In step 4, the completion time for all tasks is fixed at 408.66 days, and individual task costs are calculated for each of the three human-machine options. The resulting numbers are shown in Table 4.9.

TABLE 4.9: CALCULATED DATA FOR SELECTION OF IMPROVED COMBINATION OF ASSIGNMENTS (FIRST ITERATION: OptJobTime = 408.66 days)

	If done by HUM		If done by TEL		If done by AEQ	
Task	HumNumber	HumCost	TelNumber	TelCost	AeqNumber	AeqCost
1	4.49	1871.534	2.55	1814.043	0.55	93.897
2	0.34	143.031	0.21	150.091	456.12	77611.640
3	0.64	265.078	0.71	501.708	73.41	12491.140
4	0.04	15.621	0.02	13.588	73.41	12491.140

Notes:
1) Costs are in $million.
2) Constant costs (e.g. R&D) are not included in the table; (Rw + Rh + Rt + Ra + Cw) = 3782.145 $million.
3) Each task's costs do not include (Hw * OptJobTime) costs; (Hw * OptJobTime) = 232.729 $million.

This table resembles Table 4.8, but the task times have been omitted, since they are fixed at 408.66 days. Also, the cost numbers in the two tables are not directly comparable, since the costs in Table 4.8 include timecosts (Hw * task time), and the costs in Table 4.9 do not.

When OptJobTime is constrained to 408.66 days, the starting combination of assignments leads to a total program cost of:

ProgCost = 3782.145	(Rw + Rh + Rt + Ra + Cw)
+ 93.897	Task 1 costs
+ 150.091	Task 2 costs
+ 265.078	Task 3 costs
+ 13.588	Task 4 costs
+ 232.729	(Hw * OptJobTime)
= 4537.528 $million.	

This cost can be compared to the 4643 $M computed by the line-item cost model when the tasks were individually optimized (with different task times) in step 1. Thus fixing the completion times at 408.66 days has reduced the program cost by 105 $million, which gives indirect confirmation of the theorem of equal task times.

Step 5 is the comparison of options for each task, and the selection of the least expensive assignments. The data in Table 4.9 indicates that Task 1 can be performed in 408.66 days by 4.49 humans, or by 2.55 teleoperators, or by 0.55 units of automated equipment. Of these options, automated equipment is the cheapest. Similar comparisons for the other tasks lead to the <u>improved combination of assignments</u>:

Task 1: AEQ
Task 2: HUM
Task 3: HUM
Task 4: TEL

In the starting combination, the preferred option for Task 2 was teleoperators. In this new combination, however, that choice has shifted to humans. Physically, teleoperators were cheaper than humans when both options could be optimized independently, with different completion times; but when the completion times are fixed at 408.66 days, the humans are less expensive.

The process now iterates back to step 2, with the improved combination of assignments. The second iteration of step 2 produces a ProgCost equation similar to the one in step 2 above, but with the Task 2 terms switched to human versions.

The second iteration of step 3 calculates OptJobTime for the new combination of assignments:

OptJobTime = 465.20 days.

For that value of OptJobTime, the second iteration of step 4 produces individual task costs for each human-machine option, shown in Table 4.10. With OptJobTime constrained to 465.20 days, the improved combination of assignments leads to a total program cost of:

ProgCost = 4526.015 $million.

Thus the second iteration, by switching Task 2 to humans and recalculating the optimum Jobtime, has reduced program costs by a further 11.5 $million.

TABLE 4.10: CALCULATED DATA FOR SELECTION OF IMPROVED COMBINATION OF ASSIGNMENTS
(SECOND ITERATION: OptJobTime = 465.20 days)

	If done by HUM		If done by TEL		If done by AEQ	
Task	HumNumber	HumCost	TelNumber	TelCost	AeqNumber	AeqCost
1	3.94	1741.855	2.24	1795.375	0.48	85.667
2	0.30	133.120	0.19	148.547	400.69	70808.700
3	0.56	246.711	0.62	496.545	64.49	11396.250
4	0.03	14.538	0.02	13.448	64.49	11396.250

Notes:
1) Costs are in $million.
2) Constant costs (e.g. R&D) are not included in the table;
   (Rw + Rh + Rt + Ra + Cw) = 3782.145 $million.
3) Each task's costs do not include (Hw * OptJobTime) costs;
   (Hw * OptJobTime) = 264.924 $million.

The second iteration of step 5 selects the following minimum-cost combination of assignments:

Task 1: AEQ
Task 2: HUM
Task 3: HUM
Task 4: TEL

This is the same as the improved combination selected at the end of the first iteration. Therefore a third iteration is unnecessary, because it would duplicate exactly the results of the second iteration. The Staged Optimization Method has reached the optimal solution for this four-task three-option case example. The characteristics of this solution are displayed in Table 4.11.

4.4.4 Optimization of Reduced-Option Cases:

As discussed in section 3.5.3, it is not sufficient to apply the Staged Optimization Method to the three-option case. It must also be used on the reduced-option cases, e.g.:

Humans and teleoperators
Humans and automated equipment
etc.

TABLE 4.11: OPTIMAL SOLUTION FOR FOUR-TASK CASE EXAMPLE
(THREE HUMAN-MACHINE OPTIONS: HUM, TEL, AEQ)

Optimal selection of options:
Task 1: AEQ
Task 2: HUM
Task 3: HUM
Task 4: TEL

Task	Nhum	HumCost	Ntel	TelCost	Naeq	AeqCost
1	0.00		0.00		0.48	85.667
2	0.30	133.120	0.00		0.00	
3	0.56	246.711	0.00		0.00	
4	0.00		0.02	13.448	0.00	

Optimal Jobtime = 465.20 days.

Constant costs (Rw + Rh + Rt + Ra + Cw) = 3782.145 $million.
Whole-job timecosts (Hw * Jobtime) = 264.924 $million.

Minimum total program cost ProgCost = 4526.015 $million.

The SOM algorithms demonstrated above can easily handle these cases, by requiring that one or two options be clamped to zero during the optimization (e.g. AeqNumber[j] = 0 for all j, in the "Humans and teleoperators" case). The appropriate R&D costs are also set to zero. The optimal solutions for all cases are presented in Table 4.12.

The truly optimal solution to the case example turns out to be from the "Teleoperators and automated equipment" case:

Task 1:	AEQ	0.76 aeqs	118.928 $M
Task 2:	TEL	0.29 tels	154.789 $M
Task 3:	TEL	0.97 tels	517.410 $M
Task 4:	TEL	0.03 tels	14.013 $M

Optimal Jobtime = 298.39 days

Constant costs (Rw + Rt + Ra + Cw) = 2207.83 $M
Whole-job timecosts (Hw * Jobtime) = 169.928 $M

Minimum total cost ProgCost = 3182.900 $million.

TABLE 4.12: OPTIMAL SOLUTIONS FOR ALL HUMAN-MACHINE CASES

Humans, teleoperators, and automated equipment:
- Task 1: AEQ
- Task 2: HUM
- Task 3: HUM
- Task 4: TEL
- Jobtime = 465.20 days ProgCost = 4526 $M

Humans and teleoperators:
- Task 1: HUM
- Task 2: HUM
- Task 3: HUM
- Task 4: HUM
- Jobtime = 969.09 days ProgCost = 5527 $M

Humans and automated equipment:
- Task 1: AEQ
- Task 2: HUM
- Task 3: HUM
- Task 4: HUM
- Jobtime = 471.14 days ProgCost = 4359 $M

Teleoperators and automated equipment:
- Task 1: AEQ
- Task 2: TEL
- Task 3: TEL
- Task 4: TEL
- Jobtime = 298.39 days ProgCost = 3183 $M

Humans only:
- Tasks 1,2,3,4: HUM
- Jobtime = 969.09 days ProgCost = 5359 $M

Teleoperators only:
- Task 1,2,3,4: TEL
- Jobtime = 388.16 days ProgCost = 4574 $M

Automated equipment only:
- Task 1,2,3,4: AEQ
- Jobtime = 7290.46 days ProgCost = 38964 $M

It is interesting to compare this solution to the full three-option solution. The removal of the human option forces the reassignment of Tasks 2 and 3 to teleoperators in the optimal solution. Despite this, the reduced-option case is cheaper, because it does not include the substantial R&D costs for human support hardware (Rh = 1574 $M).

Also interesting is the effect of removing the AEQ option in the second case in Table 4.12. It is logical that this forces the reassignment of Task 1, since this task is performed by automated equipment in the three-option solution. However, the removal of the AEQ option also reassigns Task 4 from TEL to HUM. The removal of AEQ can affect a HUM-TEL tradeoff because it changes the optimal Jobtime, from 465 to 969 days. The longer completion time favors humans over teleoperators enough to switch the assignment of Task 4.

4.4.5 Summary of Four-Task Case Example:

This section has demonstrated the application of the Staged Optimization Method to this study's four-task three-option case example. Before the use of the SOM, when the tasks were modeled as separate jobs and individually optimized, the program cost had a minimum value of 4643 $million: 3782 $M in constant costs, and 861 $M in "operations" costs (i.e. task costs and whole-job timecosts).

By modeling and optimizing the tasks as an integrated job, the SOM reduces this minimum ProgCost to 4526 $million for the three-option case. This represents a savings of 117 $million from the operations costs of the program, or 13.6% of those costs. In this first application of the SOM, all of the cost reductions come from the operations costs; the constant costs are untouched.

However, when the SOM is applied several more times, to each of the reduced-option cases, the removal of some of the R&D costs results in a further decrease of 1343 $M. Within each reduced-option case, the SOM minimizes the operations costs.

In this example, the strong effect of the constant costs is due to the small scale of the construction job: the Radiometer mass is only 140 tons. Therefore the constant costs are a large part of the total costs, and the reduced-option cases can lead to large cost reductions.

In a larger, more work-intensive project (e.g. the assembly of a Solar Power Satellite), a larger fraction of the program costs would be operations costs. In that case, the first application of the SOM (with all the human-machine options included) would yield proportionally greater savings.

4.5 Application of SOM to Case Examples with Learning Curves:

This section presents examples of the application of the Staged Optimization Method to nonlinear cost functions, by adding learning curves to the overall cost equation. These learning curves are applied to the procurement of habitation modules (HAB), teleoperators (TEL), and automated equipment (AEQ). For clarity, only the initial hardware procurements are affected; the purchase of spares is done at fixed prices.

4.5.1 Modification of SOM for Learning Curves:

When the learning curve terms are added to the general cost equation (from Table 3.1), the equation becomes:

```
ProgCost =

     (Rw + Rh + Rt + Ra + Cw)
          .
          .
          .

     + Kh * Nhum[j]
     + (part of learning curve term dependent on NtotHUMi)
     + Hh * (Nhum[j] * Ttime[j])
     + Kt * Ntel[j]
     + (part of learning curve term dependent on NtotTELi)
     + Ht * (Ntel[j] * Ttime[j])
     + Ka * Naeq[j]
     + (part of learning curve term dependent on NtotAEQi)
     + Ha * (Naeq[j] * Ttime[j])
          .
          .
          .

     + Hw * Jobtime
```

where Rw, Rh, Rt, Ra, Cw, Kh, Hh, Kt, Ht, Ka, Ha, Hw = constants.

The learning curves alter the number-of-units terms in the cost equation, but they do not change the relationship between the number-of-units costs and the usage costs. Therefore the theorem of equal task times is also true for the equation above. Applying this theorem, and collecting the learning curve terms together, the equivalent cost equation is:

```
ProgCost =

     (Rw + Rh + Rt + Ra + Cw)
          .
          .
```

(continued)

$$
\begin{aligned}
&\qquad\vdots\\
&+ Kh * Nhum[j]\\
&+ Hh * (Nhum[j] * Jobtime)\\
&+ Kt * Ntel[j]\\
&+ Ht * (Ntel[j] * Jobtime)\\
&+ Ka * Naeq[j]\\
&+ Ha * (Naeq[j] * Jobtime)\\
&\qquad\vdots\\
&+ CTprHAB\\
&+ CTprTEL\\
&+ CTprAEQ\\
&+ Hw * Jobtime
\end{aligned}
$$

In this equation,

CTprHAB = total procurement cost for initial habitation modules

$$
\begin{aligned}
&= (\text{First-unit cost } CFprHAB)\\
&\quad * (1^{-q} + 2^{-q} + \ldots + \mathrm{trunc}(NtotHABi)^{-q}\\
&\qquad + (NtotHABi - \mathrm{trunc}(NtotHABi)) * NtotHABi^{-q})
\end{aligned}
$$

where NtotHABi = total number of initial habitation modules at worksite

$$= \frac{(\text{Total number of humans } NtotHUMi)}{(\text{Number of humans per HAB module})}$$

$$\text{where } NtotHUMi = \sum_{j=1}^{NTSK} Nhum[j]$$

The components of the first-unit cost CFprHAB and the number of humans per habitation module are cost model inputs.

The learning curve terms for CTprTEL (total procurement cost for initial teleoperators) and CTprAEQ (total procurement cost for initial automated equipment) are similar to those above, but in terms of the initial number of teleoperators NtotTELi and of the initial number of automated equipment units NtotAEQi.

For any given combination of assignments of options to tasks, there are explicit relationships between Jobtime and the numbers of units Nhum, Ntel, and Naeq. For example, if Task j is done by humans:

$$\text{Nhum}[j] = \frac{\text{NREPS}[j]}{\text{ODhum}[j] * \text{Jobtime}}$$

So the summation terms NtotHABi, NtotTELi, and NtotAEQi are also functions of Jobtime. Therefore, for any given combination of assignments, ProgCost can be expressed as a function of Jobtime, as required by step 2 of the SOM.

Because this function includes learning curve series, it cannot be minimized by setting its derivative equal to zero. Therefore step 3 of the SOM uses a simple gradient-search method to find an approximate minimum for ProgCost:

a) Arbitrary initial values are selected for Jobtime (e.g. 800 days) and for a time increment TimeIncr (e.g. 243 days);

b) Values for ProgCost are calculated at the initial value of Jobtime and close to it (e.g at 800 and 801 days); the difference between these values indicates the downward direction of the ProgCost curve;

c) Jobtime is incremented by TimeIncr in the downward direction, and a new value of ProgCost is calculated; if this new value is lower than the old value, this step is repeated;

d) If the new value of ProgCost is higher than the old value, TimeIncr is divided by 3 and its direction is reversed;

e) The process iterates back to step (c), until TimeIncr drops below an arbitrary value (e.g. 0.3 day); then the optimal values of Jobtime and ProgCost are output.

There are many gradient-search techniques available, some more efficient than the above. However, this technique is easy to implement, and given the shape of the ProgCost-Jobtime curves it is guaranteed to succeed. It is therefore appropriate for this illustrative example.

In step 4 of the SOM, the overall cost equation is decomposed into costs for individual tasks. This poses a problem, because the learning curve terms couple the tasks together. For example, the per-unit procurement cost for teleoperators is a function of the total number of teleoperators in all the tasks. To decouple the tasks, it was therefore decided to apply the learning curves separately within each task.

For example, if Task j is performed by teleoperators, the Task j costs are:

$$
\begin{aligned}
&Kt * Ntel[j] \\
&+ (\text{First-unit cost } CFprTEL) \\
&\quad * (1^{-q} + 2^{-q} + \ldots + trunc(Ntel[j])^{-q} \\
&\qquad + (Ntel[j] - trunc(Ntel[j])) * Ntel[j]^{-q}) \\
&+ Ht * (Ntel[j] * Jobtime)
\end{aligned}
$$

where $Ntel[j] = \dfrac{NREPS[j]}{ODtel[j] * Jobtime}$

Thus thee learning curve terms in the task costs may be slightly higher than actual costs, because they do not include other units of the same options purchased for other tasks. Fortunately, the SOM is an iterative process, and it can therefore converge on the optimal solution even though the intermediate cost models in step 4 of each iteration are not precisely accurate.

Step 4 computes task costs for each human-machine option, and step 5 compares these costs to select an improved combination of assignments. The SOM then iterates to step 2, until two successive selections of assignments are the same.

The starting combination of assignments is selected by separating the problem into individual tasks, and modeling each task as a separate job. Therfore the term (Hw * Jobtime) is added to each task's costs. As in step 4, the learning curves are applied to each task alone. Each task is then optimized with each human-machine option in turn, and the minimum costs are compared to select that task's starting assignment.

4.5.2 Four-Task Case Examples with Learning Curves:

This section presents a numerical example of the application of the SOM to this study's four-task case example, including learning curves in procurement. For comparison, the example is first shown with a 100% learning curve, i.e. no learning. Then an 80% curve is applied to purchases of habitation modules, teleoperators, and automated equipment.

To make the effects of the learning curves more obvious, the SOM is applied to the construction of 10 Microwave Radiometers (still a small job compared to the assembly of a Solar Power Satellite). To model this, the numbers of repetitions of each task and the numbers of units of low-tech and high-tech Radiometer hardware are multiplied by 10. The Radiometer's R&D cost Rw is kept at 437 $million.

In addition, the learning curves are slightly modified so that the per-unit cost of less-than-one unit of hardware is the same at the first-unit cost. For example, 0.4 teleoperators cost

(0.4) * (First-unit cost CFprTEL)

rather that the value from the learning curve series. The reason for this alteration is that the mathematical definition of the learning curve is not an accurate model of reality below 1 unit. For small fractions of one unit, the learning curve calculates a per-unit cost which can be much higher than the first-unit cost. To avoid this unrealistic behavior, the less-than-one-unit segment of the learning curves are flattened at the first-unit cost.

For brevity, the following examples only include the three-option optimization; the reduced-option cases are omitted. The three-option case is sufficient to illustrate the application of the SOM to cost functions with learning curves.

The parameters of the total program cost equation are shown in Table 4.13, including the first-unit costs for the learning curve terms. Many of the parameters are similar to those for the earlier SOM example (section 4.4.1, Table 4.7). However, the value of Cw is significantly higher, because it now includes the procurement of hardware for 10 Radiometers. The parameters Ch, Ct, and Ca have been replaced by the smaller Kh, Kt, and Ka, because per-unit costs have been taken out of them to set up the learning-curve terms. Finally, the number of repetitions of each task has been multiplied by 10.

When the learning curve parameter lcparam is set at 100% (no learning) and the SOM is applied, the starting combination of assignments is:

Task 1: AEQ
Task 2: HUM
Task 3: HUM
Task 4: TEL

After iterative optimization, the SOM's results are shown in Table 4.14. Because the learning curve parameter is set at 100%, the learning curves have no effect, and an application of the original SOM to this 10-Radiometer example would have generated the same results.

Next, the learning curve parameter is set at 80%, and the optimization process is repeated. As in the 100% case, the SOM selects the following starting combination:

Task 1: AEQ
Task 2: HUM
Task 3: HUM
Task 4: TEL

TABLE 4.13: PARAMETER VALUES IN THE GENERAL COST EQUATION (FOUR-TASK THREE-OPTION CASE, WITH LEARNING CURVES)

Whole-job R&D cost ($M)	Rw =	437.403
Human-related R&D cost ($M)	Rh =	1574.313
Teleoperator-related R&D cost ($M)	Rt =	168.430
Automated-equipment-related R&D cost ($M)	Ra =	345.330
Whole-job general costs ($M)	Cw =	8890.329
Number-of-humans cost factor ($M/pers)	Kh =	74.839
First-unit cost for HAB modules ($M/hab)	CFprHAB =	488.884
Human usage cost factor ($M/pers-day)	Hh =	0.439
Number-of-teleoperators cost factor ($M/tel)	Kt =	7.228
First-unit cost for teleoperators ($M/tel)	CFprTEL =	52.962
Teleoperator usage cost factor ($M/tel-day)	Ht =	1.592
Number-of-automated-equipment-units cost factor ($M/aeq)	Ka =	11.809
First-unit cost for auto. equip. ($M/aeq)	CFprAEQ =	110.922
Automated equipment usage cost factor ($M/aeq-day)	Ha =	0.116
Whole-job timecosts ($M/day)	Hw =	0.569

Task	Number of reps NREPS	Human productivity ODhum (reps/pers-day)	Teleoperator productivity ODtel (reps/tel-day)	Auto. equip. productivity ODaeq (reps/aeq-day)
1	166,880	9.10	16.00	74.00
2	37,280	26.60	43.20	0.02
3	6,000	2.31	2.08	0.02
4	6,000	39.20	76.80	0.02

TABLE 4.14: OPTIMAL SOLUTION FOR FOUR-TASK CASE EXAMPLE (THREE HUMAN-MACHINE OPTIONS: HUM, TEL, AEQ) (LEARNING CURVE PARAMETER LCPARAM = 100%)

Optimal selection of options:

Task 1: AEQ
Task 2: HUM
Task 3: HUM
Task 4: HUM

Calculated data for selection of final combination of assignments:

	If done by HUM		If done by TEL		If done by AEQ	
Task	HumNumber	HumCost	TelNumber	TelCost	AeqNumber	AeqCost
1	12.30	10969.080	7.00	17025.310	1.51	447.33
2	0.94	838.304	0.58	1408.649	1250.17	369749.30
3	1.74	1553.626	1.93	4708.674	201.21	59509.00
4	0.10	91.553	0.05	127.527	201.21	59509.00

Notes:
1) Costs are in $million.
2) Constant costs (e.g. R&D) are not included in the table;
(Rw + Rh + Rt + Ra + Cw) = 11415.800 $million.
3) Each task's costs include learning curve terms.
4) Each task's costs do not include (Hw * OptJobTime) costs;
(Hw * OptJobTime) = 849.105 $million.

Optimal Jobtime = 1491.00 days.
Minimum total program cost ProgCost = 15195.730 $million.

With the 80% learning curves in the cost equation, the iterative optimization produces the results in Table 4.15. The optimal selection of assignments is the same as for the 100% case, but the numerical results are different.

The most important difference is that the optimal Jobtime drops from 1491 days (for 100% learning curves) to 1455 days for the 80% case. Physically, the 80% learning curves are reducing the number-of-units costs, making larger workforces more affordable. Therefore the optimization increases the workforces and reduces Jobtime. For example, the number of units of automated equipment for Task 1 grows from 1.51 to 1.55.

The larger workforces lead to higher or lower task costs, depending on the relative effects of the increases in numbers of units (which push up costs) and of the learning curves (which reduce per-unit procurement costs). For example, the Task 1 costs decrease from 447 $M to 444 $M, but the Task 2 costs increase from 838 $M to 844 $M.

TABLE 4.15: OPTIMAL SOLUTION FOR FOUR-TASK CASE EXAMPLE
(THREE HUMAN-MACHINE OPTIONS: HUM, TEL, AEQ)
(LEARNING CURVE PARAMETER LCPARAM = 80%)

Optimal selection of options:

Task 1: AEQ
Task 2: HUM
Task 3: HUM
Task 4: HUM

Calculated data for selection of final combination of assignments:

	If done by HUM		If done by TEL		If done by AEQ	
Task	HumNumber	HumCost	TelNumber	TelCost	AeqNumber	AeqCost
1	12.61	10586.130	7.17	16916.880	1.55	443.94
2	0.96	843.887	0.59	1409.520	1281.39	252287.00
3	1.79	1563.973	1.98	4701.285	206.23	43222.69
4	0.11	92.163	0.05	127.605	206.23	43222.69

Notes:
1) Costs are in $million.
2) Constant costs (e.g. R&D) are not included in the table;
(Rw + Rh + Rt + Ra + Cw) = 11415.800 $million.
3) Each task's costs include learning curve terms.
4) Each task's costs do not include (Hw * OptJobTime) costs;
(Hw * OptJobTime) = 828.414 $million.

Optimal Jobtime = 1454.67 days.
Minimum total program cost ProgCost = 15188.180 $million.

It is expected that the 80% learning curves in procurement should reduce the overall program cost. The largest part of this reduction comes from the shortening of Jobtime: the whole-job timecosts decrease from 849 $M to 828 $M. As a result, the total program cost does decrease by 7.6 $million, to a new minimum of 15188 $M. (It should be noted that this learning curve does not affect the procurement of the space structure to be assembled; this cost has included an 80% learning curve all along, which caused no problems because it is part of the constant term Cw.)

The decrease in ProgCost is not very large, but it is sufficient to illustrate the effect of the learning curves and the application of the SOM to a nonlinear cost function. If the job's workforces were larger, the learning curves would have a larger effect. For example, if the whole-job timecost factor Hw is increased to 1.26 (by charging a 10%-per-year interest cost on the program's R&D costs), then the optimization favors shorter Jobtimes, i.e. larger workforces. Then the 100% learning curves lead to a minimum ProgCost of 16024 $M; the 80% learning curves drop this total to 15968 $M, a decrease of 56 $million.

4.5.3 Ten-Task Case Example with Learning Curves:

As a final example, the SOM is applied to the full Radiometer construction scenario, i.e. to the ten tasks described in Appendix C. As in the example above, 10 Radiometers are built. Based on the task descriptions in section C.3.1, a set of nominal productivities for humans, teleoperators, and automated equipment are estimated. The tasks, their numbers of repetitions (for 10 Radiometers), and the productivities are displayed in Table 4.16. Tasks 1, 2, 3, and 4 of the four-task case example are Tasks T1, T2, T5, and T8 in this table, respectively. The remaining parameters in the cost equation have the values in Table 4.13.

TABLE 4.16: PARAMETERS FOR TEN-TASK CASE EXAMPLE

Task	Number of reps	Human productivity ODhum (reps/pers-day)	Teleoperator productivity ODtel (reps/tel-day)	Auto. equip. productivity ODaeq (reps/aeq-day)
T1: Manufacture of truss beams	166,880	9.10	16.00	74.00
T2: Assembly of tetrahedral truss	37,280	26.60	43.20	0.02
T3: Manufacture of support columns	20	0.28	0.48	2.20
T4: Assembly of support columns	40	2.80	8.00	0.02
T5: Assembly of membrane and surface controls	6000	2.31	2.08	0.02
T6: Checkout of membrane surface controls	6000	5.60	17.60	0.02
T7: Assembly of feed horns	6000	9.10	16.00	0.02
T8: Installation of feed horn assemblies	6000	39.20	76.80	0.02
T9: Alignment of feed horn assemblies	6000	21.00	48.00	0.02
T10: Assembly of attitude control momentum wheels	6000	38.50	78.40	0.02

Step 1 of the SOM selects a starting combination of assignments by considering each task as a separate job, and optimizing each task for each human-machine option. The resulting data is presented in Table 4.17.

TABLE 4.17: CALCULATED DATA FOR SELECTION OF STARTING COMBINATION OF ASSIGNMENTS (TEN-TASK CASE EXAMPLE)

	If done by HUM		If done by TEL		If done by AEQ	
Task	HumNumber	HumCost	TelNumber	TelCost	AeqNumber	AeqCost
	(HumTime)		(TelTime)		(AeqTime)	
T1	7.43	12059.930	14.13	17558.280	4.00	978.54
	(2468.00)		(738.33)		(563.33)	
T2	1.83	1486.093	3.46	1689.710	180.75	229775.80
	(764.33)		(249.67)		(10312.66)	
T3	0.41	228.037	0.63	141.917	0.21	51.47
	(172.33)		(66.00)		(44.00)	
T4	0.19	94.236	0.21	34.146	3.80	911.98
	(77.00)		(23.33)		(527.00)	
T5	2.49	2325.537	6.92	5133.478	64.79	40893.23
	(1041.33)		(417.00)		(4630.67)	
T6	1.60	1231.821	2.01	749.582	64.79	40893.23
	(668.33)		(170.00)		(4630.67)	
T7	1.26	886.854	2.11	813.364	64.79	40893.23
	(524.33)		(178.00)		(4630.67)	
T8	0.61	355.087	0.86	227.870	64.79	40893.23
	(252.33)		(90.67)		(4630.67)	
T9	0.83	518.739	1.13	329.740	64.79	40893.23
	(345.00)		(111.00)		(4630.67)	
T10	0.61	358.914	0.85	224.270	64.79	40893.23
	(254.67)		(89.67)		(4630.67)	

Notes:
1) Costs are in $million.
2) Task times are in days.
3) Each task is modeled as an individual job, so each task's costs include (Hw * task time) costs.
4) Constant costs (e.g. R&D) are not included in the table.

One feature of the data in Table 4.17 is that the optimal times for teleoperators (TelTime) are lower than the optimal times for humans (HumTime). This is a logical result, since teleoperators have relatively low number-of-units costs but high usage costs. Another feature is that the AEQ figures for Tasks T5 through T10 are the same. This is because each of these tasks consists of 600 repetitions, and has an AEQ productivity of 0.001 reps/aeq-hr (the near-zero value used to avoid divide-by-zero problems). From the data in the table, the starting combination of assignments is:

Task T1: AEQ
Task T2: HUM
Task T3: AEQ
Task T4: TEL
Task T5: HUM
Task T6: TEL
Task T7: TEL
Task T8: TEL
Task T9: TEL
Task T10: TEL

After two iterations, the SOM reaches the optimal solution shown in Table 4.18. The iterative optimization has reasssigned Tasks T7 through T10 to humans. This is because the optimal Jobtime is 1599 days, closer to the optimal times for humans in Table 4.17.

The minimum total program cost is 16406 $million, including 11416 $M in constant costs and 4990 $M in "operations" costs. The ten-task case costs 1218 $M more than the four-task case, indicating that the additional six tasks only account for one-quarter of the operations costs. The job is dominated by Tasks T1, T2, T5, and T6, three of which were in the four-task example.

4.5.4 Summary of Cases with Learning Curves:

This section has demonstrated the application of the Staged Optimization Method to case examples including learning curves in the procurement of habitation modules, teleoperators, and automated equipment. Learning curves are representative of non-linearities in the overall cost equation, which can be handled by the SOM because they do not violate any of the method's assumptions.

To handle learning curves, the SOM uses a simple gradient-search algorithm to minimize the nondifferentiable equation for ProgCost. In the SOM's step 4, the decomposition of the cost equation into individual tasks requires learning curves for each task, which is not a strictly accurate model of the actual costs.

TABLE 4.18: OPTIMAL SOLUTION FOR TEN-TASK CASE EXAMPLE
(THREE HUMAN-MACHINE OPTIONS: HUM, TEL, AEQ)
(LEARNING CURVE PARAMETER LCPARAM = 80%)

Optimal selection of options:

Task T1:	AEQ
Task T2:	HUM
Task T3:	AEQ
Task T4:	TEL
Task T5:	HUM
Task T6:	TEL
Task T7:	HUM
Task T8:	HUM
Task T9:	HUM
Task T10:	HUM

Calculated data for selection of final combination of assignments:

	If done by HUM		If done by TEL		If done by AEQ	
Task	HumNumber	HumCost	TelNumber	TelCost	AeqNumber	AeqCost
T1	11.47	10387.480	6.52	16894.300	1.41	430.06
T2	0.88	823.250	0.54	1406.303	1165.97	249625.80
T3	0.04	41.957	0.03	67.901	0.01	1.75
T4	0.01	8.391	0.00	8.148	1.25	383.70
T5	1.62	1525.726	1.80	4693.459	187.66	42627.41
T6	0.67	629.362	0.21	555.553	187.66	42627.41
T7	0.41	387.300	0.23	611.108	187.66	42627.41
T8	0.10	89.909	0.05	127.314	187.66	42627.41
T9	0.18	167.830	0.08	203.703	187.66	42627.41
T10	0.10	91.544	0.05	124.716	187.66	42627.41

Notes:
1) Costs are in $million.
2) Constant costs (e.g. R&D) are not included in the table;
(Rw + Rh + Rt + Ra + Cw) = 11415.800 $million.
3) Each task's costs include learning curve terms.
4) Each task's costs do not include (Hw * OptJobTime) costs;
(Hw * OptJobTime) = 910.420 $million.

Optimal Jobtime = 1598.67 days.
Minimum total program cost ProgCost = 16405.830 $million.

However, since the learning curves do not cause large variations in cost behavior, and because the SOM is an iterative method, the method does reach the optimal solution.

Application of the SOM to a four-task case example with 100% and 80% learning curves shows that the 80% learning curve reduces the overall program cost. This reduction comes mainly from the shortening of the optimal Jobtime, which lowers the whole-job timecosts. Jobtime is shortened because the procurement learning curves make larger workforces more affordable.

A ten-task example with learning curves demonstrates that the SOM can be applied to larger, more realistic scenarios. Adding tasks lengthens the optimal Jobtime and increases the total program cost. The example also shows that the value of Jobtime favors human-machine options according to the relative sizes of their number-of-units and usage costs. For example, small values of Jobtime favor teleoperators over humans, because teleoperators have lower number-of-units costs.

CHAPTER 5:

STUDY CONCLUSIONS

This chapter presents a list of the study's major assumptions, and describes the study's major conclusions.

The list of assumptions (Table 5.1) is included because some of the study's conclusions are dependent on them. A more detailed discussion of these assumptions appears in section 3.7.1.

TABLE 5.1: MAJOR ASSUMPTIONS IN THIS STUDY

1) Future space scenarios are modeled as jobs consisting of individual tasks. All of a job's tasks start simultaneously, and run concurrently until individual tasks end.

2) Each task is performed by one or more human-machine options.

3) The available numbers of each option are unconstrained.

4) Fractional workforces can be assigned to tasks.

5) The intrinsic characteristics of each human-machine option are independent of the parameters of other options, and of the assignment of other options to tasks.

The study's seven major conclusions are listed in Table 5.2, and discussed below.

Conclusion 1:

In a space scenario performed by a selection of human-machine options, the total program costs include:

a) Constant costs
b) Number-of-units costs
c) Usage costs
d) Whole-job timecosts

The constant costs include R&D costs and procurement costs of hardware not directly connected to particular human-machine options (e.g. the structure to be assembled in a construction scenario). Number-of-units costs are dependent only on the numbers of units of human-machine options at the worksite (e.g. habitation module costs proportional to the number of humans in space).

TABLE 5.2: MAJOR CONCLUSIONS OF THIS STUDY

1) In a space scenario performed by a selection of human-machine options, the total program costs include:
   a) Constant costs
   b) Number-of-units costs
   c) Usage costs
   d) Whole-job timecosts

2) In the optimization process, the whole-job timecosts couple the individual task costs together, making single-task optimizations insufficient to solve the complete problem.

3) Within a space job, each task is optimized by one human-machine option. So each combination of assignments of options to tasks is a potential solution to the overall problem. Therefore the problem is fundamentally combinatorial.

4) Under this study's concurrent-tasks assumption, the minimum total program cost occurs when all the tasks have the same completion time.

5) This study's Staged Optimization Method (SOM) finds the problem's minimum-cost solution, in a computationally efficient manner.

6) Within the assumptions of the study, the SOM can handle a variety of space scenarios, including scenarios with nonlinear cost functions.

7) To apply the SOM to actual space scenarios, a comprehensive effort to develop cost model inputs is required, with particular emphasis on the estimation of productivities and R&D and procurement cost factors.

Usage costs are dependent on the numbers of unit-hours expended (e.g. teleoperator consumables costs proportional to the number of teleoperator-hours of work done). Whole-job timecosts are costs proportional to the whole-job completion time Jobtime, e.g. costs for general worksite support, and interest costs on R&D investment (if included in the model).

Most of these cost categories are quite separate in their behaviors. For example, the number-of-units costs are unaffected by the task completion times, while the whole-job timecosts are unaffected by the numbers of units.

The costs for any individual task include number-of-units costs and usage costs, both of which depend on the numbers of units of human-machine options assigned to that task.

Conclusion 2:

In the optimization process, the whole-job timecosts couple the individual task costs together, making single-task optimizations insufficient to solve the complete problem.

The reason for this coupling is that the timecosts are proportional to Jobtime, which is equal to the longest task's completion time. Since the selection of the longest task is a function of all the tasks, optimizing individual tasks may or may not have an effect on the whole-job timecosts. Therefore the optimization must deal with all of the tasks togeteher, including their effect of these timecosts.

Conclusion 3:

Within a space job, each task is optimized by one human-machine option. So each combination of assignments of options to tasks is a potential solution to the overall problem. Therefore the problem is fundamentally combinatorial.

For any individual task within the job, the costs display a robust nonlinear behavior. If two human-machine options are available, the task's cost surface shows a saddle-shape with two local minima in the feasible region. Variations in the equation parameters move and reshape this cost surface, but without changing its fundamental behavior. The productivities of the human-machine options have the strongest effect on the relative values of the minima.

Given the independence of options' productivities from each other (Assumption 5 in Table 5.1), the local minima in this saddle-shape always occur on the option axes, i.e. where only one option is applied to the task. Physically, this means that one human-machine option will be inherently cheaper than the others to perform the task.

This argument can be extended to multiple-option cases, so that each task in a multi-task job will be optimized by one option (e.g. humans only or teleoperators only). Therefore the potential solutions to the problem are combinations of single-option assignments to the tasks.

Conclusion 4:

Under this study's concurrent-tasks assumption, the minimum total program cost occurs when all the tasks have the same completion time.

For jobs consisting of concurrent tasks (Assumption 1 in Table 5.1) and with the cost structure described in Conclusion 1, there is a tradeoff between the size of workforces for individual tasks and the task's completion time. This tradeoff favors lengthening the task times up to the completion time of the longest task. Therefore, in the optimal solution, all of the tasks must have tho same completion time Jobtime.

This behavior is not usually seen in jobs on earth, whose costs consist mostly of usage costs. In space, where number-of-units costs are a relatively larger fraction of the total costs, the need to minimize workforces by lengthening task times is greater.

Conclusion 5:

This study's Staged Optimization Method (SOM) finds the problem's minimum-cost solution, in a computationally efficient manner.

The SOM is an iterative method which takes advantage of the combinatorial character of the problem (Conclusion 3) and of the theorem of equal task times (Conclusion 4) to minimize the total program cost through combinatorial selections and one-parameter optimizations. The SOM identifies the optimal combination of human-machine assignments for the tasks, and calculates the optimal number of units assigned to each task.

Because it works with one-parameter optimizations, and because it only calculates minima for a small fraction of the possible combinations, the SOM considerably reduces the problem's computational load. Although the complexity of the space scenario grows exponentially with additional tasks and human-machine options, the SOM's computation load grows only polynomially.

Conclusion 6:

Within the assumptions of the study, the SOM can handle a variety of space scenarios, including some with nonlinear cost functions.

This study's overall cost equation is sufficiently general that a wide range of space activities can be modeled, such as space construction, research operations, and satellite servicing. A variety of human-machine options can be included (e.g. several types of teleoperators).

Because the Staged Optimization Method converts the problem into a series of one-parameter optimizations, nonlinear terms can be included in the cost function. Then the optimizations may require special minimization techniques (e.g. gradient-search methods), but the one-parameter equations can still be optimized efficiently.

As long as the nonlinearities do not violate the study's assumptions, the SOM can iterate to an optimal solution. An example of a difficult nonlinearity is procurement learning curves, which couple the costs of individual tasks together (section 4.5.1). Although this forces the use of slightly inaccurate cost models in one step of the SOM, the method iterates to the correct optimal solution.

In applying the SOM to space activities, the most restrictive assumption is the concurrent-tasks requirement (Assumption 1 in Table 5.1). In general, this limits the SOM's application to scenarios (or parts of scenarios) consisting of simultaneous tasks. Future work may extend the SOM beyond this assumption, as discussed in section 3.9.

Conclusion 7:

To apply the SOM to actual space scenarios, a comprehensive effort to develop cost model inputs is required, with particular emphasis on the estimation of productivities and R&D and procurement cost factors.

Besides the decision variables to be calculated by the optimization, the total program cost equation also includes a number of parameters quantifying the costs. For example, number-of-units cost terms consist of number-of-units variables multiplied by number-of-units cost factors (in $M/unit). Such factors are algebraic combinations of many cost model inputs, representing the physical characteristics of the scenario. For example, the masses of teleoperators and of their support stations contribute to the number-of-teleoperators cost factor.

The accuracy of these parameters is important to the proper application of the SOM, since the overall cost equation must model the cost behavior of the actual scenario. (In some cases, an uncertain model input can be parametrically varied to assess its effect on the optimal solution.) Estimates for the model inputs are based on literature search, consultation, preliminary designs, sim ılations, and past data.

In this study's case example, several inputs with important effects on the problem solution were particularly difficult to estimate. The productivities of future human-machine options (e.g. free-flying teleoperators) are difficult to project from past data or preliminary designs. Fortunately, simulations of space tasks in laboratory and Shuttle experiments are beginning to build a data base on human and teleoperator productivities.

This study also developed and used a data base on the costs of 21 past and present spacecraft. However, the scatter in this data makes further research necessary, including some study of the effect of hardware complexity on future R&D and procurement costs.

REFERENCES

Chapter 1:

1-1) Bradley, S.P., A.C. Hax, and T.L. Magnanti, Applied Mathematical Programming, Addison-Wesley, Reading, MA, 1977.

1-2) Akin, D.L., "A Systems Analysis of Space Industrialization", Sc.D. Thesis, MIT Dept. of Aeronautics and Astronautics, Aug. 1981.

1-3) Miller, R.H., and D.L. Akin, "Logistics Costs of Solar Power Satellites", IAF paper no. 78-186, Sep. 1978. Also MIT SSL report 3-78.

1-4) Miller, R.H., "The Effect of Demand on Optimal Launch Vehicle Size", in the Journal of Spacecraft and Rockets, 16, 4, p.287, July/Aug. 1979.

1-5) Bock, et al, Lunar Resources Utilization for Space Construction, NASA contract NAS9-15560, General Dynamics Convair Div., Final Report, Apr. 1979.

1-6) Miller, R.H., and D.B.S. Smith, Extraterrestrial Processing and Manufacturing of Large Space Systems, Final Report, NASA contractor report CR-161293, Sep. 1979.

1-7) Bowden, M.L., and D.L. Akin, "Underwater Simulation of Human Dynamics and Productivity in Extra-Vehicular Assembly", IAF paper no. 79-109, Sep. 1979. Also MIT SSL report 12-79.

1-8) Miller, R.H., D.B.S. Smith, D.L. Akin, and M.L. Bowden, "Men or Machines to Build in Space?", in Astronautics and Aeronautics, Oct. 1980.

1-9) Akin, D.L., and M.L. Bowden, "EVA Capabilities for the Assembly of Large Space Structures", IAF paper no. 82-383, Sep. 1982.

1-10) French, R., and J.O. Burney, Teleoperator Maneuvering System: Mission Requirements and System Definition Study, Vought Corp., NASA contract NAS8-33903, Jan.1983.

1-11) Meintel, A.J. Jr., and R.T. Schappell, "Remote Orbital Servicing System Concept", in Satellite Services Workshop, vol. 2, NASA Johnson Space Center, June 22-24, 1982.

1-12) Sheridan, T.B., and W.L. Verplank, Human and Computer Control of Undersea Teleoperators, Office of Naval Research contract N00014-77-0256, MIT Man-Machine Systems Lab, 1978.

1-13) Brooks, T.L., "SUPERMAN: A System for Supervisory Manipulation and the Study of Human/Computer Interaction", M.S. Thesis, MIT Dept. of Mechanical Engineering, May 1979.

1-14) Spofford, J.R., and D.L. Akin, "Results of the MIT Beam Assembly Teleoperator and Integrated Control Station", AIAA paper 84-1890, in AIAA Guidance and Control Conference, Seattle, WA, Aug. 20-22, 1984.

1-15) Bodle, J.G., "Development of a Beam Builder for Automatic Fabrication of Large Composite Space Structures", in 13th Aerospace Mechanisms Symposium, NASA Johnson Space Center, NASA Conference Publication 2081, April 26-27, 1979.

1-16) von Tiesenhausen, G.F., An Approach Toward Function Allocation Between Humans and Machines in Space Station Activities, NASA Technical Memorandum TM-82510, Nov. 1982.

1-17) Price, H.E., "The Allocation of Functions in Systems", Human Factors, vol. 27, no.1, Feb. 1985.

1-18) Miller, R.H., M.L. Minsky, and D.B.S. Smith, Space Applications of Automation, Robotics, and Machine Intelligence Systems (ARAMIS), Phase I Final Report, NASA contractor reports CR-162079 through CR-162083, Aug. 1982.

1-19) Akin, D.L., et al, Space Applications of Automation, Robotics, and Machine Intelligence Systems (ARAMIS) - Phase II, Final Report, NASA contractor reports 3734 through 3736, Oct. 1983.

1-20) The Human Role in Space, Final Report, McDonnell Douglas Astronautics Co. report MDC H1295, NASA contract NAS8-35611, Oct. 1984.

1-21) Zimmerman, W.F., J. Bard, and A. Feinberg, Space Station Man-Machine Automation Trade-off Analysis, NASA Jet Propulsion Laboratory publication 85-13, Feb. 15, 1985.

Chapter 2:

2-1) Prasad, B., and C.L. Magee, "Application of Optimization Techniques to Vehicle Design - A Review", in Recent Experiences in Multidisciplinary Analysis and Optimization, NASA Langley Res. Ctr., NASA Conference Publication 2327 (Part I), April 24-26, 1984.

2-2) Sobieszczanski-Sobieski, J., "Structural Optimization by Multilevel Decomposition", AIAA paper 83-0832, in proc. of 24th Structures, Structural Dynamics, and Materials Conference, Lake Tahoe, 1983.

2-3) Hauser, F., "A Nonlinear Programming Method for System Design with Results that Have Been Implemented", in Recent Experiences in Multidisciplinary Analysis and Optimization, NASA Langley Res. Ctr., NASA Conference Publication 2327 (Part I), April 24-26, 1984.

2-4) Vanderplaats, G.N., "A Robust Feasible Directions Algorithm for Design Synthesis", AIAA paper 83-0938, in proceedings of 24th Structures, Structural Dynamics, and Materials Conference, Lake Tahoe, 1983.

2-5) Adelman, H.M., "Structural Sensitivity Analysis: Methods, Applications, and Needs", in Recent Experiences in Multidisciplinary Analysis and Optimization, NASA Langley Res. Ctr., NASA Conference Publication 2327 (Part I), April 24-26, 1984.

2-6) Lawler, E.L., J.K. Lenstra, and A.H.G. Rinnooy Kan, "Recent Developments in Deterministic Sequencing and Scheduling: A Survey", Preprint from The Mathematical Centre, Amsterdam, 1980. (To be included in Deterministic and Stochastic Scheduling, M.A.H. Dempster et al, eds., Reidel, Dordrecht 1982.)

2-7) Personal communication to Professor A.R. Odoni from Professor H.N. Psaraftis, MIT Dept. of Ocean Engineering, March 1985.

2-8) Personal communication from Dr. D.F.X. Mathaisel, MIT Flight Transportation Laboratory, March 1985.

2-9) Personal communication from R.S. Hiller, MIT Operations Research Center, March 11, 1985.

2-10) Bradley, S.P., A.C. Hax, and T.L. Magnanti, Applied Mathematical Programming, Addison-Wesley, Reading, MA, 1977. (Particularly Chapter 13: Nonlinear Programming.)

2-11) Charnes, A., and W.W. Cooper, "Programming with Linear Fractional Functionals", Naval Research Logistics Quarterly, vol. 9, pp. 181-186, 1962.

2-12) Bradley, S.P., and S.C. Frey Jr., "Fractional Programming with Homogeneous Functions", Operations Research, vol. 22, pp. 350-357, 1974.

2-13) Hildebrand, F.B., Methods of Applied Mathematics, Second edition, pp. 119-123, Prentice-Hall, Englewood Cliffs, NJ, 1965.

Chapter 3:

3-1) Personal communication from Professor A.R. Odoni, MIT Dept. of Aeronautics and Astronautics, March 1986.

3-2) Beyer, W.H. (ed.), CRC Standard Mathematical Tables, 26th Edition, p.64, CRC Press, Boca Raton, FL, 1981.

Chapter 4:

4-1) Lovelace, U.M., "Executive Summary", in The Microwave Radiometer Spacecraft: A Design Study, R.L. Wright (ed.), NASA Langley Research Center, NASA Reference Publication 1079, Dec 1981.

4-2) Jensen, J.K., and R.L. Wright, "Weight Estimates and Packaging Techniques for the Microwave Radiometer Spacecraft" (also "Appendix B: Spacecraft Drawings"), in The Microwave Radiometer Spacecraft: A Design Study, R.L. Wright (ed.), NASA Langley Research Center, NASA Reference Publication 1079, Dec 1981.

Appendix B:

B-1) Miller, R.H., and D.L. Akin, "Logistics Costs of Solar Power Satellites", International Astronautical Federation Congress paper 78-186, Sep. 1978.

B-2) Miller, R.H., and D.L. Akin, "Space-Manufactured Satellite Power Systems, AIAA J. of Energy, vol. 3, no. 6, Nov.-Dec. 1979.

Appendix C:

C-1) Lovelace, U.M., "Executive Summary", in The Microwave Radiometer Spacecraft: A Design Study, R.L. Wright (ed.), NASA Langley Research Center, NASA Reference Publication 1079, Dec 1981.

C-2) Jensen, J.K., and R.L. Wright, "Weight Estimates and Packaging Techniques for the Microwave Radiometer Spacecraft" (also "Appendix B: Spacecraft Drawings"), in The Microwave Radiometer Spacecraft: A Design Study, R.L. Wright (ed.), NASA Langley Research Center, NASA Reference Publication 1079, Dec 1981.

C-3) Akin, D.L., M.L. Minsky, E.D. Thiel, and C.R. Kurtzman, *Space Applications of Automation, Robotics, and Machine Intelligence Systems (ARAMIS) - Phase II*, v. 3: Executive Summary, sec. 3.2.2.5, NASA CR-3736, Oct. 1983.

C-4) Smith, D.B.S., and C.S. Major, *What is Productivity?*, MIT Space Systems Lab report 1-77, 1977.

C-5) Bowden, M.L., and D.L. Akin, "Underwater Simulation of Human Dynamics and Productivities in Extra-Vehicular Assembly", IAF paper 79-109, Sep. 1979.

C-6) Akin, D.L., and M.L. Bowden, "EVA Capabilities for the Assembly of Large Space Structures", IAF paper no. 82-383, Sep. 1982.

C-7) Personal communication from Alan Shalders, Universal Builders Supply, June 1984.

C-8) Sarver, G.L., et al, *NB-42 Neutral Buoyancy Simulation of EVA Assembly - Final Report*, MIT Space Systems Laboratory report 40-81, Dec. 1981.

C-9) Akin, D.L., M.L. Bowden, and J.R. Spofford, "Neutral Buoyancy Evaluation of Technologies for Space Station External Operations", International Astronautical Federation Congress paper IAF-84-38, Oct. 1984.

C-10) Spofford, J.R., and D.L. Akin, "Results of the MIT Beam Assembly Teleoperator and Integrated Control Station", AIAA paper 84-1890, in *AIAA Guidance and Control Conference*, Seattle, WA, Aug. 20-22, 1984.

C-11) *Space Fabrication Demonstration System*, Final Report, Grumman Aerospace Corp., NASA contractor reports CR-161286 and CR-161287, Mar. 15, 1979.

C-12) *Large Space Structures Fabrication Experiment*, Final Report, General Dynamics Convair, NASA contractor report CR-161098, Jan. 25, 1978.

C-13) Akin, D.L., et al, *Final Report on Experimental Assembly of Structures in EVA (EASE)*, MIT Space Systems Lab, in preparation.

C-14) Sheridan, T.B., and W.L. Verplank, *Human and Computer Control of Undersea Teleoperators*, MIT Man-Machine Systems Lab, Off. of Naval Research contract N00014-77-C-0256, 7-14 through 7-22, July 1978.

Appendix D:

D-1) *Space Station Reference Configuration Description*, Space Station Program Office, NASA Johnson Space Center report JSC-19989, August 1984.

D-2) *Shuttle Flight Operations Manual, Vol. 15: EVA Systems*, Basic, Rev A, NASA Johnson Space Center report JSC-12770, Jan. 6, 1984.

D-3) *Space Transportation System: EVA Description and Design Criteria*, Revision A, NASA Johnson Space Center report JSC-10615, May 1983.

D-4) Personal communication from C. Ed Whitsett Jr., NASA Johnson Space Center, Sep. 18, 1984.

D-5) *Manned Maneuvering Unit (MMU) Operational Data Book*, vol.1, Martin Marietta Aerospace report MMU-SE-17-73(Rev A), NASA contract NAS9-17018, Jan. 1984.

D-6) Meintel, A.J. Jr., and R.T. Schappell, "Remote Orbital Servicing System Concept", in *Satellite Services Workshop*, vol. 2, NASA Johnson Space Center, June 22-24, 1982.

D-7) Bodle, J.G., "Development of a Beam Builder for Automatic Fabrication of Large Composite Space Structures", in *13th Aerospace Mechanisms Symposium*, NASA Johnson Space Center, NASA Conference Publication 2081, April 26-27, 1979.

D-8) *Large Space Structures Fabrication Experiment*, Final Report, General Dynamics Convair, NASA contractor report CR-161098, Jan. 25, 1978.

D-9) *U.S. Standard Atmosphere, 1976*, NASA contractor report CR-153748, also report NOAA-S/T 76-1562, 1976.

D-10) Kendrick, J.B. (ed.), *TRW Space Data*, Third edition, TRW Systems Group, 1967. (Pages 32 and 67.)

D-11) *Mission Operation Report: High Energy Astronomy Observatory HEAO-A*, NASA Off. of Space Science report no. S-832-77-01, 1977.
Also *Mission Operation Report: High Energy Astronomy Observatory HEAO-B*, NASA Off. of Space Science report no. S-832-78-02, 1978.
Also *Mission Operation Report: High Energy Astronomy Observatory HEAO-C*, NASA Off. of Space Science report no. S-832-79-03, 1979.

D-12) National Aeronautics and Space Administration Budget Estimates, submitted yearly to the U.S. Congress, Fiscal Years 1959 through 1986. Each year's submission contains "actual" cost data (direct obligations) for one prior fiscal year.

D-13) NASA News Press Kit for HEAO-1, no. 77-45, Apr. 5, 1977.
Also NASA News Press Kit for HEAO-2, no. 78-165, Nov. 9, 1978.
Also NASA News Press Kit for HEAO-3, no. 79-113, Sep. 4, 1979.

D-14) Shuttle Systems Weight and Performance, Monthly Status Report, NASA Johnson Space Center report JSC-09095-77, Apr. 30, 1984. (Particularly Table 2.1.)

D-15) Shapland, D., and M. Rycroft, Spacelab: Research in Earth Orbit, Cambridge University Press, 1984.

D-16) Fong, F.K., et al, Space Division Unmanned Spacecraft Cost Model, Fifth Edition, Air Force Space Division, Los Angeles, CA, June 1981.

D-17) Mission Operation Report: Orbiting Astronomical Observatory OAO-A, NASA report no. S-831-66-01, 1966.
Also Mission Operation Report: Orbiting Astronomical Observatory OAO-A2, NASA Off. of Space Science and Applications report no. S-831-68-02, 1968.
Also Mission Operation Report: Orbiting Astronomical Observatory OAO-B, NASA Off. of Space Science and Applications report no. S-831-70-03, 1970.
Also Mission Operation Report: Orbiting Astronomical Observatory (OAO-C), NASA Off. of Space Science report no. S-831-72-04, 1972.

D-18) Personal communication from Professor Rene H. Miller, MIT Dept. of Aeronautics and Astronautics, Nov. 8, 1985.

D-19) Dryden, J.A., and J.P. Large, A Critique of Spacecraft Cost Models, Rand Corp. report R-2196-1-AF, Sep. 1977.

D-20) Personal communications from Robert S. Wolf and Ping Lee, MIT Space Systems Laboratory, Nov. 16, 1985.

Appendix E:

E-1) Personal communication from George Whittinghill, NASA Johnson Space Center, May 13, 1985.

E-2) Personal communication from Professor David L. Akin, MIT Space Systems Laboratory, Nov. 17, 1985.

E-3) Personal communication from Don Capone, NASA Kennedy Space Center, Sep. 14, 1984.

E-4) Moore, D.H., Pricing Options for the Space Shuttle, Special Study by the Congressional Budget Office of the U.S. Congress, March 1985. (Particularly Table 7.)

E-5) Stuart, D.G., and M. Martinez-Sanchez, Progess Report on Shuttle Tether Use for Payload Deployment to Circular Orbits, MIT Space Systems Laboratory, for Martin Marietta Aerospace contract no. RH4-394007, Aug. 10, 1984.

E-6) Personal communication from Chris Roberts, Center for Space Policy, June 5, 1985.

E-7) Space Transportation System Reimbursement Guide, NASA Johnson Space Center report JSC-11802, May 1980. (Particularly p. 3-4.)

E-8) Personal communication from Eugene Ferrick, NASA Headquarters TDRSS Office, Sep. 17, 1984.
Also Tracking and Data Relay Satellite System (TDRSS) User's Guide, Revision 4, NASA Goddard Space Flight Center report STDN no. 101.2, Jan. 1980.

Appendix K:

K-1) National Aeronautics and Space Administration Budget Estimates, submitted yearly to the U.S. Congress, Fiscal Years 1959 through 1986. Each year's submission contains "actual" cost data (direct obligations) for one prior fiscal year.

APPENDIX A:

LEXICON

A.1 Summary of Important Variables:

The following constants, parameters, and variables are used throughout the discussions in Chapters 2 and 3. Most of them appear in the study's general cost equation (Table 2.4 or Table 3.1). Mathematical expressions for Rw, Rh, Rt, Ra, Cw, Ch, Hh, Ct, Ht, Ca, Ha, and Hw appear in Appendix G, section G.4.

A.1.1 Constants in the General Cost Equation:

Rw = Whole-job R&D costs ($)
These are R&D costs not tied to specific human-machine options. Examples include the R&D of the space structure to be assembled, and of general worksite and ground support equipment.

Rh = R&D cost for human-related equipment ($)
This includes the R&D of habitation modules, pressure suits, manned maneuvering units, and human-specific ground support equipment.

Rt = R&D cost for teleoperators ($)
This is the R&D for the teleoperators, their in-orbit support stations, and their ground control stations.

Ra = R&D cost for automated equipment ($)
This is the R&D for automated equipment, its in-orbit support equipment, and its ground monitoring stations.

Cw = Whole-job general costs ($)
These are worksite costs not tied to specific human-machine options. Examples include the procurement and launch costs for the space structure to be assembled, the procurement of general ground support stations, and the training of the general ground support crew.

A.1.2 Parameters in the General Cost Equation:

Ch = Number-of-humans cost factor ($/person)
The number-of-humans cost terms (Ch * Nhum[j]) include the procurement and launch of habitation modules, pressure suits, and manned maneuvering units. Also included are the training costs for humans in space crews and for their ground support personnel.

Hh = Human usage cost factor ($/person-day)
The human usage cost terms (Hh * Nhum[j] * Ttime[j]) include the procurement and launch of spares and consumables for habitation modules, pressure suits, and manned maneuvering units. Also included are salaries for humans in space, launch costs for crew rotation, salaries for human-specific ground support personnel, and human-specific communications costs.

Ct = Number-of-teleoperators cost factor ($/teleoperator)
The number-of-teleoperators cost terms (Ct * Ntel[j]) include the procurement and launch of teleoperators, the procurement of teleoperator control stations, and the training costs for the teleoperators' ground support personnel.

Ht = Teleoperator usage cost factor ($/tel-day)
The teleoperator usage cost terms (Ct * Ntel[j] * Ttime[j]) include the procurement and launch of teleoperator spares and consumables, the procurement of spares for ground control stations, the salaries of ground control personnel, and the teleoperator-specific communications costs.

Ca = Number-of-automated-equipment-units cost factor ($/aeq)
The number-of-automated-equipment-units terms (Ca * Naeq[j]) include the procurement and launch of the automated equipment, the procurement of its ground monitoring stations, and the training costs of its ground support personnel.

Ha = Automated equipment usage cost factor ($/aeq-day)
Automated equipment usage terms (Ha * Naeq[j] * Ttime[j]) include procurement and launch of spares for the automated equipment, procurement of spares for its ground support equipment, salaries of its ground support personnel, and automated-equipment-specific communications costs.

Hw = Whole-job timecost factor ($/day)
The whole-job timecosts depend only on the whole-job completion time Jobtime. This (Hw * Jobtime) term includes the launch of construction site propellant (for orbit maintenance), the salaries of general ground support personnel, and the general worksite communications costs. Also included are discounting costs, i.e. interest costs on the early R&D and procurement costs, and any penalty costs for delays in the completion of the program. (Interest and penalty costs are not modeled in this study's case example, but in an actual scenario they would increase the value and effect of Hw.)

A.1.3 Variables in the General Cost Equation:

Nhum[j] = Number of humans assigned to task j (pers)

Ntel[j] = Number of teleoperators assigned to task j (tels)

Naeq[j] = Number of units of automated equipment assigned to task j (aeqs)

Ttime[j] = Completion time for task j (days)

Jobtime = Completion time for the whole job (days)
This is equal to the completion time for the longest task.

A.1.4 Other Important Quantities:

NTSK = Number of different tasks to be performed (#)

NREPS[j] = Number of repetitions of task j to be done (reps)

ODhum[j] = Daily productivity of humans in performing task j (reps/pers-day)

ODtel[j] = Daily productivity of teleoperators in performing task j (reps/tel-day)

ODaeq[j] = Daily productivity of automated equipment in performing task j (reps/aeq-day)

A.2 Abbreviations Within Variable Names:

This section presents a directory of the abbreviations used within variable names, i.e. for inputs, internal variables, and outputs of the cost model. Most variable names consist of several abbreviations joined together, alternating capital and lower case letters. For example:

NunSSTh =	N	un	SST	h
	= Number of	unit	space structure	high-tech hardware
	= Number of units of high-tech space structure			

A.2.1 One-Letter Abbreviations:

b = salary and overhead (bucks) *

c = consumables *

D = dollars

(continued)

h (or H) = high-technology hardware *
i = initial (at start of work in space) *
l (or L) = low-technology hardware *
N = number of...
s = spares *
t = training *

* when they appear as last letter in the variable name.

A.2.2 Two-Letter Abbreviations:

AD = aeq-day (automated equipment unit-day)
AH = aeq-hour (automated equipment unit-hour)
CF = first-item cost
CT = total cost of... (i.e. cost of that item for whole program)
DK = dollars/kilogram
DM = dollars/cubic meter
DP (or Dp) = dollars/person
DU = dollars/unit
eh (or EH) = emu-hour (extravehicular mobility unit-hour)
FR = fraction of...
GH = gsc-hour (ground support crewperson-hour)
gs (or GS) = ground support
HD = hours/day (at front of variable)
HD = hab-day (at end of variable)
hh (or HH) = hcc-hour (high-capacity channel-hour)
in = insurance
KD = kilograms/day
KG = kilograms
la (or LA) = launch
lc = learning curve
lh (or LH) = lcc-hour (low-capacity channel-hour)
mh (or MH) = mmu-hour (manned maneuvering unit-hour)
mp = mission preparation
Nd (or ND) = number of days...
OD = repetitions/day
of = ...of...
OH = repetitions/hour
pd (or PD) = person-day
ph = person-hour
pr (or PR) = procurement
QH = eqp-hour (misc. equipment-hour)
rd (or RD) = research and development
TD = teleoperator-day
th (or TH) = teleoperator-hour
un = unit
WF = wearout factor

A.2.3 Three-Letter Abbreviations:

add = additional

AEQ (or aeq) = automated equipment (in space)

AMS (or ams) = automated equipment monitoring station

crw = crew (humans, teleoperators, and automated equipment)

CSE = construction support equipment

CSP (or csp) = construction site propellant

Dhh = dollars/hcc-hour (dollars/high capacity channel-hour)

Dlh = dollars/lcc-hour (dollars/low capacity channel-hour)

Dpd = dollars/person-day

Dph = dollars/person-hour

EMU (or emu) = extravehicular mobility unit (pressure suit)

EQP (or eqp) = miscellaneous tools and equipment

GEN = general worksite

GHb = gsc-hours to be paid (in salary and overhead)

GMS (or gms) = general monitoring station

GSC (or gsc) = ground support crew (or crewperson)

GSE = ground support equipment

HAB (or hab) = habitation module

HCC = high-capacity channel (for communications)

hmo = human-machine option

HMS (or hms) = human monitoring station

HSE = human support equipment

HUM (or hum) = human (in space)

INS = insurance

LCC = low-capacity channel (for communications)

MMU (or mmu) = manned maneuvering unit

PDb = person-days to be paid (in salary and overhead)

pts = points

SST = space structure (to be assembled)

TCS (or tcs) = teleoperator control station

TDR = tracking and data relay

TEL (or tel) = teleoperator (in space)

tot = total (i.e. for the whole program)

TSK = tasks (to be done)

A.2.4 Four-Letter Abbreviations:

AMSH = ams-hour (automated equipment monitoring station-hour)

Dens = density

GMSH = gms-hour (general monitoring station-hour)

HMSH = hms-hour (human monitoring station-hour)

KGeh = kilograms/emu-hour

KGmh = kilograms/mmu-hour

KGof = kilograms of...

KGpd = kilograms/person-day

KGth = kilograms/teleoperator-hour

KGun = kilograms/unit

lcpm = learning curve parameter

Ntot = Total number of...

REPS = repetitions (of each task to be done)

TCSH = tcs-hour (teleoperator control station-hour)

YSNO = yes/no (indicates a 0-1 variable)

A.2.5 Miscellaneous Variable Names:

lcparam = learning curve parameter (for hardware other than SST)
stint = stint (tour of duty in space)

A.3 Abbreviations Within Units:

This section presents a directory of the abbreviations used within variable units throughout the cost model.

A.3.1 Abbreviations:

aeq = unit of automated equipment (in space)
ams = automated equipment monitoring station
emu = extravehicular mobility unit (pressure suit)
eqp = unit of miscellaneous tools and equipment
gms = general monitoring station
gsc = ground support crewperson
hab = habitation module
hcc = high-capacity channel (for communications)
hms = human monitoring station
hr = hour
kg = kilograms
kW = kilowatts
lbf = pound-force
lcc = low-capacity channel (for communications)
mmu = manned maneuvering unit
pers = person (in space)
reps = repetitions
tcs = teleoperator control station
tel = teleoperator (in space)

$ = dollars
$M = millions of dollars
% = percentage

A.3.2 Unabbreviated units:

cubic meter
day
launch
shift = work shift
stint = tour of duty in space
unit = unit of hardware

A.4 Miscellaneous Abbreviations:

* = multiplication symbol
/ = division symbol
LN = natural logarithm
S/C = spacecraft
STS = Space Transportation System (Shuttle)
LiOH = lithium hydroxide
sqrt = square root of...
$\binom{z}{k}$ = "z choose k" = number of k-combinations of z distinguishable items

APPENDIX B:

THE LINE-ITEM COST MODEL

B.1 Reasons for Choice of Line-Item Costing:

The purpose of this study is to explore the economic tradeoffs between humans and machines in space activities. Therefore the study's first major task is the development of a model of the cost contributions of the human and machine options to the overall program cost.

It is also anticipated that the overall program costs will be significantly impacted by a number of parameters, such as productivities of humans and machines, launch costs to low-earth orbit, and consumables requirements (Refs. B-1 and B-2). Therefore the cost model should also be able to trace the effect of these individual parameters.

In addition, the study's optimization methods will be developed from this cost model. Therefore, to improve the credibility of these optimization methods, the cost model should be checkable, i.e. it should display enough of the details of the cost calculations that the users of the method can verify the accuracy of the process. In other words, the cost model should present intermediate quantities of interest, such as total numbers of humans and machines in space, times to complete tasks, quantities of consumables required, and size of ground support crews.

Therefore the first step in this study is the formulation of a line-item cost model (sometimes called a Work Breakdown Structure). Such a model calculates the costs of small parts of the overall program, and adds these individual contributions to form the total program cost. Thus the individual contributions of humans and machines are separately calculated, and can easily be compared.

B.2 Scope and Generality of Model:

There is a tradeoff between the generality of a line-item cost model and its level of detail. It is possible to write a large cost model to handle all space activities, e.g. Shuttle, Space Station, space platform, satellite, and planetary probe missions. However, the inclusion of all the detailed subcases is not necessary for this study's theoretical development; a clear model of the effects of humans and machines is more important.

It was therefore decided to formulate the cost model with a limited scope, but with a general structure which could be adapted to other space activities. So the model described in this appendix concentrates on space construction at a permanent in-orbit facility. This takes advantage of the MIT Space Systems Lab's extensive experience in neutral buoyancy simulation of space construction by humans and teleoperators.

However, the resulting cost equations are applicable to other activities at space facilities, such as:

1) Operation of research and production facilities in zero-gravity

2) Servicing and checkout of orbit transfer vehicles

3) Satellite servicing and repair

because of two reasons. First, most of the cost items in the model are not affected by the character of the tasks being performed. For example, the three activities listed above have similar habitation requirements for humans. In general, construction activities require the most comprehensive cost modeling; the other activities would require a subset of the construction line-items. For example, a teleoperator for servicing satellites would probably require the same cost items as a construction teleoperator, except that it might not require mobility propellant.

Second, the optimization method developed later in this study does not depend on the type of activity being performed. Once the tasks and the human and machine options have been modeled in the format of the line-item cost model, the optimization technique will assign humans and machines to tasks to minimize overall cost, regardless of the physical description of the tasks to be done.

Thus this study's cost model format and its optimization method are designed to be general, so that the construction-scenario case example presented in this report could be modified to apply to any of the other three types of activities. Most of the changes required would be the definition of new tasks to be performed, and the generation of new values for some of the model inputs (e.g. new mass estimates for teleoperators and automated equipment).

In fact, this study's line-item cost model produces an overall cost equation (described in section 2.2) which is sufficiently general that it can be applied to other space scenarios, beyond those at permanent in-orbit facilities. For example, activities from the Space Shuttle can be modeled by adding a Shuttle user fee for keeping the vehicle in space for extended periods. Other space activities might also require additional cost terms, but the fundamental form of the equation would remain unchanged.

B.3 General Approach to the Line-Item Cost Model:

The overall function of the cost model is shown in schematic form in Figure B.1. The model itself consists of algebraic relationships which combine the various model inputs to generate intermediate results of interest and the line item costs. The cost model does not contain any numbers - these are provided to it by the inputs.

B.3.1 Line-Item Cost Model Inputs:

Three types of inputs are used. First, the task-related inputs define the job to be done in space in terms relevant to the cost model: number of different tasks to be done, and number of repetitions of each task. The values of these inputs are derived from descriptions of jobs to be done in space.

Second, the human and machine inputs describe the characteristics of the human and machine options to perform the tasks in space. These include numbers of humans and machines assigned to each task, masses of equipment and consumables, maintenance factors, and ground support requirements. Sources for these numbers include historical data (e.g. manuals on pressure suits, reports on consumables requirements), consultations with experts, and the preliminary designs of teleoperators, habitation modules, and other worksite equipment.

Some of the task-related and human and machine inputs overlap. These are the productivities of humans and machines in performing the various tasks. Values for these productivities are estimated from MIT Space Systems Lab experimental data, and from literature search and consultations. These productivity inputs are discussed in more detail in the next section.

Third, the situation inputs are parameters of the problem which are not tied to the choice of humans or machines for the tasks. Examples include R&D and procurement cost factors, launch costs, and general requirements for worksite support. Sources for these values include statistical analysis of historical data, literature search, and consultations with experts.

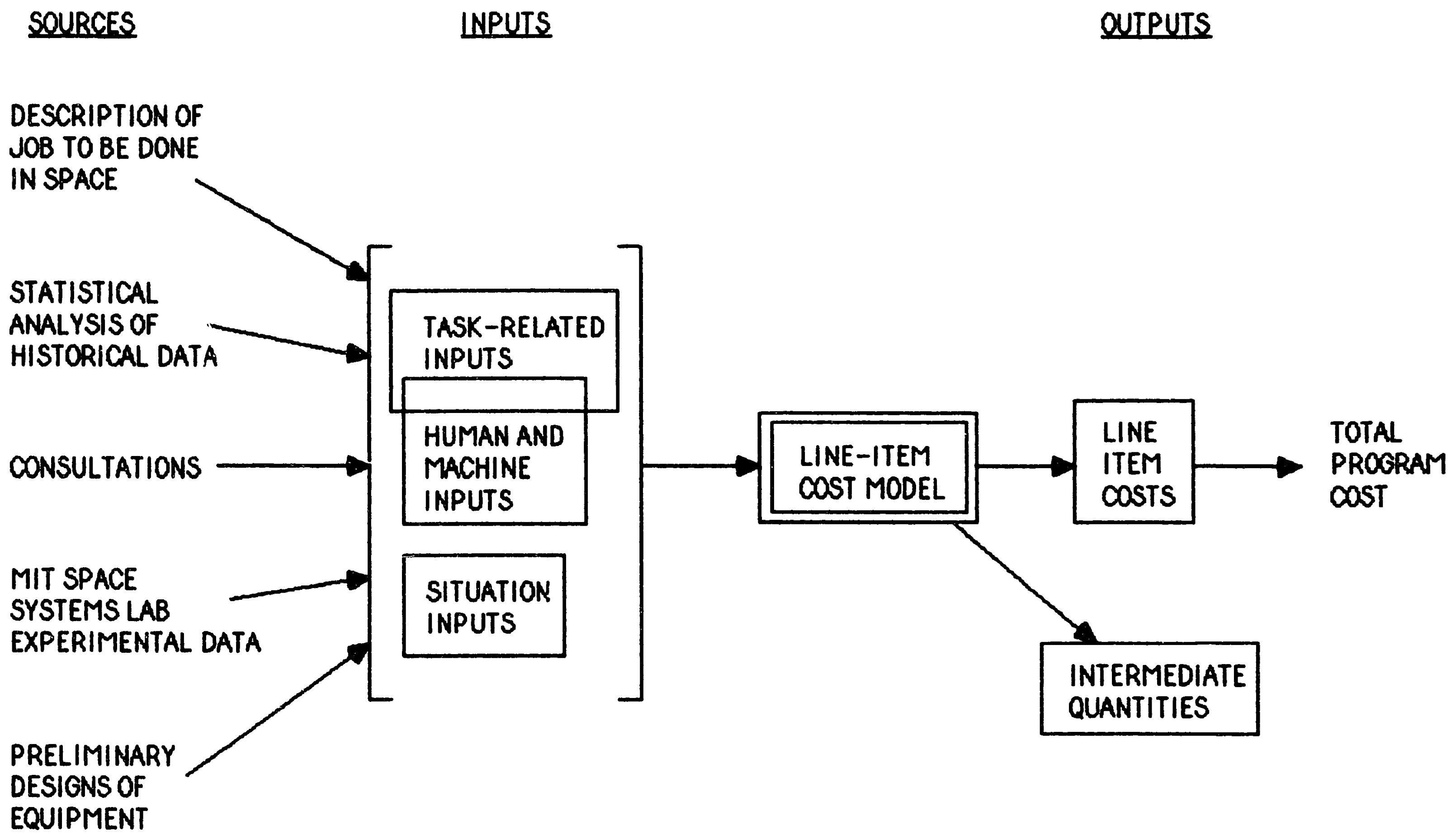


FIGURE B.1: SCHEMATIC OF THE LINE-ITEM COSTING PROCESS

The derivations of actual values of cost model inputs will be presented in Chapter 4 and Appendices C, D and E, for a case example of space construction.

B.3.2 The Importance of Productivities:

In costing activities at permanent space facilities, three overall factors have important effects on program costs:

a) the size of the job to be done

b) the numbers of humans or machines working on each task

c) the amount of time required for each task

The size of the job drives many of the research and development, procurement, and launch costs - for example, the costs for the basic hardware and raw materials for a space factory, or the costs of the space structure in a construction scenario. The numbers of humans and machines at the worksite determine their procurement and launch costs. And the times to complete tasks define certain program expenses such as consumables, maintenance, and salary costs.

Therefore all three factors must be quantified before a line-item cost model can accurately estimate the program cost of an activity in space. One possibility is to define all three factors explicitly as inputs to the model. For example, it could be stated that 500 repetitions of a task must be performed, using 4 machines, in 20 days.

Although complete, this definition of the problem is not very flexible. The completion of 500 repetitions by 4 machines in 20 days implies a machine design capable of this output, and a set of costs for that particular design. However, if one of the three inputs is changed independently of the others, e.g. requiring that 500 repetitions be done by 4 machines in 10 days, then this implies a different machine design, and requires a recalculation of the associated costs.

In this example, when the time allocation is cut in half, most people would suggest keeping the original machine design and doubling the number of machines. Similarly, most people would also suggest doubling the number of machines if the job size were doubled to 1000 repetitions. The advantage of this scaling process is that the same machine design would be applied to each case, the R&D costs would be unaffected, and the procurement and operations costs could be scaled in a staightforward manner.

Implicit in this logic is an assumption that there is a predictable triangular relationship between the three factors: increasing the size of the job requires an increase in the numbers of humans and machines, or an increase in the time to complete the tasks; for a fixed job size, however, the number of humans and machines and the time required are inversely related to each other.

For each task, this triangular relationship can be expressed as a productivity:

$$\frac{\text{(repetitions of the task)}}{\text{(number of humans or machines) x (time)}}$$

The use of productivities allows the calculation of any one of the three factors listed above from the other two. For example, knowledge of the productivity relationship, the job size (numbers of repetitions of each task), and the numbers of humans or machines assigned to each task, lead to an explicit calculation for the time required to complete the job.

In this study, it is assumed that the productivity relationship of a machine performing a task depends only on:

a) the machine design

b) the physical description of the task.

For example, if a machine has a productivity of 15 repetitions/machine-hour in doing Task 1, this figure is determined uniquely by the design of the machine and by the physical description of Task 1. This productivity figure could possibly be increased by spending more in R&D and procurement, but then the machine's design would be different.

The result of this assumption is that once a machine's design is fixed (and therefore its R&D, procurement, launch costs, and its consumables and maintenance requirements), then its productivity for each task is also fixed. Then, for each task, the number of repetitions of the task and the number of machines assigned to the task can be varied independently, and in each case the time to complete the task (and therefore the machine's operations costs) can be readily calculated.

This study's line-item cost model (described below) fixes the designs of the various human and machine options. Therefore the productivities of these options for each task are also fixed, and appear as constant inputs in the model. However, the analysis is not limited to such constants, and the productivities could be made functions of variable aspects of the designs, e.g. machine productivity as a function of R&D and procurement cost factors, and human productivity as a function of training and salary costs.

Therefore, to calculate all of the program cost elements, the line-item cost model requires input descriptions of:

1) the productivity relationships for all the tasks to be done;

2) two of the three factors:
   a) the number of repetitions of each task
   b) the number of humans or machines assigned to each task
   c) the time to complete each task.

The decision on which two of the three factors to provide as inputs depends on the expected uses of the cost model.

If the number of humans and machines for each task and the time for completion of each task are inputs, then the model will compute how many repetitions of each task will be performed. This might be useful to a project manager looking at alternatives to operate a continuous industrial process in space, with a fixed facility size and a need to maximize repetitions/year.

If the number of repetitions of each task and the time for completion of each task are inputs, the model will calculate numbers of humans and machines to do the job. This might be of interest to a project manager who needs to perform a fixed number of operations/year in space, and who has no restrictions on facility size.

However, it should be noted that for each task, if the number of repetitions and the completion time are set, then there are an infinite number of combinations of humans and machines which will satisfy the equation. Exact solutions would require either choosing a specific option for the task, or indicating specific ratios between the options.

If the number of repetitions of each task and the number of humans and machines assigned to each task are inputs, then the model will compute the time required for each task. This might be of interest to a project manager with a definite job to perform in space, with fixed resources of humans and hardware, but with a flexible deadline.

In making this choice for this study, it was first decided that an explicit description of the tasks to be performed would be more useful, i.e. the number of repetitions of each task should be inputs. This allows the modeling of specific examples of space projects (e.g. the construction of a particular structure, with x number of beam assemblies, y number of component installations). This eliminates the first alternative described above.

As described above, task times are not complete descriptions of space activity scenarios; however, the numbers of humans and machines assigned to each task are. Also, the expected results of this study's optimization method are the optimal numbers of humans and machines for each task. The line-item cost model will be used to verify these results by manipulating these numbers to verify their optimality. For these reasons, the numbers of humans and machines are the more appropriate inputs to the cost model.

In summary, it was decided to use:
1) the number of repetitions of each task
2) the numbers of humans and machines assigned to each task
as inputs to the line-item cost model.

B.4 Line-Item Cost Model for Construction Scenarios:

This section introduces the line-item cost model for in-orbit construction, which is described in detail in the remainder of Appendix B. The computer programs which implement this line-item cost model are printed out in Appendix H. The inputs to the cost model are discussed in Appendix E. A description of the abbreviations used in the names of the model variables is presented in the Lexicon in Appendix A.

B.4.1 General Cost Model Strategy:

The line item cost model breaks the program cost into four general categories: research and development (rd), procurement (pr), mission preparation and launch (la), and ground support (gs). These categories are further broken down into line-item cost elements. Table B.1 lists these cost elements, including each element's variable name.

There was some difficulty in settling on the four categories listed above. In past space programs, line-items tended to fall neatly into time-bounded categories: before-launch (consisting of R&D and procurement, usually combined), launch (including costs of launch vehicle and launch operations), and post-launch (usually mission operations and data analysis, which are entirely ground support costs).

TABLE B.1: COST ELEMENTS IN LINE-ITEM COST MODEL

Research & Development (CTrd):

CTrdSST	R&D of space structure to be assembled
CTrdHAB	R&D of human habitation modules
CTrdHSE	R&D of human support equipment (e.g. EMU, MMU, and misc. tools)
CTrdTEL	R&D of teleoperators and their support facilities (e.g. racks, power systems, storage facilities for spares and consumables)
CTrdAEQ	R&D of automated equipment and support facilities (e.g. power systems, storage facilities for spares)
CTrdCSE	R&D of construction support equipment (low-tech equipment, e.g. construction jigs, work platforms)
CTrdTSK	R&D of task procedures (e.g. simulations)
CTrdGSE	R&D of ground support equipment

Procurement (CTpr):

CTprSST	Proc. of space structure to be assembled
CTprHAB	Proc. of human habitation modules (including spares)
CTprHSE	Proc. of human support equipment (including spares)
CTprTEL	Proc. of teleoperators (including spares)
CTprAEQ	Proc. of automated equipment (including spares)
CTprCSE	Proc. of construction support equipment (including spares)
CTprHUMt	Cost of training humans in space crews
CTprHUMb	Salary and overhead for humans in space crews
CTprGSE	Proc. of ground support equipment (including spares)

Mission Preparation and Launch (CTla):

CTlaSST	Mission prep. and launch of space structure to be assembled
CTlaHUM	Launch of humans in space crews
CTlaHAB	Mission prep. and launch of human habitation modules (including spares and consumables)
CTlaHSE	Mission prep. and launch of human support equipment (including spares and consumables)
CTlaTEL	Mission prep. and launch of teleoperators (including spares and consumables)
CTlaAEQ	Mission prep. and launch of automated equipment (including spares)
CTlaCSE	Mission prep. and launch of construction support equipment
CTlaCSP	Launch of construction site propellant
CTlaINS	Launch insurance

Ground Support (CTgs):

CTgsGSCt	Ground support crew training
CTgsGSCb	Salary and overhead for ground support crew
CTgsTDR	User fees for tracking and data relay

With the advent of permanent space facilities, however, these distinctions are blurring, and new cost items are emerging. A major project at a space facility is likely to include several launches, for the initial hardware (perhaps in pieces) and later for spare parts, consumables, and perhaps new hardware to be added in orbit. The R&D costs can extend well into the operational phase, if new hardware is developed for later addition. Procurement extends past the first launch, for spare parts, consumables, and possibly additional hardware. Mission operations now includes in-space costs for human or machine intervention, perhaps over many years.

To avoid these ambiguities, it was decided to place all procurements into one category, including the procurement of ground support equipment. Costs of space crew training and salaries are also placed in "Procurement". All launch-preparation and launch costs for initial hardware, spares, humans, machines, and consumables, are placed in "Mission Preparation and Launch", which also includes insurance premiums. Ground support crew costs and communications costs are included in "Ground Support".

Since the cost model concerns itself only with line items, these general categories are a convenience, not a necessity. Other cost categories could be developed to clarify the cost elements.

B.4.2 Overall Scenario Description:

A space structure (SST) is assembled at a low-earth orbit space facility. The partially completed structure is kept at the facility during construction. The program includes the R&D, procurement, and launch of the space structure to be assembled. It also includes the R&D, procurement, and operation of those parts of the space facility involved in the construction.

All of the hardware, spare parts, consumables, and personnel required in space are launched on the Space Shuttle. The construction site is assumed to be in a 28.5-degree inclination orbit, at an altitude of 370 km (200 n. mi.).

B.4.3 Options for Construction: Humans, Teleoperators, and Automated Equipment:

Three fundamental options for space construction are examined in this study: humans, teleoperators, and automated equipment. These are considered separate options because their cost behaviors are different. It is expected that in a realistic construction scenario, some tasks would be assigned to humans, some to teleoperators, some to automated equipment, and some to combinations of the options.

Humans (HUM) perform construction in extravehicular activity (EVA). They therefore require extravehicular mobility units (pressure suits) and manned maneuvering units. They also carry miscellaneous tools and equipment. They need habitation modules, consumables for themselves and their equipment, launch services, training and salary, and some ground support. Humans are versatile, i.e. they have nonzero productivities for all tasks.

The humans at the construction site are expected to work 7-8 hours/day on construction tasks and to sleep 8-9 hours/day. This leaves roughly 8 hours/day for other activities. The humans are therefore expected to provide their own worksite support (e.g. housekeeping, food preparation, and maintenance of habitation modules, pressure suits, and manned maneuvering units), a situation analogous to Skylab operations. Construction management functions are shared between the worksite crew and the ground support crew, with the help of computers.

