Equipment Selection and Task Assignment for Multiproduct Assembly System Design

by

Stephen C. Graves*
and
Carol A. Holmes**

WP #1895-87 June 1987

* Sloan School of Management
Massachusetts Institute of Technology
E53-390
Cambridge, MA 02139

** Decision Focus Incorporated
4984 El Camino Real
Los Altos, CA 94022

ABSTRACT

A multiproduct assembly system produces a family of similar products, where the assembly of each product entails an ordered set of tasks. An assembly system consists of a sequence of work stations. For each work station, we assign a subset of the assembly tasks to be performed at the work station and select the type of assembly equipment or resource to be used by the work station. The assembly of each product requires a visit to each work station in the fixed sequence. The problem of system design is to find the minimum cost system that is capable of producing all of the products in the desired volumes. The system cost includes the fixed capital costs for the assembly equipment and tools, and the variable operating costs for the various work stations. We present and illustrate an optimization procedure that assigns tasks to work stations and selects assembly equipment for each work station.

INTRODUCTION

Historically, the design of an assembly line has been synonomous with line balancing. Assembly line balancing is appropriate for a labor-intensive assembly environment where the primary goal is to distribute the work equally among assembly workers to minimize labor costs. Today, the assembly environment is becoming less labor intensive and more capital intensive. In this new environment, various forms of automated equipment, such as robots, are available for inclusion in an assembly system. This new equipment has the potential both of being more cost effective, and of providing better quality and reliability than manual assembly. Furthermore, this equipment, especially robots, is becoming more adaptable and flexible, although manual assembly will always dominate on these dimensions. Acquisition of this automated equipment, however, requires a substantial capital investment. High fixed costs for equipment can create a situation where the least-cost assembly system is not necessarily the most-balanced assembly line. This suggests the need for new methods for assembly system design that evaluate the equipment choices with the goal of finding the least-cost assembly system.

We consider the problem of designing a flexible assembly system that is least cost and is capable of assembling a family of products. We are given the exact order of assembly operations (or tasks) for each product, and a list of candidate resources (or equipment) available to complete the operations. The design problem is to decide which resource types to select and which operations to assign to each resource so as to meet production requirements for the set of products and minimize total system cost.

We expect that the set of products consists of a family of related products, or different models of the same product. The variation between products is not great and may entail one or more task substitutions, or may consist of additional tasks that one product requires which the others do not.

Resources to perform the assembly operations include humans, fixed automation such as a transfer line station, and various programmable machines such as robots. A resource type may have the capability of performing all of the assembly tasks, or it may be a special-purpose machine capable of doing only a particular subset of the operations. The annualized fixed cost of each resource and any necessary tooling are explicitly considered in the computation of total cost. Associated with each resource is the time needed to complete each operation and the time needed to change tools.

Given this information, the problem is to design a minimum-cost assembly system with sufficient capacity to meet annual production requirements. An assembly system consists of a sequence of work stations. Associated with each work station is a resource (e.g., a robot) and a set of tasks that are assigned to the station. The assembly of each product entails visiting each work station in the fixed sequence. Production requirements are met by imposing a cycle time requirement for each product on the system. The cycle time of a product is the maximum of the processing times for the work stations in the assembly systems, and is equivalent to the maximum production rate possible for the product.

The remainder of the paper is organized into four sections. In the next section, we briefly review related work on assembly system design, and motivate the formulation that we address. We then explain our solution algorithm to the multiproduct equipment-selection problem by means of a detailed example. In the third section, we illustrate the multiproduct equipment-selection problem with a case study for the assembly of an automobile steering column. Finally, in the last section, we discuss possible model extensions and refinements.

BACKGROUND AND MOTIVATION FOR THE CURRENT FORMULATION

Almost all of the work on assembly system design is on the problem of assembly line balancing. This problem is appropriate for manual assembly environments in which the primary controllable cost is the labor cost. Assembly line balancing attempts to reduce labor cost by balancing the work load over the line to eliminate idle time. A recent survey of optimization approaches is given by Baybars [1986].

Pinto, Dannenbring and Khumawala [1983] extend the simple assembly line balancing problem to include processing alternatives (limited equipment selection) by relaxing the assumption that all work stations are identical. Their method determines if, for an incremental fixed cost, the balance of an assembly line can be improved and the total system cost reduced by selecting one or more of a set of processing alternatives. Since the core of their method is still assembly line balancing, its applicability is geared to systems that are primarily manual assembly.

Both of the above methods are restricted to the assembly of a single product type. Thomopolous [1967] extends line balancing to mixed model assembly, but he does not address the equipment selection problem.

The formulation of the equipment selection problem that we present is part of a two phase approach to assembly system design developed by researchers at the Charles Stark Draper Laboratory, Inc. (CSDL). This work stresses a global approach to assembly system design, integrating product and process design with equipment selection. In the first phase of CSDL's approach, candidate assembly sequences for a product are enumerated by a liaison sequence analysis method (DeFazio and Whitney, 1986). The second phase, the equipment selection problem, then takes as given a fixed sequence of assembly tasks for each product and seeks to find the least-cost assembly system design for that sequence. Since there may be several candidate assembly sequences, the equipment selection problem would have to be solved for each candidate.

