REAL-TIME DISPATCHING OF DELIVERY VEHICLES

by

ALAN SETH MINKOFF

A.B. Statistics, Princeton University
(1980)

SUBMITTED TO THE DEPARTMENT OF
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS OF THE DEGREE OF

DOCTOR OF PHILOSOPHY
IN OPERATIONS RESEARCH

September 1985

Signature of Author ____________________
Department of Electrical Engineering and Computer Science
July 26, 1985

Certified by ____________________
Richard C. Larson
Thesis Superivsor

Accepted by ____________________
Arthur C. Smith, Chairman
Departmental Committee on Graduate Students

REAL-TIME DISPATCHING OF DELIVERY VEHICLES

by

ALAN SETH MINKOFF

Submitted to the Department of Electrical Engineering and Computer Science

on July 26, 1985 in partial fulfillment of the requirements

for the Degree of Doctor of Philosophy in Operations Research

ABSTRACT

This dissertation formulates and develops solution procedures for a new problem archetype for a class of physical goods distribution operations. The generic problem is called the *deliverer dispatch problem*, or DDP. The key entities of the DDP are the supplier, who controls a stock of inventory of a certain good and a fleet of delivery vehicles, and a set of geographically distributed customers who consume that good in a stochastic manner. Activity in the DDP consists of the repeated dispatch of vehicles over an infinite time horizon to replenish the customers' depleting inventories. Costs are incurred through the transport of the good by the vehicles, through the holding of the good by the customers, and through customer inventory stockouts. The objective in the DDP is to develop a strategy for dispatching vehicles to minimize the systemwide expected costs incurred per unit time.

One principal assumption of the deliverer dispatch problem is that the dispatcher has access to information about the customers' current inventory levels at all times. This assumption and the presence of demand uncertainty render dispatch plans that are fixed cyclic schedules inferior to plans where each vehicle's assignment is determined just prior to its leaving the supplier. In our formulation of the DDP, the supplier dispatches a vehicle by assigning it a set of delivery instructions called an *itinerary* as soon as it becomes available for duty.

The variant of the deliverer dispatch problem we choose for closer study is a Markov decision problem. However, the size of the Markov decision problem grows rapidly with the number of customers and vehicles in the associated DDP, making exact solution of the DDP impractical for all problems but those of the most trivial size. We develop an heuristic algorithm modeled after an exact procedure, particularly in the way it achieves tradeoffs in short-term versus long-term objectives. For each dispatch, a finite horizon dispatching problem is solved, but the objective function of the problem includes penalty terms that are assessed according to the status of the system at the end of the horizon. These penalty functions are obtained by decomposing by customer the underlying infinite horizon problem. The decomposition is necessarily imperfect, but does seem to provide a good basis for devising penalties for poor dispatching.

Computational tests are performed on both abstract problems and on a case study involving the New York City Department of Sanitation's marine waste transport system. In both situations, the newly-developed algorithm outperforms techniques that fix one or more dispatches in advance of their execution. While this result may not be surprising, the magnitudes of the improvements in dispatching performance are clear indications that substantial savings in systemwide costs are in store by switching from a mode of scheduling dispatches in advance to one of deciding each vehicle's assignment only when it becomes ready. Judging from the results, it does not seem unreasonable to place these savings magnitudes on the order of 5% when the variability of the inventory depletion process is light, and 15% or more when variability is moderate to heavy.

Thesis Supervisor: Professor Richard C. Larson

Title: Professor of Electrical Engineering and Urban Studies;

Codirector of the Operations Research Center

DEDICATION

To Raisin

BIOGRAPHICAL NOTE

ALAN S. MINKOFF received his bachelor's degree in statistics from Princeton University in June of 1980. He graduated with high honors and also received the Phi Beta Kappa award. He started his graduate studies at the Massachusetts Institute of Technology in September of 1980. His teaching experience includes courses in statistics and Markov models. The theoretical work in this dissertation represents the author's principal research activity, stretching over two years. The case study of the second part of the dissertation is drawn from his participation on a consulting project for the New York City Department of Sanitiation performed by the ENFORTH Corporation of Cambridge, Massachusetts.

ACKNOWLEDGMENTS

I wish to express my sincerest gratitude to Professor Richard Larson, my thesis advisor. He planted the seeds of my interest in this dissertation topic when he hired me to work at the ENFORTH Corporation on the Department of Sanitation barge fleet sizing project. As my initial research expeditions floundered or ran aground, he was there to set me on course again. I thank him for his vigilance over the exposition of my ideas; if the reader finds certain passages of the dissertation opaque, I take sole credit.

I also thank the other members of my Thesis Committee for their contributions to this work. Professor Stephen Graves kept me honest as I dabbled in inventory theory and mathematical programming, some of his areas of expertise. Professor Amedeo Odoni offered useful suggestions and a great deal of encouragement. Professor Alvin Drake gave me perspective on the entire doctoral endeavor, and I am grateful to him for that.

If not for the ENFORTH Corporation, I do not know whether I would have found as interesting a dissertation subject as this one. For their support and friendship, I extend my thanks to the people of ENFORTH, in particular to Michael Cahn, Tom Rich, Jackee Kee, Steve McMorrow, and George Fosque. I also wish to thank Paul Gregory of the New York City Department of Sanitation for his initiation and continued support of the barge fleet sizing project.

My officemates, past and present, deserve a note of thanks for lending an ear when my research occasionally became audible, and for putting up with the growing chaos of paper amidst which my desk was located. So here's to Ed Kaplan, Jan Hammond, Paul Chapman, and Paul Thompson.

I would like to thank the rest of the Operations Reseach Center's Apollo community, who kept the machines on their feet in the early days of its installation, and who shared in the wonderful mysteries of graphics programming. Thank you, Jon Eckstein, Paul Schiodtz, and Romano

Richetta. Also, I would be remiss not to voice my appreciation to the Apollo Computer Company, whose equipment made many good computations possible.

For their cheerful support and competent intercession in administrative matters, I owe my gratitude to Jo Ann Bohmfalk and Marcia Ryder at the Operations Research Center.

I thank my parents, Mr. and Mrs. David Minkoff, for giving me the twenty-plus years of momentum that helped carry me through this ordeal. I also thank my in-laws, Rabbi and Mrs. Melvin Glatt, for their concern and their support.