Therefore the only humans at the worksite are the EVA workers. They perform construction tasks 7-8 hours/day, 7 days/ week, on the average: if the crew can work 10-11 hours/day, 5 days a week, the same productivity is achieved. This is a hard work schedule, but it is not continuous: each human alternates "stints" in space with stints on the ground. A typical stint length would be three months.

Teleoperators (TEL) consist of free-flying teleoperator units and support equipment. This support equipment includes racks for the teleoperators, power generation and handling systems (e.g. solar arrays), and storage facilities for spares and consumables.

The model assumes that the teleoperators are controlled from the ground, so they require high-capacity communications and considerable ground support. Teleoperators can do some tasks well, but they have low productivities in tasks outside their design specifications.

Automated equipment (AEQ) are specialized devices (e.g. beam-builders) and their support equipment (e.g. power generation and handling systems, storage racks for spare parts). Automated equipment requires launch of basic hardware and spare parts. However, it does not need consumables.

Units of automated equipment are relatively autonomous, i.e. they need little ground support. Automated equipment is very productive in the tasks for which it was designed, but has near-zero productivities for other tasks.

Preliminary designs of teleoperators and automated equipment will be presented in Appendix D, for a case example of space construction.

B.4.4 Tasks and Productivities:

The model assumes that the whole construction job consists of a set of "tasks" (TSK). For example, in the construction of a microwave radiometer, the assembly of trusswork might be Task 1, and the insertion of feed horns might be Task 2.

Each task consists of one or more repetitions (REPS), e.g. the assembly of a single trusswork cell within Task 1. Thus the amount of Task 1 to be done in the whole construction job is defined by NREPS[1], the number of repetitions within Task 1 (i.e. the total number of trusswork cells to be assembled).

The model also assumes that the construction is performed by a "crew" of humans, teleoperators, and automated equipment. Each task can be performed by the humans, teleoperators, or automated equipment, but with different productivities in each case (possibly zero, if a task cannot be performed by one of the options).

It is possible to assign fractional humans or machines to tasks. These represent humans or machines which share their work time between two or more of the construction tasks. For example, 0.7 humans assigned to Task 1 and 1.3 humans assigned to Task 2 indicates that 2 humans are assigned to the tasks, with one splitting time between the two tasks.

It is assumed that at a permanent space facility, there will be opportunities to share one human's time (or one machine's time) with other programs. Therefore it is possible to have fractional totals of humans or machines at the worksite. For example, 0.7 humans assigned to Task 1 and 1.9 humans assigned to Task 2 can be interpreted as one human splitting time between the two tasks, one human doing only Task 2, and a third human spending 0.6 time on Task 2. The remaining 0.4 of the third human's time is devoted to some other activity, at another program's cost.

This time-sharing with another program may be a questionable assumption. In its defense, it should be noted that only one human or machine is shared, the 'last' one to be assigned. The advantage of this assumption is that the optimization process will not have to reach whole-number solutions, which greatly simplifies the theoretical development. If an integral solution is required, it is easier to calculate a non-integral solution, and then to examine the nearest whole-number possibilities to find the minimum-cost solution.

B.4.5 Anatomy of the Computer Model:

The flowchart in Figure B.2 shows the overall organization of the computer model. Starting at the top, the model inputs required to define the scenario are converted into the data file INPUTS1 by the program INPMKR1. The data file INPUTS1 is then available to all the sections of the model. Similarly, the program TASKMKR1 fills the data file TASKS1 with information of the tasks to be performed in space.

The main program is called COSTMDL1. Its primary function is to call the startup subroutine SETUP1 and the various modeling subroutines: PRDTIME1, ARENDEE1, PROCURE1, LAUNCH1, GRNDSUP1. The main program also prints out some of the results calculated by the model.

SETUP1 reads in the model inputs from INPUTS1 and TASKS1, makes them available to the rest of the model, and echoprints the inputs in the model outputs. It also sets up the global variables required by the other subroutines.

PRDTIME1 does productivity and time calculations, computing tasks completion times, job completion time, and the initial numbers of crew members at the worksite. These quantities are provided to the other subroutines as inputs to their models.

The main program COSTMDL1 then calls the subroutines ARENDEE1, PROCURE1, LAUNCH1, and GRNDSUP1 to model R&D, procurement, mission preparation and launch, and ground support costs, respectively. Besides the quantities from PRDTIME1, these subroutines also receive input data from INPUTS1 as needed. Some variables computed by these subroutines are used by other parts of the model, and the line-item cost elements are printed out as results.

B.5 Productivities, Task Times, and Initial Numbers:

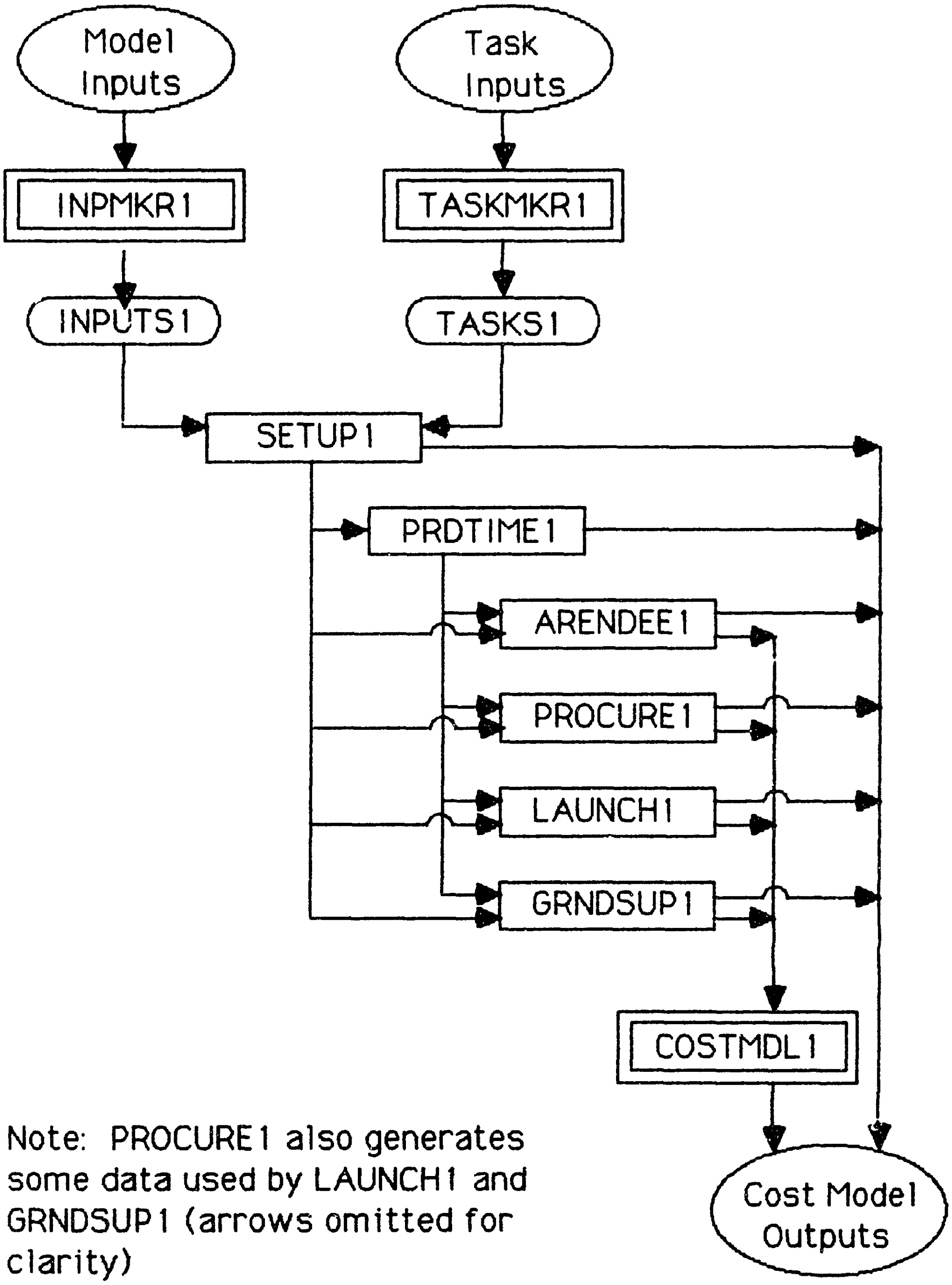
(Subroutine PRDTIME1)

Before the calculation of individual line-item cost elements can begin, a number of productivity and time calculations must take place. Results of these computations are used throughout the cost model. The model begins with productivity calculations to compute:

1) the daily productivities of humans (reps/pers-day), of teleoperators (reps/tel-day), and of automated equipment (reps/aeq-day) for each task;

2) the combined crew productivity (reps/day) for each task;

3) the time to complete each task (Ttime[j] days).



Note: PROCURE1 also generates some data used by LAUNCH1 and GRNDSUP1 (arrows omitted for clarity)

FIGURE B.2: GENERAL FLOWCHART OF COMPUTER COST MODEL

The model also computes the time to complete the entire job, and the initial numbers of humans, teleoperators, and units of automated equipment at the worksite. All of these calculations are handled by subroutine PRDTIME1.

B.5.1 Daily Productivities:

From INPUTS1:
HDhum = Hours/day worked by humans in space (hrs/day)
HDtel = Hours/day worked by teleops in space (hrs/day)
HDaeq = Hours/day worked by automated equipment in space (hrs/day)

From TASKS1:
OHhum[j] = Productivity of humans for task j (reps/pers-hr)
OHtel[j] = Productivity of teleops for task j (reps/tel-hr)
OHaeq[j] = Productivity of automated equipment for task j (reps/aeq-hr)
Nhum[j] = Number of humans assigned to task j (pers)
Ntel[j] = Number of teleops assigned to task j (tel)
Naeq[j] = Number of units of automated equipment assigned to task j (aeq)

The model calculates the daily productivities of humans, teleops, and automated equipment for each task. For task j, the daily productivity of humans is

```
     ODhum[j] = (OHhum[j] reps/pers-hr) * (HDhum hrs/day)
              = OHhum[j] * HDhum                 (reps/pers-day)
```

Similarly, the daily productivity of teleoperators is

```
     ODtel[j] = (OHtel[j] reps/tel-hr) * (HDtel hrs/day)
              = OHtel[j] * HDtel                  (reps/tel-day)
```

And the daily productivity of automated equipment is

```
     ODaeq[j] = (OHaeq[j] reps/aeq-hr) * (HDaeq hrs/day)
              = OHaeq[j] * HDaeq                  (reps/aeq-day)
```

The crew includes humans, teleops, and automated equipment. Therefore the daily crew productivity for task j is

```
     ODcrw[j] = ODhum[j] * Nhum[j]
              + ODtel[j] * Ntel[j]
              + ODaeq[j] * Naeq[j]                    (reps/day)
```

B.5.2 Task Times and Job Completion Time:

The model assumes that all tasks start on day 1, and that they are performed concurrently (until individual tasks end), as shown schematically in Figure B.3.

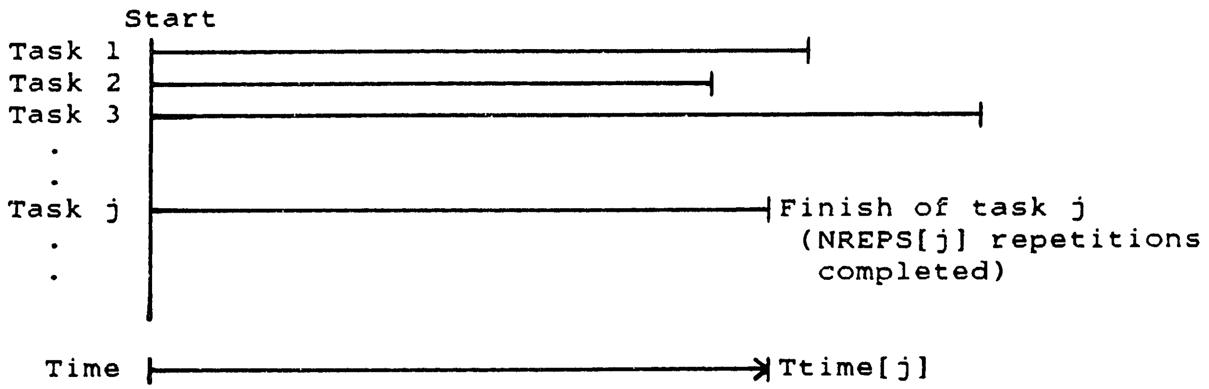


FIGURE B.3: TASK TIMES IN THE LINE-ITEM COST MODEL

In the line-item cost model, the completion time for each individual task is a function of the number of repetitions of the task and the crew productivity for that task. Therefore these completion times are independent of each other, and the tasks are finished at various times, as shown in Figure B.3.

Therefore some of the crew (humans and/or machines) will run out of work as tasks are finished. For example, the humans assigned to Task 2 in the schedule above will be idle after Ttime[2] days. The disposition of idle crew members is an issue because the cost model uses the individual task times Ttime[j] to compute time-dependent costs (e.g. salaries and consumables). So it does not compute any costs for idle crew members after their tasks are complete.

It is beyond the scope of this model to reassign idle humans and machines to other not-yet-completed tasks. Instead, the model assumes that idle crew members are immediately sent back to the ground and discharged (or reassigned to a different program at someone else's cost).

This assumption is also questionable. However, in the optimal solution, it will be proved (in Chapter 3) that in the minimum-cost situation, all the tasks must take the same time for completion. So in the optimal case the issue of idle crew members is irrelevant: everyone finishes their tasks at the same time, and the entire crew returns to the ground (or is reassigned to another program in space).

Therefore, since the purpose of the line-item cost model is to generate a cost equation for optimization, and to check the validity of the calculated optimal solutions, the assumption that crew members do not incur costs after their tasks are finished is acceptable.

From TASKS1:
NTSK = Number of different tasks to be performed (#)
NREPS[j] = Number of repetitions within task j (reps)

For each task j, the time to complete the task is:

$$\mathrm{Ttime}[j] = (\mathrm{NREPS}[j]\ \mathrm{reps}) / (\mathrm{ODcrw}[j]\ \mathrm{reps/day}) = \mathrm{Nreps}[j] / \mathrm{ODcrw}[j] \qquad \text{(days)}$$

The line-item cost model then calculates the time to complete the whole construction job, which is the time to complete the longest task (e.g. Ttime[3] in Figure B.3):

$$\mathrm{Jobtime} = \mathrm{MAX}(\mathrm{Ttime}[j]) \qquad \text{for } j=1 \text{ to NTSK} \qquad \text{(days)}$$

B.5.3 Initial Numbers of Humans, Teleoperators, and Automated Equipment:

Next, the cost model computes the initial numbers of humans, teleoperators, and automated equipment in space. It also checks the existence of humans, teleoperators, and automated equipment at the worksite, and records this in (0-1) variables; these will be used later in the model to calculate R&D costs.

From TASKS1:
Nhum[j] = Number of humans assigned to task j (pers)
Ntel[j] = Number of teleops assigned to task j (tel)
Naeq[j] = Number of units of automated equipment assigned to task j (aeq)

At the start of the work in space, when all tasks are being performed concurrently, the initial number of humans in space =

$$\mathrm{NtotHUMi} = \sum_{j=1}^{\mathrm{NTSK}} \mathrm{Nhum}[j] \qquad \text{(pers)}$$

Similarly, the initial number of teleoperators at the worksite =

$$\mathrm{NtotTELi} = \sum_{j=1}^{\mathrm{NTSK}} \mathrm{Ntel}[j] \qquad \text{(tels)}$$

And the initial number of units of automated equipment at the worksite =

$$\mathrm{NtotAEQi} = \sum_{j=1}^{\mathrm{NTSK}} \mathrm{Naeq}[j] \qquad \text{(aeqs)}$$

The model also checks the existence of HUM, TEL, and AEQ at the worksite, and records this in (0-1) variables:

```
If NtotHUMi = 0, then YSNOhum = 0
otherwise YSNOhum = 1

If NtotTELi = 0, then YSNOtel = 0
otherwise YSNOtel = 1

If NtotAEQi = 0, then YSNOaeq = 0
otherwise YSNOaeq = 1
```

(Ttime[j], Jobtime, NtotHUMi, NtotTELi, NtotAEQi, YSNOhum, YSNOtel, YSNOaeq are global variables, used by other parts of the model)

(End of PRDTIME1)

B.6 Calculation of Research and Development Costs:

(Subroutine ARENDEE1)

The following research and development cost elements are reprinted from Table B.1:

Research & Development:
- R&D of space structure to be assembled
- R&D of human habitation modules
- R&D of human support equipment (e.g. EMU, MMU, and misc. tools)
- R&D of teleoperators and their support facilities (e.g. racks, power systems, storage facilities for spares and consumables)
- R&D of automated equipment and support facilities (e.g. power systems, storage facilities for spares)
- R&D of construction support equipment (low-tech equipment, e.g. construction jigs, work platforms)
- R&D of task procedures (e.g. simulations)
- R&D of ground support equipment

In general, the cost model computes the R&D costs of various items by multiplying the mass of one unit of the item by a dollar/kilogram factor. In those cases, the model makes a distinction between low-technology and high-technology hardware, by using two different dollar/kilogram factors.

The cost model checked for the complete absence of humans, teleoperators, or automated equipment in subroutine PRDTIME1. If any of these options are absent, ARENDEE1 zeroes the appropriate R&D costs. For example, if the number of humans assigned is zero for all the tasks, then the human-specific R&D elements (i.e. HAB, EMU, MMU, EQP, and HMS) are set to zero.

B.6.1 R&D of Space Structure to be Assembled (SST):

(Subroutine RDofSST)

Structures large enough to require in-orbit construction have not yet been built, so there is little data from which to model their R&D cost. However, the physical design of the structure is expected to be essentially the same, whether it is assembled by humans or machines. Therefore the model assumes that its R&D cost will be the same in all cases.

So this cost will appear as a constant in the overall cost equation, and its value will not have any effect on the tradeoff between humans and machines. Therefore a simplified R&D cost model is acceptable.

The cost model calculates R&D costs for space structure by multiplying the mass of one unit of material by a dollar/kilogram factor, and adding a whole-structure integration factor based on the total mass of the structure. Space structure is made up of high-technology and low-technology hardware, and two different cost/kilogram factors are used to reflect this distinction.

From INPUTS1:
DKrdH = Dollars of R&D cost per kg of high-tech hardware ($/kg)
KGunSSTh = Mass of one unit of high-tech space structure (kg/unit)
NunSSTh = Number of units of high-tech space structure (units)
DKrdL = Dollars of R&D cost per kg of low-tech hardware ($/kg)
KGunSSTl = Mass of one unit of low-tech space structure (kg/unit)
NunSSTl = Number of units of low-tech space structure (units)

Total cost of R&D of high-tech space structure =

$$CTrdSSTh = DKrdH * KGunSSTh * 1 \text{ unit} + (DKrdH * KGunSSTh * NunSSTh * 0.035) \qquad (\$)$$

Total cost of R&D of low-tech space structure =

$$CTrdSSTl = DKrdL * KGunSSTl * 1 \text{ unit} + (DKrdL * KGunSSTl * NunSSTl * 0.035) \qquad (\$)$$

Total cost of R&D of space structure to be assembled =

$$CTrdSST = CTrdSSTh + CTrdSSTl \qquad (\$)$$

B.6.2 R&D of Human Habitation Modules (HAB):

(Subroutine RDofHAB)

The habitation modules house the human crew members while they are in space. The cost model assumes that HAB modules consist of low-tech hardware (the pressure shell) and high-tech hardware (the innards and support hardware, e.g. power systems).

The HAB R&D cost is therefore calculated by multiplying the high-tech hardware mass of one module by the dollar/kilogram factor for R&D of high-tech hardware, then multiplying the low-tech mass by the low-tech dollar/kilogram factor, and adding these products together.

If there are no humans at the worksite, then this R&D cost is not incurred. The model handles this by multiplying the calculated R&D cost by the (0-1) variable for the existence of humans.

From INPUTS1:
DKrdH = Dollars of R&D cost per kg of high-tech hardware ($/kg)
KGunHABh = Mass of high-tech hardware in one habitation module (kg/hab)
DKrdL = Dollars of R&D cost per kg of low-tech hardware ($/kg)
KGunHABl = Mass of low-tech hardware in one habitation module (kg/hab)

From PRDTIME1:
YSNOhum = Variable indicating existence of humans (0-1)

Total cost of R&D of habitation modules =

```
     CTrdHAB = ((DKrdH * KGunHABh * 1 hab)
                 + (DKrdL * KGunHABl * 1 hab)) * YSNOhum      ($)
```

B.6.3 R&D of Human Support Equipment (HSE):

(Subroutine RDofHSE)

Human support equipment consists of extravehicular mobility units (EMU), manned maneuvering units (MMU), and miscellaneous tools and equipment (EQP). The R&D costs for EMU and MMU are direct inputs to the model, since they represent improvements to existing designs rather than development of new hardware. The EQP R&D cost is calculated by multiplying the mass of one unit of EQP by the dollar/kilogram cost factor for R&D of high-tech hardware.

If there are no humans at the worksite, then this R&D cost is not incurred. The model handles this by multiplying the calculated R&D cost by the (0-1) variable for the existence of humans.

From INPUTS1:
DKrdH = Dollars of R&D cost per kg of high-tech hardware ($/kg)
KGunEQP = Mass of one unit of misc. equipment (kg/unit)
CTrdEMU = Total cost of R&D of pressure suits ($)
CTrdMMU = Total cost of R&D of manned maneuvering units ($)

From PRDTIME1:
YSNOhum = Variable indicating existence of humans (0-1)

Total cost of R&D of misc. equipment =
CTrdEQP = DKrdH * KGunEQP * 1 unit ($)

So total cost of R&D of human support equipment =
CTrdHSE = (CTrdEMU + CTrdMMU + CTrdEQP) * YSNOhum ($)

B.6.4 R&D of Teleoperators (TEL):

(Subroutine RDofTEL)

The cost model calculates R&D costs for teleoperators by multiplying the mass of one teleoperator (including flight support hardware, e.g. racks, power systems, storage facilities for spares and consumables) by the dollar/kilogram factor for R&D of high-tech hardware.

If there are no teleoperators at the worksite, then this R&D cost is not incurred. The model handles this by multiplying the calculated R&D cost by the (0-1) variable for the existence of teleoperators.

From INPUTS1:
DKrdH = Dollars of R&D cost per kg of high-tech hardware ($/kg)
KGunTEL = Mass of one teleoperator (including flight support hardware) (kg/tel)

From PRDTIME1:
YSNOtel = Variable indicating existence of teleoperators (0-1)

Total cost of R&D of teleoperator =
CTrdTEL = (DKrdH * KGunTEL * 1 tel) * YSNOtel ($)

B.6.5 R&D of Automated Equipment (AEQ):

(Subroutine RDofAEQ)

The cost model calculates R&D costs for automated equipment by multiplying the mass of one unit of automated equipment (including flight support hardware, e.g. power systems, storage facilities for spares)by the dollar/kilogram factor for R&D of high-tech hardware.

If there is no automated equipment at the worksite, then this R&D cost is not incurred. The model handles this by multiplying the calculated R&D cost by the (0-1) variable for the existence of automated equipment.

From INPUTS1:
DKrdH = Dollars of R&D cost per kg of high-tech hardware ($/kg)
KGunAEQ = Mass of one unit of automated equipment (including flight support hardware) (kg/aeq)

From PRDTIME1:
YSNOaeq = Variable indicating existence of auto. equipment (0-1)

Total cost of R&D of automated equipment =
CTrdAEQ = (DKrdH * KGunAEQ * 1 aeq) * YSNOaeq ($)

B.6.6 R&D of Construction Support Equipment (CSE):

(Subroutine RDofCSE)

The construction support equipment is low-tech equipment, e.g. construction jigs. Therefore its R&D cost is calculated by multiplying the mass of one unit of CSE by the dollar/kilogram factor for R&D of low-tech hardware.

From INPUTS1:
DKrdL = Dollars of R&D cost per kg of low-tech hardware ($/kg)
KGunCSE = Mass of one unit of construction support equipment (kg/unit)

Total cost of R&D of construction equipment =
CTrdCSE = DKrdL * KGunCSE * 1 unit ($)

B.6.7 Development of Task Procedures (TSK):

(Subroutine RDofTSK)

This line item is the cost of developing and perfecting techniques to perform the space tasks, e.g. through ground simulations. This number is read in from INPUTS1 as a global estimate for all tasks, based on expert consultation.

From INPUTS1:
CTrdTSK = Total cost of R&D of task procedures ($)

B.6.8 R&D of Ground Support Equipment (GSE):

(Subroutine RDofGSE)

The model assumes that ground support equipment (GSE) consists of general monitoring stations (GMS), human monitoring stations (HMS), teleoperator control stations (TCS), and automated equipment monitoring stations (AMS). These various stations are in the mission control center.

The R&D costs for each of these stations are read in from INPUTS1. If there are no humans, teleoperators, or automated equipment at the worksite, then some of these R&D costs are not incurred. The model handles this by multiplying the calculated R&D costs by the appropriate (0-1) existence variables.

From INPUTS1:
CTrdGMS = Total cost of R&D of general monitoring stations ($)
CTrdHMS = Total cost of R&D of human monitoring stations ($)
CTrdTCS = Total cost of R&D of teleoperator control stations ($)
CTrdAMS = Total cost of R&D of auto. equipment monitoring
stations ($)

From PRDTIME1:
YSNOhum = Variable indicating existence of humans (0-1)
YSNOtel = Variable indicating existence of teleoperators (0-1)
YSNOaeq = Variable indicating existence of auto. equipment (0-1)

Total R&D cost of ground support equipment =
CTrdGSE = CTrdGMS + (CTrdHMS * YSNOhum)
+ (CTrdTCS * YSNOtel) + (CTrdAMS * YSNOaeq) ($)

B.6.9 Calculation of Overall R&D Costs (CTrd):

(Subroutine ARENDEE1 cont.)

Overall R&D cost =
CTrd = CTrdSST
+ CTrdHAB + CTrdHSE
+ CTrdTEL + CTrdAEQ
+ CTrdCSE + CTrdTSK
+ CTrdGSE ($)

(CTrd is a global variable)

(End of ARENDEE1)

B.7 Calculation of Procurement Costs:

(Subroutine PROCURE1)

The following procurement cost elements are reprinted from Table B.1:

Procurement:
Proc. of space structure to be assembled
Proc. of human habitation modules (including spares)

Proc. of human support equipment (including spares)
Proc. of teleoperators (including spares)
Proc. of automated equipment (including spares)
Proc. of construction support equipment (including spares)
Cost of training humans in space crews
Salary and overhead for humans in space crews
Proc. of ground support equipment (including spares)

In general, the cost model calculates the procurement costs for hardware by multiplying the quantity of material (units or kilograms) by a procurement cost factor (dollars/unit or dollars/kilogram). For most items the quantity of material consists of initial hardware and spares (for some items, there are no spares).

These initial hardware and spares requirements are some of the intermediate results produced by the line-item cost model. Other results of interest are total person-days, teleoperator-days, and automated-equipment-days in orbit, and the unit-days for human support equipment and various types of ground support equipment. These results are printed out to provide checks on the realism of the modeling process.

This section of the cost model also calculates the required masses of consumables for humans and machines, and the required mass of construction site propellant. The procurement costs of these consumables are so low that they are not modeled, but the quantities are used later to compute launch costs. They are calculated here for ease of implementation, since they are computed from the intermediate results described above.

B.7.1 Initial Hardware and Spares:

The needed quantities of initial hardware are calculated from model inputs. For example, the numbers of units and masses of space structure to be assembled are read in from INPUTS1. The quantity of initial human support hardware (e.g. HAB modules, EMUs and MMUs) depends on the total initial number of humans at the worksite. This number was computed earlier by PRDTIME1, as well as the initial hardware requirements for teleoperators and automated equipment.

The quantities of spares required are calculated by multiplying an item's "usage" by its "wearout factor". For example, the teleoperator usage is the total number of teleoperator-days (tel-days) in orbit. This is calculated by multiplying the number of teleoperators working on each task by the completion time for that task, and summing these products over all tasks. Multiplying by the teleoperator wearout factor (with units tel/tel-day) yields the total spares requirement in fractional teleoperators (tels).

For both initial hardware and spares requirements, fractional items can be purchased. For initial hardware, this represents either "short" versions of hardware items (e.g. a short habitat module), or hardware which is shared with other programs (as discussed in section B.4.4). Therefore this program pays only a prorated share of the procurement. It is assumed that, at a permanent space facility, there will be opportunities to share hardware with other programs, or to rent out the hardware to other projects on a part-time basis.

The distinction between initial hardware and spares is important to this study's modeling process. As will be shown in Chapter 2 and Appendix G, the overall cost equation from the line-item cost model can be simplified by combining similar terms in several categories. Two of these categories are "number-of-units" terms (terms dependent on the number of humans or machines at the worksite) and "usage" terms (terms dependent on the number of person-days or machine-days at the worksite).

The quantities of initial hardware are functions of the number of humans and machines in space, and therefore the initial hardware costs contribute to the number-of-units terms. On the other hand, the spares requirements are due to wear-and-tear of the equipment, and are therefore functions of person-days or machine-days of use. So the spares costs contribute to usage terms in the cost equation.

The separation of cost terms into such categories will produce a reduced cost equation of valuable simplicity. It is therefore useful to model procurement costs so that they can be cleanly separated into number-of-units and usage terms.

For some types of hardware (e.g. manned maneuvering units), the procurement cost is simply the product of the quantity of hardware and a constant procurement factor. This cost is easily separated into two products, one for initial hardware and one for spares. For example:

Total procurement cost for MMUs =

(Total initial req. for MMUs) * (Dollars/MMU cost factor)

+ (Total spares req. for MMUs) * (Dollars/MMU cost factor)

The first term is in the number-of-humans category; the second term is in the human usage category.

B.7.2 Learning Curves in Procurement:

In several cases, however, the dollar/unit cost factors are not simple constants: they are calculated by applying *learning curves*. The following discussion first defines learning curves, and then describes their effect on the separation of procurement terms.

A learning curve is a mathematical formula which models the economies of scale in producing large numbers of identical units. These economies come from increased use of automation in large production processes, and improved workforce productivities in repeated manufacturing operations.

Mathematically, the learning curve calculation uses the cost of the first production unit, the total number of units to be produced, and a "learning curve parameter" to compute the total procurement cost. The learning curve parameter indicates the steepness of the curve, and thus the importance of the economy of scale; it is usually expressed as a percentage, e.g. 80%.

If an 80% learning curve is applied to a production run of N units, with a first-unit cost of A dollars, this means that the first unit costs A dollars; the second unit costs 80% of the first = (.8)(A) dollars; the *fourth* unit costs 80% of the *second* = (.8)(.8)(A) dollars; and so on, doubling the unit number every time. This leads to the formula:

$$\text{Total cost} = (\text{First-unit cost}) * (1^{-p} + 2^{-p} + \ldots + N^{-p})$$

where $-p = (\log_2 e) * \ln(l) = (1.4427) * \ln(l)$

and $l = (\text{learning curve parameter}) / 100$

Because of the series in the formula above, this equation can be applied only to integral numbers of units. Therefore the formula is slightly modified for use in the cost model:

$$\text{Total cost} = (\text{First-unit cost}) * (1^{-p} + 2^{-p} + \ldots + \text{trunc}(N)^{-p} + (N - \text{trunc}(N)) * N^{-p})$$

where

N = total number of units (can be a decimal number)

trunc(N) = truncation of N

This formula generates a smooth cost function, which matches the integral-unit cost values of the earlier formula.

For example, for an 80% learning curve, l = 0.8, and so

$$-p = (1.4427) * \ln(0.8) = -0.32193$$

If N = 6.3 units, the new formula gives:

$$\text{Total cost} = (\text{First-unit cost}) * (1^{-p} + 2^{-p} + \ldots + 6^{-p} + (6.3 - 6) * (6.3)^{-p})$$

$$= (\text{First-unit cost}) * (4.4653)$$

Figure B.4 presents examples of learning curves, with steepness ranging from 70% to 100%. In manufacture on earth, a typical learning curve is 80%. A 100% learning curve indicates no economy of scale, and therefore its per-unit cost is constant.

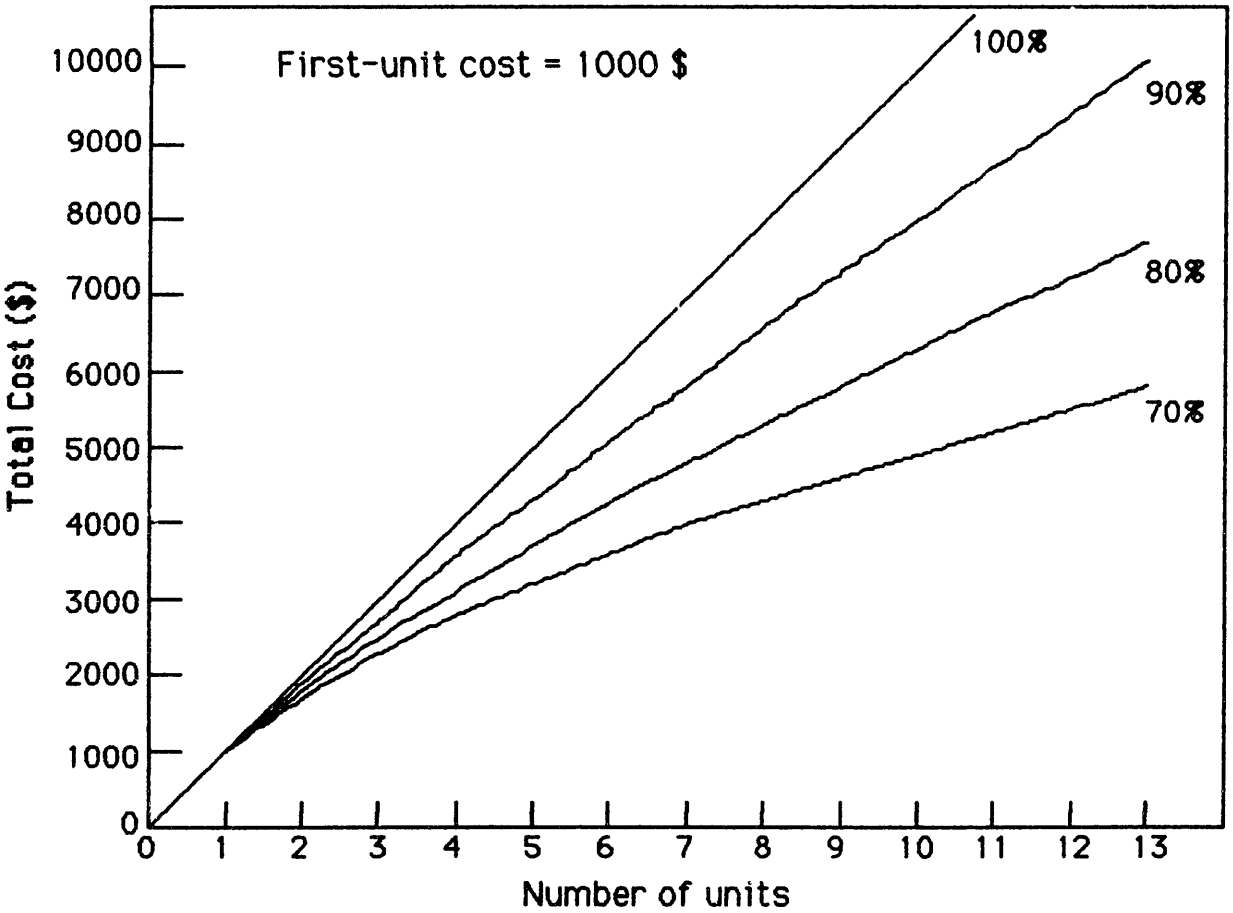


FIGURE B.4: EXAMPLES OF LEARNING CURVES

Learning curves are included in the procurement section of the line-item cost model because they make the model more realistic. However, they can also cause problems if they prevent the later separation of cost terms into number-of-units and usage categories. For example, if a learning curve is applied to the procurement of teleoperator hardware, it might generate a term of the form:

$$\begin{aligned}&(\text{First-unit cost of TEL}) * (1^{-p} + 2^{-p} + \ldots + \text{trunc}(\text{NtotTEL})^{-p}\\ &\qquad + (\text{NtotTEL} - \text{trunc}(\text{NtotTEL})) * \text{NtotTEL}^{-p})\end{aligned}$$

where
NtotTEL = Total requirement for teleoperators
= (Total initial requirement for TEL)
+ (Total spares requirement for TEL)

Because the term NtotTEL is taken to an exponent, this cost term cannot be separated into initial and spares terms. So this learning curve would couple the number-of-teleoperators and teleoperator-usage terms in the general cost equation. This would reduce the equation's value as a clear model of human and machine costs, and would greatly complicate the development of optimization algorithms.

This problem occurs because a learning curve is being applied to the sum of the initial hardware and spares requirements. To avoid this problem, the line-item cost model uses an alternative approach: it applies the learning curves separately to the initial hardware and to the spares. So for the procurement of teleoperators, the formula becomes:

$$\begin{aligned}&((\text{First-unit cost of TEL}) * (1^{-p} + 2^{-p} + \ldots + \text{trunc}(\text{NtotTELi})^{-p}\\ &\qquad + (\text{NtotTELi} - \text{trunc}(\text{NtotTELi})) * \text{NtotTELi}^{-p}))\\ &+ ((\text{First-unit cost of TEL}) * (1^{-p} + 2^{-p} + \ldots + \text{trunc}(\text{NtotTELi})^{-p}\\ &\qquad + (\text{NtotTELi} - \text{trunc}(\text{NtotTELi})) * \text{NtotTELi}^{-p}))\end{aligned}$$

where
NtotTELi = Total initial requirement for TEL
NtotTELs = Total spares requirement for TEL

For an 80% learning curve, an initial requirement of 5 teleoperators, and a spares requirement equivalent to 1.3 teleoperators, the cost terms would be:

$$\begin{aligned}&((\text{First-unit cost of TEL}) * (3.7377))\\ &\quad + ((\text{First-unit cost of TEL}) * (1.2757))\\ &= (\text{First-unit cost of TEL}) * (5.0134)\end{aligned}$$

This result can be compared to the result of a single 80% learning curve applied to 6.3 units (shown earlier). In this example, the separate learning curves give a total cost 11% higher than a single learning curve. If the learning curve parameter were 90%, the results would be closer; if the parameter were 100%, the results would be equal.

If the initial hardware and spares are bought together as a "block-buy" procurement, then it can be argued that the single learning curve is more realistic. However, such a block-buy would require an accurate prediction of the spares requirement at initial contract time.

Because the cost predictions from separate learning curves do not vary significantly from those from single curves, and because the general cost equation is considerably simplified by separate curves, the cost model uses separate learning curves for initial hardware and spares.

A more fundamental question is whether learning curves should be applied to small-quantity items. This is not an issue for the space structure to be assembled: it consists of large numbers of identical items (e.g. thousands of trusswork beams) and it is therefore expected that a learning curve would apply to its procurement.

However, for the small-number items (e.g. 3 habitats or 4.6 teleoperators) there is some question as to whether a block-buy could be negotiated with the manufacturer. Therefore the cost model uses two learning curve parameters, one for the space structure and one for the rest of the block-bought hardware. These can be varied independently.

B.7.3 Procurement of Space Structure to be Assembled (SST):

(Subroutine PRofSST)

The space structure is modeled as a number of units of high-tech hardware and a (different) number of units of low-tech hardware. Each of these types of hardware is treated separately, and two different cost/kilogram factors are used.

For each type of hardware, the cost model first computes the procurement cost of the first unit by multiplying the mass of one unit by a dollar/kilogram factor. Subroutine LERNCURV then uses that first-item cost, the total number of units, and the learning curve parameter for space structure to compute the total procurement cost.

The space structure requires no spares and no consumables.

High-tech space structure:

From INPUTS1:

DKprH = Dollars of proc. cost per kg of high-tech hardware ($/kg)

KGunSSTh = Mass of one unit of high-tech space structure (kg/unit)

NunSSTh = Number of units of high-tech space structure (units)

lcpmSST = Learning curve parameter for the space structure (%)

First-item cost of procurement of high-tech SST =
CFprSSTh = DKprH * KGunSSTh * 1 unit ($)

LERNCURV CFprSSTh, NunSSTh, lcpmSST into CTprSSTh
= total cost of procurement of high-tech space structure ($)

Low-tech space structure:

From INPUTS1:

DKprL = Dollars of proc. cost per kg of low-tech hardware ($/kg)

KGunSSTl = Mass of one unit of low-tech space structure (kg/unit)

NunSSTl = Number of units of low-tech space structure (units)

First-item cost of procurement of low-tech SST =
CFprSSTl = DKprL * KGunSSTl * 1 unit ($)

LERNCURV CFprSSTl, NunSSTl, lcpmSST into CTprSSTl
= total cost of procurement of low-tech space structure ($)

Complete structure:

Total procurement cost for space structure to be assembled =
CTprSST = CTprSSTh + CTprSSTl ($)

B.7.4 Procurement of Human Habitation Modules (HAB):

(Subroutine PRofHAB)

The habitation modules consist of high-tech and low-tech hardware. The initial hardware requirement is based on the number of humans at the worksite. The model assumes that each module provides personal and common areas, storage facilities, power generation and life-support equipment for NhumHAB humans.

The high-tech hardware to be procured consists of initial hardware and spares. The spares requirement is computed from the HAB usage. Separate learning curves are applied, and the procurement cost of high-tech HAB hardware is the sum of the two results.

For the low-tech HAB hardware, the cost model assumes that no spares are required. A learning curve applies to the initial hardware, using the learning curve parameter for hardware other than the space structure.

From PRDTIME1:
NtotHUMi = Initial number of humans at the worksite (pers)
Ttime[j] = Time to complete task j (days)

From INPUTS1:
NhumHAB = Number of humans in one habitation module (pers/hab)
WFhab = Wearout factor for habitation modules (habs/hab-day)
lcparam = Learning curve parameter for hardware other than SST (%)

From TASKS1:
NTSK = Number of different tasks to be performed (#)
Nhum[j] = Number of humans assigned to task j (pers)

Total initial hardware requirement for habitation modules =
NtotHABi = (NtotHUMi pers) / (NhumHAB pers/hab)
= NtotHUMi / NhumHAB (habs)

The first-item cost for the high-tech hardware in the habitation modules is computed by multiplying the high-tech mass in one module by the dollar/ kilogram factor for procurement of high-tech hardware. LERNCURV is then applied to this first-item cost and to the initial high-tech HAB requirement to generate the procurement cost for initial high-tech HAB hardware.

From INPUTS1:
KGunHABh = Mass of high-tech hardware in one habitation module (kg/hab)
DKprH = Dollars of proc. cost per kg of high-tech hardware ($/kg)

Thus the first-item high-tech HAB cost =
CFprHABh = KGunHABh * DKprH * 1 hab ($)

LERNCURV CFprHABh, NtotHABh, lcparam into CTprHABh
= cost of procurement of initial high-tech HAB hardware ($)

The high-tech spares requirement for the modules is based on the normal wear-and-tear during the program tasks. It is therefore a function of habs-days in space. These habs-days are multiplied by a wearout factor to calculate the equivalent number of habs needed as spares.

The cost model computes the total number of habs-days charged to this program by multiplying the habitation requirement <u>for each task</u> by the total duration of that task, and adding these individual products together. In other words, while the humans assigned to Task j work on that task, their HAB spares are charged to this program. Once Task j is finished, the model assumes that these humans are either sent groundside or reassigned to another program, and their HAB spares are then no longer charged to this program.

The model first computes the total number of person-days in space. For task j, the number of person-days in space is

$$(\text{Nhum}[j]\ \text{pers}) * (\text{Ttime}[j]\ \text{days}) \qquad (\text{pers-days})$$

So the total number of person-days in space for the program =

$$\text{NtotPD} = \sum_{j=1}^{\text{NTSK}} \text{Nhum}[j] * \text{Ttime}[j] \qquad (\text{pers-days})$$

For Task j, the total HAB usage is

$$[(\text{Nhum}[j]\ \text{pers}) / (\text{NhumHAB}\ \text{pers/hab})] * (\text{Ttime}[j]\ \text{days})$$

$$= (\text{Nhum}[j] / \text{NhumHAB}) * \text{Ttime}[j] \qquad (\text{hab-days})$$

So the total HAB usage for the program =

$$\text{NtotHD} = \sum_{j=1}^{\text{NTSK}} (\text{Nhum}[j] / \text{NhumHAB}) * \text{Ttime}[j]$$

$$= \text{NtotPD} / \text{NhumHAB} \qquad (\text{hab-days})$$

And the total high-tech HAB spares requirement for the program =

$$\text{NtotHABs} = (\text{NtotHD}\ \text{hab-days}) * (\text{WFhab}\ \text{habs/hab-day})$$

$$= \text{NtotHD} * \text{WFhab} \qquad (\text{habs})$$

LERNCURV CFprHABh, NtotHABs, lcparam into CTprHABs
= cost of procurement of high-tech HAB spares ($)

From INPUTS1:
KGunHABl = Mass of low-tech hardware in one habitation module (kg/hab)
DKprL = Dollars of proc. cost per kg of low-tech hardware ($/kg)

Since the low-tech hardware in the HAB modules requires no spares, the total low-tech HAB requirement =
NtotHABi (habs)

And the first-item low-tech HAB cost =

$$\text{CFprHABl} = \text{KGunHABl} * \text{DKprL} * 1\ \text{hab} \qquad (\$)$$

LERNCURV CFprHABl, NtotHABi, lcparam into CTprHABl
= total cost of procurement of low-tech HAB hardware ($)

Then we have total procurement cost for HAB hardware =

$$\text{CTprHAB} = \text{CTprHABh} + \text{CTprHABs} + \text{CTprHABl}$$

The habitation consumables are the crew consumables, e.g. oxygen, food, and water. The mass of these consumables is calculated by multiplying the total person-days at the worksite by a kg/person-day factor. This estimate is slightly conservative, because the humans spend part of those person-days in EVA (using EMU consumables instead).

From INPUTS1:
KGpdHABc = Kgs. of consumables required by one human in one day (kg/pers-day)

As discussed above, the total number of person-days in space for the program is NtotPD. So total consumables requirement for habitation modules =

KGofHABc = NtotPD * KGpdHABc (kg)

The low procurement cost of these consumables is not modeled.

B.7.5 Procurement of Human Support Equipment (HSE):

(Subroutine PRofHSE)

Human support equipment consists of extravehicular mobility units (EMU), manned maneuvering units (MMU), and miscellaneous tools and equipment (EQP). Each requires procurement of initial hardware and spares. The initial hardware requirement is based on the number of humans at the worksite. The spares requirement is based on normal wear-and-tear during program tasks.

Extravehicular mobility units (EMU):

The cost model assumes that a pressure suit is purchased for each of the humans in the space crews. The model also assumes that the humans in the space crews will be on a rotation schedule: three months in space, three months on the ground. Therefore two complements of humans will alternate at the worksite, and EMUs must be bought for both complements.

From PRDTIME1:
NtotHUMi = Initial number of humans at the worksite (pers)

From PRofHAB:
NtotPD = Total number of person-days in space for the program (pers-days)

From TASKS1:
NTSK = Number of different tasks to be performed (#)
Nhum[j] = Number of humans assigned to task j (pers)

From INPUTS1:
HDhum = Hours/day worked by humans in space (hrs/day)
WFemu = Wearout factor for pressure suits (emus/emu-hr)
KGunEMU = Mass of one extravehicular mobility unit, including support hardware (kg/emu)
DKprH = Dollars of proc. cost per kg of high-tech hardware ($/kg)

Total initial hardware requirement for pressure suits =

$$\text{NtotEMUi} = 2 * (\text{NtotHUMi pers}) * (1 \text{ emu/pers}) = 2 * \text{NtotHUMi} \quad \text{(emus)}$$

The spares requirement is a function of the number of emu-hours of space activity. These emu-hrs are multiplied by a wearout factor to compute the equivalent number of emus needed as spares. The cost model computes the total number of emu-hrs charged to this program by multiplying the EMU requirement <u>for each task</u> by the total duration of that task, and adding these individual products together.

For Task j, the total EMU usage is

$$(\text{Nhum}[j] \text{ pers}) * (1 \text{ emu/pers}) * (\text{Ttime}[j] \text{ days}) * (\text{HDhum hrs/day}) = \text{Nhum}[j] * \text{Ttime}[j] * \text{HDhum} \quad \text{(emu-hrs)}$$

So the total EMU usage for the program =

$$\text{NtotEH} = \sum_{j=1}^{\text{NTSK}} (\text{Nhum}[j] * \text{Ttime}[j] * \text{HDhum})$$

$$= \text{NtotPD} * \text{HDhum} \quad \text{(emu-hrs)}$$

And the total EMU spares requirement for the program =

$$\text{NtotEMUs} = (\text{NtotEH emu-hrs}) * (\text{WFemu emus/emu-hr}) = \text{NtotEH} * \text{WFemu} \quad \text{(emus)}$$

So the total hardware procurement requirement for pressure suits =

$$\text{NtotEMU} = \text{NtotEMUi} + \text{NtotEMUs} \quad \text{(emus)}$$

For simplicity, the cost model does not apply a learning curve to EMU procurement. Pressure suits are high-tech hardware, so the cost is calculated by multiplying the total required mass of EMU by the dollar/kilogram factor for procurement of high-tech hardware.

So the total cost of procurement of EMU hardware =

$$\text{CTprEMU} = (\text{NtotEMU emus}) * (\text{KGunEMU kg/emu}) * (\text{DKprH \$/kg}) = \text{NtotEMU} * \text{KGunEMU} * \text{DKprH} \quad (\$)$$

The EMU consumables consist of oxygen, water, air-scrubbing canisters, and batteries. The mass of these consumables is calculated by multiplying the cotal emu-hours at the worksite by a kg/emu-hr factor.

From INPUTS1:
KGehEMUc = Kgs. of consumables required by one pressure suit in one hour (kg/emu-hr)

So total consumables requirement for pressure suits =

$$\text{KGofEMUc} = \text{NtotEH} * \text{KGehEMUc} \qquad \text{(kg)}$$

The low procurement cost of these consumables is not modeled.

Manned maneuvering units (MMU):

The model assumes that a maneuvering unit is provided for each of the humans at the worksite. Therefore the total initial hardware requirement for manned maneuvering units =

$$\begin{aligned}\text{NtotMMUi} &= (\text{NtotHUMi pers}) * (1\ \text{mmu/pers}) \\ &= \text{NtotHUMi} \qquad \text{(mmus)}\end{aligned}$$

From INPUTS1:
WFmmu = Wearout factor for manned maneuv. units (mmus/mmu-hr)
KGunMMU = Mass of one manned maneuvering unit, including support hardware (kg/mmu)

The spares requirement is a function of the number of mmu-hours of space activity. These mmu-hrs are multiplied by a wearout factor to compute the equivalent number of mmus needed as spares.

For Task j, the total MMU usage is

$$\begin{aligned}&(\text{Nhum}[j]\ \text{pers}) * (1\ \text{mmu/pers}) \\ &\ * (\text{Ttime}[j]\ \text{days}) * (\text{HDhum hrs/day}) \\ &= \text{Nhum}[j] * \text{Ttime}[j] * \text{HDhum} \qquad \text{(mmu-hrs)}\end{aligned}$$

So the total MMU usage for the program =

$$\begin{aligned}\text{NtotMH} &= \sum_{j=1}^{\text{NTSK}} (\text{Nhum}[j] * \text{Ttime}[j] * \text{HDhum}) \\ &= \text{NtotEH} \qquad \text{(mmu-hrs)}\end{aligned}$$

And the total MMU spares requirement for the program =

$$\begin{aligned}\text{NtotMMUs} &= (\text{NtotMH mmu-hrs}) * (\text{WFmmu mmus/mmu-hr}) \\ &= \text{NtotMH} * \text{WFmmu} \qquad \text{(mmus)}\end{aligned}$$

So the total hardware procurement requirement for maneuvering units =

NtotMMU = NtotMMUi + NtotMMUs (mmus)

For simplicity, the cost model does not apply a learning curve to MMU procurement. Maneuvering units are high-tech hardware, so the cost is calculated by multiplying the total required mass of MMU by the dollar/kilogram factor for procurement of high-tech hardware.

So the total cost of procurement of MMU hardware =

CTprMMU = (NtotMMU mmus) * (KGunMMU kg/mmu) * (DKprH $/kg)
= NtotMMU * KGunMMU * DKprH ($)

The MMU consumables consist of nitrogen and batteries. The mass of these consumables is calculated by multiplying the total mmu-hours at the worksite by a kg/mmu-hr factor.

From INPUTS1:
KGmhMMUc = Kgs. of consumables required by one maneuvering unit in one hour (kg/mmu-hr)

So total consumables requirement for maneuvering units =

KGofMMUc = NtotMH * KGmhMMUc (kg)

The low procurement cost of these consumables is not modeled.

Miscellaneous tools and equipment (EQP):

The model assumes that one unit of tools and equipment is provided for each of the humans at the worksite. Therefore the total initial hardware requirement for tools and equipment =

NtotEQPi = (NtotHUMi pers) * (1 eqp/pers)
= NtotHUMi (eqps)

From INPUTS1:
WFeqp = Wearout factor for misc. tools and equip. (eqps/eqp-hr)
KGunEQP = Mass of one unit of misc. tools and equip. (kg/eqp)

The spares requirement is a function of the number of eqp-hours of space activity. These eqp-hrs are multiplied by a wearout factor to compute the equivalent number of eqps needed as spares.

For Task j, the total EQP usage is

(Nhum[j] pers) * (1 eqp/pers)
* (Ttime[j] days) * (HDhum hrs/day)
= Nhum[j] * Ttime[j] * HDhum (eqp-hrs)

So the total EQP usage for the program =

$$NtotQH = \sum_{j=1}^{NTSK} (Nhum[j] * Ttime[j] * HDhum)$$

$$= NtotEH \qquad \text{(eqp-hrs)}$$

And the total EQP spares requirement for the program =

$$NtotEQPs = (NtotQH \text{ eqp-hrs}) * (WFeqp \text{ eqps/eqp-hr})$$
$$= NtotQH * WFeqp \qquad \text{(eqps)}$$

So the total hardware procurement requirement for miscellaneous tools and equipment =

$$NtotEQP = NtotEQPi + NtotEQPs \qquad \text{(eqps)}$$

For simplicity, the cost model does not apply a learning curve to EQP procurement. Miscellaneous tools and equipment are high-tech hardware, so the cost is calculated by multiplying the total required mass of EQP by the dollar/kilogram factor for procurement of high-tech hardware.

So the total cost of procurement of EQP hardware =

$$CTprEQP = (NtotEQP \text{ eqps}) * (KGunEQP \text{ kg/eqp}) * (DKprH \text{ \$/kg})$$
$$= NtotEQP * KGunEQP * DKprH \qquad (\$)$$

The miscellaneous equipment and tools requires no consumables.

Therefore the total procurement cost for human support equipment =

$$CTprHSE = CTprEMU + CTprMMU + CTprEQP \qquad (\$)$$

B.7.6 Procurement of Teleoperators (TEL):

(Subroutine PROfTEL)

The teleoperator hardware to be procured consists of initial hardware and spares, and separate learning curves are applied to each. The initial hardware requirement is the initial number of teleoperators at the worksite (calculated by PRDTIME1 earlier). The spares requirement is computed from TEL usage. The mass of TEL consumables is calculated by multiplying the total tel-hours at the worksite by a kg/tel-hr factor.

From PRDTIME1:
NtotTELi = Initial number of teleoperators at the worksite (tels)
Ttime(j) = Time to complete task j (days)

From TASKS1:
NTSK = Number of different tasks to be performed (#)
Ntel(j) = Number of teleops assigned to task j (tels)

From INPUTS1:
KGunTEL = Mass of one teleoperator, including support hardware (kg/tel)
DKprH = Dollars of proc. cost per kg of high-tech hardware ($/kg)
HDtel = Hours/day worked by teleoperators (hrs/day)
WFtel = Wearout factor for teleoperators (tels/tel-hr)
lcparam = Learning curve parameter for hardware other than SST (%)

The total initial hardware requirement for teleoperators =
NtotTELi (tels)

The cost model assumes that teleoperators are entirely made of high-tech hardware. Therefore their first-item cost is computed by multiplying the mass of one teleoperator by the dollar/ kilogram factor for procurement of high-tech hardware. LERNCURV is then applied to this first-item cost and to the initial TEL requirement to generate the procurement cost of initial TEL hardware.

Thus the first-item TEL cost =
CFprTEL = KGunTEL * DKprH * 1 tel ($)

LERNCURV CFprTEL, NtotTELi, lcparam into CTprTELi
= total cost of procurement of initial TEL hardware ($)

The spares requirement for the teleops is based on the normal wear-and-tear during the program tasks. It is therefore a function of tel-hrs of work in space. These tel-hrs are multiplied by a wearout factor to calculate the equivalent number of tels needed as spares.

The cost model computes the total number of tel-hrs charged to this program by multiplying the teleoperator requirement for each task by the total duration of that task, and adding these individual products together. (This is similar to the HAB and HSE spares discussed above.)

For Task j, the total TEL usage is
(Ntel(j) tels) * (Ttime(j) days) * (HDtel hrs/day)
= Ntel(j) * Ttime(j) * HDtel (tel-hrs)

So the total TEL usage for the program =

$$\text{NtotTH} = \sum_{j=1}^{\text{NTSK}} (\text{Ntel}(j) * \text{Ttime}(j) * \text{HDtel}) \qquad \text{(tel-hrs)}$$

And the total TEL spares requirement for the program =

NtotTELs = (NtotTH tel-hrs) * (WFtel tels/tel-hr)
= NtotTH * WFtel (tels)

LERNCURV CFprTEL, NtotTELs, lcparam into CTprTELs
= total cost of procurement of TEL spares ($)

Then we have total procurement cost for TEL hardware =
CTprTEL = CTprTELi + CTprTELs ($)

The TEL consumables consist of nitrogen propellant and batteries. The mass of these consumables is calculated by multiplying the total tel-hours at the worksite by a kg/tel-hr factor.

From INPUTS1:
KGthTELc = Kgs. of consumables required by one teleoperator in one hour (kg/tel-hr)

So total consumables requirement for teleoperators =
KGofTELc = NtotTH * KGthTELc (kg)

The low procurement cost of these consumables is not modeled.

B.7.7 Procurement of Automated Equipment (AEQ):

(Subroutine PRofAEQ)

The automated equipment hardware to be procured consists of initial hardware and spares, and separate learning curves are applied to each. The initial hardware requirement is the initial number of units of automated equipment at the worksite (calculated by PRDTIME1 earlier). The spares requirement is computed from AEQ usage.

From PRDTIME1:
NtotAEQi = Initial number of units of automated equipment at the worksite (aeqs)
Ttime(j) = Time to complete task j (days)

From TASKS1:
NTSK = Number of different tasks to be performed (#)
Naeq(j) = Number of units of automated equipment assigned to task j (aeqs)

From INPUTS1:
KGunAEQ = Mass of one unit of automated equipment, including support hardware (kg/aeq)
DKprH = Dollars of proc. cost per kg of high-tech hardware ($/kg)
HDaeq = Hours/day worked by automated equipment (hrs/day)
WFaeq = Wearout factor for automated equipment (aeqs/aeq-hr)
lcparam = Learning curve parameter for hardware other than SST (%)

The total initial hardware requirement for units of automated equipment =
NtotAEQi (aeqs)

The cost model assumes that the units of automated equipment are entirely made of high-tech hardware. Therefore their first-item cost is computed by multiplying the mass of one unit of AEQ by the dollar/ kilogram factor for procurement of high-tech hardware. LERNCURV is then applied to this first-item cost and to the initial AEQ requirement to generate the procurement cost for initial AEQ hardware.

Thus the first-item AEQ cost =
CFprAEQ = KGunAEQ * DKprH * 1 aeq ($)

LERNCURV CFprAEQ, NtotAEQi, lcparam into CTprAEQi
= total cost of procurement of initial AEQ hardware ($)

The spares requirement for the automated equipment is based on the normal wear-and-tear during the program tasks. It is therefore a function of aeq-hrs of work in space. These aeq-hrs are multiplied by a wearout factor to calculate the equivalent number of aeqs needed as spares.

The cost model computes the total number of aeq-hrs charged to this program by multiplying the automated equipment requirement for each task by the total duration of that task, and adding these individual products together. (This is similar to the HAB and HSE spares discussed above.)

For Task j, the total AEQ usage is
(Naeq(j) aeqs) * (Ttime(j) days) * (HDaeq hrs/day)
= Naeq(j) * Ttime(j) * HDaeq (aeq-hrs)

So the total AEQ usage for the program =

$$\text{NtotAH} = \sum_{j=1}^{\text{NTSK}} (\text{Naeq}(j) * \text{Ttime}(j) * \text{HDaeq}) \qquad \text{(aeq-hrs)}$$

And the total AEQ spares requirement for the program =
NtotAEQs = (NtotAH aeq-hrs) * (WFaeq aeqs/aeq-hr)
= NtotAH * WFaeq (aeqs)

LERNCURV CFprAEQ, NtotAEQs, lcparam into CTprAEQs
= total cost of procurement of AEQ spares ($)

Then we have total procurement cost for AEQ hardware =
CTprAEQ = CTprAEQi + CTprAEQs ($)

The automated equipment requires no consumables. For example, a beam-builder requires no propellant, and would not require batteries if it were cableconnected to its power system.

B.7.8 Procurement of Construction Support Equipment (CSE):

(Subroutine PROfCSE)

The construction support equipment is low-tech equipment, e.g. construction jigs. The cost model assumes that it requires no spares and no consumables, and that there is no learning curve in its procurement. Therefore its cost is calculated by multiplying the total required mass of CSE by the dollar/kilogram factor for procurement of low-tech hardware.

From INPUTS1:
NtotCSE = Total number of units of construction support equipment (units)
KGunCSE = Mass of one unit of construction support equipment (kg/unit)
DKprL = Dollars of proc. cost per kg of low-tech hardware ($/kg)

Total procurement cost for construction support equipment =
CTprCSE = NtotCSE * KGunCSE * DKprL ($)

B.7.9 Procurement of Construction Site Propellant (CSP):

(Subroutine PROfCSP)

The construction site propellant is needed to compensate for the orbital drag of the worksite. The model assumes an orbital altitude of 260 km for the worksite. Calculations to estimate the average drag of the radiometer during construction, and the associated daily requirement for propellant, are presented in Appendix D.

The total CSP mass required is calculated by multiplying the daily requirement by the total time to complete the tasks, i.e. the time to complete the longest task (which was computed by PRDTIME1).

From PRDTIME1:
Jobtime = Time to complete all space tasks (days)

From INPUTS1:
KDcsp = Daily requirement of construction site propellant (kg/day)

Total requirement for construction site propellant =
KGofCSP = (KDcsp kg/day) * (Jobtime days)
= KDcsp * Jobtime (kg)

The procurement cost for this propellant is expected to be small compared to its launch cost, and therefore it is not modeled.

B.7.10 Training of Humans in Space Crews (HUMt):

(Subroutine PRofHUMt)

The cost model assumes that the humans in the space crews are on a rotation schedule: one stint in space (e.g. three months), one stint on the ground. Therefore two complements of humans alternate at the worksite, and both complements must be trained prior to the start of the space tasks. The cost of training is estimated by multiplying the number of humans to be trained by a dollar/person factor. (Note: this cost does not include the salaries of the crew members during training; they are handled in the next section below.)

From PRDTIME1:
NtotHUMi = Initial number of humans at the worksite (pers)

From INPUTS1:
DpHUMt = Dollars/person for training of space crew humans ($/pers)

Total number of space crew humans requiring training =
NtotHUMt = 2 * NtotHUMi (pers)

Total cost of training space crew humans =
CTprHUMt = NtotHUMt * DpHUMt ($)

B.7.11 Salary and Overhead for Humans in Space Crews (HUMb):

(Subroutine PRofHUMb)

The salary and overhead costs are computed by multiplying the total number of person-days by a dollar/person-day factor.

The cost model assumes that the humans in the space crews are on a rotation schedule: one stint in space, one stint on the ground. Therefore two complements of humans alternate at the worksite, and both complements are paid full-time, in space or on the ground. In addition, all the humans receive NdHUMt days of training before the start of the space tasks, and they draw salary during that time.

The cost model calculates the total number of person-days to be paid by multiplying the number of humans working on each task by the total time (training plus task), and adding the individual totals together.

From PRDTIME1:
NtotHUMi = Initial number of humans at the worksite (pers)

From PROfHAB:
NtotPD = Total number of person-days in space for the program (pers-days)

From INPUTS1:
NdHUMt = Number of days of training for humans in space crews (days)
DpdHUM = Salary and overhead for one human in space for one day ($/pers-day)

For task j, the number of person-days to be paid is

$$2 * (\mathrm{Nhum}(j)\ \mathrm{pers}) * [(\mathrm{NdHUMt\ days}) + (\mathrm{Ttime}(j)\ \mathrm{days})]$$
$$= 2 * \mathrm{Nhum}(j) * [\mathrm{NdHUMt} + \mathrm{Ttime}(j)] \qquad (\text{pers-days})$$

So the total number of person-days to be paid =

$$\mathrm{NtotPDb} = \sum_{j=1}^{\mathrm{NTSK}} 2 * \mathrm{Nhum}(j) * [\mathrm{NdHUMt} + \mathrm{Ttime}(j)]$$

$$= 2 * \sum_{j=1}^{\mathrm{NTSK}} \mathrm{Nhum}(j) * \mathrm{NdHUMt} \quad + \quad 2 * \sum_{j=1}^{\mathrm{NTSK}} \mathrm{Nhum}(j) * \mathrm{Ttime}(j)$$

$$= (2 * \mathrm{NdHUMt} * \mathrm{NtotHUMi}) + (2 * \mathrm{NtotPD}) \qquad (\text{pers-days})$$

So salary and overhead for humans in space crews=

$$\mathrm{CTprHUMb} = (\mathrm{NtotPDb\ pers\text{-}days}) * (\mathrm{DpdHUM}\ \$/\mathrm{pers\text{-}day})$$
$$= \mathrm{NtotPDb} * \mathrm{DpdHUM} \qquad (\$)$$

B.7.12 Procurement of Ground Support Equipment (GSE):

(Subroutine PRofGSE)

The model assumes that ground support equipment (GSE) consists of general monitoring stations (GMS), human monitoring stations (HMS), teleoperator control stations (TCS), and automated equipment monitoring stations (AMS). For each of these monitoring and control stations, the hardware to be procured consists of initial hardware and spares, and separate learning curves are applied.

The general monitoring stations provide overall worksite support: their initial hardware requirement is a direct input to the model, and their spares requirement depends on the total time for the entire job. The HMS, TCS, and AMS initial hardware requirements are based on the initial numbers of humans, teleoperators, and automated equipment at the worksite, respectively; their spares requirements depend on person-hours, teleoperator-hours, and automated-equipment-hours in space.

Other types of ground support equipment are included in other cost items. For example, the costs of ground simulators for training are included in the R&D cost for task procedures and the procurement cost for training humans; and the cost of pre-launch support equipment (e.g. payload cradles and checkout equipment) is included in the mission preparation and launch costs.

General monitoring stations (GMS):

The GMS hardware to be procured consists of initial hardware and spares, and separate learning curves are applied to each.

From PRDTIME1:
Jobtime = Time to complete all space tasks (days)

From INPUTS1:
NgmsGEN = Number of general monitoring stations for general worksite support (gms)
WFgms = Wearout factor for general monitoring stations (gms/gms-hr)
DUprGMS = Dollar/unit cost of general monitoring stations ($/gms)
lcparam = Learning curve parameter for hardware other than SST (%)

The cost model assumes that NgmsGEN worksite monitoring stations provide general support for the worksite for the duration of the program, regardless of the level of activity in space. Therefore the total initial requirement for general monitoring stations =

NtotGMSi = NgmsGEN (gms)

The cost model takes DUprGMS as the first-item cost for worksite monitoring stations. LERNCURV is then applied to this first-item cost and to the initial GMS requirement to generate the procurement cost for initial GMS hardware.

LERNCURV DUprGMS, NtotGMSi, lcparam into CTprGMSi
= total cost of procurement of initial GMS hardware ($)

The spares requirement for the general monitoring stations is based on the normal wear-and-tear during the program tasks. It is therefore a function of gms-hrs of work. These gms-hrs are multiplied by a wearout factor to calculate the equivalent number of gms needed as spares.

The cost model computes the total number of gms-hrs of general worksite support by multiplying the GMS requirement by the time to complete the whole job. These stations operate around the clock, so this GMS usage =

NtotGMSH = (NtotGMSi gms) * (Jobtime days) * (24 hrs/day)
= NtotGMSi * Jobtime * 24 (gms-hrs)

And the total GMS spares requirement for the program =

NtotGMSs = (NtotGMSH gms-hrs) * (WFgms gms/gms-hr)
= NtotGMSH * WFgms (gms)

LERNCURV DUprGMS, NtotGMSs, lcparam into CTprGMSs
= total cost of procurement of GMS spares ($)

Then the total cost of procurement of GMS hardware =

CTprGMS = CTprGMSi + CTprGMSs ($)

Human monitoring stations (HMS):

The HMS hardware to be procured consists of initial hardware and spares, and separate learning curves are applied to each.

From PRDTIME1:
NtotHUMi = Initial number of humans at the worksite (pers)

From INPUTS1:
NhmsHUM = Number of human monitoring stations to support one human in space (hms/pers)
WFhms = Wearout factor for human monitoring stations (hms/hms-hr)
DUprHMS = Dollar/unit cost of human monitoring stations ($/hms)
lcparam = Learning curve parameter for hardware other than SST (%)

From PROfHSE:
NtotEH = Total usage of pressure suits for the program (emu-hrs)

The cost model assumes that each human at work in space requires NhmsHUM human monitoring stations on the ground. Therefore the total initial requirement for human monitoring stations =

$$\text{NtotHMSi} = \text{NtotHUMi} * \text{NhmsHUM} \qquad \text{(hms)}$$

The cost model takes DUprHMS as the first-item cost for human monitoring stations. LERNCURV is then applied to this first-item cost and to the initial HMS requirement to generate the procurement cost for initial HMS hardware.

LERNCURV DUprHMS, NtotHMSi, lcparam into CTprHMSi
= total cost of procurement of initial HMS hardware ($)

The spares requirement for the human monitoring stations is based on the normal wear-and-tear during the program tasks. It is therefore a function of hms-hrs of work. These hms-hrs are multiplied by a wearout factor to calculate the equivalent number of hms needed as spares.

The model computes the total number of hms-hrs by multiplying the human monitoring station requirement for each task by the total duration of that task, and adding these individual products together.

For Task j, the total HMS usage is

$$\begin{aligned} &((\text{Nhum}[j]\ \text{pers}) * (\text{NhmsHUM hms/pers})) \\ &\quad * (\text{Ttime}[j]\ \text{days}) * (\text{HDhum hrs/day}) \\ &= \text{Nhum}[j] * \text{NhmsHUM} * \text{Ttime}[j] * \text{HDhum} \qquad \text{(hms-hrs)} \end{aligned}$$

So the total HMS usage for the program =

$$\text{NtotHMSH} = \sum_{j=1}^{\text{NTSK}} \text{Nhum}[j] * \text{Ttime}[j] * \text{HDhum} * \text{NhmsHUM}$$

$$= \text{NtotEH} * \text{NhmsHUM} \qquad \text{(hms-hrs)}$$

And the total HMS spares requirement for the program =

$$\begin{aligned} \text{NtotHMSs} &= (\text{NtotHMSH hms-hrs}) * (\text{WFhms hms/hms-hr}) \\ &= \text{NtotHMSH} * \text{WFhms} \qquad \text{(hms)} \end{aligned}$$

LERNCURV DUprHMS, NtotHMSs, lcparam into CTprHMSs
= total cost of procurement of HMS spares ($)

Then the total cost of procurement of HMS hardware =
CTprHMS = CTprHMSi + CTprHMSs ($)

Teleoperator control stations (TCS):

The TCS hardware to be procured consists of initial hardware and spares, and separate learning curves are applied to each.

From PRDTIME1:
NtotTELi = Initial number of teleoperators at the worksite (tels)

From PRofTEL:
NtotTH = Total teleoperator usage for the program (tel-hrs)

From INPUTS1:
NtcsTEL = Number of teleoperator control stations to support one teleoperator in space (tcs/tel)
WFtcs = Wearout factor for teleop. control stations (tcs/tcs-hr)
DUprTCS = Dollar/unit cost of teleop. control stations ($/tcs)
lcparam = Learning curve parameter for hardware other than SST (%)

The cost model assumes that each teleoperator at work in space requires NtcsTEL teleoperator control stations on the ground. Therefore the total initial requirement for teleoperator control stations =
NtotTCSi = NtotTELi * NtcsTEL (tcs)

The cost model takes DUprTCS as the first-item cost for teleoperator control stations. LERNCURV is then applied to this first-item cost and to the initial TCS requirement to generate the procurement cost for initial TCS hardware.

LERNCURV DUprTCS, NtotTCSi, lcparam into CTprTCSi
= total cost of procurement of initial TCS hardware ($)

The spares requirement for the teleoperator control stations is based on the normal wear-and-tear during the program tasks. It is therefore a function of tcs-hrs of work. These tcs-hrs are multiplied by a wearout factor to calculate the equivalent number of tcs needed as spares.

The cost model computes the total number of tcs-hrs charged to this program by multiplying the teleoperator control station requirement for each task by the total duration of that task, and adding these individual products together.

For Task j, the total TCS usage is
(Ntel[j] tels) * (NtcsTEL tcs/tel)
* (Ttime[j] days) * (HDtel hrs/day)
= Ntel[j] * NtcsTEL * Ttime[j] * HDtel (tcs-hrs)

So the total TCS usage for the program =

$$\mathrm{NtotTCSH} = \sum_{j=1}^{\mathrm{NTSK}} \mathrm{Ntel}[j] * \mathrm{Ttime}[j] * \mathrm{HDtel} * \mathrm{NtcsTEL}$$

$$= \mathrm{NtotTH} * \mathrm{NtcsTEL} \qquad \text{(tcs-hrs)}$$

And the total TCS spares requirement for the program =
NtotTCSs = (NtotTCSH tcs-hrs) * (WFtcs tcs/tcs-hr)
= NtotTCSH * WFtcs (tcs)

LERNCURV DUprTCS, NtotTCSs, lcparam into CTprTCSs
= total cost of procurement of TCS spares ($)

Then the total cost of procurement of TCS hardware =
CTprTCS = CTprTCSi + CTprTCSs ($)

Automated equipment monitoring stations (AMS):

The AMS hardware to be procured consists of initial hardware and spares, and separate learning curves are applied to each.

From PRDTIME1:
NtotAEQi = Initial number of units of automated equipment at the worksite (aeqs)

From INPUTS1:
NamsAEQ = Number of auto. equipment monitoring stations to support one unit of auto. equipment in space (ams/aeq)
WFams = Wearout factor for auto. equipment monitoring stations (ams/ams-hr)
DUprAMS = Dollar/unit cost of auto. equipment monitoring stations ($/ams)
lcparam = Learning curve parameter for hardware other than SST (%)

From PROfAEQ:
NtotAH = Total usage of automated equipment for the program (aeq-hrs)

The cost model assumes that each unit of automated equipment at work in space requires NamsAEQ automated equipment monitoring stations on the ground. Therefore the total initial requirement for automated equipment monitoring stations =

NtotAMSi = NtotAEQi * NamsAEQ (ams)

The cost model takes DUprAMS as the first-item cost for automated equipment monitoring stations. LERNCURV is then applied to this first-item cost and to the initial AMS requirement to generate the procurement cost for initial AMS hardware.

LERNCURV DUprAMS, NtotAMSi, lcparam into CTprAMSi
= total cost of procurement of initial AMS hardware ($)

The spares requirement for the automated equipment monitoring stations is based on the normal wear-and-tear during the program tasks. It is therefore a function of ams-hrs of work. These ams-hrs are multiplied by a wearout factor to calculate the equivalent number of ams needed as spares.

The model computes the total number of ams-hrs by multiplying the automated equipment monitoring station requirement <u>for each task</u> by the total duration of that task, and adding these individual products together.

For Task j, the total AMS usage is

$$((\text{Naeq}[j]\ \text{aeqs}) * (\text{NamsAEQ ams/aeq})) * (\text{Ttime}[j]\ \text{days}) * (\text{HDaeq hrs/day})$$

$$= \text{Naeq}[j] * \text{NamsAEQ} * \text{Ttime}[j] * \text{HDaeq} \qquad \text{(ams-hrs)}$$

So the total AMS usage for the program =

$$\text{NtotAMSH} = \sum_{j=1}^{\text{NTSK}} \text{Naeq}[j] * \text{Ttime}[j] * \text{HDaeq} * \text{NamsAEQ}$$

$$= \text{NtotAH} * \text{NamsAEQ} \qquad \text{(ams-hrs)}$$

And the total AMS spares requirement for the program =

NtotAMSs = (NtotAMSH ams-hrs) * (WFams ams/ams-hr)
= NtotAMSH * WFams (ams)

LERNCURV DUprAMS, NtotAMSs, lcparam into CTprAMSs
= total cost of procurement of AMS spares ($)

Bhen the total cost of procurement of AMS hardware =
CTprAMS = CTprAMSi + CTprAMSs ($)

B.7.13 Calculation of Overall Procurement Costs (CTpr):

(Subroutine PROCURE1 cont.)

```
CTpr =    CTprSST
       + CTprHAB + CTprHSE
       + CTprTEL + CTprAEQ
       + CTprCSE + CTprHUMt
       + CTprHUMb + CTprGSE
```

(CTpr, NtotHABh, NtotHABl, NtotPD, KGofHABc, NtotEMU, KGofEMUc, NtotMMU, KGofMMUc, NtotEQP, NtotTEL, KGofTELc, NtotAEQ, KGofCSP, NtotEH, NtotTH, NtotAH, NtotGMSH, NtotHMSH, NtotTCSH, NtotAMSH, CTprSST, CTprHAB, CTprHSE, CTprTEL, CTprAEQ, CTprCSE are global variables)

(End of PROCURE1)

B.8 Calculation of Mission Preparation and Launch Costs:

(Subroutine LAUNCH1)

The following mission preparation and launch cost elements are reprinted from Table B.1:

Mission Preparation and Launch:
- Mission prep. and launch of space structure to be assembled
- Launch of humans in space crews
- Mission prep. and launch of human habitation modules (including spares and consumables)
- Mission prep. and launch of human support equipment (including spares and consumables)
- Mission prep. and launch of teleoperators (including spares and consumables)
- Mission prep. and launch of automated equipment (including spares)
- Mission prep. and launch of construction support equipment
- Launch of construction site propellant
- Launch insurance

This cost category includes two types of costs, mission preparation and launch. Mission preparation is the checkout, packaging, and integration of payloads into the Shuttle, prior to launch. These costs are modeled by multiplying the masses to be prepared by dollar/kilogram factors. Two different factors are used for high-tech and low-tech hardware.

The cost model uses three fundamental options to calculate launch costs. For the space structure to be assembled, the model checks the packing density of the structure and decides whether mass or volume dominates in the Shuttle bay. It then applies either a dollars/kilogram or a dollars/cubic-meter factor to calculate launch cost.

For the humans in the space crew, the model calculates the number of person-launches required during the program, and multiplies this by a dollar/person-launch factor to compute their launch cost (this includes the cost of return to earth as well). This is not applied to the Shuttle pilots; they are part of the Shuttle launch service.

For other hardware, the model assumes sufficient density that these materials are launch-costed by mass, using the dollars/kilogram factor. All of the material to be launched incurs launch costs. However, the mission preparation costs for consumables and construction site propellant are not modeled, because these costs are expected to be very small.

The model does not apply any learning curve to the launch costs, because the actual learning curve would be over all Shuttle uses, not just those for this project. Therefore the launch cost factors are set prices, reflecting that larger learning curve. In effect, the launch cost factors are administrative decisions, based on the Shuttle owner's pricing policy. In this analysis, it is expected that those launch cost factors will be varied as parameters to the model.

This section of the cost model also calculates insurance premiums for the launches. These premiums are based on the replacement cost of the hardware being launched, i.e. the sum of its procurement and launch costs. The procurement costs (calculated earlier) are added to the launch costs computed here, and their sum is multiplied by an insurance factor.

B.8.1 Mission Preparation and Launch of Space Structure to be Assembled (SST):

(Subroutine LAofSST)

The space structure is modeled as a number of units of high-tech hardware and a (different) number of units of low-tech hardware. Each of these types of hardware is treated separately.

Depending on packing density, launch cost for space structure is based either on mass (kg) or volume (cubic meters). The Shuttle is assumed to carry 27,281 kg (60,155 lbs) to 370-km altitude, in a payload bay roughly 4.5 meters diameter x 18 meters long (15 ft. diameter x 60 ft. long). So volume is roughly 295 cubic meters, for a crossover density of 92.5 kg/cubic meter.

High-tech space structure:

From INPUTS1:
NunSSTh = Number of units of high-tech space structure (units)
KGunSSTh = Mass of one unit of high-tech space structure (kg/unit)
DKmpH = Dollar/kg cost to mission-prep. high-tech hardware ($/kg)
DensSSTh = Packing density of high-tech space structure
(kg/cubic meter)
DKla = Dollars to launch one kilogram on STS ($/kg)
DMla = Dollars to launch one cubic meter on STS ($/cubic meter)

The mission preparation cost for high-tech space structure is the product of the mass to be launched and a dollar/kilogram factor. This cost is

(NunSSTh units) * (KGunSSTh kg/unit) * (DKmpH $/kg)
= NunSSTh * KGunSSTh * DKmpH ($)

If (DensSSTh kg/cubic meter) $\underline{>}$ (92.5 kg/cubic meter) then mass dominates, and the launch-only cost is the product of the mass to be launched and DKla. This cost is

(NunSSTh units) * (KGunSSTh kg/unit) * (DKla $/kg)
= NunSSTh * KGunSSTh * DKla ($)

Then the total cost to launch the high-tech space structure =

CTlaSSTh = (NunSSTh * KGunSSTh * DKmpH)
+ (NunSSTh * KGunSSTh * DKla) ($)

Otherwise, (DensSSTh kg/cubic meter) < (92.5 kg/cubic meter) and volume dominates, and the launch-only cost is the product of the volume to be launched and DMla. This cost is

((NunSSTh * KGunSSTh kg) / (DensSSTh kg/cubic meter))
* (DMla $/cubic meter)
= (NunSSTh * KGunSSTh / DensSSTh) * DMla ($)

Then the total cost to launch the high-tech space structure =

CTlaSSTh = (NunSSTh * KGunSSTh * DKmpH)
+ (NunSSTh * KGunSSTh / DensSSTh) * DMla ($)

Low-tech space structure:

From INPUTS1:
NunSSTl = Number of units of low-tech space structure (units)
KGunSSTl = Mass of one unit of low-tech space structure (kg/unit)
DKmpL = Dollar/kg cost to mission-prep. low-tech hardware ($/kg)
DensSSTl = Packing density of low-tech space structure
(kg/cubic meter)

The mission preparation cost for low-tech space structure is the product of the mass to be launched and a dollar/kilogram facter. This cost is

(NunSSTl units) * (KGunSSTl kg/unit) * (DKmpL $/kg)
= NunSSTl * KGunSSTl * DKmpL ($)

If (DensSSTl kg/cubic meter) $\geq$ (92.5 kg/cubic meter) then mass dominates, and the launch-only cost is the product of the mass to be launched and DKla. This cost is

(NunSSTl units) * (KGunSSTl kg/unit) * (DKla \$/kg)

= NunSSTl * KGunSSTl * DKla (\$)

Then the total cost to launch the low-tech space structure =

CTlaSSTl = (NunSSTl * KGunSSTl * DKmpL)

+ (NunSSTl * KGunSSTl * DKla) (\$)

Otherwise, (DensSSTl kg/cubic meter) < (92.5 kg/cubic meter) and volume dominates, and the launch-only cost is the product of the volume to be launched and DMla. This cost is

((NunSSTl * KGunSSTl kg) / (DensSSTl kg/cubic meter))

* (DMla \$/cubic meter)

= (NunSSTl * KGunSSTl / DensSSTl) * DMla (\$)

Then the total cost to launch the low-tech space structure =

CTlaSSTl = (NunSSTl * KGunSSTl * DKmpL)

+ (NunSSTl * KGunSSTl / DensSSTl) * DMla (\$)

Complete structure:

From INPUTS1:

FRlaINS = Fraction of procurement and launch cost paid as insurance premium (\$/\$)

From PROCURE1:

CTprSST = Total proc. cost for space structure to be assembled (\$)

Total launch cost for space structure to be assembled =

CTlaSST = CTlaSSTh + CTlaSSTl (\$)

Launch insurance is computed as a percentage of the procurement and launch costs of the hardware. So total launch insurance premium for space structure =

CTinSST = (CTprSST + CTlaSST) * FRlaINS (\$)

B.8.2 Launch of Humans (HUM):

(Subroutine LAofHUM)

The cost model assumes that the humans at the worksite work for a tour of duty (called a "stint") and are then rotated groundside for the next stint. For example, if each stint is three months long, each human works three months in space, then spends the next three months on the ground. Therefore two complements of humans alternate at the worksite.

The model calculates the number of person-stints required during the program, which is also the number of person-launches. This is multiplied by a dollar/person-launch factor to compute the cost of launching the human crew members. This includes the cost of return to earth as well.