Both heuristic and optimization methods for the equipment selection problem have been developed at CSDL. Gustavson [1986] has implemented heuristic methods which seem to find very good solutions to both the single and multiple product case of the equipment selection problem. As in any heuristic method, these solution methods cannot guarantee that an optimal solution will be found. One reason for developing an optimization approach to the multiproduct equipment selection problem (MESP) was to calibrate the effectiveness of Gustavson's heuristic methods.

Graves and Whitney [1979] first formulated an optimization method for the single-product equipment selection problem in 1979. Similar to the MESP, the stated goal of the problem was to select equipment and make task assignments so as to minimize the sum of fixed and variable costs. They assumed a fixed sequence of tasks as in the MESP, but they permitted non-serial line layouts in which an assembly unit may return more than once to a given station. Their model did not account explicitly for tool change times and tool costs, which was a serious drawback to the model. The problem was formulated as a mixed integer program and was solved using branch and bound and a subgradient optimization procedure.

Graves and Lamar [1981] extended the formulation of Graves and Whitney to explicitly include tool change times in the formulation. The problem was formulated as an integer program and solved by finding upper and lower bounds for a linear relaxation of the problem. The integer programming problem as formulated was a very large problem whose computational requirements grew exponentially with the number of candidate resources. As a result of allowing unresticted floor layouts for the problem, the solutions found by both of these early formulations were not necessarily physically realizable.

The present MESP formulation is intended to address the limitations of the two earlier methods. In particular, the new formulation includes the following four extensions:

(i) guarantees the feasiblity of the layout by restricting the system to a serial linear floor layout;

(ii) explicitly models tool costs, as well as tool change times;

(iii) is implemented on a PC;

(iv) and considers multiproduct assembly systems.

DETAILS OF THE OPTIMIZATION METHOD

In this section we present the optimization method for the multiproduct equipment-selection problem. The optimization criterion is to find the assembly system that is least cost among all design possibilities. The first step of the solution enumerates all candidate work stations for the system and selects the least-cost resource type for each candidate work station. To find the least-cost assembly system, we then construct a graph in which each candidate work station corresponds to an arc. We obtain the least-cost assembly system by solving a shortest path problem on this graph. We describe the solution procedure in detail using a simple example as a means of explanation.

Consider an assembly system that is to assemble two different model types. Model A requires tasks (1,2,3,5,6,8,9,10,12) for assembly, and Model B requires tasks (1,2,4,5,6,7,10,11,12). Thus, the models are nearly identical except that model B substitutes task 4 for task 3 on model A, substitutes task 7 for task 8 and 9 on model A, and has an additional task 11. For each model, the tasks must be completed in the exact sequence given. However, between models there is flexibility in the order in which tasks may be assigned to work stations. For example, since Model A does not require task 4 and Model B does not require task 3, there is no fixed order between these tasks. Thus, we may assign task 4 to a work station that precedes the work station that performs task 3. As a result, we can construct a partial ordering of the total set of tasks. A convenient way to represent this partial ordering is with a network diagram, as given in Figure 1 for this example, in which the nodes represents tasks and the arcs denote a precedence relationship.

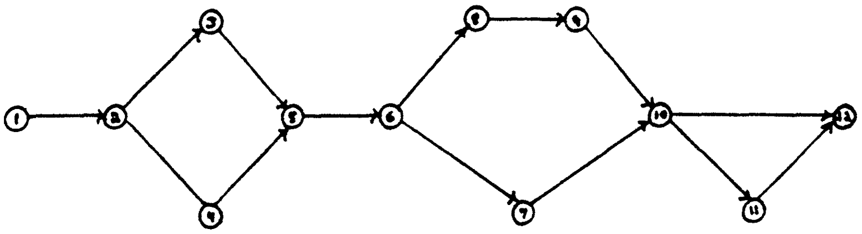


Figure 1

Figure 2 gives information on the two resource types that are available to complete the assembly tasks. As shown, for each resource the operational times, tool numbers and annualized tool costs needed to complete each task are given. It is important to note that a resource may not be able to perform all tasks. For instance, resource type 2 cannot perform tasks 1,11,and 12. Also listed in Figure 2 are the annualized fixed costs and the variable cost per hour associated with each resource. This variable cost includes both the labor cost and the variable operating cost for the resource.

Each resource may require an additional piece of tooling to perform a particular task. For example, a robot might require a special gripper or a bolt-driver depending on the task. However, tasks that are assigned to the same work station may share the same tool. For instance, in the example in Figure 2, the same tool can be used for tasks 3, 4, and 5. Thus, if these tasks are assigned to the same work station, only one tool need be purchased; however, if these tasks are assigned to different work stations, we must purchase one tool for each work station. Furthermore, if successive tasks that require the same tool are assigned to the same work station, we may avoid having to change tools between these tasks at that work station. Otherwise, the work station will always incur a tool change time between successive tasks for each model.