It would take another volume of a size comparable to this dissertation for me to express my full appreciation for my wife Shari, better known to me as Raisin. Not only was she married to a graduate student for over four years, but for the last year, that graduate student was hardly ever around. Even after undergoing major surgery several months before the dissertation was due and returning to motherhood in a greatly weakened condition, she always gave me her full support and love. I do not think I could have made it without her.

My daughter Aviva earned a special assist in the "keeping me cheerful when I was at home" department. I am grateful for her insistent invitations to "pay a dame wid you." May her daddy never have to go through 15-hour work days again.

The early stages of my research, where the foundations of Chapters II and III of this dissertation were developed, were supported in part by a grant from the UPS Foundation through the Center for Transportation Studies at M.I.T. Support for the work of Chapters IV and V, the development and testing of the decomposition heuristic, was partially supplied by NSF Grant Number 8204318-ECS.

TABLE OF CONTENTS

LIST OF EXHIBITS

CHAPTER I

REAL-TIME DISPATCHING

1.0 Introduction

Many physical distribution operations are distinguished by uncertainty in the demand for the delivered goods and a transfer to the supplier of the responsibility for maintaining adequate inventory levels at the customer locations. Yet many popular optimization models employed to provide decision support for these operations tend to ignore these salient characteristics. Instead, they assume delivery requirements to be fixed and imposed from an outside source, and concentrate on optimizing only the transportation component of the system. The potential impacts of demand uncertainty on delivery schedules derived from such models are either to render them less and less effective the further they extend into the future, or to require frequent intervention to correct them. In this age of accelerated computer and information technology development, it seems unreasonable to have to accept these limitations. One can now visualize sophisticated decision support systems for delivery vehicle fleet dispatchers that can suggest effective dispatches in real time on the basis of current customer inventory information. This dissertation aims to demonstrate the feasibility of a physical distribution operation dispatch decision support system with a prescriptive capability operating in an environment of substantial customer demand uncertainty.

We refer to a physical distribution operation dispatch decision support system as a *computer-aided dispatching* (CAD) system, in the terminology of Lee and Larson [17]. Several features of a prototypical CAD system come readily to mind. The system should allow the exploration of alternative dispatches with respect to their effects on the total performance of the operation. It should have access to current and historical information about the distribution environment, and permit the manipulation of the data in various ways. The system should contain user-database and user-model interfaces that do not interfere with the facile utilization of the system. Unquestionably, none of these features requires a significant extension of the state of the art for its successful implementation, as is evidenced by today's operational CAD systems. But there is one last feature which may or may not be handled adequately by modern means, and that is a *prescriptive mode of operation*.

Prescriptive computer-aided dispatching systems currently in use generally derive their dispatch decisions by solving variations of the *vehicle routing problem* (VRP). The VRP is a standard network optimization problem in which a fleet of vehicles is assigned routes and deliveries to satisfy customer delivery requirements at minimum transportation cost (see, for example, Bodin *et al.* [5]). The routes are paths through the network that begin at a depot node and end there after passing through a subset of the customer nodes. The lengths of paths and/or the sum of deliveries along paths may be constrained by vehicle capacities. Customer delivery demands are imposed exogeneously to the problem and must be met completely in any feasible solution to the problem. Hence, the implicit responsibility of the dispatcher is solely to meet the current set of demands.

For the types of physical distribution operations we devote our attention to, we believe prescriptive CAD capability based on VRP-type models to be inadequate. In their stead, we suggest the employment of a kind of problem we designate the *deliverer dispatch problem* (DDP), which to our knowledge constitutes a new problem archetype within the operations research literature. Although the deliverer dispatch problem is discussed in more detail later in this chapter, some points of comparison with the VRP are merited here. While the VRP generates a one-time-period (e.g., one-day) schedule, the DDP provides schedule guidelines for the infinite horizon. The VRP concerns itself solely with transportation objectives; the DDP adds inventory considerations to its list of objectives (to be fair, some recently researched extensions to the VRP support inventory considerations, too). Finally, VRP-guided deliveries correspond exactly to deterministic, exogeneously enforced demands; in the DDP, they are left to the discretion of the decision-maker, and ill-planned deliveries are costly rather than infeasible.

It is rightly conjectured that an optimization problem which extends in so many ways upon the already hard-to-solve vehicle routing problem will prove impenetrable to exact solution in relevant cases. Indeed, one might consider it extremely fortunate to be able to derive any analytic insight into this class of problems. We have sought to fulfill our aims primarily by the *demonstration* of the techniques we have developed in a real world operation, rather than by pure analysis. Therefore, the theoretical development of the solution procedures examined in this dissertation is complemented by a case study, one drawn from the New York City Department of Sanitation's marine waste transport system. We introduce this system later on in this chapter.