To avoid integral step functions in these costs, fractional person-launches are allowed. For example, if a human finishes all assigned tasks halfway through a stint, then the program incurs only half of the launch cost for that stint. This assumption is not entirely realistic, but sensitivity analyses (presented in Appendix F) indicate that the personal launch cost for humans has relatively little effect on the program cost. Therefore this simplifying assumption is acceptable.

From PROCURE1:
NtotPD = Total number of person-days in space for the program (pers-days)

From INPUTS1:
NDstint = Number of days in one human tour of duty in space (days/stint)
DPla = Dollars to launch one human ($/pers-launch)

For task j, the number of person-launches required is

$$\begin{aligned}&(\text{Nhum}(j)\ \text{pers}) * [(\text{Ttime}(j)\ \text{days}) / (\text{NDstint days/stint})] \\ &\quad * (1\ \text{pers-launch} / 1\ \text{pers-stint}) \\ &= \text{Nhum}(j) * (\text{Ttime}(j) / \text{NDstint}) \qquad \text{(pers-launch)}\end{aligned}$$

The total number of person-launches required by the program is the sum of the requirements for the individual tasks =

$$\text{HUMtotLA} = \sum_{j=1}^{\text{NTSK}} \text{Nhum}(j) * (\text{Ttime}(j) / \text{NDstint})$$

$$= \text{NtotPD} / \text{NDstint} \qquad \text{(pers-launch)}$$

So the total cost to launch human crew members =

$$\text{CTlaHUM} = \text{HUMtotLA} * \text{DPla} \qquad (\$)$$

There is no insurance premium on the launch of people.

B.8.3 Mission Preparation and Launch of Habitation Modules (HAB):

(Subroutine LAofHAB)

As described in the Procurement section above, the total HAB hardware requirement consists of initial habitation modules (high-tech and low-tech hardware) and high-tech spares. All of these must be mission-prepped and launched.

The habitation consumables are the crew consumables, e.g. oxygen, food, and water. The model assumes that mission preparation of consumables cost very little, and it is therefore not modeled. The cost of launch of hardware and consumables is the product of the mass to be launched and a dollar/kilogram factor.

From PROCURE1:
NtotHABh = Total high-tech hardware requirement for HAB modules (habs)
NtotHABl = Total low-tech hardware requirement for HAB modules (habs)
KGofHABc = Total consumables requirement for HAB modules (kg)

From INPUTS1:
KGunHABh = Mass of high-tech hardware in one habitation module (kg/hab)
DKmpH = Dollar/kg cost to mission-prep. high-tech hardware ($/kg)
DKla = Dollars to launch one kilogram on STS ($/kg)
KGunHABl = Mass of low-tech hardware in one habitation module (kg/hab)
DKmpL = Dollar/kg cost to mission-prep. low-tech hardware ($/kg)

The mission preparation cost for high-tech hardware is the product of the mass to be launched and a dollar/kilogram factor. For the high-tech HAB hardware, this cost is

$$(\text{NtotHABh habs}) * (\text{KGunHABh kg/hab}) * (\text{DKmpH \$/kg}) = \text{NtotHABh} * \text{KGunHABh} * \text{DKmpH} \qquad (\$)$$

The launch-only cost for the hardware is the product of the mass to be launched and DKla. For the high-tech HAB hardware, this cost is

$$(\text{NtotHABh habs}) * (\text{KGunHABh kg/hab}) * (\text{DKla \$/kg}) = \text{NtotHABh} * \text{KGunHABh} * \text{DKla} \qquad (\$)$$

Similarly, the mission preparation cost for low-tech HAB hardware is

$$(\text{NtotHABl habs}) * (\text{KGunHABl kg/hab}) * (\text{DKmpL \$/kg}) = \text{NtotHABl} * \text{KGunHABl} * \text{DKmpL} \qquad (\$)$$

And the launch cost for low-tech HAB hardware is

$$(\text{NtotHABl habs}) * (\text{KGunHABl kg/hab}) * (\text{DKla \$/kg}) = \text{NtotHABl} * \text{KGunHABl} * \text{DKla} \qquad (\$)$$

The launch cost for the HAB consumables is

$$(\text{KGofHABc kg}) * (\text{DKla \$/kg}) = \text{KGofHABc} * \text{DKla} \qquad (\$)$$

Then the total cost to launch the habitation modules (including spares and consumables) =

```
     CTlaHAB =  (NtotHABh * KGunHABh * DKmpH)
              + (NtotHABh * KGunHABh * DKla)
              + (NtotHABl * KGunHABl * DKmpL)
              + (NtotHABl * KGunHABl * DKla)
              + (KGofHABc * DKla)                         ($)
```

From INPUTS1:
FRlaINS = Fraction of procurement and launch cost paid as insurance premium ($/$)

From PROCURE1:
CTprHAB = Total procurement cost for habitation modules ($)

Launch insurance is computed as a percentage of the procurement and launch costs of the hardware. So total launch insurance premium for habitation modules =

```
     CTinHAB = (CTprHAB + CTlaHAB) * FRlaINS              ($)
```

B.8.4 Mission Preparation and Launch of Human Support Equipment (HSE):

(Subroutine LAofHSE)

As described in the Procurement section above, human support equipment consists of extravehicular mobility units (EMU), manned maneuvering units (MMU), and miscellaneous tools and equipment (EQP). The hardware for each of these includes initial equipment and spares, which must be mission-prepped and launched. The model assumes that EMU, MMU, and EQP are all high-tech hardware.

The EMU consumables consist of oxygen, water, air-scrubbing canisters, and batteries. The MMU consumables are nitrogen and batteries. The EQP does not require consumables.

The very low cost of mission preparation of consumables is not modeled. The cost of launch of hardware and consumables is the product of the mass to be launched and a dollar/kilogram factor.

Extravehicular mobility units (EMU):

From PROCURE1:
NtotEMU = Total hardware requirement for pressure suits (emus)
KGofEMUc = Total consumables requirement for pressure suits (kg)

From INPUTS1:
KGunEMU = Mass of one extravehicular mobility unit, including support hardware (kg/emu)
DKmpH = Dollar/kg cost to mission-prep. high-tech hardware ($/kg)
DK1a = Dollars to launch one kilogram on STS ($/kg)

The mission preparation cost for high-tech hardware is the product of the mass to be launched and a dollar/kilogram factor. For the EMU hardware, this cost is

(NtotEMU emus) * (KGunEMU kg/emu) * (DKmpH $/kg)
= NtotEMU * KGunEMU * DKmpH ($)

The launch-only cost for the hardware is the product of the mass to be launched and DK1a. This cost is

(NtotEMU emus) * (KGunEMU kg/emu) * (DK1a $/kg)
= NtotEMU * KGunEMU * DK1a ($)

The launch cost for the EMU consumables is

(KGofEMUc kg) * (DK1a $/kg)
= KGofEMUc * DK1a ($)

Then the total cost to launch the pressure suits (including spares and consumables) =

CT1aEMU = (NtotEMU * KGunEMU * DKmpH)
+ (NtotEMU * KGunEMU * DK1a)
+ (KGofEMUc * DK1a) ($)

Manned maneuvering units (MMU):

From PROCURE1:
NtotMMU = total requirement for manned maneuvering units (mmus)
KGofMMUc = Total consumables requirement for maneuvering units (kg)

From INPUTS1:
KGunMMU = Mass of one manned maneuvering unit, including support hardware (kg/mmu)
DKmpH = Dollar/kg cost to mission-prep. high-tech hardware ($/kg)
DK1a = Dollars to launch one kilogram on STS ($/kg)

The mission preparation cost for high-tech hardware is the product of the mass to be launched and a dollar/kilogram factor. For the MMU hardware, this cost is

(NtotMMU mmus) * (KGunMMU kg/mmu) * (DKmpH $/kg)
= NtotMMU * KGunMMU * DKmpH ($)

The launch-only cost for the hardware is the product of the mass to be launched and DK1a. This cost is

(NtotMMU mmus) * (KGunMMU kg/mmu) * (DK1a $/kg)
= NtotMMU * KGunMMU * DK1a ($)

The launch cost for the MMU consumables is
(KGofMMUc kg) * (DKla $/kg)
= KGofMMUc * DKla ($)

Then the total cost to launch the maneuvering units (including spares and consumables) =
CTlaMMU = (NtotMMU * KGunMMU * DKmpH)
+ (NtotMMU * KGunMMU * DKla)
+ (KGofMMUc * DKla) ($)

Miscellaneous tools and equipment (EQP):

From PROCURE1:
NtotEQP = total requirement for miscellaneous tools and equipment (eqps)

From INPUTS1:
KGunEQP = Mass of one unit of misc. tools and equip. (kg/eqp)
DKmpH = Dollar/kg cost to mission-prep. high-tech hardware ($/kg)
DKla = Dollars to launch one kilogram on STS ($/kg)

The mission preparation cost for high-tech hardware is the product of the mass to be launched and a dollar/kilogram factor. For the EQP hardware, this cost is
(NtotEQP eqps) * (KGunEQP kg/eqp) * (DKmpH $/kg)
= NtotEQP * KGunEQP * DKmpH ($)

The launch-only cost for the hardware is the product of the mass to be launched and DKla. This cost is
(NtotEQP eqps) * (KGunEQP kg/eqp) * (DKla $/kg)
= NtotEQP * KGunEQP * DKla ($)

Then the total cost to launch the miscellaneous tools and equipment (including spares) =
CTlaEQP = (NtotEQP * KGunEQP * DKmpH)
+ (NtotEQP * KGunEQP * DKla) ($)

Therefore the total cost to launch the human support equipment =
CTlaHSE = CTlaEMU + CTlaMMU + CTlaEQP ($)

From INPUTS1:
FRlaINS = Fraction of procurement and launch cost paid as insurance premium ($/$)

From PROCURE1:
CTprHSE = total procurement cost for human support equipment ($)

Launch insurance is computed as a percentage of the procurement and launch costs of the hardware. So total launch insurance premium for human support equipment =
CTinHSE = (CTprHSE + CTlaHSE) * FRlaINS ($)

B.8.5 Mission Preparation and Launch of Teleoperators (TEL):

(Subroutine LAofTEL)

The teleoperator hardware consists of initial equipment and spares, which must be mission-prepped and launched. Teleoperators are high-tech hardware.

The TEL consumables consist of nitrogen and batteries. The very low cost of mission preparation of consumables is not modeled. The cost of launch of hardware and consumables is the product of the mass to be launched and a dollar/kilogram factor.

From PROCURE1:
NtotTEL = total requirement for teleoperators (tels)
KGofTELc = Total consumables requirement for teleoperators (kg)

From INPUTS1:
KGunTEL = Mass of one teleoperator, including support hardware (kg/tel)
DKmpH = Dollar/kg cost to mission-prep. high-tech hardware ($/kg)
DK1a = Dollars to launch one kilogram on STS ($/kg)

The mission preparation cost for high-tech hardware is the product of the mass to be launched and a dollar/kilogram factor. For the TEL hardware, this cost is
(NtotTEL tels) * (KGunTEL kg/tel) * (DKmpH $/kg)
= NtotTEL * KGunTEL * DKmpH ($)

The launch-only cost for the hardware is the product of the mass to be launched and DK1a. This cost is
(NtotTEL tels) * (KGunTEL kg/tel) * (DK1a $/kg)
= NtotTEL * KGunTEL * DK1a ($)

The launch cost for the TEL consumables is
(KGofTELc kg) * (DK1a $/kg)
= KGofTELc * DK1a ($)

Then the total cost to launch the teleoperators (including spares and consumables) =
CT1aTEL = (NtotTEL * KGunTEL * DKmpH)
+ (NtotTEL * KGunTEL * DK1a)
+ (KGofTELc * DK1a) ($)

From INPUTS1:
FR1aINS = Fraction of procurement and launch cost paid as insurance premium ($/$)

From PROCURE1:
CTprTEL = Total procurement cost for teleoperators ($)

Launch insurance is computed as a percentage of the procurement and launch costs of the hardware. So total launch insurance premium for teleoperators =

CTinTEL = (CTprTEL + CTlaTEL) * FRlaINS ($)

B.8.6 Mission Preparation and Launch of Automated Equipment (AEQ):

(Subroutine LAofAEQ)

The automated equipment hardware consists of initial equipment and spares, which must be mission-prepped and launched. Automated equipment is high-tech hardware. The model assumes that AEQ requires no consumables.

From PROCURE1:
NtotAEQ = total requirement for automated equipment (aeqs)

From INPUTS1:
KGunAEQ = Mass of one unit of automated equipment (kg/aeq)
DKmpH = Dollar/kg cost to mission-prep. high-tech hardware ($/kg)
DKla = Dollars to launch one kilogram on STS ($/kg)

The mission preparation cost for high-tech hardware is the product of the mass to be launched and a dollar/kilogram factor. For the AEQ hardware, this cost is

(NtotAEQ aeqs) * (KGunAEQ kg/aeq) * (DKmpH $/kg)
= NtotAEQ * KGunAEQ * DKmpH ($)

The launch-only cost for the hardware is the product of the mass to be launched and DKla. This cost is

(NtotAEQ aeqs) * (KGunAEQ kg/aeq) * (DKla $/kg)
= NtotAEQ * KGunAEQ * DKla ($)

Then the total cost to launch the automated equipment (including spares) =

CTlaAEQ = (NtotAEQ * KGunAEQ * DKmpH)
+ (NtotAEQ * KGunAEQ * DKla) ($)

From INPUTS1:
FRlaINS = Fraction of procurement and launch cost paid as insurance premium ($/$)

From PROCURE1:
CTprAEQ = Total procurement cost for automated equipment ($)

Launch insurance is computed as a percentage of the procurement and launch costs of the hardware. So total launch insurance premium for automated equipment =

CTinAEQ = (CTprAEQ + CTlaAEQ) * FRlaINS ($)

B.8.7 Mission Preparation and Launch of Construction Support Equipment (CSE):

(Subroutine LAofCSE)

As described in the Procurement section above, the CSE is low-tech hardware (e.g. construction jigs), requiring no spares or consumables. However, it must be mission-prepped and launched.

From INPUTS1:

NtotCSE = Total number of units of construction support equipment (units)

KGunCSE = Mass of one unit of construction support equipment (kg/unit)

DKmpL = Dollar/kg cost to mission-prep. low-tech hardware ($/kg)

DKla = Dollars to launch one kilogram on STS ($/kg)

The mission preparation cost for low-tech hardware is the product of the mass to be launched and a dollar/kilogram factor. For the CSE hardware, this cost is

(NtotCSE units) * (KGunCSE kg/unit) * (DKmpL $/kg)
= NtotCSE * KGunCSE * DKmpL ($)

The launch-only cost for the hardware is the product of the mass to be launched and DKla. This cost is

(NtotCSE units) * (KGunCSE kg/unit) * (DKla $/kg)
= NtotCSE * KGunCSE * DKla ($)

Then the total cost to launch the construction support equipment =

CTlaCSE = (NtotCSE * KGunCSE * DKmpL)
+ (NtotCSE * KGunCSE * DKla) ($)

From INPUTS1:

FRlaINS = Fraction of procurement and launch cost paid as insurance premium ($/$)

From PROCURE1:

CTprCSE = Total procurement cost for construction support equipment ($)

Launch insurance is computed as a percentage of the procurement and launch costs of the hardware. So total launch insurance premium for construction support equipment =

CTinCSE = (CTprCSE + CTlaCSE) * FRlaINS ($)

B.8.8 Launch of Construction Site Propellant (CSP):

(Subroutine LAofCSP)

This is the cost of launching the construction site propellant, including its tankage. The mission preparation costs are very low and therefore not modeled. The launch cost is the product of the mass to be launched and a dollar/kilogram factor.

From PROCURE1:
KGofCSP = Total requirement for construction site propellant (kg)

From INPUTS1:
DK1a = Dollars to launch one kilogram on STS ($/kg)
FR1aINS = Fraction of procurement and launch cost paid as insurance premium ($/$)

Total cost to launch the construction site propellant =
CT1aCSP = (KGofCSP kg) * (DK1a $/kg)
= KGofCSP * DK1a ($)

Since the small procurement cost of the construction site propellant is not modeled, its launch insurance is computed as a percentage of the launch cost only. So total launch insurance premium for construction site propellant =
CTinCSP = CT1aCSP * FR1aINS ($)

B.8.9 Launch Insurance (INS):

(Subroutine LAofINS)

The insurance premiums from the various items to be launched are summed into an insurance total.

From LAofSST:
CTinSST = Insurance premium for space structure ($)

From LAofHAB:
CTinHAB = Insurance premium for habitation modules ($)

From LAofHSE:
CTinHSE = Insurance premium for human support equipment ($)

From LAofTEL:
CTinTEL = Insurance premium for teleoperators ($)

From LAofAEQ:
CTinAEQ = Insurance premium for automated equipment ($)

From LAofCSE:
CTinCSE = Insurance premium for construction support equipment ($)

From LAofCSP:
CTinCSP = Insurance premium for construction site propellant ($)

The total cost of insurance for the program =

$$\begin{aligned} \mathrm{CTlaINS} = {} & \mathrm{CTinSST} + \mathrm{CTinHAB} + \mathrm{CTinHSE} \\ & + \mathrm{CTinTEL} + \mathrm{CTinAEQ} + \mathrm{CTinCSE} + \mathrm{CTinCSP} \qquad (\$) \end{aligned}$$

B.8.10 Calculation of Overall Launch Costs (CTla):

(Subroutine LAUNCH1 cont.)

$$\begin{aligned} \mathrm{CTla} = {} & \mathrm{CTlaSST} \\ & + \mathrm{CTlaHUM} + \mathrm{CTlaHAB} + \mathrm{CTlaHSE} \\ & + \mathrm{CTlaTEL} + \mathrm{CTlaAEQ} \\ & + \mathrm{CTlaCSE} + \mathrm{CTlaCSP} \\ & + \mathrm{CTlaINS} \end{aligned}$$

(CTla is a global variable)

(End of LAUNCH1)

B.9 Calculation of Ground Support Costs:

(Subroutine GRNDSUP1)

The following ground support cost elements are reprinted from Table B.1:

Ground Support:
- Ground support crew training
- Salary and overhead for ground support crew
- User fees for tracking and data relay

The ground support costs consist of training and salary costs for the ground support crew (GSC), and costs of communications between the ground and the in-orbit worksite. The ground support crew consists of a general monitoring crew for the worksite, a monitoring crew for the humans in space, a control crew for the teleoperators, and a monitoring crew for the automated equipment.

B.9.1 Ground Support Crew Training (GSCt):

(Subroutine GSofGSCt)

For simplicity, the cost model assumes that the members of the ground support crew are chosen early in the program, and that they stay with the program until completion. Therefore the training cost is calculated by multiplying the initial number of ground support crewpeople (gsc) by a dollar/person factor.

As described in the Procurement section above, the cost model calculates the number of general monitoring stations (GMS), human monitoring stations (HMS), teleoperator control stations (TCS), and automated equipment monitoring stations (AMS) required for ground support of the worksite. The model assumes that each type of ground station requires a specific number of ground support crewpeople to operate it.

However, the model also assumes that ground support crewpeople work 8 hours/day. Therefore some monitoring stations will require several shifts of people each day, and all of these people must be trained.

From PRDTIME1:
NtotHUMi = Initial number of humans at the worksite (pers)
NtotTELi = Initial number of teleoperators at the worksite (tels)
NtotAEQi = Initial number of units of automated equipment at the worksite (aeqs)

From INPUTS1:
NgmsGEN = Number of general monitoring stations for general worksite support (gms)
NgscGMS = Number of ground support crewpeople to operate one general monitoring station (gsc/gms)
NhmsHUM = Number of worksite monitoring stations to support one human in space (hms/pers)
HDhum = Hours/day worked by humans in space (hrs/day)
NgscHMS = Number of ground support crewpeople to operate one human monitoring station (gsc/hms)
NtcsTEL = Number of teleoperator control stations to support one teleoperator in space (tcs/tel)
HDtel = Hours/day worked by teleoperators (hrs/day)
NgscTCS = Number of ground support crewpeople to operate one teleoperator control station (gsc/tcs)
NamsAEQ = Number of auto. equipment monitoring stations to support one unit of automated equipment in space (ams/aeq)
HDaeq = Hours/day worked by automated equipment (hrs/day)
NgscAMS = Number of ground support crewpeople to operate one auto. equipment monitoring station (gsc/ams)
DpGSCt = Dollars to train one ground support crewperson ($/gsc)

To calculate the number of GSC people to be trained, the cost model considers the first day of construction, when all tasks are in progress. On that day, the general monitoring stations operate for

(NgmsGEN gms) * (24 hours/day) * (1 day)   (gms-hrs)

and therefore require

(NgmsGEN gms) * (24 hrs) * (NgscGMS gsc/gms)   (gsc-hrs)

gsc-hours of ground support crew time. Since there are 8 hours per shift, the number of shifts required for general monitoring is   NgmsGEN * (24 hrs) * NgscGMS / (8 hrs/shift)

= NgmsGEN * NgscGMS * (24/8 shifts)   (gsc-shifts)

which is the number of people to be trained for general worksite support.

Similarly, for the support of humans in space, the number of HMS required is

(NtotHUMi pers) * (NhmsHUM hms/pers)   (hms)

and on the first day of construction these operate for

NtotHUMi * NhmsHUM * (HDhum hrs/day) * (1 day)   (hms-hrs)

So they require

NtotHUMi * NhmsHUM * HDhum * (NgscHMS gsc/hms)   (gsc-hrs)

gsc-hours of ground support crew time. At 8 hours/shift, the number of shifts required is

NtotHUMi * NhmsHUM * HDhum * NgscHMS / (8 hrs/shift)

= NtotHUMi * NhmsHUM * HDhum * NgscHMS / 8   (gsc-shifts)

which is the number of people to be trained for human support.

Also similar is the support for teleoperators in space, for which the number of TCS required is

(NtotTELi tels) * (NtcsTEL tcs/tel)   (tcs)

and on the first day of construction these operate for

NtotTELi * NtcsTEL * (HDtel hrs/day) * (1 day)   (tcs-hrs)

So they require

NtotTELi * NtcsTEL * HDtel * (NgscTCS gsc/tcs)   (gsc-hrs)

gsc-hours of ground support crew time. At 8 hours/shift, the number of shifts required is

NtotTELi * NtcsTEL * HDtel * NgscTCS / (8 hrs/shift)

= NtotTELi * NtcsTEL * HDtel * NgscTCS / 8   (gsc-shifts)

which is the number of people to be trained for ground support of teleoperators in space.

Finally, for the support for automated equipment in space, the number of AMS required is

(NtotAEQi aeqs) * (NamsAEQ ams/aeq)   (ams)

and on the first day of construction these operate for

NtotAEQi * NamsAEQ * (HDaeq hrs/day) * (1 day)   (ams-hrs)

So they require
NtotAEQi * NamsAEQ * HDaeq * (NgscAMS gsc/ams) (gsc-hrs)
gsc-hours of ground support crew time. At 8 hours/shift, the number of shifts required is
NtotAEQi * NamsAEQ * HDaeq * NgscAMS / (8 hrs/shift)
= NtotAEQi * NamsAEQ * HDaeq * NgscAMS / 8 (gsc-shifts)
which is the number of people to be trained for ground support of automated equipment in space.

Therefore the total number of ground support crewpeople to be trained =
NtotGSCt = (NgmsGEN * NgscGMS * (24/8))
+ (NtotHUMi * NhmsHUM * HDhum * NgscHMS / 8)
+ (NtotTELi * NtcsTEL * HDtel * NgscTCS / 8)
+ (NtotAEQi * NamsAEQ * HDaeq * NgscAMS / 8)
(gsc)

and the total cost of training for ground support crewpeople =
CTgsGSCt = (NtotGSCt gscs) * (DpGSCt $/gsc)
= NtotGSCt * DpGSCt ($)

B.9.2 Salary and Overhead for Ground Support Crew (GSCb):

(Subroutine GSofGSCb)

Salary and overhead expenses are calculated as the product of the total number of gsc-hours put in by the ground support crew and a dollar/gsc-hour factor. These gsc-hours occur in two phases, during training and during space operations.

The model assumes that the entire ground crew goes through the same period of training as the space crew (NdHUMt days), and that they draw regular salaries during that period. During space activities, for each type of ground support station (GMS, HMS, TCS, AMS), the total number of gsc-hours is proportional to that station's usage.

From GSofGSCt:
NtotGSCt = Total number of ground support crewpeople to be trained (gsc)

From PROCURE1:
NtotGMSH = Total general monitoring station usage for the program (gms-hrs)
NtotHMSH = Total human monitoring station usage for the program (hms-hrs)
NtotTCSH = Total teleoperator control station usage for the program (tcs-hrs)
NtotAMSH = Total automated equipment monitoring station usage for the program (ams-hrs)

From INPUTS1:
NdHUMt = Number of days of training for humans in space crews (days)
NgscGMS = Number of ground support crewpeople to operate one general monitoring station (gsc/gms)
NgscHMS = Number of ground support crewpeople to operate one human monitoring station (gsc/hms)
NgscTCS = Number of ground support crewpeople to operate one teleoperator control station (gsc/tcs)
NgscAMS = Number of ground support crewpeople to operate one auto. equipment monitoring station (gsc/ams)
DphGSC = Salary and overhead for one ground support crewperson for one hour ($/gsc-hr)

The number of ground support crewpeople drawing salary during training =
NtotGSCt (gsc)

These people draw salary 8 hours/day, for NdHUMt days. So the number of gsc-hrs to be paid during training =
(NtotGSCt gsc) * (NdHUMt days) * (8 hrs/day) (gsc-hrs)

As described in the section on Procurement of ground support equipment, the cost model has calculated NtotGMSH, the total general monitoring station usage for the program. Since each GMS requires NgscGMS ground support crewpeople during its operation, the total number of gsc-hours for GMS operation is
(NtotGMSH gms-hrs) * (NgscGMS gsc/gms) (gsc-hrs)

Similarly, the number of gsc-hours for HMS operation is
(NtotHMSH hms-hrs) * (NgscHMS gsc/hms) (gsc-hrs)

Similarly, the number of gsc-hours for TCS operation is
(NtotTCSH tcs-hrs) * (NgscTCS gsc/tcs) (gsc-hrs)

Similarly, the number of gsc-hours for AMS operation is
(NtotAMSH ams-hrs) * (NgscAMS gsc/ams) (gsc-hrs)

So the total requirement for gsc-hours during space activities =
NtotGH = (NtotGMSH * NgscGMS)
+ (NtotHMSH * NgscHMS)
+ (NtotTCSH * NgscTCS)
+ (NtotAMSH * NgscAMS) (gsc-hrs)

So the total number of gsc-hrs to be paid =
NtotGHb = (NtotGSCt * NdHUMt * 8) + NtotGH (gsc-hrs)

And the total cost of salary and overhead for the ground support crew =
CTgsGSCb = (NtotGHb gsc-hrs) * (DphGSC $/gsc-hr)
= NtotGHb * DphGSC ($)

B.9.3 User Fees for Tracking and Data Relay (TDR):

(Subroutine GSofTDR)

These are the costs for communicating between the worksite and the ground, via relay satellites (e.g. TDRSS). The model assumes that two types of channel are available: low-capacity and high-capacity.

The model assumes that three (3) low-capacity channels are used for the entire duration of the program for general support of the worksite. These channels operate around the clock.

Low-capacity channels are also used for communication to humans and automated equipment. Specifically, the model assumes that each human and each unit of automated equipment at the worksite requires one low-capacity channel. To handle its multiple video views, each teleoperator at the worksite requires one high-capacity channel.

The cost model computes the costs of these channels by multiplying the number of channel-hours required by dollar/ channel-hour factors for low-capacity and high-capacity channels.

To compute the numbers of channel-hours needed, the cost model multiplies the numbers of low-capacity channels (lccs) and high-capacity channels (hccs) required for each task by the total duration of that task, and adds the individual products together.

From PRDTIME1:
Jobtime = Time to complete all space tasks (days)

From PROCURE1:
NtotEH = Total usage of pressure suits for the program (emu-hrs)
NtotAH = Total usage of automated equipment for the program (aeq-hrs)
NtotTH = Total teleoperator usage for the program (tel-hrs)

From INPUTS1:
DlhTDR = Cost of one low-capacity TDR channel for one hour ($/lcc-hr)
DhhTDR = Cost of one high-capacity TDR channel for one hour ($/hcc-hr)

Low-capacity channels (LCC):

The number of low-capacity channel-hrs for general support is
(3 lccs) * (Jobtime days) * (24 hrs/day)
= 72 * Jobtime (lcc-hrs)

For task j, the number of low-capacity channel-hours required is

$$\begin{aligned}
&(\text{Nhum}[j]\ \text{pers}) * (1\ \text{lcc/pers}) \\
&\qquad * (\text{Ttime}[j]\ \text{days}) * (\text{HDhum hrs/day}) \\
&+ (\text{Naeq}[j]\ \text{aeqs}) * (1\ \text{lcc/aeq}) \\
&\qquad * (\text{Ttime}[j]\ \text{days}) * (\text{HDaeq hrs/day}) \\
&= \text{Nhum}[j] * \text{Ttime}[j] * \text{HDhum} \\
&\qquad + \text{Naeq}[j] * \text{Ttime}[j] * \text{HDaeq} \qquad \text{(lcc-hrs)}
\end{aligned}$$

So the total LCC usage for the program =

$$\begin{aligned}
\text{NtotLH} &= (72 * \text{Jobtime}) + \sum_{j=1}^{\text{NTSK}} \text{Nhum}[j] * \text{Ttime}[j] * \text{HDhum} + \text{Naeq}[j] * \text{Ttime}[j] * \text{HDaeq} \\
&= (72 * \text{Jobtime}) + \text{NtotEH} + \text{NtotAH} \qquad \text{(lcc-hrs)}
\end{aligned}$$

And the total cost for low-capacity channels for the program =

$$\begin{aligned}
\text{CTgsLCC} &= (\text{NtotLH lcc-hrs}) * (\text{DlhTDR \$/lcc-hr}) \\
&= \text{NtotLH} * \text{DlhTDR} \qquad (\$)
\end{aligned}$$

High-capacity channels (HCC):

For task j, the number of high-capacity channel-hours required is

$$\begin{aligned}
&(\text{Ntel}[j]\ \text{tels}) * (1\ \text{hcc/tel}) \\
&\qquad * (\text{Ttime}[j]\ \text{days}) * (\text{HDtel hrs/day}) \\
&= \text{Ntel}[j] * \text{Ttime}[j] * \text{HDtel} \qquad \text{(hcc-hrs)}
\end{aligned}$$

So the total HCC usage for the program =

$$\begin{aligned}
\text{NtotHH} &= \sum_{j=1}^{\text{NTSK}} \text{Ntel}[j] * \text{Ttime}[j] * \text{HDtel} \\
&= \text{NtotTH} \qquad \text{(hcc-hrs)}
\end{aligned}$$

And the total cost for high-capacity channels for the program =

$$\begin{aligned}
\text{CTgsHCC} &= (\text{NtotHH hcc-hrs}) * (\text{DhhTDR \$/hcc-hr}) \\
&= \text{NtotHH} * \text{DhhTDR} \qquad (\$)
\end{aligned}$$

Then total user fees for tracking and data relay =

$$\text{CTgsTDR} = \text{CTgsLCC} + \text{CTgsHCC} \qquad (\$)$$

B.9.4 Calculation of Overall Ground Support Costs (CTgs):

(Subroutine GRNDSUP1 cont.)

```
CTgs = CTgsGSCt
       + CTgsGSCb
       + CTgsTDR
```

(CTgs is a global variable)

(End of GRNDSUP1)

APPENDIX C:

RADIOMETER MODEL AND CONSTRUCTION TASKS

This appendix describes the Microwave Radiometer to be constructed in this study's case example. Sections C.1 and C.2 present the preliminary design and a physical model of the Radiometer. From this design, section C.3 defines and describes ten construction tasks.

Four of these construction tasks are selected as an illustrative case, which is used in most of the numerical examples in Chapter 4 (the full ten tasks are used in a more complex example at the end of that chapter). Finally, section C.4 estimates the productivities of the human-machine options for the four tasks.

C.1 Description of Microwave Radiometer:

C.1.1 General Description:

The Microwave Radiometer of interest to this study was defined by the NASA Langley Research Center in 1981, as "a potential future space mission requiring large space structures technology" (Ref. C-1). The purpose of the Microwave Radiometer is to map the earth's soil moisture content from space, to aid in global crop forecasting. The radiometer is passive: it does not broadcast any microwaves, but only receives microwaves from the earth's surface.

Preliminary designs of several structural concepts for the radiometer were produced by the NASA Langley design team. The chosen baseline configuration (described in Refs. C-1 and C-2) is sketched in Figures C.1 and C.2. The radiometer consists of four main components:

1) a 725-m diameter collecting antenna;
2) a stiffener ring surrounding the antenna;
3) a linear array of 600 feed horn assemblies;
4) two 580-meter support columns extending from the stiffener ring to the feed horn array.

Several structural versions of the radiometer were defined by the NASA design team, including deployable and erectable concepts with varying structural member lengths. For a case example in this study, the móst appropriate version is the erectable structure design, using 18-meter-long structural elements.

Table C.1 presents a mass breakdown of that version of the Microwave Radiometer. The following sections describe the radiometer components in more detail.

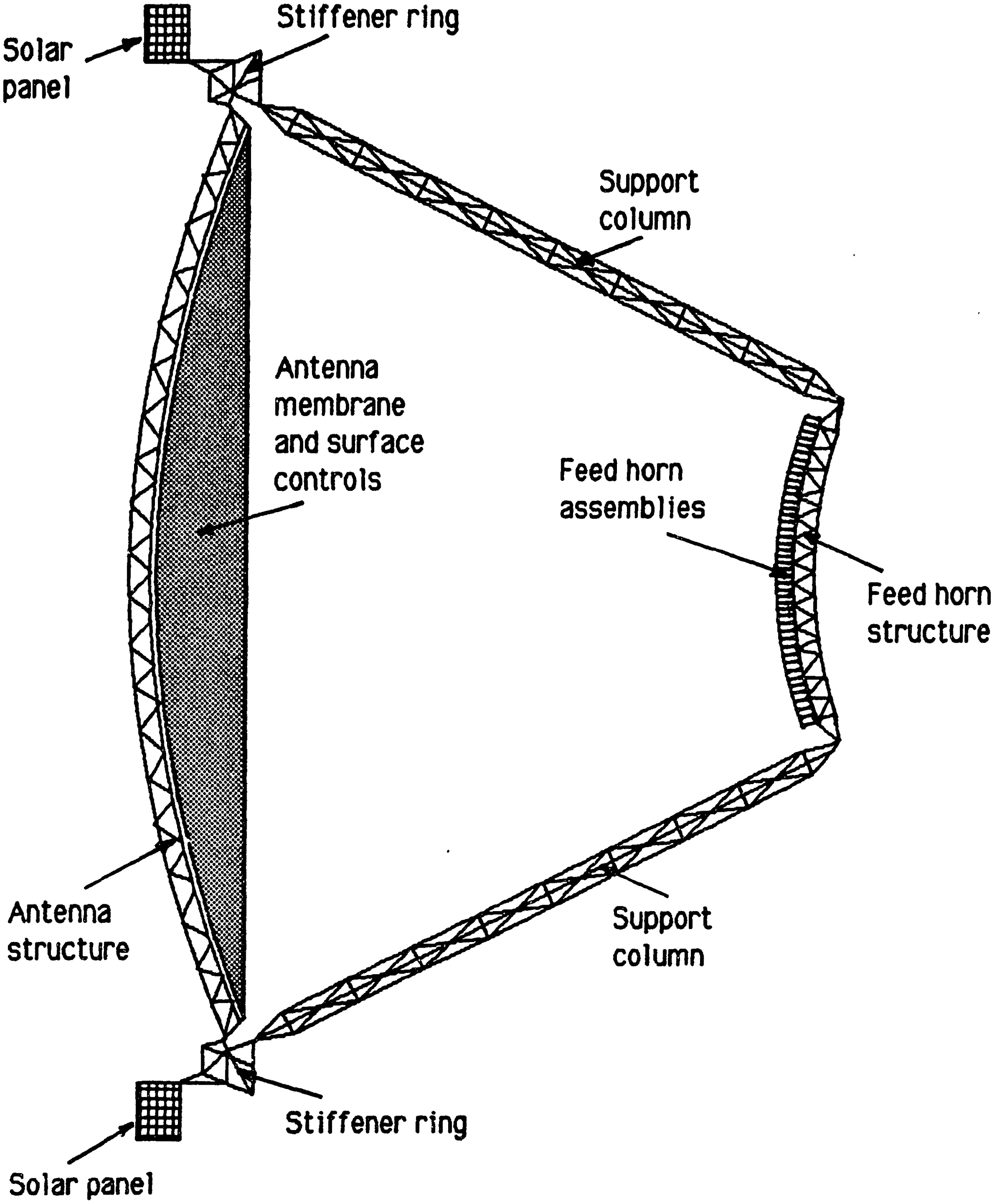


FIGURE C.1: SIDE CUTAWAY VIEW OF MICROWAVE RADIOMETER

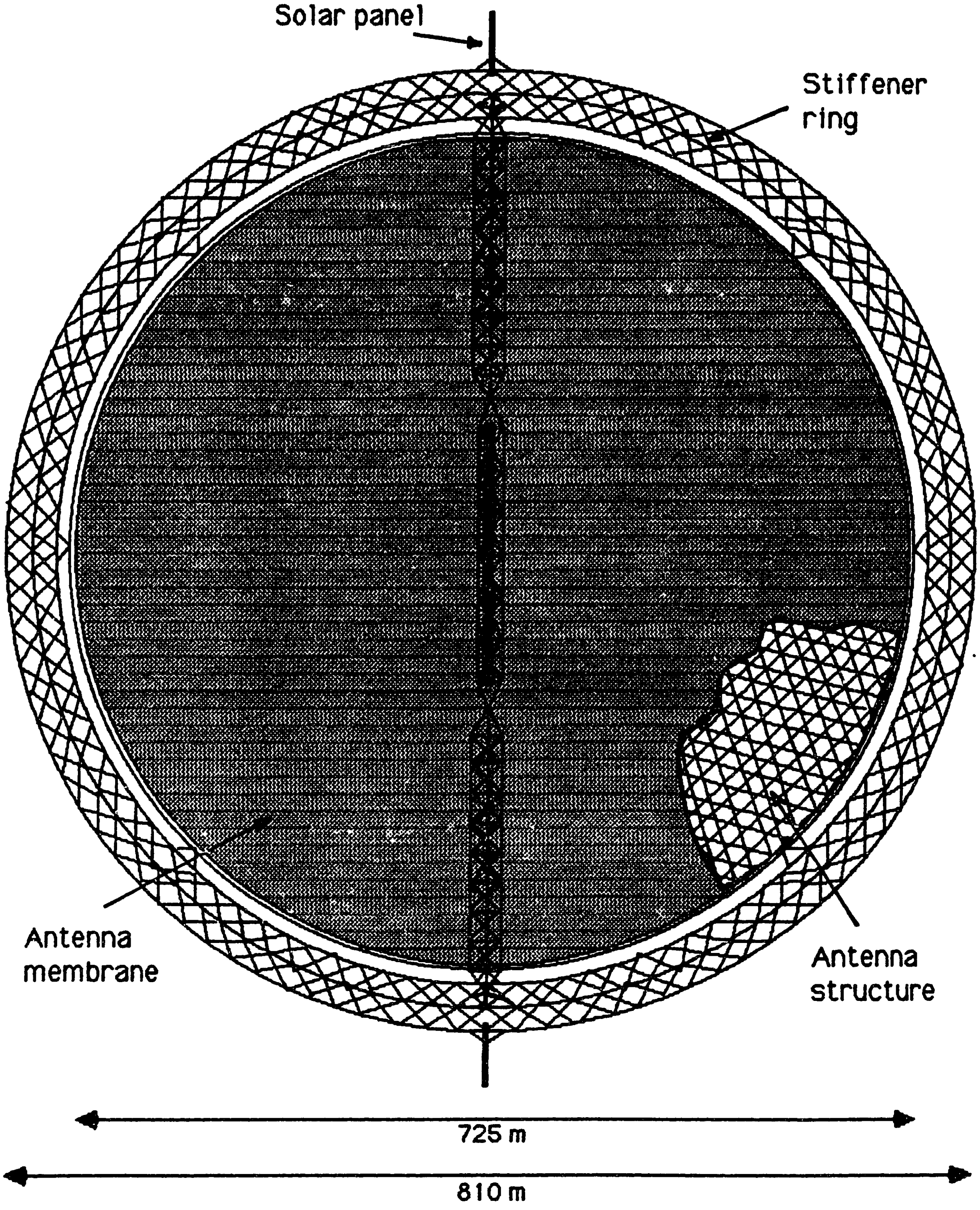


FIGURE C.2: FRONT VIEW OF MICROWAVE RADIOMETER

TABLE C.1: MASS BREAKDOWN FOR BASELINE MICROWAVE RADIOMETER
(DOUBLE-LAYER TETRAHEDRAL TRUSS, ERECTABLE)

	Mass(kg)
Antenna structure	45,813
Antenna membrane	35,000
Membrane surface controls (a)	30,000
Antenna actuators	110
Stiffener ring structure	9,990
Solar panels (a)	450
Attitude control momentum wheels (a)	2,000
Feed horn structure	708
Feed horns	13,800
Electronics modules (a)	600
Support columns	1,480
Total microwave radiometer	139,951 (b)

Notes:
Table extracted from Table III in Ref. C-2.
(a) Estimated values.
(b) For a Shuttle payload capacity of 27,281 kg (to 370 km altitude) this radiometer would require 5.13 Shuttle flights for launch.

C.1.2 Collecting Antenna:

The antenna consists of a radio-frequency membrane attached to a tetrahedral truss structure. It is a 725-meter-diameter circular section of a sphere.

The antenna structure is made up of 13,590 structural beams, joined at 3,020 node fittings into a double-layer tetrahedral truss. Each structural beam is roughly 18.0 meters long, with some length variations to create the curvature of the truss. Each beam masses 3.32 kg, and each node fitting masses 0.23 kg, for a total antenna structure mass of 45,813 kg.

The antenna membrane is a mesh which reflects the collected microwaves into the array of feed horn assemblies. It is a metallized membrane (e.g. a gold-plated molybdenum mesh) with a total area of 423,700 square meters and a total mass of 35,000 kg.

To achieve the desired <1km resolution, 70-dB gain, and 90% efficiency, the membrane's surface must be positioned within 1/50 of the wavelength being measured. This results in a surface-accuracy-to-diameter ratio on the order of 10^{-7}. To achieve this, the NASA study proposed a combination of mechanical and electrostatic systems, but these are still in the research stage and detailed designs are not yet available. The NASA design team estimated a total mass of 30,000 kg for the membrane surface controls.

The tetrahedral structure is attached at several points to the stiffener ring through control actuators, which allow the entire antenna to move within the ring. One drawing in Ref. C-2 suggests that there are eight of these actuators. Their total mass is only 110 kg.

C.1.3 Stiffener Ring:

The stiffener ring surrounds the collecting antenna, and serves as the principal structure to which the radiometer sections are attached. Details of this structure are not reported in Ref. C-2, so it is assumed that it is made up of 18-meter truss beams and node fittings, in the same ratios as in the antenna's tetrahedral truss. Using NASA Langley's estimate of 9,990 kg for the whole ring, this works out to 2,964 18-meter beams and 658 node fittings.

Also attached to the stiffener ring are solar panels (with total area 200 square meters and mass 450 kg).

The stiffener ring also supports the attitude control momentum wheels for the radiometer. The NASA design study proposes a novel "dual-momentum vector control system", using two counter-rotating wheels as large as the stiffener ring. Detailed designs are not yet available, but the total mass is estimated at 2000 kg.

C.1.4 Radiometer Feed Horns:

The linear array of radiometer feed horns consists of a truss structure and the feed horn assemblies attached to the structure. The support structure is a 200-meter-long tetrahedral truss, made of 134 18-meter truss beams and 50 node fittings, with a total mass of 421 kg. When compared to the masses for antenna truss components, this value is a little low, indicating that the feed horn structure elements are of slightly different construction.

Also included in the feed horn structure are attachment fittings for the feed horn assemblies. There are 600 such supports, with a total mass of 287 kg.

Each of the 600 feed horn assemblies consists of the feed horn itself and an electronics module. Each feed horn masses 23 kg, for a total feed horn mass of 13,800 kg. Each electronics module masses 1 kg, for a total mass of 600 kg.

C.1.5 Support Columns:

Two 580-meter-long support columns extend between the stiffener ring and the feed horn array. In the baseline design, these columns are tension-stabilized. Each column consists of a central graphite/epoxy compression tube (645 kg), sixteen aluminum outrigger tubes (85 kg), and a set of Kevlar tension cables (10 kg), for a total column mass of 740 kg.

C.2 Physical Model of Radiometer:

C.2.1 General Discussion of Physical Model:

The physical model of the radiometer consists of the number of units and unit masses of various types of hardware, their packing densities, and the procurement learning curve parameter. As described in Appendix B, these inputs are used in the line-item cost model to calculate the R&D, procurement, and launch costs for the space structure to be assembled.

To keep the case example simple, it is assumed that the physical description of the radiometer is not involved in the tradeoffs between humans and machines in space. In other words, the space structure's R&D, procurement, and launch costs are the same, whether humans, teleoperators, or automated equipment are used in its construction. The result of this assumption is that these costs of the physical space structure appear as constants in the overall line-item cost equation.

How far does this assumption bend reality? The R&D, procurement, and launch costs depend on the physical design of the hardware. That physical design might depend on the expected means of construction. For example, a structure to be assembled by teleoperators might have different joints than one assembled by humans.

However, it is expected that the design of a space structure will be driven more by its intended use than by its construction. It is the construction options which should be adapted to the structure, not the other way around. For example, well-designed teleoperators should be able to assemble the same structures as humans (Ref. C-3).

As for automated equipment, it is unrealistic to expect machines to do tasks in the same fashion as humans; but the design of construction machines should not force major changes in the hardware to be built. It is expected that minor adaptations in the space structure's hardware would be made, but that these would not lead to major differences in structure costs.

The assumption described above places the study's emphasis on the construction tasks (described later) rather than on the radiometer design. Therefore a simple physical model of this structure is appropriate. The radiometer is modeled as two types of hardware: low-technology and high-technology.

C.2.2 Low-technology Hardware:

The low-technology hardware consists of all the trusswork elements of the radiometer: antenna structure, stiffener ring, feed horn structure, and support columns. These elements are summarized in Table C.2.

In the NASA Langley design study, the baseline beams are graphite/epoxy tapered tubes, which are brought to the worksite as 9-meter nested cones. For this study, however, it is more interesting to assume 18-meter truss beams as the basic structural element. This is because truss beams are likely candidates for onsite manufacture by automated equipment, such as the Grumman and General Dynamics beam-builder designs. Truss beams were therefore assumed for the case example, but the Langley estimates for numbers and masses of beams were kept unchanged.

To define a basic unit of low-technology hardware, a hypothetical trusswork "cell" was selected, with 1 node fitting and 4.5 truss beams (this is their ratio in the antenna's tetrahedral truss). The unit mass is therefore 15.17 kg.

To calculate the total number of units of low-technology hardware, the total trusswork mass 57,704 kg was divided by the unit mass, to yield 3,804 units. In effect, this models all trusswork, including the support columns, as beam-and-node trusses. This is acceptable in this simplified physical model of the radiometer. (As discussed later, however, the construction task for the support columns is kept separate.)

TABLE C.2: TRUSSWORK ELEMENTS IN RADIOMETER

Basic structural elements:

	Length(m)	Mass(kg)
Truss beam	18.0	3.32
Node fitting		0.23

Radiometer trusswork elements:

	Number	Mass(kg)
Antenna structure:		
Truss beams	13,590	45,118
Node fittings	3,020	695
Stiffener ring:		
Truss beams	2,964	9,839
Node fittings	658	151
Feed horn structure:		
Truss beams	134	409 (a)
Node fittings	50	12
Support columns:		
Center shafts	2	1,290
Outriggers	32	170
Kevlar cables		20

Total trusswork mass		57,704

Notes:
(a) These beams are slightly lighter than basic beams.

C.2.3 High-technology Hardware:

The high-technology hardware consists of the antenna membrane and surface controls, the attitude control momentum wheels, and the feed horn assemblies. These components are summarized in Table C.3.

Specific designs for the antenna membrane and its surface controls are not yet available. As will be explained below, the case example can be kept simpler by assuming that the membrane and the membrane surface controls each consist of 600 units of hardware. In other words, each section of membrane and

TABLE C.3: HIGH-TECHNOLOGY HARDWARE IN RADIOMETER

Component:	Mass(kg)
Antenna membrane	35,000
Membrane surface controls	30,000
Attitude control momentum wheels	2,000
Feed horn attachment fittings	287
Feed horns	13,800
Electronics modules	600
Total high-tech hardware	81,687

associated surface controls covers 1/600 of the total antenna area, or roughly 706 square meters (equivalent to a 30-meter-diameter circle). So each membrane unit masses 58.33 kg, and each surface control unit masses 50 kg.

For simplicity in the case example, the antenna actuators are not modeled. They are low in number (8) and mass (110 kg). Similarly, the solar panels (200 square meters and 450 kg) are also omitted from the model.

Detailed designs for the attitude control momentum wheels are not yet available. It is therefore assumed that these wheels consist of 600 units of hardware, with mass 3.33 kg each.

The feed horn assemblies consist of 600 feed horns (23 kg each), and 600 electronics modules (1 kg each). In addition, the 600 feed horn attachment fittings (0.48 kg each) are included in the high-technology hardware, because they are precision components.

The assumptions of 600 units of hardware for antenna membrane, surface controls, and momentum wheels simplify the case example because they allow the definition of a single "unit" of high-technology hardware, which consists of one unit of antenna membrane, one unit of antenna surface controls, one unit of momentum wheel, and one feed horn assembly (with attachment fitting). The components of the high-tech hardware unit are summarized in Table C.4.

This lumping of different kinds of hardware into one high-tech unit is allowable, because the line-item cost model will produce the same results from the lumped unit as it would from several separate units.

TABLE C.4: COMPONENTS OF ONE HIGH-TECH HARDWARE UNIT

One unit of high-technology hardware:	Mass(kg)
Antenna membrane	58.33
Membrane surface controls	50
Attitude control momentum wheels	3.33
Feed horn attachment fittings	0.48
Feed horns	23
Electronics modules	1

High-tech hardware unit mass	136.15

C.2.4 Summary of Physical Model:

The physical model of the radiometer is presented in Table C.5. Besides the number of units and unit masses of hardware, it includes packing densities for launch, and a learning curve parameter for procurement. As shown by note (a) of the Table, the physical model includes almost all of the actual radiometer design.

TABLE C.5: SUMMARY OF PHYSICAL MODEL OF RADIOMETER

	Number of units	Unit mass(kg)	Total mass(kg)
Low-technology hardware	3,804	15.17	57,704
High-technology hardware	600	136.15	81,687

Total mass of physical model			139,391 (a)

Packing density (kg/cubic meter):

Low-tech hardware:	200
High-tech hardware:	300

Learning curve parameter: 80%

Note:
(a) Unmodeled components:

Antenna actuators	110
Solar panels	450

Total radiometer mass	139,951

The packing density of this erectable structure is expected to be higher than the critical value of 92.5 kg/cubic meter. Values of 200 kg/cubic meter for low-technology hardware and 300 kg/cubic meter for high-technology hardware are assumed. As long as the densities are above 92.5 kg/cubic meter, the exact values are not important.

The learning curve parameter (defined in section B.7.2) is set at 80%, a typical value for production of large lots of hardware (e.g. 600 or 3,804 units).

C.3 Radiometer Construction Tasks:

The definition of radiometer construction tasks is a three-step process:

1) Definition of specific construction tasks;
2) Calculation of the number of repetitions of each task;
3) Estimation of the productivities of humans, teleoperators, and automated equipment in each task.

This section deals with steps (1) and (2). Step (3) is discussed in section C.4.

C.3.1 Ten-Task Construction Model:

Based on the descriptions in Refs. C-1 and C-2, ten construction tasks are defined to complete the Microwave Radiometer. Tables C.2, C.4, and C.5 are used to compute the number of repetitions of each task. These are presented in Table C.6, and described below.

Task T1 is the manufacture of graphite/epoxy truss beams from as-delivered stock. This stock is packaged for high density, so it could be in the form of raw material (e.g. rolls of pre-preg) or components of beams (e.g. longerons and crosspieces). This task produces the standard 18-meter truss beams (with some length variations as needed), and the slightly lighter beams for the feed horn structure. Each beam produced is a repetition of the task.

Task T2 is the assembly of tetrahedral trusses for the antenna, the stiffener ring, and the feed horn structure. These trusses are made up of truss beams and node fittings, and one repetition of the task is defined as the attachment of beams to one node fitting. Therefore there are as many repetitions as node fittings, as listed in Table C.2.

TABLE C.6: CONSTRUCTION TASKS FOR THE MICROWAVE RADIOMETER

Task:	Number of repetitions
T1: Manufacture of truss beams:	
For antenna structure	13,590
For stiffener ring	2,964
For feed horn structure	134

	16,688
T2: Assembly of tetrahedral truss:	
For antenna structure	3,020
For stiffener ring	658
For feed horn structure	50

	3,728
T3: Manufacture of support columns	2
T4: Assembly of support columns	4
T5: Assembly of membrane and surface controls	600
T6: Checkout of membrane surface controls	600
T7: Assembly of feed horns	600
T8: Installation of feed horn assemblies	600
T9: Alignment of feed horn assemblies	600
T10: Assembly of attitude control momentum wheels	600

Task T3 is the manufacture of the two 580-meter-long support columns from as-delivered stock. The manufacture of each column is a repetition of the task.

Task T4 is the attachment of the two support columns to the stiffener ring and to the feed horn structure. This involves four attachment points, therefore four repetitions of the task. (Note: the NASA Langley design includes a set of bracing cables to fix the position of these columns; installing these cables is part of this task.)

Task T5 is the assembly of the antenna membrane and associated surface controls to the antenna truss structure. For the physical model of the radiometer, it is assumed that the antenna membrane consists of 600 units. This assumption is carried over to this task definition. The installation of one such unit is a repetition of the task.

Task T6 is the checkout of each subsection of the antenna membrane after assembly: the proper function of the membrane surface controls must be verified. The checkout of each of the 600 membrane units is a repetition of this task.

Task T7 is the assembly of the radiometer feed horns from as-delivered components. The NASA design team proposes a low-mass graphite/epoxy 10-meter-long feed horn, delivered in stacks of half-horns (split lengthwise). These half-horns must be re-assembled, and their electronics modules and attachment fittings must be added. Since there are 600 feed horns, 600 repetitions of this task are required.

Task T8 is the installation of the feed horn assemblies into the feed horn truss structure. This involves the structural connection of the horns through their attachment fittings, and the electrical connection of power and data cables. The installation of each assembly is a repetition of this task.

Task T9 is the alignment of each feed horn assembly, after its installation. This would involve the use of fiducial sources in known locations on the radiometer, and perhaps the use of completed sections of the antenna membrane. Onsite test and adjustment equipment would also be needed. The alignment of each feed horn is a repetition of the task.

Task T10 is the assembly of the dual-momentum vector control wheels attached to the stiffener ring. For the physical model of the radiometer, it is assumed that these wheels consist of 600 units of hardware. This assumption is carried over to this task definition: the installation of one such unit is a repetition of the task. This task resembles Task T8, the installation of the feed horn assemblies, in that it consists of structural and electrical connections of precision components.

This ten-task construction model is used as a complex case example in Chapter 4 of this study.

C.3.2 Four-Task Construction Model:

For the exploration of the cost equation's behavior, and for the development of the study's methods, a simpler four-task example is more useful. Therefore four tasks are selected from the construction tasks described above; they are presented in Table C.7. These tasks are used in most of the numerical examples in Chapter 4.

TABLE C.7: FOUR-TASK CASE EXAMPLE

Task:		Number of repetitions
1:	Manufacture of truss beams	16,688
2:	Assembly of tetrahedral truss	3,728
3:	Assembly of membrane and surface controls	600
4:	Installation of feed horn assemblies	600

C.4 Human and Machine Productivities:

The case example's human and machine options perform the tasks listed in Table C.7 with various levels of productivity. Estimates of these productivities are based on preliminary task definitions, simulations of construction activities, and on the equipment designs in Appendix D.

C.4.1 General Discussion on Estimation of Productivities:

The first requirement in the estimation of productivities is an appropriate definition of productivity itself. In general, productivity is the ratio of production output to production time by workers. In earth industry, "output" is measured by a variety of quantities, e.g. mass, volume, number of items produced, or value added. However, these scales may not be useful in measuring the output of space activities, and different models of productivity may be required (Ref. C-4).

In space construction, the variety of the tasks complicates the definition of productivity. 'Mass of hardware assembled' and 'operations performed' have been used as measures of output by several studies (e.g. Ref. C-5). However, mass may not be the significant factor in construction time. For example, Ref. C-6 finds that in the simulated assembly of truss structures, the moment of inertia of the beams has a far stronger correlation with work time. This might suggest 'cumulative moment of inertia' as a measure of output for truss assembly.

The use of 'operations' as a measure of output requires the definition of 'operation'. Since construction requires a variety of activities (e.g. beam manufacture, truss assembly, component installation), a different definition of 'operation' may be required for each task.

Despite this drawback, this is the approach used in this study: individual tasks are defined, and each task is made up of "repetitions" (reps). For example, the manufacture of truss beams is a task, and the manufacture of one beam is a "repetition" of the task. One advantage of this measure is that it can be applied to tasks with no material output, e.g. the alignment of an instrument.

Therefore the productivities in this study have the units "reps/pers-hr" for humans, "reps/tel-hr" for teleoperators, and "reps/aeq-hr" for automated equipment, with definitions of "repetition" for each task. To estimate the numerical value of each of these productivities, four techniques are available. They are described below, in order of increasing accuracy.

The first technique is a hypothetical time and motion study for a repetiton of a task. The repetition is broken down into its component actions: for example, the manufacture of truss beams is decomposed into the alignment of beam components and the welding of beam joints.

For each of these actions, prior experience and engineering judgment are used to estimate a range of time, e.g. "each weld will take more than 8 seconds but less than 17 seconds". The most likely ("nominal") value within this range can also be estimated. Then the time ranges for all the actions are combined into a time range for the complete repetition, and this is converted into a productivity range for the task.

This technique is quick and cheap, since it relies mostly on judgment and consultation. It is useful for estimating upper and lower bounds on a task's productivity, but its "nominal" productivity estimates are very preliminary.

The second technique uses earth analogies to estimate space productivities. If a task to be done in space resembles a documented task on earth, the earth productivity can perhaps be adjusted into an estimate for the space task. This technique includes several steps. First, an earth analogy must be found and its appropriateness must be evaluated. For example, if space truss assembly is thought to be analogous to the construction of scaffolding, which type of scaffolding is the closest equivalent, and how close an analogy is it?

Next, the productivity of the earth task must be found. This requires literature search and consultation, and may run into proprietary data barriers. For example, the total work hours used to erect the scaffolding around the Statue of Liberty are not available (Ref. C-7). If the earth productivity is available, then it must be adjusted into a space estimate by evaluating the effect of the different task environment.

The third technique is the use of simulations on earth. For space construction, such simulations include human experiments in parabolic flights and neutral buoyancy (Refs. C-5, C-6, C-8), teleoperator simulations in laboratories (e.g. on air-bearing floors at NASA Marshall) and in neutral buoyancy (Refs. C-9, C-10), and the development of prototypes of automated beam-builders (Refs. C-11, C-12).

Such simulations are relatively recent, and the resulting data bases are still limited, particularly for teleoperators and automated equipment. Their results must also be adjusted to the space environment, since the laboratory and underwater environments cannot duplicate zero-g and vacuum conditions exactly. When available, however, this data is more accurate than the results of the first two techniques.

The fourth technique is the calculation of productivities from actual space hardware and procedures, either from earlier space programs or from new experiments specifically to measure productivities. Relevant earlier programs might include the Skylab sunshade deployment, the recovery of Westar and Palapa, and the repairs of Solar Max and Leasat. It should be noted, however, that these jobs were done without any attempt at high productivity: the priority was on successful completion, not minimum time.

To date, the only space experiment simulating construction has been the Experimental Assembly of Structures in EVA (EASE), an MIT Space Systems Lab/NASA Marshall Space Flight Center experiment on Shuttle flight 61-B (Ref. C-13). At this time, only qualitative results are available, and they validate the earlier neutral buoyancy simulations. It is expected that future space simulations will develop data on construction teleoperators and automated equipment. Based on detailed time and motion studies of these simulations, productivity predictions for a wide range of space tasks will become possible.

The following sections use several of the above-described techniques to estimate productivities for the four construction tasks in this study's case example. The purpose of this example is to illustrate the application of the study techniques, not to decide the best way to build a Radiometer. Therefore precise values for the productivities are not required, and estimates based on available data are sufficient.

C.4.2 Productivities in Manufacture of Truss Beams:

Task 1 is the manufacture of 18-meter-long truss beams from high-density stock (either stacked beam components or rolls of graphite/epoxy material). The beams are modeled as 10-bay tribeams, each consisting of 3 longerons, 33 crossmembers, and 30 diagonals. The ends of the longerons are brought together and attached to 2 endjoints, so there are 68 components in each beam.

These pieces are attached together by ultrasonic welding: 66 welds for the crossmembers, 60 for the diagonals, and 6 for the endjoints, for a total of 132 welds per beam.

Table C.8 presents estimated ranges of times for the alignment of the 68 components and the completion of the 132 welds by humans in EVA. The total times for completion of one beam are converted to productivities (beams/person-hour). Since these values apply only to the actual manufacture, the productivities are then lowered by a 20% time overhead factor, to account for worksite setup, fetching of raw materials, delivery of completed beams, and miscellaneous workbreaks. Each beam's manufacture is one repetition of Task 1. Therefore:

OHhum[1] = Productivity of humans in performing task 1
= a range: low value = 0.9 reps/pers-hr
nominal value = 1.3 reps/pers-hr
high value = 1.8 reps/pers-hr

TABLE C.8: ESTIMATED PRODUCTIVITY OF HUMANS IN THE MANUFACTURE OF TRUSSWORK BEAMS

		Slow	Nominal	Fast
Alignment times (secs):				
	Each piece	15	10	7
	68 pieces	1020	680	476
Welding times (secs):				
	Each weld	17	12	8
	132 welds	2244	1584	1056
		------	-----	-----
Total times for one beam (secs):		3264	2264	1532
	(mins):	54.4	37.7	25.5
		Low	Nominal	High
Human productivity (beams/pers-hr):		1.1	1.6	2.3
With 20% time overhead (reps/pers-hr):		0.9	1.3	1.8

The productivity of the teleoperator is conceptually simpler to estimate by a comparison to human productivity. At the upper end of the range, since the teleoperator is controlled by a human across a time delay, it is unlikely that its productivity will be higher than that of an onsite human. Just to achieve a productivity equal to the human's, the teleoperator requires some methods to mitigate the time delay, such as predictive displays for the operator and trouble-avoidance sensors and logic aboard the teleoperator.

For a productivity higher than an onsite human's, a teleoperator would require certain abilities beyond the human's, such as some preprogrammed or learned functions to be performed faster than humanly feasible. This level of independent control (including trouble avoidance) would require a significant investment in machine intelligence, beyond the teleoperator R&D cost modeled in this example. Therefore the upper limit of teleoperator productivity is estimated as equal to the EVA human's.

At the low end of the range, the teleoperator operates under a move-and-wait strategy. Ref. C-14 reviews some experimental evidence for a teleoperator productivity roughly one-third of the human's, for master-slave operations across a 1-to-1.5-second time delay. Therefore the low value for teleoperator productivity is set at one-third of the human's.

The manufacture of truss beams involves well-defined operations in a controlled situation. Therefore the teleoperator's nominal productivity is estimated at (0.8) of the human's. Because force-feedback cannot be usefully implemented across a time delay, the onsite human has a slight advantage in this activity. Therefore:

```
OHtel[1] = Teleoperator productivity in performing task 1
         = a range: low value = 0.3 reps/tel-hr
                    nominal value = 1.0 reps/tel-hr
                    high value = 1.8 reps/tel-hr
```

Note that the low and high values are extreme values. If the actual human productivity were (1.3 reps/pers-hr), then the expected range for the teleoperator would be (0.4 reps/tel-hr) to (1.3 reps/tel-hr).

For the automated equipment, the design speed of the beam-builder is 1.1 meters of truss beam per minute (Ref. C-12). This corresponds to 16.4 minutes per beam, or 3.7 beams/aeq-hr. Since this is a preliminary estimate, a range of ±30% is estimated. Unlike the human and teleoperator, the beam-builder incurs no time overhead. Therefore:

OHaeq[1] = Productivity of automated equipment in task 1
= a range: low value = 2.6 reps/aeq-hr
nominal value = 3.7 reps/aeq-hr
high value = 4.8 reps/aeq-hr

C.4.3 Productivities in Assembly of Tetrahedral Truss:

Task 2 is the assembly of tetrahedral truss for the radiometer collecting antenna, stiffener ring, and feed horn structure. Each repetition of this task is the assembly of one trusswork "cell" consisting of 1 node fitting and 4.5 beams (requiring 9 joint connections).

The performance of this task by humans has been simulated in underwater experiments by the MIT Space Systems Laboratory (Ref. C-8, Table 2, p. 54, and Ref. C-9). Human productivities were repeatedly measured for the neutral buoyancy assembly of a 36-member tetrahedral truss. This truss simulates 8 of the case example's truss cells, corresponding to 8 repetitions of this task.

Typical assembly productivities in the MIT tests were 600 kg/pers-hr. Since the MIT 36-member truss masses 1008 kg, this corresponds to 4.8 truss cells per person-hour. Productivities in the MIT tests ranged from a high of 900 kg/pers-hr (7.1 cells/pers-hr) for experienced test subjects, to a low of roughly 200 kg/pers-hr (1.6 cells/pers-hr) for inexperienced subjects. As in task 1, these productivities are reduced by a 20% time overhead. Therefore:

OHhum[2] = Productivity of humans in performing task 2
= a range: low value = 1.3 reps/pers-hr
nominal value = 3.8 reps/pers-hr
high value = 5.7 reps/pers-hr

As in task 1, the teleoperator's estimated productivity ranges from (one-third of the human's) to (equal to the human's). This task involves well-defined operations, but in a worksite that is only partially predictable. Physical and perceptual difficulties are anticipated in maneuvering the teleoperator near the partly completed structure (Ref. C-10). Therefore the nominal teleoperator productivity is estimated at (0.7) of the human's. With the 20% time overhead factor:

OHtel[2] = Productivity of teleoperator in performing task 2
= a range: low value = 0.4 reps/tel-hr
nominal value = 2.7 reps/tel-hr
high value = 5.7 reps/tel-hr

This study's case example is kept simple by including only one type of automated equipment, the beam-builder. This device is not capable of assembling tetrahedral truss, and therefore its productivity in task 2 is set very close to zero (but not quite zero, to avoid divide-by-zero problems in the computer programs). Therefore:

```
OHaeq[2] = Productivity of automated equipment in task 2
         = 0.001 reps/aeq-hr
```

(It is possible to define automated equipment for this task: a variety of automated assemblers have been proposed for construction of large trusses, e.g. solar power satellites. In fact, specific automated equipment could be designed and modeled for each task in this case example. But this would complicate the example and make it more difficult to illustrate clearly the study's theoretical methods.)

C.4.4 Productivities in Assembly of Membrane and Surface Controls:

As mentioned in section C.1, detailed designs for the antenna membrane and surface controls are not available. Therefore only rough estimates of productivity are possible. Each repetition of Task 3 is the assembly of 706 square meters of membrane and associated surface controls to the antenna truss structure. It is expected that this includes many connections of control devices to the membrane and to the truss, and that this work must be done carefully to avoid damaging the hardware.

Therefore it is estimated that a human will take nominally 3 hours for one repetition of the task. The range of this time is estimated to run from 1 hour to 6 hours. Therefore:

```
OHhum[3] = Productivity of humans in performing task 3
         = a range: low value = 0.17 reps/pers-hr
                    nominal value = 0.33 reps/pers-hr
                    high value = 1.0 reps/pers-hr
```

The teleoperator's productivity range is estimated to run from (one-third of the human's) to (equal to the human's). Since this task involves delicate work in an environment which is only partially structured, the teleoperator's nominal productivity is estimated at (0.4) of the human's. Therefore:

```
OHtel[3] = Productivity of teleoperator in performing task 3
         = a range: low value = 0.06 reps/tel-hr
                    nominal value = 0.13 reps/tel-hr
                    high value = 1.0 reps/tel-hr
```

As in task 2, the automated equipment is incapable of performing this task, and is therefore assigned a near-zero productivity:

OHaeq[3] = Productivity of automated equipment in task 3
= 0.001 reps/aeq-hr

C.4.5 Productivities in Installation of Feed Horn Assemblies:

Task 4 is the installation of feed horn assemblies into the feed horn truss structure. for this example, it is assumed that each repetition of this task consists of positioning a feed horn assembly, making five structural connections, and making four cable connections (2 redundant power cables, 2 redundant data cables).

Estimates of ranges of times for humans performing these operations are presented in Table C.9. Based on those times, estimated human productivity is:

OHhum[4] = Productivity of humans in performing task 4
= a range: low value = 3.1 reps/pers-hr
nominal value = 5.6 reps/pers-hr
high value = 9.9 reps/pers-hr

TABLE C.9: ESTIMATED PRODUCTIVITY OF HUMANS IN THE INSTALLATION OF FEED HORN ASSEMBLIES

		Slow	Nominal	Fast
Positioning times (secs):		180	100	60
Structural connection times (secs):				
	Each connection	100	55	30
	5 connections	500	275	150
Cable connection times (secs):				
	Each cable	60	35	20
	4 cables	240	140	80
Total times for one assembly (secs):		920	515	290
	(mins):	15.3	8.6	4.8

	Low	Nominal	High
Human productivity (assemblies/pers-hr):	3.9	7.0	12.4
With 20% time overhead (reps/pers-hr):	3.1	5.6	9.9

As in earlier tasks, the teleoperator's productivity range is estimated as (one-third of the human's) to (equal to the human's). This task consists of well-defined but dextrous operations, in a controlled environment. Assuming that the teleoperator benefits from some preprogrammed routines for making structural and cable connections, and from its third arm for positioning, its nominal productivity is estimated at (0.85) of the human's. Therefore:

OHtel[4] = Productivity of teleoperator in performing task 4
= a range: low value = 1.0 reps/tel-hr
nominal value = 4.8 reps/tel-hr
high value = 9.9 reps/tel-hr

The automated equipment modeled in this case example is not capable of installing feed horn assemblies. As in task 2, it is therefore assigned a near-zero productivity:

OHaeq[4] = Productivity of automated equipment in task 4
= 0.001 reps/aeq-hr

APPENDIX D:

SPACE EQUIPMENT DESIGNS AND PARAMETERS

This appendix presents preliminary designs of major pieces of space hardware. This equipment is featured in this study's numerical examples of space construction (in Chapter 4). This hardware includes:

1) Habitation module
2) Extravehicular mobility unit (pressure suit)
3) Manned maneuvering unit
4) Construction teleoperator
5) Automated equipment
6) Construction site propulsion.

This appendix also presents a data base of past spacecraft costs, and uses it to estimate R&D and procurement cost factors for space hardware.

D.1 Construction Equipment:

For this study's case examples, the human option is represented by humans in extravehicular activity (EVA), using extravehicular mobility units (EMU, or pressure suits) and manned maneuvering units (MMU). The humans at the worksite also require habitation modules. All of this hardware includes support equipment (e.g. power systems) and requires spare parts and consumables.

The teleoperator option consists of a free-flying construction teleoperator and a teleoperator support station. The support station includes power systems and storage for spare parts and consumables.

The automated equipment option is represented by an automatic beam-builder, which is capable of producing a variety of trussbeam designs. This beam-builder includes a power generation system, and it requires spare parts but no consumables.

D.1.1 Habitation Module:

The habitation module design is based on the habitation modules in the August 1984 Space Station Reference Configuration Description (Ref. D-1). This reference configuration is the "Power Tower" Space Station, which consists of five pressurized modules, a considerable amount of truss structure, a variety of attached components (e.g. solar arrays, radiators, navigation equipment), and the Space Station payloads.

In the Space Station design, the human crew (nominally 6 people) lives in two Habitation Modules (HM) and benefits from some crew support equipment in the Logistics Module (LM). The Station also includes two Laboratory Modules, and a variety of payload attachment points are provided throughout the structure. A mass breakdown of the Space Station Reference Configuration is listed in Table D.1 (adapted from Table 4.2.4-1 in Ref. D-1).

The habitation module design used in this study is shown in Figure D.1. The principal element is a pressurized cylinder, which houses a nominal crew of three humans. Therefore:

NhumHAB = Number of humans in one habitation module
= 3 pers/hab

Projecting from one endcap is a trusswork structure which supports a cluster of replaceable fluid tanks. At the end of this trusswork is the rotation bearing for the two solar arrays and two thermal radiators. These are arranged in a windmill pattern so they will not shadow each other. Besides the central rotation bearing, individual gimbals give each array and radiator the two degrees of freedom needed to keep the arrays face-on and the radiators edge-on to the sun.

Two cutaway views of the habitation module are shown in Figures D.2 and D.3. The module's pressurized cylinder is a hybrid of the two Habitation Modules in the Space Station reference configuration, but it also includes the crew support equipment of the Logistics Module (e.g. refrigerator/freezer).

A mass breakdown of the habitation module is presented in Table D.2. These masses are for the habitation module only, and do not include the EMU and MMU primary and support hardware, although some of that hardware is shown in the cutaway drawings. The component masses are discussed in the following sections.

The Space Station reference configuration provides 75 kW of electrical power, using 5693 kg of solar arrays and gimbals, 3568 kg of storage and distribution hardware attached to the truss structure, and various masses of distribution hardware in the pressurized modules. However, of that 75 kW, only 25 kW are for housekeeping of the pressurized modules; the remainder is provided for the payloads. The solar arrays have a total area of 1785 square meters.

This study's habitation module only needs housekeeping power (plus support power for EMUs and MMUs, discussed later). Assuming that the Space Station's two Habitation Modules and part of the Logistics Module consume roughly two-thirds of the housekeeping power, the Station's estimated habitation power is 17 kW. This

TABLE D.1 SPACE STATION ELEMENT MASSES (KILOGRAMS)
(1984 REFERENCE CONFIGURATION)

SUBSYSTEM	HAB. MOD. 1	HAB. MOD. 2	LOG. MOD.	LAB. MOD. 1	LAB. MOD. 2	TRUSS STRUC.	ATTACHED COMPONENTS	PAYLOADS
ELECTRICAL POWER	292	210	94	229	229	3568	5693	-
GUIDANCE, NAV., & CONTROL	-	-	-	-	-	1693	-	-
COMMUNICATIONS & TRACKING	378	955	29	256	256	298	1045	59
DATA HANDLING	456	347	54	365	274	-	567	-
PROPULSION	-	-	166 (a)	-	-	-	823 (b)	-
ECLSS (c)	1744 (d)	1744 (d)	257 (e)	257 (e)	240 (f)	-	-	-
ATCS (g)	215	215	96	551	290	1092	2447	-
STRUCTURES & MECHANICAL	5879	5879	3315	5879	5879	6946	1462	-
CREW SYSTEMS	7125 (h)	5025 (i)	4138 (j)	1761	1761	-	-	-
PAYLOADS	-	-	5963	8449	15998	-	-	83612
TOTALS	16089	14375	14112	17747	24927	13597	12037	83671

(a) Dry mass (wet mass is 1255 kg).
(b) Dry mass (wet mass is 3152 kg).
(c) Environmental Control and Life Support System.
(d) Dry mass (wet mass is 2863 kg); includes 19 kg of EVA servicing equipment.
(e) Dry mass (wet mass is 421 kg).
(f) Dry mass (wet mass is 394 kg).
(g) Automated Thermal Control System.
(h) Includes 2926 kg of EVA hardware.
(i) Includes 675 kg of EVA hardware.
(j) Includes 850 kg of secondary structure and 1541 kg of crew support equipment.

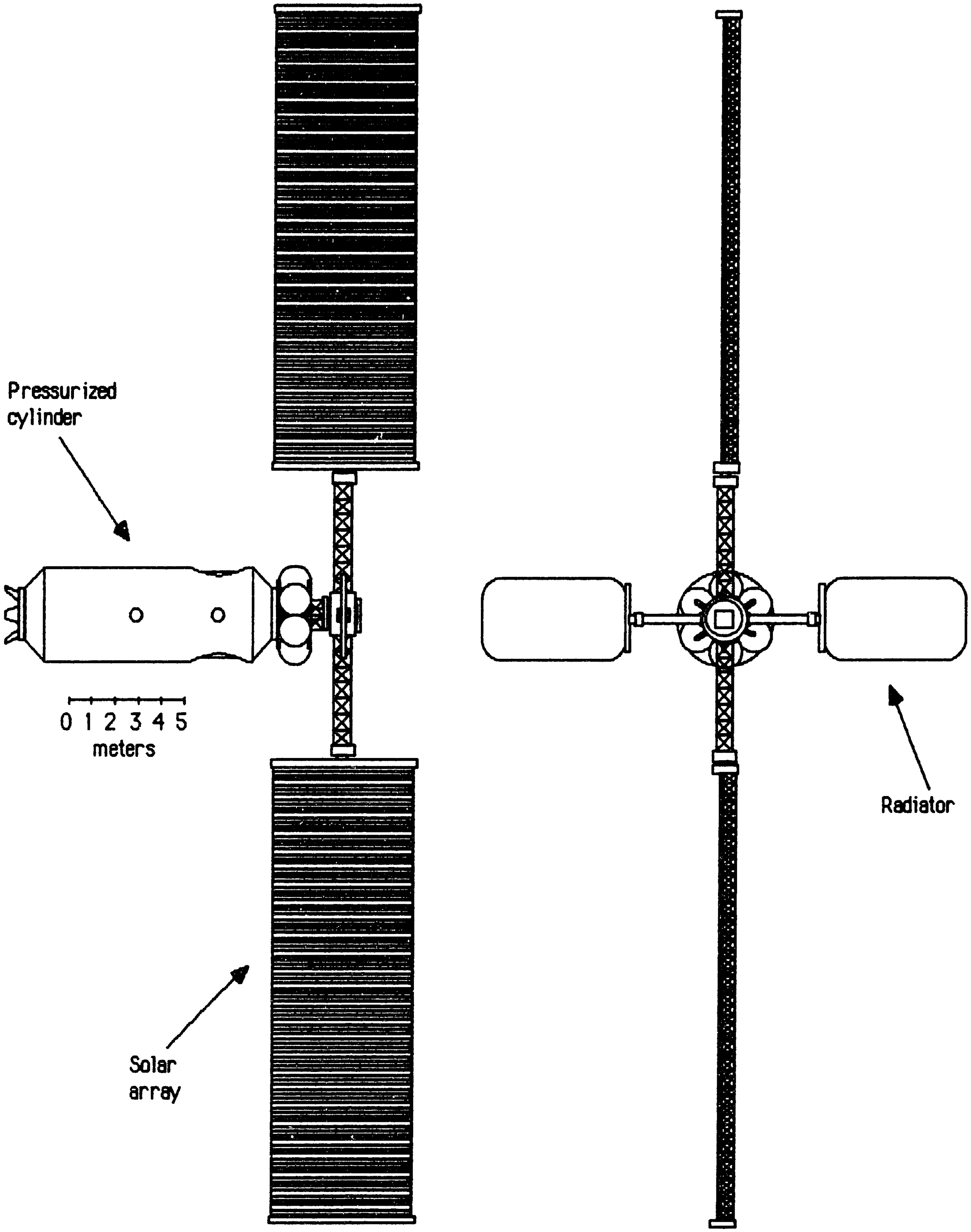


FIGURE D.1: OVERALL VIEW OF HABITATION MODULE

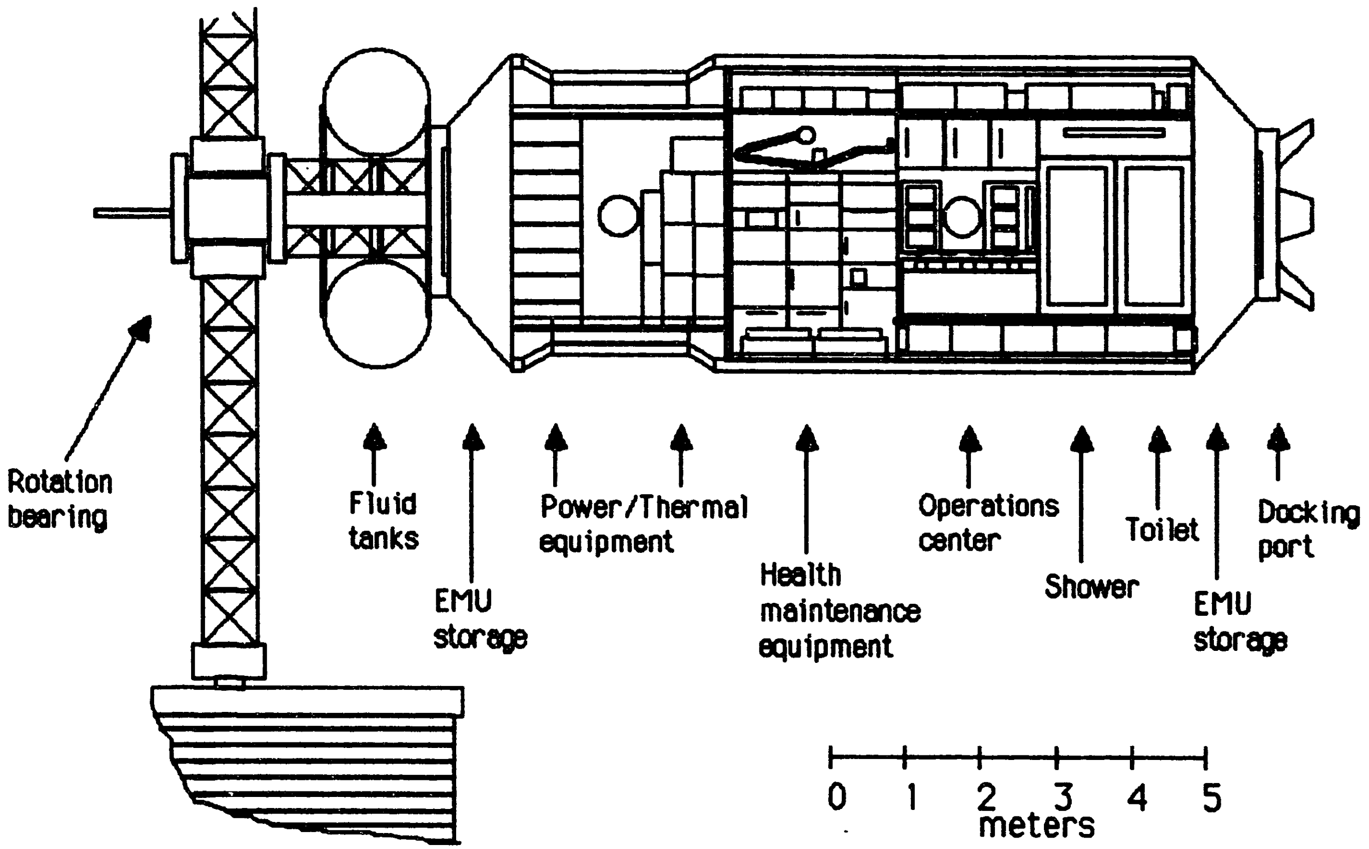


FIGURE D.2: LEFT CUTAWAY VIEW OF HABITATION MODULE

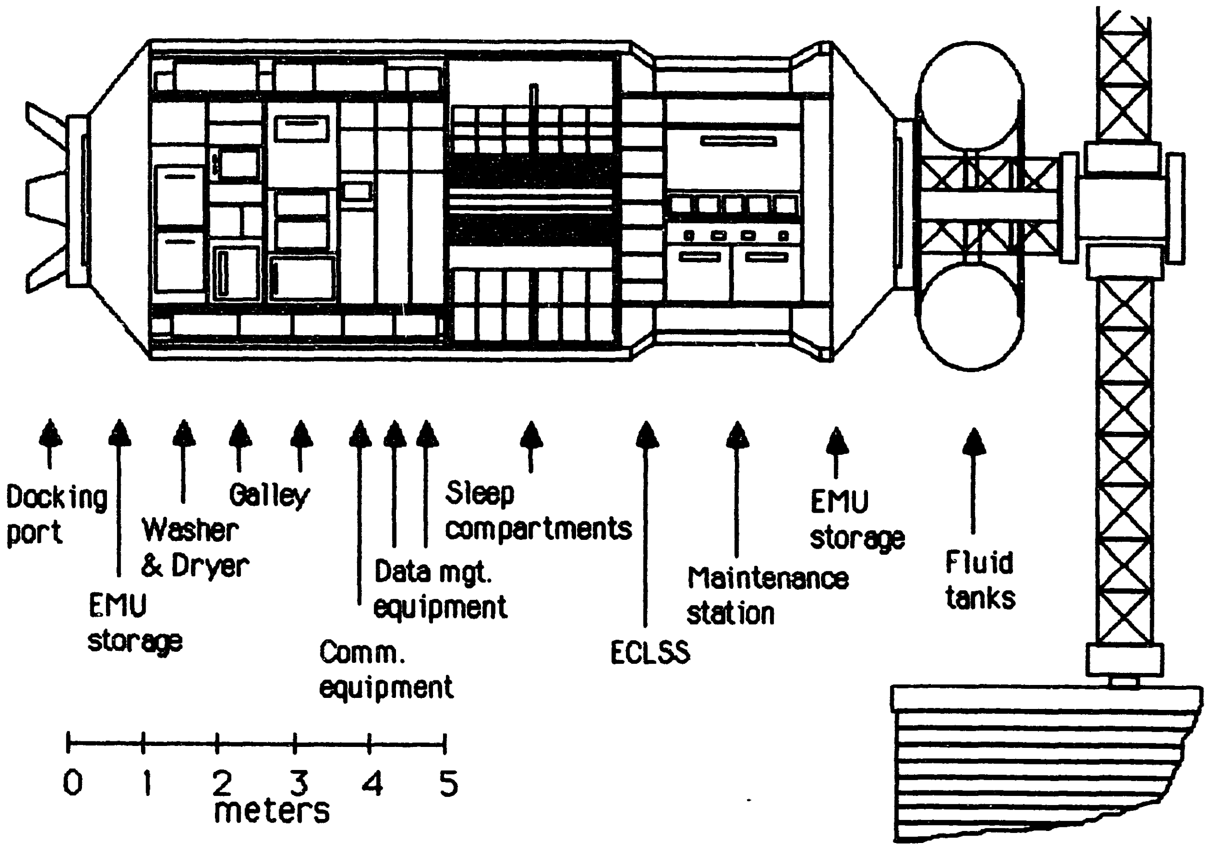


FIGURE D.3: RIGHT CUTAWAY VIEW OF HABITATION MODULE

TABLE D.2: HABITATION MODULE MASS BREAKDOWN
(KILOGRAMS)

Subsystem	Pressurized Cylinder	Truss Structure	Attached Components
Electrical Power	345	404 (a)	645 (a)
Guidance, Nav. & Control	-	282 (b)	-
Communications & Tracking	667	50 (b)	174 (b)
Data Handling	402	-	95 (b)
Propulsion	-	-	-
ECLSS	1725 (c)	-	-
ATCS	311	182 (b)	408 (b)
Structures & Mechanical	5879	1158 (b)	244 (b)
Crew Systems	5816 (d)	-	-
	------	-----	-----
TOTALS	15,145	2,076	1,566

Notes:
(a) Sized for 8.5 KW = (0.113) of reference Station power.
(b) One-sixth of reference Station value.
(c) Dry mass.
(d) Includes airlock and hatches (660 kg).

is for a nominal crew of 6 in two Habitation Modules, or twice the requirement for this study's module. Therefore the estimated power consumption of this study's habitation module is 8.5 kW.