	RESOURCE TYPE 1			RESOURCE TYPE 2		
Task	Operation Time (seconds)	Tool #	Tool Cost ($)	Operation Time (seconds)	Tool #	Tool Cost ($)
1	5.6	100	11000	---	---	----
2	3.6	120	8000	2.4	220	8000
3	3.6	121	3000	2.4	221	3000
4	1.8	121	3000	1.8	221	3000
5	1.8	121	3000	1.8	221	3000
6	3.6	131	8000	2.4	231	3000
7	3.6	141	7000	3.0	241	7000
8	2.0	142	2000	2.0	242	2000
9	4.0	150	7000	3.6	250	7000
10	7.2	160	4000	7.2	260	4000
11	5.4	170	10000	---	---	----
12	5.4	170	10000	---	---	----
Annualized fixed cost		$40,000			$50,000	
Variable cost/hour		$4.8			$5.3	
Tool change time(seconds)		2.0			2.0	

Figure 2

We assume a required production of 216,000 units per year for each model. This annual volume will dictate the cycle time for each model. The cycle time is the rate at which the assembly system needs to complete one unit to ensure that the annual volume requirement is met. For each model, the cycle time is given by the production time available per year for the model divided by its desired annual volume. In our example, we assume that the available production time per year for assembly of both models is one eight-hour shift per day, operating for 240 days per year (i.e., 1920 hours per year). If we assume that each model gets exactly half of this available time, then each model will have 960 hours of assembly time to produce 216,000 units. Thus, the assembly system needs to be capable of producing at least one unit of either model each 16 seconds. That is, the cycle times for both model A and B can be at most 16 seconds. Note that we do not require that the cycle time be the same for every model. If we were to make a different assumption about how the available time is split between the two models, we would obtain differing cycle times in this example.

The station-to-station move time for a system is the time required to move(load) a unit of product into a work station and subsequently to remove(unload) the unit from the work station. Since this time is not available for assembly, the effective cycle time is the cycle time computed above minus the time to load and unload a work station. In this example, this move time is 2 seconds, so that the cycle time is reduced to 14 seconds per unit for each model. We call this the target cycle time for each model for the system.

Given the above data, the first step of the solution is to enumerate all candidate work stations for the system. A candidate work station is a subset of tasks, which can be assigned to a single work station. That is, it is a subset of tasks that could be performed at a single work station, provided that all preceding tasks are assigned to earlier work stations.

To explain how we generate these candidate work stations, consider Figure 3. Define a cut set as a group of tasks such that for any task in the set, all the predecessors of that task are also in the set. Each dashed line shown in Figure 3 is a "cut" that determines a cut set. For example, cut E determines the cut set (1,2,4) and cut K determines the

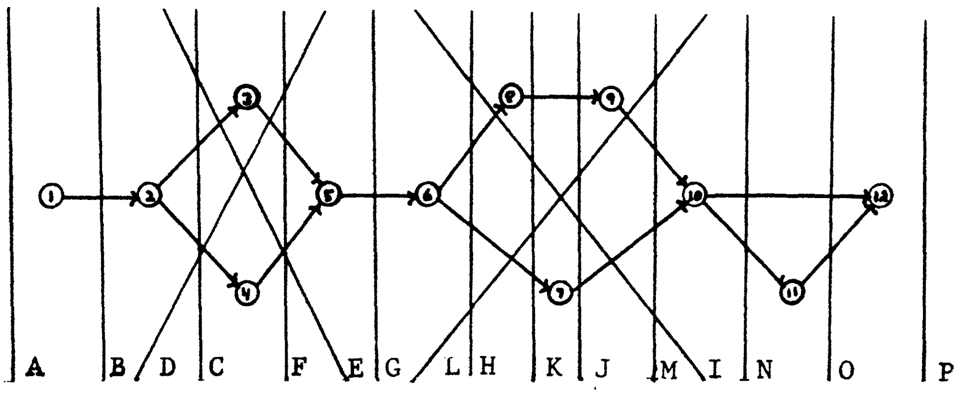


Figure 3

cut set (1,2,3,4,5,6,8). A candidate work station can now be represented as the difference between two cut sets. For instance, the work station {2,3,4,5} is given by G - B. A work station may be identified by more than one pair of cuts. For instance, the work station {4} is given by E - D, by E - C, and by F - D. In general, to generate all candidate work stations, we need consider all pairs of cut sets in which one cut set is a subset of the other. For example, the pair (A,G) is the work station (1,2,3,4,5) and the pair (E,G) is the work station (3,5). Pairs such as (D,E), where cut set D is not a subset of cut set E, need not be considered since another pair will generate the same work station.

To enumerate candidate work stations, we need an efficient algorithm for enumerating the cut sets in a network. We use an algorithm by Schrage and Baker [1978], which extends to a partially-ordered set the standard binary counting method for enumerating all subsets of an unordered set. The implementation of this algorithm identifies all of the cut sets. The candidate work stations are then enumerated by considering pairs of these cut sets.

Not every work station that satisfies the precedence constraints is a feasible work station. A candidate work station is feasible if it satisfies the condition that at least one resource type is able to accomplish all of the tasks within the target cycle time for each model. To determine whether a candidate work station is feasible, we need to determine for each resource and for each model the time to complete the set of tasks.

The time to complete a set of tasks at one work station is the sum of the task times plus any necessary tool change times. If the tool change for the initial task occurs simultaneously with the station-to-station move time, then it may be possible to overlap part of the tool change time with the station-to-station move time.