1.1 Example

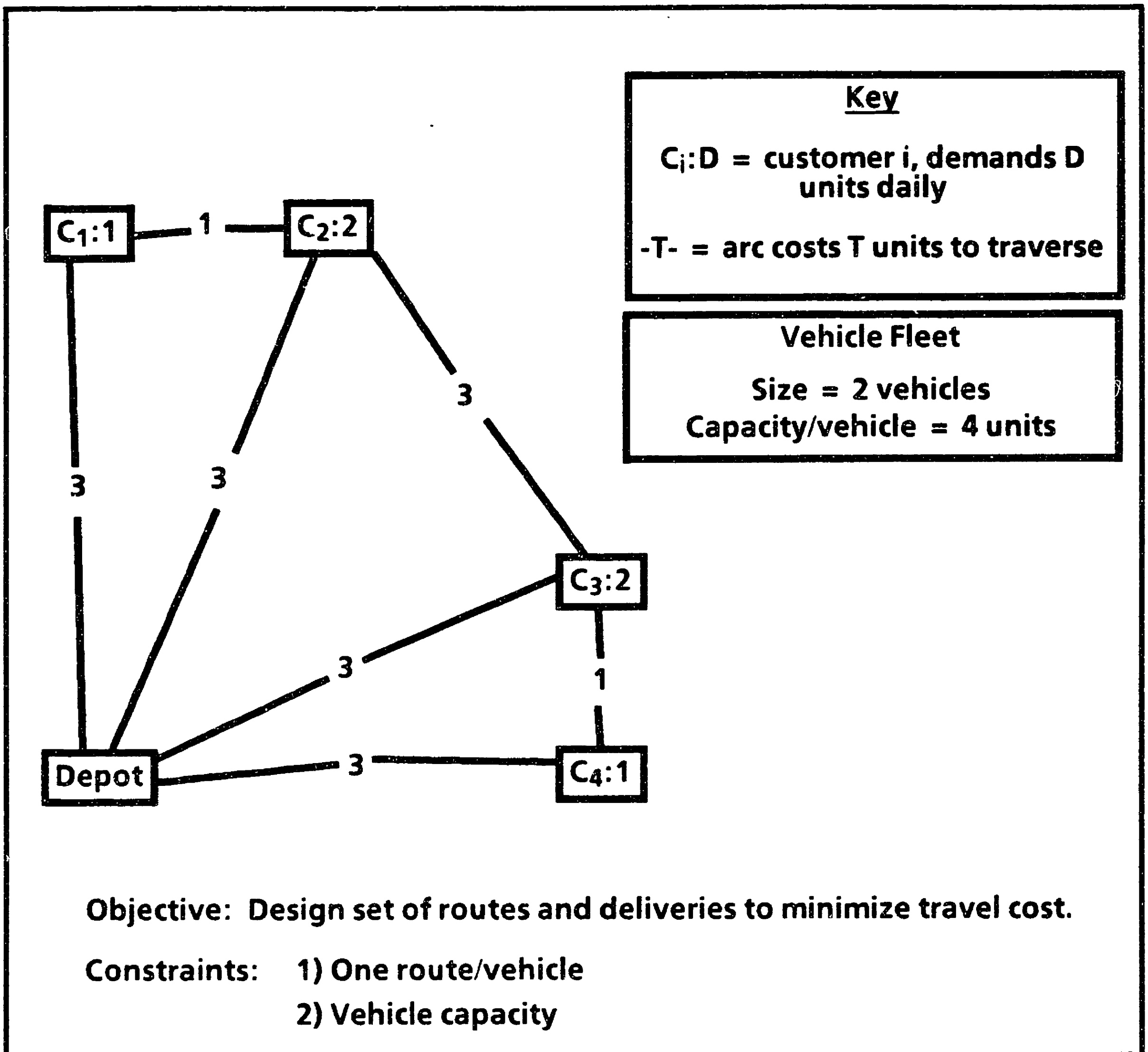
Some of the complexities of vehicle dispatching under demand variability and supplier inventory responsibility can be conveyed via the following small example.

Consider a physical distribution operation in which a fleet of two delivery vehicles, each of which can carry up to four item units, provides regular delivery service to four customers. Such an operation is depicted in Exhibit 1.1. Each customer is represented as a node labeled by an identification number and a daily demand (assume for now that the demand is deterministic). Each arc is labeled with its traversal cost. By formulating the appropriate vehicle routing problem and solving it, we obtain a set of route and delivery assignments for each vehicle. These assignments can also be extended to represent a delivery schedule yielding minimum transportation cost among those whose assignments do not vary by day of a certain (perhaps infinite) planning horizon. One solution, at an objective cost of 14 units, is:

Vehicle 1: deliver one unit to customer 1 and two units to customer 2.

Vehicle 2: deliver two units to customer 3 and one unit to customer 4.

Exhibit 1.1: Sample Physical Distribution Operation



This example is based on one discussed in [2].

The total daily cost of this schedule is 14 cost units.

Now suppose, in the same example, that deliveries over and above daily demand may be put into inventory and be used on subsequent days. Then consider this two-day schedule:

On odd-numbered days:

Vehicle 1: deliver two units each to customers 1 and 2.

Vehicle 2: deliver two units each to customers 3 and 4.

On even-numbered days:

Vehicle 1: deliver two units each to customers 2 and 3.

Vehicle 2: remain idle.

This schedule incurs 23 cost units every two days while completely satisfying customer demands. The first schedule costs 28 units over the same timespan. The basic VRP is not geared to detect such economies. More complex problem formulations must be and have been devised for optimal or near-optimal scheduling when the planning horizon extends beyond one day (see [5]).

The multi-day schedule given above required that two of the customers hold an item in inventory every other night. A more general model of a physical distribution operation such as this one would attach a cost to the activity of holding items in inventory. If this were the case in our example, then the second schedule would be rejected in favor of the first one if the systemwide holding cost per item were greater than 2.5 cost units. Additionally, suppose costs or frequency constraints are ascribed to customer stockouts. Then, for instance, it may be more economical to forego any replenishment of customers 1 and 4 under some stockout objective structures. Useful models for delivery scheduling support in such environments need to accomodate the potential tradeoffs among transportation, inventory holding and inventory stockout objectives. Recent research efforts in this area include the work of Federgruen and Zipkin [10], Fisher, Jaikumar, and Bell [11], and Dror [8].

Up until this point, we have assumed that the customer demands presented in the example are deterministic quantities. Now suppose they are merely averages of probabilistically varying demands. The presence of uncertainty in customer demand patterns transforms the vehicle scheduling task in several critical ways. It is no longer possible to time deliveries so that they arrive just as customers run out of stock, because the time of stockout is uncertain. To guarantee that each customer is well-stocked at all times may require intolerably high inventory holding levels, so that the transportation-holding-stockout tradeoffs are felt more acutely here. In multi-day scheduling, the quality of dispatch performance is likely to decrease for dispatches made later in the planning period, because the true inventory status of the system becomes more and more likely to depart from its anticipated status (some customers may stock out more quickly than expected, while others may encounter lighter demand than usual). Indeed, uncertainty may even be associated with the assumed current values of inventory levels at the time a schedule is designed.

For illustrative purposes, suppose it is true for each customer that, given its average daily demand of D units, either $D-1$ units or $D+1$ units are demanded on any particular day. Each outcome has a probability of occurrence of one-half, and these probabilities do not depend on previous days' or other customers' demands. Suppose, also, that deliveries occur at the beginning of the day, and demands occur at the end of the day, with the implication that each day's demands are unknown at the time of dispatch. Demands experienced by a stocked-out customer are lost rather than backlogged, by assumption here.