(For comparison, the Shuttle Orbiter generates roughly 8.5 kW, of which 1.5 kW is for Orbiter use and 7 kW for payloads. When a pressurized Spacelab flies, roughly 3 of those 7 kW are used for Spacelab life support, and 4 kW are available for experiments. But the Shuttle and Spacelab are considered very lean on power, and the Orbiter uses flash evaporation for cooling, rather than active thermal control.)

Thus the habitation module's power requirement is (0.113) times the Station requirement. The masses for the solar arrays and gimbals and for the storage and main distribution components are scaled accordingly in Table D.2.

Within the pressurized cylinder, the distribution hardware is expected to be similar to that of the reference Station, and therefore the mass is the average of the HM values in Table D.1, plus the mass of the LM's electrical hardware (which is for crew support equipment).

For guidance, navigation, and control hardware, the reference Station's needs are substantial (1693 kg), since the Station consists of five modules and various attached payloads on a gravity-gradient-stabilized truss. It was therefore estimated that this study's module requires one-sixth of the Station's GN&C hardware, or 282 kg.

In communications and tracking, the reference Station is very busy, due to a variety of onboard activities and the need to monitor co-orbiting platforms. This study's module has lesser requirements. The pressurized cylinder's requirements are similar to those of the reference HMs, and the mass is therefore the average of the HM values in Table D.1. The masses of the truss structure components and attached components are estimated at one-sixth of the reference values.

A similar argument applies to data handling: the pressurized cylinder's requirement is the average of the HM data handling masses in Table D.1. The attached components mass is one-sixth of the reference value.

Although some propulsion components appear in the reference LM, they have been left out of this study's habitation module. Propulsion hardware for the entire worksite is included in the line item "construction support equipment".

The environmental control and life support system (ECLSS) in the habitation module is similar to those in the reference HMs. The Logistics Module ECLSS is there to provide shirtsleeve access to that module, and is therefore not needed here. Therefore the ECLSS mass is the same as for the Habitation Modules in Table D.1, minus 19 kg of EVA servicing equipment.

The automated thermal control system (ATCS) in the reference Station is sized to handle the solar load on the modules, the metabolic load of the humans in the modules, the thermal waste from housekeeping, and the thermal waste from payloads. In this study's habitation module, the pressurized cylinder's thermal loads are similar to those of the reference HMs, with the addition of the LM thermal load, which is mainly due to the refrigerator and freezer. Therefore the ATCS mass for the cylinder is the sum of the reference HM and LM ATCS masses.

The ATCS truss structure components and attached radiators are sized by the total thermal load of the habitation module. The solar load is one-fifth of the reference Station's; the metabolic and housekeeping loads are less than one-half of the Station's; there is no payload thermal waste. Therefore it was estimated that the habitation module's thermal load is one-sixth of the reference Station's, and the truss structure and attached component masses are scaled accordingly in Table D.2.

The structures and mechanical components are the cylindrical pressure shell and endcaps, the trusswork elements, and structural support fittings for attached components. The pressurized cylinder's shell is the same as the reference HMs, and therefore its mass is the same also. The truss structures and attached component structures are estimated to mass one-sixth of those in the reference Station.

Crew systems includes all the interior furnishings of the pressurized cylinder, including the airlock and hatches, most of the internal hardware (e.g. galley, health maintenance, hygiene, sleeping quarters), stowage containers, and housekeeping equipment. To calculate its mass, the EVA hardware masses (2926 kg and 675 kg) were first subtracted from the reference HM masses (7125 kg and 5025 kg); the results were then averaged. The 1541 kg of crew support equipment from the LM was then added, for a total of 5816 kg.

The physical model of the habitation module used in the line-item cost model consists of low-technology hardware and high-technology hardware. The low-tech hardware consists of the structures and mechanical components (7281 kg) plus the airlock's 660 kg from crew systems, for a low-tech total mass of 7941 kg. Therefore:

KGunHABl = Mass of low-tech hardware in one habitation module
= 7941 kg/hab

The high-tech hardware consists of everything else, with a total mass of 10,846 kg. However, since these are preliminary estimates, a ±30% range is applied. Therefore:

KGunHABh = Mass of high-tech hardware in one habitation module
= a range: low value = 7600 kg/hab
nominal value = 10,846 kg/hab
high value = 14,100 kg/hab

The spares requirements for the habitation module are based in part on the spares estimates in Table 4.2.4-6 of Ref. D-1. That table presents spares requirements for the various Station subsystems over their maintenance cycles. Based on this data, the 10-year spares requirement for each Station subsystem was calculated. These requirements were then divided by the total dry masses of the Station subsystems to generate a 10-year spares fraction for each subsystem.

These 10-year spares fractions were then applied to this study's habitation module design, as shown in Table D.3. This process estimates a total spares requirement of 2267 kg over 10 years for one habitation module. For simplicity, all of the habitation module spares are modeled as high-tech hardware in the

TABLE D.3: HABITATION MODULE SPARES REQUIREMENT (KILOGRAMS)

Subsystem	Total Dry Mass	10-year Fraction	10-year Spares Mass
Electrical Power	1394	.929	1295
Guidance, Nav. & Control	282	0	0
Communications & Tracking	891	0	0
Data Handling	497	.315	157
Propulsion	-	.089	-
ECLSS	1725	.208	359
ATCS	901	.134	121
Structures & Mechanical	7281	.046	335
Crew Systems	5816	0	0
TOTALS	18,787		2267

line-item cost model. Using the habitation module's high-tech hardware mass (10,846 kg), the 2267 kg to be replaced over ten years corresponds to 2.1% of the high-tech hardware per year (0.006% per day).

This value appears low. Current unmanned spacecraft have lifetimes approaching 10 years. If they failed completely at that time, this would correspond to 100% replacement at 10 years, or an average of 10% per year. Of course, unmanned spacecraft do not fail all at once, and some of their components could last well beyond ten years. On the other hand, unmanned spacecraft are not subject to the stresses of human habitation or worksite operations. So for the habitation module, 15% per year (0.04% per day) was taken as the high end of the spares fraction range. Therefore:

WFhab = Fraction of one habitation module worn out in one day
= a range: low value = 0.00006 hab/hab-day
nominal value = 0.0002 hab/hab-day
high value = 0.0004 hab/hab-day

The consumables requirements for the habitation module are based on the data in Tables 4.2.4-5 and 3.3.2.2-1 of Ref. D-1. These consumables are itemized in Table D.4.

The crew systems consumables for the reference Station are sized for a crew of six, and are therefore cut in half for this study's habitation module. This leads to a crew systems consumables estimate of 964 kg for three humans for 90 days.

TABLE D.4: CONSUMABLES REQUIREMENTS FOR HABITATION MODULE

Item	Consumables Requirement (kg/pers-day)
Crew Systems:	
Clothing	0.16
Personal Hygiene	0.18
Equipment Containers	0.06
Housekeeping Supplies	0.12
Ancillary Provisions	0.01
Galley	0.02
Food	2.95
Health Maintenance	0.05
Maintenance Workstation	0.02

Crew Systems Subtotal	3.57
ECLSS	1.11

Total Consumables Requirement	4.68

The ECLSS consumables for the reference Station (1515 kg for 90 days) include 671 kg of nitrogen for MMU propellant, 204 kg of air to replace cabin leakage from the entire Station, and 122 kg of air to make up airlock losses. The remaining 518 kg of ECLSS consumables is scaled for 6 people, and is therefore cut in half for the habitation module. The propellant for MMUs and the airlock losses will be incorporated in MMU and EMU consumables, discussed later. The estimated cabin leakage from one habitation module is one-fifth of the Station's, or 41 kg for 90 days.

This leads to an ECLSS consumables estimate of 300 kg for 90 days. It should be noted that the habitation module design includes atmospheric regeneration systems that do not require expendable LiOH canisters (such as those in the Shuttle), and that all the waste water is reprocessed. This keeps the ECLSS consumables requirement at this low figure. Therefore:

KGpdHABc = Kilograms/day of consumables for one human in space
= 4.68 kg/pers-day

D.1.2 Extravehicular Mobility Unit:

The line-item cost model assumes that humans in EVA use an Extravehicular Mobility Unit (EMU) which is a more advanced "blue-collar" version of the current Shuttle EMU. This study's EMU is not significantly different in mass or appearance from the Shuttle's, but it includes a number of improvements in performance:

1) all of its elements are rechargeable or replaceable on-orbit;

2) it can provide a nominal 8-hour EVA (0.25 hr egress, 7 hrs useful work, 0.25 hr ingress, 0.5 hr reserve);

3) its battery can be charged 12 times before replacement.

The 7 hours of useful work available per EVA sets the daily work time of the humans at the worksite. Therefore:

HDhum = Hours/day worked by humans in space
= 7 hours/day

To achieve these improvements in the EMU, an R&D program cost of 30 $million is estimated. Therefore:

CTrdEMU = Total cost of R&D of blue-collar pressure suit
= 30,000,000 $

The EMU and its support equipment are shown in Figure D.4. The EMU mass breakdown is presented in Table D.5. The pressure suit mass is from Ref. D-2 (p.2.3-39), with the contaminant control cartridge and battery scaled up slightly to provide 8 hours of EVA. The mass of EMU ancillary equipment is estimated from descriptions in Refs. D-2 and D-3. Masses of specific components are listed when available.

The EMU backup equipment consists of duplicates of other pieces of EMU equipment, so that a failure does not force an EMU out of action. The mass of this backup equipment is estimated at 30% of the EMU and ancillary equipment masses.

A mass breakdown of EMU support equipment and EMU consumables appears in Table D.6. The support equipment services the EMU while it is in the airlock. Most of the masses of EMU support equipment are estimated from descriptions in Refs. D-1, D-2, and D-3.

The EMU power requirement is roughly 420 watt-hr for one 8-hour EVA. Assuming one EVA per EMU per day, and a 50% charging efficiency for the EMU battery, this corresponds to 35 watts required continuously to recharge batteries for one EMU. More significant is the power for operation of pumps and valves in the EMU support equipment, for reprocessing of EMU waste water in the ECLSS, and for operation of the airlock. This requirement is estimated at 265 watts per EMU, for a total power requirement of 0.3 kW per EMU.

Using the sizing relationships developed for the habitation module, the masses of power equipment for support of one EMU are 12 kg in the pressurized cylinder, 14 kg in the truss structure, and 23 kg of attached components.

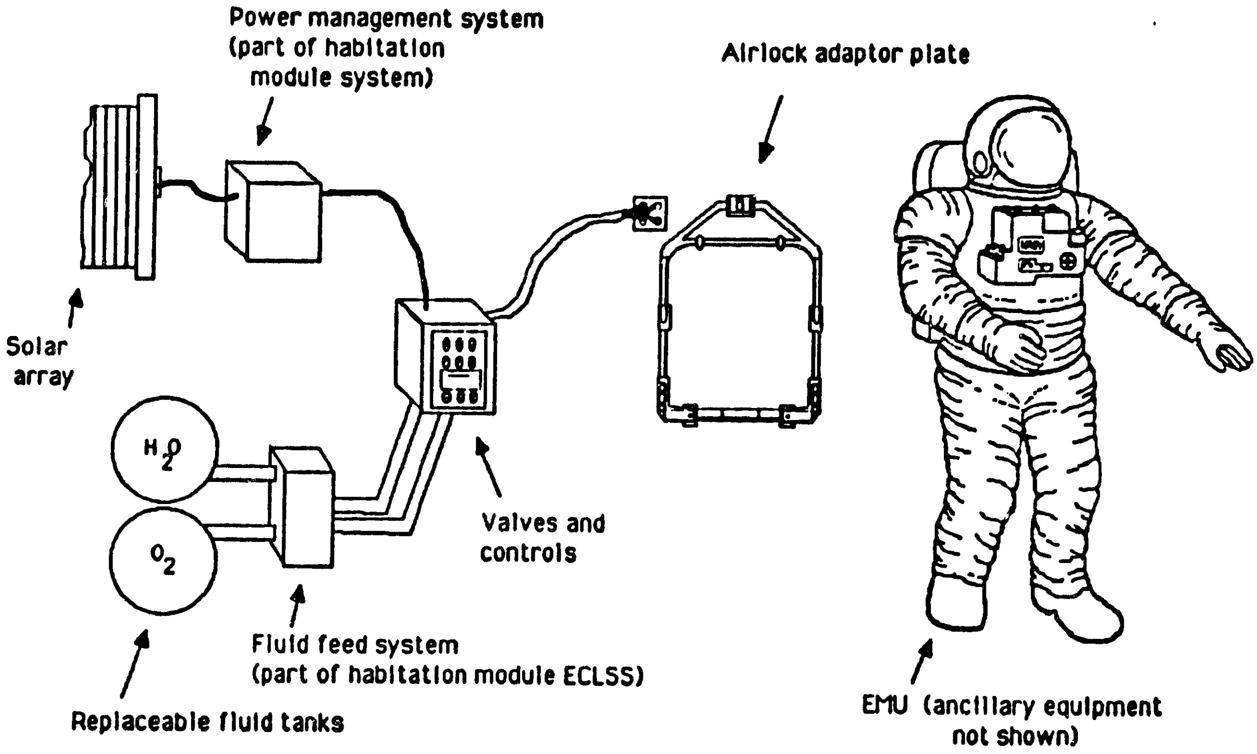


FIGURE D.4: EXTRAVEHICULAR MOBILITY UNIT (EMU) AND SUPPORT EQUIPMENT

The initial hardware mass for one EMU is the sum of the masses for the pressure suit, ancillary equipment, backup equipment, and support equipment. Therefore:

KGunEMU = Mass of one Extravehicular Mobility Unit (including
support equipment)
= 300 kg/emu

Ref. D-2 (p. 2.3-70) estimates that for each 7-hour EVA, 0.73 kg of oxygen and 4.54 kg of water are provided to each EMU, and 0.91 kg of waste water are collected from each EMU. Scaling these figures up for an 8-hour EVA yields 0.82 kg of oxygen and 4.15 kg of water required for each EMU; these figures are then increased by 10% for tankage.

TABLE D.5: MASS BREAKDOWN FOR EXTRAVEHICULAR MOBILITY UNIT

Item	Mass (kg)	
Pressure Suit		
Hard Upper Torso		
Arms (2)		
Gloves (2)		
Lower Torso Assembly		
Helmet/Visor Assembly		
Communications Carrier Assembly		
Liquid Cooling and Vent Garment	2.9	
Urine Collection Device		
Bioinstrumentation System		
Primary Life Support System	29.6	
Displays and Control Module	5.2	
Contaminant Control Cartridge	3.3	
Battery (420 W-hr)	5.1	
Secondary Oxygen Pack	9.5	
Extravehicular Communicator & Antenna		
Oxygen (primary and backup)	1.8	
Water	5.3	
PRESSURE SUIT TOTAL (DRY)		105
EMU Ancillary Equipment		
EMU lights	1.7	
EMU TV	13.4	
Tethers	0.5	
Wrist Mirror		
Cuff Checklist		
EV Watch		
Insuit Drinking Bag	0.1	
Insuit Food Dispenser		
EMU Scissors		
Mini-Work Station		
ANCILLARY EQUIPMENT TOTAL		25
EMU Backup Equipment		39
EMU TOTAL		169

One LiOH cartridge is expended per EVA. The EMU battery can be charged 12 times to support 12 EVAs, so its average consumption is one-twelfth of a battery per EVA. Ref. D-1 estimates 1.4 kg airlock losses for one EVA. This figure is multiplied by 1.1 (10% tankage); but three people go through the airlock at one time, so the result is divided by three to get the per-EMU figure.

TABLE D.6: EMU SUPPORT EQUIPMENT AND CONSUMABLES

Item	Mass (kg)
EMU Support Equipment	
Airlock Adaptor Plate	13.6
Service and Cooling Umbilical	5.5
Airlock Stowage Bag	1
EMU Prep Kit	0.5
EMU Maintenance Kit	2
Donning Handles	0.5
Pumps, Valves, Controls	40
ECLSS Hardware	19
Electrical Power (0.3 kW)	49

SUPPORT EQUIPMENT TOTAL	131.1
EMU Consumables (for one 8-hour EVA)	
Oxygen (incl. tankage)	0.90
Water (incl. tankage)	4.57
Contaminant Control Cartridge (1)	3.29
Battery (one-twelfth of one)	0.42
Airlock losses	0.51

CONSUMABLES TOTAL	9.69

The total requirement is 9.69 kg per EMU per EVA. Since each EVA yields 7 hours of useful work, the consumables requirement is:

KGehEMUc = Kilograms/hour of consumables for one EMU
= 1.38 kg/emu-hr

The EMU spares requirement is estimated by assuming that the EMU hardware (including support hardware) has a two-year lifetime in every-day operation. Since each day provides 7 hours of useful work, the entire EMU mass of 300 kg will be replaced over 5110 useful-work hours. Therefore:

WFemu = Fraction of one EMU worn out in one hour
= 0.0002 emu/emu-hr

D.1.3 Manned Maneuvering Unit:

The Manned Maneuvering Unit (MMU) assumed in this study is similar to the current MMU, except that:

1) all of its elements are rechargeable or replaceable on-orbit;

2) it can provide a nominal 7 hours of useful work during each EVA;

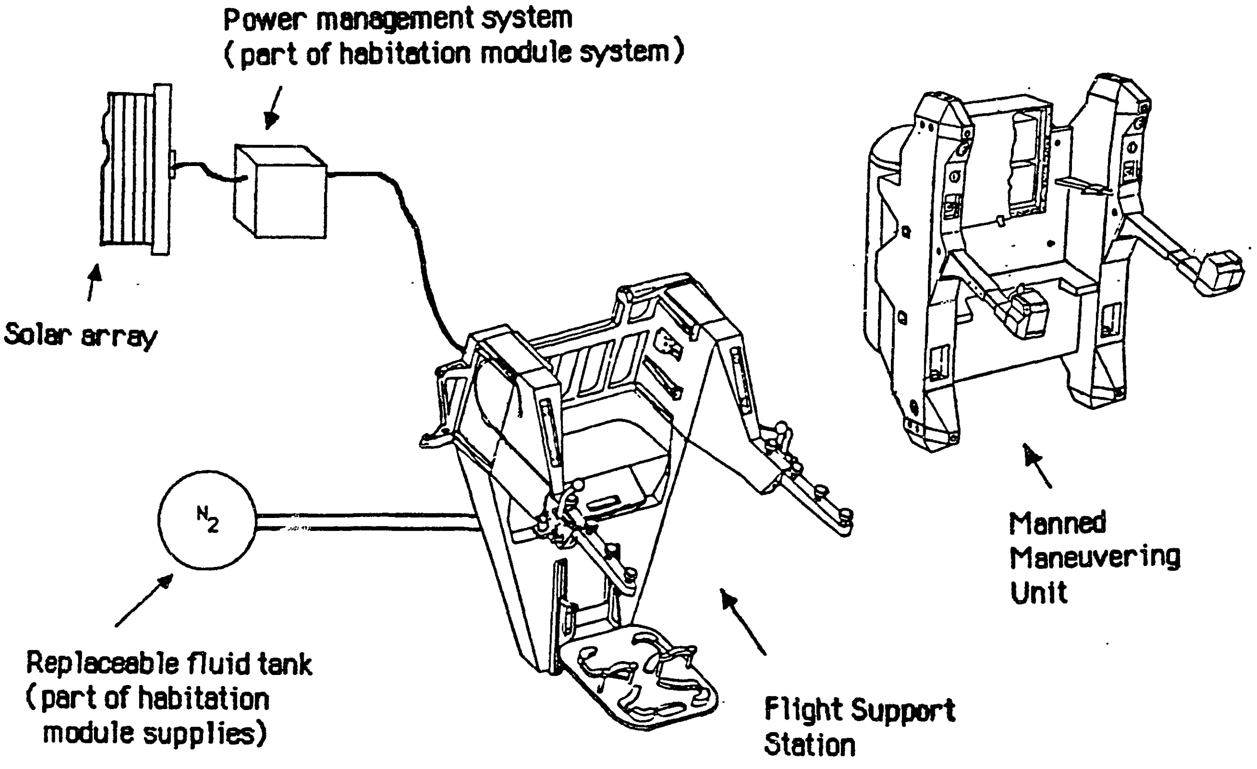
3) its propellant tanks have twice the current volume, to extend its time between recharges;

4) the MMU can be recharged from a portable propellant station.

The current MMU was developed from a Skylab prototype for 45 $million (1982$) (Ref. D-4). To achieve the above improvements, it is estimated that a further 20 $million is required. Therefore:

CTrdMMU = Total cost of R&D for Manned Maneuvering Unit
= 20,000,000 $

The Manned Maneuvering Unit is shown in Figure D.5. A mass breakdown of the MMU, its support equipment, and its consumables appears in Table D.7. The masses are from Table 1.3.3-2 in Ref. D-5; the MMU propulsion mass has been increased by 25 kg to account for the larger nitrogen tanks.



Note: Portable Propellant Station not shown.

FIGURE D.5: MANNED MANEUVERING UNIT AND SUPPORT EQUIPMENT

TABLE D.7: MASS BREAKDOWN AND CONSUMABLES FOR MANNED MANEUVERING UNIT

Item	Mass (kg)
Manned Maneuvering Unit (MMU)	
Propulsion (dry)	70
Electrical (incl. 2 batteries)	20
Structures & Mechanical	60
Thermal Control	3
Guidance & Navigation	13
Propellant	21

MMU TOTAL (DRY)	166
Flight Support Station (FSS)	
Propulsion/Charging (dry)	6
Electrical	2
Structures and Mechanical	95
Thermal Control	7

FSS TOTAL	110
Portable Propellant Station (PPS)	55
Electrical Power (0.14 kW)	24

MMU HARDWARE TOTAL	355
MMU Consumables (for 7 hours of EVA)	
Batteries (one-twelfth of two)	0.85
Nitrogen (incl. tankage)	80.85

CONSUMABLES TOTAL	81.7

The Portable Propellant Station allows recharging the MMU elsewhere than the FSS (which is attached to the habitation module). The PPS can be located within the worksite for easy access during construction. Its mass is one-half that of the FSS.

The MMU power requirement is roughly 840 watt-hr for one 7-hour EVA. Assuming one EVA per MMU per day, and a 50% charging efficiency for the MMU battery, this corresponds to 70 watts required to recharge batteries for one MMU. It is estimated that the power requirement for the FSS (heaters, nitrogen valves) adds another 70 watts, for a total power requirement of 0.14 kW per MMU.

Using the sizing relationships developed for the habitation module, the masses of power equipment for support of one MMU are 6 kg in the pressurized cylinder, 7 kg in the truss structure, and 11 kg of attached components.

The MMU's initial hardware mass is the sum of the masses of the MMU itself, the FSS, the PPS, and the electrical power hardware. Therefore:

KGunMMU = Mass of one Manned Maneuvering Unit (including support equipment)
= 355 kg

The MMU consumables are batteries and nitrogen propellant. The MMU carries two batteries, each similar to the 420 watt-hour EMU battery. The two batteries are sufficient for one EVA, and since they can be charged 12 times, the average consumption is one-twelfth of two batteries per MMU per EVA.

The current MMU has a usable nitrogen capacity of 10.5 kg (Ref. D-5), which is typically expended in one hour of flying time (Ref. D-4). Assuming that the construction activities require this typical consumption, 7 hours of MMU operation consumes 73.5 kg of nitrogen, plus 10% for tankage. Therefore:

KGmhMMUc = Kilograms/hour of consumables for one Manned Maneuvering Unit
= 11.67 kg/mmu-hr

The MMU spares requirement is estimated by assuming that the MMU hardware (including support hardware) has a four-year lifetime in every-day, 7-hours-per-day operation. Then the entire MMU mass of 355 kg will be replaced over 10,220 useful-work hours. Therefore:

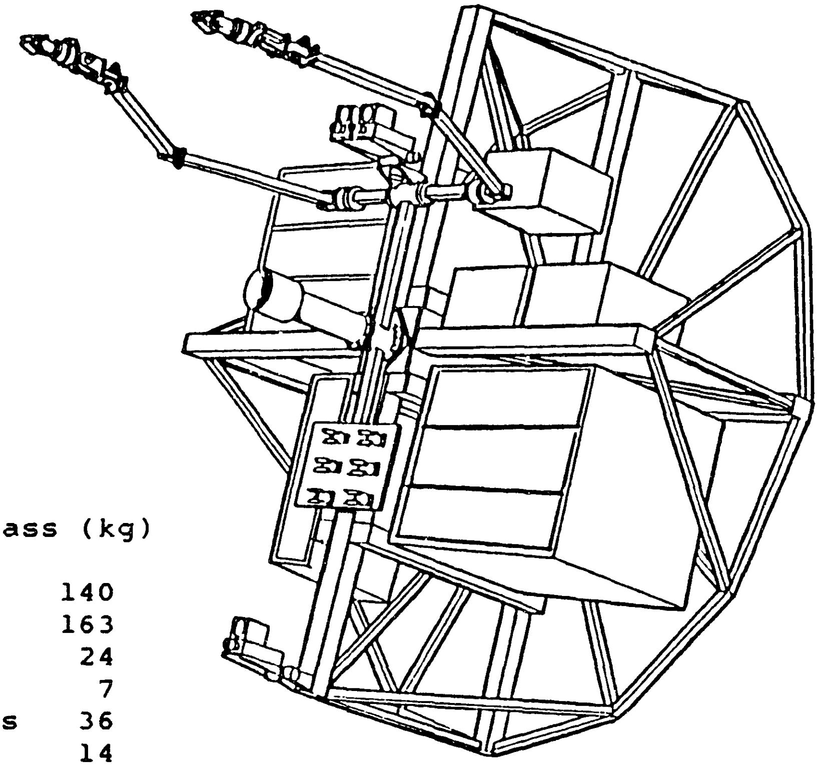
WFmmu = Fraction of one MMU worn out in one hour
= 0.0001 mmu/mmu-hr

D.1.4 Teleoperator Design:

The teleoperator design developed for this study is a combination of a remote servicing teleoperator concept developed by industry and a scaled-up version of this study's MMU.

A human in EVA consists of a human crewmember in an EMU, using an MMU as a mobility frame. The teleoperator's design philosophy is to replace the MMU's payload (the human and EMU) by a teleoperator payload, and then to redesign the MMU into an appropriate mobility frame. The teleoperator payload consists of a set of manipulator arms, cameras, electronics, power systems, and miscellaneous tools.

This payload resembles the Remote Orbital Servicing System (ROSS) developed by Martin Marietta (Ref. D-6), which is shown in Figure D.6. The ROSS is a multipurpose teleoperator kit to be attached to a mobility frame such as the Orbital Maneuvering System. Its principal element is a "manipulator assembly", consisting of a pair of 6-degree-of-freedom manipulator arms and stereo cameras mounted on a rotatable carriage. Each manipulator arm is 2.4 meters (8 feet) long.



Item	Mass (kg)
Manipulator Assembly	140
Stowage Rack	163
Thermal Closure	24
Peripheral TV	7
Electronic Assemblies	36
Electrical Cabling	14
SUBTOTAL	384
Batteries	Mission-specific

FIGURE D.6: REMOTE ORBITAL SERVICING SYSTEM
(MARTIN MARIETTA AEROSPACE, 1982)

The ROSS concept was adapted into this study's teleoperator, specialized for construction tasks. This teleoperator is shown in Figure D.7. It also uses a manipulator assembly with 2.4-meter arms and stereo cameras, but the carriage is only partially rotatable. An additional "anchor arm" is used either to hold the teleoperator fixed relative to the work area, or to hold a supply of construction materials.

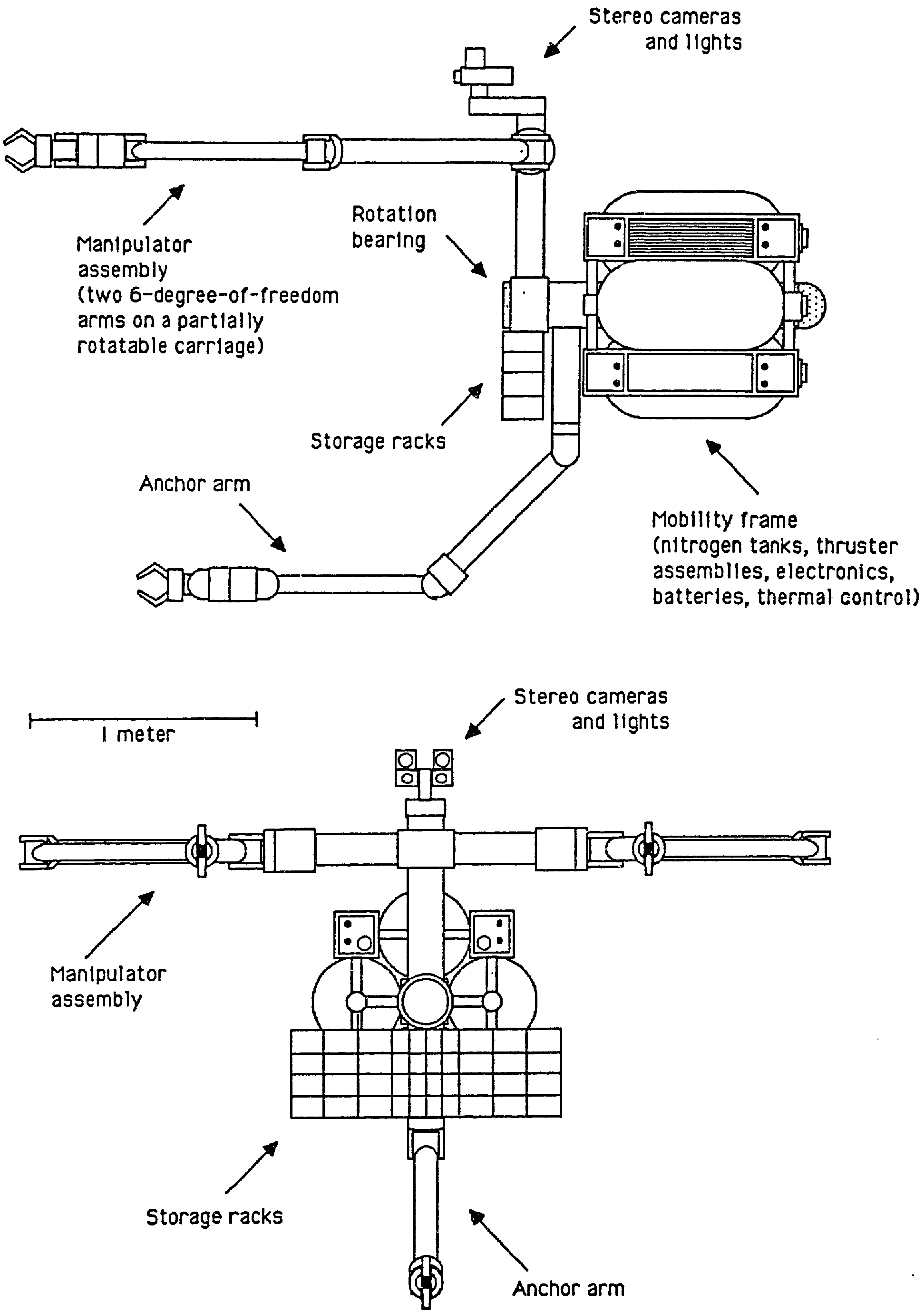

Stereo cameras and lights
Manipulator assembly (two 6-degree-of-freedom arms on a partially rotatable carriage)
Rotation bearing
Storage racks
Anchor arm
Mobility frame (nitrogen tanks, thruster assemblies, electronics, batteries, thermal control)
Stereo cameras and lights
1 meter
Manipulator assembly
Storage racks
Anchor arm

FIGURE D.7: TELEOPERATOR FOR CONSTRUCTION TASKS

The mass breakdown and consumables requirements of the teleoperator is presented in Table D.8. Most of the "Teleoperator Section" masses are carried over from the ROSS design. The large stowage rack is not needed, and is reduced to 20 kg of structure. The anchor arm mass is estimated as one-third of the manipulator assembly mass. The teleoperator also uses 27 kg of miscellaneous end-effectors and tools.

The operational power consumption of the "Teleoperator Section" is estimated to be 600 watts (five times that of the MMU). Since this design uses the same batteries as the MMU, and is intended to operate as long as the MMU between battery changes, the Teleoperator Section requires 10 onboard batteries.

The current MMU carries a typical payload of 178 kg (73 kg person, 105 kg EMU). The "teleoperator section" is the payload of the teleoperator's mobility frame. Since the teleoperator section masses 366 kg (2.1 times the MMU payload), the MMU components are scaled up by that factor for the teleoperator's mobility frame. Its electrical system therefore requires four 420-watt-hour batteries. The onboard propellant quantity is also scaled up, which results in the same delta-V capability as the MMU with human payload.

The Teleoperator Support Station provides propellant and battery recharge for the teleoperator. It is a rack mounted to the worksite, into which the teleoperator can be parked. It also includes storage for spare parts. The TSS component masses are scaled up from those of the MMU's Flight Support Station, using the (2.1) factor.

In addition, the TSS also includes power generation equipment: a gimballed solar array, storage cells, and a power management and distribution system, all mounted on a short truss.

The teleoperator's 14 420-watt-hour batteries power it for 7 hours of work. However, the teleoperator can operate longer than 7 hours per day, because the ground-based operators can work in shifts. Assuming that the teleoperator works as many as 21 hours per day, the power system must be scaled to recharge up to 42 batteries per day. At a 50% charging efficiency, this requires 1470 watts. Adding 230 watts for other TSS operations, the maximum power requirement for one teleoperator is 1.7 kW.

Using the sizing relationships developed for the habitation module, the power generation mass is 279 kg. An additional 30 kg is estimated for guidance, navigation and control hardware, and another 20 kg for monitoring and communications assemblies.

TABLE D.8: MASS BREAKDOWN AND CONSUMABLES FOR TELEOPERATOR

Item	Mass (kg)
Teleoperator Section	
Manipulator Assembly	140
Structure	20
Thermal closure	24
Peripheral TV	7
Electronic Assemblies	36
Electrical Cabling	14
Anchor Arm	47
Miscellaneous Tools	27
Batteries (10)	51

TELEOPERATOR SECTION TOTAL	366
Mobility Frame	
Propulsion (dry)	144
Electrical (incl. 4 batteries)	41
Structures and Mechanical	123
Thermal Control	6
Guidance and Navigation	27
Propellant	43

MOBILITY FRAME TOTAL (DRY)	341
Teleoperator Support Station (TSS)	
Propulsion/Charging (dry)	12
Electrical	4
Structures and Mechanical	195
Thermal Control	14
Electrical Power Generation (1.7 kW)	279
Guidance, Nav. & Control	30
Electronics	20

TSS TOTAL	554
TELEOPERATOR HARDWARE TOTAL	1261
Teleoperator Consumables (for 16 hours of work)	
Batteries (one-twelfth of 32)	13.6
Propellant (incl. tankage)	380.0

CONSUMABLES TOTAL	393.6

The initial hardware mass of the teleoperator is the sum of the masses of the Teleoperator Section, Mobility Frame, and Teleoperator Support Station. However, since these are preliminary estimates, a ±50% range is assumed. Therefore:

KGunTEL = Mass of one teleoperator (including support equipment)
= a range: low value = 631 kg/tel
nominal value = 1261 kg/tel
high value = 1892 kg/tel

The daily work time of the teleoperator can vary. Therefore:

HDtel = Hours/day worked by teleoperators in space
= a range: low value = 8 hours/day
nominal value = 16 hours/day
high value = 21 hours/day

For a nominal 16-hour workday, the consumables requirements for one teleoperator are listed in Table D.8. Sixteen hours of work requires 32 batteries' worth of energy, therefore 32 battery recharges. Each recharge wears out one-twelfth of a battery's life. The propellant consumption is 21.59 kg/hour (2.1 times that of the MMU), plus 10% for tankage. Therefore:

KGthTELc = Kilograms/hour of consumables for one teleoperator
= 24.6 kg/tel-hr

As in the habitation module design, there is some uncertainty in the spares requirement for the teleoperator and its support equipment. Assuming an every-day, 16-hours-per-day operation, the teleoperator works 5840 hours in a year. A low estimate for replacement is 10% of the hardware per year (10-year lifetime); a high estimate is 50% per year (2-year lifetime). The nominal spares estimate is 20% per year. Therefore:

WFtel = Fraction of one teleoperator worn out in one hour of operation
= a range: low value = 0.00002 tel/tel-hr
nominal value = 0.00003 tel/tel-hr
high value = 0.00009 tel/tel-hr

D.1.5 Automated Equipment Design:

The automated equipment used in this study's case example is an adjustable beam-builder, capable of producing graphite/epoxy trusswork beams in a variety of sizes and configurations. Its design is based on a beam-builder concept developed by General Dynamics, shown in Figure D.8.

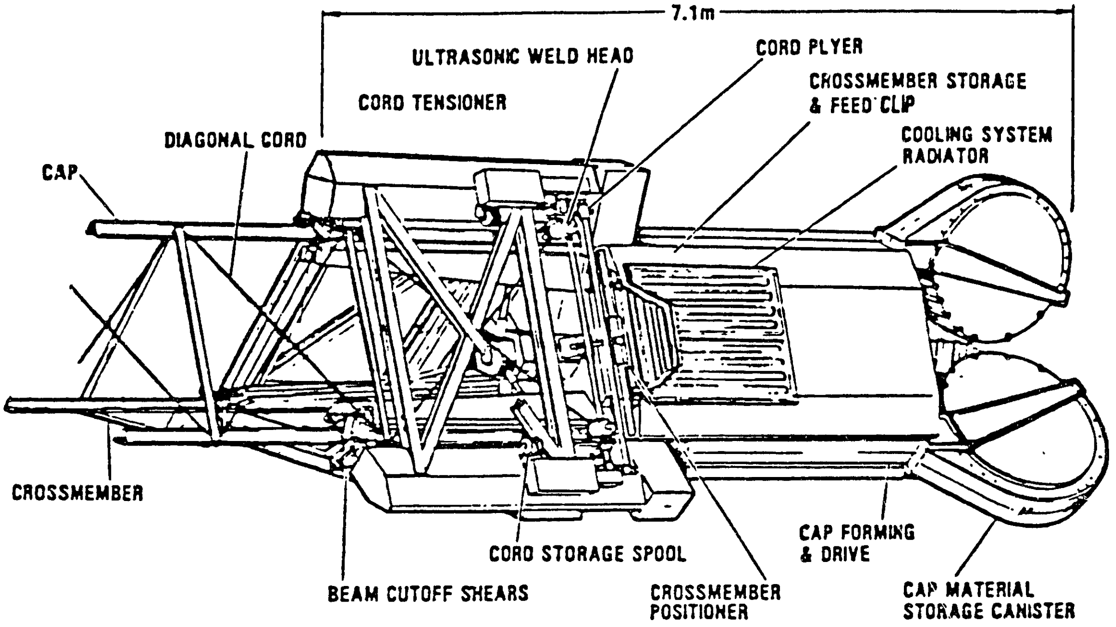


FIGURE D.8: GRAPHITE/EPOXY BEAM-BUILDER CONCEPT
(GENERAL DYNAMICS CONVAIR)

This General Dynamics design (Ref. D-7) has an estimated mass of 2600 kg, not including beam material. A more detailed design of this beam-builder is proposed as a Shuttle flight experiment (Ref. D-8). This proposed design has a mass of 1830 kg, plus 1685 kg for support structure in the Shuttle payload bay, and 2187 kg of test instrumentation. The nominal truss forming speed is 1.1 meters/minute, and the average power requirement is 2 kW.

The automated equipment design developed for this study is shown in Figure D.9. Besides a beam-builder section, it includes an endjoint installer, which brings the ends of the beam caps together and welds them to beam endjoints. The relative positions of the components can be changed by adjusting the external structural framework, to vary the cross-section of the output beam. This framework also includes a short truss supporting the electrical power components: a gimballed solar array, storage cells, and a power management and distribution system.

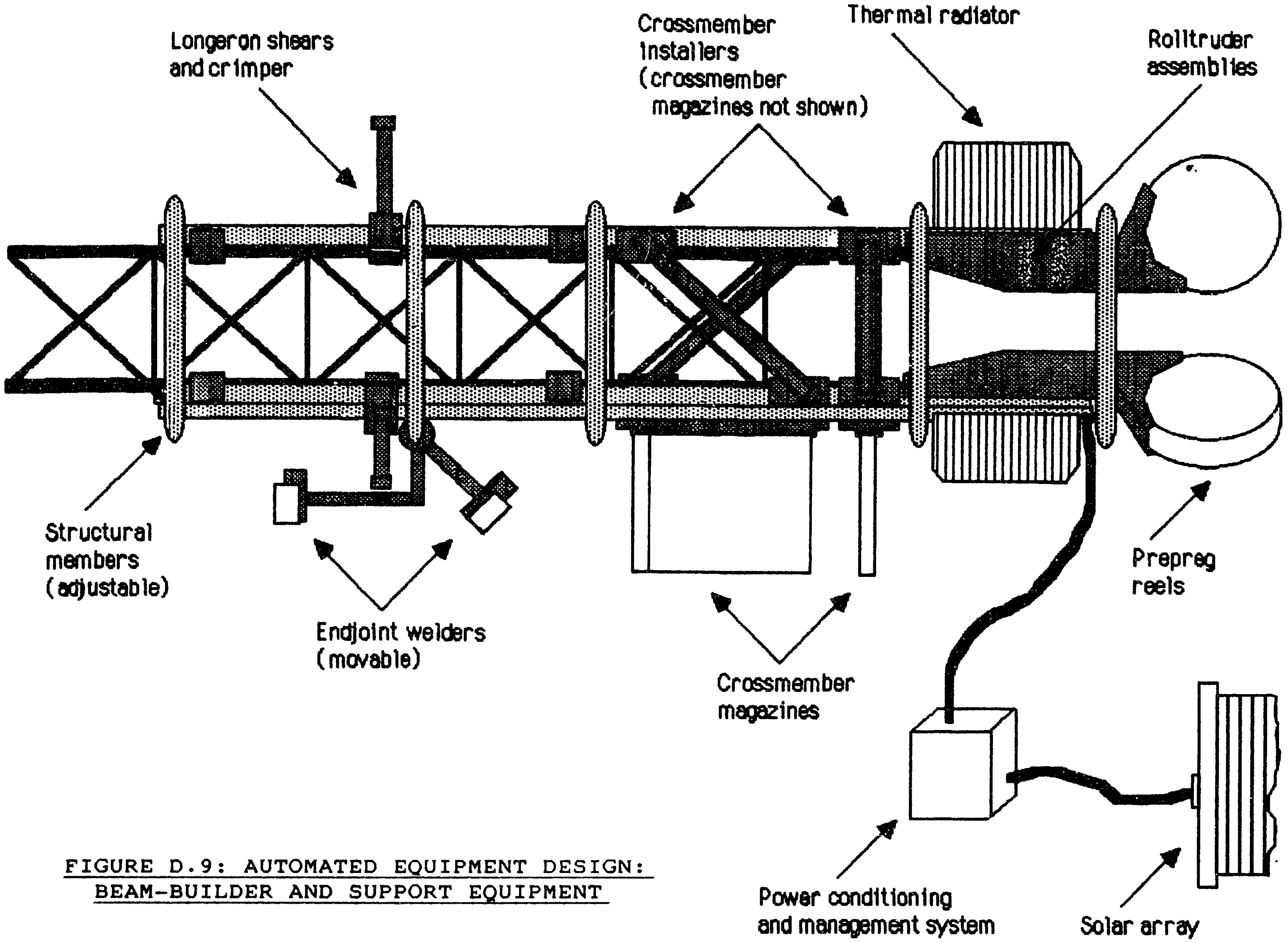

Longeron shears and crimper
Crossmember installers (crossmember magazines not shown)
Thermal radiator
Rolltruder assemblies
Structural members (adjustable)
Endjoint welders (movable)
Crossmember magazines
Prepreg reels
Power conditioning and management system
Solar array

FIGURE D.9: AUTOMATED EQUIPMENT DESIGN: BEAM-BUILDER AND SUPPORT EQUIPMENT

A mass breakdown for this automated equipment design is shown in Table D.9. The beam-buider component masses are from Ref. D-8. The endjoint installer component masses are estimates based on the beam-builder masses.

TABLE D.9: MASS BREAKDOWN OF AUTOMATED EQUIPMENT

Item	Mass (kg)
Beam Builder	
Beam Builder Structure	281
Cross-Member Cutoff Mechanism (3)	18
Cross-Member Translation Mechanism (3)	42
Bobbin and Traverse Mechanism (3)	48
Ultrasonic Weld Heads (6)	30
Weld Head Positioning Mechanism (6)	24
Cross-Member Rolltruder Assembly (3)	417
Beam Cap Rolltruder (3)	713
Beam Cutoff Mechanism (3)	21
Electrical Equipment	236

BEAM BUILDER TOTAL	1830
Endjoint Installer	
Endjoint Installer Structure	10
Beam Cap Positioner (3)	12
Endjoint Positioner (1)	5
Ultrasonic Weld Heads (3)	15
Weld Head Positioning Mechanism (3)	12
Electrical Equipment	10

ENDJOINT INSTALLER TOTAL	64
Electronics	40
Thermal Control	50
External Structure (incl. truss)	250
Electrical Power Generation (2.3 kW)	377
Guidance, Nav., and Control	30

AUTOMATED EQUIPMENT TOTAL	2641

The mass of electronic assemblies for process monitoring and communications is estimated at 40 kg. Thermal control hardware for the rolltrusion processes adds another 50 kg (in Ref. D-8's Shuttle flight experiment, the Orbiter provides cooling). The external structure, including the truss for the power generation equipment and attachment fittings to the worksite, has an estimated mass of 250 kg.

The beam-builder needs 2 kW of average power. The endjoint installer only runs occasionally; its average power is estimated at 50 watts. Another 250 watts is estimated to operate various electronic hardware. Using the sizing relationships developed for the habitation module, the total power generation of 2.3 kW requires 377 kg of hardware. The guidance, navigation, and control equipment, used to point the solar array, adds 30 kg.

The sum of the masses in Table D.9 is 2641 kg. This figure is used as the nominal mass of one unit of automated equipment. However, since the component masses are based on a flight experiment design rather than an operational design, this figure may be low. Therefore 2641 kg is also used as a low estimate; the high end of the range is estimated as 1.5 times higher. So:

KGunAEQ = Mass of one unit of automated equipment (including
support equipment)
= a range: low/nominal value = 2641 kg
high value = 3962 kg

Similarly to the teleoperator, the automated equipment's daily work time can be long, since its ground monitoring crew can work in shifts. Therefore:

HDaeq = Hours/day worked by automated equipment
= a range: low value = 8 hours/day
nominal value = 20 hours/day
high value = 23 hours/day

The automated equipment needs no consumables, but it does require spare parts. Assuming the nominal every-day, 20-hours-per-day operation, each unit of automated equipment will work 7300 hours per year. Because its functions are well defined, automated equipment can be designed for longevity. However, its heavy duty cycle offsets some of that advantage. A low estimate for replacement is 10% of the hardware per year (10-year lifetime); a high estimate is 50% per year (2-year lifetime). The nominal spares estimate is 20% per year. Therefore:

WFaeq = Fraction of one unit of automated equipment worn out in
one hour of operation
= a range: low value = 0.00001 aeq/aeq-hr
nominal value = 0.00003 aeq/aeq-hr
high value = 0.0C007 aeq/aeq-hr

D.1.6 Construction Site Propulsion:

At an orbital altitude of 370 km, the construction site moves through the rarified upper atmosphere, thus incurring an orbital drag. This drag is a function of the worksite's cross-sectional area in its direction of travel, and of the dynamic pressure applied by the atmosphere.

During construction, the radiometer's drag includes the drag of the trusswork beams and the drag of the antenna membrane. The overall effect can be minimized by keeping the radiometer edge-on to the direction of travel.

Assuming that each 18-meter truss beam has a longeron width of 5 cm, and crossmember and diagonal widths of 2 cm, then the beam has a plan-view cross-sectional area of roughly 4 square meters. The tetrahedral truss puts the beams at a variety of angles relative to the velocity vector, reducing the overall cross-sectional area by roughly 30%. Assuming another 30% reduction because of "shadowing" of some beams by others, the 16,688 beams in the complete structure have an effective drag area of:

(0.7) * (0.7) * (16,688 beams) * (4 square meters/beam)

= 32,700 square meters

The completed antenna membrane, edge-on to the velocity vector, has an area of roughly 20,500 square meters. When the structure and membrane are combined, however, considerable shadowing of the beams by the membrane occurs, for an estimated total drag area of:

(20,500) + ((0.6) * (32,700)

= 40,120 square meters

It is anticipated that judicious scheduling of construction work (to minimize cross-sectional area until late in the assembly process) can keep the average cross-sectional area of the radiometer at about one-third of this figure, i.e. 13,400 square meters.

(By comparison, the Shuttle Orbiter has a belly-on cross-sectional area of 340 square meters. So the worksite has average drag area equivalent to roughly 39 Orbiters.)

At 370 km (200 nautical miles), the atmospheric density is 4.82×10^{-12} kg/cubic meter (Ref. D-9), but this figure can vary by a factor of ten due to sunspot activity. At this altitude, the orbital velocity is 7687 meters/second (Ref. D-10), so the dynamic pressure is:

(0.5) * (density) * (velocity^{2})

= 0.00014 kg/m-sec^{2}

Modeling the worksite components as spheres in particle flow, the coefficient of drag Cd = 2, and the total drag is:

(Cd) * (area) * (dynamic pressure)

= (2) * (13,400) * (0.00014)

$= 3.82 \text{ kg-m/sec}^2 = 3.82 \text{ Newtons}$

Therefore the worksite requires an average thrust of 3.82 Newtons (0.86 lbf) to offset the orbital drag.

Assuming that the drag compensation is provided by a set of bipropellant engines (monomethyl hydrazine and nitrogen tetroxide: specific impulse = 340 secs; Ref. D-10), the thrust is:

(massflow) * (exhaust velocity) = 3.82 Newtons (0.86 lbf)

Since:

$(\text{exhaust velocity}) = (340 \text{ secs}) * (9.8 \text{ meters/sec}^2)$
= 3332 meters/sec

the propellant massflow is:

(massflow) = (3.82 Newtons) / (3332 meters/sec)
= 0.00115 kg/sec
= 99.1 kg/day

This is the nominal propellant requirement. To account for uncertainties, mostly in the sunspot cycle, this figure is extended into a range. Also, the figures are increased by 10% to include tankage. Therefore:

KDcsp = Kilograms/day of construction site propellant requirement
= a range: low value = 54.5 kg/day
nominal value = 109 kg/day
high value = 545 kg/day

The propulsion hardware for the worksite is part of the "construction support equipment". To estimate its mass, it is compared to the propulsion system for the NASA Space Station Reference Configuration (Ref. D-1). The Space Station uses a monopropellant system, with a thrust of 25 to 75 lbf, and a mass of 989 kg (see Table D.1).

For a bipropellant system with a thrust of 40 Newtons (9 lbf), a scaling-down factor of 3 is estimated. (The thrust is set well above the average 3.82 Newtons to deal with drag variations during construction and contigencies.) This yields an estimated propulsion hardware mass of 330 kg. Because low-thrust bipropellant propulsion is a well-established technology, this equipment is considered to be low-technology hardware.

The construction support equipment also includes various jigs and fixtures used during worksite operations. The mass of this hardware is estimated at 1000 kg. Therefore:

KGunCSE = Mass of one unit of construction support equipment
= 1330 kg/unit

D.2 Cost Factors for R&D and Procurement:

The line-item cost model calculates R&D and procurement costs for most of the space equipment by using dollar/kilogram cost factors for high-technology and low-technology hardware. This section estimates values for these cost factors from data on past NASA spacecraft.

D.2.1 Cost Factors for High-Technology Hardware:

In this study, "high-technology hardware" is defined as the type of hardware used in typical past NASA spacecraft: structure, electronics, power and thermal systems, experiments. Therefore the R&D and procurement cost factors for high-technology hardware are based on the costs and masses of past spacecraft presented in Table D.10.

Most of Table D.10's cost data is developed from Mission Operation Reports (MOR), such as the HEAO MORs (Ref. D-11). In some cases (e.g. Solar Max, Relay) the MOR data is supplemented or replaced by data from the NASA Budget Estimates (Ref. D-12). The costs from these sources are entered on computer spreadsheets, and the yearly costs are adjusted into 1984 dollars, using a set of inflation factors provided by NASA Headquarters. The numbers of spacecraft built and the spacecraft masses are based on data in the MORs, the NASA Budget Estimates, and the NASA press releases at the time of launch (e.g. the HEAO Press Kits, Ref. D-13).

The Shuttle Orbiter cost data is from a detailed spreadsheet of Shuttle development and production costs (presented in Appendix K), produced from data in the NASA Budget Estimates (Ref. D-12, fiscal years 1972 through 1976). The number of Shuttle Orbiters is set at 5.5: Enterprise, the four flight vehicles (Columbia, Challenger, Discovery, Atlantis), and the equivalent of half an Orbiter in major spare parts (e.g. fuselage and wing structures). The Orbiter mass is taken from Ref. D-14. Note that the Orbiter costs and mass do not include the Space Shuttle Main Engines; rocket engines are not representative of the space hardware being modeled in this study.

TABLE D.10: COST AND MASS DATA FOR PAST SPACECRAFT

Spacecraft	R&D and procurement of all spacecraft ($million; 1984$)	Number of spacecraft built	Typical spacecraft mass (kg)
HEAO	395.2	3	2867
OAO	1325.9	4	2022
Landsat 1&2	385.2	2	947
Landsat 3	75.9	1	900
Landsat 4	600.1	2	1941
Solar Max	113.3	1	2315
Relay	74.5	2	78
Voyager	470.4	2	825
Viking	1805.4	2	3519
OGO	800.5	6	555
OSO-1	18.8	1	200
OSO-2	43.5	2	247
OSO-3	20.5	1	284
OSO-4	52.2	1	272
OSO-5	41.7	1	291
OSO-6	42.5	1	290
OSO-7	82.0	1	635
OSO-8	120.2	1	1064
TIROS-N	87.0	1	723
Shuttle Orb.	15998	5.5	67347
Spacelab	1239	1	9225

Notes: HEAO = High Energy Astronomy Observatory
OAO = Orbiting Astronomical Observatory
OGO = Orbiting Geophysical Observatory
OSO = Orbiting Solar Observatory
TIROS = Television Infra-Red Observation Satellite
Shuttle Orb. = Shuttle Orbiter (without Main Engines)

The Spacelab cost is the 1984-dollar equivalent of an overall program cost estimate (including delivery of one flight unit) in Ref. D-15. The mass estimate is from the same source.

Each cost figure in Table D.10 includes both the R&D costs and the procurement costs for one or more spacecraft of that type. In general, the only available cost data is in this form, i.e. without any breakdown between R&D costs and procurement costs, and without any breakdown between individual spacecraft in a group.

However, this study's line-item cost model requires separate cost factors for R&D and procurement, applicable to individual pieces of hardware. To break down the data in Table D.10 into more useful forms, some models of spacecraft expenditures are developed below.

Some of the items in Table D.10 are individual spacecraft: Landsat 3, Solar Max, OSO-1, OSO-3 through OSO-8, and TIROS-N. For these spacecraft, the cost in the Table includes the procurement of only one flight unit. To separate these spacecraft's R&D and procurement costs, it is assumed that the overall cost can be expresed as:

$$\text{(R\&D and procurement of spacecraft)}$$

$$= \text{(R\&D cost)} + \text{(Unit procurement cost of spacecraft)}$$

$$= \left(\frac{\text{(R\&D cost)}}{\text{(Unit proc. cost)}} * \text{(Unit proc. cost)}\right) + \text{(Unit proc. cost)}$$

and that $\frac{\text{(R\&D cost)}}{\text{(Unit proc. cost)}}$ can be modeled as a constant.

Some data on the relative sizes of R&D and unit procurement costs is available. The detailed Shuttle spreadsheet in Appendix K yields a procurement cost of 5014 $M (in 1984$) for 3 Orbiters, the last 3 in the production run of 5.5. Assuming a 90% learning curve in Orbiter production, this corresponds to a first-unit cost of 2092 $M, and a total procurement cost of 9873 $M. Subtracting this figure from the R&D-and-procurement total of 15,998 $M gives an R&D estimate of 6125 $M (1984$) for the Orbiter (without Main Engines). The ratio of R&D cost to first-unit cost is therefore 2.9; the ratio of R&D cost to average unit cost is 3.4.

The Spacelab cost figure in Table D.10 includes the procurement of one flight unit. Ref. D-15 suggests that a second flight unit costs 212 $M (1984$). Assuming a 90% learning curve, the first-unit cost is 235.3 $M. Subtracting this from the total cost of 1239 $M gives an R&D estimate of 1004 $M (1984$). Therefore the ratio of R&D cost to first-unit cost is 4.3.

The Air Force's Space Division Unmanned Spacecraft Cost Model (Ref. D-16) uses statistical analyses of spacecraft costs to generate separate cost estimating relationships for R&D and procurement of future spacecraft. Although this model's data base is not available, and its cost relationships are not linear with mass, it can be applied to hypothetical spacecraft and the R&D-to-procurement ratio computed in each case. For a low-mass spacecraft (185 kg) the predicted R&D cost is 24.4 $M and the procurement cost is 6.9 $M (both in 1979$), and the ratio is 3.5. For a medium-mass spacecraft (609 kg), the R&D prediction is 55.3 $M, the procurement is 17.3 $M (both 1979$), and the ratio is 3.2.

Based on this evidence, it was decided to set:

$$\frac{\text{(R\&D cost)}}{\text{(Unit proc. cost)}} = 3.5$$

Therefore, for the individual spacecraft in Table D.10:

(R&D and procurement of spacecraft)

= (3.5 * (Unit proc. cost)) + (Unit proc. cost)

= (4.5) * (Unit proc. cost)

Other items in Table D.10 are groups of identical spacecraft: Relay, Voyager, Viking, and OSO-2. In some of these cases, redundant spacecraft were built for greater chance of mission success (Relay, Voyager, Viking). The first OSO-2 was destroyed during launch; an identical replacement was launched later. Thus the e cost figures for these spacecraft types include the procurement of several identical flight units.

To separate the R&D and procurement costs of these spacecraft, the formula developed above was extended:

(R&D and procurement of all spacecraft)

= (3.5 * (Unit proc. cost))

+ (Number of spacecraft built) * (Unit proc. cost)

This preserves the R&D-to-unit-procurement ratio of 3.5, while multiplying the unit procurement cost by the number of flight units produced.

Finally, some of the items in Table D.10 are families of spacecraft: HEAO, OAO, Landsat 1&2, Landsat 4, and OGO. Within each family, all the spacecraft have generally similar designs, but with some modifications specific to each spacecraft. Their cost figures include R&D for the family of spacecraft, R&D for the specific spacecraft designs, and procurement for the flight units. (It can be argued that Landsat 3 belongs in the Landsat family, and that the OSO spacecraft make up a family; however, since individual costs are available for each spacecraft, they are treated separately.)

The costs of families of spacecraft are modeled as follows:

(R&D and procurement of all spacecraft)

= (Basic R&D for spacecraft family)

+ (Number of spacecraft built)
* ((Spacecraft-specific R&D) + (Unit proc. cost))

Thus the total R&D cost consists of basic R&D cost for the spacecraft family and individual R&D costs for each spacecraft. To rewrite this equation in terms of (Unit proc. cost) alone, some partial breakdowns of individual HEAO and OAO program costs (Refs. D-11 and D-17) were examined. These figures suggest that:

(Basic R&D for spacecraft family) = (2.7) * (Unit proc. cost)

and

(Spacecraft-specific R&D) = (0.8) * (Unit proc. cost)

So the spacecraft family formula becomes:

(R&D and procurement of all spacecraft)

= ((2.7) * (Unit proc. cost))

+ (Number of s/c)
* ((0.8 * Unit proc. cost) + (Unit proc. cost))

Note that for a family of one spacecraft, the R&D-to-procurement ratio of 3.5 is maintained.

The costs predicted by this formula are within 15% (above and below) of the available OAO costs, and within 35% (above and below) of the quoted HEAO costs. Since these published estimates are only partial (not all OAO spacecraft are broken down) and very few details on the figures are provided, this agreement is considered acceptable. Also, because these OAO and HEAO figures are so fragmentary, the formula above is also applied to these families to generate estimated cost breakdowns.

The above-derived formulas are applied to the cost figures in Table D.10 to generate the cost components shown in Table D.11. (The Shuttle Orbiter and Spacelab are handled differently, as described above.) The figures in Table D.11 are the basic R&D cost for spacecraft families, the R&D cost specific to each spacecraft, and the procurement cost for each flight unit. For individual spacecraft or groups of identical spacecraft, some of these categories do not apply.

To produce dollar/kilogram R&D factors for the spacecraft, the R&D costs in Table D.11 are divided by the spacecraft masses in Table D.10. For the spacecraft families, the (basic R&D for s/c family) and (spacecraft-specific R&D) terms are first added together. This total represents the R&D cost that would be incurred if a single spacecraft of that family were produced "from scratch", i.e. without benefit of any prior R&D.

The dollar/kilogram procurement factors are produced by dividing the (unit procurement cost) terms in Table D.11 by the spacecraft masses in Table D.10. The resulting estimated cost factors for the past spacecraft are shown in Table D.12.

It should be noted that in most cases the R&D cost factors are 3.5 times the procurement cost factors, within roundoff error. This is expected, since that ratio is maintained in all the formulas which separate the total costs into R&D and procurement components. (The exceptions are the Shuttle Orbiter and Spacelab, for which detailed data is available.) Therefore the relationship between R&D cost and procurement cost is somewhat artificial. However, within each category (R&D or procurement), the cost factors are independent of each other, and thus make up a statistical data base.

A more general comment on this R&D-to-procurement ratio of 3.5 is that it is considerably lower than its counterpart (typically 30) in the transport aircraft industry (Ref. D-18). This is because a transport aircraft is developed with the expectation of production of at least 100 units. Therefore a substantial part of the R&D effort is the development of fabrication tooling and techniques to reduce the first-unit cost of the production articles. This results in a higher R&D cost but a lower unit procurement cost, and therefore a very high R&D-to-procurement ratio.

By comparison, the spacecraft listed in the tables above were developed with no expectation of mass-production. In fact, the number of flight units of a single type is so low that they are almost part of the development process: in a family of spacecraft, later models are often improved versions of the early designs.

TABLE D.11: ESTIMATED COST BREAKDOWNS FOR PAST SPACECRAFT
($MILLION; 1984$)

Spacecraft	Estimated basic R&D for s/c family	Estimated spacecraft-specific R&D	Estimated unit procurement cost
HEAO	131.7	39.0	48.8
OAO	337.1	99.9	124.8
Landsat 1&2	165.0	48.9	61.1
Landsat 3	N/A	59.0	16.9
Landsat 4	257.2	76.2	95.3
Solar Max	N/A	88.1	25.2
Relay	47.4	N/A	13.5
Voyager	299.3	N/A	85.5
Viking	1149	N/A	328.3
OGO	160.1	47.4	59.3
OSO-1	N/A	14.6	4.2
OSO-2	27.7	N/A	7.9
OSO-3	N/A	15.9	4.6
OSO-4	N/A	40.6	11.6
OSO-5	N/A	32.4	9.3
OSO-6	N/A	33.1	9.4
OSO-7	N/A	63.8	18.2
OSO-8	N/A	93.5	26.7
TIROS-N	N/A	67.7	19.3
Shuttle Orb.	6125	N/A	2092
Spacelab	1004	N/A	235.3

Notes: s/c = spacecraft

N/A = not applicable

Unit procurement costs for Shuttle Orb. and Spacelab are first-unit costs (90% learning curve)

TABLE D.12: ESTIMATED R&D AND PROCUREMENT COST FACTORS FOR PAST SPACECRAFT (1984$)

Spacecraft	Estimated R&D cost factor ($/kg)	Estimated procurement cost factor ($/kg)
HEAO	60,000	17,000
OAO	216,000	62,000
Landsat 1&2	226,000	65,000
Landsat 3	66,000	19,000
Landsat 4	172,000	49,000
Solar Max	38,000	11,000
Relay	608,000	173,000
Voyager	363,000	104,000
Viking	327,000	93,000
OGO	374,000	107,000
OSO-1	73,000	21,000
OSO-2	112,000	32,000
OSO-3	56,000	16,000
OSO-4	149,000	43,000
OSO-5	111,000	32,000
OSO-6	114,000	32,000
OSO-7	100,000	29,000
OSO-8	88,000	25,000
TIROS-N	94,000	27,000
Shuttle Orb.	91,000	31,000
Spacelab	109,000	26,000

Note: Figures are rounded to nearest thousand.

Fortunately, in the radiometer construction scenario, it is expected that the number of units of most types of hardware (e.g. habitation modules, teleoperators, beam-builders) will be low (e.g. less than 10). Therefore the R&D-to-procurement ratio developed in this section should be sufficiently realistic to model the hardware's cost behavior. If the cost model is applied to a larger scenario (e.g. the assembly of a solar power satellite), the hardware cost model should be reconsidered.

Another exception is the space structure to be assembled, which has enough repeated components to warrant mass-production. To compensate, the procurement of space structure is assigned a fairly steep learning curve (80%). In any case, since the procurement cost of the space structure appears as a constant in the overall cost equation, any inaccuracy in its value will not affect this study's human and machine tradeoffs.

The R&D cost factors from Table D.12 are plotted in Figure D.10. Also shown is a straight line fitted to the data by least-squares linear regression. The data points for the Shuttle Orbiter and Spacelab are not plotted, because they fall far to the right of the other points; but their $/kg levels are indicated, and they are included in the regression fit for the line.

The use of a constant dollar/kilogram R&D cost factor in the cost model is supported by the flatness of this line and the near-zero value (-0.14) of the correlation coefficient. However, the low magnitude of this coefficient is also due to the large scatter in the data. This scatter is due to a variety of causes, including:

1) Levels of complexity of spacecraft: the more complex the components, the more expensive they are to develop. Two spacecraft can have similar masses but different levels of complexity, and therefore significantly different R&D costs. Examples of complex spacecraft are OAO, OGO, and Viking.

2) Maturity of spacecraft technology: some spacecraft benefit from earlier R&D of related spacecraft, and of improvements in the R&D process in general. Conversely, spacecraft which require new technology are more expensive (e.g. Relay, the first active communications satellite).

3) Spacecraft reliability requirements: some spacecraft are made more expensive by stringent reliability requirements, for long-term missions or difficult assignments. Planetary probes (Voyager, Viking) are examples of this.

4) Stochastic effects, e.g. failures: on-orbit failures of spacecraft can require extensive development efforts to correct the problems before further launches. Such extra costs were incurred by the OAO family and the two Landsat 4 spacecraft.

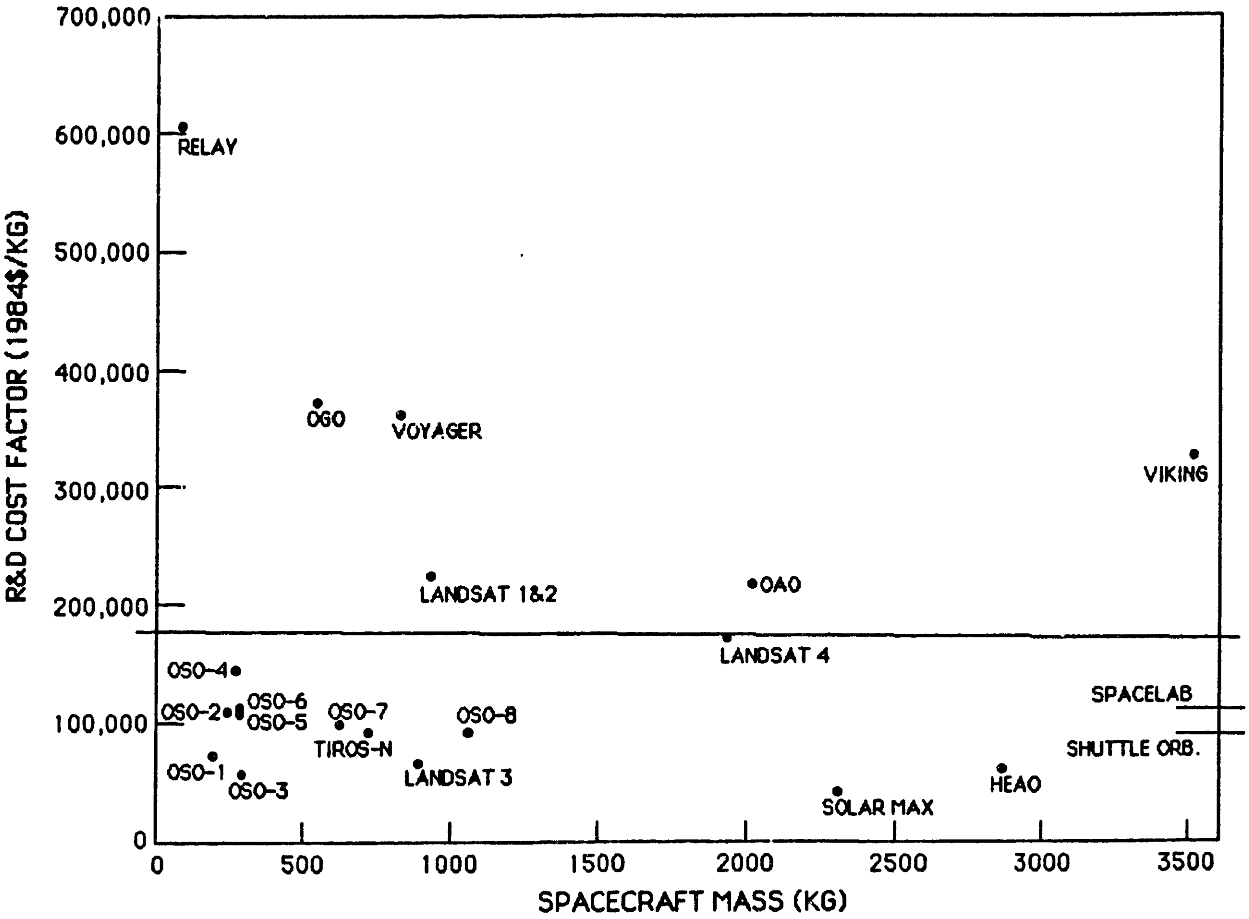


FIGURE D.10: R&D COST FACTORS FOR PAST SPACECRAFT

Some of these effects have been included in spacecraft cost models, with varying degrees of success (Ref. D-19). The complexity levels of spacecraft components have been incorporated into detailed cost models by Werner Gruhl of NASA Headquarters. Adjustments for technological maturity (sometimes called "technology carryover" or "inheritance") are included in models by Gruhl, SAI, and the Air Force SAMSO. The SAI model is for planetary probes, and therefore also reflects their high reliability requirements.

The development of a detailed cost prediction method for space hardware is outside the scope of this study. The principal focus of this study is the development of optimization methods to be applied to the cost model. Therefore, for this study's illustrative case example, a simple linear-with-mass model for space hardware costs is appropriate. As will be seen in Chapter 5, the methods developed in this study can be applied to a variety of nonlinear cost models, which could include the ones mentioned above.

The procurement cost factors from Table D.12 are plotted in Figure D.11, with a linear regression line. Based on the data in Figures D.10 and D.11, the R&D and procurement cost factors for high-technology hardware are defined as ranges of values:

DKrdH = Dollars of R&D cost per kilogram of high-tech hardware
= a range: low value = 50,000 $/kg
nominal value = 130,000 $/kg
high value = 320,000 $/kg

DKprH = Dollars of procurement cost per kilogram of high-tech hardware
= a range: low value = 15,000 $/kg
nominal value = 40,000 $/kg
high value = 90,000 $/kg

Because this study's example hardware is expected to require less technological advance than Relay, and less complexity and reliability than Voyager and Viking, the nominal values of the cost factors are slightly lower than the regression lines.

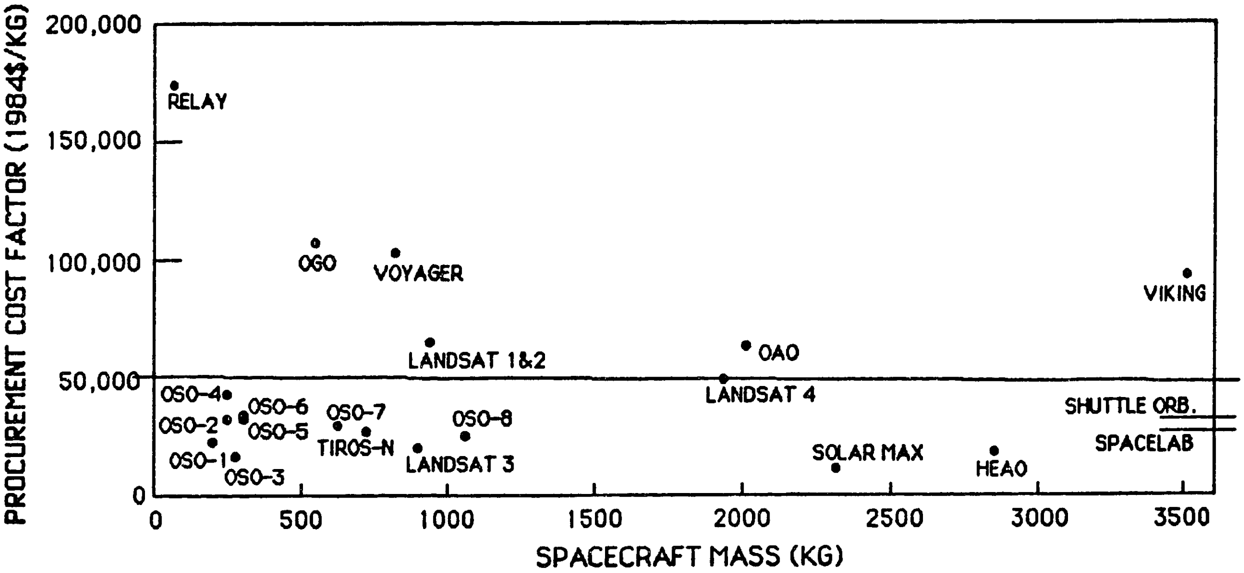


FIGURE D.11: PROCUREMENT COST FACTORS FOR PAST SPACECRAFT

D.2.2 Cost Factors for Low-Technology Hardware:

Low-technology hardware is defined as bulk structure, such as erectable trusswork, construction jigs, and pressure shells and airlock structures for habitation modules. Such hardware has almost never been launched, but two examples are available for general cost estimates.

The Shuttle External Tank was developed at a cost (in 1984$) of 1033.8 $M (Ref. D-12). This includes the procurement of four flight units for Shuttle flights STS-1 through STS-4. Assuming once again that:

$$\frac{\text{(R\&D cost)}}{\text{(Unit proc. cost)}} = 3.5$$

then the estimated R&D cost is 482.4 $M and the estimated unit procurement cost is 137.8 $M (both 1984$). Typical mass for the External Tank is 30,385 kg (Ref. D-14). This suggests an R&D cost factor of 15,878 $/kg and a procurement cost factor of 4536 $/kg, roughly one-tenth of the nominal values for high-tech hardware.

The other example is the flight hardware for the Experimental Assembly of Structures in EVA (EASE). This consists of an erectable tetrahedral structure, designed and built by the MIT Space Systems Laboratory under the supervision of the NASA Marshall Space Flight Center. The mass of the flight hardware is 313 kg, and the R&D and procurement costs incurred by MIT total 0.83 $M (1984$; Ref. D-20).

Doubling this figure to account for NASA Marshall's supervision and review activities, the total EASE hardware cost is estimated at 1.7 $M. Assuming an R&D-to-procurement ratio of 3.5, the R&D cost is 1.29 $M and the procurement cost is 0.37 $M. This corresponds to an R&D cost factor of 4121 $/kg and a procurement cost factor of 1182 $/kg. These values are less than one-tenth of the low values estimated for high-tech hardware, reflecting the cost-effectiveness of flight hardware built by students with academic salaries.

Based on these examples, the cost factors for low-technology hardware are estimared to be roughly one-tenth of the equivalent factors for high-technology hardware. Therefore:

DKrdL = Dollars of R&D cost per kilogram of low-tech hardware
= a range: low value = 5000 $/kg
nominal value = 13,000 $/kg
high value = 32,000 $/kg

DKprL = Dollars of proc. cost per kilogram of low-tech hardware
= a range: low value = 1500 $/kg
nominal value = 4000 $/kg
high value = 9000 $/kg

APPENDIX E:

LINE-ITEM COST MODEL: INPUTS TO MODEL

This appendix presents the values of inputs to this study's line-item cost model. These values are used for illustrative examples in Chapters 2 and 3, and in the study's case examples of space construction in Chapter 4. Some of these values are developed in Appendices C and D, on hardware designs. Other values are developed in this appendix.

It takes 81 inputs to define a space construction scenario for the cost model: 74 single-number inputs and 7 arrays. They are grouped into ten categories and listed in Table E.1. The following sections describe the inputs and their values, and discuss their application to other scenarios besides space construction.

TABLE E.1: LIST OF INPUTS TO THE LINE-ITEM COST MODEL

Description of tasks to be done in space:

NTSK (#) = Number of different tasks to be done
NREPS[j] (repetitions) = Number of repetitions of task j to be done
Nhum[j] (pers) = Number of humans assigned to task j
OHhum[j] (reps/pers-hr) = Productivity of humans in performing task j
Ntel[j] (tels) = Number of teleoperators assigned to task j
OHtel[j] (reps/tel-hr) = Productivity of teleoperators in performing task j
Naeq[j] (aeqs) = Number of units of automated equipment assigned to task j
OHaeq[j] (reps/aeq-hr) = Productivity of automated equipment in performing task j

Description of space structure to be constructed:

NunSSTh (units) = Number of units of high-tech space structure
KGunSSTh (kg/unit) = Mass of one unit of high-tech space structure
DensSSTh (kg/cubic meter) = Packing density of high-tech space structure
NunSSTl (units) = Number of units of low-tech space structure
KGunSSTl (kg/unit) = Mass of one unit of low-tech space structure
DensSSTl (kg/cubic meter) = Packing density of low-tech space structure
lcpmSST (%) = Learning curve parameter for space structure

TABLE E.1 cont.: LIST OF INPUTS TO THE LINE-ITEM COST MODEL

Research and development parameters:

DKrdH ($/kg) = Dollars of R&D cost per kg of high-tech space hardware

DKrdL ($/kg) = Dollars of R&D cost per kg of low-tech space hardware

CTrdTSK ($) = Total cost of R&D of task procedures

Procurement parameters:

DKprH ($/kg) = Dollars of proc. cost per kg of high-tech space hardware

DKprL ($/kg) = Dollars of proc. cost per kg of low-tech space hardware

Mission preparation and launch parameters:

DKmpH ($/kg) = Dollars of mission prep. cost per kg of high-tech hardware

DKmpL ($/kg) = Dollars of mission prep. cost per kg of low-tech hardware

DKla ($/kg) = Dollars to launch one kilogram on STS

DMla ($/cubic meter) = Dollars to launch one cubic meter on STS

DPla ($/pers-launch) = Dollars to launch one human

FRlaINS ($/$) = Fraction of proc. and launch cost paid as insurance premium

Ground support parameters:

General worksite support:

CTrdGMS ($) = Total cost of R&D of general monitoring stations

NgmsGEN (gms) = Number of general monitoring stations for worksite support

WFgms (gms/gms-hr) = Wearout factor for general monitoring stations

DUprGMS ($/gms) = Proc. cost for one general monitoring station

NgscGMS (gsc/gms) = No. of GSC people to operate one general monitoring station

Support for humans:

CTrdHMS ($) = Total cost of R&D of human monitoring stations

NhmsHUM (hms/pers) = Number of HMS to support one human in space

WFhms (hms/hms-hr) = Wearout factor for human monitoring stations

DUprHMS ($/hms) = Proc. cost for one human monitoring station

NgscHMS (gsc/hms) = No. of GSC people to operate one human monitoring station

TABLE E.1 cont.: LIST OF INPUTS TO THE LINE-ITEM COST MODEL

Support for teleoperators:
CTrdTCS ($) = Total cost of R&D of teleop. control stations
NtcsTEL (tcs/tel) = Number of TCS to support one teleop. in space
WFtcs (tcs/tcs-hr) = Wearout factor for teleop. control stations
DUprTCS ($/tcs) = Proc. cost for one teleoperator control station
NgscTCS (gsc/tcs) = Number of GSC people to operate one teleop. control station

Support for automated equipment:
CTrdAMS ($) = Total cost of R&D of auto. equipment monitoring stations
NamsAEQ (ams/aeq) = Number of AMS to support one unit of auto. equipment in space
WFams (ams/ams-hr) = Wearout factor for auto. equip. monitoring stations
DUprAMS ($/ams) = Proc. cost for one auto. equip. monitoring station
NgscAMS (gsc/ams) = No. of GSC people to operate one auto. equip. monitoring station

Miscellaneous:
DpGSCt ($/gsc) = Dollars/person for ground support crew training
DphGSC ($/gsc-hr) = Dollars/hour salary and overhead for one ground support crewmember

Description of humans in space:

Habitation modules:
NhumHAB (pers/hab) = Number of humans in one habitation module
KGunHABh (kg/hab) = Mass of high-tech hardware in one hab. module
KGunHABl (kg/hab) = Mass of low-tech hardware in one hab. module
WFhab (hab/hab-day) = Fraction of one hab. module worn out in one day
KGpdHABc (kg/pers-day) = Kilograms/day of consumables consumed by one human in space

Extravehicular mobility units:
CTrdEMU ($) = Total cost of R&D of blue-collar pressure suit
KGunEMU (kg/emu) = Mass of one pressure suit (incl. support hardware)
WFemu (emu/emu-hr) = Fraction of 1 pressure suit worn out in one hour
KGehEMUc (kg/emu-hr) = Kilos/hour of consumables for 1 pressure suit

Manned maneuvering units:
CTrdMMU ($) = Total cost of R&D for manned maneuvering unit
KGunMMU (kg/mmu) = Mass of one MMU (incl. support hardware)
WFmmu (mmu/mmu-hr) = Fraction of one manned maneuvering unit worn out in one hour
KGmhMMUc (kg/mmu-hr) = Kilograms/hour of consumables for one manned maneuvering unit

TABLE E.1 cont.: LIST OF INPUTS TO THE LINE-ITEM COST MODEL

Miscellaneous tools and equipment:
KGunEQP (kg/eqp) = Mass of one unit of misc. tools and equipment
WFeqp (eqp/eqp-hr) = Fraction of 1 unit of EQP worn out in one hour

Other human parameters:
HDhum (hrs/day) = Hours/day worked by humans in space
DpHUMt ($/pers) = Dollars/person for training of humans in space crews
DpdHUM ($/pers-day) = Dollars/day salary and overhead for one human in space
NdHUMt (days) = Number of days of training for humans in space crews
NDstint (days/stint) = Number of days in one human tour of duty in space

Description of teleoperators in space:

HDtel (hrs/day) = Hours/day worked by teleoperators in space
KGunTEL (kg/tel) = Mass of one free-flying teleoperator (incl. support equip.)
WFtel (tel/tel-hr) = Fraction of 1 teleoperator worn out in 1 hour
KGthTELc (kg/tel-hr) = Kilograms/hour of consumables consumed by one teleoperator

Description of automated equipment in space:

HDaeq (hrs/day) = Hours/day worked by automated equipment in space
KGunAEQ (kg/aeq) = Mass of one unit of automated equipment (incl. support equip.)
WFaeq (aeq/aeq-hr) = Fraction of 1 unit of AEQ worn out in 1 hour

Miscellaneous inputs:

Construction support equipment:
NtotCSE (units) = Number of units of construction support equipment at the worksite
KGunCSE (kg/unit) = Mass of one unit of construction support equip.