Consider the feasibility of workstation (4,5,6,7). Such a workstation would perform tasks 5 and 6 for model A, and tasks 4,5,6, and 7 for model B. The computation of total station time for resource type 1 is: (See Figure 2 for task times.)

Resource 1	Product	Operation Time(secs)	Tool Change Time(secs)	Total Station Time(secs)
	1	5.4	2.0	7.4
	2	10.8	4.0	14.8

Resource 1 is feasible for product 1 since the total station time, 7.4 seconds, is less than the target 14 seconds. However, since the total station time for product 2, 14.8 seconds, is greater than 14 seconds, resource type 1 is infeasible for product 2, and hence, is infeasible for this work station. The computation of total station time for resource type 2 is:

Resource 2	Product	Operation Time(secs)	Tool Change Time(secs)	Total Station Time(secs)
	1	4.2	2.0	6.2
	2	10.0	4.0	14.0

The feasibility of both products in this case implies that resource 2 is feasible for the work station. Clearly then, resource 2 is the least-cost resource type for the work station since it is the only feasible resource type.

As a second example consider the work station (3,4,5) which is feasible for both resources.

Resource 1	Product	Operation Time(secs)	Tool Change Time(secs)	Total Station Time(secs)
	1	5.4	0	5.4
	2	3.6	0	3.6

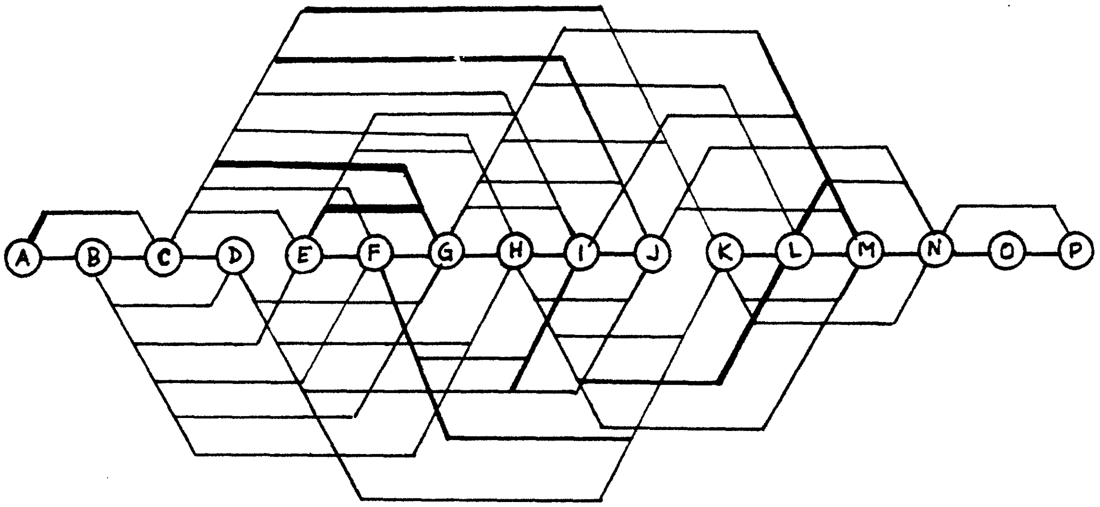
Resource 2	Product	Operation Time(secs)	Tool Change Time(secs)	Total Station Time(secs)
	1	4.2	0	4.2
	2	3.6	0	3.6

The work station cost for each resource type must be computed to select the least-cost resource for this work station. Work station cost has both a fixed and a variable part. The fixed part is the sum of the annualized fixed cost for the resource type and the annualized cost of the necessary tools. The variable cost is the product of the variable cost per hour for the given resource and the total hours the system will be in operation. For the current example, total system time is 1920 hours from before. The total station cost is computed for each feasible resource type and the least-cost resource type for the candidate work station is selected. For work station (3,4,5) the following chart shows the total station cost computation. For this work station, resource 1 is the least cost resource.

	Resource 1		Resource 2	
Annualized Fixed Cost		$40000		$50000
Annualized Tool Cost		$3000		$3000
Variable Cost/hour	$4.8		$5.3	
Hours in production	1920		1920	
Total Variable Cost		$9216		$10176
Total Station Cost		$52216		$62176

Once the least-cost resource type has been determined for each candidate work station, we seek to find the least-cost set of work stations such that each task is assigned to exactly one work station. To find the least cost system we consider the network diagram in Figure 4.

NETWORK REPRESENTATION
OF FEASIBLE WORK STATIONS



Each node in the diagram is a cut
from the original task set diagram.

Each arc connects a pair of cuts and
represents a candidate work station.

Figure 4

The nodes in this diagram represent the cuts from the original task set diagram. The arcs in this diagram are feasible work stations that correspond to pairs of cuts. For example, Arc (E,G) is the work station with tasks (3,5). The cost for each arc (work station) is the station cost for that resource which was selected as least cost in the previous step. Finding the least-cost system is then a matter of finding the least-cost path from the start node A to the terminal node O in the diagram. This is a shortest path problem where the length of each arc is defined to be its associated cost.

The least-cost system for the example, given in Figure 5, has three stations of resource type 1 and one station of resource type 2.