Exhibit 1.2 displays a number of scenarios marked by different demand quantities over a two-day planning horizon and/or different initial inventories. In each scenario, the two-day delivery schedule given above is adhered to. The first two scenarios begin with each customer holding no inventory, as in the deterministic case. The fortunate circumstances of scenario 1 allow the system to return to its original state with no demand lost. The operation does not go unscathed in scenario 2, though, where aggregate demand is only two units higher. One may quickly surmise that it is advantageous to hold some inventory at all times to act as a buffer in periods of high demand. Scenarios 3 and 4 feature such safety stocks. No lost demand is suffered in scenario 3, even though demand there is much like the disastrous scenario 2. In scenario 4, demand is lighter than in scenario 3, yet one unit of demand is lost. The point to be learned is that the second day's

Exhibit 1.2: Initial Inventory/Demand Scenarios

S/C	Init Inv	Day 1			Day 2		
		Delivery	Demand	Inventory	Delivery	Demand	Inventory
1/1	0	+2	-0	=2	+0	-2	=0
2	0	+2	-1	=1	+2	-3	=0
3	0	+2	-1	=1	+2	-3	=0
4	0	+2	-2	=0	+0	-0	=0
2/1	0	+2	-0	=2	+0	-0	=2
2	0	+2	-3	=0*	+2	-3	=0*
3	0	+2	-1	=1	+2	-3	=0
4	0	+2	-2	=0	+0	-2	=0**
3/1	1	+2	-0	=3	+0	-2	=1
2	2	+2	-3	=1	+2	-3	=0
3	2	+2	-1	=3	+2	-3	=2
4	1	+2	-2	=1	+0	-0	=1
4/1	1	+2	-0	=3	+0	-2	=1
2	2	+2	-1	=3	+2	-1	=4
3	2	+2	-3	=1	+2	-1	=2
4	1	+2	-2	=1	+0	-2	=0*

Each * represents one unit of lost demand.
S/C = scenario / customer

schedule, fixed in advance, could have been easily improved if rescheduling at the beginning of the second day had been allowed. Taking this further, a better way to go about scheduling might be to forestall the decision about each day's routes to be run to the beginning of that day, when the latest inventory information is available. It is the formalization of this type of scheduling that we pursue in this dissertation.

Before we move on, it seems only fair to say that the degree of variability in the foregoing scenarios may have exaggerated the levels to be found in most applications. Nevertheless, we feel that this example conveys our main points effectively. We also acknowledge that there is some cost to be associated with implementing a computer-aided dispatching system more responsive to system characteristic changes. But with the costs of the hardware components of CAD systems dropping, and the penalties of poor dispatching tending to increase, we believe that the time is ripe to study these sorts of dispatching problems and to develop methods for generating effective dispatches.

1.2 Characterization of the Deliverer Dispatch Problem

The deliverer dispatch problem is intended to be a representation of a physical distribution system more realistic than the vehicle routing problem in cases where the deliverer bears some responsibility for customer inventories. A more rigorous treatment of the DDP is performed in Chapter III. This section offers a brief characterization of the deliverer dispatch problem.

A *dispatch decision*, the focus of our study, is a set of instructions listing when and to where a vehicle is to be dispatched, and how much is to be delivered at each stop on the dispatch assignment. In our deliverer dispatch model, dispatch decisions are made at moments in time, over a time horizon that extends infinitely far into the future. We call a moment when a dispatch decision is to be made a *decision point*. Relative to the decision point that establishes it, a dispatch may occur immediately or at some future time. A point in time at which a dispatch occurs is a *dispatch point*. Depending on the decision-making procedure (one might say, on how *adaptive* it is), there may or may not be a one-to-one correspondence between decision points and dispatch points. Note, also, that *not* to dispatch a vehicle is a valid dispatch decision. The result of the dispatch of a vehicle is that it undergoes a period of unavailability for dispatch, during which time it is carrying out its dispatch assignment. As it visits each customer on its list, it delivers the assigned amount. From each customer's perspective, inventory is depleting over time. The depletion process results from stochastic, exogeneous demands for goods from the customer. A customer's inventory level may drop to zero; in this case, the customer is stocked out, and further demands are either backordered or lost to the system. When a vehicle visits a customer, that customer's inventory level jumps up by the quantity of the delivery.

The objective in the DDP is one of cost-minimization. Costs arise in both the transportation and the inventory components of the system. Each *itinerary*, or list of customers and their deliveries, incurs a certain transportation cost upon its execution. Inventory costs are assessed on the holding of units in inventory and on demands which are backordered or lost. The time horizon against which the quality of decisions is judged is an infinite one, so minimization of total costs will not serve as a useful principle for the evaluation of competing decision-making procedures, or decision rules. Instead, we characterize the objective as one of minimizing cost per unit time over the long run. Because the systems we deal with are stochastic, the objective cost will be expressed as an expectation.

Given a more mathematically explicit problem statement like the one put forward in Chapter III, the DDP can be readily formulated as a Markov decision problem. Deriving a dispatch decision rule from such a formulation is an entirely different matter. Only the smallest of DDPs may be solved exactly using Markov decision theory, since the size of the problem to be solved grows exponentially in the number of customers. To make practical use of the DDP in computer-aided dispatching (CAD) systems, the development of good approximate techniques is required. We undertake this task in Chapters IV and V.

1.3 Characterization of a Marine Waste Transport System

To help bridge the gap between theoretical formulation and practical implementation of adaptive dispatching methodology, we intend to demonstrate the development of a computer-aided dispatching system for a complex operation, namely marine waste transport in New York City. This demonstration comprises the second part of the dissertation. This section provides a quick synopsis of the marine waste transport operation and its relationship to the deliverer dispatch problem. Chapter VI is devoted to a fuller explication of this system and our model thereof.