Construction site propellant:
KDcsp (kg/day) = Kilograms/day of construction site propellant requirement

Communications:
DlhTDR ($/lcc-hr) = Dollars/hour for one low-capacity TDR channel
DhhTDR ($/hcc-hr) = Dollars/hour for one high-capacity TDR channel

Learning curves:
lcparam (%) = Learning curve parameter for hardware other than SST

E.1 Inputs Describing Tasks to be Done in Space:

The cost model defines the job to be done as a set of concurrent tasks, which are described by eight inputs to the model. The first is NTSK, the number of different tasks to be performed. The remaining seven are arrays of values: each array contains as many elements as there are tasks to be done. The array NREPS[j] indicates how many repetitions of each task must be performed. It is this inpt which scales the overall job.

Three of the arrays (Nhum[j], Ntel[j], Naeq[j]) indicate the numbers of humans, teleoperators, and units of automated equipment assigned to each task. It is important to note that in this line-item cost model these values are explicit inputs; as discussed in Chapter 3, in this study's optimization method these quantities are variables.

The remaining three arrays (OHhum[j], OHtel[j], OHaeq[j]) are the productivities of humans, teleoperators, and automated equipment in doing each task, e.g. in repetitions/person-hour for humans. Because their values are difficult to predict for future equipment, it is expected that they will be systematically varied as parameters of the model, to explore their influences on the optimal choices of humans and machines.

It should also be noted that these task-related model inputs are not specific to space construction. The cost model deals in tasks, and these tasks could be construction, repair, research, or any other type of space tasks. The cost model can deal with any job which can be defined as a set of NTSK tasks, with each task to be done NREPS times by Nhum humans, Ntel teleoperators, and Naeq units of automated equipment, with productivities OHhum, OHtel, OHaeq.

NTSK = Number of different tasks to be done
= 4
(Four-task case example defined in Appendix C, Table C.7)

NREPS[j] = Number of repetitions of task j to be done
NREPS[1] = 16,688 reps
NREPS[2] = 3,728 reps
NREPS[3] = 600 reps
NREPS[4] = 600 reps
(From Appendix C, Table C.7)

Nhum[j] = Number of humans assigned to task j
Ntel[j] = Number of teleoperators assigned to task j
Naeq[j] = Number of units of automated equipment assigned to task j

These values are arbitrary, i.e. assigned by the Program Manager. For this case example, illustrative values are presented in Table E.2.

TABLE E.2: NUMBERS OF HUMANS AND MACHINES ASSIGNED TO EACH TASK

Task number [j]	Nhum[j] (pers)	Ntel[j] (tel)	Naeq[j] (aeq)
1	0.6	0.4	4.3
2	0.5	0.4	0.3
3	1.2	1.0	0.2
4	0.4	0.3	0.0

OHhum[j] = Productivity of humans in performing task j
(From Appendix C, section C.4)

OHhum[1] = a range: low value = 0.9 reps/pers-hr
nominal value = 1.3 reps/pers-hr
high value = 1.8 reps/pers-hr

OHhum[2] = a range: low value = 1.3 reps/pers-hr
nominal value = 3.8 reps/pers-hr
high value = 5.7 reps/pers-hr

OHhum[3] = a range: low value = 0.17 reps/pers-hr
nominal value = 0.33 reps/pers-hr
high value = 1.0 reps/pers-hr

OHhum[4] = a range: low value = 3.1 reps/pers-hr
nominal value = 5.6 reps/pers-hr
high value = 9.9 reps/pers-hr

OHtel[j] = Productivity of teleoperators in performing task j
(From Appendix C, section C.4)

OHtel[1] = a range: low value = 0.3 reps/tel-hr
nominal value = 1.0 reps/tel-hr
high value = 1.8 reps/tel-hr

OHtel[2] = a range: low value = 0.4 reps/tel-hr
nominal value = 2.7 reps/tel-hr
high value = 5.7 reps/tel-hr

OHtel[3] = a range: low value = 0.06 reps/tel-hr
nominal value = 0.13 reps/tel-hr
high value = 1.0 reps/tel-hr

OHtel[4] = a range: low value = 1.0 reps/tel-hr
nominal value = 4.8 reps/tel-hr
high value = 9.9 reps/tel-hr

OHaeq[j] = Productivity of automated equipment in doing task j
(From Appendix C, section C.4)

OHaeq[1] = a range: low value = 2.6 reps/aeq-hr
nominal value = 3.7 reps/aeq-hr
high value = 4.8 reps/aeq-hr

OHaeq[2] = 0.001 reps/aeq-hr

OHaeq[3] = 0.001 reps/aeq-hr

OHaeq[4] = 0.001 reps/aeq-hr

E.2 Inputs Describing Space Structure to be Assembled:

These inputs describe the space structure to be assembled, modeling it as units of high-technology structure and of low-technology structure. The inputs are the numbers of units, the unit masses, the packing densities of hightech and low-tech structure, and the learning curve parameter used in the procurement of the structure. For the case example in this study, these quantities are adequate to describe the Radiometer to be constructed. However, other structures might require more inputs, e.g. to describe more types of hardware to be assembled.

In general, this category describes hardware to be procured and launched, but this is hardware being operated on: it does not include any hardware from the human-machine options themselves. Therefore, for a scenario other than space construction, this category would be replaced by a similar set of inputs.

For example, for the servicing and repair of spacecraft, these inputs would describe the replacement parts for the satellites. And for the operation of production facilities in zero-g, these inputs would describe the production facilities themselves, and the raw materials to feed them.

These inputs are from Appendix C, Table C.5:

NunSSTh = Number of units of high-tech space structure
= 600 units

KGunSSTh = Mass of one unit of high-tech space structure
= 136.15 kg/unit

DensSSTh = Packing density of high-tech space structure
= 300 kg/cubic meter

NunSSTl = Number of units of low-tech space structure
= 3804 units

KGunSST1 = Mass of one unit of low-tech space structure
= 15.17 kg/unit

DensSST1 = Packing density of low-tech space structure
= 200 kg/cubic meter

lcpmSST = Learning curve parameter for space structure
= 80%

E.3 Research and Development Parameters:

This category includes inputs for R&D cost calculations, such as dollar/kilogram factors for high-tech and low-tech hardware. These are significant parameters whose values are difficult to estimate for future technologies. Techniques to calculate the values of these inputs are discussed and applied in Appendix D. Also included in this category is the R&D cost to develop construction task procedures. These inputs would be included in any space scenario.

DKrdH = Dollars of R&D cost per kg of high-tech space hardware
= a range: low value = 50,000 $/kg
nominal value = 130,000 $/kg
high value = 320,000 $/kg
(From section D.2.1)

DKrdL = Dollars of R&D cost per kg of low-tech space hardware
= a range: low value = 5000 $/kg
nominal value = 13,000 $/kg
high value = 32,000 $/kg
(From section D.2.2)

CTrdTSK = Total cost of R&D of task procedures
= 1,270,000 $
(Estimate by D.G. Stuart, based on 150 hours in simulators equivalent to Shuttle Mission Simulator (6000 $/hr; Ref. E-1), and 100 hours of neutral buoyancy simulations (3700 $/hr; Ref. E-2).)

E.4 Procurement parameters:

The inputs in the procurement category consist of dollar/kilogram factors for high-tech and low-tech hardware. The estimation of their values is also discussed in Appendix D. Such inputs would appear in any scenario.

DKprH = Dollars of proc. cost per kilogram of high-tech hardware
= a range: low value = 15,000 $/kg
nominal value = 40,000 $/kg
high value = 90,000 $/kg
(From section D.2.1)

DKprL = Dollars of proc. cost per kilogram of low-tech hardware
= a range: low value = 1500 $/kg
nominal value = 4000 $/kg
high value = 9000 $/kg
(From section D.2.2)

E.5 Mission Preparation and Launch Parameters:

The mission preparation and launch parameters include:
a) dollar/kilogram factors for the mission preparation of hardware (i.e. payload integration and checkout prior to launch);
b) launch cost factors (in dollars/kilogram, dollars/cubic meter, and dollars/person);
c) the fraction of procurement and launch costs to be charged as launch insurance.
Such inputs would appear in any space activities scenario.

DKmpH = Dollars of mission prep. cost per kg of high-tech hardware
= 500 $/kg
(Based on mission preparation cost of 10-15 $M for 12,000-kg Spacelab 1 (Ref. E-3), which suggests 1042 $/kg; this figure is reduced because the Spacelab 1 integration and checkout was particularly long and difficult, and because mission prep. costs are expected to decrease with experience.)

DKmpL = Dollars of mission prep. cost per kg of low-tech hardware
= 170 $/kg
(Based on mission prep. costs (Ref. E-3) of 0.1 $M for 438-kg Remote Manipulator System (228 $/kg) and 1 $M for typical 4500-kg already-integrated communications satellite (222 $/kg); these figures are reduced because future mission prep. costs are expected to decrease.)

DKla = Dollars to launch one kilogram on STS
= a range: low value = 1730 $/kg
nominal value = 3500 $/kg
high value = 6180 $/kg
(Values based on candidate prices for Shuttle launches (Ref. E-4) and Shuttle payload of 27,281 kg for direct insertion to 370-km orbit (Ref. E-5). Low value corresponds to "short-run marginal cost" price of 42 $M/flight (1982$); nominal price is for "average operational cost" of 85 $M (1982$); high value is for "average total cost, including all capital costs" price of 150 $M (1982$). Values are adjusted to 1984$.)

DM1a = Dollars to launch one cubic meter on STS
= a range: low value = 160,000 $/cubic meter
nominal value = 324,000 $/cubic meter
high value = 572,000 $/cubic meter
(Same calculations as for DM1a, but with Shuttle payload = 295 cubic meters.)

DP1a = Dollars to launch one human on STS
= a range: low value = 155,700 $/pers-launch
nominal value = 315,000 $/pers-launch
high value = 556,200 $/pers-launch
(Based on a 90-kg human (including personal effects) times the values of DK1a.)

FR1aINS = Fraction of proc. and launch cost paid as insurance premium
= 0.05 $/$
(5% premium to low-earth-orbit aboard STS (Ref. E-6). Note: it's the LEO-to-GEO transfer that incurs large premiums in current spacecraft launches.)

E.6 Ground Support Parameters:

The ground support hardware consists of a variety of monitoring and control stations. For each type of ground station, the inputs include: the R&D cost for that type of station; a factor indicating how many stations are required; a wearout factor used to calculate spares requirements; the first-unit procurement cost for one station; and the number of ground support crewpeople to operate that type of station. Also in this category are cost factors for the training and salary of the ground support crew.

These ground support inputs would be included in any scenario which uses humans, teleoperators, and/or automated equipment in space. If other aspects of the space activities needed ground support, other ground stations and personnel could be added with similar inputs.

General worksite support:

CTrdGMS = Total cost of R&D of general monitoring stations
= 3,000,000 $
(Estimated relative to the more complex teleoperator control stations. Includes development of automated monitoring systems to assist human operator.)

NgmsGEN = Number of general monitoring stations for worksite support
= 6 gms
(Estimated from the ~15 stations currently in the Mission Operations Control Room for Shuttle missions (Ref. E-1); some of the Shuttle stations do not apply to this worksite, and other functions are combined into 6 general monitoring stations.)

WFgms = Wearout factor for general monitoring stations
= 0.00006 gms/gms-hr
(Based on an estimated 2-year lifetime in 24-hours-per-day, 7-days-a-week operation (17,520 hours).)

DUprGMS = Procurement cost for one general monitoring station
= 300,000 $/gms
(Estimated as one-tenth of GMS R&D cost.)

NgscGMS = No. of GSC people to operate one general monitoring station
= 3 gsc/gms
(1 person at station, 2 in back room. Based on current 1 at station, 3-to-4 in back room for Shuttle monitoring (Ref. E-1); number reduced for routine monitoring of worksite, with the assistance of some automated monitoring systems.)

Support for humans:

CTrdHMS = Total cost of R&D of human monitoring stations
= 2,000,000 $
(Similar to general monitoring stations, but with fewer functions. R&D includes development of some automated systems, e.g. for health monitoring.)

NhmsHUM = Number of HMS to support one human in space
= 1 hms/pers
(Assumption: HMS is designed to monitor 1 human.)

WFhms = Wearout factor for human monitoring stations
= 0.00009 hms/hms-hr
(Based on an estimated 4-year lifetime in 8-hours-per-day, 7-days-a-week operation (11,680 hours).)

DUprHMS = Procurement cost for one human monitoring station
= 200,000 $/hms
(Estimated as one-tenth of HMS R&D cost.)

NgscHMS = No. of GSC people to operate one human monitoring station
= 2 gsc/hms
(1 person at station, 1 in back room. Routine EVA using blue-collar EMU and MMU and partially automated HMS does not require many ground support personnel.)

Support for teleoperators:

CTrdTCS = Total cost of R&D of teleoperator control station
= 4,500,000 $
(Based on estimate of 5 $M to develop and procure 1 integrated teleoperator control station (Ref. E-2). R&D includes development of flexible human-machine interface, automated monitoring systems for teleoperator hardware, trouble-avoidance systems (e.g. force-setting controls, predictive displays).)

NtcsTEL = Number of TCS to support one teleoperator in space
= 1 tcs/tel
(Assumption: TCS is designed to control 1 teleoperator.)

WFtcs = Wearout factor for teleop. control stations
= 0.00009 tcs/tcs-hr
(Based on an estimated 2-year lifetime in 16-hours-per-day, 7-days-a-week operation (11,680 hours).)

DUprTCS = Procurement cost for one teleoperator control station
= 450,000 $/tcs
(Estimated as one-tenth of TCS R&D cost.)

NgscTCS = Number of GSC people to operate one teleoperator control station
= 3 gsc/tcs
(2 persons at station, 1 in back room. The persons at the station are 1 operator and 1 systems monitor. Monitoring a teleoperator is more complex than monitoring a human.)

Support for automated equipment:

CTrdAMS = Total cost of R&D of automated equipment monitoring station
= 2,000,000 $
(Similar to human monitoring station.)

NamsAEQ = Number of AMS to support one unit of automated equipment in space
= 0.5 ams/aeq
(For the relatively simple task of monitoring automated equipment, 1 station can keep track of 2 automated machines.)

WFams = Wearout factor for automated equipment monitoring stations
= 0.00007 ams/ams-hr
(Based on an estimated 2-year lifetime in 20-hours-per-day, 7-days-a-week operation (14,600 hours).)

DUprAMS = Procurement cost for one automated equipment monitoring station
= 200,000 $/ams
(Estimated as one-tenth of AMS R&D cost.)

NgscAMS = Number of GSC people to operate one automated equipment monitoring station
= 2 gsc/ams
(1 person at station, 1 in back room.)

Miscellaneous:

DpGSCt = Dollars/person for ground support crew training
= 42,500 $/gsc
(Estimated (four years of tuition at MIT); does not include person's salary during training, which is calculated separately. Also, some of this training is covered under the R&D of task procedures CTrdTSK.)

DphGSC = Dollars/hour salary and overhead for one ground support crewmember
= 35 $/gsc-hr
(Based on 35,000 $/year salary, 100% overhead; work year = 250 days x 8 hours/day.)

E.7 Inputs Describing Humans in Space:

This category describes the humans working in space, including their in-orbit support equipment. These inputs include the number of humans housed in each habitation module, masses of high-tech and low-tech hardware, wearout factors, and consumables factors for habitation modules and support equipment.

In addition, "other" inputs in this category include the number of hours per day worked by humans in space, cost factors for their training and salary, the number of days of training required, and the number of days in space in a human's tour of duty (called a "stint").

This category's inputs would be included in any space scenario which uses humans in extravehicular activity (EVA). For other human options, the needed inputs would probably be a subset of the EVA inputs. For example, intravehicular activity (IVA) would be described by the habitation module inputs, the miscellaneous equipment inputs, and the "other" inputs.

Habitation modules: (From section D.1.1)

NhumHAB = Number of humans in one habitation module
= 3 pers/hab

KGunHABh = Mass of high-tech hardware in one habitation module
= a range: low value = 7600 kg/hab
nominal value = 10,846 kg/hab
high value = 14,100 kg/hab

KGunHABl = Mass of low-tech hardware in one habitation module
= 7941 kg/hab

WFhab = Fraction of one habitation module worn out in one day
= a range: low value = 0.00006 hab/hab-day
nominal value = 0.0002 hab/hab-day
high value = 0.0004 hab/hab-day

KGpdHABc = Kilograms/day of consumables for one human in space
= 4.68 kg/pers-day

Extravehicular mobility units: (From section D.1.2)

CTrdEMU = Total cost of R&D of blue-collar pressure suit
= 30,000,000 $

KGunEMU = Mass of one Extravehicular Mobility Unit (including support equipment)
= 300 kg/emu

WFemu = Fraction of one EMU worn out in one hour
= 0.0002 emu/emu-hr

KGehEMUc = Kilograms/hour of consumables for one EMU
= 1.38 kg/emu-hr

Manned maneuvering units: (From section D.1.3)

CTrdMMU = Total cost of R&D for Manned Maneuvering Unit
= 20,000,000 $

KGunMMU = Mass of one Manned Maneuvering Unit (including support equipment)
= 355 kg

WFmmu = Fraction of one MMU worn out in one hour
= 0.0001 mmu/mmu-hr

KGmhMMUc = Kilograms/hour of consumables for one manned maneuvering unit
= 11.67 kg/mmu-hr

Miscellaneous tools and equipment:

KGunEQP = Mass of one unit of misc. tools and equipment
= 70 kg/eqp
(Modeled as 40 kg of misc. tools in a 15-kg toolbox (including attachment clamps), and a 15-kg portable foot restraint.)

WFeqp = Fraction of 1 unit of EQP worn out in one hour
= 0.00009 eqp/eqp-hr
(Based on an estimated 4-year lifetime in 8-hours-per-day, 7-days-a-week operation (11,680 hours).)

Other human parameters:

HDhum = Hours/day worked by humans in space
= 7 hours/day
(From section D.1.2)

DpHUMt = Dollars/person for training of humans in space crews
= 500,000 $/pers
(Estimated as four times the cost of training a Payload Specialist, since EVA is involved. Ref. E-7 suggests a training cost for Payload Specialist of ~125,000 $ (1984$).)

DpdHUM = Dollars/day salary and overhead for one human in space
= 320 $/pers-day
(Based on 40,000 $/year salary, 100% overhead; work year = 250 days.)

NdHUMt = Number of days of training for humans in space crews
= 730 days
(Estimated: 2 years of training.)

NDstint = Number of days in one human tour of duty in space
= 91 days/stint
(One-quarter of a year. Estimated to be a safe duration for a zero-g stay, based on Skylab and Salyut data.)

E.8 Inputs Describing Teleoperators in Space:

This category describes space teleoperators, including number of hours per day of operation, unit mass (including in-orbit support hardware), and wearout and consumables factors.

These inputs would be included in any scenario using teleoperators in space. If more than one type of teleoperator were used, similar sets of inputs would be added, one set for each version to be modeled.

These inputs are from section D.1.4:

HDtel = Hours/day worked by teleoperators in space
      = a range: low value = 8 hours/day
                 nominal value = 16 hours/day
                 high value = 21 hours/day

KGunTEL = Mass of one teleoperator (including support equipment)
        = a range: low value = 631 kg/tel
                   nominal value= 1261 kg/tel
                   high value = 1892 kg/tel

WFtel = Fraction of one teleoperator worn out in one hour of operation
      = a range: low value = 0.00002 tel/tel-hr
                 nominal value = 0.00003 tel/tel-hr
                 high value = 0.00009 tel/tel-hr

KGthTELc = Kilograms/hour of consumables for one teleoperator
         = 24.6 kg/tel-hr

E.9 Inputs Describing Automated Equipment in Space:

The automated equipment inputs include hours/day of operation, unit mass, and wearout factor. Automated equipment does not require consumables. Such inputs would appear in any scenario using automated equipment. Other types of automated equipment would be described by additional sets of these inputs.

These inputs are from section D.1.5:

HDaeq = Hours/day worked by automated equipment in space
      = a range: low value = 8 hours/day
                 nominal value = 20 hours/day
                 high value = 23 hours/day

KGunAEQ = Mass of one unit of automated equipment (including support equipment)
        = a range: low/nominal value = 2641 kg
                   high value = 3962 kg

WFaeq = Fraction of one unit of automated equipment worn out in one hour of operation
      = a range: low value = 0.00001 aeq/aeq-hr
                 nominal value = 0.00003 aeq/aeq-hr
                 high value = 0.00007 aeq/aeq-hr

E.10 Miscellaneous Inputs:

The final category contains miscellaneous inputs. The construction support equipment is general worksite equipment, not tied to any human or machine option; it is described by its number of units at the worksite, and its unit mass. Also in this category is the daily mass of construction site propellant required for orbital drag cancellation - the estimation of its value is described in Appendix D. The hourly costs for low-capacity and high-capacity communications channels are also listed. Finally, the learning curve parameter for procurement of multiple items of hardware (other than space structure) is also included.

The construction support equipment represents low-tech hardware needed at the worksite; other scenarios might well require such equipment, with similar inputs. The requirement for propellant, the costs for communications channels, and the learning curves in procurement would be inputs to any space scenario.

Construction support equipment:

NtotCSE = Number of units of construction support equipment at the worksite
= 1 unit
(Assumption: one batch of construction support equipment for the entire worksite.)

KGunCSE = Mass of one unit of construction support equipment
= 1330 kg/unit
(From section D.1.6)

Construction site propellant:

KDcsp = Kilograms/day of construction site propellant requirement
= a range: low value = 54.5 kg/day
nominal value = 109 kg/day
high value = 545 kg/day
(From section D.1.6)

Communications:

DlhTDR = Dollars/hour for one low-capacity TDR channel
= 2100 $/lcc-hr
(Based on commercial-user price for Tracking and Data Relay Satellite System: one multiple-access forward channel (up to 10 kilobits/sec; 27 $/min.) and one multiple-access return channel (up to 50 kilobits/sec; 8 $/min.) (Ref. E-8).)

DhhTDR = Dollars/hour for one high-capacity TDR channel
= 7200 $/hcc-hr
(Based on commercial-user price for Tracking and Data Relay Satellite System: one single-access channel (up to 25 megabits/sec forward; up to 300 megabits/sec return; 120 $/min.) (Ref. E-8).)

Learning curves:

lcparam = Learning curve parameter for hardware other than SST
= 90%
(Typical learning curve for small-lot space hardware production.)

APPENDIX F:

SENSITIVITY ANALYSES OF THE FOUR-TASK CASE EXAMPLE

This appendix presents the sensitivities of the case example's total program cost to variations in input values. Some of the line-item cost model's inputs are systematically varied through ranges of values, and the resulting changes in ProgCost are shown in the graphs below. The cost behaviors are used to identify the important cost-driving parameters in the four-task case example.

F.1 Effects of Task-Related Parameters:

The first parameters of interest are the productivities of the human-machine options in performing the tasks. In Chapter 4's line-item case example, Task 1 is performed by a mix of 0.6 humans, 0.4 teleoperators, and 4.3 units of automated equipment. The effect of each option's productivity in this task is shown in Figure F.1.

To produce each graph, only the associated productivity is varied, through the range estimated in section C.4. All other model inputs have the baseline values listed in Appendix E. The productivities are normalized by dividing them by their baseline values (e.g. OHhum[1] is divided by 1.3 reps/pers-hr). At those baseline values, the program cost has its baseline value of 5611 $million.

The three curves in Figure F.1 illustrate both the effect of productivity and the interaction between productivities. The typical effect of productivity appears in the bottom graph in the figure: as the automated equipment productivity increases, the total program costs decrease. In fact, all three curves show this effect, but the upper two graphs' variations are too small to show on this vertical scale.

The reason for the relative sizes of these effects is that the automated equipment performs the large majority of Task 1. Because of the automated equipment's larger assignment (4.3 units), its higher hourly productivity (3.7 reps/aeq-hr), and its longer daily work time (20 hrs/day), variations in its productivity have a larger impact on the overall cost than changes in human and teleoperator productivities.

Figure F.2 shows the effects of human and teleoperator productivities in Task 2. In this case, these two options perform almost all of the task. Their assignments (0.5 humans and 0.4 teleoperators) and nominal daily productivities (26.6 reps/pers-day and 43.2 reps/tel-day) give each option roughly half of the task to do. Therefore variations in either productivity lead to similar overall cost behaviors.

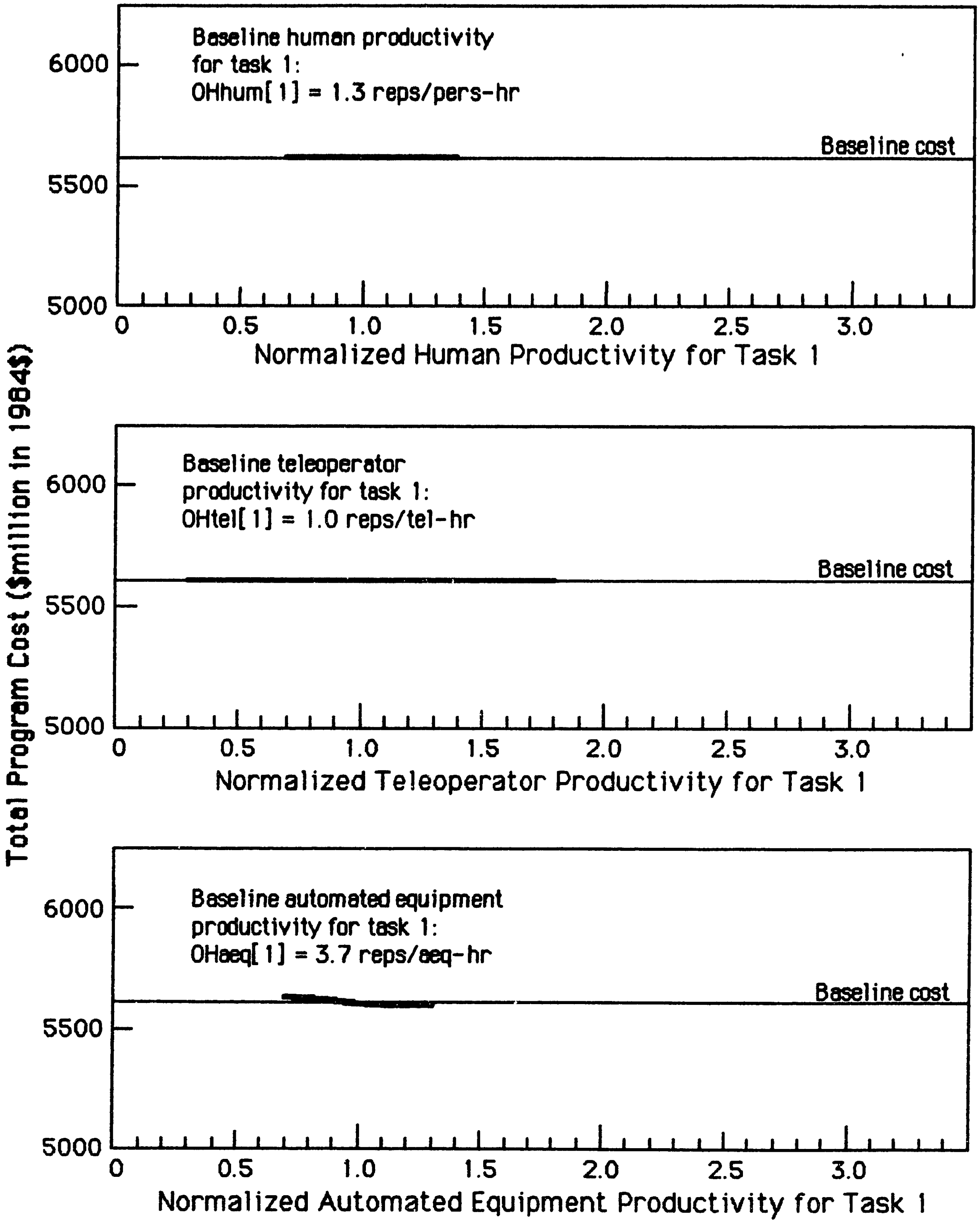


FIGURE F.1: EFFECT OF PRODUCTIVITIES IN TASK 1 ON CASE EXAMPLE'S OVERALL PROGRAM COST

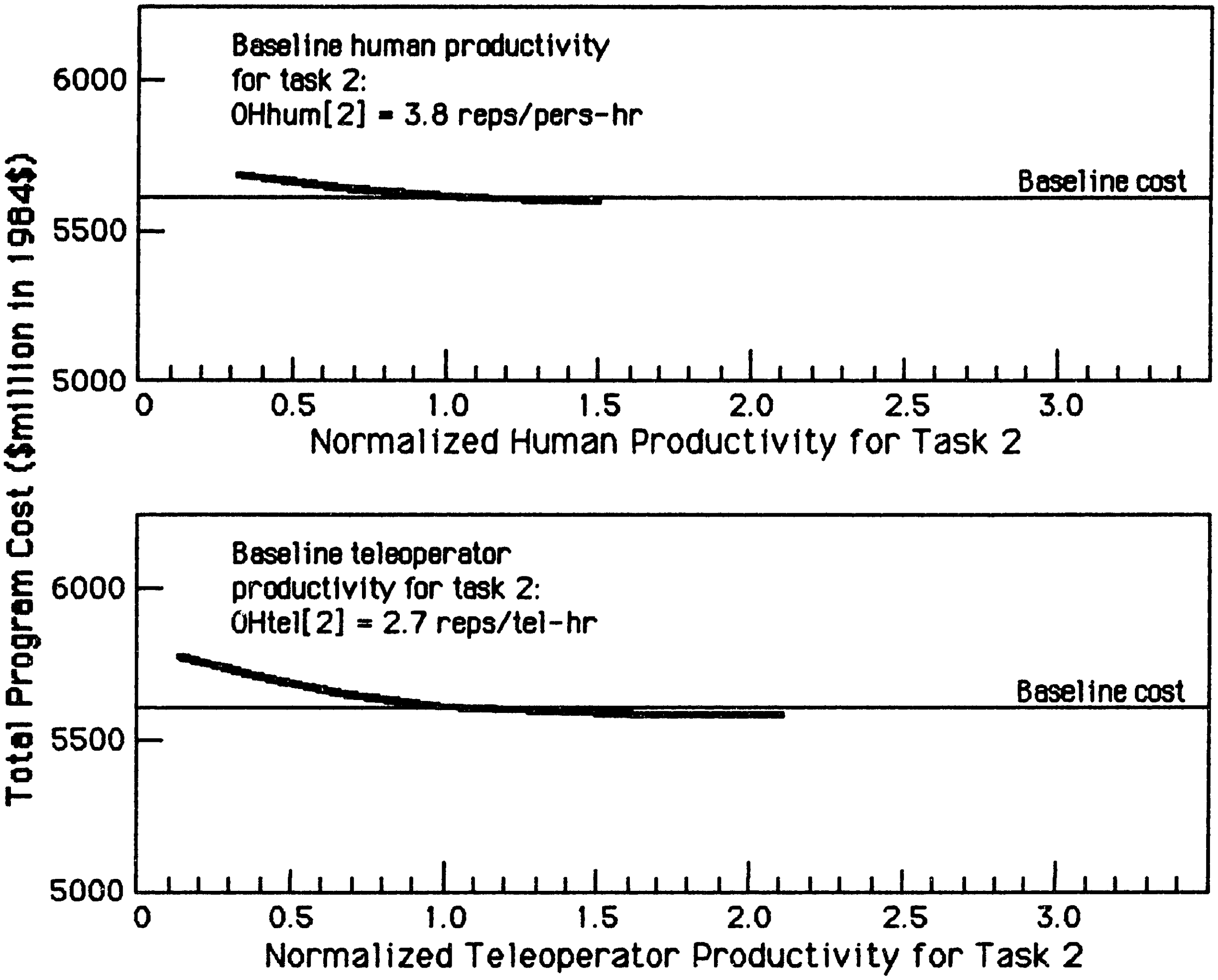


FIGURE F.2: EFFECT OF PRODUCTIVITIES IN TASK 2
ON CASE EXAMPLE'S OVERALL PROGRAM COST

The effects of the productivities in Task 3 are shown in Figure F.3. These effects are stronger than those in Figure F.2 because Task 3 is the longest task (as shown in Table 4.4) and thus sets the overall job time. Therefore variations in the duration of this task not only affect costs specific to the task, but also whole-job costs, such as the launch cost for construction site propellant.

Figure F.4 displays the effects of productivities in Task 4. Although they also show decreasing costs with increasing productivities, these variations are not very strong. This illustrates the indirect effect of task size: the 600 repetitions of Task 4 are less affected by changes in productivity than (for example) the 3728 repetitions of Task 2.

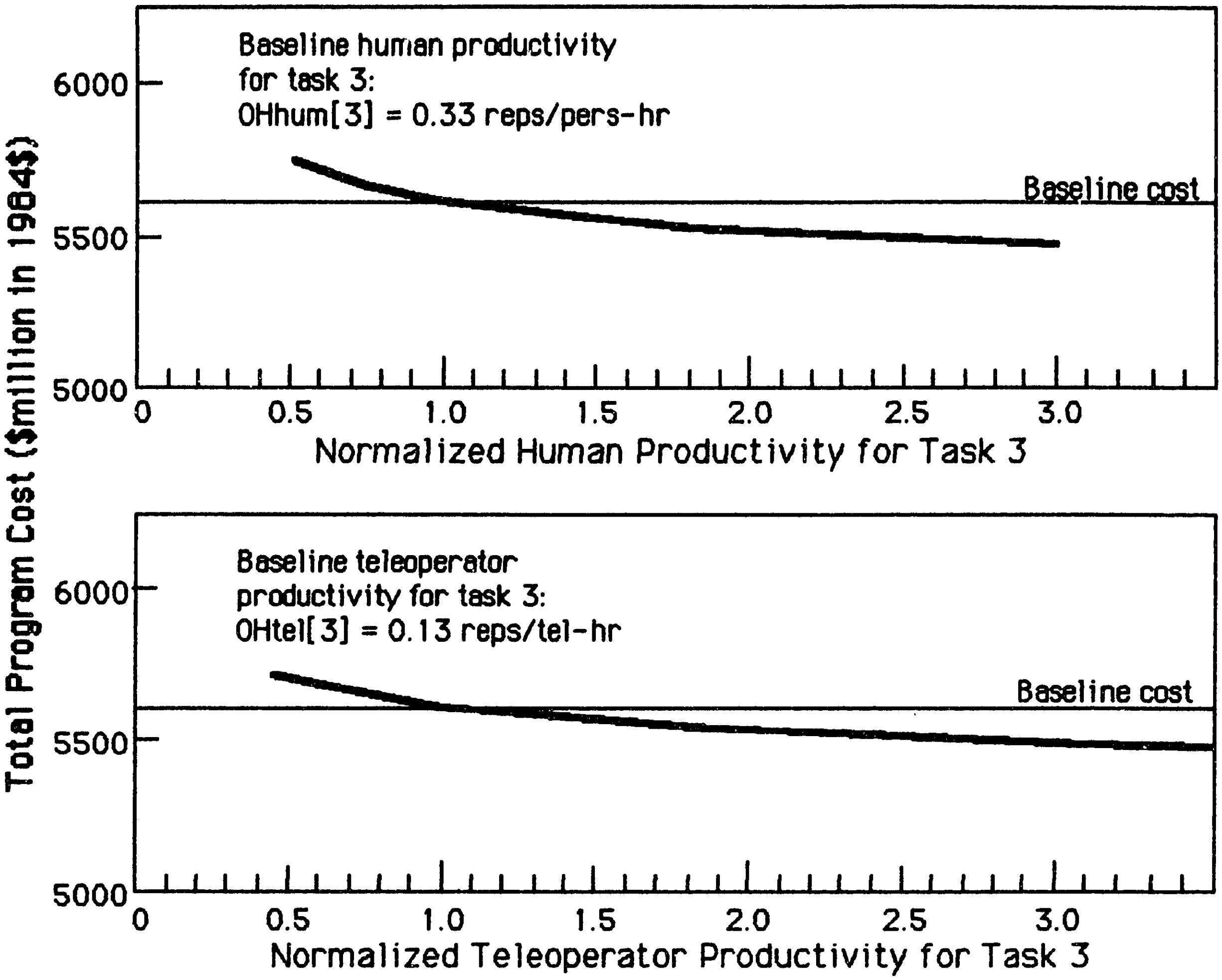


FIGURE F.3: EFFECT OF PRODUCTIVITIES IN TASK 3
ON CASE EXAMPLE'S OVERALL PROGRAM COST

Therefore the effect of increasing any single productivity is to decrease overall program costs. However, the size of that reduction depends in part on other parameters:

1) the numbers and productivities of other human-machine options assigned to the task;
2) the number of repetitions of the task to be performed;
3) whether the task is the longest task in the job.

In part, it is these interdependences that complicate the application of optimization methods to the overall cost equation (as discussed in Chapter 2). This study's Staged Optimization Method was developed to solve this problem while keeping these relationships intact.

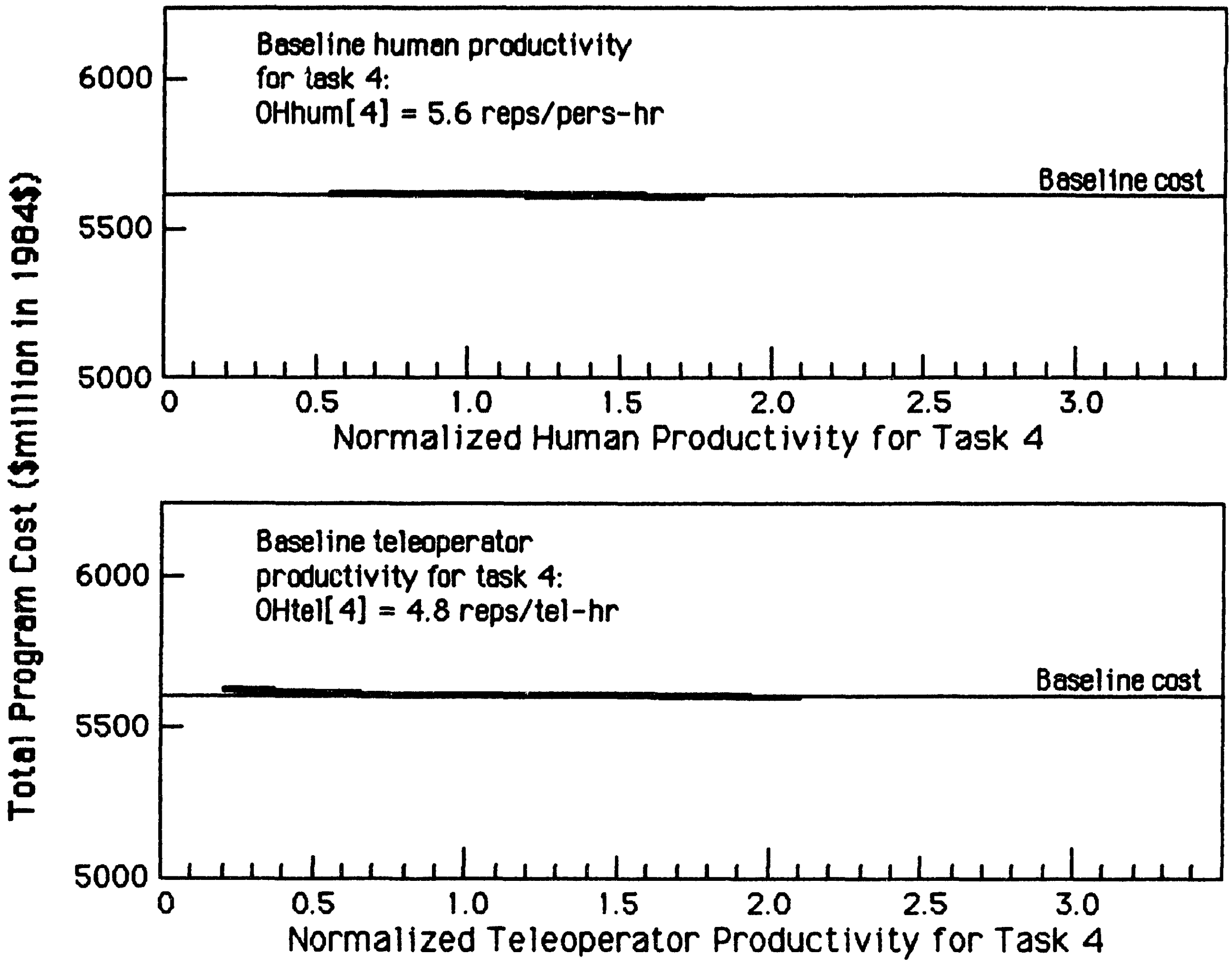


FIGURE F.4: EFFECT OF PRODUCTIVITIES IN TASK 4
ON CASE EXAMPLE'S OVERALL PROGRAM COST

Another parameter of interest is the overall scale of the job to be done, i.e. the number of Radiometers to be assembled. Although this is not an explicit input to the cost model, its effect can be modeled by adjusting the numbers of units of radiometer hardware and the numbers of repetitions of construction tasks, while keeping constant the R&D cost for the Radiometer. The results are shown in Figure F.5.

The marginal cost of building a second radiometer (the difference between 2-radiometer and 1-radiometer costs in Figure F.5) would be 1497 $million. This indicates that the case example's construction of one radiometer (for 5611 $M) includes a substantial up-front investment in R&D and procurement of work-site hardware, before the actual construction costs.

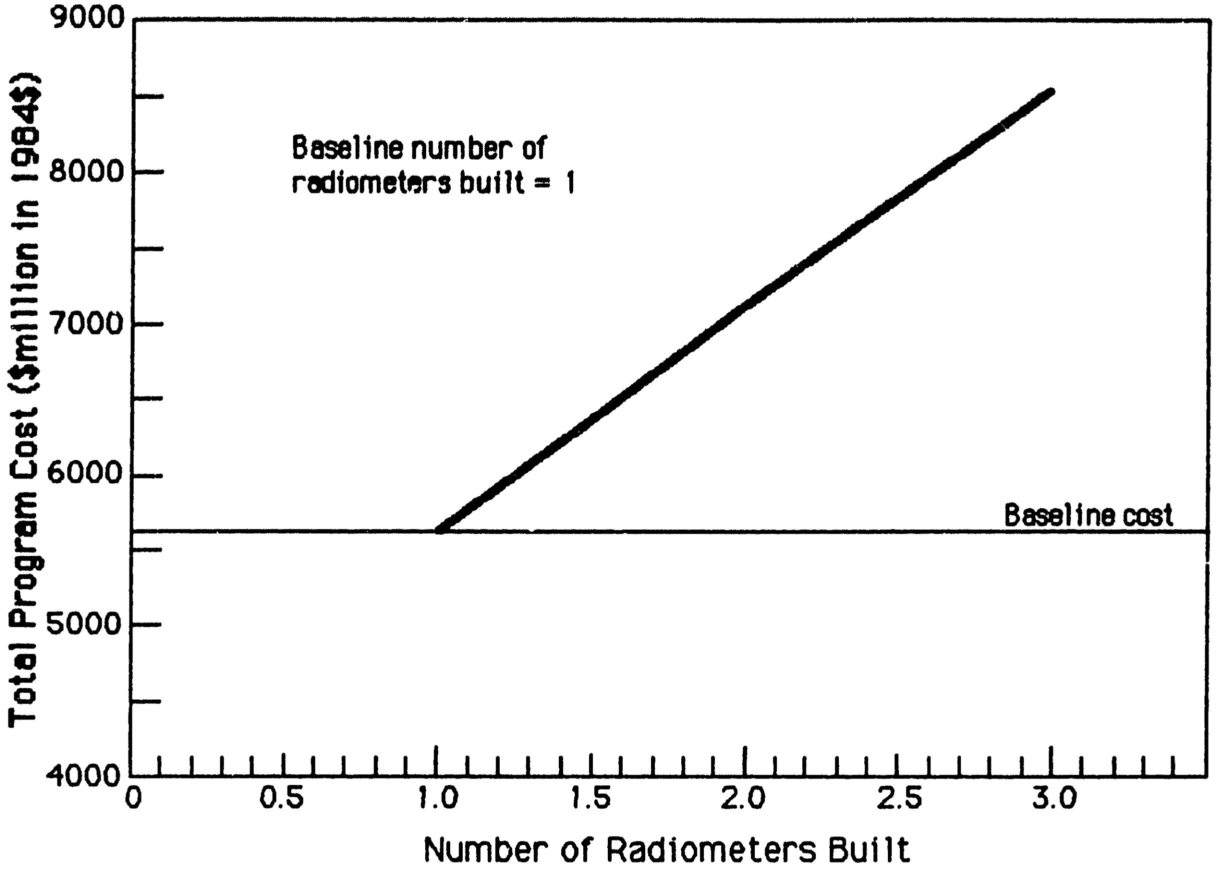


FIGURE F.5: EFFECT OF NUMBER OF RADIOMETERS BUILT ON CASE EXAMPLE'S OVERALL PROGRAM COST

The marginal cost of a third radiometer would be 1432 $M, slightly lower than the second radiometer because of learning curves in the procurement of larger amounts of hardware. Thus the line in Figure F.5 is actually slightly concave-downward.

F.2 Effects of Space-Structure-Related Parameter:

Figure F.6 presents the impact of the launch packing density of the Radiometer's low-tech structure (i.e. the trusswork components). If this structure is packed with a density higher than 92.5 kilogram/cubic meter, it is launched by mass, and the program stays at its baseline cost. Below the critical density, however, the structure is launched by volume, and the increased number of flights pushes the launch cost sharply upward.

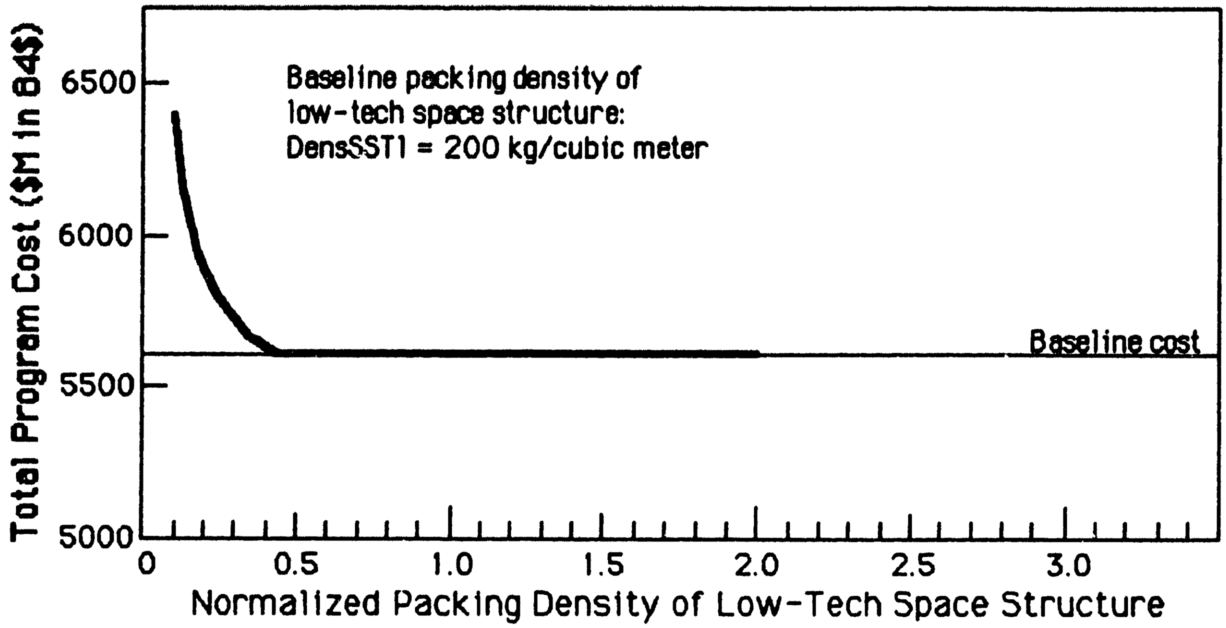


FIGURE F.6: EFFECT OF PACKING DENSITY OF LOW-TECH STRUCTURE ON CASE EXAMPLE'S OVERALL PROGRAM COST

One conclusion from this figure is that if deployable structures are used, their packing density should still be near or above 92.5 kg/cubic meter, or any improvements in assembly productivity may be overwhelmed by the increased launch costs.

F.3 Effects of Research and Development Parameters:

Figure F.7 presents the total program cost as a function of the dollar/kilogram factors for R&D of high-technology and low-technology hardware. As the figure shows, the R&D cost factor for high-tech hardware has a strong impact on total program cost. This is because a substantial fraction of the overall cost is for R&D of high-tech hardware (e.g. parts of the radiometer and habitation module, the teleoperator, and the automated equipment), and because the range of uncertainty in the R&D cost factor is large (the high estimate is more than 6 times the low estimate).

The low-tech R&D cost factor is also a cost-driver, but not as important as the high-tech R&D factor. (Note that the vertical scale in the upper graph is twice as compressed as in the lower graph.) The low-tech factor is not as significant because there is less low-tech hardware than high-tech hardware to be developed for this scenario.

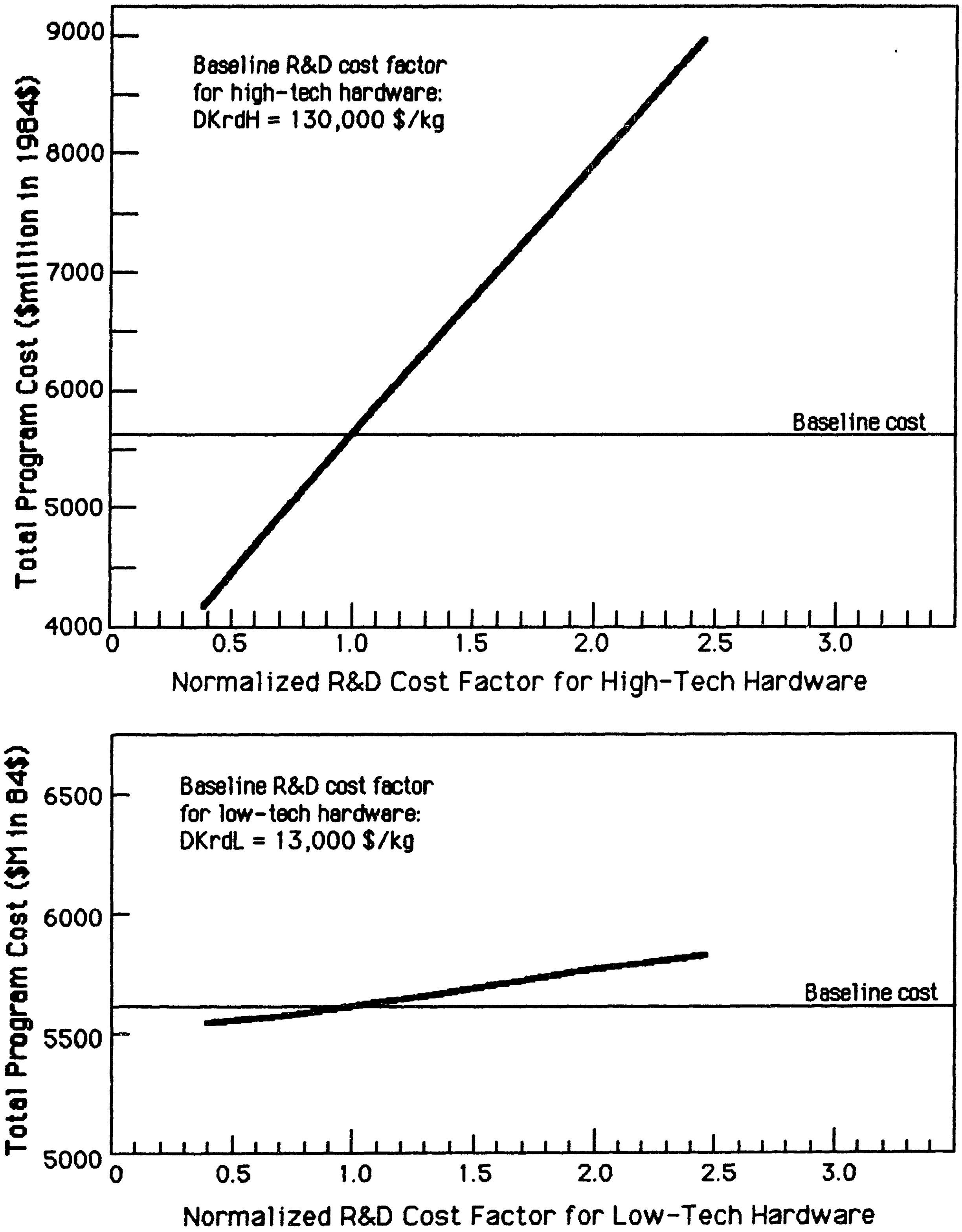


FIGURE F.7: EFFECTS OF R&D COST FACTORS ON CASE EXAMPLE'S OVERALL PROGRAM COST

F.4 Effects of Procurement Parameters:

Figure F.8 (on next page) shows the effects of the dollar/kilogram procurement cost factors for high-tech and low-tech hardware. (Note that the vertical scale in the upper graph is twice as compressed as in the lower graph.) Because this scenario involves mostly high-tech hardware, the high-tech procurement factor is a strong cost-driver; the low-tech factor is not very significant.

F.5 Effects of Launch Parameters:

Figure F.9 shows the effect of the dollar/kilogram cost to launch payload on the Space Shuttle. In the case example, launch costs are roughly one-fifth of the program cost, and almost all of these costs are for launching payload by mass. Therefore the per-kilogram launch price is a strong cost-driving parameter.

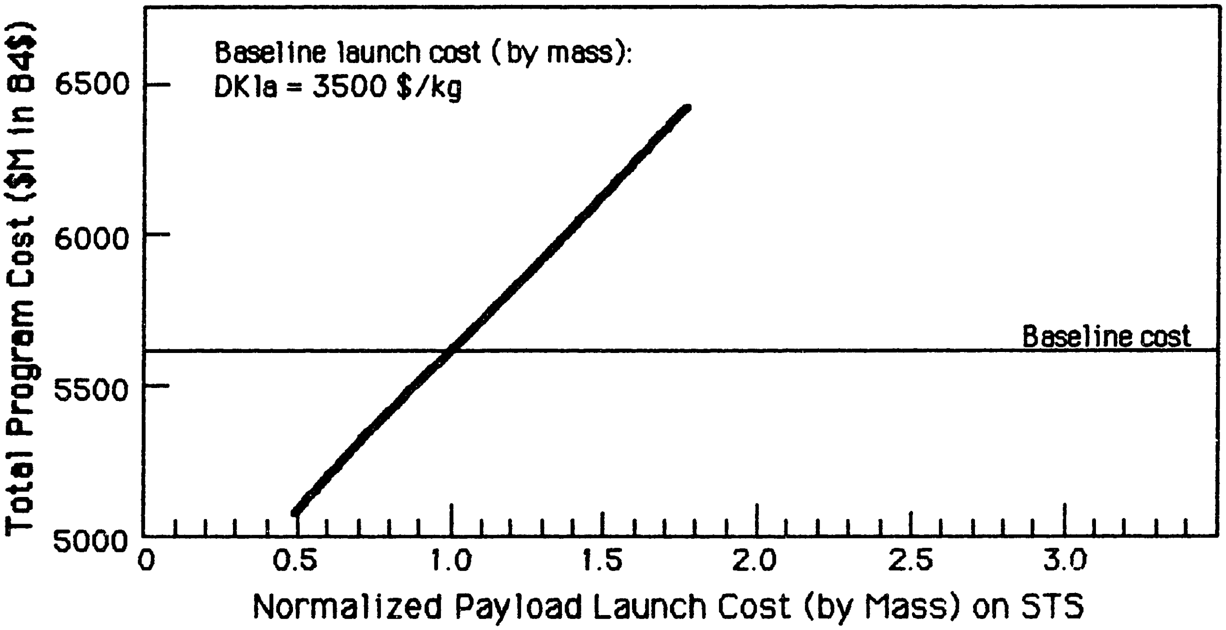


FIGURE F.9: EFFECT OF SPACE SHUTTLE PAYLOAD LAUNCH COST ON CASE EXAMPLE'S OVERALL PROGRAM COST

By comparison, the cost to launch people nas very little effect, as shown in Figure F.10 (in that figure, the curve rises very slightly to the right). Although this factor is proportional to the payload launch cost, its impact is minimal in this case example, because only 2.7 person-launches are required. For a more human-intensive job, this sensitivity would increase.

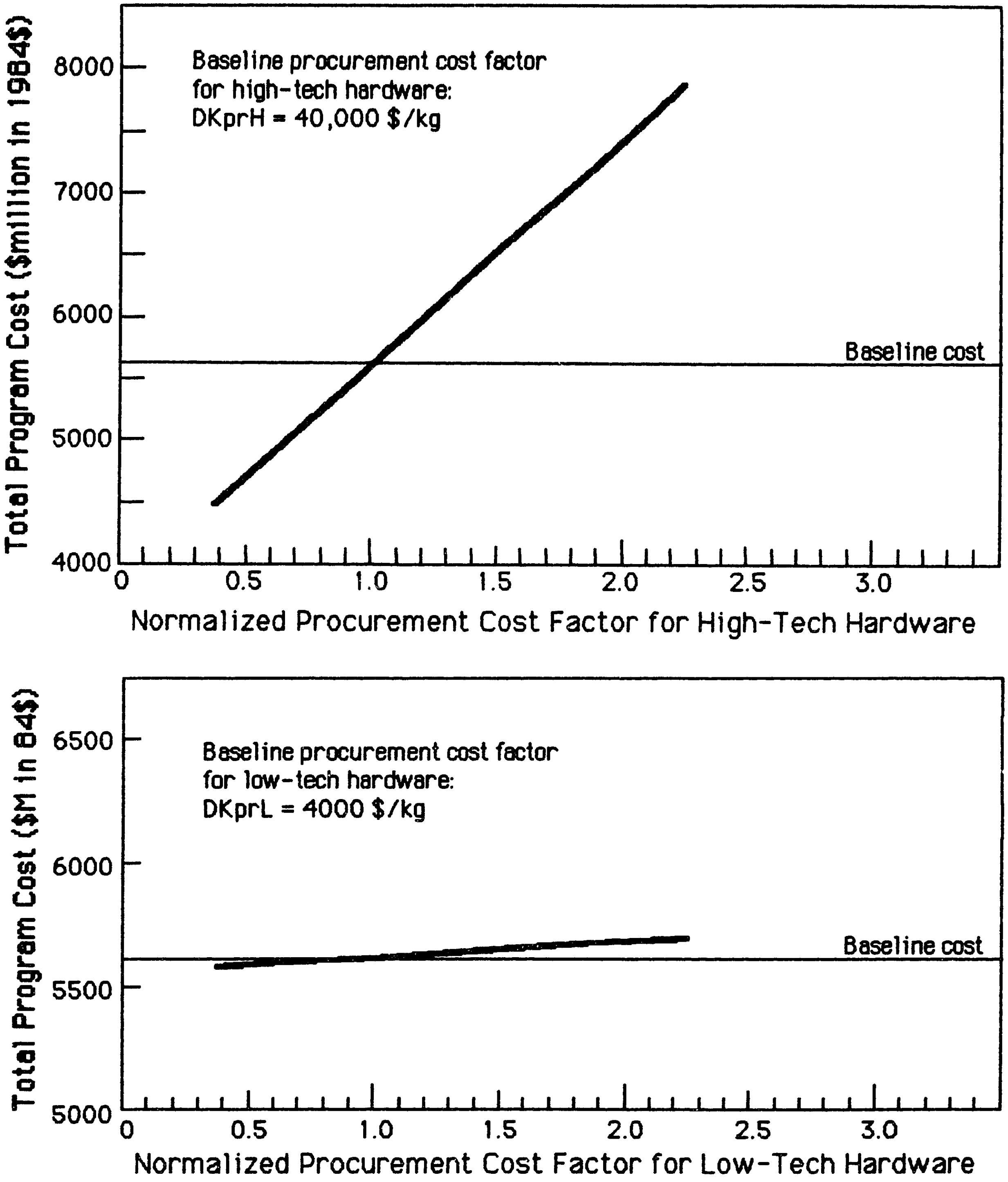


FIGURE F.8: EFFECT OF PROCUREMENT COST FACTORS ON CASE EXAMPLE'S OVERALL PROGRAM COST

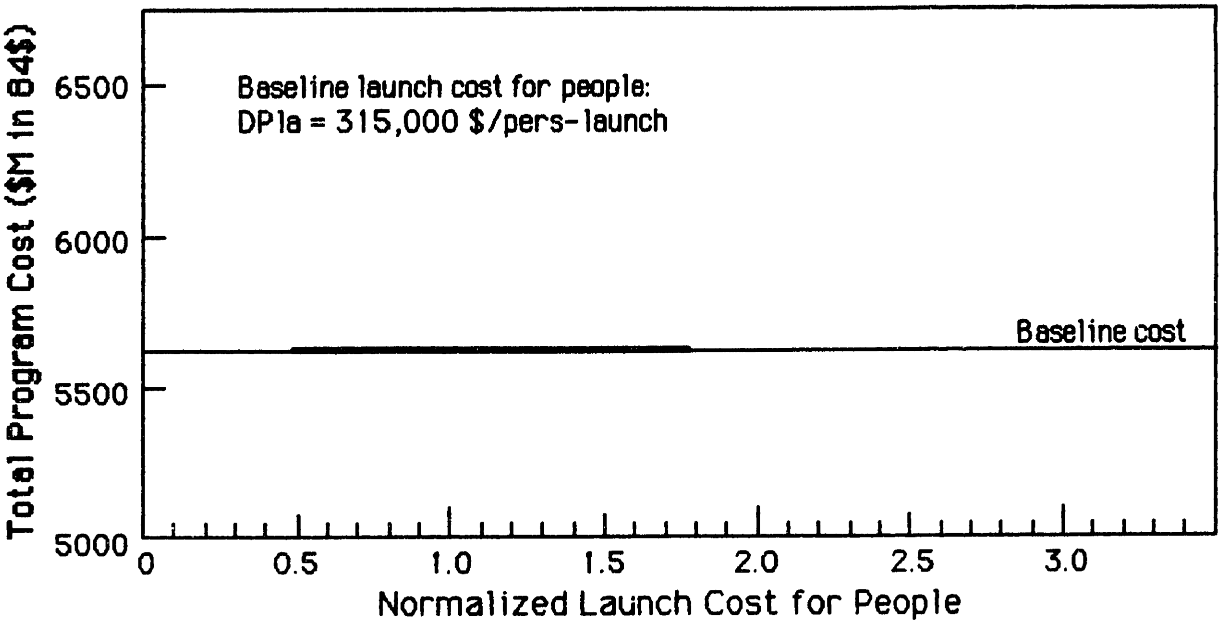


FIGURE F.10: EFFECT OF SPACE SHUTTLE LAUNCH COST FOR PEOPLE ON CASE EXAMPLE'S OVERALL PROGRAM COST

F.6 Effect of Ground Support Parameter:

The cost model calculates the size of the ground crew as a single number rather than a range of values. However, the impact of ground crew size is of interest, and is therefore plotted in Figure F.11. The crew size is varied through an arbitrary range of (0.6 * baseline) to (2.3 * baseline).

The effect of ground support crew size is relatively small, because in this case example much of the support activity takes place in space, and therefore the ground crew is not very numerous. If some other scenario required a larger baseline ground crew (e.g. 10 times larger), then its program cost would be more sensitive to this parameter.

F.7 Effects of Human-Related Parameters:

Another parameter of interest is the mass of the habitation module. Most of this is high-technology hardware, in fact the largest single piece of high-technology hardware to be provided for the construction scenario. The effect of this high-tech mass on the program cost is shown in Figure F.12.

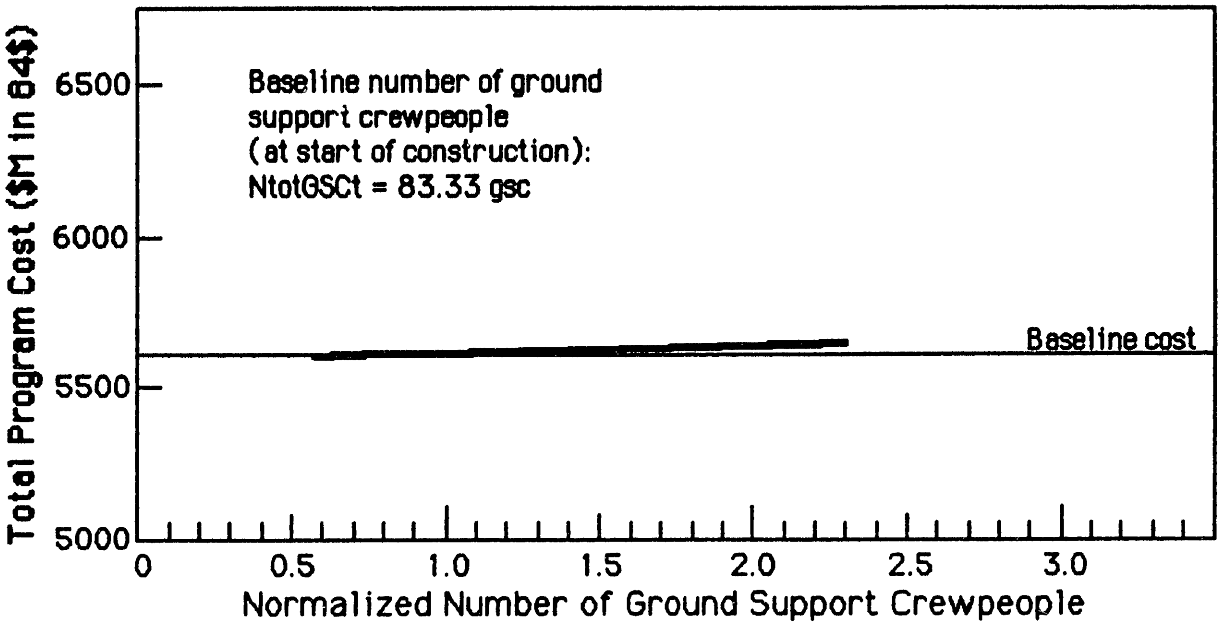


FIGURE F.11: EFFECT OF GROUND SUPPORT CREW SIZE ON CASE EXAMPLE'S OVERALL PROGRAM COST

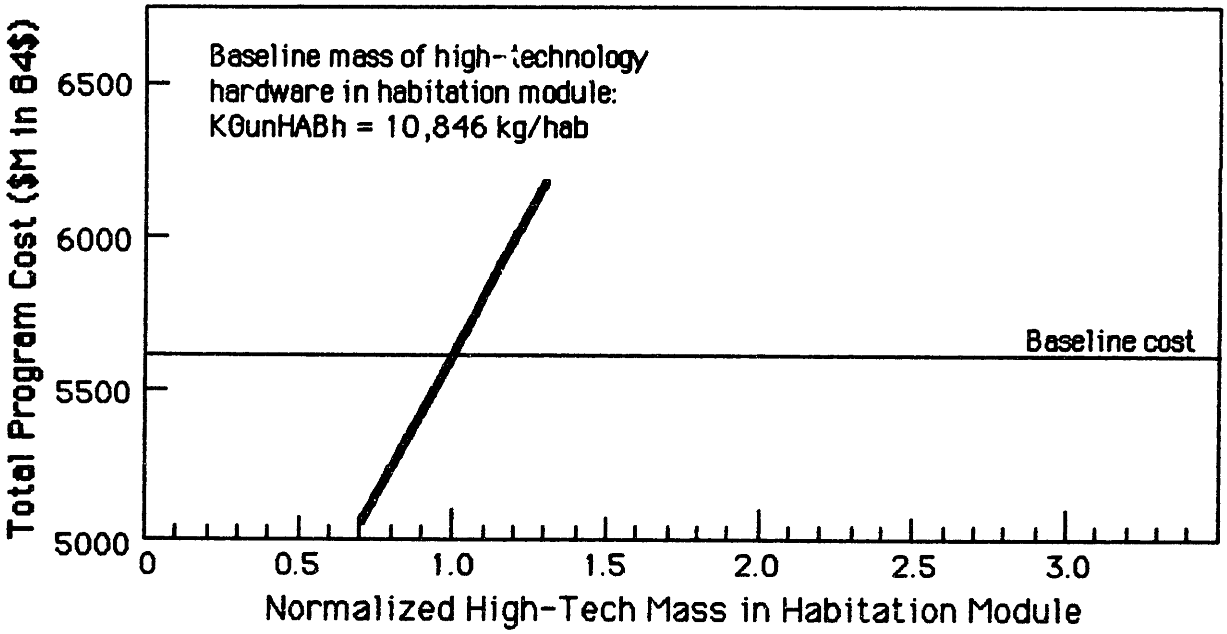


FIGURE F.12: EFFECT OF HIGH-TECH MASS IN HABITATION MODULE ON CASE EXAMPLE'S OVERALL PROGRAM COST

Figure F.13 displays the effect of the habitation module wearout factor (a measure of the module's maintenance requirements). This effect is not very important, suggesting that a higher wearout rate for the habitation module would be acceptable if this were accompanied by lower R&D and procurement cost factors. In other words, the baseline design in Appendix D may be somewhat over-engineered.

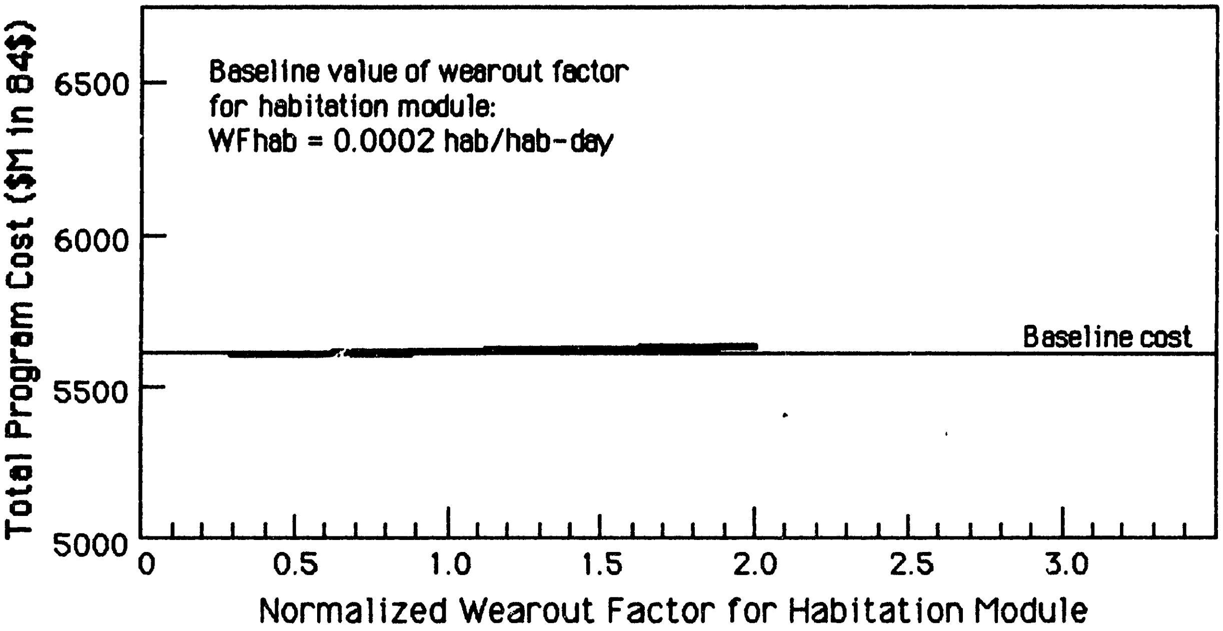


FIGURE F.13: EFFECT OF HABITATION MODULE WEAROUT FACTOR ON CASE EXAMPLE'S OVERALL PROGRAM COST

F.8 Effects of Teleoperator-Related Parameters:

Figure F.14 presents the effect of the mass of the construction teleoperator. The cost behavior is as expected: the larger the mass, the higher the R&D, procurement, and launch costs, and therefore the greater the program cost.

In section D.1.4 on teleoperator design, the consumables requirement is presented as a single value rather than a range. However, this factor can have a significant impact on total program cost if it varies substantially from its single-number estimate. (For example, the teloperator propellant requirement might be reduced by designing the teleoperator to "walk" along the structure rather than flying around it.) The teleoperator consumables requirement is varied through an arbitrary range of (0.3 * baseline) to (1.5 * baseline) in Figure F.15.

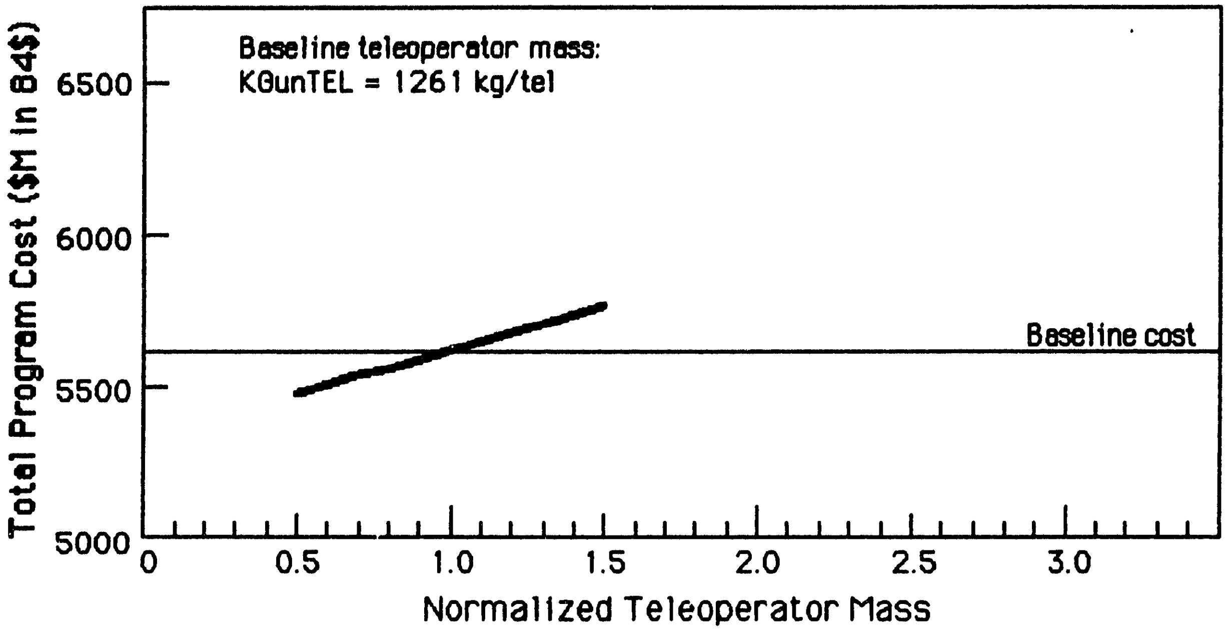


FIGURE F.14: EFFECT OF TELEOPERATOR MASS ON CASE EXAMPLE'S OVERALL PROGRAM COST

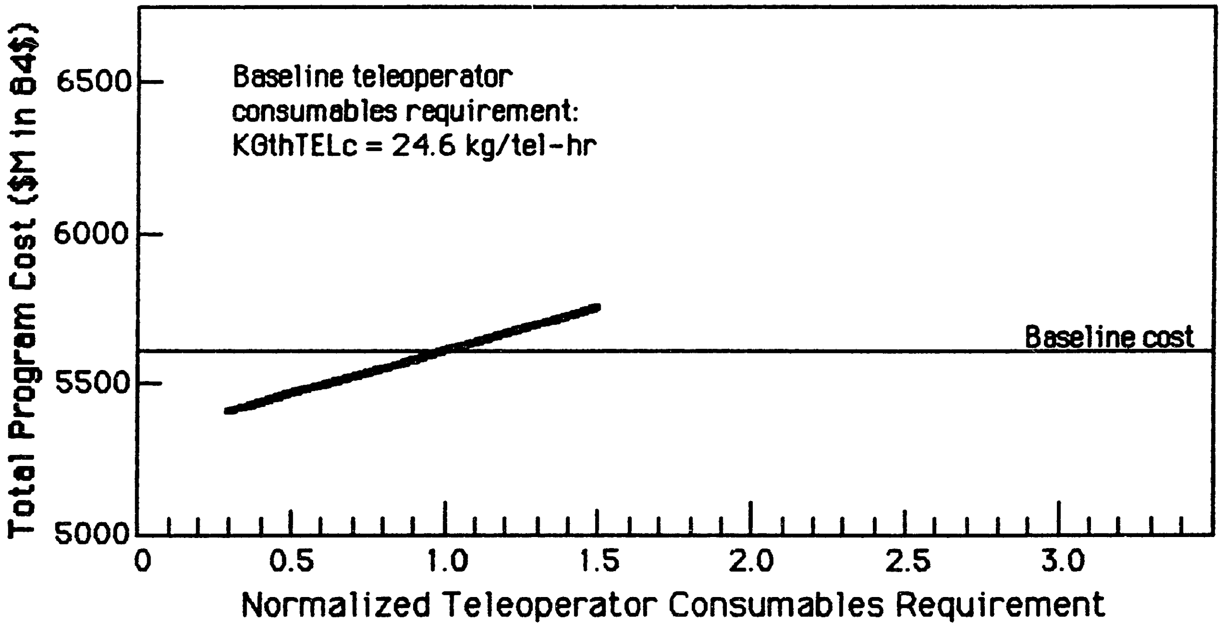


FIGURE F.15: EFFECT OF TELEOPERATOR CONSUMABLES REQUIREMENT ON CASE EXAMPLE'S OVERALL PROGRAM COST

The teleoperators spend a nominal 16 hours per day in construction at the worksite. If this daily work time is varied, the resulting change in total program cost is shown in Figure F.16. This displays the surprising result that costs increase with higher teleoperator duty cycles, suggesting that teleoperators should be left idle to minimize cost.

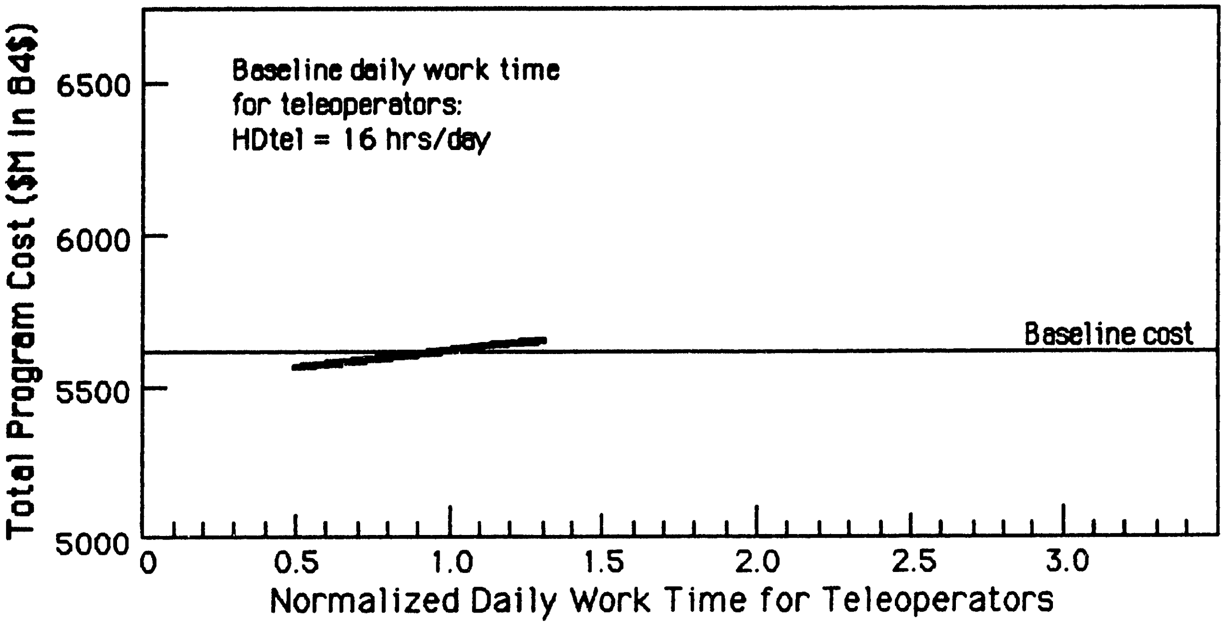


FIGURE F.16: EFFECT OF TELEOPERATOR DAILY WORK TIME ON CASE EXAMPLE'S OVERALL PROGRAM COST

The reason for this behavior is that in this case example, the teleoperators are more expensive to use in most tasks than the other options. For example, when a task is assigned to a combination of humans and teleoperators, some fraction of the total repetitions of the task will be performed by humans, and the rest by teleoperators. These fractions are determined by the relative daily productivities of these two options. If the daily work time of the teleoperators is increased, so is their daily productivity, and therefore they will perform a greater fraction of the total task. If the teleoperators are more expensive than humans in that task, then the costs of that task will increase.

It should be noted that this cost behavior only indicates that teleoperators are more expensive when all tasks are considered together. It is possible that for some tasks teleoperators are the cheapest alternative; if the daily work time were increased for those tasks only, then program costs would decrease. However, when the daily work time is varied for all of the tasks together, the less-favorable tasks dominate the overall cost behavior.

(As will be shown below, the effect of the automated equipment's daily work time is the opposite: increasing that daily work time reduces overall costs.)

Figure F.17 displays the effect of the teleoperator's wearout factor on overall program cost. As in the habitation module (Figure F.13), this effect is not very important, suggesting that a higher wearout rate for the teleoperator would be acceptable if this were accompanied by lower R&D and procurement costs.

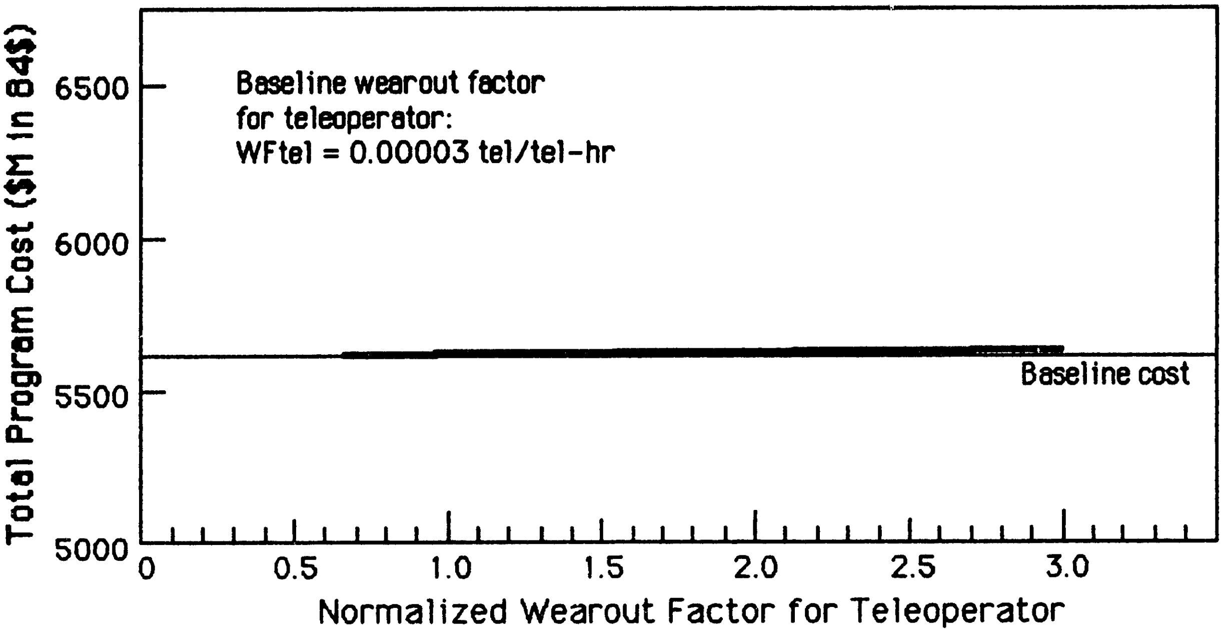


FIGURE F.17: EFFECT OF TELEOPERATOR WEAROUT FACTOR ON CASE EXAMPLE'S OVERALL PROGRAM COST

F.9 Effects of Automated-Equipment-Related Parameters:

Figure F.18 shows the overall cost effect of the unit mass for automated equipment. As mentioned in section D.1.5, the baseline mass estimate is also the low estimate for its range. The cost behavior is as expected: higher equipment mass leads to increased program costs. This mass can have a significant effect, because it drives the automated equipment's sizable R&D and procurement costs.