LEAST COST SYSTEM

Work Station	Resource Type	Task #'s	Tool #'s	Station Cost
1	1	1,2	100, 120	$68216
2	2	3,4,5,6,7,8	221,231,241,242	$75176
3	1	9,10	150, 160	$60216
4	1	11,12	170	$59216
TOTAL SYSTEM COST		$262,824		

WORK STATION SUMMARY BY PRODUCT

	Product 1		Product 2	
Work Station	Station Time(secs)	Tasks	Station Time(secs)	Tasks
1	11.2	1,2	11.2	1,2
2	12.6	3,5,6,8	13.0	4,5,6,7
3	13.2	9,10	7.2	10
4	5.4	12	10.8	11,12

Figure 5

Resource 2 is least cost for the second work station because resource 1 requires more time to do the assembly operations than resource 2 and cannot complete the tasks at the second work station in the allotted cycle time. Six tasks have been assigned to the second work station, but examining the system by product shows that a maximum of four tasks will be done at station 2 at one time, four by product 1 and four by product 2.

The maximum station time of the least-cost system is 13.2 seconds for product 1 and 13.0 seconds for product 2. These maximum station times are the actual cycle times of the system. Since the required annual volume implied a target cycle time of 14 seconds for each product, this system has a capability to exceed the required annual production volume.

Implementation Issues

We have implemented this solution algorithm for the MESP in BASIC on an IBM PC XT. Our implementation has been more than adequate for the problems that we have attempted to solve. For instance, a problem with 28 tasks, 3 models, and 5 resource types ran in under 3 minutes on the IBM PC XT. We have not used any special data structures or special implementations of the recursive computations to improve performance or reduce storage requirements. Nevertheless, this is possible (Kao and Queyranne, 1982) and may be necessary if we encounter problems with many more tasks or with less structure.

The theoretical maximum on computation time and storage requirements for this algorithm is extremely large. In the worst case, the algorithm requires the complete enumeration of all possible subsets of a given task set. However, realistic task sets will require substantially less work due to the structure of their precedence diagram. For instance, the example required the examination of only 120 candidate work stations (equal to the number of pairs of cuts in Figure 3), whereas there are 4095 ($2^{12}-1$) subsets of the task set.

CASE STUDY FOR STEERING COLUMN ASSEMBLY

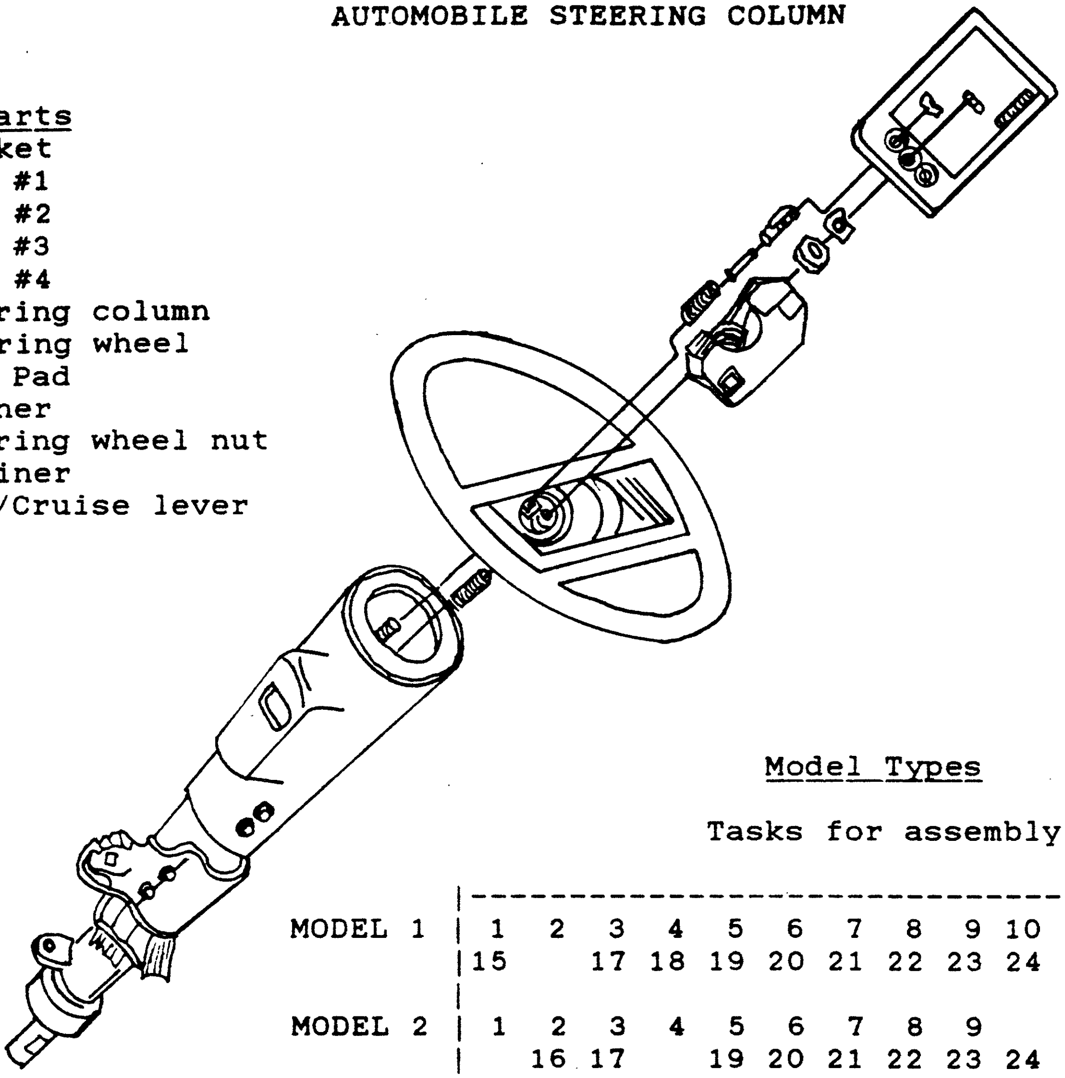
As an application of the multiproduct equipment selection system, consider a design problem from the automotive industry. A manufacturer has to assemble three models of a steering column as a subassembly for an automobile. The three models have most of their parts in common, but options include a turn/cruise lever, a hazard switch, and a tilt option on the steering wheel. The targeted annual volume of product is 250,000 units with models one, two, and three representing 63, 29, and 8 percent of the total volume. This is a hypothetical example based on the assembly specifications of a real product. The multi-model dimension of the problem was fabricated to demonstrate the MESP system.