Each day, New York City generates tens of thousands of tons of refuse. It is the responsibility of the New York City Department of Sanitation (DOS for short) to put that refuse where it is least offensive to the most people. A great proportion of it finds its final resting place in Fresh Kills Landfill (FKL), a sizable chunk of land on Staten Island. The landfill is located a comfortable distance away from most citizens of New York City. This degree of separation invites the question, How does the refuse get from the trash can to the landfill? The most economical mode of transport is the marine operations system, which works as follows:

Sanitation trucks carry their loads several times a day to marine transfer stations (MTSs), two-story structures that jut out into the local waterway. They drive onto the second story of the MTS, back up to one of several apertures in the floor of the building, and dump their load through the aperture onto a large barge in the water beneath (the first story). Tugboats also visit the MTS from time to time, dropping off empty barges and picking up full ones. The barges are towed, typically in a train of three or four barges, by tugs to FKL, where they are unloaded by large cranes or "diggers." The empty barges are towed back to the MTSs by the tugs to be filled again.

Human decision-making faculties impinge on the DOS marine waste transport operation most critically in the activity of directing where tugs are to pick up and drop off barges. The dispatch decision-making function as it is currently practiced consists of two phases. First, each morning, DOS Operations personnel decide how many empty barges must be present at each MTS by the *next* morning. These requirements are then radioed to the Marine Dispatchers at FKL. In the second phase, Marine Dispatchers draw on their experience to direct tugs and barges throughout the day to meet the next day's requirements. The second phase is by no means straightforward to execute because the marine system is subject to many idiosyncrasies. The number of barges available for delivery varies constantly over time. Tug travel is strongly affected by tides, which operate on a lunar cycle, and by weather conditions. Tug service is provided by an independent firm, so that the degree of control dispatchers may exert on route selection is limited. Key components of the system may go out of service for varying lengths of time; these service outages often affect other system components as well. Since refuse inflow to MTSs cannot be predicted precisely, barge requirements may change during the day, sometimes necessitating changes in the dispatching schedule. Other system quirks also must be borne in mind by the dispatcher when making decisions.

That the task of routing tugs through the marine system can be modeled as a (suitably detailed) deliverer dispatch problem may not be evident at first glance. Most difficulties are resolved, however, when it is realized that "inventory" in this system refers to empty space on barges. An MTS suffering a "stockout" has no empty barge capacity, i.e. nowhere to put the incoming refuse. In practice, what happens in this situation is that sanitation trucks either queue up at the MTS and wait for an empty barge to be moved into loading position, or are rerouted to a more distant facility. Both solutions cost the city in terms of additional salary and fuel expenditures, hence stockouts are costly and are to be avoided. The cost of holding empty barges in inventory at MTSs corresponds, in the short term, to the risk of diminished barge availability elsewhere in the system, and in the long term, to increased capital expense. Refuse inflow rates, or the rates at which inventories deplete, are always subject to some degree of random variation. Given these observations, one should feel much more comfortable modeling the system with a DDP than with the vehicle routing problem. But modeling is one thing, and deriving good policies from a model is quite another. Chapters VII and VIII detail our successes and failures at the latter endeavor.

1.4 Preview of the Thesis

Chapter I of this thesis has introduced the topic of our research, real-time dispatching of delivery vehicles.

Hereafter, the dissertation is divided into two parts. The first part lays a theoretical foundation for a prescriptive component of a computer-aided dispatching system. The second part demonstrates the application of the material contained in the first part by means of a case study.

Chapter II provides a perspective on the broader class of vehicle dispatching problems, and suggests a framework for the characterization of such problems. In doing so, it reviews the current literature on problems related to the one we study in this thesis.

Chapter III describes in detail, both verbal and mathematical, the deliverer dispatch problem. The general problem and some interesting special cases are treated. This chapter also establishes a basis for treating the DDP as a Markov decision problem.

After demonstrating that exact solution via Markov decision theory is infeasible, Chapter IV explores a heuristic solution approach to the DDP. The approach is to solve a finite horizon scheduling problem for each state of the underlying system *as it arises*, with a penalty function incorporated in the objective function. The dispatches derived from solving the finite horizon problems are influenced to serve long-range objectives through the imposition of strong penalties when the current decision will tend to leave the system in undesirable states in the future. The resulting heuristic is shown to parallel in several ways the policy iteration algorithm for solving Markov decision problems.

To illustrate the working of the somewhat complex heuristic proposed in Chapter IV, the heuristic is applied to a small DDP in Chapter V. This chapter continues by documenting computational experiments involving various instances of the DDP and the heuristic.

Chapter VI begins the case study. It describes the marine waste transport system (MWTS) and presents a mathematical model of it. The subject of dispatching within this system is postponed for one chapter.

It is noted in Chapter VII that many aspects of the operation described in the case study do not conform to the deliverer dispatch model. Chapter VII details what must be done to adapt the previously developed DDP solution algorithms for application in the MWTS model. It also describes a dispatching algorithm that emulates to some degree what is presently done to route tugs and barges in the MWTS.

Chapter VIII provides extensive numerical results relating the performance of the DDP algorithms in their implementation in the MWTS model.

Chapter IX summarizes both the answered and the unanswered questions raised during the course of the dissertation.

PART I

THEORETICAL FOUNDATIONS:
THE DELIVERER DISPATCH PROBLEM

CHAPTER II

A PERSPECTIVE ON VEHICLE DISPATCHING PROBLEMS

2.0 Introduction

A multitude of common operations in both the private and the public sectors of daily commerce can be characterized as vehicle dispatching operations. Attempts to model and/or improve these operations have resulted in the establishment of a wide body of operations research literature devoted to the formulation and solution of vehicle dispatching problems of various types. However, little has been done to unify these disparate theoretical works within a cohesive framework. This chapter is intended to provide a global perspective on vehicle dispatching problems and to lay the foundation of a comprehensive system for their portrayal.

The general structure of the class of vehicle dispatching operations that we address consists of the dispatching of vehicles over time from some facility to perform a number of tasks in a geographic region before returning to the same (or possibly a different) dispatching facility. Many real world activities are described by this structure. One of the more common examples is the distribution of goods from a warehouse to local retailers. The dispatched vehicle may render a service, rather than deliver a good, so that police patrol car and fire engine dispatching fall under this domain, too. And vehicles may pick up, rather than deliver, a given quantity; thus, local refuse collection and airport limousine service operations may be modeled as vehicle dispatching operations. The key decisions that must be made in all these operations are when to dispatch a vehicle, and what the vehicle is to do on its dispatch assignment.