Figure F.19 presents the effect of variation in the daily work time of automated equipment. Unlike the teleoperator work time curve in Figure F.16, in this case the total program cost decreases with increased use of the automated equipment. This

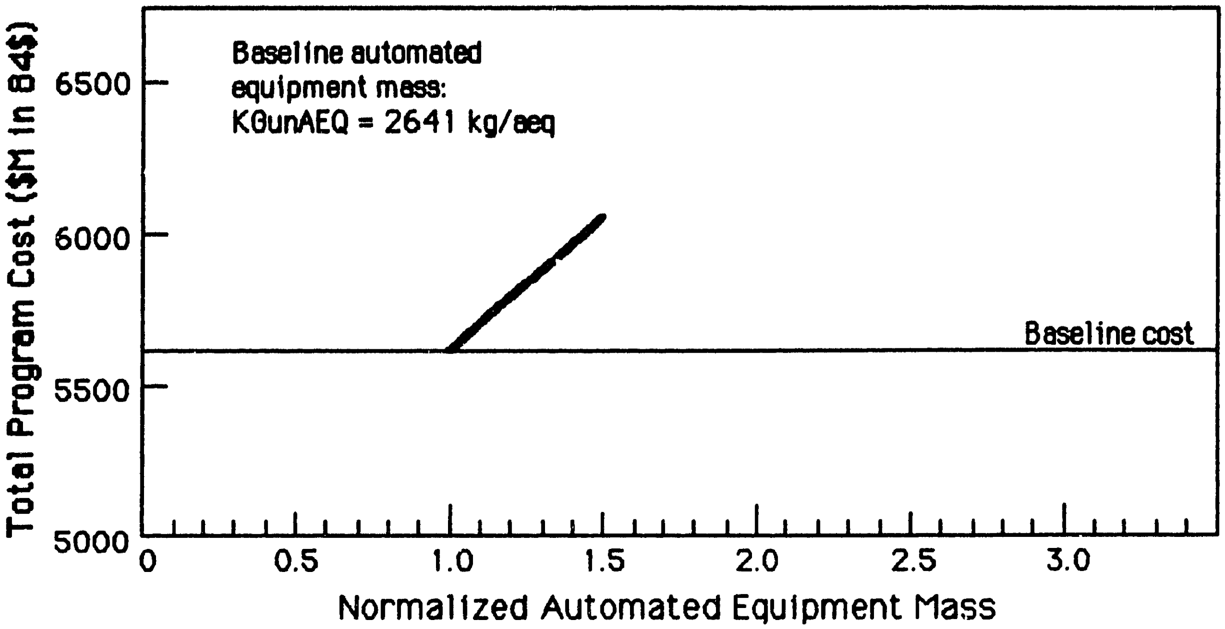


FIGURE F.18: EFFECT OF AUTOMATED EQUIPMENT MASS ON CASE EXAMPLE'S OVERALL PROGRAM COST

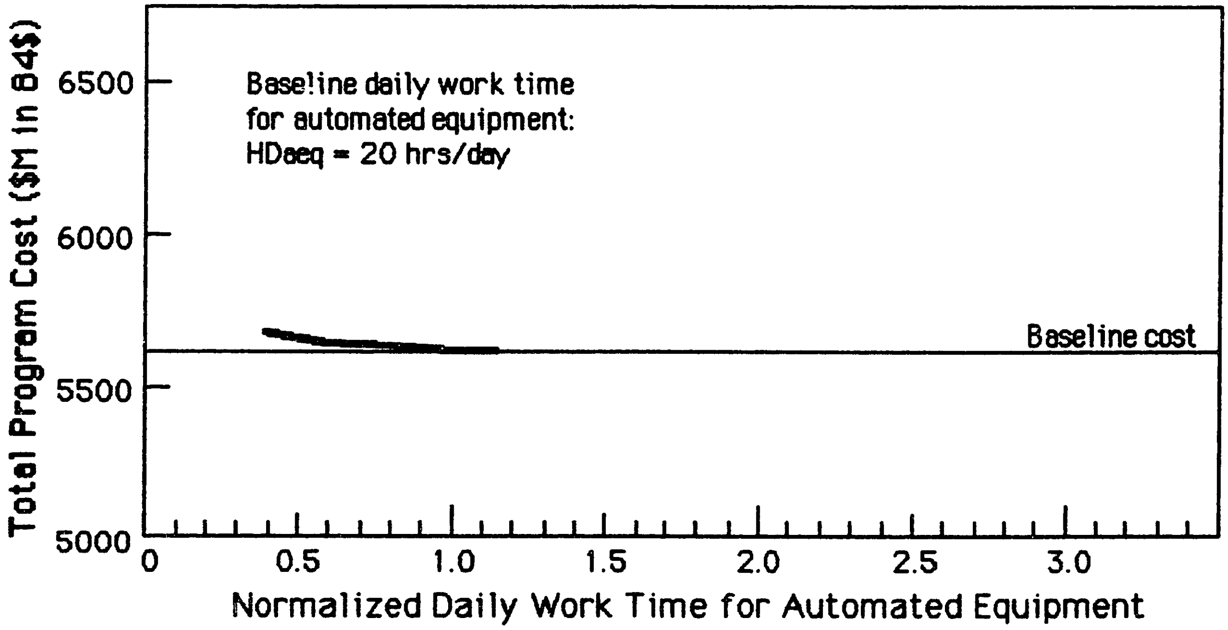


FIGURE F.19: EFFECT OF AUTOMATED EQUIPMENT DAILY WORK TIME ON CASE EXAMPLE'S OVERALL PROGRAM COST

indicates that automated equipment is generally a favorable option for the tasks to which it is applied: if the automated equipment performs a greater share of each of those tasks, total program cost goes down.

Similarly to the teleoperator work time, this result does not indicate that automated equipment is the cheapest option in any particular task. It only indicates that the favorable applications of automated equipment dominate the cost behavior when all tasks are considered together.

The effect of the automated equipment wearout factor is shown in Figure F.20. As in the habitation module and the teleoperator (Figures F.13 and F.17), this effect is not very important. This suggests that the automated equipment design is over-engineered: a higher wearout rate would be acceptable if accompanied by lower R&D and procurement costs.

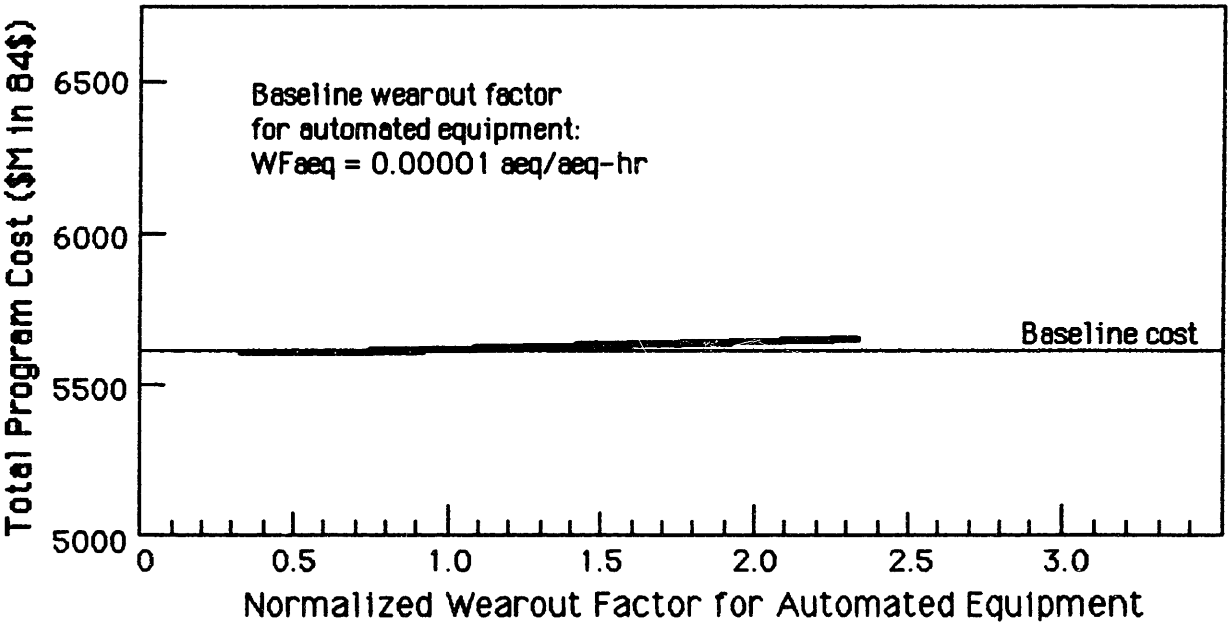


FIGURE F.20: EFFECT OF AUTOMATED EQUIPMENT WEAROUT FACTOR ON CASE EXAMPLE'S OVERALL PROGRAM COST

F.10 Effect of Miscellaneous Parameter:

Figure F.21 shows the effect of the construction site propellant requirement on overall costs. For consistency, the figure's horizontal axis has the same scale as the other graphs in this section. However, the range estimated in section D.1.6 extends to (5 * baseline), and so the cost curve actually reaches beyond the figure's right boundary.

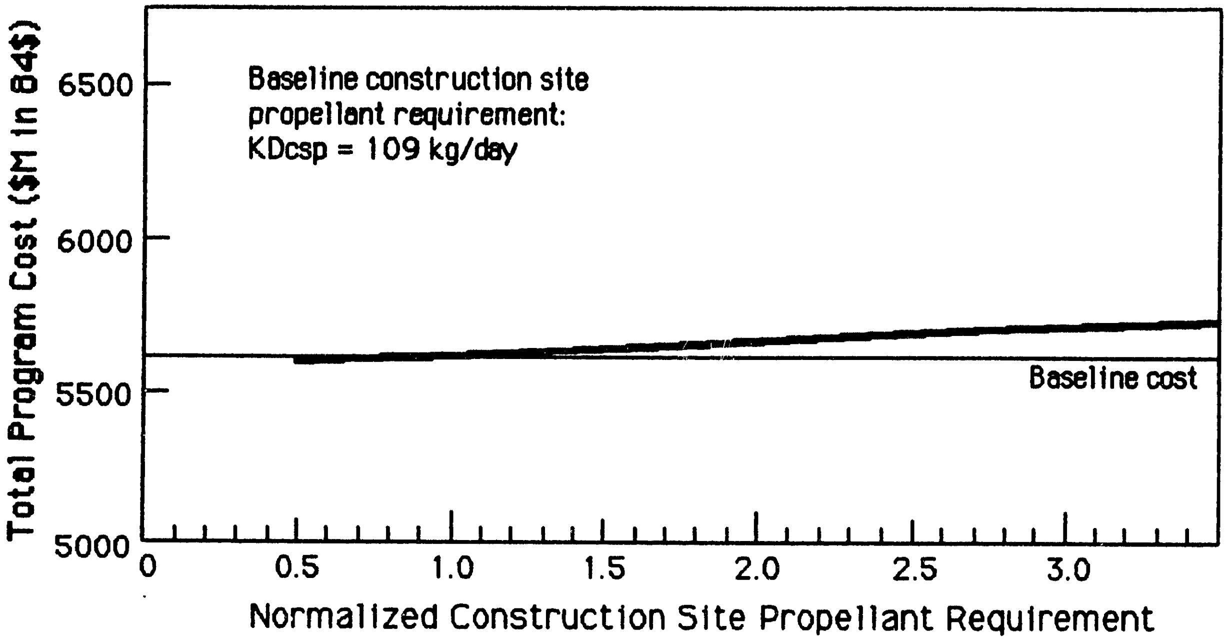


FIGURE F.21: EFFECT OF CONSTRUCTION SITE PROPELLANT REQUIREMENT ON CASE EXAMPLE'S OVERALL PROGRAM COST

Thus this parameter can have a significant impact on overall cost. This propellant requirement depends on orbital drag, which is a strong function of altitude. Therefore, if the drag varies much from baseline (e.g. during sunspot maximum), the altitude tradeoff between the launch cost and this propellant requirement should be recalculated.

APPENDIX G:

REDUCTION OF LINE-ITEM COST EQUATION

G.1 Line-Item Cost Equation:

This section presents the equation for total program cost, as implemented in the line-item cost model (i.e. in the computer programs in Appendix H). The individual line items are identified in curly brackets in the right-hand column.

Total program cost ProgCost =

$$\begin{aligned}
&\mathrm{DKrdH} * \mathrm{KGunSSTh} && \{\mathrm{CTrdSST}\}\\
&+ (\mathrm{DKrdH} * \mathrm{KGunSSTh} * \mathrm{NunSSTh} * 0.035)\\
&+ \mathrm{DKrdL} * \mathrm{KGunSSTl}\\
&+ (\mathrm{DKrdL} * \mathrm{KGunSSTl} * \mathrm{NunSSTl} * 0.035)\\[6pt]
&+ ((\mathrm{DKrdH} * \mathrm{KGunHABh}) + (\mathrm{DKrdL} * \mathrm{KGunHABl})) * \mathrm{YSNOhum} && \{\mathrm{CTrdHAB}\}\\[6pt]
&+ (\mathrm{CTrdEMU} + \mathrm{CTrdMMU} + (\mathrm{DKrdH} * \mathrm{KGunEQP})) * \mathrm{YSNOhum} && \{\mathrm{CTrdHSE}\}\\[6pt]
&+ (\mathrm{DKrdH} * \mathrm{KGunTEL}) * \mathrm{YSNOtel} && \{\mathrm{CTrdTEL}\}\\[6pt]
&+ (\mathrm{DKrdH} * \mathrm{KGunAEQ}) * \mathrm{YSNOaeq} && \{\mathrm{CTrdAEQ}\}\\[6pt]
&+ \mathrm{DKrdL} * \mathrm{KGunCSE} && \{\mathrm{CTrdCSE}\}\\[6pt]
&+ \mathrm{CTrdTSK} && \{\mathrm{CTrdTSK}\}\\[6pt]
&+ \mathrm{CTrdGMS} + (\mathrm{CTrdHMS} * \mathrm{YSNOhum}) + (\mathrm{CTrdTCS} * \mathrm{YSNOtel})\\
&+ (\mathrm{CTrdAMS} * \mathrm{YSNOaeq}) && \{\mathrm{CTrdGSE}\}
\end{aligned}$$

$$\begin{aligned}
&+ (\mathrm{DKprH} * \mathrm{KGunSSTh}) * (1^{-p} + 2^{-p} + \ldots + \mathrm{trunc}(\mathrm{NunSSTh})^{-p}\\
&\qquad + (\mathrm{NunSSTh} - \mathrm{trunc}(\mathrm{NunSSTh})) * \mathrm{NunSSTh}^{-p}\ ) && \{\mathrm{CTprSSTh}\}\\
&\qquad\qquad \text{where } -p = (1.4427) * (\mathrm{LN}(\mathrm{lcpmSST}/100)
\end{aligned}$$

$$\begin{aligned}
&+ (\mathrm{DKprL} * \mathrm{KGunSSTl}) * (1^{-p} + 2^{-p} + \ldots + \mathrm{trunc}(\mathrm{NunSSTl})^{-p}\\
&\qquad + (\mathrm{NunSSTl} - \mathrm{trunc}(\mathrm{NunSSTl})) * \mathrm{NunSSTl}^{-p}\ ) && \{\mathrm{CTprSSTl}\}
\end{aligned}$$

$$+ (\mathrm{DKprH} * \mathrm{KGunHABh}) * (1^{-q} + 2^{-q} + \ldots + \mathrm{trunc}(\mathrm{NtotHABi})^{-q}$$

$$+ (\mathrm{NtotHABi} - \mathrm{trunc}(\mathrm{NtotHABi})) * \mathrm{NtotHABi}^{-q}) \qquad \{\mathrm{CTprHABh}\}$$

$$\text{where } -q = (1.4427) * (\mathrm{LN}(\mathrm{lcparam}/100)$$

$$\text{and where } \mathrm{NtotHABi} = (\mathrm{NtotHUMi} / \mathrm{NhumHAB})$$

$$\text{where } \mathrm{NtotHUMi} = \sum_{j=1}^{\mathrm{NTSK}} \mathrm{Nhum}[j]$$

$$+ (\mathrm{DKprH} * \mathrm{KGunHABh}) * (1^{-q} + 2^{-q} + \ldots + \mathrm{trunc}(\mathrm{NtotHABs})^{-q}$$

$$+ (\mathrm{NtotHABs} - \mathrm{trunc}(\mathrm{NtotHABs})) * \mathrm{NtotHABs}^{-q}) \qquad \{\mathrm{CTprHABs}\}$$

$$\text{where } \mathrm{NtotHABs} = ((\mathrm{NtotPD} / \mathrm{NhumHAB}) * \mathrm{WFhab})$$

$$\text{where } \mathrm{NtotPD} = \sum_{j=1}^{\mathrm{NTSK}} \mathrm{Nhum}[j] * \mathrm{Ttime}[j]$$

$$+ (\mathrm{DKprL} * \mathrm{KGunHABl}) * (1^{-q} + 2^{-q} + \ldots + \mathrm{trunc}(\mathrm{NtotHABi})^{-q}$$

$$+ (\mathrm{NtotHABi} - \mathrm{trunc}(\mathrm{NtotHABi})) * \mathrm{NtotHABi}^{-q}) \qquad \{\mathrm{CTprHABl}\}$$

$$+ ((2 * \mathrm{NtotHUMi}) + (\mathrm{NtotPD} * \mathrm{HDhum} * \mathrm{WFemu})) * \mathrm{KGunEMU} * \mathrm{DKprH} \qquad \{\mathrm{CTprEMU}\}$$

$$+ (\mathrm{NtotHUMi} + (\mathrm{NtotPD} * \mathrm{HDhum} * \mathrm{WFmmu})) * \mathrm{KGunMMU} * \mathrm{DKprH} \qquad \{\mathrm{CTprMMU}\}$$

$$+ (\mathrm{NtotHUMi} + (\mathrm{NtotPD} * \mathrm{HDhum} * \mathrm{WFeqp})) * \mathrm{KGunEQP} * \mathrm{DKprH} \qquad \{\mathrm{CTprEQP}\}$$

$$+ (\mathrm{DKprH} * \mathrm{KGunTEL}) * (1^{-q} + 2^{-q} + \ldots + \mathrm{trunc}(\mathrm{NtotTELi})^{-q}$$

$$+ (\mathrm{NtotTELi} - \mathrm{trunc}(\mathrm{NtotTELi})) * \mathrm{NtotTELi}^{-q}) \qquad \{\mathrm{CTprTELi}\}$$

$$\text{where } \mathrm{NtotTELi} = \sum_{j=1}^{\mathrm{NTSK}} \mathrm{Ntel}[j]$$

$$+ (\mathrm{DKprH} * \mathrm{KGunTEL}) * (1^{-q} + 2^{-q} + \ldots + \mathrm{trunc}(\mathrm{NtotTELs})^{-q}$$

$$+ (\mathrm{NtotTELs} - \mathrm{trunc}(\mathrm{NtotTELs})) * \mathrm{NtotTELs}^{-q}) \qquad \{\mathrm{CTprTELs}\}$$

where $\mathrm{NtotTELs} = \mathrm{NtotTH} * \mathrm{WFtel}$

and $\mathrm{NtotTH} = \mathrm{HDtel} * \sum_{j=1}^{\mathrm{NTSK}} \mathrm{Ntel}[j] * \mathrm{Ttime}[j]$

$$+ (\mathrm{DKprH} * \mathrm{KGunAEQ}) * (1^{-q} + 2^{-q} + \ldots + \mathrm{trunc}(\mathrm{NtotAEQi})^{-q}$$

$$+ (\mathrm{NtotAEQi} - \mathrm{trunc}(\mathrm{NtotAEQi})) * \mathrm{NtotAEQi}^{-q}) \qquad \{\mathrm{CTprAEQi}\}$$

where $\mathrm{NtotAEQi} = \sum_{j=1}^{\mathrm{NTSK}} \mathrm{Naeq}[j]$

$$+ (\mathrm{DKprH} * \mathrm{KGunAEQ}) * (1^{-q} + 2^{-q} + \ldots + \mathrm{trunc}(\mathrm{NtotAEQs})^{-q}$$

$$+ (\mathrm{NtotAEQs} - \mathrm{trunc}(\mathrm{NtotAEQs})) * \mathrm{NtotAEQs}^{-q}) \qquad \{\mathrm{CTprAEQs}\}$$

where $\mathrm{NtotAEQs} = \mathrm{NtotAH} * \mathrm{WFaeq}$

where $\mathrm{NtotAH} = \mathrm{HDaeq} * \sum_{j=1}^{\mathrm{NTSK}} \mathrm{Naeq}[j] * \mathrm{Ttime}[j]$

$$+ \mathrm{NtotCSE} * \mathrm{KGunCSE} * \mathrm{DKprL} \qquad \{\mathrm{CTprCSE}\}$$

$$+ (2 * \mathrm{NtotHUMi}) * \mathrm{DpHUMt} \qquad \{\mathrm{CTprHUMt}\}$$

$$+ ((2 * \mathrm{NdHUMt} * \mathrm{NtotHUMi}) + (2 * \mathrm{NtotPD})) * \mathrm{DpdHUM} \qquad \{\mathrm{CTprHUMb}\}$$

$$+ \mathrm{DUprGMS} * (1^{-q} + 2^{-q} + \ldots + \mathrm{trunc}(\mathrm{NtotGMSi})^{-q}$$

$$+ (\mathrm{NtotGMSi} - \mathrm{trunc}(\mathrm{NtotGMSi})) * \mathrm{NtotGMSi}^{-q}) \qquad \{\mathrm{CTprGMSi}\}$$

where $\mathrm{NtotGMSi} = \mathrm{NgmsGEN}$

$$+ \mathrm{DUprGMS} * (1^{-q} + 2^{-q} + \dots + \mathrm{trunc}(\mathrm{NtotGMSs})^{-q} + (\mathrm{NtotGMSs} - \mathrm{trunc}(\mathrm{NtotGMSs})) * \mathrm{NtotGMSs}^{-q}) \quad \{\mathrm{CTprGMSs}\}$$

where NtotGMSs = NtotGMSi * Jobtime * 24 * WFgms
where NtotGMSi = NgmsGEN

$$+ \mathrm{DUprHMS} * (1^{-q} + 2^{-q} + \dots + \mathrm{trunc}(\mathrm{NtotHMSi})^{-q} + (\mathrm{NtotHMSi} - \mathrm{trunc}(\mathrm{NtotHMSi})) * \mathrm{NtotHMSi}^{-q}) \quad \{\mathrm{CTprHMSi}\}$$

where NtotHMSi = NtotHUMi * NhmsHUM

$$+ \mathrm{DUprHMS} * (1^{-q} + 2^{-q} + \dots + \mathrm{trunc}(\mathrm{NtotHMSs})^{-q} + (\mathrm{NtotHMSs} - \mathrm{trunc}(\mathrm{NtotHMSs})) * \mathrm{NtotHMSs}^{-q}) \quad \{\mathrm{CTprHMSs}\}$$

where NtotHMSs = NtotPD * HDhum * NhmsHUM * WFhms

$$+ \mathrm{DUprTCS} * (1^{-q} + 2^{-q} + \dots + \mathrm{trunc}(\mathrm{NtotTCSi})^{-q} + (\mathrm{NtotTCSi} - \mathrm{trunc}(\mathrm{NtotTCSi})) * \mathrm{NtotTCSi}^{-q}) \quad \{\mathrm{CTprTCSi}\}$$

where NtotTCSi = NtotTELi * NtcsTEL

$$+ \mathrm{DUprTCS} * (1^{-q} + 2^{-q} + \dots + \mathrm{trunc}(\mathrm{NtotTCSs})^{-q} + (\mathrm{NtotTCSs} - \mathrm{trunc}(\mathrm{NtotTCSs})) * \mathrm{NtotTCSs}^{-q}) \quad \{\mathrm{CTprTCSs}\}$$

where NtotTCSs = NtotTH * NtcsTEL * WFtcs

$$+ \mathrm{DUprAMS} * (1^{-q} + 2^{-q} + \dots + \mathrm{trunc}(\mathrm{NtotAMSi})^{-q} + (\mathrm{NtotAMSi} - \mathrm{trunc}(\mathrm{NtotAMSi})) * \mathrm{NtotAMSi}^{-q}) \quad \{\mathrm{CTprAMSi}\}$$

where NtotAMSi = NtotAEQi * NamsAEQ

$$+ \mathrm{DUprAMS} * (1^{-q} + 2^{-q} + \dots + \mathrm{trunc}(\mathrm{NtotAMSs})^{-q} + (\mathrm{NtotAMSs} - \mathrm{trunc}(\mathrm{NtotAMSs})) * \mathrm{NtotAMSs}^{-q}) \quad \{\mathrm{CTprAMSs}\}$$

where NtotAMSs = NtotAH * NamsAEQ * WFams

$$+ (\mathrm{NunSSTh} * \mathrm{KGunSSTh} * \mathrm{DKmpH}) + (\mathrm{NunSSTh} * \mathrm{KGunSSTh} * \mathrm{DKla}) \quad \{\mathrm{CTlaSSTh}, \text{ for } \mathrm{DensSSTh} >= 92.5\}$$

or

$$+ (\mathrm{NunSSTh} * \mathrm{KGunSSTh} * \mathrm{DKmpH}) + (\mathrm{NunSSTh} * \mathrm{KGunSSTh} / \mathrm{DensSSTh}) * \mathrm{DMla} \quad \{\mathrm{CTlaSSTh}, \text{ for } \mathrm{DensSSTh} < 92.5\}$$

```
+ (NunSST1 * KGunSST1 * DKmpL) + (NunSST1 * KGunSST1 * DK1a)
                                   {CT1aSST1, for DensSST1 >= 92.5}
                           or
                           + (NunSST1 * KGunSST1 * DKmpL)
                           + (NunSST1 * KGunSST1 / DensSST1) * DM1a
                                    {CT1aSST1, for DensSST1 < 92.5}

+ (NtotPD / NDstint) * DP1a                               {CT1aHUM}

+ (NtotHABh * KGunHABh * DKmpH) + (NtotHABh * KGunHABh * DK1a)
                    where NtotHABh = (NtotHUMi / NhumHAB)
                                     + ((NtotPD / NhumHAB) * WFhab)

+ (NtotHAB1 * KGunHAB1 * DKmpL) + (NtotHAB1 * KGunHAB1 * DK1a)
                              where NtotHAB1 = (NtotHUMi / NhumHAB)

+ NtotPD * KGpdHABc * DK1a
                                                          {CT1aHAB}

+ (NtotEMU * KGunEMU * DKmpH) + (NtotEMU * KGunEMU * DK1a)
                        where NtotEMU = (2 * NtotHUMi)
                                        + (NtotPD * HDhum * WFemu)

+ NtotPD * HDhum * KGehEMUc * DK1a
                                                          {CT1aEMU}

+ (NtotMMU * KGunMMU * DKmpH) + (NtotMMU * KGunMMU * DK1a)
                        where NtotMMU = NtotHUMi
                                        + (NtotPD * HDhum * WFmmu)

+ NtotPD * HDhum * KGmhMMUc * DK1a
                                                          {CT1aMMU}

+ (NtotEQP * KGunEQP * DKmpH) + (NtotEQP * KGunEQP * DK1a)
                                                          {CT1aEQP}
                        where NtotEQP = NtotHUMi
                                        + (NtotPD * HDhum * WFeqp)

+ (NtotTEL * KGunTEL * DKmpH) + (NtotTEL * KGunTEL * DK1a)
                              where NtotTEL = NtotTELi
                                              + (NtotTH * WFtel)

+ NtotTH * KGthTELc * DK1a
                                                          {CT1aTEL}

+ (NtotAEQ * KGunAEQ * DKmpH) + (NtotAEQ * KGunAEQ * DK1a)
                                                          {CT1aAEQ}
                              where NtotAEQ = NtotAEQi
                                              + (NtotAH * WFaeq)
```

+ (NtotCSE * KGunCSE * DKmpL) + (NtotCSE * KGunCSE * DKla)
{CTlaCSE}

+ KDcsp * Jobtime * DKla {CTlaCSP}

+ (CTprSST + CTlaSST) * FRlaINS {CTinSST}
where CTprSST =

(DKprH * KGunSSTh) * $(1^{-p} + 2^{-p} + \ldots + \mathrm{trunc}(NunSSTh)^{-p}$

+ (NunSSTh - trunc(NunSSTh)) * $NunSSTh^{-p}$)

+ (DKprL * KGunSSTl) * $(1^{-p} + 2^{-p} + \ldots + \mathrm{trunc}(NunSSTl)^{-p}$

+ (NunSSTl - trunc(NunSSTl)) * $NunSSTl^{-p}$)

and CTlaSST =
(NunSSTh * KGunSSTh * DKmpH) + (NunSSTh * KGunSSTh * DKla)
or
+ (NunSSTh * KGunSSTh * DKmpH)
+ (NunSSTh * KGunSSTh / DensSSTh) * DMla
{For DensSSTh < 92.5}
+ (NunSSTl * KGunSSTl * DKmpL) + (NunSSTl * KGunSSTl * DKla)
or
+ (NunSSTl * KGunSSTl * DKmpL)
+ (NunSSTl * KGunSSTl / DensSSTl) * DMla
{For DensSSTl < 92.5}

+ (CTprHAB + CTlaHAB) * FRlaINS {CTinHAB}
where CTprHAB =

(DKprH * KGunHABh) * $(1^{-q} + 2^{-q} + \ldots + \mathrm{trunc}(NtotHABi)^{-q}$

+ (NtotHABi - trunc(NtotHABi)) * $NtotHABi^{-q}$)

+ (DKprH * KGunHABh) * $(1^{-q} + 2^{-q} + \ldots + \mathrm{trunc}(NtotHABs)^{-q}$

+ (NtotHABs - trunc(NtotHABs)) * $NtotHABs^{-q}$)

+ (DKprL * KGunHABl) * $(1^{-q} + 2^{-q} + \ldots + \mathrm{trunc}(NtotHABi)^{-q}$

+ (NtotHABi - trunc(NtotHABi)) * $NtotHABi^{-q}$)

and CTlaHAB =
(NtotHABh * KGunHABh * DKmpH) + (NtotHABh * KGunHABh * DKla)
+ (NtotHABl * KGunHABl * DKmpL) + (NtotHABl * KGunHABl * DKla)
+ NtotPD * KGpdHABc * DKla

+ (CTprHSE + CTlaHSE) * FRlaINS {CTinHSE}

where CTprHSE =
((2 * NtotHUMi) + (NtotPD * HDhum * WFemu)) * KGunEMU * DKprH
+ (NtotHUMi + (NtotPD * HDhum * WFmmu)) * KGunMMU * DKprH
+ (NtotHUMi + (NtotPD * HDhum * WFeqp)) * KGunEQP * DKprH

and CTlaHSE =
(NtotEMU * KGunEMU * DKmpH) + (NtotEMU * KGunEMU * DKla)
+ NtotPD * HDhum * KGehEMUc * DKla
+ (NtotMMU * KGunMMU * DKmpH) + (NtotMMU * KGunMMU * DKla)
+ NtotPD * HDhum * KGmhMMUc * DKla
+ (NtotEQP * KGunEQP * DKmpH) + (NtotEQP * KGunEQP * DKla)

+ (CTprTEL + CTlaTEL) * FRlaINS {CTinTEL}

where CTprTEL =
$(DKprH * KGunTEL) * (1^{-q} + 2^{-q} + \ldots + \mathrm{trunc}(NtotTELi)^{-q}$
$+ (NtotTELi - \mathrm{trunc}(NtotTELi)) * NtotTELi^{-q})$
$+ (DKprH * KGunTEL) * (1^{-q} + 2^{-q} + \ldots + \mathrm{trunc}(NtotTELs)^{-q}$
$+ (NtotTELs - \mathrm{trunc}(NtotTELs)) * NtotTELs^{-q})$

and CTlaTEL =
(NtotTEL * KGunTEL * DKmpH) + (NtotTEL * KGunTEL * DKla)
+ NtotTH * KGthTELc * DKla

+ (CTprAEQ + CTlaAEQ) * FRlaINS {CTinAEQ}

where CTprAEQ =
$(DKprH * KGunAEQ) * (1^{-q} + 2^{-q} + \ldots + \mathrm{trunc}(NtotAEQi)^{-q}$
$+ (NtotAEQi - \mathrm{trunc}(NtotAEQi)) * NtotAEQi^{-q})$
$+ (DKprH * KGunAEQ) * (1^{-q} + 2^{-q} + \ldots + \mathrm{trunc}(NtotAEQs)^{-q}$
$+ (NtotAEQs - \mathrm{trunc}(NtotAEQs)) * NtotAEQs^{-q})$

and CTlaAEQ =
(NtotAEQ * KGunAEQ * DKmpH) + (NtotAEQ * KGunAEQ * DKla)

+ (CTprCSE + CTlaCSE) * FRlaINS {CTinCSE}

where CTprCSE = NtotCSE * KGunCSE * DKprL

and CTlaCSE =
(NtotCSE * KGunCSE * DKmpL) + (NtotCSE * KGunCSE * DKla)

+ KDcsp * Jobtime * DKla * FRlaINS {CTinCSP}

```
+ (NgmsGEN * NgscGMS * (24/8)) * DpGSCt

+ (NtotHUMi * NhmsHUM * HDhum * NgscHMS / 8) * DpGSCt

+ (NtotTELi * NtcsTEL * HDtel * NgscTCS / 8) * DpGSCt

+ (NtotAEQi * NamsAEQ * HDaeq * NgscAMS / 8) * DpGSCt
                                                                 {CTgsGSCt}

+ (NgmsGEN * NgscGMS * (24/8)) * NdHUMt * 8 * DphGSC

+ (NtotHUMi * NhmsHUM * HDhum * NgscHMS / 8) * NdHUMt * 8 * DphGSC

+ (NtotTELi * NtcsTEL * HDtel * NgscTCS / 8) * NdHUMt * 8 * DphGSC

+ (NtotAEQi * NamsAEQ * HDaeq * NgscAMS / 8) * NdHUMt * 8 * DphGSC

+ (NgmsGEN * Jobtime * 24 * NgscGMS) * DphGSC

+ (NtotPD * HDhum * NhmsHUM * NgscHMS) * DphGSC

+ (NtotTH * NtcsTEL * NgscTCS) * DphGSC

+ (NtotAH * NamsAEQ * NgscAMS) * DphGSC
                                                                 {CTgsGSCb}

+ ((72 * Jobtime) + (NtotPD * HDhum) + NtotAH) * DlhTDR
                                                                  {CTgsLCC}

+ NtotTH * DhhTDR                                                 {CTgsHCC}
```

G.2 Line-Item Equation for ProgCost, Rearranged:

With the decision variables Nhum[j], Ntel[j], Naeq[j], and Ttime[j] in mind, the cost equation in section G.1 can be rearranged into twelve useful categories:

1) Whole-job R&D costs
2) Human-related R&D costs
3) Teleoperator-related R&D costs (constants)
4) Automated-equipment-related R&D costs

5) Whole-job general costs

6) Number-of-humans costs
   (costs dependent on NtotHUMi = number of humans in space)
7) Human usage costs
   (costs dependent on NtotPD = number of person-days in space)

8) Number-of-teleoperators costs
   (costs dependent on NtotTELi = number of teleoperators in space)
9) Teleoperator usage costs
   (costs dependent on NtotTH = number of teleoperator-hours in space)

10) Number-of-automated-equipment-units costs
    (costs dependent on NtotAEQi = number of units of automated equipment in space)
11) Automated equipment usage costs
    (costs dependent on NtotAH = number of automated equipment unit-hours in space)

12) Whole-job timecosts
    (functions of Jobtime = time to complete whole job)

Categories 1 through 5 contribute constant terms which do not affect the outcome of the optimization. Categories 6 through 11 include the decision variables of interest. It may appear that category 12 introduces another decision variable Jobtime. However, since the time to complete the job is the time to complete the longest task, Jobtime is actually equal to the largest of the Ttime[j] terms.

The rearrangement of the line-item cost equation into these categories is presented in sections G.2 and G.3. This section expands and rearranges the original line items. Section G.3 will combine some of the terms within each category, compressing the overall cost equation down to 6 pages.

At this stage of the reduction, the equation terms are not yet in terms of the desired decision variables Nhum[j], Ntel[j], Naeq[j], and Ttime[j]. For conciseness, the equation keeps certain summation terms, such as:

```
NtotHUMi = summation of Nhum[j] over all j
NtotPD = summation of (Nhum[j] * Ttime[j]) over all j
```

Similar terms are used for teleoperators and automated equipment. The cost equation can be recast in terms of the desired decision variables by expanding these summation terms. For the moment, however, this concise form is more convenient.

The original line items from which the terms were extracted are identified in curly brackets.

Whole-job R&D costs = Rw:

```
DKrdH * KGunSSTh
+ (DKrdH * KGunSSTh * NunSSTh * 0.035)
+ DKrdL * KGunSSTl
+ (DKrdL * KGunSSTl * NunSSTl * 0.035)                 {CTrdSST}

+ DKrdL * KGunCSE                                      {CTrdCSE}

+ CTrdTSK                                              {CTrdTSK}

+ CTrdGMS                                         {From CTrdGSE}
```

Human-related R&D costs = Rh:

```
((DKrdH * KGunHABh) + (DKrdL * KGunHABl)) * YSNOhum    {CTrdHAB}

+ (CTrdEMU + CTrdMMU + (DKrdH * KGunEQP)) * YSNOhum    {CTrdHSE}

+ (CTrdHMS * YSNOhum)                             {From CTrdGSE}
```

Teleoperator-related R&D costs = Rt:

```
(DKrdH * KGunTEL) * YSNOtel                            {CTrdTEL}

+ (CTrdTCS * YSNOtel)                             {From CTrdGSE}
```

Automated-equipment-related R&D costs = Ra:

```
(DKrdH * KGunAEQ) * YSNOaeq                            {CTrdAEQ}

+ (CTrdAMS * YSNOaeq)                             {From CTrdGSE}
```

Whole-job general costs = Cw:

$$(\text{DKprH} * \text{KGunSSTh}) * (1^{-p} + 2^{-p} + \ldots + \text{trunc}(\text{NunSSTh})^{-p} + (\text{NunSSTh} - \text{trunc}(\text{NunSSTh})) * \text{NunSSTh}^{-p}) \qquad \{\text{CTprSSTh}\}$$

where $-p = (1.4427) * (\text{LN}(\text{lcpmSST}/100)$

$$+ (\text{DKprL} * \text{KGunSSTl}) * (1^{-p} + 2^{-p} + \ldots + \text{trunc}(\text{NunSSTl})^{-p} + (\text{NunSSTl} - \text{trunc}(\text{NunSSTl})) * \text{NunSSTl}^{-p}) \qquad \{\text{CTprSSTl}\}$$

$$+ \text{NtotCSE} * \text{KGunCSE} * \text{DKprL} \qquad \{\text{CTprCSE}\}$$

$$+ \text{DUprGMS} * (1^{-q} + 2^{-q} + \ldots + \text{trunc}(\text{NtotGMSi})^{-q} + (\text{NtotGMSi} - \text{trunc}(\text{NtotGMSi})) * \text{NtotGMSi}^{-q}) \qquad \{\text{CTprGMSi}\}$$

where $-q = (1.4427) * (\text{LN}(\text{lcparam}/100)$

and where NtotGMSi = NgmsGEN

$$+ (\text{NunSSTh} * \text{KGunSSTh} * \text{DKmpH}) + (\text{NunSSTh} * \text{KGunSSTh} * \text{DKla})$$

{CTlaSSTh, for DensSSTh >= 92.5}

or

$$+ \text{NunSSTh} * \text{KGunSSTh} * (\text{DKmpH} + (\text{DMla} / \text{DensSSTh}))$$

{CTlaSSTh, for DensSSTh < 92.5}

$$+ (\text{NunSSTl} * \text{KGunSSTl} * \text{DKmpL}) + (\text{NunSSTl} * \text{KGunSSTl} * \text{DKla})$$

{CTlaSSTl, for DensSSTl >= 92.5}

or

$$+ \text{NunSSTl} * \text{KGunSSTl} * (\text{DKmpL} + (\text{DMla} / \text{DensSSTl}))$$

{CTlaSSTl, for DensSSTl < 92.5}

$$+ \text{NtotCSE} * \text{KGunCSE} * (\text{DKmpL} + \text{DKla}) \qquad \{\text{CTlaCSE}\}$$

$$+ (\text{DKprH} * \text{KGunSSTh}) * (1^{-p} + 2^{-p} + \ldots + \text{trunc}(\text{NunSSTh})^{-p} + (\text{NunSSTh} - \text{trunc}(\text{NunSSTh})) * \text{NunSSTh}^{-p}) * \text{FRlaINS}$$

$$+ (\text{DKprL} * \text{KGunSSTl}) * (1^{-p} + 2^{-p} + \ldots + \text{trunc}(\text{NunSSTl})^{-p} + (\text{NunSSTl} - \text{trunc}(\text{NunSSTl})) * \text{NunSSTl}^{-p}) * \text{FRlaINS}$$

$$+ (\text{NunSSTh} * \text{KGunSSTh} * (\text{DKmpH} + \text{DKla}) * \text{FRlaINS}$$

or

$$+ \text{NunSSTh} * \text{KGunSSTh} * (\text{DKmpH} + (\text{DMla} / \text{DensSSTh})) * \text{FRlaINS}$$

{For DensSSTh < 92.5}

```
+ (NunSST1 * KGunSST1 * (DKmpL + DK1a) * FR1aINS
                     or
                     + NunSST1 * KGunSST1
                       * (DKmpL + (DM1a / DensSST1)) * FR1aINS
                     {For DensSST1 < 92.5}
                                                           {CTinSST}

+ NtotCSE * KGunCSE * DKprL * FR1aINS
+ (NtotCSE * KGunCSE * (DKmpL + DK1a) * FR1aINS            {CTinCSE}

+ (NgmsGEN * NgscGMS * (24/8)) * DpGSCt               {From CTgsGSCt}

+ (NgmsGEN * NgscGMS * (24/8)) * NdHUMt * 8 * DphGSC
                                                      {From CTgsGSCb}
```

<u>Number-of-humans costs (functions of NtotHUMi):</u>

```
                       -q   -q                        -q
(DKprH * KGunHABh) * (1   + 2   + ... + trunc(NtotHABi)
                                                -q
    + (NtotHABi - trunc(NtotHABi)) * NtotHABi    )       {CTprHABh}
                          where NtotHABi = (NtotHUMi / NhumHAB)

                                                  NTSK
                                                  \
                                  where NtotHUMi = >   Nhum[j]
                                                  /
                                                  j=1

                          -q   -q                        -q
+ (DKprL * KGunHABl) * (1   + 2   + ... + trunc(NtotHABi)
                                                -q
    + (NtotHABi - trunc(NtotHABi)) * NtotHABi    )       {CTprHABl}

+ (2 * NtotHUMi) * KGunEMU * DKprH                   {From CTprEMU}

+ NtotHUMi * KGunMMU * DKprH                         {From CTprMMU}

+ NtotHUMi * KGunEQP * DKprH                         {From CTprEQP}

+ (2 * NtotHUMi) * DpHUMt                                {CTprHUMt}

+ (2 * NdHUMt * NtotHUMi) * DpdHUM                  {From CTprHUMb}

                  -q   -q                        -q
+ DUprHMS * (1   + 2   + ... + trunc(NtotHMSi)
                                                -q
    + (NtotHMSi - trunc(NtotHMSi)) * NtotHMSi    )       {CTprHMSi}
                          where NtotHMSi = NtotHUMi * NhmsHUM
```

$$\begin{aligned}&\text{+ (NtotHUMi / NhumHAB) * KGunHABh * (DKmpH + DKla)}\\&\text{+ (NtotHUMi / NhumHAB) * KGunHABl * (DKmpL + DKla)}\\&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\{\text{From CTlaHAB}\}\end{aligned}$$

$$\text{+ (2 * NtotHUMi) * KGunEMU * (DKmpH + DKla)}\qquad\{\text{From CTlaEMU}\}$$

$$\text{+ NtotHUMi * KGunMMU * (DKmpH + DKla)}\qquad\{\text{From CTlaMMU}\}$$

$$\text{+ NtotHUMi * KGunEQP * (DKmpH + DKla)}\qquad\{\text{From CTlaEQP}\}$$

$$\begin{aligned}&\text{+ (DKprH * KGunHABh) * (}1^{-q}\text{ + }2^{-q}\text{ + ... + trunc(NtotHABi)}^{-q}\\&\qquad\text{+ (NtotHABi - trunc(NtotHABi)) * NtotHABi}^{-q}\text{) * FRlaINS}\\&\text{+ (DKprL * KGunHABl) * (}1^{-q}\text{ + }2^{-q}\text{ + ... + trunc(NtotHABi)}^{-q}\\&\qquad\text{+ (NtotHABi - trunc(NtotHABi)) * NtotHABi}^{-q}\text{) * FRlaINS}\\&\text{+ (NtotHUMi / NhumHAB) * KGunHABh * (DKmpH + DKla) * FRlaINS}\\&\text{+ (NtotHUMi / NhumHAB) * KGunHABl * (DKmpL + DKla) * FRlaINS}\\&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\{\text{From CTinHAB}\}\end{aligned}$$

$$\begin{aligned}&\text{+ (2 * NtotHUMi) * KGunEMU * DKprH * FRlaINS}\\&\text{+ NtotHUMi * KGunMMU * DKprH * FRlaINS}\\&\text{+ NtotHUMi * KGunEQP * DKprH * FRlaINS}\\&\text{+ (2 * NtotHUMi) * KGunEMU * (DKmpH + DKla) * FRlaINS}\\&\text{+ NtotHUMi * KGunMMU * (DKmpH + DKla) * FRlaINS}\\&\text{+ NtotHUMi * KGunEQP * (DKmpH + DKla) * FRlaINS}\qquad\{\text{From CTinHSE}\}\end{aligned}$$

$$\begin{aligned}&\text{+ (NtotHUMi * NhmsHUM * HDhum * NgscHMS / 8) * DpGSCt}\\&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\{\text{From CTgsGSCt}\}\end{aligned}$$

$$\begin{aligned}&\text{+ (NtotHUMi * NhmsHUM * HDhum * NgscHMS / 8) * NdHUMt * 8 * DphGSC}\\&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\{\text{From CTgsGSCb}\}\end{aligned}$$

Human usage costs (functions of NtotPD):

$$\begin{aligned}&\text{(DKprH * KGunHABh) * (}1^{-q}\text{ + }2^{-q}\text{ + ... + trunc(NtotHABs)}^{-q}\\&\qquad\text{+ (NtotHABs - trunc(NtotHABs)) * NtotHABs}^{-q}\text{)}\qquad\{\text{CTprHABs}\}\\&\qquad\qquad\qquad\text{where NtotHABs = ((NtotPD / NhumHAB) * WFhab)}\end{aligned}$$

$$\text{where NtotPD} = \sum_{j=1}^{\text{NTSK}} \text{Nhum[j] * Ttime[j]}$$

```
+ (NtotPD * HDhum * WFemu) * KGunEMU * DKprH       {From CTprEMU}

+ (NtotPD * HDhum * WFmmu) * KGunMMU * DKprH       {From CTprMMU}

+ (NtotPD * HDhum * WFeqp) * KGunEQP * DKprH       {From CTprEQP}

+ (2 * NtotPD) * DpdHUM                           {From CTprHUMb}

              -q   -q                       -q
+ DUprHMS * (1  + 2  + ... + trunc(NtotHMSs)
                                              -q
     + (NtotHMSs - trunc(NtotHMSs)) * NtotHMSs  )     {CTprHMSs}
                where NtotHMSs = NtotPD * HDhum * NhmsHUM * WFhms

+ (NtotPD / NDstint) * DPla                             {CTlaHUM}

+ ((NtotPD / NhumHAB) * WFhab) * KGunHABh * (DKmpH + DKla)
+ NtotPD * KGpdHABc * DKla                         {From CTlaHAB}

+ (NtotPD * HDhum * WFemu) * KGunEMU * (DKmpH + DKla)
+ NtotPD * HDhum * KGehEMUc * DKla                 {From CTlaEMU}

+ (NtotPD * HDhum * WFmmu) * KGunMMU * (DKmpH + DKla)
+ NtotPD * HDhum * KGmhMMUc * DKla                 {From CTlaMMU}

+ (NtotPD * HDhum * WFeqp) * KGunEQP * (DKmpH + DKla)
                                                   {From CTlaEQP}

                         -q   -q                        -q
+ (DKprH * KGunHABh) * (1  + 2  + ... + trunc(NtotHABs)
                                              -q
     + (NtotHABs - trunc(NtotHABs)) * NtotHABs  ) * FRlaINS
+ ((NtotPD / NhumHAB) * WFhab) * KGunHABh * (DKmpH + DKla) * FRlaINS
+ NtotPD * KGpdHABc * DKla * FRlaINS               {From CTinHAB}

+ (NtotPD * HDhum * WFemu) * KGunEMU * DKprH * FRlaINS
+ (NtotPD * HDhum * WFmmu) * KGunMMU * DKprH * FRlaINS
+ (NtotPD * HDhum * WFeqp) * KGunEQP * DKprH * FRlaINS
+ (NtotPD * HDhum * WFemu) * KGunEMU * (DKmpH + DKla) * FRlaINS
+ NtotPD * HDhum * KGehEMUc * DKla * FRlaINS
+ (NtotPD * HDhum * WFmmu) * KGunMMU * (DKmpH + DKla) * FRlaINS
+ NtotPD * HDhum * KGmhMMUc * DKla * FRlaINS
+ (NtotPD * HDhum * WFeqp) * KGunEQP * (DKmpH + DKla) * FRlaINS
                                                   {From CTinHSE}

+ (NtotPD * HDhum * NhmsHUM * NgscHMS) * DphGSC    {From CTgsGSCb}

+ (NtotPD * HDhum) * DlhTDR                        {From CTgsLCC}
```

Number-of-teleoperators costs (functions of NtotTELi):

$$+ (\text{DKprH} * \text{KGunTEL}) * (1^{-q} + 2^{-q} + \dots + \text{trunc}(\text{NtotTELi})^{-q}$$
$$+ (\text{NtotTELi} - \text{trunc}(\text{NtotTELi})) * \text{NtotTELi}^{-q}) \qquad \{\text{CTprTELi}\}$$

$$\text{where NtotTELi} = \sum_{j=1}^{\text{NTSK}} \text{Ntel}[j]$$

$$+ \text{DUprTCS} * (1^{-q} + 2^{-q} + \dots + \text{trunc}(\text{NtotTCSi})^{-q}$$
$$+ (\text{NtotTCSi} - \text{trunc}(\text{NtotTCSi})) * \text{NtotTCSi}^{-q}) \qquad \{\text{CTprTCSi}\}$$
$$\text{where NtotTCSi} = \text{NtotTELi} * \text{NtcsTEL}$$

$$+ \text{NtotTELi} * \text{KGunTEL} * (\text{DKmpH} + \text{DKla}) \qquad \{\text{From CTlaTEL}\}$$

$$+ (\text{DKprH} * \text{KGunTEL}) * (1^{-q} + 2^{-q} + \dots + \text{trunc}(\text{NtotTELi})^{-q}$$
$$+ (\text{NtotTELi} - \text{trunc}(\text{NtotTELi})) * \text{NtotTELi}^{-q}) * \text{FRlaINS}$$
$$+ \text{NtotTELi} * \text{KGunTEL} * (\text{DKmpH} + \text{DKla}) * \text{FRlaINS} \qquad \{\text{From CTinTEL}\}$$

$$+ (\text{NtotTELi} * \text{NtcsTEL} * \text{HDtel} * \text{NgscTCS} / 8) * \text{DpGSCt} \qquad \{\text{From CTgsGSCt}\}$$

$$+ (\text{NtotTELi} * \text{NtcsTEL} * \text{HDtel} * \text{NgscTCS} / 8) * \text{NdHUMt} * 8 * \text{DphGSC} \qquad \{\text{From CTgsGSCb}\}$$

Teleoperator usage costs (functions of NtotTH):

$$+ (\text{DKprH} * \text{KGunTEL}) * (1^{-q} + 2^{-q} + \dots + \text{trunc}(\text{NtotTELs})^{-q}$$
$$+ (\text{NtotTELs} - \text{trunc}(\text{NtotTELs})) * \text{NtotTELs}^{-q}) \qquad \{\text{CTprTELs}\}$$
$$\text{where NtotTELs} = \text{NtotTH} * \text{WFtel}$$

$$\text{and NtotTH} = \text{HDtel} * \sum_{j=1}^{\text{NTSK}} \text{Ntel}[j] * \text{Ttime}[j]$$

$$+ \text{DUprTCS} * (1^{-q} + 2^{-q} + \dots + \text{trunc}(\text{NtotTCSs})^{-q}$$
$$+ (\text{NtotTCSs} - \text{trunc}(\text{NtotTCSs})) * \text{NtotTCSs}^{-q}) \qquad \{\text{CTprTCSs}\}$$
$$\text{where NtotTCSs} = \text{NtotTH} * \text{NtcsTEL} * \text{WFtcs}$$

+ (NtotTH * WFtel) * KGunTEL * (DKmpH + DKla)
+ NtotTH * KGthTELc * DKla {From CTlaTEL}

+ (DKprH * KGunTEL) * (1^{-q} + 2^{-q} + ... + trunc(NtotTELs)$^{-q}$
+ (NtotTELs - trunc(NtotTELs)) * NtotTELs^{-q}) * FRlaINS
+ (NtotTH * WFtel) * KGunTEL * (DKmpH + DKla) * FRlaINS
+ NtotTH * KGthTELc * DKla * FRlaINS {From CTinTEL}

+ (NtotTH * NtcsTEL * NgscTCS) * DphGSC {From CTgsGSCb}

+ NtotTH * DhhTDR {CTgsHCC}

Number-of-automated-equipment-units costs (functions of NtotAEQi):

(DKprH * KGunAEQ) * (1^{-q} + 2^{-q} + ... + trunc(NtotAEQi)$^{-q}$
+ (NtotAEQi - trunc(NtotAEQi)) * NtotAEQi^{-q}) {CTprAEQi}

where $\text{NtotAEQi} = \sum_{j=1}^{NTSK} \text{Naeq[j]}$

+ DUprAMS * (1^{-q} + 2^{-q} + ... + trunc(NtotAMSi)$^{-q}$
+ (NtotAMSi - trunc(NtotAMSi)) * NtotAMSi^{-q}) {CTprAMSi}
where NtotAMSi = NtotAEQi * NamsAEQ

+ NtotAEQi * KGunAEQ * (DKmpH + DKla) {From CTlaAEQ}

+ (DKprH * KGunAEQ) * (1^{-q} + 2^{-q} + ... + trunc(NtotAEQi)$^{-q}$
+ (NtotAEQi - trunc(NtotAEQi)) * NtotAEQi^{-q}) * FRlaINS
+ NtotAEQi * KGunAEQ * (DKmpH + DKla) * FRlaINS {From CTinAEQ}

+ (NtotAEQi * NamsAEQ * HDaeq * NgscAMS / 8) * DpGSCt
{From CTgsGSCt}

+ (NtotAEQi * NamsAEQ * HDaeq * NgscAMS / 8) * NdHUMt * 8 * DphGSC
{From CTgsGSCb}

Automated equipment usage costs (functions of NtotAH):

$$(\mathrm{DKprH} * \mathrm{KGunAEQ}) * (1^{-q} + 2^{-q} + \dots + \mathrm{trunc}(\mathrm{NtotAEQs})^{-q} + (\mathrm{NtotAEQs} - \mathrm{trunc}(\mathrm{NtotAEQs})) * \mathrm{NtotAEQs}^{-q}) \qquad \{\mathrm{CTprAEQs}\}$$

where $\mathrm{NtotAEQs} = \mathrm{NtotAH} * \mathrm{WFaeq}$

where $\mathrm{NtotAH} = \mathrm{HDaeq} * \sum_{j=1}^{\mathrm{NTSK}} \mathrm{Naeq}[j] * \mathrm{Ttime}[j]$

$$+ \mathrm{DUprAMS} * (1^{-q} + 2^{-q} + \dots + \mathrm{trunc}(\mathrm{NtotAMSs})^{-q} + (\mathrm{NtotAMSs} - \mathrm{trunc}(\mathrm{NtotAMSs})) * \mathrm{NtotAMSs}^{-q}) \qquad \{\mathrm{CTprAMSs}\}$$

where $\mathrm{NtotAMSs} = \mathrm{NtotAH} * \mathrm{NamsAEQ} * \mathrm{WFams}$

$$+ (\mathrm{NtotAH} * \mathrm{WFaeq}) * \mathrm{KGunAEQ} * (\mathrm{DKmpH} + \mathrm{DKla}) \qquad \{\mathrm{From\ CTlaAEQ}\}$$

$$+ (\mathrm{DKprH} * \mathrm{KGunAEQ}) * (1^{-q} + 2^{-q} + \dots + \mathrm{trunc}(\mathrm{NtotAEQs})^{-q} + (\mathrm{NtotAEQs} - \mathrm{trunc}(\mathrm{NtotAEQs})) * \mathrm{NtotAEQs}^{-q}) * \mathrm{FRlaINS}$$
$$+ (\mathrm{NtotAH} * \mathrm{WFaeq}) * \mathrm{KGunAEQ} * (\mathrm{DKmpH} + \mathrm{DKla}) * \mathrm{FRlaINS} \qquad \{\mathrm{From\ CTinAEQ}\}$$

$$+ (\mathrm{NtotAH} * \mathrm{NamsAEQ} * \mathrm{NgscAMS}) * \mathrm{DphGSC} \qquad \{\mathrm{From\ CTgsGSCb}\}$$

$$+ \mathrm{NtotAH} * \mathrm{DlhTDR} \qquad \{\mathrm{From\ CTgsLCC}\}$$

Whole-job timecosts (functions of Jobtime):

$$\mathrm{DUprGMS} * (1^{-q} + 2^{-q} + \dots + \mathrm{trunc}(\mathrm{NtotGMSs})^{-q} + (\mathrm{NtotGMSs} - \mathrm{trunc}(\mathrm{NtotGMSs})) * \mathrm{NtotGMSs}^{-q}) \qquad \{\mathrm{CTprGMSs}\}$$

where $\mathrm{NtotGMSs} = \mathrm{NtotGMSi} * \mathrm{Jobtime} * 24 * \mathrm{WFgms}$

where $\mathrm{NtotGMSi} = \mathrm{NgmsGEN}$

$$+ \mathrm{KDcsp} * \mathrm{Jobtime} * \mathrm{DKla} \qquad \{\mathrm{CTlaCSP}\}$$

$$+ \mathrm{KDcsp} * \mathrm{Jobtime} * \mathrm{DKla} * \mathrm{FRlaINS} \qquad \{\mathrm{CTinCSP}\}$$

$$+ (\mathrm{NgmsGEN} * \mathrm{Jobtime} * 24 * \mathrm{NgscGMS}) * \mathrm{DphGSC} \qquad \{\mathrm{From\ CTgsGSCb}\}$$

$$+ (72 * \mathrm{Jobtime}) * \mathrm{DlhTDR} \qquad \{\mathrm{From\ CTgsLCC}\}$$

G.3 Cost Equation, Rearranged and Compressed:

This section compresses the cost equation in section G.2 by combining terms together. The resulting equation has the form:

ProgCost =

(Rw + Rh + Rt + Ra + Cw)	Constants
+ (learning curve terms dependent on NtotHUMi) + (constant 1) * (NtotHUMi)	Number-of- humans
+ (learning curve terms dependent on NtotPD) + (constant 2) * (NtotPD)	Human usage
+ (learning curve terms dependent on NtotTELi) + (constant 3) * (NtotTELi)	Number-of- teleop.
+ (learning curve terms dependent on NtotTD) + (constant 4) * (NtotTD)	Teleop. usage
+ (learning curve terms dependent on NtotAEQi) + (constant 5) * (NtotAEQi)	Number-of- auto. eq.
+ (learning curve terms dependent on NtotAD) + (constant 6) * (NtotAD)	Auto. eq. usage
+ (learning curve term dependent on Jobtime) + (constant 7) * (Jobtime)	Whole-job timecosts

These program costs include four general types of terms:

1) Some constant terms:
   a) Rw, the R&D costs for the space structure to be assembled, some construction support equipment, and some ground support hardware;
   b) Rh, the R&D costs for space and ground support equipment for humans;
   c) Rt, the R&D costs for teleoperators and their control stations;
   d) Ra, the R&D costs for automated equipment and its ground support equipment;
   e) Cw, the procurement and launch costs for space structure, construction support equipment, and some ground support hardware.

2) For each human or machine option, some terms dependent on the total number of units of the option at the worksite, i.e. terms dependent on:

a) the number of humans NtotHUMi;
b) the number of teleoperators NtotTELi;
c) the number of automated equipment units NtotAEQi.

In general, these terms cover the procurement and launch of initial space hardware, the procurement of initial ground support equipment, and the training of space and ground support crews.

3) For each human or machine option, some terms dependent on the total usage of the option at the worksite, i.e. terms dependent on:

a) the number of person-days NtotPD;
b) the number of teleoperator-days NtotTD;
c) the number of automated equipment-days NtotAD.

These terms cover the procurement and launch of spares and consumables for hardware and crew, the crew rotation launches, the procurement of ground support equipment spares, the salaries of humans in space and on the ground, and the cost of communications to support the humans and machines in space.

4) Some terms dependent on the whole-job time Jobtime. These terms cover the procurement of spares for the general ground support equipment, the launch of construction site propellant, the salary of the general support ground crew, and the cost of the associated communications.

In this section, the cost terms dependent on teleoperator-hours (TH) and automated equipment unit-hours (AH) in the previous section have been redefined in terms of teleoperator-days (TD) and automated equipment unit-days (AD).

All the terms in the original cost equation are represented here, and these expressions yield the same total program cost as the line-item cost model.

Whole-job R&D costs = Rw:

(DKrdH * KGunSSTh) * (1 + (NunSSTh * 0.035))

+ (DKrdL * KGunSSTl) * (1 + (NunSSTl * 0.035))

+ DKrdL * KGunCSE

+ CTrdTSK

+ CTrdGMS

Human-related R&D costs = Rh:

```
((DKrdH * KGunHABh) + (DKrdL * KGunHABl)) * YSNOhum

+ (CTrdEMU + CTrdMMU + (DKrdH * KGunEQP)) * YSNOhum

+ (CTrdHMS * YSNOhum)
```

Teleoperator-related R&D costs = Rt:

```
(DKrdH * KGunTEL) * YSNOtel

+ (CTrdTCS * YSNOtel)
```

Automated-equipment-related R&D costs = Ra:

```
(DKrdH * KGunAEQ) * YSNOaeq

+ (CTrdAMS * YSNOaeq)
```

Whole-job general costs = Cw:

```
(DKprH * KGunSSTh) * (1 + FRlaINS)
          -p    -p                        -p
     * (1   + 2   + ... + trunc(NunSSTh)
                                                 -p
        + (NunSSTh - trunc(NunSSTh)) * NunSSTh     )
                               where -p = (1.4427) * (LN(lcpmSST/100)

+ (NunSSTh * KGunSSTh) * (DKmpH + DKla) * (1 + FRlaINS)
                                                    {For DensSSTh >= 100}
                     or
                     + (NunSSTh * KGunSSTh)
                       * (DKmpH + (DMla / DensSSTh))
                       * (1 + FRlaINS)
                                                     {For DensSSTh < 100}

+ (DKprL * KGunSSTl) * (1 + FRlaINS)
          -p    -p                        -p
     * (1   + 2   + ... + trunc(NunSSTl)
                                                 -p
        + (NunSSTl - trunc(NunSSTl)) * NunSSTl     )

+ (NunSSTl * KGunSSTl) * (DKmpL + DKla) * (1 + FRlaINS)
                                                    {For DensSSTl >= 100}
                     or
                     + (NunSSTl * KGunSSTl)
                       * (DKmpL + (DMla / DensSSTl))
                       * (1 + FRlaINS)
                                                     {For DensSSTl < 100}
```

```
+ (NtotCSE * KGunCSE) * (DKprL + DKmpL + DK1a) * (1 + FR1aINS)

              -q   -q                         -q
+ DUprGMS * (1  + 2  + ... + trunc(NtotGMSi)
                                                -q
     + (NtotGMSi - trunc(NtotGMSi)) * NtotGMSi    )
                            where -q = (1.4427) * (LN(1cparam/100)
                                      and where NtotGMSi = NgmsGEN

+ (NgmsGEN * NgscGMS * (24/8)) * (DpGSCt + (NdHUMt * 8 * DphGSC))
```

Number-of-humans costs (functions of NtotHUMi):

```
((DKprH * KGunHABh) + (DKprL * KGunHAB1)) * (1 + FR1aINS)
         -q   -q                          -q
     * (1  + 2  + ... + trunc(NtotHABi)
                                                   -q
        + (NtotHABi - trunc(NtotHABi)) * NtotHABi    )
                        where NtotHABi = NtotHUMi * (1 / NhumHAB)
```

$$\text{where NtotHUMi} = \sum_{j=1}^{\text{NTSK}} \text{Nhum[j]}$$

```
              -q   -q                         -q
+ DUprHMS * (1  + 2  + ... + trunc(NtotHMSi)
                                                -q
     + (NtotHMSi - trunc(NtotHMSi)) * NtotHMSi    )
                              where NtotHMSi = NtotHUMi * NhmsHUM

+ NtotHUMi
     * (((1 / NhumHAB) * KGunHABh * (DKmpH + DK1a) * (1 + FR1aINS))
        + ((1 / NhumHAB) * KGunHAB1 * (DKmpL + DK1a) * (1 + FR1aINS))

        + (((2 * KGunEMU) + KGunMMU + KGunEQP)
           * (DKprH + DKmpH + DK1a) * (1 + FR1aINS))

        + (2 * DpHUMt)

        + (2 * NdHUMt * DpdHUM)

        + ((NhmsHUM * HDhum * NgscHMS / 8)
           * (DpGSCt + (NdHUMt * 8 * DphGSC))))
```

Human usage costs (functions of NtotPD):

$$
\begin{aligned}
&(\mathrm{DKprH} * \mathrm{KGunHABh}) * (1 + \mathrm{FRlaINS})\\
&\quad * (1^{-q} + 2^{-q} + \dots + \mathrm{trunc}(\mathrm{NtotHABs})^{-q}\\
&\qquad + (\mathrm{NtotHABs} - \mathrm{trunc}(\mathrm{NtotHABs})) * \mathrm{NtotHABs}^{-q})\\
&\qquad\qquad \text{where } \mathrm{NtotHABs} = \mathrm{NtotPD} * (\mathrm{WFhab} / \mathrm{NhumHAB})
\end{aligned}
$$

$$
\text{where } \mathrm{NtotPD} = \sum_{j=1}^{\mathrm{NTSK}} \mathrm{Nhum}[j] * \mathrm{Ttime}[j]
$$

$$
\begin{aligned}
&+ \mathrm{DUprHMS} * (1^{-q} + 2^{-q} + \dots + \mathrm{trunc}(\mathrm{NtotHMSs})^{-q}\\
&\qquad + (\mathrm{NtotHMSs} - \mathrm{trunc}(\mathrm{NtotHMSs})) * \mathrm{NtotHMSs}^{-q})\\
&\qquad\qquad \text{where } \mathrm{NtotHMSs} = \mathrm{NtotPD} * (\mathrm{HDhum} * \mathrm{NhmsHUM} * \mathrm{WFhms})
\end{aligned}
$$

$$
\begin{aligned}
&+ \mathrm{NtotPD}\\
&\quad * (((1 + \mathrm{FRlaINS})\\
&\qquad * (((1 / \mathrm{NhumHAB}) * \mathrm{WFhab} * \mathrm{KGunHABh} * (\mathrm{DKmpH} + \mathrm{DKla}))\\
&\qquad\quad + (\mathrm{KGpdHABc} * \mathrm{DKla})))\\
&\qquad + ((1 + \mathrm{FRlaINS})\\
&\qquad\quad * ((\mathrm{HDhum} * \mathrm{WFemu} * \mathrm{KGunEMU}) * (\mathrm{DKprH} + \mathrm{DKmpH} + \mathrm{DKla})\\
&\qquad\qquad + (\mathrm{HDhum} * \mathrm{KGehEMUc} * \mathrm{DKla})\\
&\qquad\qquad + (\mathrm{HDhum} * \mathrm{WFmmu} * \mathrm{KGunMMU}) * (\mathrm{DKprH} + \mathrm{DKmpH} + \mathrm{DKla})\\
&\qquad\qquad + (\mathrm{HDhum} * \mathrm{KGmhMMUc} * \mathrm{DKla})\\
&\qquad\qquad + (\mathrm{HDhum} * \mathrm{WFeqp} * \mathrm{KGunEQP}) * (\mathrm{DKprH} + \mathrm{DKmpH} + \mathrm{DKla})))\\
&\qquad + (2 * \mathrm{DpdHUM})\\
&\qquad + ((1 / \mathrm{NDstint}) * \mathrm{DPla})\\
&\qquad + (\mathrm{HDhum} * \mathrm{NhmsHUM} * \mathrm{NgscHMS} * \mathrm{DphGSC})\\
&\qquad + (\mathrm{HDhum} * \mathrm{DlhTDR}))
\end{aligned}
$$

Number-of-teleoperators costs (functions of NtotTELi):

$$
\begin{aligned}
&(\mathrm{DKprH} * \mathrm{KGunTEL}) * (1 + \mathrm{FRlaINS})\\
&\quad * (1^{-q} + 2^{-q} + \dots + \mathrm{trunc}(\mathrm{NtotTELi})^{-q}\\
&\qquad + (\mathrm{NtotTELi} - \mathrm{trunc}(\mathrm{NtotTELi})) * \mathrm{NtotTELi}^{-q})
\end{aligned}
$$

$$\text{where } NtotTELi = \sum_{j=1}^{NTSK} Ntel[j]$$

$$+ DUprTCS * (1^{-q} + 2^{-q} + \ldots + trunc(NtotTCSi)^{-q}$$
$$+ (NtotTCSi - trunc(NtotTCSi)) * NtotTCSi^{-q})$$
$$\text{where } NtotTCSi = NtotTELi * NtcsTEL$$

$$+ NtotTELi$$
$$* ((KGunTEL * (DKmpH + DK1a) * (1 + FR1aINS))$$
$$+ ((NtcsTEL * HDtel * NgscTCS / 8)$$
$$* (DpGSCt + (NdHUMt * 8 * DphGSC))))$$

Teleoperator usage costs (functions of NtotTD):

$$(DKprH * KGunTEL) * (1 + FR1aINS)$$
$$* (1^{-q} + 2^{-q} + \ldots + trunc(NtotTELs)^{-q}$$
$$+ (NtotTELs - trunc(NtotTELs)) * NtotTELs^{-q})$$
$$\text{where } NtotTELs = NtotTD * (HDtel * WFtel)$$

$$\text{and } NtotTD = \sum_{j=1}^{NTSK} Ntel[j] * Ttime[j]$$

$$+ DUprTCS * (1^{-q} + 2^{-q} + \ldots + trunc(NtotTCSs)^{-q}$$
$$+ (NtotTCSs - trunc(NtotTCSs)) * NtotTCSs^{-q})$$
$$\text{where } NtotTCSs = NtotTD * (HDtel * NtcsTEL * WFtcs)$$

$$+ NtotTD$$
$$* ((HDtel * WFtel * KGunTEL * (DKmpH + DK1a) * (1 + FR1aINS))$$
$$+ (HDtel * KGthTELc * DK1a * (1 + FR1aINS))$$
$$+ (HDtel * NtcsTEL * NgscTCS * DphGSC)$$
$$+ (HDtel * DhhTDR))$$

Number-of-automated-equipment-units costs (functions of NtotAEQi):

$$\begin{aligned}&(\mathrm{DKprH} * \mathrm{KGunAEQ}) * (1 + \mathrm{FR1aINS})\\ &\quad * (1^{-q} + 2^{-q} + \ldots + \mathrm{trunc}(\mathrm{NtotAEQi})^{-q}\\ &\qquad + (\mathrm{NtotAEQi} - \mathrm{trunc}(\mathrm{NtotAEQi})) * \mathrm{NtotAEQi}^{-q})\end{aligned}$$

$$\text{where } \mathrm{NtotAEQi} = \sum_{j=1}^{\mathrm{NTSK}} \mathrm{Naeq}[j]$$

$$\begin{aligned}&+ \mathrm{DUprAMS} * (1^{-q} + 2^{-q} + \ldots + \mathrm{trunc}(\mathrm{NtotAMSi})^{-q}\\ &\quad + (\mathrm{NtotAMSi} - \mathrm{trunc}(\mathrm{NtotAMSi})) * \mathrm{NtotAMSi}^{-q})\\ &\qquad\qquad \text{where } \mathrm{NtotAMSi} = \mathrm{NtotAEQi} * \mathrm{NamsAEQ}\end{aligned}$$

$$\begin{aligned}&+ \mathrm{NtotAEQi}\\ &\quad * ((\mathrm{KGunAEQ} * (\mathrm{DKmpH} + \mathrm{DK1a}) * (1 + \mathrm{FR1aINS}))\\ &\qquad + ((\mathrm{NamsAEQ} * \mathrm{HDaeq} * \mathrm{NgscAMS} / 8)\\ &\qquad\quad * (\mathrm{DpGSCt} + (\mathrm{NdHUMt} * 8 * \mathrm{DphGSC}))))\end{aligned}$$

Automated equipment usage costs (functions of NtotAD):

$$\begin{aligned}&(\mathrm{DKprH} * \mathrm{KGunAEQ}) * (1 + \mathrm{FR1aINS})\\ &\quad * (1^{-q} + 2^{-q} + \ldots + \mathrm{trunc}(\mathrm{NtotAEQs})^{-q}\\ &\qquad + (\mathrm{NtotAEQs} - \mathrm{trunc}(\mathrm{NtotAEQs})) * \mathrm{NtotAEQs}^{-q})\\ &\qquad\qquad \text{where } \mathrm{NtotAEQs} = \mathrm{NtotAD} * (\mathrm{HDaeq} * \mathrm{WFaeq})\end{aligned}$$

$$\text{where } \mathrm{NtotAD} = \sum_{j=1}^{\mathrm{NTSK}} \mathrm{Naeq}[j] * \mathrm{Ttime}[j]$$

$$\begin{aligned}&+ \mathrm{DUprAMS} * (1^{-q} + 2^{-q} + \ldots + \mathrm{trunc}(\mathrm{NtotAMSs})^{-q}\\ &\quad + (\mathrm{NtotAMSs} - \mathrm{trunc}(\mathrm{NtotAMSs})) * \mathrm{NtotAMSs}^{-q})\\ &\qquad\qquad \text{where } \mathrm{NtotAMSs} = \mathrm{NtotAD} * (\mathrm{HDaeq} * \mathrm{NamsAEQ} * \mathrm{WFams})\end{aligned}$$

```
+ NtotAD
     * ((HDaeq * WFaeq * KGunAEQ * (DKmpH + DKla) * (1 + FRlaINS))

        + (HDaeq * NamsAEQ * NgscAMS * DphGSC)

        + (HDaeq * DlhTDR))
```

Whole-job timecosts (functions of Jobtime):

```
            -q   -q                        -q
DUprGMS * (1  + 2  + ... + trunc(NtotGMSs)
                                               -q
     + (NtotGMSs - trunc(NtotGMSs)) * NtotGMSs   )
               where NtotGMSs = Jobtime * (NtotGMSi * 24 * WFgms)
                                          where NtotGMSi = NgmsGEN
+ Jobtime
     * ((KDcsp * DKla * (1 + FRlaINS))

        + (NgmsGEN * 24 * NgscGMS * DphGSC)

        + (72 * DlhTDR))
```

G.4 Cost Equation, Without Certain Learning Curves:

The cost equation in section G.3 includes learning curves in some of the procurement terms. These terms couple decision variables in expressions with fractional exponents. For example, the learning curve terms dependent on NtotHUMi have the form:

$$\text{(a constant)} * (1^{-q} + 2^{-q} + \ldots + \text{trunc}(\#\text{units})^{-q}$$

$$+ (\#\text{units} - \text{trunc}(\#\text{units})) * (\#\text{units})^{-q})$$

$$\text{where } (\#\text{units}) = \text{(another constant)} * (\text{NtotHUMi})$$

$$= \text{number of units to be procured}$$

When expanded, these learning curve series will include the term:

$$\text{NtotHUMi}^{-q}$$

$$= (\text{Nhum}[1] + \text{Nhum}[2] + \ldots + \text{Nhum}[j] + \ldots)^{-q}$$

where $-q = -0.32193$ for an 80% learning curve. This type of nonlinearity is very difficult to deal with in formal optimization, and usually requires numerical methods. But learning curves are useful because they improve the realism of the cost model, and because they are representative of nonlinear models for individual line items in a program.

Fortunately, these learning curves have a relatively small impact on overall cost behavior. In the case example in Chapter 4, it will be seen that most hardware procurements involve small numbers of units (e.g. 2.1 teleoperators, 2.7 manned maneuvering units), and therefore their learning curves do not have a major effect on overall program costs.

The exception is the procurement of the large quantity of space structure to be assembled: 3,804 low-tech units and 600 high-tech units in the case example. The learning curve has an important effect on this cost element. However, the space structure procurement contributes only to the constant term Cw, and therefore its learning curve does not affect the optimization process.

If the learning curve parameter is set at 100% (no learning) then $-q = 0$, and the general learning curve form above reduces to:

$$\text{(a constant)} * (\#\text{units})$$

$$\text{where } (\#\text{units}) = \text{(another constant)} * (\text{NtotHUMi})$$

This is the case when each item is bought at a fixed price, with no economies of mass production.

The input lcparam is the learning curve parameter for hardware other than space structure. If lcparam is set to 100%, the overall cost equation is greatly simplified. In this section, this process is carried out, and the simplified procurement terms are combined with the other terms in their categories. The space structure procurement terms are left unchanged.

Whole-job R&D costs = Rw:

$$(\text{DKrdH} * \text{KGunSSTh}) * (1 + (\text{NunSSTh} * 0.035))$$

$$+ (\text{DKrdL} * \text{KGunSSTl}) * (1 + (\text{NunSSTl} * 0.035))$$

$$+ \text{DKrdL} * \text{KGunCSE}$$

$$+ \text{CTrdTSK}$$

$$+ \text{CTrdGMS}$$

Human-related R&D costs = Rh:

$$((\text{DKrdH} * \text{KGunHABh}) + (\text{DKrdL} * \text{KGunHABl})) * \text{YSNOhum}$$

$$+ (\text{CTrdEMU} + \text{CTrdMMU} + (\text{DKrdH} * \text{KGunEQP})) * \text{YSNOhum}$$

$$+ (\text{CTrdHMS} * \text{YSNOhum})$$

Teleoperator-related R&D costs = Rt:

$$(\text{DKrdH} * \text{KGunTEL}) * \text{YSNOtel}$$

$$+ (\text{CTrdTCS} * \text{YSNOtel})$$

Automated-equipment-related R&D costs = Ra:

$$(\text{DKrdH} * \text{KGunAEQ}) * \text{YSNOaeq}$$

$$+ (\text{CTrdAMS} * \text{YSNOaeq})$$

Whole-job general costs = Cw:

$$(\text{DKprH} * \text{KGunSSTh}) * (1 + \text{FRlaINS})$$

$$* (1^{-p} + 2^{-p} + \ldots + \text{trunc}(\text{NunSSTh})^{-p}$$

$$+ (\text{NunSSTh} - \text{trunc}(\text{NunSSTh})) * \text{NunSSTh}^{-p})$$

$$\text{where } -p = (1.4427) * (\text{LN}(\text{lcpmSST}/100)$$

```
+ (NunSSTh * KGunSSTh) * (DKmpH + DKla) * (1 + FRlaINS)
                                                {For DensSSTh >= 100}
                    or
                    + (NunSSTh * KGunSSTh)
                      * (DKmpH + (DMla / DensSSTh))
                      * (1 + FRlaINS)
                                                 {For DensSSTh < 100}

+ (DKprL * KGunSSTl) * (1 + FRlaINS)
         -p   -p                     -p
    * (1   + 2   + ... + trunc(NunSSTl)
                                             -p
        + (NunSSTl - trunc(NunSSTl)) * NunSSTl   )

+ (NunSSTl * KGunSSTl) * (DKmpL + DKla) * (1 + FRlaINS)
                                                {For DensSSTl >= 100}
                    or
                    + (NunSSTl * KGunSSTl)
                      * (DKmpL + (DMla / DensSSTl))
                      * (1 + FRlaINS)
                                                 {For DensSSTl < 100}

+ (NtotCSE * KGunCSE) * (DKprL + DKmpL + DKla) * (1 + FRlaINS)

+ DUprGMS * NgmsGEN

+ (NgmsGEN * NgscGMS * (24/8)) * (DpGSCt + (NdHUMt * 8 * DphGSC))
```

Number-of-humans costs = (NtotHUMi * Ch):

```
NtotHUMi
     * (((1 / NhumHAB) * KGunHABh * (DKmpH + DKla) * (1 + FRlaINS))
       + ((1 / NhumHAB) * KGunHABl * (DKmpL + DKla) * (1 + FRlaINS))

       + (((2 * KGunEMU) + KGunMMU + KGunEQP)
          * (DKprH + DKmpH + DKla) * (1 + FRlaINS))

       + (2 * DpHUMt)

       + (2 * NdHUMt * DpdHUM)

       + ((NhmsHUM * HDhum * NgscHMS / 8)
          * (DpGSCt + (NdHUMt * 8 * DphGSC)))

       + (((DKprH * KGunHABh) + (DKprL * KGunHABl))
          * (1 + FRlaINS) * (1 / NhumHAB))

       + NhmsHUM * DUprHMS)
```

$$\text{where NtotHUMi} = \sum_{j=1}^{NTSK} Nhum[j]$$

Human usage costs = (NtotPD * Hh):

```
NtotPD
     * (((1 + FRlaINS)
         * (((1 / NhumHAB) * WFhab * KGunHABh * (DKmpH + DKla))
            + (KGpdHABc * DKla)))

        + ((1 + FRlaINS)
           * ((HDhum * WFemu * KGunEMU) * (DKprH + DKmpH + DKla)
              + (HDhum * KGehEMUc * DKla)
              + (HDhum * WFmmu * KGunMMU) * (DKprH + DKmpH + DKla)
              + (HDhum * KGmhMMUc * DKla)
              + (HDhum * WFeqp * KGunEQP) * (DKprH + DKmpH + DKla)))

        + (2 * DpdHUM)

        + ((1 / NDstint) * DPla)

        + (HDhum * NhmsHUM * NgscHMS * DphGSC)

        + (HDhum * DlhTDR)

        + ((WFhab / NhumHAB) * (DKprH * KGunHABh) * (1 + FRlaINS))

        + ((HDhum * NhmsHUM * WFhms) * DUprHMS))
```

$$\text{where NtotPD} = \sum_{j=1}^{\text{NTSK}} \text{Nhum}[j] * \text{Ttime}[j]$$

Number-of-teleoperators costs = (NtotTELi * Ct):

```
NtotTELi
     * ((KGunTEL * (DKmpH + DKla) * (1 + FRlaINS))

        + ((NtcsTEL * HDtel * NgscTCS / 8)
              * (DpGSCt + (NdHUMt * 8 * DphGSC)))

        + ((DKprH * KGunTEL) * (1 + FRlaINS))

        + (NtcsTEL * DUprTCS))
```

$$\text{where NtotTELi} = \sum_{j=1}^{\text{NTSK}} \text{Ntel}[j]$$

Teleoperator usage costs = (NtotTD * Ht):

```
NtotTD
     * ((HDtel * WFtel * KGunTEL * (DKmpH + DKla) * (1 + FRlaINS))

        + (HDtel * KGthTELc * DKla * (1 + FRlaINS))

        + (HDtel * NtcsTEL * NgscTCS * DphGSC)

        + (HDtel * DhhTDR)

        + (HDtel * WFtel * (DKprH * KGunTEL) * (1 + FRlaINS))

        + ((HDtel * NtcsTEL * WFtcs) * DUprTCS))
```

$$\text{and NtotTD} = \sum_{j=1}^{NTSK} \text{Ntel}[j] * \text{Ttime}[j]$$

Number-of-automated-equipment-units costs = (NtotAEQi * Ca):

```
NtotAEQi
     * ((KGunAEQ * (DKmpH + DKla) * (1 + FRlaINS))

        + ((NamsAEQ * HDaeq * NgscAMS / 8)
           * (DpGSCt + (NdHUMt * 8 * DphGSC)))

        + ((DKprH * KGunAEQ) * (1 + FRlaINS))

        + (NamsAEQ * DUprAMS))
```

$$\text{where NtotAEQi} = \sum_{j=1}^{NTSK} \text{Naeq}[j]$$

Automated equipment usage costs = (NtotAD * Ha):

```
NtotAD
     * ((HDaeq * WFaeq * KGunAEQ * (DKmpH + DKla) * (1 + FRlaINS))

        + (HDaeq * NamsAEQ * NgscAMS * DphGSC)

        + (HDaeq * DlhTDR)

        + (HDaeq * WFaeq * (DKprH * KGunAEQ) * (1 + FRlaINS))

        + ((HDaeq * NamsAEQ * WFams) * DUprAMS))
```

$$\text{where NtotAD} = \sum_{j=1}^{\text{NTSK}} \text{Naeq[j]} * \text{Ttime[j]}$$

<u>Whole-job timecosts = (Jobtime * Hw):</u>

```
Jobtime
     * ((KDcsp * DKla * (1 + FRlaINS))

        + (NgmsGEN * 24 * NgscGMS * DphGSC)

        + (72 * DlhTDR)

        + ((NgmsGEN * 24 * WFgms) * DUprGMS))
```

APPENDIX H:

LINE-ITEM COST MODEL: COMPUTER PROGRAMS

H.1 General Notes:

This study's line-item cost model is implemented as a set of computer programs. The overall structure of these programs is shown in Figure H.1.