Steering Column Assembly Data

Figures 6 and 7 give the assembly specifications for the three models of the steering column and the resources available to complete the assembly. The tasks necessary for the assembly of each model, given in Figure 6, identify which options will be included on each model type. Model 1 has all the options including a dampner, a turn/cruise lever, a hazard switch, and a tilt lever. Model 2 is the basic model with no options. Model 3 has the dampner and turn/cruise lever, with no tilt lever or hazard switch.

Figure 7 identifies the three resource types available to complete the steering column assembly. The resource types available for this application are manual, fixed automation, and a paint machine (used only for painting the steering column). A programmable machine (robot) was

AUTOMOBILE STEERING COLUMN



Parts

A	Bracket
B	Bolt #1
C	Bolt #2
D	Bolt #3
E	Bolt #4
F	Steering column
G	Steering wheel
H	Horn Pad
I	Dampner
J	Steering wheel nut
K	Retainer
L	Turn/Cruise lever

Model Types

Tasks for assembly

MODEL 1	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	15		17	18	19	20	21	22	23	24	25	26	27	28
MODEL 2	1	2	3	4	5	6	7	8	9		11	12	13	14
		16	17		19	20	21	22	23	24				28
MODEL 3	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	15		17		19	20	21	22	23	24		26		28

Task #'s refer to Figure 7

Figure 6

AUTOMOBILE STEERING COLUMN
Available Resources

	Manual	Fixed Automation	Paint Machine
Fixed Resource Cost($)	262.5	7875	18375
Var. Operating Cost($/hr)	23.8	5.7	24.8
Tool Change Time(secs)	2.5	2.0	2.0

Task Data

		Manual Labor			Fixed Automation		
		Task		Tool	Task		Tool
#	Task Description	Time	Tool	Cost	Time	Tool	Cost
		(secs)	#	($)	(secs)	#	($)
1	Schedule steering column	23	100	0	---		
2	Bracket & bolt to column	8	100	0	---		
3	Finger start bolt #2	3	100	0	---		
4	Finger start bolt #3	3	100	0	---		
5	Finger start bolt #4	3	100	0	---		
6	Paint steering column	---			---		
7	Schedule steering wheel	17	100	0	---		
8	Schedule horn pad	9	100	0	---		
9	Steering wheel to column	7	100	0	6	201	8925
10	Place dampner to column	7	102	9975	6	202	12600
11	Drive steering wheel nut	6	103	9975	3	203	12600
12	Inspect nut torque	2	103	9975	2	203	12600
13	Install nut retainer	2	104	1313	1	204	5775
14	Visual inspect retainer	2	100	0	---		
15	Install trn/cruise lever	14	105	525	---		
16	Install turn lever	8	100	0	4	205	7350
17	Horn pad to wheel	8	100	0	5	206	8925
18	Install tilt lever	7	100	0	3	207	6825
19	Secure bracket bolt #1	3	108	6825	1	208	14700
20	Secure bracket bolt #2	3	108	6825	1	208	14700
21	Secure bracket bolt #3	3	108	6825	1	208	14700
22	Secure bracket bolt #4	3	108	6825	1	208	14700
23	Test bracket secureness	2	100	0	---		
24	Test horn pad secureness	2	100	0	---		
25	Install hazard switch	9	109	1313	---		
26	Test turn/cruise lever	21	110	15750	---		
27	Test hazard switch	7	110	15750	---		
28	Electrical test horn pad	12	110	15750	---		

Task #6 can only be completed by a Paint Machine.
Operation Time = 43 secs. Tool # = 401. Tool Cost = $7875.

A dash(---) indicates that a resource cannot perform the task.

Figure 7

considered in the original problem formulation, but it was too expensive to ever enter the optimal solution. Figure 7 gives the annualized fixed costs and the variable costs associated with each resource. The annualized fixed cost for each resource includes both the fixed investment cost of a work station plus the expected cost to install the work station. Figure 7 also gives, for each resource, the operation time to complete one unit of the task, the tool needed, and the annualized fixed cost of the tool for each ˛f the 28 tasks.