The operations mentioned above, as well as many others, have been the subject of quantitative modeling efforts, and have produced interesting theoretical problems. The variety of approaches taken toward modeling vehicle dispatching operations tends to obscure the common bonds shared by the activities and the models. We consider below three prototypical problems that abstract vehicle dispatching operations (we refer to such problems as vehicle dispatching problems, or VDPs). These problems also serve to focus subsequent discussion of VDP characteristics in Section 2.4.

The vehicle routing problem (as described in Bodin *et al.* [5]) models a one-shot assignment of customers to vehicle delivery routes. Each customer is represented by its delivery requirement or demand for the planning period (typically one day) and by its location in the transportation network. Each vehicle in the fleet has a certain carrying capacity for the delivered good, and may be constrained in total travel time or distance. The objective in the vehicle routing problem (VRP) is to design a set of vehicle routes and deliveries that incurs minimum transportation costs, while assuring that each customer's demand is met and that all vehicle constraints are respected.

We use the term "emergency dispatch problem" (EDP) to label models of emergency service systems such as police, fire, and ambulance systems. In these systems, vehicles are dispatched in response to calls for service. Most problems abstracted from these systems model call arrival rates and service administration durations as stochastic quantities. Planning horizons are usually taken as infinite in these problems, and decisions are usually evaluated for their performance in the long run or steady-state.

The third vehicle dispatching problem we consider here is the deliverer dispatch problem which was introduced in Chapter I. To review, the DDP models certain kinds of physical distribution operations in a more detailed and comprehensive way. The vehicle fleet-customer structure resembles that of the VRP, except that customer demands are replaced by stochastically depleting inventories. Further, the planning horizon in the DDP extends arbitrarily far into the future, and dispatch decisions are made generally when vehicles are available for dispatch. Customer inventory concerns become intertwined with transportation concerns in the problem objective. Hence, good dispatch decisions must seek a balance between avoidance of inventory stockouts and overburdening the delivery and inventory holding systems.

As diverse as these problems seem, it is nonetheless possible to detect common threads among them. This chapter has been organized with the intention of developing such a structure. Section 2.1 reviews previous VDP characterization efforts. Section 2.2 sets up an appropriate paradigm for understanding the basic processes active in any VDP. From the standpoint of this paradigm, a set of core components of VDPs is extracted in Section 2.3. Section 2.4 discusses each of these components in detail, with discussion centering on the three VDP prototypes described in the introduction to this chapter. Several other VDPs are examined in Section 2.5 to reinforce certain VDP component concepts. Section 2.6 presents conclusions and points to further research topics in this area.

2.1 Background

The small number of characterization efforts that have preceded this work have focused on variations of the vehicle routing problem described in Section 2.0. Although the resultant VDP subclasses treated encompass many of those vehicle dispatching problems that have been studied in the literature, several important classes lie outside their domain. Furthermore, the characterization schemes themselves are incomplete with respect to our conceptions of these problems. We refer to two studies, one by Bodin and Golden [4] and the other by Bodin et al. [5]. The two characterization schemes do not differ greatly, in that the latter is based explicitly on the former. We outline their approaches before proceeding with our proposal.

The referenced characterization schemes initially group the vehicle dispatching problems of their selected domain into three categories:

(1) *Routing problems* are problems in which vehicles are assigned to visit points in the geographic region to pick up or drop off goods, and are unconstrained by time-of-day and other point-specific factors.

(2) *Scheduling problems* require vehicles to visit assigned points at certain prescribed moments in time.

(3) *Combined routing and scheduling problems* cover all types of vehicle routing and scheduling problems that do not fit into the other two categories, including problems that specify "time windows" on visiting times, and those that enforce a full or partial ordering on the points assigned in vehicle routes.

The cited works extend their characterization schemes by listing a series of characteristics common to all vehicle routing and scheduling problems. These characteristics include the size and type of the vehicle fleet, demand characteristics, network configuration, and system objectives. For each characteristic, several potential values are given. The listings of characteristic values are clearly not meant to be exhaustive of all possible values. This work seems intended as a loose framework for identifying members of the class of vehicle routing problems.

2.2 A Dynamic Decision Paradigm

The framework that we elect for the characterization of vehicle dispatching problems relates to the fundamental evolutionary process of any vehicle dispatching operation. To reiterate, a vehicle dispatching operation involves the periodic dispatch of vehicles to perform tasks in the geographic region of responsibility. The action of the dispatched vehicles, together with external forces, produce changes, or *evolution*, in the environment in which the vehicles operate. The vehicle actions are dictated in turn by *decisions* made somewhere within the system. Alternative decisions are appraised according to the anticipated level of *performance* of the system, where performance is the interpretation of evolution in light of system objectives. The interplay of these concepts suggests that a dynamic decision problem paradigm would serve as a useful foundation for a cohesive VDP characterization framework.

The formal dynamic decision problem paradigm we utilize is as follows: Each decision during the course of evolution of a system takes into account potential future evolutionary paths. For a given future evolutionary path, a sequence of points in time at which decisions can be made is indexed by the set of nonnegative integers k, where $k=0$ represents the current decision point. At the k-th decision point in the path, the system occupies state x_k, the decision u_k is made, and uncontrollable factors w_k act upon the system from then until decision point $k+1$. The decision u_k depends solely on information contained in the system state x_k. The state of the system at decision point $k+1$ derives from the previous state, decision, and uncontrollable factor activity. System performance during that decision point interval is a product of the same three system elements. The dynamic decision problem may then be expressed symbolically as

$$min_{\{u_k\}} E_{\{w_k\}} g\left(x_0, u_0, w_0, x_1, u_1, w_1, x_2, u_2, w_2, \ldots \right)$$

subject to

$$x_{k+1} = f_k(x_k, u_k, w_k)$$

and

$$u_k = U_k(x_k),$$

where g, f_k, and U_k are the performance, evolution, and decision functions, respectively. In order to assure that the value of the objective function is finite, g must incorporate either a discount factor or a performance averaging function. Often the function g may be reexpressed as

$$\sum_{k=0}^{\infty} g_k\left(x_k, u_k, w_k\right).$$

We withhold further discussion of the elements of this formulation until Section 2.4.

2.3 Core Components of Vehicle Dispatching Problems

We identify four core components of the general vehicle dispatching problem. By ascribing specific attributes to each of these components, we can generate any instance of the vehicle dispatching problem. The core components are:

(1) the set of events that precipitate decision points;

(2) the structure of the system;

(3) the way that the system evolves;

(4) the system's performance function.