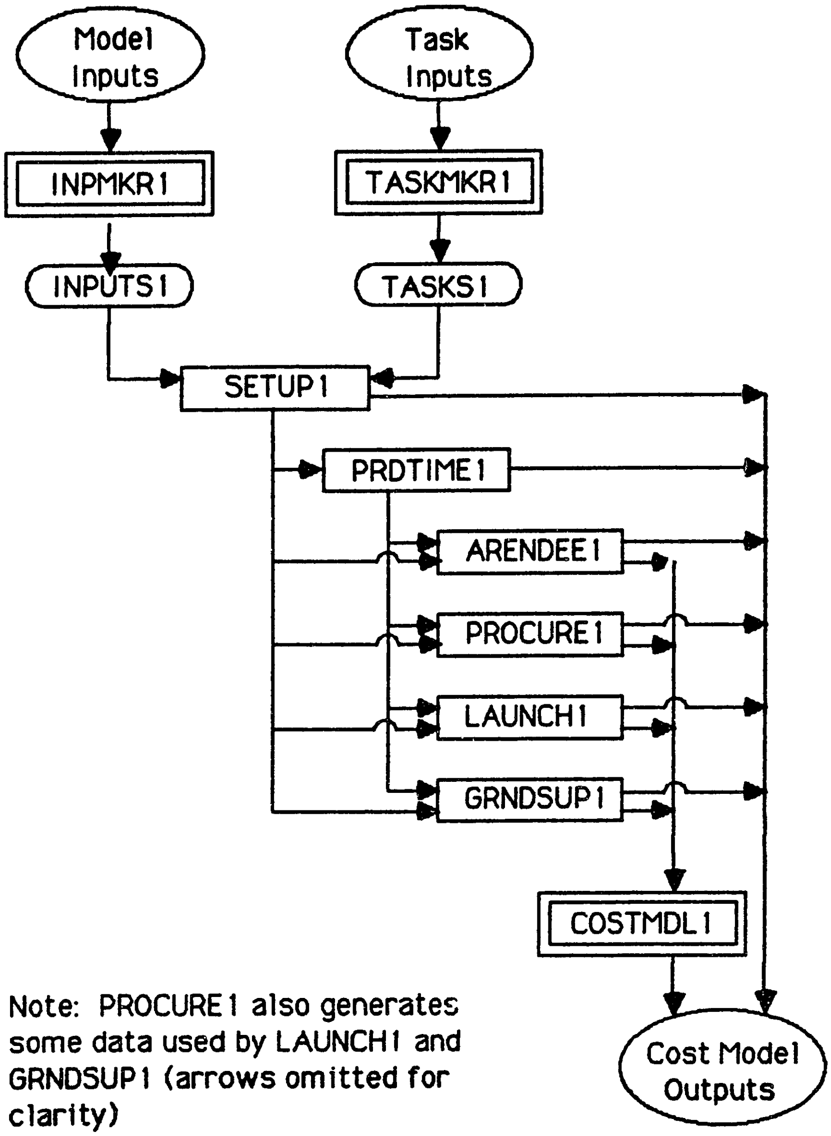


FIGURE H.1: STRUCTURE OF COMPUTER PROGRAMS FOR LINE-ITEM COST MODEL

Three main programs are used in the model. Programs INPMKR1 and TASKMKR1 load the model inputs into the input files INPUTS1 and TASKS1. The program COSTMDL1 runs the line-item costing algorithms, using subroutines SETUP1, PRDTIME1, ARENDEE1, PROCURE1, LAUNCH1, and GRNDSUP1. Various parts of the model contribute to the intermediate and final cost model outputs.

These programs are written in UCSD Pascal and run on a Corvus Concept workstation. The Corvus Concept has its own Corvus Concept Operating System, and requires unit structures and segmentation to compile large programs. The associated commands are described later in this appendix.

The following sections list the cost model programs. Because the costing algorithms are discussed in detail in Appendix B, only minimal comments are included in the programs here.

H.2 Program INPMKR1:

Program INPMKR1 ("input-maker 1") loads most of the cost model's inputs into file INPUTS1. Each input is a record, including record number, variable name, numerical value, units, and a brief description of the input. Because of the large size of INPMKR1, it is internally divided into three "procedures" (Pascal name for subroutines) INPMKR11, INPMKR12, and INPMKR13.

Each of these procedures first defines the values of a set of input records, then writes these records into INPUTS1. The input records are also printed out by procedures RECPRINT and WFPRINT, which differ only in the precision of the displayed numerical values. For brevity, some of the definition and printout statements are omitted from this listing.

```
PROGRAM INPMKR1;

   TYPE
      inputrec =
         RECORD
            recnumber : integer;
            varname : STRING[8];
            varvalue : real;
            varunit : STRING[30];
            vardescrip : STRING[80];
         END;
      inputfile = FILE OF inputrec;
   VAR
      INPUTS1 : inputfile;
      temprec : inputrec;
      blankrec : inputrec;


PROCEDURE RECPRINT;
   BEGIN     {RECPRINT}
      WITH temprec DO
         BEGIN
            write(recnumber:2,': ');
            writeln(varname,' = ',varvalue:3:1,' ', varunit);
            writeln('    Description: ',vardescrip);
            writeln;
         END;
   END;   {RECPRINT}


PROCEDURE WFPRINT;
   BEGIN     {WFPRINT}
      WITH temprec DO
         BEGIN
            write(recnumber:2,': ');
            writeln(varname,' = ',varvalue:8:6,' ', varunit);
            writeln('    Description: ',vardescrip);
            writeln;
         END;
   END;   {WFPRINT}


PROCEDURE INPMKR11;
   VAR
      number : integer;
      Scenario : inputrec;
      NunSSTh, KGunSSTh, DensSSTh, NunSSTl, KGunSSTl,
      DensSSTl, lcpmSST : inputrec;
      DKrdH, DKrdL, CTrdTSK : inputrec;
      DKprH, DKprL : inputrec;
      DKmpH, DKmpL, DKla, DMla, DPla, FRlaINS : inputrec;

   BEGIN      {INPMKR11}

      WITH Scenario DO
         BEGIN
            recnumber := 1;
            varname := 'Scenario';
            varvalue := 1;
            varunit := '      ';
            vardescrip := 'Humans, teleops, and auto. eq. for ',
                          'construction';
         END;
```

```
WITH NunSSTh DO
   BEGIN
      recnumber := 2;
      varname := 'NunSSTh';
      varvalue := 600;
      varunit := 'units';
      vardescrip := 'Number of units of high-tech space ',
                    'structure';
   END;

       o

       o   (lots more record definitions)

       o

WITH FRlaINS DO
   BEGIN
      recnumber := 26;
      varname := 'FRlaINS';
      varvalue :=  0.05;
      varunit := '$/$';
      vardescrip := 'Fraction of proc. and launch cost ',
                    'paid as ins. premium';
   END;

rewrite(INPUTS1,'/VDAVE/INPUTS1');
write(INPUTS1, blankrec, Scenario,
      NunSSTh, KGunSSTh, DensSSTh,
      NunSSTl, KGunSSTl, DensSSTl, lcpmSST,
      blankrec, blankrec,
      DKrdH, DKrdL, CTrdTSK,
      blankrec, blankrec,
      DKprH, DKprL,
      blankrec, blankrec, blankrec,
      DKmpH, DKmpL, DKla, DMla, DPla, FRlaINS,
      blankrec, blankrec);
close(INPUTS1, lock);

writeln;
writeln;
writeln('         DATA GOING INTO INPUTS1');
writeln('           (PROCEDURE INPMKR11)');
writeln;

reset(INPUTS1,'/VDAVE/INPUTS1');
seek(INPUTS1,1);
   FOR number := 1 TO 25 DO
      BEGIN
         seek (INPUTS1,number);
         read (INPUTS1,temprec);
         RECPRINT;
      END;
```

```
         read (INPUTS1,temprec);          {26}
         WFPRINT;

         FOR number := 27 TO 28 DO
            BEGIN
               seek (INPUTS1,number);
               read (INPUTS1,temprec);
               RECPRINT;
            END;
      close (INPUTS1,lock);

   END;       {INPMKR11}


PROCEDURE INPMKR12;

   VAR
      number : integer;
      CTrdGMS, NgmsGEN, WFgms, DUprGMS, NgscGMS,
      CTrdHMS, NhmsHUM, WFhms, DUprHMS, NgscHMS,
      CTrdTCS, NtcsTEL, WFtcs, DUprTCS, NgscTCS,
      CTrdAMS, NamsAEQ, WFams, DUprAMS, NgscAMS,
      DpGSCt, DphGSC : inputrec;

   BEGIN      {INPMKR12}

      WITH CTrdGMS DO
         BEGIN
            recnumber := 31;
            varname := 'CTrdGMS';
            varvalue :=  3000000;
            varunit := 'dollars';
            vardescrip := 'Total cost of R&D of general ',
                          'monitoring stations';
         END;

             o

             o   (lots more record definitions)

             o


      WITH DphGSC DO
         BEGIN
            recnumber := 52;
            varname := 'DphGSC';
            varvalue :=  35;
            varunit := '$/gsc-hr';
            vardescrip :=
               'Dollars/hour salary and overhead for one GSC ',
               'member';
         END;
      reset(INPUTS1,'/VDAVE/INPUTS1');
      seek(INPUTS1,29);
      write(INPUTS1,
            blankrec, blankrec,
            CTrdGMS, NgmsGEN, WFgms, DUprGMS, NgscGMS,
            CTrdHMS, NhmsHUM, WFhms, DUprHMS, NgscHMS,
            CTrdTCS, NtcsTEL, WFtcs, DUprTCS, NgscTCS,
            CTrdAMS, NamsAEQ, WFams, DUprAMS, NgscAMS,
            DpGSCt, DphGSC,
            blankrec, blankrec, blankrec, blankrec, blankrec,
            blankrec);
      close(INPUTS1, lock);

      writeln;
      writeln('          DATA GOING INTO INPUTS1');
      writeln('            (PROCEDURE INPMKR12)');
      writeln;

      reset(INPUTS1,'/VDAVE/INPUTS1');
         seek (INPUTS1,29);
         FOR number := 29 TO 32 DO
            BEGIN
               read (INPUTS1,temprec);
               RECPRINT;
            END;

         read (INPUTS1,temprec);            {33}
         WFPRINT;

                o

                o    (lots more records printed out)

                o

         FOR number := 49 TO 58 DO
            BEGIN
               read (INPUTS1,temprec);
               RECPRINT;
            END;
      close (INPUTS1,lock);

   END;        {INPMKR12}


PROCEDURE INPMKR13;

   VAR
      number : integer;
      HDhum, NhumHAB, KGunHABh, KGunHABl, WFhab, KGpdHABc,
      CTrdEMU, KGunEMU, WFemu, KGehEMUc,
      CTrdMMU, KGunMMU, WFmmu, KGmhMMUc,
   KGunEQP, WFeqp, DpHUMt, DpdHUM, NdHUMt, NDstint : inputrec;
   HDtel, KGunTEL, WFtel, KGthTELc : inputrec;
   HDaeq, KGunAEQ, WFaeq : inputrec;
   NtotCSE, KGunCSE, KDcsp, DlhTDR, DhhTDR, lcparam : inputrec;

BEGIN      {INPMKR13}

   WITH HDhum DO
      BEGIN
         recnumber := 59;
         varname := 'HDhum';
         varvalue :=  7;
         varunit := 'hrs/day';
         vardescrip := 'Hours/day worked by humans in space';
      END;

          o

          o   (lots more record definitions)

          o

   WITH lcparam DO
      BEGIN
         recnumber := 102;
         varname := 'lcparam';
         varvalue :=  90;
         varunit := '%';
         vardescrip := 'Learning curve param. for hardware ',
                       'other than SST';
      END;

   reset(INPUTS1,'/VDAVE/INPUTS1');
   seek(INPUTS1,59);
   write(INPUTS1,
         HDhum, NhumHAB, KGunHABh, KGunHABl, WFhab, KGpdHABc,
         CTrdEMU, KGunEMU, WFemu, KGehEMUc,
         CTrdMMU, KGunMMU, WFmmu, KGmhMMUc,
         KGunEQP, WFeqp, DpHUMt, DpdHUM, NdHUMt, NDstint,
         blankrec, blankrec,
         HDtel, KGunTEL, WFtel, KGthTELc,
         blankrec, blankrec, blankrec, blankrec, blankrec,
         blankrec,
         HDaeq, KGunAEQ, WFaeq,
         blankrec, blankrec, blankrec,
         NtotCSE, KGunCSE, KDcsp, DlhTDR, DhhTDR, lcparam,
         blankrec);
   close(INPUTS1, lock);
   writeln;
   writeln('        DATA GOING INTO INPUTS1');
   writeln('          (PROCEDURE INPMKR13)');
   writeln;
```

```
      reset(INPUTS1,'/VDAVE/INPUTS1');
      seek(INPUTS1,59);
         FOR number := 59 TO 62 DO
            BEGIN
               read (INPUTS1,temprec);
               RECPRINT;
            END;

         read (INPUTS1,temprec);          {63}
         WFPRINT;

                o

                o   (lots more records printed out)

                o

         FOR number := 94 TO 103 DO
            BEGIN
               read (INPUTS1,temprec);
               RECPRINT;
            END;
      close (INPUTS1,lock);

   END;       {INPMKR13}


BEGIN    {INPMKR1}

   WITH blankrec DO
      BEGIN
         recnumber := 0;
         varname := 'blankrec';
         varvalue := 0;
         varunit := '      ';
         vardescrip := 'Blank input record';
      END;

   INPMKR11;
   INPMKR12;
   INPMKR13;

END.     {INPMKR1}
```

H.3 Program TASKMKR1:

Program TASKMKR1 ("task-maker 1") loads task-related model inputs into file TASKS1. Each input is a record, including the task number, a brief description of the task, the number of repetitions of the task to be performed, and the numbers and productivities of humans, teleoperators, and units of automated equipment assigned to the task.

```
PROGRAM TASKMKR1;

   TYPE
      taskrec =
         RECORD
            tasknumber : integer;
            taskdescrip : STRING[100];
            repsintask : real;
            numberhum : real;
            prodhum : real;
            numbertel : real;
            prodtel : real;
            numberaeq : real;
            prodaeq : real;
         END;
      taskfile = FILE OF taskrec;
      index = 1..20;

   VAR
      TASKS1 : taskfile;
      temptask : taskrec;

      NTSK : taskrec;
      task1, task2, task3, task4 : taskrec;

      j : index;

   PROCEDURE TASKPRINT;
      BEGIN     {TASKPRINT}
         WITH temptask DO
            BEGIN
               writeln('Task number ',tasknumber:1,':   ',
                                                taskdescrip);
               writeln('Number of repetitions in task = ',
                                                repsintask:7:1);
               write('Number of humans = ',numberhum:5:2);
               writeln('   with productivity = ',prodhum:6:3,
                                                ' reps/pers-hr');
               write('Number of teleops = ',numbertel:5:2);
               writeln('   with productivity = ',prodtel:6:3,
                                                ' reps/tel-hr');
               write('Number of units of AEQ = ',numberaeq:5:2);
               writeln('   with productivity = ',prodaeq:6:3,
                                                ' reps/aeq-hr');
               writeln;
            END;
      END;   {TASKPRINT}

   BEGIN      {TASKMKR1}

      WITH NTSK DO
         BEGIN
      tasknumber:= 4;
      taskdescrip := 'Number of tasks to be performed';
      repsintask := 0;
      numberhum := 0;
      prodhum := 0;
      numbertel := 0;
      prodtel := 0;
      numberaeq := 0;
      prodaeq := 0;
   END;

WITH task1 DO
   BEGIN
      tasknumber:= 1;
      taskdescrip := 'Manufacture of truss beams';
      repsintask := 16688;
      numberhum := 0.6;
      prodhum := 1.3;
      numbertel := 0.4;
      prodtel := 1.0;
      numberaeq := 4.3;
      prodaeq := 3.7;
   END;

       o

       o   (more task records defined)

       o


WITH task4 DO
   BEGIN
      tasknumber:= 4;
      taskdescrip := 'Installation of feed horn assemblies';
      repsintask := 600;
      numberhum := 0.4;
      prodhum := 5.6;
      numbertel := 0.3;
      prodtel := 4.8;
      numberaeq := 0.0;
      prodaeq := 0.001;
   END;

rewrite(TASKS1,'/VDAVE/TASKS1');
seek(TASKS1,0);
write(TASKS1, NTSK, task1, task2, task3, task4);
close(TASKS1, lock);
writeln;
writeln;
writeln('         DATA GOING INTO TASKS1');
writeln;
```

```
   reset(TASKS1,'/VDAVE/TASKS1');
      read(TASKS1,temptask);
      writeln('Total number of tasks NTSK = ',
                                            NTSK.tasknumber:2);
      writeln;

      FOR j:=1 TO NTSK.tasknumber DO
         BEGIN
            seek(TASKS1,j);
            read(TASKS1,temptask);
            TASKPRINT;
         END;
   close (TASKS1,lock);

END.    {TASKMKR1}
```

H.4 Program COSTMDL1:

Program COSTMDL1 ("cost model 1") calculates and displays the line-item costs of the space scenario under study. To do this, the program calls the procedures SETUP1, PRDTIME1, ARENDEE1, PROCURE1, LAUNCH1, and GRNDSUP1. Since COSTMDL1 and these procedures are too large to be compiled all together, each procedure is in a separate "unit". These units are compiled by themselves, and "used" by the main program.

This section presents the main program only. The following sections list the procedures.

```
PROGRAM COSTMDL1;

   USES {$U SETUP1.U} SETUP1U,
        {$U PRDTIME1.U} PRDTIM1U,
        {$U ARENDEE1.U} ARENDE1U,
        {$U PROCURE1.U} PROCUR1U,
        {$U LAUNCH1.U} LAUNCH1U,
        {$U GRNDSUP1.U} GRDSUP1U;

   VAR
      ProgCost : real;

   BEGIN   {COSTMDL1}
      writeln;
      writeln('               OUTPUT OF PROGRAM COSTMDL1');

      writeln;
      writeln('INPUT DATA:');

      writeln;
      SETUP1;
```

```
writeln;
writeln('INTERMEDIATE OUTPUTS:');

writeln;
PRDTIME1;

ARENDEE1;

PROCURE1;
writeln;

LAUNCH1;
writeln;

GRNDSUP1;

ProgCost := CTrd + CTpr + CTla + CTgs;

writeln;
writeln('        LINE-ITEM COSTS FOR CONSTRUCTION SCENARIO');
writeln('               (Costs in $Million (1984$))');
writeln;

write('Total R&D cost (CTrd):                           ');
writeln(CTrd/1E6:9:3);
write('   Space structure to be assembled (CTrdSST):',
                                               '         ');
writeln(CTrdSST/1E6:9:3);
write('   Human habitation modules (CTrdHAB):',
                                   '                ');
writeln(CTrdHAB/1E6:9:3);
write('   Human support equipment (CTrdHSE):',
                                   '                 ');
writeln(CTrdHSE/1E6:9:3);
write('   Teleoperators (CTrdTEL):',
                            '                           ');
writeln(CTrdTEL/1E6:9:3);
write('   Automated equipment (CTrdAEQ):',
                                   '                     ');
writeln(CTrdAEQ/1E6:9:3);
write('   Construction support equipment (CTrdCSE):',
                                         '          ');
writeln(CTrdCSE/1E6:9:3);
write('   Task procedures (CTrdTSK):',
                            '                         ');
writeln(CTrdTSK/1E6:9:3);
write('   Ground support equipment (CTrdGSE):',
                                   '                ');
writeln(CTrdGSE/1E6:9:3);
writeln;
```

```
write('Total procurement cost (CTpr):                   ');
writeln(CTpr/1E6:9:3);
write('   Space structure to be assembled (CTprSST):',
                                         '         ');
writeln(CTprSST/1E6:9:3);
write('   Human habitation modules (CTprHAB):',
                                   '                 ');
writeln(CTprHAB/1E6:9:3);
write('   Human support equipment (CTprHSE):',
                                   '                 ');
writeln(CTprHSE/1E6:9:3);
write('   Teleoperators (CTprTEL):',
                      '                           ');
writeln(CTprTEL/1E6:9:3);
write('   Automated equipment (CTprAEQ):',
                               '                     ');
writeln(CTprAEQ/1E6:9:3);
write('   Construction support equipment (CTprCSE):',
                                         '          ');
writeln(CTprCSE/1E6:9:3);
write('   Training of humans in space crews (CTprHUMt):',
                                               '      ');
writeln(CTprHUMt/1E6:9:3);
write('   Salary and overhead for humans in space ',
                                         '(CTprHUMb):');
writeln(CTprHUMb/1E6:9:3);
write('   Ground support equipment (CTprGSE):',
                                   '                ');
writeln(CTprGSE/1E6:9:3);
writeln;

write('Total mission prep. and launch cost (CTla):      ');
writeln(CTla/1E6:9:3);
write('   Space structure to be assembled (CTlaSST):',
                                         '         ');
writeln(CTlaSST/1E6:9:3);
write('   Launch of humans in space crews (CTlaHUM):',
                                         '         ');
writeln(CTlaHUM/1E6:9:3);
write('   Human habitation modules (CTlaHAB):',
                                   '                ');
writeln(CTlaHAB/1E6:9:3);
write('   Human support equipment (CTlaHSE):',
                                   '                 ');
writeln(CTlaHSE/1E6:9:3);
write('   Teleoperators (CTlaTEL):',
                       '                           ');
writeln(CTlaTEL/1E6:9:3);
write('   Automated equipment (CTlaAEQ):',
                           '                     ');
writeln(CTlaAEQ/1E6:9:3);
```

```
      write('   Construction support equipment (CTlaCSE):',
                                                '         ');
      writeln(CTlaCSE/1E6:9:3);
      write('   Construction site propellant (CTlaCSP):',
                                                '           ');
      writeln(CTlaCSP/1E6:9:3);
      write('   Launch insurance (CTlaINS):',
                                '                       ');
      writeln(CTlaINS/1E6:9:3);
      writeln;

      write('Total ground support cost (CTgs):                   ');
      writeln(CTgs/1E6:9:3);
      write('   Training of ground support crew (CTgsGSCt):',
                                                      '       ');
      writeln(CTgsGSCt/1E6:9:3);
      write('   Salary & overhead for ground sup. crew ',
                                               '(CTgsGSCb): ');
      writeln(CTgsGSCb/1E6:9:3);
      write('   User fees for tracking and data relay ',
                                               '(CTgsTDR):   ');
      writeln(CTgsTDR/1E6:9:3);

      writeln('                                                  ',
                                               '    ----------');
      write('Total program cost (ProgCost):                      ');
      writeln(ProgCost/1E6:9:3);

   END.       {COSTMDL1}
```

H.5 Procedure SETUP1:

Procedure SETUP1 sets up the cost modeling process by reading input values from files INPUTS1 and TASKS1 into global variables. These variables are then available to all parts of the cost model, because they are declared in the INTERFACE section of SETUP1. Procedure SETUP1 also prints out these inputs as it reads them. For brevity, some of the input reading and printout statements are omitted from this listing.

SETUP1 also declares other global variables, which are used to pass calculated values between various parts of the model. For example, masses of consumables calculated in PROCURE1 are passed to LAUNCH1 in this fashion.

The {$S SEGONE} command is a segmentation command. Procedure SETUP1 is assigned to a separate segment than COSTMDL1 and the other procedures (which are in the default segment; no command required). This is done to avoid overloading the Linker. Procedure SETUP1 is also nested inside UNIT SETUP1U, which allows it to be compiled by itself.

```
{$S SEGONE}

UNIT SETUP1U;

INTERFACE
   TYPE
      index = 1..20;
      taskarray = ARRAY[index] of real;

   VAR
      Scenario : integer;
      NunSSTh, KGunSSTh, DensSSTh,
      NunSSTl, KGunSSTl, DensSSTl, lcpmSST : real;

      DKrdH, DKrdL, CTrdTSK : real;

      DKprH, DKprL : real;

      DKmpH, DKmpL, DKla, DMla, DPla, FRlaINS : real;
      CTrdGMS, NgmsGEN, WFgms, DUprGMS, NgscGMS,
      CTrdHMS, NhmsHUM, WFhms, DUprHMS, NgscHMS,
      CTrdTCS, NtcsTEL, WFtcs, DUprTCS, NgscTCS,
      CTrdAMS, NamsAEQ, WFams, DUprAMS, NgscAMS,
      DpGSCt, DphGSC : real;

      HDhum, NhumHAB, KGunHABh, KGunHABl, WFhab, KGpdHABc,
      CTrdEMU, KGunEMU, WFemu, KGehEMUc,
      CTrdMMU, KGunMMU, WFmmu, KGmhMMUc,
      KGunEQP, WFeqp, DpHUMt, DpdHUM, NdHUMt, NDstint : real;

      HDtel, KGunTEL, WFtel, KGthTELc : real;

      HDaeq, KGunAEQ, WFaeq : real;

      NtotCSE, KGunCSE, KDcsp, DlhTDR, DhhTDR, lcparam : real;

      NTSK : index;
      NREPS, Nhum, OHhum, Ntel, OHtel, Naeq, OHaeq : taskarray;

      Ttime : taskarray;
      Jobtime : real;
      NtotHUMi, NtotTELi, NtotAEQi : real;
      YSNOhum, YSNOtel, YSNOaeq : integer;

      CTrd : real;
      CTrdSST, CTrdHAB, CTrdHSE,
      CTrdTEL, CTrdAEQ, CTrdCSE, CTrdGSE : real;

      CTpr : real;
      CTprSST, CTprHAB, CTprHSE, CTprTEL, CTprAEQ,
      CTprCSE, CTprHUMt, CTprHUMb, CTprGSE : real;
      NtotHABh, NtotHABl, NtotPD, KGofHABc,
      NtotEMU, KGofEMUc, NtotMMU, KGofMMUc,
      NtotEQP, NtotTEL, KGofTELc, NtotAEQ,
      KGofCSP, NtotEH, NtotTH, NtotAH,
      NtotGMSH, NtotHMSH, NtotTCSH, NtotAMSH : real;

      CTla : real;
      CTlaSST, CTlaHUM, CTlaHAB, CTlaHSE,
      CTlaTEL, CTlaAEQ, CTlaCSE, CTlaCSP, CTlaINS : real;

      CTgs : real;
      NtotGSCt, CTgsGSCt, CTgsGSCb, CTgsTDR : real;

   PROCEDURE SETUP1;

IMPLEMENTATION

   PROCEDURE SETUP1;
      TYPE
         inputrec =
            RECORD
               recnumber : integer;
               varname : STRING[8];
               varvalue : real;
               varunit : STRING[30];
               vardescrip : STRING[80];
            END;
         inputfile = FILE OF inputrec;

         spacetask =
            RECORD
               tasknumber : integer;
               taskdescrip : STRING[100];
               repsintask : real;
               numberhum : real;
               prodhum : real;
               numbertel : real;
               prodtel : real;
               numberaeq : real;
               prodaeq : real;
            END;
         taskfile = FILE OF spacetask;

      VAR
         INPUTS1 : inputfile;
         tempinput : inputrec;
         TASKS1 : taskfile;
         temptask : spacetask;
         j : index;
         tasknum : integer;
```

```
PROCEDURE INPUTPRINT;
   BEGIN    {INPUTPRINT}
      WITH tempinput DO
         writeln(' ',varunit,' = ',vardescrip);
   END;     {INPUTPRINT}


BEGIN    {SETUP1}

   writeln('From Procedure SETUP1:');
   writeln;
   writeln('Data from file INPUTS1:');

   reset(INPUTS1,'/VDAVE/INPUTS1');
      seek(INPUTS1,2);
      read(INPUTS1,tempinput);
      NunSSTh := tempinput.varvalue;
      write('NunSSTh = ',NunSSTh:3:2);
      INPUTPRINT;

      seek(INPUTS1,3);
      read(INPUTS1,tempinput);
      KGunSSTh := tempinput.varvalue;
      write('KGunSSTh = ',KGunSSTh:3:2);
      INPUTPRINT;

             o

             o   (lots more inputs read in and printed out)

             o

      seek(INPUTS1,102);
      read(INPUTS1,tempinput);
      lcparam := tempinput.varvalue;
      write('lcparam = ',lcparam:3:1);
      INPUTPRINT;

   close(INPUTS1,lock);

   writeln;
   writeln('Data from file TASKS1:');

   reset(TASKS1,'/VDAVE/TASKS1');
      seek(TASKS1,0);
      read(TASKS1,temptask);
      NTSK := temptask.tasknumber;
      writeln('Number of different tasks to be done: ',
                                              NTSK:1);
         write('Task                         Prod.');
         writeln('               Prod.               Prod.');
         write(' #        NREPS       Nhum   OHhum');
         writeln('        Ntel   OHtel        Naeq   OHaeq');
         writeln;

         FOR j:=1 TO NTSK DO
            BEGIN
               seek(TASKS1,j);
               read(TASKS1,temptask);

               WITH temptask DO
                  BEGIN
                     tasknum := tasknumber;
                     NREPS[j] := repsintask;
                     Nhum[j] := numberhum;
                     OHhum[j] := prodhum;
                     Ntel[j] := numbertel;
                     OHtel[j] := prodtel;
                     Naeq[j] := numberaeq;
                     OHaeq[j] := prodaeq;
                  END;

               write(tasknum:2,'      ',NREPS[j]:7:2,'      ');
               write(Nhum[j]:5:2,'  ',OHhum[j]:6:2,'       ');
               write(Ntel[j]:5:2,'  ',OHtel[j]:6:2,'       ');
               writeln(Naeq[j]:5:2,'  ',OHaeq[j]:6:2);
               writeln;
            END;   {FOR}

         close(TASKS1,lock);

      END;      {SETUP1}
END.        {UNIT SETUP1U}
```

H.6 Procedure PRDTIME1:

Procedure PRDTIME1 ("productivity-time 1") calculates daily productivities, task times, the time for the entire job, and the initial numbers of humans, teleoperators, and units of automated equipment at the worksite. These values are printed out as intermediate results of the cost model.

PRDTIME1 also checks for the total absence of each option, assigning 0-or-1 values to YSNOhum, YSNOtel, and YSNOaeq (these will be used by ARENDEE1).

PRDTIME1 is nested inside UNIT PRDTIM1U, which allows it to be compiled separately.

```
UNIT PRDTIM1U;

INTERFACE

   USES {$U SETUP1.U} SETUP1U;

   PROCEDURE PRDTIME1;

IMPLEMENTATION

   PROCEDURE PRDTIME1;
      VAR
         j : index;
         ODhum, ODtel, ODaeq, ODcrw : taskarray;
         tempmax : real;

      BEGIN  {PRDTIME1}
         writeln('From Procedure PRDTIME1:');
         writeln;

         FOR j:=1 TO NTSK DO
            BEGIN
               ODhum[j] := OHhum[j] * HDhum;
               ODtel[j] := OHtel[j] * HDtel;
               ODaeq[j] := OHaeq[j] * HDaeq;

               ODcrw[j] := ODhum[j] * Nhum[j]
                           + ODtel[j] * Ntel[j]
                           + ODaeq[j] * Naeq[j];

               Ttime[j] := NREPS[j] / ODcrw[j];
            END;  {FOR}

         writeln('       Daily     Daily     Daily       Daily',
                                      '     Task');
         writeln('Task   Prod.     Prod.     Prod.       Prod.',
                                      '     Time');
         writeln(' #     ODhum     ODtel     ODaeq       ODcrw',
                                      '     Ttime');
         writeln;

         FOR j:=1 TO NTSK DO
            BEGIN
               write(j:2,'   ',ODhum[j]:7:2,'   ');
               write(ODtel[j]:7:2,'   ',ODaeq[j]:7:2,'    ');
               writeln(ODcrw[j]:8:2,'   ',Ttime[j]:8:5);
               writeln;
            END;   {FOR}
         tempmax := 0;
         FOR j:=1 TO NTSK DO
            IF (Ttime[j] > tempmax) THEN tempmax := Ttime[j];
         Jobtime := tempmax;
         writeln('Longest task time Jobtime = ',Jobtime:8:5);
         writeln;

         NtotHUMi := 0;
         NtotTELi := 0;
         NtotAEQi := 0;

         FOR j:=1 TO NTSK DO
            BEGIN
               NtotHUMi := NtotHUMi + Nhum[j];
               NtotTELi := NtotTELi + Ntel[j];
               NtotAEQi := NtotAEQi + Naeq[j];
            END;  {FOR}
         writeln('Initial number of humans NtotHUMi = ',
                                               NtotHUMi:4:3);
         writeln('Initial number of teleoperators NtotTELi = ',
                                               NtotTELi:4:3);
         writeln('Initial number of units of AEQ NtotAEQi = ',
                                               NtotAEQi:4:3);
         writeln;

         IF (NtotHUMi = 0) THEN
            YSNOhum := 0
         ELSE
            YSNOhum := 1;

         IF (NtotTELi = 0) THEN
            YSNOtel := 0
         ELSE
            YSNOtel := 1;

         IF (NtotAEQi = 0) THEN
            YSNOaeq := 0
         ELSE
            YSNOaeq := 1;

   END;   {PRDTIME1}
END.    {PRDTIM1U}
```

H.7 Procedure ARENDEE1:

Procedure ARENDEE1 ("R&D 1") calculates research and development line items, and adds them together into the total R&D cost. These results are made available to COSTMDL1 as global variables.

ARENDEE1 is nested within UNIT ARENDE1U, which allows it to be compiled separately.

```
UNIT ARENDE1U;

INTERFACE

   USES {$U SETUP1.U} SETUP1U;

   PROCEDURE ARENDEE1;

IMPLEMENTATION

   PROCEDURE ARENDEE1;

      PROCEDURE RDofSST;
         VAR
            CTrdSSTh, CTrdSSTl : real;
         BEGIN {RDofSST}
            CTrdSSTh := DKrdH * KGunSSTh
                        + (DKrdH * KGunSSTh * NunSSTh * 0.035);
            CTrdSSTl := DKrdL * KGunSSTl
                        + (DKrdL * KGunSSTl * NunSSTl * 0.035);
            CTrdSST := CTrdSSTh + CTrdSSTl;
         END;  {RDofSST}

      PROCEDURE RDofHAB;
         BEGIN {RDofHAB}
            CTrdHAB := ((DKrdH * KGunHABh) + (DKrdL * KGunHABl))
                        * YSNOhum;
         END;  {RDofHAB}

      PROCEDURE RDofHSE;
         VAR
            CTrdEQP : real;
         BEGIN {RDofHSE}
            CTrdEQP := DKrdH * KGunEQP;
            CTrdHSE := (CTrdEMU + CTrdMMU + CTrdEQP) * YSNOhum;
         END;  {RDofHSE}

      PROCEDURE RDofTEL;
         BEGIN {RDofTEL}
            CTrdTEL := (DKrdH * KGunTEL) * YSNOtel;
         END;  {RDofTEL}

      PROCEDURE RDofAEQ;
         BEGIN {RDofAEQ}
            CTrdAEQ := (DKrdH * KGunAEQ) * YSNOaeq;
         END;  {RDofAEQ}

      PROCEDURE RDofCSE;
         BEGIN {RDofCSE}
            CTrdCSE := DKrdL * KGunCSE;
         END;  {RDofCSE}
      {No procedure needed for CTrdTSK - available from INPUTS1}

      PROCEDURE RDofGSE;
         BEGIN {RDofGSE}
            CTrdGSE := CTrdGMS + (CTrdHMS * YSNOhum)
                        + (CTrdTCS * YSNOtel)
                                         + (CTrdAMS * YSNOaeq);
         END;  {RDofGSE}


      BEGIN {ARENDEE1}

         RDofSST;
         RDofHAB;
         RDofHSE;
         RDofTEL;
         RDofAEQ;
         RDofCSE;
         RDofGSE;

         CTrd := CTrdSST + CTrdHAB + CTrdHSE + CTrdTEL
                + CTrdAEQ + CTrdCSE + CTrdTSK + CTrdGSE;
      END; {ARENDEE1}

END.   {ARENDE1U}
```

H.8 Procedure PROCURE1:

Procedure PROCURE1 ("procurement 1") calculates procurement line items, using the learning curve procedure LERNCURV where necessary. In calculating procurement costs, PROCURE1 computes a number of variables of interest (e.g. quantities of initial hardware, spares, and consumables). These are printed out as intermediate results of the model.

PROCURE1 also computes the total procurement cost. The line-item costs and the total are made available to COSTMDL1 as global variables.

PROCURE1 is nested within UNIT PROCUR1U, which allows it to be compiled separately.

```
UNIT PROCUR1U;

INTERFACE

   USES {$U SETUP1.U} SETUP1U;

   PROCEDURE PROCURE1;
```

```
IMPLEMENTATION

   PROCEDURE PROCURE1;

      PROCEDURE LERNCURV(Firstcost,numitems,lcpm : real;
                         VAR Totcost : real);
         VAR
            q, costfactor, remainder : real;
            n, lastitem : integer;

         FUNCTION IPWR (A : integer; B : real): real;
          BEGIN  {IPWR}
           IF A<=0 THEN BEGIN
            writeln('ERROR:  Argument of IPWR <= 0');
            IPWR := 0;
           END
           ELSE IPWR := EXP(B * LN(A));
         END;   {IPWR}

         FUNCTION RPWR (C : real; D : real): real;
          BEGIN  {RPWR}
           IF C<=0 THEN BEGIN
            writeln('ERROR:  Argument of RPWR <= 0');
            RPWR := 0;
           END
           ELSE RPWR := EXP(D * LN(C));
         END;   {RPWR}

         BEGIN  {LERNCURV}
            IF (numitems = 0) THEN
               Totcost := 0
            ELSE IF (numitems = 3804) AND (lcpm = 80) THEN
               Totcost := Firstcost * (0.103546) * numitems
            ELSE IF (numitems = 600) AND (lcpm = 80) THEN
               Totcost := Firstcost * (0.186611) * numitems
            ELSE
               BEGIN
                  lastitem := trunc(numitems);
                  q := (1.4427) * (LN(lcpm/100));
                  costfactor := 0;
                  FOR n := 1 TO lastitem DO
                     costfactor := costfactor + IPWR(n,q);
                  remainder := (numitems - lastitem);
                  costfactor := costfactor
                                + (remainder * RPWR(numitems,q));
                  Totcost := Firstcost * costfactor;
               END;    {IF}
         END;  {LERNCURV}
```

```
PROCEDURE PROfSST;
   VAR
      CFprSSTh, CTprSSTh,
      CFprSSTl, CTprSSTl : real;

   BEGIN  {PROfSST}
      CFprSSTh := DKprH * KGunSSTh;
      LERNCURV(CFprSSTh, NunSSTh, lcpmSST, CTprSSTh);

      CFprSSTl := DKprL * KGunSSTl;
      LERNCURV(CFprSSTl, NunSSTl, lcpmSST, CTprSSTl);

      CTprSST := CTprSSTh + CTprSSTl;
   END;  {PROfSST}


PROCEDURE PROfHAB;
   VAR
      NtotHABi, CFprHABh, CTprHABh,
      NtotHD, NtotHABs, CTprHABs,
      CFprHABl, CTprHABl : real;
      j : index;

   BEGIN  {PROfHAB}
      NtotHABi := NtotHUMi / NhumHAB;
      writeln('Initial number of habitats NtotHABi = ',
                                         NtotHABi:5:3);

      CFprHABh := KGunHABh * DKprH;
      LERNCURV(CFprHABh, NtotHABi, lcparam, CTprHABh);
      writeln('Calculated by PROfHAB:  CTprHABh = ',
                                         CTprHABh:7:2);

      NtotPD := 0;
      FOR j:=1 TO NTSK DO
         NtotPD := NtotPD + (Nhum[j] * Ttime[j]);
      writeln('Total number of person-days in space ',
                               'NtotPD = ',NtotPD:7:4);
      NtotHD := NtotPD / NhumHAB;

      NtotHABs := NtotHD * WFhab;
      write('High-tech habitation spares (in equiv. ',
                                   'habs) NtotHABs = ');
      writeln(NtotHABs:5:3);

      LERNCURV(CFprHABh, NtotHABs, lcparam, CTprHABs);
      writeln('Calculated by PROfHAB:  CTprHABs = ',
                                         CTprHABs:7:2);
      CFprHABl := KGunHABl * DKprL;
      LERNCURV(CFprHABl, NtotHABi, lcparam, CTprHABl);
      writeln('Calculated by PROfHAB:  CTprHABl = ',
                                         CTprHABl:7:2);

      CTprHAB := CTprHABh + CTprHABs + CTprHABl;

      NtotHABh := NtotHABi + NtotHABs;

      NtotHABl := NtotHABi;

      KGofHABc := NtotPD * KGpdHABc;
      writeln('Total habitation consumables KGofHABc = ',
                                         KGofHABc:7:2);
      writeln;
   END;  {PROfHAB}


PROCEDURE PROfHSE;
   VAR
      NtotEMUi, NtotEMUs, CTprEMU,
      NtotMMUi, NtotMH, NtotMMUs, CTprMMU,
      NtotEQPi, NtotQH, NtotEQPs, CTprEQP : real;

   BEGIN  {PROfHSE}
      NtotEMUi := 2 * NtotHUMi;
      writeln('Initial number of emus NtotEMUi = ',
                                         NtotEMUi:5:3);

      NtotEH := NtotPD * HDhum;
      writeln('Total number of emu-hrs in space NtotEH = ',
                                         NtotEH:7:4);

      NtotEMUs := NtotEH * WFemu;
      writeln('EMU spares (in equiv. emus) NtotEMUs = ',
                                         NtotEMUs:5:3);

      NtotEMU := NtotEMUi + NtotEMUs;

      CTprEMU := NtotEMU * KGunEMU * DKprH;
      writeln('Calculated by PROfHSE:  CTprEMU = ',
                                         CTprEMU:7:2);

      KGofEMUc := NtotEH * KGehEMUc;
      writeln('Total EMU consumables KGofEMUc = ',
                                         KGofEMUc:7:2);
      writeln;


      NtotMMUi := NtotHUMi;
      writeln('Initial number of mmus NtotMMUi = ',
                                         NtotMMUi:5:3);
      NtotMH := NtotEH;
      writeln('Total number of mmu-hrs in space NtotMH = ',
                                         NtotMH:7:4);

      NtotMMUs := NtotMH * WFmmu;
      writeln('MMU spares (in equiv. mmus) NtotMMUs = ',
                                         NtotMMUs:5:3);

      NtotMMU := NtotMMUi + NtotMMUs;

      CTprMMU := NtotMMU * KGunMMU * DKprH;
      writeln('Calculated by PRofHSE:  CTprMMU = ',
                                         CTprMMU:7:2);

      KGofMMUc := NtotMH * KGmhMMUc;
      writeln('Total MMU consumables KGofMMUc = ',
                                         KGofMMUc:7:2);
      writeln;


      NtotEQPi := NtotHUMi;
      writeln('Initial number of eqps NtotEQPi = ',
                                         NtotEQPi:5:3);

      NtotQH := NtotEH;

      NtotEQPs := NtotQH * WFeqp;
      writeln('EQP spares (in equiv. eqps) NtotEQPs = ',
                                         NtotEQPs:5:3);

      NtotEQP := NtotEQPi + NtotEQPs;

      CTprEQP := NtotEQP * KGunEQP * DKprH;
      writeln('Calculated by PRofHSE:  CTprEQP = ',
                                         CTprEQP:7:2);
      writeln;

      CTprHSE := CTprEMU + CTprMMU + CTprEQP;
   END;  {PRofHSE}


PROCEDURE PRofTEL;
   VAR
      NtotTELs, CFprTEL, CTprTELi, CTprTELs : real;
      j : index;

   BEGIN  {PRofTEL}
      writeln('Initial number of tels NtotTELi = ',
                                         NtotTELi:5:3);
      CFprTEL := KGunTEL * DKprH;
      LERNCURV(CFprTEL, NtotTELi, lcparam, CTprTELi);
      writeln('Calculated by PRofTEL:  CTprTELi = ',
                                           CTprTELi:7:2);

      NtotTH := 0;
      FOR j:=1 TO NTSK DO
         NtotTH := NtotTH + (Ntel[j] * Ttime[j] * HDtel);
      writeln('Total number of tel-hrs in space NtotTH = ',
                                           NtotTH:7:4);

      NtotTELs := NtotTH * WFtel;
      writeln('TEL spares (in equiv. tels) NtotTELs = ',
                                           NtotTELs:5:3);

      LERNCURV(CFprTEL, NtotTELs, lcparam, CTprTELs);
      writeln('Calculated by PRofTEL:  CTprTELs = ',
                                           CTprTELs:7:2);

      CTprTEL := CTprTELi + CTprTELs;

      NtotTEL := NtotTELi + NtotTELs;

      KGofTELc := NtotTH * KGthTELc;
      writeln('Total TEL consumables KGofTELc = ',
                                           KGofTELc:7:2);
      writeln;
   END;  {PRofTEL}


PROCEDURE PRofAEQ;
   VAR
      NtotAEQs, CFprAEQ, CTprAEQi, CTprAEQs : real;
      j : index;

   BEGIN  {PRofAEQ}
      writeln('Initial number of aeqs NtotAEQi = ',
                                           NtotAEQi:5:3);

      CFprAEQ := KGunAEQ * DKprH;
      LERNCURV(CFprAEQ, NtotAEQi, lcparam, CTprAEQi);
      writeln('Calculated by PRofAEQ:  CTprAEQi = ',
                                           CTprAEQi:7:2);

      NtotAH := 0;
      FOR j:=1 TO NTSK DO
         NtotAH := NtotAH + (Naeq[j] * Ttime[j] * HDaeq);
      writeln('Total number of aeq-hrs in space NtotAH = ',
                                           NtotAH:7:4);
      NtotAEQs := NtotAH * WFaeq;
      writeln('AEQ spares (in equiv. aeqs) NtotAEQs = ',
                                         NtotAEQs:5:3);

      LERNCURV(CFprAEQ, NtotAEQs, lcparam, CTprAEQs);
      writeln('Calculated by PRofAEQ:  CTprAEQs = ',
                                         CTprAEQs:7:2);
      writeln;

      CTprAEQ := CTprAEQi + CTprAEQs;

      NtotAEQ := NtotAEQi + NtotAEQs;
   END;  {PRofAEQ}


PROCEDURE PRofCSE;
   BEGIN  {PRofCSE}
      CTprCSE := NtotCSE * KGunCSE * DKprL;
   END;  {PRofCSE}


PROCEDURE PRofCSP;
   BEGIN  {PRofCSP}
      KGofCSP := KDcsp * Jobtime;
      writeln('Total requirement for CSP (kgs) KGofCSP = ',
                                         KGofCSP:7:2);
      writeln;
   END;  {PRofCSP}


PROCEDURE PRofHUMt;
   VAR
      NtotHUMt : real;
   BEGIN  {PRofHUMt}
      NtotHUMt := 2 * NtotHUMi;
      CTprHUMt := NtotHUMt * DpHUMt;
   END;  {PRofHUMt}


PROCEDURE PRofHUMb;
   VAR
      NtotPDb : real;
   BEGIN    {PRofHUMb}
      NtotPDb := (2 * NdHUMt * NtotHUMi) + (2 * NtotPD);
      writeln('Total no. of paid person-days for space ',
                           'crews NtotPDb = ',NtotPDb:5:3);
      writeln;
      CTprHUMb := NtotPDb * DpdHUM;
   END;    {PRofHUMb}
```

```
PROCEDURE PROfGSE;
   VAR
      NtotGMSi, CTprGMSi, NtotGMSs, CTprGMSs, CTprGMS : real;
      NtotHMSi, CTprHMSi, NtotHMSs, CTprHMSs, CTprHMS : real;
      NtotTCSi, CTprTCSi, NtotTCSs, CTprTCSs, CTprTCS : real;
      NtotAMSi, CTprAMSi, NtotAMSs, CTprAMSs, CTprAMS : real;

   BEGIN  {PROfGSE}

      {For general monitoring stations:}
      NtotGMSi := NgmsGEN;
      write('Initial number of general monitoring ',
                                 'stations NtotGMSi = ');
      writeln(NtotGMSi:5:3);
      LERNCURV(DUprGMS, NtotGMSi, lcparam, CTprGMSi);
      writeln('Calculated by PROfGSE:  CTprGMSi = ',
                                       CTprGMSi:7:2);

      NtotGMSH := (NtotGMSi * Jobtime * 24);
      writeln('Total number of gms-hrs NtotGMSH = ',
                                       NtotGMSH:7:4);

      NtotGMSs := NtotGMSH * WFgms;
      writeln('GMS spares (in equiv. gms) NtotGMSs = ',
                                       NtotGMSs:5:3);
      LERNCURV(DUprGMS, NtotGMSs, lcparam, CTprGMSs);
      writeln('Calculated by PROfGSE:  CTprGMSs = ',
                                       CTprGMSs:7:2);

      CTprGMS := CTprGMSi + CTprGMSs;
      writeln('Calculated by PROfGSE:  CTprGMS = ',
                                       CTprGMS:7:2);
      writeln;


      {For human monitoring stations:}
      NtotHMSi := (NtotHUMi * NhmsHUM);
      write('Initial number of human monitoring ',
                             'stations NtotHMSi = ');
      writeln(NtotHMSi:5:3);
      LERNCURV(DUprHMS, NtotHMSi, lcparam, CTprHMSi);
      writeln('Calculated by PROfGSE:  CTprHMSi = ',
                                       CTprHMSi:7:2);

      NtotHMSH := (NtotEH * NhmsHUM);
      writeln('Total number of hms-hrs NtotHMSH = ',
                                       NtotHMSH:7:4);

      NtotHMSs := NtotHMSH * WFhms;
      writeln('HMS spares (in equiv. hms) NtotHMSs = ',
                                       NtotHMSs:5:3);
```

```
LERNCURV(DUprHMS, NtotHMSs, lcparam, CTprHMSs);
writeln('Calculated by PRofGSE:  CTprHMSs = ',
                                  CTprHMSs:7:2);

CTprHMS := CTprHMSi + CTprHMSs;
writeln('Calculated by PRofGSE:  CTprHMS = ',
                                  CTprHMS:7:2);
writeln;


{For teleoperator control stations:}
NtotTCSi := (NtotTELi * NtcsTEL);
write('Initial number of teleoperator ',
                            'control stations ');
writeln('NtotTCSi = ',NtotTCSi:5:3);
LERNCURV(DUprTCS, NtotTCSi, lcparam, CTprTCSi);
writeln('Calculated by PRofGSE:  CTprTCSi = ',
                                  CTprTCSi:7:2);

NtotTCSH := (NtotTH * NtcsTEL);
writeln('Total number of tcs-hrs NtotTCSH = ',
                                  NtotTCSH:7:4);

NtotTCSs := NtotTCSH * WFtcs;
writeln('TCS spares (in equiv. tcs) NtotTCSs = ',
                                  NtotTCSs:5:3);
LERNCURV(DUprTCS, NtotTCSs, lcparam, CTprTCSs);
writeln('Calculated by PRofGSE:  CTprTCSs = ',
                                  CTprTCSs:7:2);

CTprTCS := CTprTCSi + CTprTCSs;
writeln('Calculated by PRofGSE:  CTprTCS = ',
                                  CTprTCS:7:2);
writeln;


{For automated equipment monitoring stations:}
NtotAMSi := (NtotAEQi * NamsAEQ);
write('Initial number of auto. equipment ',
                        'monitoring stations ');
writeln('NtotAMSi = ',NtotAMSi:5:3);
LERNCURV(DUprAMS, NtotAMSi, lcparam, CTprAMSi);
writeln('Calculated by PRofGSE:  CTprAMSi = ',
                                  CTprAMSi:7:2);

NtotAMSH := (NtotAH * NamsAEQ);
writeln('Total number of ams-hrs NtotAMSH = ',
                                  NtotAMSH:7:4);
```

```
            NtotAMSs := NtotAMSH * WFams;
            writeln('AMS spares (in equiv. ams) NtotAMSs = ',
                                               NtotAMSs:5:3);
            LERNCURV(DUprAMS, NtotAMSs, lcparam, CTprAMSs);
            writeln('Calculated by PROfGSE:  CTprAMSs = ',
                                               CTprAMSs:7:2);

            CTprAMS := CTprAMSi + CTprAMSs;
            writeln('Calculated by PROfGSE:  CTprAMS = ',
                                               CTprAMS:7:2);


            CTprGSE := CTprGMS + CTprHMS + CTprTCS + CTprAMS;
         END;  {PROfGSE}


      BEGIN  {PROCURE1}
         writeln('From Procedure PROCURE1:');
         writeln;

         PROfSST;
         PROfHAB;
         PROfHSE;
         PROfTEL;
         PROfAEQ;
         PROfCSE;
         PROfCSP;
         PROfHUMt;
         PROfHUMb;
         PROfGSE;

         CTpr := CTprSST + CTprHAB + CTprHSE + CTprTEL
                 + CTprAEQ + CTprCSE + CTprHUMt + CTprHUMb
                                                    + CTprGSE;
                                         {No proc. cost for CSP}
      END;  {PROCURE1}

END.  {PROCUR1U}
```

H.9 Procedure LAUNCH1:

Procedure LAUNCH1 calculates mission preparation, launch, and insurance line items. LAUNCH1 receives certain quantities calculated by PROCURE1 as global variables, such as masses of consumables to be launched and procurement costs used to compute insurance premiums.

Several quantities are displayed as intermediate results. LAUNCH1 also computes the total mission preparation and launch cost. The line-item costs and the total are made available to COSTMDL1 as global variables.

LAUNCH1 is nested within UNIT LAUNCH1U, which allows it to be compiled separately.

```
UNIT LAUNCH1U;

INTERFACE

   USES {$U SETUP1.U} SETUP1U;

   PROCEDURE LAUNCH1;

IMPLEMENTATION

   PROCEDURE LAUNCH1;
      VAR
         CTinSST, CTinHAB, CTinHSE, CTinTEL,
         CTinAEQ, CTinCSE, CTinCSP : real;

      PROCEDURE LAofSST;
         VAR
            CTlaSSTh, CTlaSSTl : real;

         BEGIN  {LAofSST}
            IF (DensSSTh >= 92.5) THEN
               CTlaSSTh := (NunSSTh * KGunSSTh * DKmpH)
                           + (NunSSTh * KGunSSTh * DKla)
            ELSE
               CTlaSSTh := (NunSSTh * KGunSSTh * DKmpH)
                           + (NunSSTh * KGunSSTh / DensSSTh)
                                                      * DMla;

            IF (DensSSTl >= 92.5) THEN
               CTlaSSTl := (NunSSTl * KGunSSTl * DKmpL)
                           + (NunSSTl * KGunSSTl * DKla)
            ELSE
               CTlaSSTl := (NunSSTl * KGunSSTl * DKmpL)
                           + (NunSSTl * KGunSSTl / DensSSTl)
                                                      * DMla;

            CTlaSST := CTlaSSTh + CTlaSSTl;

            CTinSST := (CTprSST + CTlaSST) * FRlaINS;
         END;  {LAofSST}


      PROCEDURE LAofHUM;
         VAR
            HUMtotLA : real;
   BEGIN  {LAofHUM}
      HUMtotLA := (NtotPD / NDstint);
      writeln('Total number of launches of humans ',
                         'HUMtotLA = ',HUMtotLA:7:3);
      writeln;

      CTlaHUM := HUMtotLA * DPla;
   END;  {LAofHUM}


PROCEDURE LAofHAB;
   BEGIN  {LAofHAB}
      CTlaHAB := (NtotHABh * KGunHABh * DKmpH)
                  + (NtotHABh * KGunHABh * DKla)
                  + (NtotHABl * KGunHABl * DKmpL)
                  + (NtotHABl * KGunHABl * DKla)
                  + (KGofHABc * DKla);

      CTinHAB := (CTprHAB + CTlaHAB) * FRlaINS;
   END;  {LAofHAB}


PROCEDURE LAofHSE;
   VAR
      CTlaEMU, CTlaMMU, CTlaEQP : real;

   BEGIN  {LAofHSE}
      CTlaEMU := (NtotEMU * KGunEMU * DKmpH)
                  + (NtotEMU * KGunEMU * DKla)
                  + (KGofEMUc * DKla);

      CTlaMMU := (NtotMMU * KGunMMU * DKmpH)
                  + (NtotMMU * KGunMMU * DKla)
                  + (KGofMMUc * DKla);

      CTlaEQP := (NtotEQP * KGunEQP * DKmpH)
                  + (NtotEQP * KGunEQP * DKla);

      CTlaHSE := CTlaEMU + CTlaMMU + CTlaEQP;

      CTinHSE := (CTprHSE + CTlaHSE) * FRlaINS;
   END;  {LAofHSE}


PROCEDURE LAofTEL;
   BEGIN  {LAofTEL}
      CTlaTEL := (NtotTEL * KGunTEL * DKmpH)
                  + (NtotTEL * KGunTEL * DKla)
                  + (KGofTELc * DKla);

      CTinTEL := (CTprTEL + CTlaTEL) * FRlaINS;
   END;  {LAofTEL}
```

```
PROCEDURE LAofAEQ;
   BEGIN  {LAofAEQ}
      CTlaAEQ := (NtotAEQ * KGunAEQ * DKmpH)
                  + (NtotAEQ * KGunAEQ * DKla);

      CTinAEQ := (CTprAEQ + CTlaAEQ) * FRlaINS;
   END;  {LAofAEQ}


PROCEDURE LAofCSE;
   BEGIN  {LAofCSE}
      CTlaCSE := (NtotCSE * KGunCSE * DKmpL)
                  + (NtotCSE * KGunCSE * DKla);

      CTinCSE := (CTprCSE + CTlaCSE) * FRlaINS;
   END;  {LAofCSE}


PROCEDURE LAofCSP;
   BEGIN  {LAofCSP}
      CTlaCSP :=  KGofCSP * DKla;

      CTinCSP := CTlaCSP * FRlaINS;
   END;  {LAofCSP}


PROCEDURE LAofINS;
   BEGIN  {LAofINS}
      writeln('Insurance premiums:');
      writeln('Space structure (CTinSST) = ',CTinSST:7:2);
      writeln('Habitation modules (CTinHAB) = ',
                                         CTinHAB:7:2);
      writeln('Human support equipment (CTinHSE) = ',
                                         CTinHSE:7:2);
      writeln('Teleoperators (CTinTEL) = ',CTinTEL:7:2);
      writeln('Automated equipment (CTinAEQ) = ',
                                         CTinAEQ:7:2);
      writeln('Construction support equipment (CTinCSE) = ',
                                         CTinCSE:7:2);
      writeln('Construction site propellant (CTinCSP) = ',
                                         CTinCSP:7:2);

      CTlaINS := CTinSST + CTinHAB + CTinHSE
                  + CTinTEL + CTinAEQ + CTinCSE + CTinCSP;
   END;  {LAofINS}


BEGIN  {LAUNCH1}
   writeln('From Procedure LAUNCH1:');
   writeln;
```

```
         LAofSST;
         LAofHUM;
         LAofHAB;
         LAofHSE;
         LAofTEL;
         LAofAEQ;
         LAofCSE;
         LAofCSP;
         LAofINS;

         CT1a := CT1aSST + CT1aHUM + CT1aHAB + CT1aHSE
                  + CT1aTEL + CT1aAEQ + CT1aCSE + CT1aCSP
                                                   + CT1aINS;
      END;  {LAUNCH1}

END.  {LAUNCH1U}
```

H.10 Procedure GRNDSUP1:

Procedure GRNDSUP1 ("ground support 1") calculates ground support line items, including communications costs. GRNDSUP1 receives several quantities calculated by PROCURE1 (e.g. various equipment usages) and the total job time from PRDTIME1 as global variables.

Several quantities are displayed as intermediate results. GRNDSUP1 also computes the total ground support cost. The line-item costs and the total are made available to COSTMDL1 as global variables.

GRNDSUP1 is nested within UNIT GRDSUP1U, which allows it to be compiled separately.

```
UNIT GRDSUP1U;

INTERFACE

   USES {$U SETUP1.U} SETUP1U;

   PROCEDURE GRNDSUP1;

IMPLEMENTATION

   PROCEDURE GRNDSUP1;

      PROCEDURE GSofGSCt;
         BEGIN   {GSofGSCt}
      NtotGSCt := (NgmsGEN * NgscGMS * (24/8))
          + (NtotHUMi * NhmsHUM * HDhum * NgscHMS / 8)
          + (NtotTELi * NtcsTEL * HDtel * NgscTCS / 8)
          + (NtotAEQi * NamsAEQ * HDaeq * NgscAMS / 8);
      write('Total no. of GSC people to be trained ',
                                         '(NtotGSCt) = ');
      writeln(NtotGSCt:5:3);

      CTgsGSCt := NtotGSCt * DpGSCt;
   END;    {GSofGSCt}


PROCEDURE GSofGSCb;
   VAR
      NtotGH, NtotGHb : real;

   BEGIN   {GSofGSCb}
      NtotGH := (NtotGMSH * NgscGMS)
                                 + (NtotHMSH * NgscHMS)
                  + (NtotTCSH * NgscTCS)
                                 + (NtotAMSH * NgscAMS);
      write('Total no. of GS pers-hours during constr. ',
                                         '(NtotGH) = ');
      writeln(NtotGH:5:4);

      NtotGHb := (NtotGSCt * NdHUMt * 8) + NtotGH;
      write('Total no. of paid GS pers-hrs (NtotGHb) = ');
      writeln(NtotGHb:8:3);
      writeln;

      CTgsGSCb := NtotGHb * DphGSC;
   END;    {GSofGSCb}


PROCEDURE GSofTDR;
   VAR
      NtotLH, CTgsLCC, NtotHH, CTgsHCC : real;

   BEGIN   {GSofTDR}
      NtotLH := (72 * Jobtime) + NtotEH + NtotAH;
      write('Total no. of low-capacity channel-hours ',
                                         '(NtotLH) = ');
      writeln(NtotLH:5:4);

      CTgsLCC := NtotLH * DlhTDR;

      NtotHH := NtotTH;
      write('Total no. of high-capacity channel-hours ',
                                         '(NtotHH) = ');
      writeln(NtotHH:5:4);
      writeln;
```

```
          CTgsHCC := NtotHH * DhhTDR;

          CTgsTDR := CTgsLCC + CTgsHCC;
       END;    {GSofTDR}


    BEGIN  {GRNDSUP1}
       writeln('From Procedure GRNDSUP1:');
       writeln;

       GSofGSCt;
       GSofGSCb;
       GSofTDR;

       CTgs := CTgsGSCt + CTgsGSCb + CTgsTDR;
    END;   {GRNDSUP1}

END.   {GRDSUP1U}
```

APPENDIX I:

STAGED OPTIMIZATION METHOD: COMPUTER PROGRAMS

I.1 General Notes:

This study's Staged Optimization Method (as described in section 3.5 and demonstrated in section 4.4) is implemented as a set of computer programs. The overall structure of these programs is shown in Figure I.1.

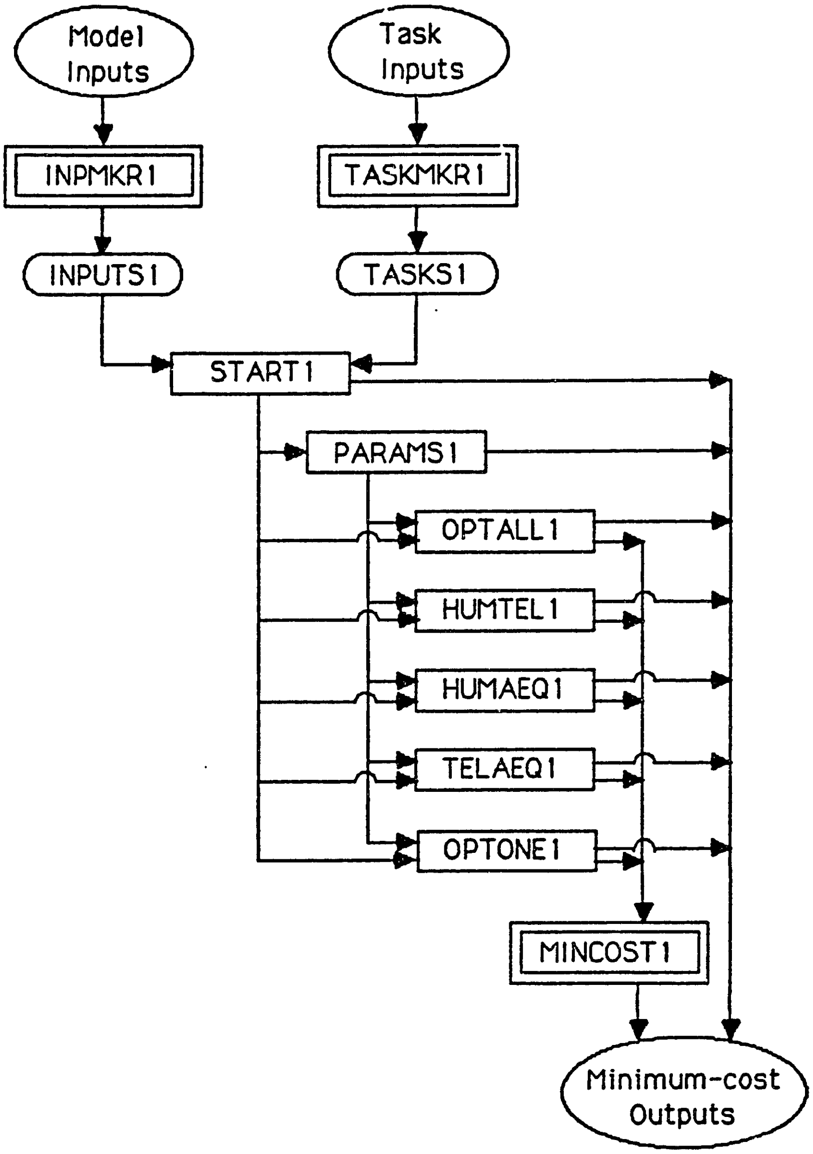


FIGURE I.1: STRUCTURE OF COMPUTER PROGRAMS FOR STAGED OPTIMIZATION METHOD

Three main programs are used in the Staged Optimization Method (SOM). Programs INPMKR1 and TASKMKR1 load the model inputs into the input files INPUTS1 and TASKS1. The program MINCOST1 runs the optimization algorithms, using subroutines START1, PARAMS1, OPTALL1, HUMTEL1, HUMAEQ1, TELAEQ1, and unit OPTONE1. Various parts of the SOM contribute to the optimal outputs.

These programs are written in UCSD Pascal and run on a Corvus Concept workstation. The Corvus Concept has its own Corvus Concept Operating System, and requires unit structures and segmentation to compile large programs.

The model and task inputs, the programs INPMKR1 and TASKMKR1, and the input files INPUTS1 and TASKS1 are the same as in the line-item cost model (see Appendix H for these programs). In this application, however, the input values for Nhum[j], Ntel[j], and Naeq[j] are ignored, since MINCOST1 will calculate optimal values for these decision variables.

The following sections list MINCOST1 and its subroutines.

I.2 Program MINCOST1:

Program MINCOST1 ("minimize cost 1") calculates and displays the minimum-cost solution for the space scenario under study. To do this, the program calls the procedures START1, PARAMS1, OPTALL1, HUMTEL1, HUMAEQ1, TELAEQ1, and the unit OPTONE1. Since MINCOST1 and these procedures are too large to be compiled all together, each procedure is in a separate "unit". These units are compiled by themselves, and "used" by the main program.

This section presents the main program only. The following sections list the procedures.

```
{$S SEGTHRE}

PROGRAM MINCOST1;

   USES {$U START1.U} START1U,
        {$U PARAMS1.U} PARAMS1U,
        {$U OPTALL1.U} OPTALL1U,
        {$U HUMTEL1.U} HUMTEL1U,
        {$U HUMAEQ1.U} HUMAEQ1U,
        {$U TELAEQ1.U} TELAEQ1U,
        {$U OPTONE1.U} OPTONE1U;

   VAR
      j : index;
```

```
BEGIN   {MINCOST1}
   writeln;
   writeln('                OUTPUT OF PROGRAM MINCOST1');
   writeln;

   START1;

   PARAMS1;

   FOR j:=1 TO NTSK DO
      BEGIN
         ODhum[j] := OHhum[j] * HDhum;
         ODtel[j] := OHtel[j] * HDtel;
         ODaeq[j] := OHaeq[j] * HDaeq;
      END;   {FOR}

   OPTALL1;

   HUMTEL1;

   HUMAEQ1;

   TELAEQ1;

   HUMONLY1;

   TELONLY1;

   AEQONLY1;

END.   {MINCOST1}
```

I.3 Procedure START1:

Procedure START1 sets up the optimization process by reading input values from files INPUTS1 and TASKS1 into global variables. These variables are then available to all parts of the optimization, because they are declared in the INTERFACE section of START1. Procedure START1 also prints out these inputs as it reads them. For brevity, some of the input reading and printout statements are omitted from this listing.

START1 also declares other global variables, which are used to pass calculated values between various parts of the model. For example, the parameters Ch, Hh, Ct, Ht, Ca, Ha, Hw are passed from PARAMS1 to the other procedures in this fashion.

The {$S SEGTHRE} command is a segmentation command. Procedure START1 is assigned to a separate segment than MINCOST1 and the other procedures, to avoid overloading the Linker. Procedure START1 is also nested inside UNIT START1U, which allows it to be compiled by itself.

```
{$S SEGTHRE}

UNIT START1U;

INTERFACE
   TYPE
      index = 1..20;
      taskarray = ARRAY[index] of real;
      optionarray = ARRAY[index] OF STRING[3];

   VAR
      Scenario : integer;
      NunSSTh, KGunSSTh, DensSSTh,
      NunSSTl, KGunSSTl, DensSSTl, lcpmSST : real;

      DKrdH, DKrdL, CTrdTSK : real;

      DKprH, DKprL : real;

      DKmpH, DKmpL, DKla, DMla, DPla, FRlaINS : real;

      CTrdGMS, NgmsGEN, WFgms, DUprGMS, NgscGMS,
      CTrdHMS, NhmsHUM, WFhms, DUprHMS, NgscHMS,
      CTrdTCS, NtcsTEL, WFtcs, DUprTCS, NgscTCS,
      CTrdAMS, NamsAEQ, WFams, DUprAMS, NgscAMS,
      DpGSCt, DphGSC : real;

      HDhum, NhumHAB, KGunHABh, KGunHABl, WFhab, KGpdHABc,
      CTrdEMU, KGunEMU, WFemu, KGehEMUc,
      CTrdMMU, KGunMMU, WFmmu, KGmhMMUc,
      KGunEQP, WFeqp, DpHUMt, DpdHUM, NdHUMt, NDstint : real;

      HDtel, KGunTEL, WFtel, KGthTELc : real;

      HDaeq, KGunAEQ, WFaeq : real;

      NtotCSE, KGunCSE, KDcsp, DlhTDR, DhhTDR, lcparam : real;

      NTSK : index;
      NREPS, Nhum, OHhum, Ntel, OHtel, Naeq, OHaeq : taskarray;

      IndvTime, IndvNhum, IndvNtel, IndvNaeq, IndvCost : real;
      Rw, Rh, Rt, Ra, Cw : real;
      Ch, Hh, Ct, Ht, Ca, Ha, Hw : real;
      ODhum, ODtel, ODaeq : taskarray;
      YESNOhum, YESNOtel, YESNOaeq : taskarray;
      OptJobTime, TskNhum, TskNtel, TskNaeq, TskCost : real;
      HumTime, HumCost, HumNumber : taskarray;
      TelTime, TelCost, TelNumber : taskarray;
      AeqTime, AeqCost, AeqNumber : taskarray;
      TskOption : optionarray;
      DollarDays, CostofTasks, RandD, OldProgCost: real;
      ProgCost : real;
      MinHTAcost : real;
      MinHTcost, MinHAcost, MinTAcost : real;
      MinHcost, MinTcost, MinAcost : real;


   PROCEDURE START1;

IMPLEMENTATION

   PROCEDURE START1;
      TYPE
         inputrec =
            RECORD
               recnumber : integer;
               varname : STRING[8];
               varvalue : real;
               varunit : STRING[30];
               vardescrip : STRING[80];
            END;
         inputfile = FILE OF inputrec;

         spacetask =
            RECORD
               tasknumber : integer;
               taskdescrip : STRING[100];
               repsintask : real;
               numberhum : real;
               prodhum : real;
               numbertel : real;
               prodtel : real;
               numberaeq : real;
               prodaeq : real;
            END;
         taskfile = FILE OF spacetask;

      VAR
         INPUTS1 : inputfile;
         tempinput : inputrec;

         TASKS1 : taskfile;
         temptask : spacetask;
         j : index;
         tasknum : integer;

      PROCEDURE INPUTPRINT;
         BEGIN    {INPUTPRINT}
            WITH tempinput DO
               writeln(' ',varunit,' = ',vardescrip);
         END;   {INPUTPRINT}
```

```
BEGIN   {START1}

   writeln('From Procedure START1:');
   writeln;
   writeln('Data from file INPUTS1:');

   reset(INPUTS1,'/VDAVE/INPUTS1');

      seek(INPUTS1,2);
      read(INPUTS1,tempinput);
      NunSSTh := tempinput.varvalue;
      write('NunSSTh = ',NunSSTh:3:2);
      INPUTPRINT;

      seek(INPUTS1,3);
      read(INPUTS1,tempinput);
      KGunSSTh := tempinput.varvalue;
      write('KGunSSTh = ',KGunSSTh:3:2);
      INPUTPRINT;

             o

             o   (lots more inputs read in and printed out)

             o

      seek(INPUTS1,102);
      read(INPUTS1,tempinput);
      lcparam := tempinput.varvalue;
      write('lcparam = ',lcparam:3:1);
      INPUTPRINT;

   close(INPUTS1,lock);

   writeln;
   writeln('Data from file TASKS1:');

   reset(TASKS1,'/VDAVE/TASKS1');
      seek(TASKS1,0);
      read(TASKS1,temptask);
      NTSK := temptask.tasknumber;
      writeln('Number of different tasks to be done: ',
                                               NTSK:1);

   write('Task                         Prod.');
   writeln('               Prod.               Prod.');
   write(' #        NREPS              OHhum');
   writeln('               OHtel               OHaeq');
   writeln;
```

```
         FOR j:=1 TO NTSK DO
            BEGIN
               seek(TASKS1,j);
               read(TASKS1,temptask);
               WITH temptask DO
                  BEGIN
                     tasknum := tasknumber;
                     NREPS[j] := repsintask;
                     OHhum[j] := prodhum;
                     OHtel[j] := prodtel;
                     OHaeq[j] := prodaeq;
                  END;

               write(tasknum:2,'      ',NREPS[j]:7:2,'      ');
               write('       ',OHhum[j]:6:2,'       ');
               write('       ',OHtel[j]:6:2,'       ');
               writeln('       ',OHaeq[j]:6:2);
               writeln;
            END;   {FOR}

         close(TASKS1,lock);

      END;      {START1}
END.        {UNIT START1U}
```

I.4 Procedure PARAMS1:

Procedure PARAMS1 ("parameters 1") calculates a number of useful parameters, making them available to the other procedures. PARAMS1 is nested inside UNIT PARAMS1U, which allows it to be compiled separately.

```
{$S SEGFOUR}

UNIT PARAMS1U;

INTERFACE

   USES {$U START1.U} START1U;

   PROCEDURE PARAMS1;

IMPLEMENTATION

   PROCEDURE PARAMS1;

      PROCEDURE RDPARAM1;
         VAR
            RandDw, RandDh, RandDt, RandDa : real;
   BEGIN  {RDPARAM1}
      RandDw := ((DKrdH * KGunSSTh) * (1 + (600 * 0.035))
                 + (DKrdL * KGunSSTl) * (1 + (3804 * 0.035))
                 + DKrdL * KGunCSE
                 + CTrdTSK
                 + CTrdGMS);
      Rw := RandDw / 1E6;

      RandDh := (((DKrdH * KGunHABh) + (DKrdL * KGunHABl))
                 + (CTrdEMU + CTrdMMU + (DKrdH * KGunEQP))
                 + CTrdHMS);
      Rh := RandDh  / 1E6;

      RandDt := ((DKrdH * KGunTEL)
                 + CTrdTCS);
      Rt := RandDt / 1E6;

      RandDa := ((DKrdH * KGunAEQ)
                 + CTrdAMS);
      Ra := RandDa / 1E6;
   END;  {RDPARAM1}

PROCEDURE CWPARAM1;
   VAR
      FrstSSTh, TotSSTh, FrstSSTl, TotSSTl : real;
      Costw : real;

   PROCEDURE LERNCURV(Firstcost,numitems,lcpm : real;
                      VAR Totcost : real);
      VAR
         q, costfactor, remainder : real;
         n, lastitem : integer;

      FUNCTION IPWR (A : integer; B : real): real;
       BEGIN  {IPWR}
        IF A<=0 THEN BEGIN
         writeln('ERROR:  Argument of IPWR <= 0');
         IPWR := 0;
        END
        ELSE IPWR := EXP(B * LN(A));
      END;   {IPWR}

      FUNCTION RPWR (C : real; D : real): real;
       BEGIN  {RPWR}
        IF C<=0 THEN BEGIN
         writeln('ERROR:  Argument of RPWR <= 0');
         RPWR := 0;
        END
        ELSE RPWR := EXP(D * LN(C));
      END;   {RPWR}
   BEGIN  {LERNCURV}
      IF (numitems = 0) THEN
         Totcost := 0

      ELSE IF (numitems = 3804) AND (lcpm = 80) THEN
         Totcost := Firstcost * (0.103546) * numitems

      ELSE IF (numitems = 600) AND (lcpm = 80) THEN
         Totcost := Firstcost * (0.186611) * numitems

      ELSE
         BEGIN
            lastitem := trunc(numitems);
            q := (1.4427) * (LN(lcpm/100));
            costfactor := 0;
            FOR n := 1 TO lastitem DO
               costfactor := costfactor + IPWR(n,q);
            remainder := (numitems - lastitem);
            costfactor := costfactor
                      + (remainder * RPWR(numitems,q));
            Totcost := Firstcost * costfactor;
         END;    {IF}
   END;  {LERNCURV}


BEGIN  {CWPARAM1}
   FrstSSTh := (DKprH * KGunSSTh) * (1 + FRlaINS);
   LERNCURV(FrstSSTh,NunSSTh,lcpmSST,TotSSTh);

   FrstSSTl := (DKprL * KGunSSTl) * (1 + FRlaINS);
   LERNCURV(FrstSSTl,NunSSTl,lcpmSST,TotSSTl);

   Costw := (TotSSTh
             + (NunSSTh * KGunSSTh) * (DKmpH + DKla)
                                         * (1 + FRlaINS)
             + TotSSTl
             + (NunSSTl * KGunSSTl) * (DKmpL + DKla)
                                         * (1 + FRlaINS)
             + (NtotCSE * KGunCSE)
                * (DKprL + DKmpL + DKla) * (1 + FRlaINS)
             + DUprGMS * NgmsGEN
             + (NgmsGEN * NgscGMS * (24/8))
                  * (DpGSCt + (NdHUMt * 8 * DphGSC)));

   Cw := Costw / 1E6;
END;  {CWPARAM1}
```

```
PROCEDURE HUMPARA1;
   VAR
      Costh, Usageh : real;

   BEGIN  {HUMPARA1}
      Costh := (((1 / NhumHAB) * KGunHABh * (DKmpH + DKla)
                                            * (1 + FRlaINS))
                + ((1 / NhumHAB) * KGunHABl * (DKmpL + DKla)
                                            * (1 + FRlaINS))
                + (((2 * KGunEMU) + KGunMMU + KGunEQP)
                   * (DKprH + DKmpH + DKla) * (1 + FRlaINS))
                + (2 * DpHUMt)
                + (2 * NdHUMt * DpdHUM)
                + ((NhmsHUM * HDhum * NgscHMS / 8)
                   * (DpGSCt + (NdHUMt * 8 * DphGSC)))
                + (((DKprH * KGunHABh) + (DKprL * KGunHABl))
                   * (1 + FRlaINS) * (1 / NhumHAB))
                + NhmsHUM * DUprHMS);
      Ch := Costh / 1E6;

      Usageh := (((1 + FRlaINS)
                  * (((1 / NhumHAB) * WFhab * KGunHABh
                                          * (DKmpH + DKla))
                     + (KGpdHABc * DKla)))
                 + ((1 + FRlaINS)
                    * ((HDhum * WFemu * KGunEMU)
                                       * (DKprH + DKmpH + DKla)
                       + (HDhum * KGehEMUc * DKla)
                       + (HDhum * WFmmu * KGunMMU)
                                       * (DKprH + DKmpH + DKla)
                       + (HDhum * KGmhMMUc * DKla)
                       + (HDhum * WFeqp * KGunEQP)
                                       * (DKprH + DKmpH + DKla)))
                 + (2 * DpdHUM)
                 + ((1 / NDstint) * DPla)
                 + (HDhum * NhmsHUM * NgscHMS * DphGSC)
                 + (HDhum * DlhTDR)
                 + ((WFhab / NhumHAB) * (DKprH * KGunHABh)
                                           * (1 + FRlaINS))
                 + ((HDhum * NhmsHUM * WFhms) * DUprHMS));
      Hh := Usageh  / 1E6;
   END;  {HUMPARA1}


PROCEDURE TELPARA1;
   VAR
      Costt, Usaget : real;
```

```
  BEGIN  {TELPARA1}
     Costt := ((KGunTEL * (DKmpH + DKla) * (1 + FRlaINS))
               + ((NtcsTEL * HDtel * NgscTCS / 8)
                    * (DpGSCt + (NdHUMt * 8 * DphGSC)))
               + ((DKprH * KGunTEL) * (1 + FRlaINS))
               + (NtcsTEL * DUprTCS));
     Ct := Costt / 1E6;

     Usaget := ((HDtel * WFtel * KGunTEL * (DKmpH + DKla)
                                         * (1 + FRlaINS))
                + (HDtel * KGthTELc * DKla * (1 + FRlaINS))
                + (HDtel * NtcsTEL * NgscTCS * DphGSC)
                + (HDtel * DhhTDR)
                + (HDtel * WFtel * (DKprH * KGunTEL)
                                         * (1 + FRlaINS))
                + ((HDtel * NtcsTEL * WFtcs) * DUprTCS));
     Ht := Usaget  / 1E6;
  END;  {TELPARA1}


PROCEDURE AEQPARA1;
   VAR
      Costa, Usagea : real;

   BEGIN  {AEQPARA1}
      Costa := ((KGunAEQ * (DKmpH + DKla) * (1 + FRlaINS))
                + ((NamsAEQ * HDaeq * NgscAMS / 8)
                   * (DpGSCt + (NdHUMt * 8 * DphGSC)))
                + ((DKprH * KGunAEQ) * (1 + FRlaINS))
                + (NamsAEQ * DUprAMS));
      Ca := Costa / 1E6;

      Usagea := ((HDaeq * WFaeq * KGunAEQ * (DKmpH + DKla)
                                          * (1 + FRlaINS))
                 + (HDaeq * NamsAEQ * NgscAMS * DphGSC)
                 + (HDaeq * DlhTDR)
                 + (HDaeq * WFaeq * (DKprH * KGunAEQ)
                                          * (1 + FRlaINS))
                 + ((HDaeq * NamsAEQ * WFams) * DUprAMS));
      Ha := Usagea  / 1E6;
   END;  {AEQPARA1}


PROCEDURE JTPARAM1;
   VAR
      Usagew : real;
   BEGIN  {JTPARAM1}
      Usagew := ((KDcsp * DKla * (1 + FRlaINS))
                 + (NgmsGEN * 24 * NgscGMS * DphGSC)
                 + (72 * DlhTDR)
                 + ((NgmsGEN * 24 * WFgms) * DUprGMS));
      Hw := (Usagew  / 1E6)
            + (((Rw + Rh + Rt + Ra) * 0.1) / 365);
   END;  {JTPARAM1}


BEGIN  {PARAMS1}
   RDPARAM1;
   CWPARAM1;
   HUMPARA1;
   TELPARA1;
   AEQPARA1;
   JTPARAM1;

   writeln;
   writeln('From Procedure PARAMS1:');
   writeln;
   writeln('Cost equation terms and coefficients:');
   writeln;
   write('Whole-job R&D cost ($M)                       ',
                                          '      Rw = ');
   writeln(Rw:9:4);
   write('Human-specific R&D cost ($M)                  ',
                                          '      Rh = ');
   writeln(Rh:9:4);
   write('Teleoperator-specific R&D cost ($M)           ',
                                          '      Rt = ');
   writeln(Rt:9:4);
   write('Automated equipment R&D cost ($M)             ',
                                          '      Ra = ');
   writeln(Ra:9:4);
   writeln;
   write('Whole-job general cost ($M)                   ',
                                          '      Cw = ');
   writeln(Cw:9:4);
   writeln;
   write('Number-of-humans cost factor ($M/pers)        ',
                                          '      Ch = ');
   writeln(Ch:9:4);
   write('Human usage cost factor ($M/pers-day)         ',
                                          '      Hh = ');
   writeln(Hh:9:5);
   write('Number-of-teleoperators cost factor ($M/tel)',
                                          '      Ct = ');
```

```
         writeln(Ct:9:4);
         write('Teleoperator usage cost factor ($M/tel-day) ',
                                             '      Ht = ');
         writeln(Ht:9:5);
         write('Number-of-auto.-equip.-units cost factor',
                                        ' ($M/aeq) Ca = ');
         writeln(Ca:9:4);
         write('Auto. equip. usage cost factor ($M/aeq-day) ',
                                             '      Ha = ');
         writeln(Ha:9:5);
         writeln;
         write('Whole-job timecost factor ($M/day)          ',
                                             '      Hw = ');
         writeln(Hw:9:5);
         writeln;

      END;   {PARAMS1}

END.  {PARAMS1U}
```