For this example, we assume that there are 235 working days per year, each day having two eight-hour shifts. We have allocated the available production time between the models according to the portion of annual demand that the model represents. This forces the cycle time to be the same for all models. This is an appropriate allocation of available production time when the time to assemble a unit of each of the models is approximately the same for all models. Assuming an annual total demand of 250,000 units and a station-to-station move time of 4 seconds, we can compute the target cycle time for each model (after removing the move times) to be about 50 seconds per unit.

Optimal System Configuration: 250,000 Units

Figure 8 gives the optimal system design for the steering column for an annual total demand of 250,000 units. At this volume, the optimal system has six work stations: five manual stations and one paint machine for task #6. For each work station, the tasks assigned to the work station and the necessary tools are given. Total run time for the computer program was just under 3 minutes on an IBM PC XT.

OPTIMAL SYSTEM CONFIGURATION

Annual Demand = 250,000 units
Available Cycle Time for all Models = 50 seconds

Station	Resource	Task #'s	Tool #'s	Station Cost ($000's)
1	manual	1 - 5	100	89.8
2	paint	6	401	119.5
3	manual	7 - 10	100,102	99.7
4	manual	11 - 17	103,104,100,105	101.6
5	manual	18 - 25	100,108,109	97.9
6	manual	26 - 28	110	105.5

Total System Cost = $614.0

SYSTEM CONFIGURATION BY MODEL

		Model 1		Model 2		Model 3	
Station	Resource	Station Time(sec)	Tasks	Station Time(sec)	Tasks	Station Time(sec)	Tasks
1	manual	40	1-5	40	1-5	40	1-5
2	paint	43	6	43	6	43	6
3	manual	42.5	7-10	33	7-9	42.5	7-10
4	manual	44	11-15,17	33	11-14,16, 17	44	11-15,17
5	manual	39.5	18-25	18.5	19-24	18.5	19-24
6	manual	40	26-28	12	28	33	26,28

Figure 8

We note that the optimal assembly system has actual cycle times, equal to the maximum station time for each model, that are significantly less than the target cycle time for each model. Whereas the target cycle times were about 50 seconds per unit for each model, the actual cycle times are 44 seconds for models 1 and 3, and 43 seconds for model 2. As a consequence, the optimal system has an actual production capability that is much larger than needed. Alternately, the system can produce the desired production volume of 250,000 units in 88% of one year or by working shorter days. Then, if the variable cost is truly variable, the actual variable cost to produce 250,000 units will be less than predicted since it depends on the actual rather than the target cycle time. In this example, the actual system cost would be reduced from $614,000 to $554,000 if we could either run the system for just 88% of the year or reduce the length of each day an equivalent amount.

Sensitivity Analysis

The all-manual solution (except for the paint machine) for the 250,000 unit problem is not a particularly interesting result for an equipment selection problem. More interesting results occur when we examine the sensitivity of the solution to different volume levels. At volumes of 300,000 units or more, we needed to permit "parallel" resources as candidate resources for the assembly system. A parallel resource is simply a double resource assigned to a single set of tasks. For instance, in Figure 9, MA2 is a station with two people working in parallel on the given set of tasks. Parallel resources can, in general, do the same set of tasks as a single resource in half the time for twice the fixed and variable cost. The MESP system has an option that will automatically add parallel resources, as necessary, when evaluating candidate work stations.

Figure 9 is a summary of the optimal assembly system found for volumes ranging from 150,000 to 500,000 units. As might be expected, fixed automation becomes increasingly attractive at higher demand levels. With greater demand, the high fixed costs associated with fixed automation are offset by the increased speed of assembly and lower variable cost. It is more surprising, though, that at 150,000 units, fixed automation also entered the optimal assembly system.

OPTIMAL SYSTEMS FOR DIFFERENT VOLUMES

ANNUAL DEMAND VOLUME	SYSTEM COST (000'S)	1	2	3	4	5	6	7	8
150000	$473	MA1 1-5	PT1 6	MA1 7-16	FXD 17	MA1 18-28			
200000	$524	MA1 1-5	PT1 6	MA1 7-14 16	MA1 15 17-24	MA1 25-28			
250000	$614	MA1 1-5	PT1 6	MA1 7-10	MA1 11-17	MA1 18-25	MA1 26-28		
300000	$769	MA1 1-5	PT2 6	MA1 7-9	FXD 10-12	MA1 13-18	MA1 19-25	MA1 26-28	
400000	$933	MA1 1	MA1 2-5	PT2 6	MA1 7-8	FXD 9-13	MA1 14-17	FXD 18-22	MA2 23-28
500000	$1111	MA2 1-5	PT2 6	MA1 7	MA1 8-9	FXD 10-13	MA1 14-16	FXD 17-22	MA1 23-25
		MA1 26	MA1 27-28						

For each station the assigned resource and tasks are shown.