2.4 Discussion of Core Components

In this section, we discuss each of the four core components of vehicle dispatching problems. The discussion is intended to illustrate how each component contributes to the formulation of particular instances of vehicle dispatching problems, and to treat the key issues associated with each component. We make extensive use of the three VDP prototypes described in Section 2.0 during the course of our discussion.

2.4.1 Decision Point-Precipitating Events

A variety of events can precipitate vehicle dispatch decision points, all depending on the nature of the problem. Some problems contain artificial events that occur solely to initiate dispatch decisions, while others reach decision points as a result of normal operational events. The interval between decision points may be of fixed or variable duration. We assume that, in any problem we shall consider, decisions are made at at least one point in time, otherwise the problem is of no interest from a decision-making perspective.

In many formulations of vehicle dispatching problems, there is only one decision point allotted; its precipitating event may be understood as the start of the time period being modeled. One VDP that has this characteristic is the vehicle routing problem. Basically, the VRP represents a single-period distribution operation. Dispatch decisions bear no consequences on the evolution of the system to the next decision point, because there is no "next" decision point. Further examples of single-decision point problems include vehicle scheduling problems (described in [5]) and the designing of fixed, repeating schedules for longer time intervals (e.g., a metropolitan bus schedule valid for three months).

With respect to the number of decision points in a typical evolutionary path, the emergency dispatch problem lies at the other end of the spectrum from the VRP. Two types of events precipitate dispatch decisions in the EDP. The first precipitating event type is the arrival of a call for service; the decision that arises is which vehicle (if any) to dispatch to the call. Because calls for service may be placed into a queue of calls awaiting dispatch (for instance, when all vehicles are busy), the return of a vehicle to the "available" status may also initiate a dispatch decision. Hence, a service completion is a second decision-precipitating event in some situations. Because call arrivals and service completions occur at random instants in time, the spacing between dispatch decision points fluctuates stochastically. Models of emergency dispatch systems, such as the HYPERCUBE queueing model presented in [14], explicitly account for this randomness in the problem formulation. Section 2.4.2 discusses the nature of decisions in these systems.

The deliverer dispatch problem also may associate dispatch decision points with the return to the dispatching location of a vehicle from its previous assignment. Prompting the need for models with this characteristic are operations in which vehicles are heavily utilized and completion times vary considerably from assignment to assignment. If a model of this type of system only allowed all vehicles to be dispatched simultaneously (as in the VRP), a considerable amount of delivery resources may be wasted as vehicles completing their assignments prior to the next decision point sit idle. Another characteristic of all-at-once dispatching further exacerbates the resource wastage problem in the DDP. Since customer inventories deplete stochastically, an all-at-once dispatching system cannot, without the aid of intervention, react timely to extraordinary inventory depletion patterns. For the types of operations that the DDP is designed to emulate, dispatch-when-ready schemes fit the circumstances better.

One system which could be modeled well as a deliverer dispatch problem is the New York City Department of Sanitation's marine waste transport system, as is indicated in Chapter I. The vehicles that are dispatched are tugboats, and their task is to transport empty refuse barges to special sites called marine transfer stations, where sanitation trucks dump their loads into the barges. Because the rates of refuse arrival to the marine transfer stations are variable, and because the lack of barge capacity at stations strains the system, dispatchers must be in a position to respond to critical barge needs in the system. This requires flexibility in dispatching and causes to be untenable the fixing of delivery schedules too far in advance, as all-at-once dispatching requires. Thus, only when a tug is ready to depart from the dispatch facility with several barges in tow is the tug's assignment determined. Of course, it is necessary to anticipate future dispatch decisions for other tugs, because performance in the domain of the marine transfer stations is dependent on the times of barge deliveries.

From this discussion, we have learned that it is important to define the set of events (start of time period, call for service, return of vehicle) that may precipitate the dispatch decision points in the context of a problem. It is also possible, for some problems, that the occurrence of a precipitating event does not guarantee that a dispatch decision besides "no dispatch" will be made, or that certain combinations of events are required. For instance, in a police dispatch system, the return of a patrol car when no calls for service are in queue requires no dispatch decision, yet the return of a car when there are calls in queue normally signals a new dispatch to one of the waiting calls. Similarly, the return of a tug in the marine waste transport system might not result in a new dispatch, if barge demand throughout the system is relatively light, but in the creation of a different precipitating event--a dispatch delay period--that initiates a dispatch decision point some time in the future. The importance, to the definition of a vehicle dispatch problem, of decision-precipitating events cannot be overemphasized, because the length of the interim period between decisions, and the capacity of the system to evolve during that period, bear on the determination of the quality of alternative dispatch decisions.

2.4.2 Structure of the System

We employ the term "system structure" to refer to those elements of a vehicle dispatching problem that contribute to problem characterization by:

(1) defining the nature of a "solution" to the problem;

(2) characterizing feasible solutions; or

(3) listing the types of information that are needed to determine feasibility.

One might argue that the length of description required to define completely the structure of a vehicle dispatching problem is so great that the goal of problem characterization is defeated. We would respond that the vehicle dispatching literature is already replete with generalized problem prototypes, so that characterization along this component may normally be performed by referencing the appropriate prototype. Indeed, many of these prototypes may be generated with more refined characterization schemes such as the one by Bodin *et al.* [5] that was mentioned earlier. The rest of this section elaborates upon the concept of system structure.

2.4.2.1 Nature of a Solution

If a vehicle dispatching problem models the periodic dispatch of vehicles to execute tasks over space and time, then the solution of such a problem ought to supply the guidelines for the timing of the dispatches of and the activities to be performed by the vehicles. Our dynamic decision problem paradigm further stipulates that a unique decision is associated with each possible state of the system at any decision point. We find that all vehicle dispatching problems share this basic solution structure, but that different classes of these problems exhibit markedly different solution characteristics. We explore particular solution structures and the state-decision correspondence in our discussion below of our three VDP prototypes.

There are two principal components to the solution of a (deterministic) vehicle routing problem: the route that each vehicle is to travel, and the activity that the vehicle must perform at each stop on the route. After the problem is solved, it can be determined exactly what transpires in the model during the planning period. The state-decision correspondence is trivial in the VRP case, since there are no future decision points, so a decision need only be found for one state of the system (the original state).