I.5 Procedure OPTALL1:

Procedure OPTALL1 ("options all 1") applies the SOM to the job with all human-machine options present (humans, teleoperators, and automated equipment). The all-option optimal solution is displayed. OPTALL1 is nested within UNIT OPTALL1U, which allows it to be compiled separately.

```
{$S SEGFOUR}

UNIT OPTALL1U;

INTERFACE

   USES {$U START1.U} START1U;

   PROCEDURE OPTALL1;

IMPLEMENTATION

   PROCEDURE OPTALL1;
      VAR
         NewTskOption : optionarray;
         CnstCost : real;
         Optioncheck : integer;
```

```
PROCEDURE INITHTA;  {Makes initial selections of HUM, TEL,}
                    {or AEQ for each task}
   VAR
      j : index;

   BEGIN  {INITHTA}
      writeln('Selection of Initial Options for Each Task:  ');
      writeln;
      write('Optimum values for individual tasks: numbers, ');
      writeln('costs, and times:');
      writeln('(Costs in $Million; task times in days)');
      writeln('(Task timecosts (Hw * tasktime) are included)');
      write('         If done by HUM        If done by TEL  ');
      writeln('      If done by AEQ ');
      write('Task    HumNumber   HumCost   TelNumber   TelCost');
      writeln('   AeqNumber   AeqCost');
      write(' #      (HumTime)             (TelTime)          ');
      writeln('    (AeqTime)');
      writeln;

      FOR j:=1 TO NTSK DO
         BEGIN
            HumTime[j] := SQRT((1/Hw) * (Ch * NREPS[j]
                                             / ODhum[j]));
              {Optimal time for the task if done by humans}
            HumNumber[j] := (NREPS[j] / (ODhum[j]
                                         * HumTime[j]));
                    {Optimal number of humans for the task}
            HumCost[j] := Ch * HumNumber[j]
                          + Hh * HumNumber[j] * HumTime[j]
                          + Hw * HumTime[j];
              {Minimum cost for the task if done by humans}

            TelTime[j] := SQRT((1/Hw) * (Ct * NREPS[j]
                                               / ODtel[j]));
        {Optimal time for the task if done by teleoperators}
            TelNumber[j] := (NREPS[j] / (ODtel[j]
                                           * TelTime[j]));
             {Optimal number of teleoperators for the task}
            TelCost[j] := Ct * TelNumber[j]
                          + Ht * TelNumber[j] * TelTime[j]
                          + Hw * TelTime[j];
       {Minimum cost for the task if done by teleoperators}

            AeqTime[j] := SQRT((1/Hw) * (Ca * NREPS[j]
                                               / ODaeq[j]));
        {Optimal time for the task if done by auto. equip.}
            AeqNumber[j] := (NREPS[j] / (ODaeq[j]
                                           * AeqTime[j]));
     {Optimal number of units of auto. equip. for the task}
```

```
            AeqCost[j] := Ca * AeqNumber[j]
                          + Ha * AeqNumber[j] * AeqTime[j]
                          + Hw * AeqTime[j];
         {Minimum cost for the task if done by auto. equip.}

            IF (HumCost[j] < TelCost[j]) THEN
                                {Select optimal task option}
               BEGIN
                  IF (AeqCost[j] < HumCost[j]) THEN
                     NewTskOption[j] := 'AEQ'
                  ELSE
                     NewTskOption[j] := 'HUM'
               END
            ELSE
               BEGIN
                  IF (AeqCost[j] < TelCost[j]) THEN
                     NewTskOption[j] := 'AEQ'
                  ELSE
                     NewTskOption[j] := 'TEL'
            END;  {IF}

            write(j:2,'         ');
            write(HumNumber[j]:6:2,' ',HumCost[j]:9:3);
            write('      ',TelNumber[j]:6:2,' ',
                                   TelCost[j]:9:3);
            writeln('      ',AeqNumber[j]:6:2,' ',
                                   AeqCost[j]:9:3);
            write('        (',HumTime[j]:7:2);
            write(')             (',TelTime[j]:7:2);
            writeln(')             (', AeqTime[j]:7:2,')');
            writeln;
         END;  {FOR}

      writeln('The initial selected options are:');
      writeln('   Task      Selected');
      writeln('     #        option');
      FOR j:=1 TO NTSK DO
         writeln('   ',j:2,'         ',NewTskOption[j]);
      writeln;
      writeln;

   END;   {INITHTA}


PROCEDURE ITERHTA;
   VAR
      j : index;
      OldTskOption : optionarray;
```

```
BEGIN  {ITERHTA}
             {Procedure receives selected task options}
             {as NewTskOption[j]. It computes optimal}
             {job time for all tasks together. For that}
             {job time, it computes costs for each}
             {task, and reselects an option for each}
             {task.}

   DollarDays := 0;        {Initialize temporary variable}
                           {DollarDays to zero}

   FOR j:=1 TO NTSK DO
      BEGIN
         IF (NewTskOption[j] = 'HUM') THEN
            BEGIN
               YESNOhum[j] := 1;  {For each task, indi-}
               YESNOtel[j] := 0;  {cate selected option}
               YESNOaeq[j] := 0;  {with 0-1 variables}
            END
         ELSE IF (NewTskOption[j] = 'TEL') THEN
            BEGIN
               YESNOhum[j] := 0;
               YESNOtel[j] := 1;
               YESNOaeq[j] := 0;
            END
         ELSE IF (NewTskOption[j] = 'AEQ') THEN
            BEGIN
               YESNOhum[j] := 0;
               YESNOtel[j] := 0;
               YESNOaeq[j] := 1;
            END;  {IF}

         DollarDays := DollarDays
                    + ((Ch * NREPS[j] / ODhum[j])
                                      * YESNOhum[j])
                    + ((Ct * NREPS[j] / ODtel[j])
                                      * YESNOtel[j])
                    + ((Ca * NREPS[j] / ODaeq[j])
                                      * YESNOaeq[j]);
                     {Sum of DollarDays over all tasks}
      END;  {FOR}

   OptJobTime := SQRT((1/Hw) * DollarDays);
            {Optimum job time for all tasks together}
   writeln('For the task options selected above, ');
   write('   the optimum job time (days) OptJobTime = ');
   writeln(OptJobTime:7:4);
   write('If all tasks are done in that time,');
   writeln('the cost values are:');
   writeln;
   CnstCost := Rw + Rh + Rt + Ra + Cw;
   write('Constant costs (R&D and whole-job general',
                                            ' costs; ');
```

```
  writeln('in $M) = ',CnstCost:9:3);
  writeln;
  writeln('Optimal values of task-related costs:');
  writeln('(Whole-job timecosts not included)');
  write('Task      If done by HUM      ',
                                ' If done by TEL  ');
  writeln('      If done by AEQ ');
  write(' #      HumNumber   HumCost   TelNumber   ',
                                     ' TelCost');
  writeln('   AeqNumber   AeqCost');
  writeln;

  FOR j:=1 TO NTSK DO
     BEGIN
        HumNumber[j] := (NREPS[j] / (ODhum[j]
                                     * OptJobTime));
               {Optimal number of humans for the task}
        HumCost[j] := Ch * HumNumber[j]
                      + Hh * HumNumber[j] * OptJobTime;
         {Minimum cost for the task if done by humans}

        TelNumber[j] := (NREPS[j] / (ODtel[j]
                                     * OptJobTime));
        {Optimal number of teleoperators for the task}
        TelCost[j] := Ct * TelNumber[j]
                      + Ht * TelNumber[j] * OptJobTime;
 {Minimum cost for the task if done by teleoperators}

        AeqNumber[j] := (NREPS[j] / (ODaeq[j]
                                     * OptJobTime));
{Optimal number of units of auto. equip. for the task}
        AeqCost[j] := Ca * AeqNumber[j]
                      + Ha * AeqNumber[j] * OptJobTime;
  {Minimum cost for the task if done by auto. equip.}

        write(j:2,'         ');
        write(HumNumber[j]:6:2,' ',HumCost[j]:9:3);
        write('      ',TelNumber[j]:6:2,' ',
                                     TelCost[j]:9:3);
        writeln('      ',AeqNumber[j]:6:2,' ',
                                     AeqCost[j]:9:3);
        writeln;
     END;   {FOR}

  write('Whole-job timecosts (Hw * OptJobTime;',
                                     ' in $M) = ');
  writeln((Hw * OptJobTime):9:3);
  writeln;
```

```
             {Next, the minimum program cost for the}
             {currently selected task options is}
             {calculated}
CostofTasks := 0;    {Initialize temporary variable}
                     {CostofTasks to zero}
FOR j:=1 TO NTSK DO
   BEGIN
      TskNhum := (NREPS[j] / (ODhum[j] * OptJobTime))
                                   * YESNOhum[j];
      TskNtel := (NREPS[j] / (ODtel[j] * OptJobTime))
                                   * YESNOtel[j];
      TskNaeq := (NREPS[j] / (ODaeq[j] * OptJobTime))
                                   * YESNOaeq[j];

      TskCost :=   Ch * TskNhum * YESNOhum[j]
                 + Hh * TskNhum * OptJobTime
                                  * YESNOhum[j]
                 + Ct * TskNtel * YESNOtel[j]
                 + Ht * TskNtel * OptJobTime
                                  * YESNOtel[j]
                 + Ca * TskNaeq * YESNOaeq[j]
                 + Ha * TskNaeq * OptJobTime
                                  * YESNOaeq[j];

      CostofTasks := CostofTasks + TskCost;
                 {Sum of CostofTasks over all tasks}
   END;  {FOR}

ProgCost := CnstCost + CostofTasks
                             + (Hw * OptJobTime);
writeln('For the options selected above');
writeln('   and a task time of ',OptJobTime:7:4,
                                         ' days,');
write('   the minimum program cost ($Million)',
                                   ' ProgCost = ');
writeln(ProgCost:9:3);
writeln;

    {Next, the optimal task options are reselected}
Optioncheck := 0;
             {Initialize checking variable to zero}

FOR j:=1 TO NTSK DO
   BEGIN
      OldTskOption[j] := NewTskOption[j];
                      {For each task, put old task}
                      {option in OldTskOption[j]}
```

```
            IF (HumCost[j] < TelCost[j]) THEN
                          {Select new optimal task option}
               BEGIN
                  IF (AeqCost[j] < HumCost[j]) THEN
                     NewTskOption[j] := 'AEQ'
                  ELSE
                     NewTskOption[j] := 'HUM'
               END
            ELSE
               BEGIN
                  IF (AeqCost[j] < TelCost[j]) THEN
                     NewTskOption[j] := 'AEQ'
                  ELSE
                     NewTskOption[j] := 'TEL'
               END;  {IF}

            IF (OldTskOption[j] <> NewTskOption[j]) THEN
               Optioncheck := Optioncheck + 1;
                    {Compare old and new task options and}
                    {store the result in Optioncheck}
         END;  {FOR}

      writeln('Based on the task-related costs above,');
      writeln('   the newly selected options are:');
      writeln('   Task      Selected');
      writeln('     #        option');
      FOR j:=1 TO NTSK DO
         writeln('    ',j:2,'         ',NewTskOption[j]);
      writeln;
      writeln;

   END;   {ITERHTA}


BEGIN   {OPTALL1}
   writeln;
   write('From Procedure OPTALL1:');
   writeln(' Humans, Teleoperators, and Automated Equipment');
   writeln;

   INITHTA;

   writeln;
   writeln('Iteration of Task Options:  ');
   writeln;

   REPEAT
      ITERHTA;
   UNTIL (Optioncheck = 0);
```

```
      writeln('These are the same as the last iteration.',
                                                ' Therefore:');
      writeln;

      MinHTAcost := ProgCost;
      writeln('Final result:  ');
      writeln('   When all options (HUM, TEL, AEQ) are',
                                                ' available,');
      write('  Minimum program cost ($Million) MinHTAcost = ');
      writeln(MinHTAcost:9:3);
      writeln;
      writeln;
    END;    {OPTALL1}

END.   {OPTALL1U}
```

I.6 Procedure HUMTEL1:

Procedure HUMTEL1 ("humans and teleoperators 1") applies the SOM to a reduced-option case with two human-machine options (humans and teleoperators). The optimal solution for the reduced-option case is displayed. HUMTEL1 is nested within UNIT HUMTEL1U, so that it can be compiled separately.

```
{$S SEGFIVE}

UNIT HUMTEL1U;

INTERFACE

   USES {$U START1.U} START1U;

   PROCEDURE HUMTEL1;

IMPLEMENTATION

   PROCEDURE HUMTEL1;
      VAR
         NewTskOption : optionarray;
         CnstCost : real;
         Optioncheck : integer;

      PROCEDURE INITHT; {Makes initial selections of HUM or TEL}
                        {for each task}
         VAR
            j : index;
```

```
BEGIN  {INITHT}
   writeln('Selection of Initial Options for Each Task:  ');
   writeln;
   write('Optimum values for individual tasks: numbers, ');
   writeln('costs, and times:');
   writeln('(Costs in $Million; task times in days)');
   writeln('(Task timecosts (Hw * tasktime) are included)');
   writeln('          If done by HUM      If done by TEL  ');
   writeln('Task    HumNumber   HumCost    TelNumber    TelCost');
   writeln(' #      (HumTime)             (TelTime)          ');
   writeln;

   FOR j:=1 TO NTSK DO
      BEGIN
         HumTime[j] := SQRT((1/Hw) * (Ch * NREPS[j]
                                    / ODhum[j]));
            {Optimal time for the task if done by humans}
         HumNumber[j] := (NREPS[j] / (ODhum[j]
                                 * HumTime[j]));
              {Optimal number of humans for the task}
         HumCost[j] := Ch * HumNumber[j]
                       + Hh * HumNumber[j] * HumTime[j]
                       + Hw * HumTime[j];
            {Minimum cost for the task if done by humans}

         TelTime[j] := SQRT((1/Hw) * (Ct * NREPS[j]
                                    / ODtel[j]));
     {Optimal time for the task if done by teleoperators}
         TelNumber[j] := (NREPS[j] / (ODtel[j]
                                    * TelTime[j]));
          {Optimal number of teleoperators for the task}
         TelCost[j] := Ct * TelNumber[j]
                       + Ht * TelNumber[j] * TelTime[j]
                       + Hw * TelTime[j];
    {Minimum cost for the task if done by teleoperators}

         IF (HumCost[j] < TelCost[j]) THEN
                        {Select optimal task option}
            NewTskOption[j] := 'HUM'
         ELSE
            NewTskOption[j] := 'TEL';

         write(j:2,'       ');
         write(HumNumber[j]:6:2,' ',HumCost[j]:9:3);
         writeln('      ',TelNumber[j]:6:2,' ',
                                      TelCost[j]:9:3);
         write('        (',HumTime[j]:7:2);
         writeln(')            (',TelTime[j]:7:2);
         writeln;
      END;  {FOR}
      writeln('The initial selected options are:');
      writeln('   Task      Selected');
      writeln('     #        option');
      FOR j:=1 TO NTSK DO
         writeln('    ',j:2,'         ',NewTskOption[j]);
      writeln;
      writeln;

   END;   {INITHT}


PROCEDURE ITERHT;
   VAR
      j : index;
      OldTskOption : optionarray;

   BEGIN  {ITERHT}
               {Procedure receives selected task options}
               {as NewTskOption[j]. It computes optimal}
               {job time for all tasks together. For that}
               {job time, it computes costs for each}
               {task, and reselects an option for each}
               {task.}

      DollarDays := 0;    {Initialize temporary variable}
                                     {DollarDays to zero}

      FOR j:=1 TO NTSK DO
         BEGIN
            IF (NewTskOption[j] = 'HUM') THEN
               BEGIN
                  YESNOhum[j] := 1; {For each task, indi-}
                  YESNOtel[j] := 0; {cate selected option}
                  YESNOaeq[j] := 0; {with 0-1 variables}
               END
            ELSE IF (NewTskOption[j] = 'TEL') THEN
               BEGIN
                  YESNOhum[j] := 0;
                  YESNOtel[j] := 1;
                  YESNOaeq[j] := 0;
               END;  {IF}

            DollarDays := DollarDays
                       + ((Ch * NREPS[j] / ODhum[j])
                                        * YESNOhum[j])
                       + ((Ct * NREPS[j] / ODtel[j])
                                        * YESNOtel[j])
                       + ((Ca * NREPS[j] / ODaeq[j])
                                        * YESNOaeq[j]);
                      {Sum of DollarDays over all tasks}
         END;  {FOR}
```

```
OptJobTime := SQRT((1/Hw) * DollarDays);
        {Optimum job time for all tasks together}
writeln('For the task options selected above, ');
write('   the optimum job time (days) OptJobTime = ');
writeln(OptJobTime:7:4);
write('If all tasks are done in that time,');
writeln('the cost values are:');
writeln;
CnstCost := Rw + Rh + Rt + Cw;
write('Constant costs (R&D and whole-job general',
                                        ' costs; ');
writeln('in $M) = ',CnstCost:9:3);
writeln;
writeln('Optimal values of task-related costs:');
writeln('(Whole-job timecosts not included)');
writeln('Task      If done by HUM       ',
                            ' If done by TEL  ');
writeln(' #      HumNumber   HumCost   TelNumber  ',
                                        ' TelCost');
writeln;

FOR j:=1 TO NTSK DO
   BEGIN
      HumNumber[j] := (NREPS[j] / (ODhum[j]
                                  * OptJobTime));
            {Optimal number of humans for the task}
      HumCost[j] := Ch * HumNumber[j]
                   + Hh * HumNumber[j] * OptJobTime;
       {Minimum cost for the task if done by humans}

      TelNumber[j] := (NREPS[j] / (ODtel[j]
                                  * OptJobTime));
       {Optimal number of teleoperators for the task}
      TelCost[j] := Ct * TelNumber[j]
                   + Ht * TelNumber[j] * OptJobTime;
 {Minimum cost for the task if done by teleoperators}


      write(j:2,'          ');
      write(HumNumber[j]:6:2,' ',HumCost[j]:9:3);
      writeln('     ',TelNumber[j]:6:2,' ',
                                  TelCost[j]:9:3);
      writeln;
   END;   {FOR}

write('Whole-job timecosts (Hw * OptJobTime;',
                                  ' in $M) = ');
writeln((Hw * OptJobTime):9:3);
writeln;
```

```
            {Next, the minimum program cost for the}
            {currently selected task options is}
            {calculated}
CostofTasks := 0;    {Initialize temporary variable}
                     {CostofTasks to zero}
FOR j:=1 TO NTSK DO
   BEGIN
      TskNhum := (NREPS[j] / (ODhum[j] * OptJobTime))
                                   * YESNOhum[j];
      TskNtel := (NREPS[j] / (ODtel[j] * OptJobTime))
                                   * YESNOtel[j];
      TskNaeq := (NREPS[j] / (ODaeq[j] * OptJobTime))
                                   * YESNOaeq[j];

      TskCost :=   Ch * TskNhum * YESNOhum[j]
                 + Hh * TskNhum * OptJobTime
                                   * YESNOhum[j]
                 + Ct * TskNtel * YESNOtel[j]
                 + Ht * TskNtel * OptJobTime
                                   * YESNOtel[j]
                 + Ca * TskNaeq * YESNOaeq[j]
                 + Ha * TskNaeq * OptJobTime
                                   * YESNOaeq[j];

      CostofTasks := CostofTasks + TskCost;
                 {Sum of CostofTasks over all tasks}
   END;  {FOR}

ProgCost := CnstCost + CostofTasks + (Hw * OptJobTime);
writeln('For the options selected above');
writeln('   and a task time of ',OptJobTime:7:4,
                                        ' days,');
write('   the minimum program cost ($Million)',
                                   ' ProgCost = ');
writeln(ProgCost:9:3);
writeln;

       {Next, the optimal task options are reselected}
Optioncheck := 0; {Initialize checking variable to zero}

FOR j:=1 TO NTSK DO
   BEGIN
      OldTskOption[j] := NewTskOption[j];
                          {For each task, put old task}
                          {option in OldTskOption[j]}

      IF (HumCost[j] < TelCost[j]) THEN
                    {Select new optimal task option}
         NewTskOption[j] := 'HUM'
      ELSE
         NewTskOption[j] := 'TEL';
```

```
               IF (OldTskOption[j] <> NewTskOption[j]) THEN
                  Optioncheck := Optioncheck + 1;
                      {Compare old and new task options and}
                      {store the result in Optioncheck}
             END;  {FOR}

          writeln('Based on the task-related costs above,');
          writeln('   the newly selected options are:');
          writeln('   Task       Selected');
          writeln('     #         option');
          FOR j:=1 TO NTSK DO
             writeln('    ',j:2,'          ',NewTskOption[j]);
          writeln;
          writeln;

       END;   {ITERHT}


     BEGIN   {HUMTEL1}
        writeln;
        write('From Procedure HUMTEL1:');
        writeln(' Humans and Teleoperators');
        writeln;

        INITHT;

        writeln;
        writeln('Iteration of Task Options:  ');
        writeln;

        REPEAT
           ITERHT;
        UNTIL (Optioncheck = 0);

        writeln('These are the same as the last iteration.',
                                                 ' Therefore:');
        writeln;

        MinHTcost := ProgCost;
        writeln('Final result:  ');
        writeln('   When options (HUM, TEL) are available,');
        write('  Minimum program cost ($Million) MinHTcost = ');
        writeln(MinHTcost:9:3);
        writeln;
        writeln;
     END;    {HUMTEL1}

END.   {HUMTEL1U}
```

I.7 Procedure HUMAEQ1:

Similarly to HUMTEL1, procedure HUMAEQ1 ("humans and automated equipment 1") applies the SOM to a reduced-option case with two human-machine options (humans and automated equipment). The optimal solution for the reducedoption case is displayed. HUMAEQ1 is nested within UNIT HUMAEQ1U, so that it can be compiled separately.

```
{$S SEGFIVE}

UNIT HUMAEQ1U;

INTERFACE

   USES {$U START1.U} START1U;

   PROCEDURE HUMAEQ1;

IMPLEMENTATION

   PROCEDURE HUMAEQ1;
      VAR
         NewTskOption : optionarray;
         CnstCost : real;
         Optioncheck : integer;

      PROCEDURE INITHA;   {Makes initial selections of HUM or AEQ}
                          {for each task}
         VAR
            j : index;

         BEGIN  {INITHA}
            writeln('Selection of Initial Options for Each Task:  ');
            writeln;
            write('Optimum values for individual tasks: numbers, ');
            writeln('costs, and times:');
            writeln('(Costs in $Million; task times in days)');
            writeln('(Task timecosts (Hw * tasktime) are included)');
            write('          If done by HUM                        ');
            writeln('      If done by AEQ ');
            write('Task    HumNumber   HumCost                      ');
            writeln('   AeqNumber   AeqCost');
            write(' #      (HumTime)                               ');
            writeln('    (AeqTime)');
            writeln;
   FOR j:=1 TO NTSK DO
      BEGIN
         HumTime[j] := SQRT((1/Hw) * (Ch * NREPS[j]
                                            / ODhum[j]));
            {Optimal time for the task if done by humans}
         HumNumber[j] := (NREPS[j] / (ODhum[j]
                                         * HumTime[j]));
               {Optimal number of humans for the task}
         HumCost[j] := Ch * HumNumber[j]
                       + Hh * HumNumber[j] * HumTime[j]
                       + Hw * HumTime[j];
           {Minimum cost for the task if done by humans}

         AeqTime[j] := SQRT((1/Hw) * (Ca * NREPS[j]
                                            / ODaeq[j]));
     {Optimal time for the task if done by auto. equip.}
         AeqNumber[j] := (NREPS[j] / (ODaeq[j]
                                      * AeqTime[j]));
  {Optimal number of units of auto. equip. for the task}
         AeqCost[j] := Ca * AeqNumber[j]
                       + Ha * AeqNumber[j] * AeqTime[j]
                       + Hw * AeqTime[j];
     {Minimum cost for the task if done by auto. equip.}

         IF (AeqCost[j] < HumCost[j]) THEN
                        {Select optimal task option}
            NewTskOption[j] := 'AEQ'
         ELSE
            NewTskOption[j] := 'HUM';

         write(j:2,'         ');
         write(HumNumber[j]:6:2,' ',HumCost[j]:9:3);
         write('                      ');
         writeln('      ',AeqNumber[j]:6:2,' ',
                                      AeqCost[j]:9:3);
         write('        (',HumTime[j]:7:2);
         write(')                     ');
         writeln('              (', AeqTime[j]:7:2,')');
         writeln;
      END;  {FOR}

   writeln('The initial selected options are:');
   writeln('   Task      Selected');
   writeln('     #        option');
   FOR j:=1 TO NTSK DO
      writeln('    ',j:2,'         ',NewTskOption[j]);
   writeln;
   writeln;

END;   {INITHA}
```

```
PROCEDURE ITERHA;
   VAR
      j : index;
      OldTskOption : optionarray;

   BEGIN  {ITERHA}
                 {Procedure receives selected task options}
                 {as NewTskOption[j]. It computes optimal}
                 {job time for all tasks together. For that}
                 {job time, it computes costs for each}
                 {task, and reselects an option for each}
                 {task.}

      DollarDays := 0;        {Initialize temporary variable}
                              {DollarDays to zero}

      FOR j:=1 TO NTSK DO
         BEGIN
            IF (NewTskOption[j] = 'HUM') THEN
               BEGIN
                  YESNOhum[j] := 1; {For each task, indi-}
                  YESNOtel[j] := 0; {cate selected option}
                  YESNOaeq[j] := 0; {with 0-1 variables}
               END
            ELSE IF (NewTskOption[j] = 'AEQ') THEN
               BEGIN
                  YESNOhum[j] := 0;
                  YESNOtel[j] := 0;
                  YESNOaeq[j] := 1;
               END;  {IF}

            DollarDays := DollarDays
                       + ((Ch * NREPS[j] / ODhum[j])
                                         * YESNOhum[j])
                       + ((Ct * NREPS[j] / ODtel[j])
                                         * YESNOtel[j])
                       + ((Ca * NREPS[j] / ODaeq[j])
                                         * YESNOaeq[j]);
                  {Sum of DollarDays over all tasks}
         END;  {FOR}

      OptJobTime := SQRT((1/Hw) * DollarDays);
                {Optimum job time for all tasks together}
      writeln('For the task options selected above, ');
      write('   the optimum job time (days) OptJobTime = ');
      writeln(OptJobTime:7:4);
      write('If all tasks are done in that time,');
      writeln('the cost values are:');
      writeln;
      CnstCost := Rw + Rh +  Ra + Cw;
      write('Constant costs (R&D and whole-job general',
                                               ' costs; ');
```

```
 writeln('in $M) = ',CnstCost:9:3);
 writeln;
 writeln('Optimal values of task-related costs:');
 writeln('(Whole-job timecosts not included)');
 write('Task      If done by HUM                        ');
 writeln('      If done by AEQ ');
 write(' #     HumNumber   HumCost                      ');
 writeln('   AeqNumber   AeqCost');
 writeln;

 FOR j:=1 TO NTSK DO
    BEGIN
       HumNumber[j] := (NREPS[j] / (ODhum[j]
                                  * OptJobTime));
             {Optimal number of humans for the task}
       HumCost[j] := Ch * HumNumber[j]
                     + Hh * HumNumber[j] * OptJobTime;
         {Minimum cost for the task if done by humans}

       AeqNumber[j] := (NREPS[j] / (ODaeq[j]
                                  * OptJobTime));
{Optimal number of units of auto. equip. for the task}
       AeqCost[j] := Ca * AeqNumber[j]
                     + Ha * AeqNumber[j] * OptJobTime;
   {Minimum cost for the task if done by auto. equip.}

       write(j:2,'          ');
       write(HumNumber[j]:6:2,' ',HumCost[j]:9:3);
       write('                      ');
       writeln('      ',AeqNumber[j]:6:2,' ',
                                   AeqCost[j]:9:3);
       writeln;
    END;  {FOR}

 write('Whole-job timecosts (Hw * OptJobTime;',
                                   ' in $M) = ');
 writeln((Hw * OptJobTime):9:3);
 writeln;
             {Next, the minimum program cost for the}
             {currently selected task options is}
             {calculated}
 CostofTasks := 0;    {Initialize temporary variable}
                      {CostofTasks to zero}
 FOR j:=1 TO NTSK DO
    BEGIN
       TskNhum := (NREPS[j] / (ODhum[j] * OptJobTime))
                                   * YESNOhum[j];
       TskNtel := (NREPS[j] / (ODtel[j] * OptJobTime))
                                   * YESNOtel[j];
       TskNaeq := (NREPS[j] / (ODaeq[j] * OptJobTime))
                                   * YESNOaeq[j];
```

```
      TskCost :=    Ch * TskNhum * YESNOhum[j]
                  + Hh * TskNhum * OptJobTime
                                    * YESNOhum[j]
                  + Ct * TskNtel * YESNOtel[j]
                  + Ht * TskNtel * OptJobTime
                                    * YESNOtel[j]
                  + Ca * TskNaeq * YESNOaeq[j]
                  + Ha * TskNaeq * OptJobTime
                                    * YESNOaeq[j];

      CostofTasks := CostofTasks + TskCost;
             {Sum of CostofTasks over all tasks}
   END;  {FOR}

ProgCost := CnstCost + CostofTasks + (Hw * OptJobTime);
writeln('For the options selected above');
writeln('   and a task time of ',OptJobTime:7:4,' days,');
write('   the minimum program cost ($Million)',
                                   ' ProgCost = ');
writeln(ProgCost:9:3);
writeln;

      {Next, the optimal task options are reselected}
Optioncheck := 0;
               {Initialize checking variable to zero}

FOR j:=1 TO NTSK DO
   BEGIN
      OldTskOption[j] := NewTskOption[j];
                       {For each task, put old task}
                       {option in OldTskOption[j]}

      IF (AeqCost[j] < HumCost[j]) THEN
                      {Select new optimal task option}
         NewTskOption[j] := 'AEQ'
      ELSE
         NewTskOption[j] := 'HUM';

      IF (OldTskOption[j] <> NewTskOption[j]) THEN
         Optioncheck := Optioncheck + 1;
              {Compare old and new task options and}
              {store the result in Optioncheck}
   END;  {FOR}

writeln('Based on the task-related costs above,');
writeln('   the newly selected options are:');
writeln('   Task      Selected');
writeln('     #        option');
FOR j:=1 TO NTSK DO
   writeln('    ',j:2,'         ',NewTskOption[j]);
writeln;
writeln;
```

```
         END;   {ITERHA}

      BEGIN   {HUMAEQ1}
         writeln;
         write('From Procedure HUMAEQ1:');
         writeln(' Humans and Automated Equipment');
         writeln;

         INITHA;

         writeln;
         writeln('Iteration of Task Options:  ');
         writeln;

         REPEAT
            ITERHA;
         UNTIL (Optioncheck = 0);

         writeln('These are the same as the last iteration.',
                                                 ' Therefore:');
         writeln;

         MinHAcost := ProgCost;
         writeln('Final result:  ');
         writeln('   When options (HUM, AEQ) are available,');
         write(' Minimum program cost ($Million) MinHAcost = ');
         writeln(MinHAcost:9:3);
         writeln;
         writeln;
      END;    {HUMAEQ1}

END.   {HUMAEQ1U}
```

I.8 Procedure TELAEQ1:

Similarly to HUMTEL1, procedure TELAEQ1 ("teleoperators and automated equipment 1") applies the SOM to a reduced-option case with two human-machine options (teleoperators and automated equipment). The optimal solution for the reducedoption case is displayed. TELAEQ1 is nested within UNIT TELAEQ1U, so that it can be compiled separately.

```
{$S SEGSIX}

UNIT TELAEQ1U;
```

```
INTERFACE

   USES {$U START1.U} START1U;

   PROCEDURE TELAEQ1;

IMPLEMENTATION

   PROCEDURE TELAEQ1;
      VAR
         NewTskOption : optionarray;
         CnstCost : real;
         Optioncheck : integer;

      PROCEDURE INITTA; {Makes initial selections of TEL or AEQ}
                        {for each task}
         VAR
            j : index;

         BEGIN  {INITTA}
            writeln('Selection of Initial Options for Each Task:  ');
            writeln;
            write('Optimum values for individual tasks: numbers, ');
            writeln('costs, and times:');
            writeln('(Costs in $Million; task times in days)');
            writeln('(Task timecosts (Hw * tasktime) are included)');
            write('                                If done by TEL  ');
            writeln('      If done by AEQ ');
            write('Task                            TelNumber   TelCost');
            writeln('   AeqNumber   AeqCost');
            write(' #                              (TelTime)          ');
            writeln('    (AeqTime)');
            writeln;

            FOR j:=1 TO NTSK DO
               BEGIN
                  TelTime[j] := SQRT((1/Hw) * (Ct * NREPS[j]
                                             / ODtel[j]));
           {Optimal time for the task if done by teleoperators}
                  TelNumber[j] := (NREPS[j] / (ODtel[j]
                                              * TelTime[j]));
                  {Optimal number of teleoperators for the task}
                  TelCost[j] := Ct * TelNumber[j]
                                + Ht * TelNumber[j] * TelTime[j]
                                + Hw * TelTime[j];
            {Minimum cost for the task if done by teleoperators}

                  AeqTime[j] := SQRT((1/Hw) * (Ca * NREPS[j]
                                             / ODaeq[j]));
           {Optimal time for the task if done by auto. equip.}
```

```
             AeqNumber[j] := (NREPS[j] / (ODaeq[j]
                                          * AeqTime[j]));
   {Optimal number of units of auto. equip. for the task}
             AeqCost[j] := Ca * AeqNumber[j]
                           + Ha * AeqNumber[j] * AeqTime[j]
                           + Hw * AeqTime[j];
       {Minimum cost for the task if done by auto. equip.}

             IF (AeqCost[j] < TelCost[j]) THEN
                              {Select optimal task option}
                NewTskOption[j] := 'AEQ'
              ELSE
                NewTskOption[j] := 'TEL';

             write(j:2,'         ');
             write('                 ');
             write('      ',TelNumber[j]:6:2,' ',
                                          TelCost[j]:9:3);
             writeln('      ',AeqNumber[j]:6:2,' ',
                                          AeqCost[j]:9:3);
             write('                 ');
             write('               (',TelTime[j]:7:2);
             writeln(')                (', AeqTime[j]:7:2,')');
             writeln;
          END;  {FOR}

      writeln('The initial selected options are:');
      writeln('   Task      Selected');
      writeln('    #         option');
      FOR j:=1 TO NTSK DO
         writeln('    ',j:2,'         ',NewTskOption[j]);
      writeln;
      writeln;

   END;   {INITTA}


PROCEDURE ITERTA;
   VAR
      j : index;
      OldTskOption : optionarray;

   BEGIN  {ITERTA}
                {Procedure receives selected task options}
                {as NewTskOption[j]. It computes optimal}
                {job time for all tasks together. For that}
                {job time, it computes costs for each}
                {task, and reselects an option for each}
                {task.}

      DollarDays := 0;      {Initialize temporary variable}
                            {DollarDays to zero}
```

```
FOR j:=1 TO NTSK DO
   BEGIN
      IF (NewTskOption[j] = 'TEL') THEN
         BEGIN
            YESNOhum[j] := 0; {For each task, indi-}
            YESNOtel[j] := 1; {cate selected option}
            YESNOaeq[j] := 0; {with 0-1 variables}
         END
      ELSE IF (NewTskOption[j] = 'AEQ') THEN
         BEGIN
            YESNOhum[j] := 0;
            YESNOtel[j] := 0;
            YESNOaeq[j] := 1;
         END;  {IF}

      DollarDays := DollarDays
                 + ((Ch * NREPS[j] / ODhum[j])
                                   * YESNOhum[j])
                 + ((Ct * NREPS[j] / ODtel[j])
                                   * YESNOtel[j])
                 + ((Ca * NREPS[j] / ODaeq[j])
                                   * YESNOaeq[j]);
           {Sum of DollarDays over all tasks}
   END;  {FOR}

OptJobTime := SQRT((1/Hw) * DollarDays);
         {Optimum job time for all tasks together}
writeln('For the task options selected above, ');
write('   the optimum job time (days) OptJobTime = ');
writeln(OptJobTime:7:4);
write('If all tasks are done in that time,');
writeln('the cost values are:');
writeln;
CnstCost := Rw + Rt + Ra + Cw;
write('Constant costs (R&D and whole-job general',
                                         ' costs; ');
writeln('in $M) = ',CnstCost:9:3);
writeln;
writeln('Optimal values of task-related costs:');
writeln('(Whole-job timecosts not included)');
write('Task                           ',
                             ' If done by TEL  ');
writeln('      If done by AEQ ');
write(' #                            TelNumber   ',
                                   ' TelCost');
writeln('   AeqNumber   AeqCost');
writeln;

FOR j:=1 TO NTSK DO
   BEGIN
      TelNumber[j] := (NREPS[j] / (ODtel[j]
                                   * OptJobTime));
```

```
  {Optimal number of teleoperators for the task}
       TelCost[j] := Ct * TelNumber[j]
                     + Ht * TelNumber[j] * OptJobTime;
  {Minimum cost for the task if done by teleoperators}

       AeqNumber[j] := (NREPS[j] / (ODaeq[j]
                                    * OptJobTime));
{Optimal number of units of auto. equip. for the task}
       AeqCost[j] := Ca * AeqNumber[j]
                     + Ha * AeqNumber[j] * OptJobTime;
   {Minimum cost for the task if done by auto. equip.}

       write(j:2,'         ');
       write('              ');
       write('      ',TelNumber[j]:6:2,' ',
                                    TelCost[j]:9:3);
       writeln('      ',AeqNumber[j]:6:2,' ',
                                    AeqCost[j]:9:3);
       writeln;
    END;   {FOR}

 write('Whole-job timecosts (Hw * OptJobTime;',
                                    ' in $M) = ');
 writeln((Hw * OptJobTime):9:3);
 writeln;
             {Next, the minimum program cost for the}
             {currently selected task options is}
             {calculated}
 CostofTasks := 0;    {Initialize temporary variable}
                      {CostofTasks to zero}
 FOR j:=1 TO NTSK DO
    BEGIN
       TskNhum := (NREPS[j] / (ODhum[j] * OptJobTime))
                                    * YESNOhum[j];
       TskNtel := (NREPS[j] / (ODtel[j] * OptJobTime))
                                    * YESNOtel[j];
       TskNaeq := (NREPS[j] / (ODaeq[j] * OptJobTime))
                                    * YESNOaeq[j];

       TskCost :=   Ch * TskNhum * YESNOhum[j]
                  + Hh * TskNhum * OptJobTime
                                    * YESNOhum[j]
                  + Ct * TskNtel * YESNOtel[j]
                  + Ht * TskNtel * OptJobTime
                                    * YESNOtel[j]
                  + Ca * TskNaeq * YESNOaeq[j]
                  + Ha * TskNaeq * OptJobTime
                                    * YESNOaeq[j];

       CostofTasks := CostofTasks + TskCost;
                  {Sum of CostofTasks over all tasks}
    END;  {FOR}
```

```
      ProgCost := CnstCost + CostofTasks + (Hw * OptJobTime);
      writeln('For the options selected above');
      writeln('   and a task time of ',OptJobTime:7:4,
                                               ' days,');
      write('   the minimum program cost ($Million)',
                                         ' ProgCost = ');
      writeln(ProgCost:9:3);
      writeln;

            {Next, the optimal task options are reselected}
      Optioncheck := 0;
                     {Initialize checking variable to zero}

      FOR j:=1 TO NTSK DO
         BEGIN
            OldTskOption[j] := NewTskOption[j];
                            {For each task, put old task}
                            {option in OldTskOption[j]}

            IF (AeqCost[j] < TelCost[j]) THEN
                          {Select new optimal task option}
               NewTskOption[j] := 'AEQ'
            ELSE
               NewTskOption[j] := 'TEL';

            IF (OldTskOption[j] <> NewTskOption[j]) THEN
               Optioncheck := Optioncheck + 1;
                   {Compare old and new task options and}
                   {store the result in Optioncheck}
         END;  {FOR}

      writeln('Based on the task-related costs above,');
      writeln('   the newly selected options are:');
      writeln('   Task      Selected');
      writeln('     #        option');
      FOR j:=1 TO NTSK DO
         writeln('    ',j:2,'         ',NewTskOption[j]);
      writeln;
      writeln;

   END;   {ITERTA}


BEGIN   {TELAEQ1}
   writeln;
   write('From Procedure TELAEQ1:');
   writeln(' Teleoperators and Automated Equipment  ');
   writeln;

   INITTA;
```

```
         writeln;
         writeln('Iteration of Task Options:  ');
         writeln;

         REPEAT
            ITERTA;
         UNTIL (Optioncheck = 0);

         writeln('These are the same as the last iteration.',
                                                ' Therefore:');
         writeln;

         MinTAcost := ProgCost;
         writeln('Final result:  ');
         writeln('   When options (TEL, AEQ) are available,');
         write('  Minimum program cost ($Million) MinTAcost = ');
         writeln(MinTAcost:9:3);
         writeln;
         writeln;
      END;    {TELAEQ1}

END.   {TELAEQ1U}
```

I.9 Unit OPTONE1:

Unit OPTONE1 includes three procedures: HUMONLY1, TELONLY1, AEQONLY1. These apply the SOM to the single-option reduced cases: humans alone, teleoperators alone, and automated equipment alone. The optimal solution for each case is displayed.

```
{$S SEGSIX}

UNIT OPTONE1U;

INTERFACE

   USES {$U START1.U} START1U;

   PROCEDURE HUMONLY1;

   PROCEDURE TELONLY1;

   PROCEDURE AEQONLY1;

IMPLEMENTATION

   PROCEDURE HUMONLY1;
      VAR
         CnstCost : real;
         j : index;
```

```
BEGIN   {HUMONLY1}
                   {Procedure computes optimal values when}
                   {all tasks are performed by humans. It}
                   {calculates the job time for all tasks,}
                   {the associated number of humans for each}
                   {task, and the minimum program cost.}
   writeln;
   write('From Procedure HUMONLY1:');
   writeln(' Humans Only  ');
   writeln;

   DollarDays := 0;         {Initialize temporary variable}
                            {DollarDays to zero}

   FOR j:=1 TO NTSK DO
      BEGIN
         YESNOhum[j] := 1;            {For each task, indi-}
         YESNOtel[j] := 0;            {cate selected option}
         YESNOaeq[j] := 0;            {with 0-1 variables}

         DollarDays := DollarDays
                    + ((Ch * NREPS[j] / ODhum[j])
                                      * YESNOhum[j])
                    + ((Ct * NREPS[j] / ODtel[j])
                                      * YESNOtel[j])
                    + ((Ca * NREPS[j] / ODaeq[j])
                                      * YESNOaeq[j]);
                      {Sum of DollarDays over all tasks}
      END;  {FOR}

   OptJobTime := SQRT((1/Hw) * DollarDays);
           {Optimum job time for all tasks together}
   writeln('When all tasks are done by humans, ');
   write('   the optimum job time (days) OptJobTime = ');
   writeln(OptJobTime:7:4);
   write('If all tasks are done in that time,');
   writeln('the cost values are:');
   writeln;
   CnstCost := Rw + Rh + Cw;
   write('Constant costs (R&D and whole-job general',
                                           ' costs; ');
   writeln('in $M) = ',CnstCost:9:3);
   writeln;
   writeln('Optimal values of task-related costs:');
   writeln('(Whole-job timecosts not included)');
   writeln('Task       If done by HUM   ');
   writeln(' #       HumNumber   HumCost');
   writeln;
   FOR j:=1 TO NTSK DO
      BEGIN
         HumNumber[j] := (NREPS[j] / (ODhum[j]
                                          * OptJobTime));
                {Optimal number of humans for the task}
         HumCost[j] := Ch * HumNumber[j]
                       + Hh * HumNumber[j] * OptJobTime;
         {Minimum cost for the task if done by humans}

         write(j:2,'         ');
         writeln(HumNumber[j]:6:2,' ',HumCost[j]:9:3);
         writeln;
      END;  {FOR}

   write('Whole-job timecosts (Hw * OptJobTime; in $M) = ');
   writeln((Hw * OptJobTime):9:3);
   writeln;
                  {Next, the minimum program cost for the}
                  {all-human task options is calculated}
   CostofTasks := 0;       {Initialize temporary variable}
                           {CostofTasks to zero}
   FOR j:=1 TO NTSK DO
      BEGIN
         TskNhum := (NREPS[j] / (ODhum[j] * OptJobTime))
                                    * YESNOhum[j];
         TskNtel := (NREPS[j] / (ODtel[j] * OptJobTime))
                                    * YESNOtel[j];
         TskNaeq := (NREPS[j] / (ODaeq[j] * OptJobTime))
                                    * YESNOaeq[j];

         TskCost :=   Ch * TskNhum * YESNOhum[j]
                    + Hh * TskNhum * OptJobTime
                                          * YESNOhum[j]
                    + Ct * TskNtel * YESNOtel[j]
                    + Ht * TskNtel * OptJobTime
                                          * YESNOtel[j]
                    + Ca * TskNaeq * YESNOaeq[j]
                    + Ha * TskNaeq * OptJobTime
                                          * YESNOaeq[j];

         CostofTasks := CostofTasks + TskCost;
                       {Sum of CostofTasks over all tasks}
      END;  {FOR}

   MinHcost := CnstCost + CostofTasks + (Hw * OptJobTime);

   writeln('Final result:  ');
   writeln('   When only the option (HUM) is available,');
   write('  Minimum program cost ($Million) MinHcost = ');
   writeln(MinHcost:9:3);
   writeln;
   writeln;
END;    {HUMONLY1}
```

```
PROCEDURE TELONLY1;
   VAR
      CnstCost : real;
      j : index;

   BEGIN   {TELONLY1}
                  {Procedure computes optimal values when}
                  {all tasks are performed by teleoperators.}
                  {It calculates the job time for all tasks,}
                  {the associated number of teleoperators}
                  {for each task, and the minimum program}
                  {cost.}
      writeln;
      write('From Procedure TELONLY1:');
      writeln(' Teleoperators Only  ');
      writeln;

      DollarDays := 0;        {Initialize temporary variable}
                              {DollarDays to zero}

      FOR j:=1 TO NTSK DO
         BEGIN
            YESNOhum[j] := 0;           {For each task, indi-}
            YESNOtel[j] := 1;           {cate selected option}
            YESNOaeq[j] := 0;           {with 0-1 variables}

            DollarDays := DollarDays
                       + ((Ch * NREPS[j] / ODhum[j])
                                           * YESNOhum[j])
                       + ((Ct * NREPS[j] / ODtel[j])
                                           * YESNOtel[j])
                       + ((Ca * NREPS[j] / ODaeq[j])
                                           * YESNOaeq[j]);
                        {Sum of DollarDays over all tasks}
         END;  {FOR}

      OptJobTime := SQRT((1/Hw) * DollarDays);
               {Optimum job time for all tasks together}
      writeln('When all tasks are done by teleoperators, ');
      write('   the optimum job time (days) OptJobTime = ');
      writeln(OptJobTime:7:4);
      write('If all tasks are done in that time,');
      writeln('the cost values are:');
      writeln;
      CnstCost := Rw + Rt + Cw;
      write('Constant costs (R&D and whole-job general',
                                                  ' costs; ');
      writeln('in $M) = ',CnstCost:9:3);
      writeln;
      writeln('Optimal values of task-related costs:');
writeln('(Whole-job timecosts not included)');
writeln('Task                           If done by TEL  ');
writeln(' #                           TelNumber   TelCost');
writeln;

FOR j:=1 TO NTSK DO
   BEGIN
      TelNumber[j] := (NREPS[j] / (ODtel[j]
                                      * OptJobTime));
         {Optimal number of teleoperators for the task}
      TelCost[j] := Ct * TelNumber[j]
                    + Ht * TelNumber[j] * OptJobTime;
   {Minimum cost for the task if done by teleoperators}

      write(j:2,'         ');
      write('                ');
      writeln('      ',TelNumber[j]:6:2,' ',
                                      TelCost[j]:9:3);
      writeln;
   END;   {FOR}

write('Whole-job timecosts (Hw * OptJobTime; in $M) = ');
writeln((Hw * OptJobTime):9:3);
writeln;
          {Next, the minimum program cost for the}
          {all-teleoperator task options is calculated}
CostofTasks := 0;       {Initialize temporary variable}
                        {CostofTasks to zero}
FOR j:=1 TO NTSK DO
   BEGIN
      TskNhum := (NREPS[j] / (ODhum[j] * OptJobTime))
                                      * YESNOhum[j];
      TskNtel := (NREPS[j] / (ODtel[j] * OptJobTime))
                                      * YESNOtel[j];
      TskNaeq := (NREPS[j] / (ODaeq[j] * OptJobTime))
                                      * YESNOaeq[j];

      TskCost :=   Ch * TskNhum * YESNOhum[j]
                 + Hh * TskNhum * OptJobTime
                                      * YESNOhum[j]
                 + Ct * TskNtel * YESNOtel[j]
                 + Ht * TskNtel * OptJobTime
                                      * YESNOtel[j]
                 + Ca * TskNaeq * YESNOaeq[j]
                 + Ha * TskNaeq * OptJobTime
                                      * YESNOaeq[j];

      CostofTasks := CostofTasks + TskCost;
                      {Sum of CostofTasks over all tasks}
   END;  {FOR}

MinTcost := CnstCost + CostofTasks + (Hw * OptJobTime);
```

```
      writeln('Final result:  ');
      writeln('   When only the option (TEL) is available,');
      write('  Minimum program cost ($Million) MinTcost = ');
      writeln(MinTcost:9:3);
      writeln;
      writeln;
   END;    {TELONLY1}


PROCEDURE AEQONLY1;
   VAR
      CnstCost : real;
      j : index;

   BEGIN    {AEQONLY1}
                      {Procedure computes optimal values when}
                      {all tasks are performed by automated}
                      {equipment. It calculates the job time}
                      {for all tasks, the associated number of}
                      {units of automated equipment for each}
                      {task, and the minimum program cost.}
      writeln;
      write('From Procedure AEQONLY1:');
      writeln(' Automated Equipment Only  ');
      writeln;

      DollarDays := 0;      {Initialize temporary variable}
                            {DollarDays to zero}

      FOR j:=1 TO NTSK DO
         BEGIN
            YESNOhum[j] := 0;           {For each task, indi-}
            YESNOtel[j] := 0;           {cate selected option}
            YESNOaeq[j] := 1;           {with 0-1 variables}

            DollarDays := DollarDays
                      + ((Ch * NREPS[j] / ODhum[j])
                                            * YESNOhum[j])
                      + ((Ct * NREPS[j] / ODtel[j])
                                            * YESNOtel[j])
                      + ((Ca * NREPS[j] / ODaeq[j])
                                            * YESNOaeq[j]);
                    {Sum of DollarDays over all tasks}
         END;  {FOR}

      OptJobTime := SQRT((1/Hw) * DollarDays);
                    {Optimum job time for all tasks together}
      writeln('When all tasks are done by automated',
                                            ' equipment, ');
      write('   the optimum job time (days) OptJobTime = ');
      writeln(OptJobTime:7:4);
```

```
write('If all tasks are done in that time,');
writeln('the cost values are:');
writeln;
CnstCost := Rw + Ra + Cw;
write('Constant costs (R&D and whole-job general costs; ');
writeln('in $M) = ',CnstCost:9:3);
writeln;
writeln('Optimal values of task-related costs:');
writeln('(Whole-job timecosts not included)');
write('Task                                           ');
writeln('      If done by AEQ ');
write(' #                                              ');
writeln('   AeqNumber   AeqCost');
writeln;

FOR j:=1 TO NTSK DO
   BEGIN
      AeqNumber[j] := (NREPS[j] / (ODaeq[j] * OptJobTime));
         {Optimal number of units of auto. equip. for',
                                           ' the task}
      AeqCost[j] := Ca * AeqNumber[j]
                    + Ha * AeqNumber[j] * OptJobTime;
            {Minimum cost for the task if done by',
                                     ' auto. equip.}

      write(j:2,'         ');
      write('                ');
      write('                     ');
      writeln('      ',AeqNumber[j]:6:2,' ',
                                     AeqCost[j]:9:3);
      writeln;
   END;   {FOR}

write('Whole-job timecosts (Hw * OptJobTime; in $M) = ');
writeln((Hw * OptJobTime):9:3);
writeln;
                 {Next, the minimum program cost for the}
                 {all-automated-equipment task options}
                 {is calculated}
CostofTasks := 0;       {Initialize temporary variable}
                        {CostofTasks to zero}
FOR j:=1 TO NTSK DO
   BEGIN
      TskNhum := (NREPS[j] / (ODhum[j] * OptJobTime))
                                     * YESNOhum[j];
      TskNtel := (NREPS[j] / (ODtel[j] * OptJobTime))
                                     * YESNOtel[j];
      TskNaeq := (NREPS[j] / (ODaeq[j] * OptJobTime))
                                     * YESNOaeq[j];
```

```
               TskCost :=   Ch * TskNhum * YESNOhum[j]
                          + Hh * TskNhum * OptJobTime
                                               * YESNOhum[j]
                          + Ct * TskNtel * YESNOtel[j]
                          + Ht * TskNtel * OptJobTime
                                               * YESNOtel[j]
                          + Ca * TskNaeq * YESNOaeq[j]
                          + Ha * TskNaeq * OptJobTime
                                               * YESNOaeq[j];

               CostofTasks := CostofTasks + TskCost;
                           {Sum of CostofTasks over all tasks}
            END;  {FOR}

         MinAcost := CnstCost + CostofTasks + (Hw * OptJobTime);

         writeln('Final result:  ');
         writeln('   When only the option (AEQ) is available,');
         write('  Minimum program cost ($Million) MinAcost = ');
         writeln(MinAcost:9:3);
         writeln;
         writeln;
      END;    {AEQONLY1}

END.  {OPTONE1U}
```

APPENDIX J:

MORE CONTOUR PLOTS OF THE RESTRICTED CASE EXAMPLE

This appendix supplements section 2.3.2 with three more contour plots of ProgCost, for a one-task two-option case. Within each contour plot, the quantities Rw, Rh, Rt, Cw, Ch, Hh, Ct, Ht, Hw, NREPS, ODhum, and ODtel are held constant. From plot to plot, however, some of these quantities are varied parametrically, to examine their effects on the equation's overall behavior.

Unless otherwise noted, the quantities Rw, Rh, Rt, Cw, Ch, Hh, Ct, Ht, and Hw are given numerical values calculated from the expressions in section G.4, using model inputs from this study's case example (in Chapter 4). The quantities NREPS, ODhum, and ODtel are given the values of NREPS[1], ODhum[1], and ODtel[1] from the case example's Task 1.

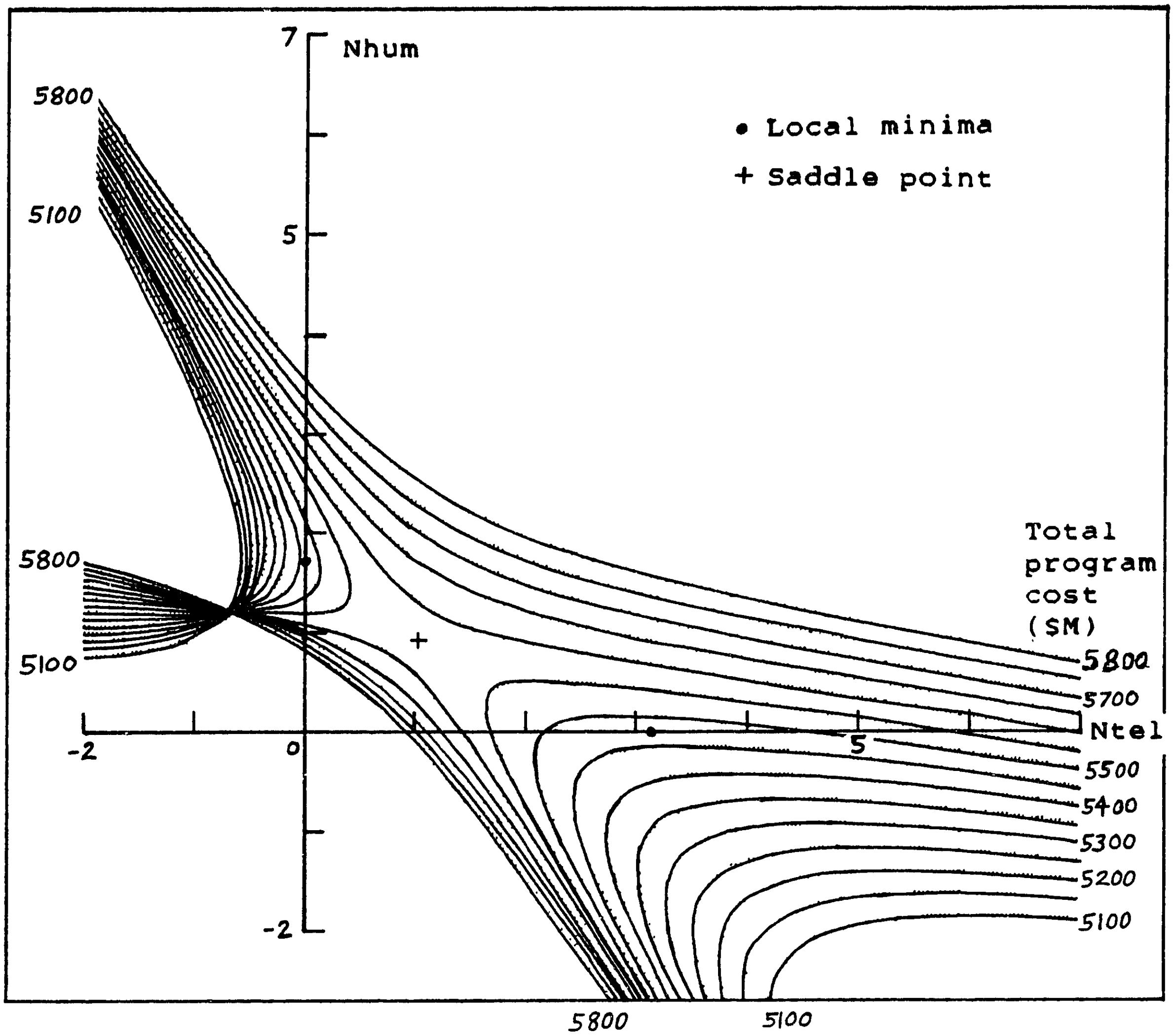
The Nhum and Ntel axes in the plots run from -2 to 7. The "feasible region" in this problem is the positive quadrant; there is no physical significance to negative humans or teleoperators. However, parts of the nonpositive quadrants are included, to show more clearly the mathematical behavior of the cost equation.

The effect of an increased number-of-humans cost factor Ch is shown in Figure J.1. The value of Ch is set at 357 $M/pers, a 50% increase over baseline. Such an increase might be due to increased procurement or launch costs for habitation module initial hardware. The other parameters have their baseline values.

In some respects this contour plot resembles the lower-ODhum plot in Figure 2.3 (in section 2.3.2). Relative to baseline, the saddle point position has been shifted upwards and to the left, and its value has increased to roughly 5580 $M. The 5500 and 5550 $M contours have gone through the saddle point to become trough contours.

However, the asymptotic plane has not tilted from its baseline angle. Also, the saddle point has not moved as far as in Figure 2.3, partly because variations in Ch do not move the contours' cross-through point through the asymptotic plane.

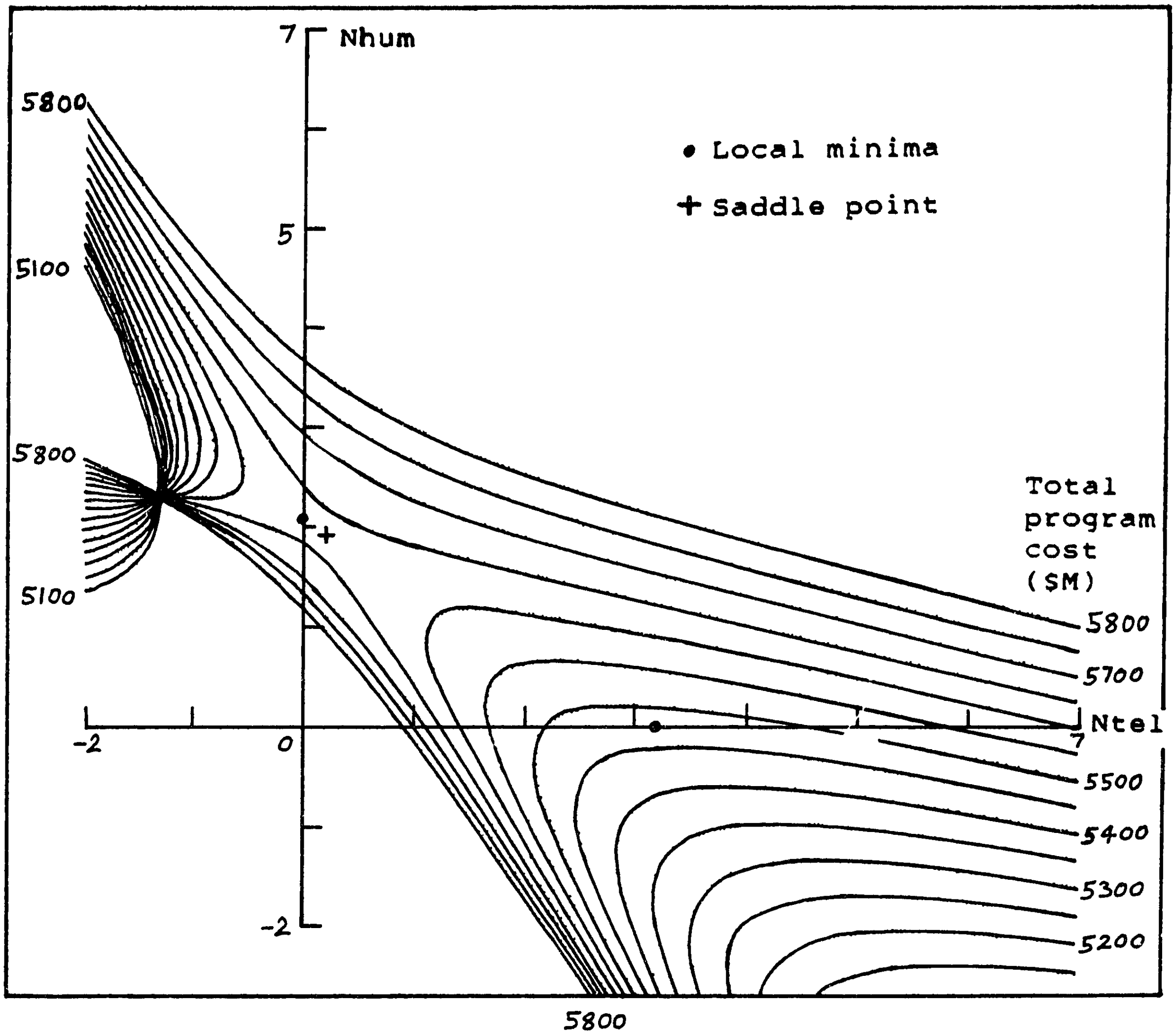
The larger Ch increases human-related costs. The Nhum trough's feasible minimum is now roughly 5465 $million, at (Nhum = 1.7). Along the Ntel axis, the change in Ch has no effect, and the contour crossings have baseline values, including the trough minimum of 5475 $M at (Ntel = 3.1). So in this case the overall feasible minimum is in the Nhum trough, but the difference between the two minima is small, and a further change in Ch would switch the choice to the other trough.



Rw = 437.403 $M	Ch = 357.000 $M/pers	Hw = 0.569 $M/day
Rh = 1574.313 $M	Hh = 0.439 $M/pers-day	NREPS = 16,688 reps
Rt = 168.430 $M	Ct = 60.190 $M/tel	ODhum = 7.3 reps/pers-day
Cw = 1256.669 $M	Ht = 1.592 $M/tel-day	ODtel = 16.0 reps/tel-day

FIGURE J.1: CONTOUR PLOT OF ProgCost (HIGHER Ch)

Figure J.2 shows the effect of a 50% increase in the human usage cost factor Hh, to 0.658 $M/pers-day. This might result from increased human consumables requirements, or increased spares requirements for human support hardware. The other parameters have their baseline values.



Rw = 437.403 $M	Ch = 237.800 $M/pers	Hw = 0.569 $M/day
Rh = 1574.313 $M	Hh = 0.658 $M/pers-day	NREPS = 16,688 reps
Rt = 168.430 $M	Ct = 60.190 $M/tel	ODhum = 7.3 reps/pers-day
Cw = 1256.669 $M	Ht = 1.592 $M/tel-day	ODtel = 16.0 reps/tel-day

FIGURE J.2: CONTOUR PLOT OF ProgCost (HIGHER Hh)

Here again, this contour plot resembles the lower-ODhum plot in Figure 2.3. The saddle point has been moved upward and to the left relative to baseline, and has almost left the feasible region. The saddle point value has increased to roughly 5635 $M, and the 5500, 5550, and 5600 $million contours have gone through the saddle point to become trough contours. The asymptotic plane has the same baseline angle, but the contours' cross-through point has moved away from the origin.

The larger usage factor increases human-related costs, e.g. roughly 5630 $M for the Nhum trough's feasible minimum at (Nhum = 2.1, Ntel = 0). Along the Ntel axis, the higher Hh has no effect, and the contour crossings have baseline values. The Ntel trough minimum of 5475 $M at (Nhum = 0, Ntel = 3.1) is clearly the overall feasible minimum.

Figures 2.3, J.1, and J.2 have shown the effects of variations in the human-related parameters ODhum, Ch, and Hh on the overall cost behavior. Similar variations on the teleoperator-related parameters ODtel, Ct, and Ht would have symmetric effects on the contour plot, moving the saddle point downward and to the right, and increasing the values of the Ntel trough contours.

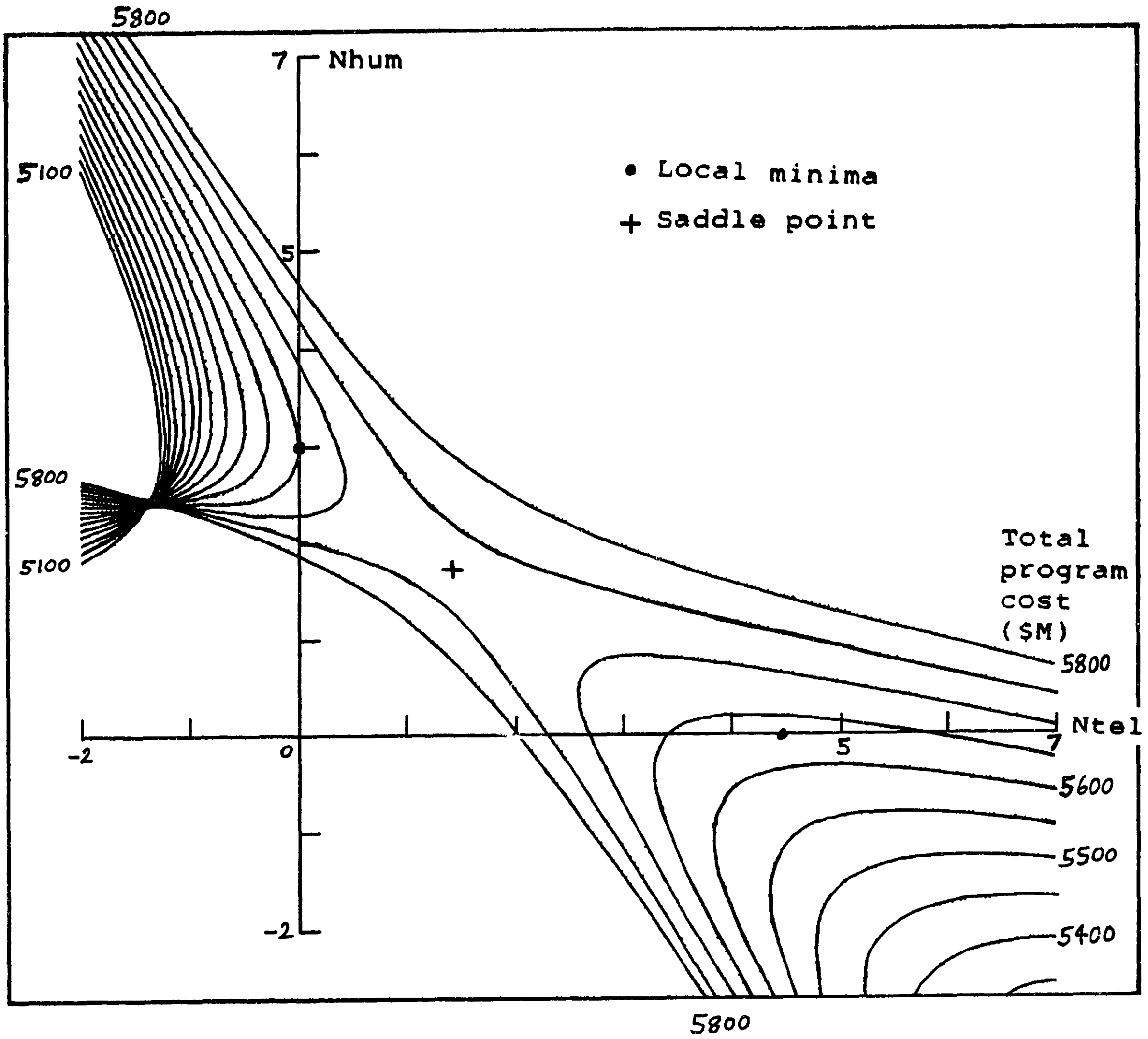
Figure J.3 displays the effect of a 100% increase in the whole-job timecost parameter Hw, from the baseline 0.569 $M/day to 1.139 $M/day. This could be due to a larger requirement for construction site propellant, or to the inclusion of interest costs. Although a 100% increase may seem high, Hw could grow to this level. For example, if the R&D costs (Rw + Rh + Rt = 2180 $M) accrued interest at 10%/year, this would add 0.597 $M/day to Hw, roughly doubling this factor. Other parameters have their baseline values.

Increasing Hw has several effects. The saddle point increases in value to roughly 5725 $million. The higher contours are compressed toward the saddle point, and the contours from 5500 through 5700 $M have gone through the saddle point to become trough contours. The saddle point and both troughs have moved away from the origin, so that the contour crossings on both axes have different values than the baseline. The asymptotic plane has not changed angle, but the contours' cross-through point has moved further from the origin.

The increase in Hw has also shifted the saddle point towards the Nhum axis. This is because in the baseline case the options in the Nhum trough (e.g. Nhum = 2, Ntel = 0) have lower daily productivities than those in the Ntel trough (e.g. Nhum = 0, Ntel = 3), and therefore take a longer time to complete the job. Therefore the increase in the cost of time has a stronger effect on human-intensive options than on teleoperator-intensive options. This raises the Nhum trough more than the Ntel trough, moving the saddle point toward the Nhum trough.

The resulting overall feasible minimum has value 5630 $M, at (Nhum = 0, Ntel = 4.3). However, the Nhum trough minimum has a value of 5650 $M, so the choice between minima is close.

The contour plots are further discussed in section 2.3.2.



Rw = 437.403 $M	Ch = 237.800 $M/pers	Hw = 1.139 $M/day
Rh = 1574.313 $M	Hh = 0.439 $M/pers-day	NREPS = 16,688 reps
Rt = 168.430 $M	Ct = 60.190 $M/tel	ODhum = 7.3 reps/pers-day
Cw = 1256.669 $M	Ht = 1.592 $M/tel-day	ODtel = 16.0 reps/tel-day

FIGURE J.3: CONTOUR PLOT OF ProgCost (HIGHER Hw)

APPENDIX K:

SPREADSHEETS OF SHUTTLE COSTS

K.1 General Notes on the Spreadsheets:

The following pages present two versions of a spreadsheet of Space Shuttle program costs. The first sheet shows expenditures as they occurred, i.e. in actual dollars. These numbers are converted into 1984 dollars in the second sheet, by multiplying them by the inflation factors listed at the bottom of the sheet.

The listed expenditures are "direct obligations", i.e. the amounts contracted out in those fiscal years (not the actual outlays, which usually lag behind the direct obligations). The data is from the yearly NASA Budget Estimates (Ref. K-1). Each year's estimates also includes the "actual" obligations for the year two years before (e.g. the 1972 estimates includes actual data for 1970).

The sheets present cost categories (e.g. DDT&E), consisting of individual line items (e.g. Vehicles). The very first column indicates the fiscal years in which each line item first appears. The costs appear in fiscal-year columns, with one column for the "Transition Quarter" between 1976 and 1977 (when the federal fiscal year was delayed by three months).

Columns of line items are summed upwards into yearly category totals. These category totals are summed upwards into yearly "Total Shuttle Costs", and also summed leftwards into the "Totals" column. In the second sheet, individual line items are also summed leftwards into the "Totals" column.

As the Shuttle program evolved, the cost categories were rearranged and sometimes renamed (e.g. between 1981 and 1982). Footnotes marked in the spreadsheets (and presented in section K.4) trace these transitions. The footnotes also list the specific pieces of hardware covered under the various categories; these lists are extracted from the program descriptions in Ref. K-1. The sheets and the footnotes use some acronyms:

DDT&E = Design, Development, Test, & Evaluation
ET = External Tank
KSC = Kennedy Space Center
MCC = Mission Control Center
SRB = Solid Rocket Booster
SSME = Space Shuttle Main Engine
STA = Structural Test Article
STS = Space Transportation System

K.2 Spreadsheet in Actual Dollars:

SHUTTLE DIRECT OBLIGATIONS	File: SHUTTLE2
Master File	Edit Date: Sep 23 1985
As copied from the NASA Budget Estimates	Edited by: Dave Stuart

	TOTALS	1970	1971
TOTAL SHUTTLE COSTS	20932475000	12500000	78500000
DDT&E: (a)	10092475000	12500000	78500000
70 Vehicles (b)		8300000(c)	
70 Engine (b)		4200000	
71 Technology and related development (b)			10600000
71 Engine definition (b)			20900000
71 Vehicle definition (b)			47000000(c)
72 Orbiter			
72 Main engine			
72 Vehicle and engine definition			
74 Solid rocket boosters			
74 External tank			
74 Launch and landing			
82 Shuttle DDT&E (e)			
PRODUCTION: (f)	2257200000		
77 Orbiter			
77 Main engine			
79 Launch and landing			
79 Spares and equipment			
SHUTTLE PRODUCTION AND OPERATIONAL CAPABILITY: (k)	4654860000		
82 Orbiter			
82 Launch and mission support			
82 Propulsion systems			
84 Changes and system upgrading			
SPACE TRANSPORTATION SYSTEM OPERATIONS CAPABILITY DEVELOPMENT: (l)	299640000		
75 Spacelab			
75 Interim upper stage			
76 STS upper stages			
76 Payload and operations support			
77 Multimission and payload support equip			
78 MCC upgrading (Level II)			
79 Thrust augmentation			
79 Teleoperator retrieval system			
80 Performance augmentation studies			
SPACE TRANSPORTATION SYSTEM OPERATIONS:	452300000		
78 Shuttla operations			
78 Payload support operations			
79 Upper stages operations			
SHUTTLE OPERATIONS: (s)	3176000000		
82 Flight operations			
82 Flight hardware			
82 Launch and landing operations			

(continued on next page)

1972	1973	1974	1975	1976	TR. QTR.
100000000	198575000	475000000	800500000	1217300000	323840000
100000000	198575000	475000000	797500000	1206000000	321000000
26100000	18552000				
15000000(c)	139480000(c)	363125000(d)	634757000	867335000	216300000
45100000	40543000	82307000	95300000	140800000	37900000
13800000					
		8567000	21143000	65700000	20400000
		18100000	34000000	82240000	26000000
		2901000	12300000	49925000	20400000
			3000000	11300000	2840000
			2000000	6100000	1300000
			1000000		
				2625000	700000
				2575000	840000

(continued on next page)

1977	1978	1979	1980	1981	1982
1429900000	1431100000	1727100000	2073200000	2340100000	2612050000
1343100000	1307500000	1270300000	1115500000	973000000	894000000
899400000	813060000	727800000	641900000	510500000	
182200000	197400000	172700000	140600000	134000000	
100400000	104998000	115400000	65200000	50500000	
84000000	88030000	104800000	79400000	63500000	
77100000	104012000	149600000	188400000	214500000	
					894000000(e)
70000000	41700000	368000000	755500000	1022000000	
65000000	29140000	264500000	572600000	779000000(g)	
5000000	12560000	75500000	123600000	112000000(h)	
		7000000	16400000	33000000(i)	
		21000000	42900000	98000000(j)	
					1282750000
					916850000
					134900000
					231000000
16800000	65400000	64500000	54100000	81700000	
8600000	21600000(m)				
1800000	8400000	19300000	18300000	38300000(n)	
4900000	24622000	10000000	10900000	9300000(o)	
1500000	5378000	10400000	9600000	15500000(o)	
	5400000	10300000	12900000	15600000(i)	
		4000000			
		10500000			
			2400000	3000000(p)	
	16500000	24300000	148100000	263400000	
	14800000	15100000	119300000	213500000(r)	
	1700000	6300000	10100000	19000000(o)	
		2900000	18700000	30900000(n)	
					435300000
					71300000
					357600000
					6400000

(continued on next page)

```
**********************************
      1983            1984
==================================
  3064510000      3048300000
==================================

----------------------------------

----------------------------------









----------------------------------

----------------------------------



----------------------------------

  1725810000      1646300000
----------------------------------
   903910000       724900000
   246300000       277700000
   575600000       638200000
                     5500000

----------------------------------

----------------------------------








----------------------------------

----------------------------------




----------------------------------
  1338700000      1402000000
----------------------------------
   317500000       343300000
   679200000       729500000
   342000000       329200000
----------------------------------
```

K.3 Spreadsheet in 1984 Dollars:

	TOTALS	1970	1971
TOTAL SHUTTLE COSTS (IN 1984$)	29369177035	37925000	224039000
DDT&E: (a)	17076634925	37925000	224039000
70 Vehicles	25182200	25182200(c)	
70 Engine	12742800	12742800	
71 Technology and related development	148122760		30252400
71 Engine definition	59648600		59648600
71 Vehicle definition	134138000		134138000(c)
72 Orbiter	10302491357		
72 Main engine	2245075446		
72 Vehicle and engine definition	37260000		
74 Solid rocket boosters	912292850		
74 External tank	968220280		
74 Launch and landing	1226604632		
82 Shuttle DDT&E	1004856000		
PRODUCTION: (f)	3034654900		
77 Orbiter	2300879580		
77 Main engine	449698420		
79 Launch and landing	73473200		
79 Spares and equipment	210603700		
SHUTTLE PRODUCTION AND OPERATIONAL CAPABILITY: (k)	4915743790		
82 Orbiter	2712680090		
82 Launch and mission support	690159300		
82 Propulsion systems	1507404400		
84 Changes and system upgrading	5500000		
SPACE TRANSPORTATION SYSTEM OPERATIONS CAPABILITY DEVELOPMENT: (l)	443677320		
75 Spacelab	69850000		
75 Interim upper stage	2151000		
76 STS upper stages	124752125		
76 Payload and operations support	97483039		
77 Multimission and payload support equip	59383856		
78 MCC upgrading (Level II)	61218600		
79 Thrust augmentation	6036000		
79 Teleoperator retrieval system	15844500		
80 Performance augmentation studies	6958200		
SPACE TRANSPORTATION SYSTEM OPERATIONS:	589505600		
78 Shuttle operations	472232900		
78 Payload support operations	49432400		
79 Upper stages operations	67840300		
SHUTTLE OPERATIONS: (s)	3308960500		
82 Flight operations	759673700		
82 Flight hardware	1850715200		
82 Launch and landing operations	698571600		
INFLATION FACTORS (TO 1984$)		3.034	2.854

(continued on next page)

1972	1973	1974	1975	1976	TR. QTR.
270000000	507359125	1131925000	1721875500	2401732900	625982720
270000000	507359125	1131925000	1715422500	2379438000	620493000
70470000	47400360				
40500000(c)	356371400(c)	865326875(d)	1365362307	1711251955	418107900
121770000	103587365	196137581	204990300	277798400	73260700
37260000					
		20415161	45478593	129626100	39433200
		43132300	73134000	162259520	50258000
		6913083	26457300	98502025	39433200
			6453000	22294900	5489720
			4302000	12035300	2512900
			2151000		
				5179125	1353100
				5080475	1623720
2.700	2.555	2.383	2.151	1.973	1.933

(continued on next page)

1977	1978	1979	1980	1981	1982
2546651900	2364177200	2606193900	2825771600	2875982900	2935944200
2392061100	2159990000	1916882700	1520426500	1195817000	1004856000
1601831400	1343175120	1098250200	874909700	627404500	
324498200	326104800	260604300	191637800	164686000	
178812400	173456696	174138600	88867600	62064500	
149604000	145425560	158143200	108222200	78041500	
137315100	171827824	225746400	256789200	263620500	
					1004856000
124670000	68888400	555312000	1029746500	1256038000	
115765000	48139280	399130500	780453800	957391000(g)	
8905000	20749120	113929500	168466800	137648000(h)	
		10563000	22353200	40557000(i)	
		31689000	58472700	120442000(j)	
					1441811000
					1030539400
					151627600
					259644000
29920800	108040800	97330500	73738300	100409300	
15316600	35683200(m)				
3205800	13876800	29123700	24942900	47070700(n)	
8726900	40675544	15090000	14856700	11429700(o)	
2671500	8884456	15693600	13084800	19049500(o)	
	8920800	15542700	17582700	19172400(i)	
		6036000			
		15844500			
			3271200	3687000(p)	
	27258000	36668700	201860300	323718600	
	24449600	22785900	162605900	262391500(r)	
	2808400	9506700	13766300	23351000(o)	
		4376100	25488100	37976100(n)	
					489277200
					80141200
					401942400
					7193600
1.781	1.652	1.509	1.363	1.229	1.124

(continued on next page)

1983	1984
3245316090	3048300000
1827632790	1646300000
957240690	724900000
260831700	277700000
609560400	638200000
	5500000
1417683300	1402000000
336232500	343300000
719272800	729500000
362178000	329200000
1.059	1.000

K.4 Footnotes for Spreadsheets:

(a) DDT&E includes procurement of:
Structural test article (STA) for orbiters
Orbiters 101 (Enterprise) and 102 (Columbia)
At least 3 SSMEs for individual tests
3 SSMEs for propulsion test article
6 flight SSMEs (for 101 and 102)
Roughly 3 ground test External Tanks
4 ETs for the development flights
2 SRB structural test articles
4 SRBs for development test firings
3 SRBs for qualification test firings
5 SRBs for ground tests (1 struc, 4 vibration)
8 SRBs for the development flights
Ground equipment at KSC to support one Shuttle
Orbital test flights STS-1 through STS-4

(b) Shuttle costs are under "Space Flight Operations" in 1972 and 1973 books.

(c) Vehicles were intended to be orbiter and (possibly flyback) booster.

(d) From here on, current Orbiter/ET/SRB configuration.

(e) Listed under Space Transportation Capability Development in 1984 book.

(f) Listed under Space Shuttle. Production includes:
Modification of STA into 099 (Challenger)
Start of orbiters 103 (Atlantis) and 104 (Discovery)
(funding completed under Shuttle Production and Operational Capab.)
Ground equipment to handle second STS at KSC
Spares for STS fleet
SSMEs (109% rated) for 099,103
Modifications to Columbia after STS-5
Note: funding for all these items continues under Shuttle Production and Operational Capability.

(g) Funding continues under Shuttle Production and Operational Capability - Orbiter.

(h) Funding continues under Shuttle Production and Operational Capability - Propulsion systems.

(i) Becomes part of Shuttle Production and Operational Capability - Launch and mission support.

(j) Funding continues throughout Shuttle Production and Operational Capability.

(k) Listed as R&D - Space Transportation Capability Development - Shuttle Production and Capability Development, in 1984 book;
then becomes Space Flight, Control and Data Communications - Space Transportation Systems - Shuttle Production and Operational Capability, in 1985 book and beyond.
Shuttle Production and Operational Capability includes:
Completion of Orbiters 103 and 104
Completion of modifications to 102
Modifications to orbiters to carry Centaur
Procurement of 109% rated SSMEs
Spares for STS fleet
Improvements to ground equipment at KSC
Upgrade of Mission Control Center at JSC
Development of filament-wound composite SRB
Upgrade of ET production processes

(l) Listed as R&D - Space Flight Operations - Space Transportation System Operations Capability Development

(m) Becomes a separate line item under Space Flight Operations in 1981 book.

(n) Becomes part of R&D - Space Transportation Capability Development - Upper Stages, in 1984 book and beyond.

(o) Becomes part of R&D - Space Transportation Capability Development - Payload Operations and Support Equipment, in 1984 book and beyond.

(p) Becomes part of Shuttle DDT&E.

(q) Listed as R&D - Space Flight Operations - Space Transportation System Operations.

Space Transportation System Operations includes:
Procurement of ETs and SRBs for operational flights (after STS-4)
Mission-specific procedures development and training
Procurement of upper stages for NASA payloads
Payload integration at KSC
Launch operations for operational flights
Propellants
Vehicle refurbishment

(r) Becomes Shuttle Operations - Flight operations, Flight hardware, Launch and landing operations, in 1984 book and beyond.

(s) Listed as R&D - Space Transportation Operations - Shuttle Operations, in 1984 book; then listed as Space Flight, Control and Data Communications - Space Transportation Operations - Shuttle Operations, in 1985 book and beyond.