MA1 = SINGLE MANUAL STATION MA2 = DOUBLE MANUAL STATION
PT1 = SINGLE PAINT STATION PT2 = DOUBLE PAINT STATION
FXD = FIXED AUTOMATION

Figure 9

The main conclusion that we can draw from this analysis is that the solution may fluctuate greatly with demand level. At 250,000 units, the least cost system is all manual, but at lower and higher demands fixed automation becomes an attractive addition to the system. Furthermore, specific tasks are alternately assigned to manual stations and fixed automation, depending on the required volume. The inconclusiveness of this analysis suggests that care is needed in using the results from the MESP program.

For many design studies, the required volume may be highly uncertain and/or there may be expectations of significant growth (or decline) in the required volume over time. As a result, one would want to understand both how sensitive the optimal solution is to the required volume as well as how the assembly system could evolve over time as the requirements grow. For now, the most we can do is a parametric analysis as illustrated in Figure 9.

UNRESOLVED ISSUES

In this paper we have described an optimization procedure for solving a multiproduct equipment-selection problem for assembly system design. This procedure is implemented on an IBM PC XT, and has been tested on realistic assemblies and subassemblies through project work at CSDL. Nevertheless, there are several outstanding issues that will require additional research and development.

Allocation Of Production Time For The Multiproduct Case

When using the MESP system, the cycle time for each product is determined before the optimal system is found. In a multiproduct case, the available production time must be allocated between the products in order to compute a cycle time for each product. We have not adequately answered the question of how to select the "best" allocation of production time for a given application. This remains an unresolved issue and further research into methods for allocating the available production time between products is needed.

System Reliability

We define system reliability as the fraction of available time that a system is expected to be functioning properly. In the MESP program, we account for this reliability by assigning a Percent Uptime Expected to each resource. This percent is used to compute the total accumulated work station time under the assumption that the station will be down a given percent of the available production time. By accounting for the reliability of each resource separately we assume that a system continues to function when one station fails (due to low utilization and/or sufficient buffer stock). This may not be the case. If the failure of one resource shuts down the entire system, then a system-wide reliability factor may be more appropriate. As the MESP is currently formulated, it is impossible to determine a system-wide reliability before the resources are selected. Using only an individual percent uptime expected for each resource oversimplifies the reliability issue, but it is the best method we have found for the given problem formulation.

Changeover Time Between Products

A final unresolved issue involves the changeover time between products. Each time an assembly line is converted to assemble a different product (model), some of the available production time may be lost due to a changeover period. The amount of time needed to convert the line depends on which resources have been selected. The longest changeover time among the selected resources determines the changeover time for the system. Before the system is configured, and the resources are selected, we can not determine an exact value for the changeover time. In the current formulation, we assume that an estimate of the annual changeover time has been subtracted from the available production time. For example, if there were 240 day/year, we might assume that 5 days would be lost to changeover time between models, which would leave 235 days/year for assembly time. Possibly, a more accurate way to account for changeover time could be found.

ACKNOWLEDGEMENT

This research was supported by the National Science Foundation under Grant DMC-84-17949 made to the Charles Stark Draper Laboratory, Inc. The authors would also like to thank Dr. Daniel E. Whitney and Richard Gustavson for their help and support throughout the duration of this research.

REFERENCES

1. Baybars, I. "A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem," Management Science, 32, August 1986, 909-932.

2. DeFazio, T.L. and D.E. Whitney, "Simplified Generation of All Mechanical Assembly Sequences," C.S. Draper Laboratory Report No. P-2709, Cambridge, Massachusetts, June 1986.

3. Graves, S.C. and B.W. Lamar, "An Integer Programming Procedure for Assembly System Design Problems," Operations Research, 31 (1983), 522-545.

4. Graves, S.C. and D.E. Whitney, "A Mathematical Programming Procedure for Equipment Selection and System Evaluation in Programmable Assembly," Proceedings of the Eighteenth IEEE Conference on Decisions and Control (1979), 531-536.

5. Gustavson, R.E., Design of Cost-Effective Assembly Systems," C.S. Draper Laboratory Report No. P-2661, Cambridge, Massachusetts, 1986.

6. Holmes, C. A., "Equipment Selection and Task Assignment for Multiproduct Assembly System Design," Master's Thesis, Operations Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1987.

7. Kao, E.P.C. and M. Queyranne, "On Dynamic Programming Methods for Assembly Line Balancing," Operations Research, 30, March 1982, 375-390.

8. Lamar, B.W., "Optimal Machine Selection and Task Assignment in an Assembly System Design Problem," Master's Thesis, Operations Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1980.

9. Pinto, P.A., D.G. Dannenbring and B.M. Khumawala, "Assembly Line Balancing with Processing Alternatives: An Application," Management Science, 29, July 1983, 817-830.

10. Schrage, L. and K.R. Baker, "Dynamic Programming Solution of Sequencing Problems with Precedence Constraints," Operations Research, 26, June 1978, 444-449.

11. Thomopolous, N.T., "Line Balancing-Sequencing for Mixed-Model Assembly," Management Science, 14, October 1967, 59-75.

12. Whitney, D.E., T.L. DeFazio, R.E. Gustavson, S.C. Graves, C. Holmes, and C.J. Klein, "Computer-Aided Design of Flexible Assembly Systems, First Report," C.S. Draper Laboratory, Inc. Report No. CSDL-R-1947, Cambridge, Massachusetts, August 1986.