In the emergency dispatch problem, actual service requirements are known a priori only in a probabilistic way, yet dispatching rules must give precise information on when and to where vehicles are to be dispatched. Hence, the solution to the EDP provides dispatching rules that are conditional on the state of the system at a decision point. The rules associating state and decision might take the form of a table of order of preference among vehicles for each customer or customer region, a nearest available vehicle function, or more generally, a reference to another problem whose solution supplies the dispatch decision.

The concept of a solution referencing other problems and their solutions may be quite useful in characterizing the solutions of more complex vehicle dispatching problems such as the deliverer dispatch problem. As in the EDP, the dispatch decision in the DDP is a nontrivial function of the state of the system at the time of dispatch. However, it is often true that the combinatorially large or infinite number of states makes tabular storage of dispatch decisions either infeasible or impossible, even in principle. The logical alternative here would be for the "solution" to simply point to a smaller-scale vehicle dispatching (or other mathematical) problem; the exact instance of the problem to be solved would derive from the current state of the system, and the solution of the smaller problem would yield the dispatch decision. Where both options are feasible, the tradeoff is one of reduced storage versus additional computation.

The form of a solution for the DDP with the system in a given state is likely to be more complex than that for the VRP. Although perhaps only one vehicle route is solved for at a decision point, the specification of an assignment might consist in part of a number of conditional actions. To illustrate, an assignment may be a list of locations to visit, with the size of each delivery to be calculated upon arrival at each location as the local fill-up-to point minus the current inventory level. The inventory level is likely to change between the time of dispatch and the time of arrival. Not only are the delivery sizes uncertain in the original assignment, but the original route itself may become ineffectual if the vehicle's supply of the delivered good is exhausted while further stops on the route remain. An additional condition may be supplied for such occurrences, stipulating that the vehicle travel to a supply point for refill and then finish its original assignment.

2.4.2.2 Feasible Solutions

The structure of a system must allow one to discriminate between feasible and infeasible solutions. Therefore, it must delineate the operational constraints that all feasible decisions must observe. Since the specifics of these constraints may change as the system evolves, the system structure's feasibility assessment aspect should not be considered simply a list of constraints. Rather, it is preferable to construe the discriminating function as being composed of two complementary entities: a description of the general format of the constraints, and a list of information requirements for establishing the actual set of operational constraints at any dispatch decision point. This section further explores how system structures characterize feasible solutions.

The feasibility discrimination mechanism of the vehicle routing problem is embedded in the constraints identified in its verbal or mathematical programming formulation. For example, assuming that all dispatches take place at the beginning of the planning period (day), then the system structure includes a constraint of the format, "the number of dispatches may not exceed the size of the vehicle fleet." Feasible routes must respect the network structure of the problem. Length of route constraints may also be active. Deliveries must satisfy demand (perhaps exactly), yet not exceed vehicle capacities. Further possible details include compatibility of vehicle and delivery locations and, to move into the realm of vehicle scheduling problems, acceptable time of visit intervals.

Feasible solutions may be characterized more succinctly for the emergency dispatch problem. Because vehicles are generally dispatched to perform a single task under a condition of emergency, the system structure is unlikely to carry constraints limiting lengths of routes or lengths of service times. Indeed, route selection from origin to destination, for a given origin-destination pair, is usually not accounted for in EDPs. (Travel time *is* a consideration in evaluating the quality of a dispatch decision; see Section 2.4.4.) The main constraint on feasibility in the EDP is of the form, "a busy vehicle may not be dispatched." In problems allowing preemption of service, even this constraint may not be present; at the very least, several classes of the "busy" state must be defined, and a table must be established showing which busy states may be interrupted for dispatch to which types of service calls.

The deliverer dispatch problem shares many of the characteristics of the vehicle routing problem and the emergency dispatch problem. Consequently, solutions for the DDP combine characteristics of feasibility outlined for both problems. The environment in which the vehicles operate is usually similar to those modelled in VRPs. Hence, system structure will define feasible routes, feasible delivery quantities, and vehicle/location compatibility. In regard to actual dispatching, the DDP more clearly resembles the EDP, for dispatch is generally addressed to one of the (non-busy) vehicles located at the dispatching facility. Some problem variations may allow the adjustment of assignments in progress as a result of new information about the system. One important example of this type of operational modification was presented in the last section, the case where the exact size of a delivery is resolved only when the vehicle arrives at the place of delivery. In this case, the vehicle may run out of supply before the assignment has been completed, thus necessitating a change in that assignment. This example also illustrates that even a feasible solution may be impossible to fully carry out within the context of the model. It is therefore essential to articulate fully the qualifications of a feasible solution in problems such as the DDP, as it is in any vehicle dispatching problem.

2.4.2.3 Information

Given a description of the format of the constraints on feasible dispatches in a vehicle dispatching problem, the constraints active at any decision point in a system's evolution are established by supplying the constraint formats with system state information. In characterizing a VDP's system structure, then, it is crucial to enumerate the types of information required to assess feasibility. Since feasibility discrimination is an integral component of the decision function (as presented in the dynamic decision paradigm), and since the decision function relies solely on information contained in the current system state, the discussion of this section also sheds light on the meaning of "system state."

The information requirements for the assessment of feasibility in the vehicle routing problem can be read quite straightforwardly from its constraint formats. For instance, the constraint that the number of dispatches may not exceed the size of the vehicle fleet cannot be tested until the size of the vehicle fleet is known; hence, "size of vehicle fleet" constitutes one item of information required to discriminate feasible solutions from nonfeasible ones. This item may be considered to belong to a class of vehicle operating characteristics information. This class will also contain information on travel range and carrying capacity limits. Similarly, the class of customer information will contain information on delivery demands. A third class of information concerns the geographic network in which the customers are located and the vehicles travel. Some types of information may belong to more than one class, such as lists of vehicles which may or may not make deliveries to a certain location.

Dispatching within the emergency dispatch problem operates under fewer constraints, so less information must be supplied than in the VRP in order to determine which dispatches are feasible. In the basic EDP (one free server dispatched to each call for service), the only constraint is that a busy server must not be dispatched. A list of which servers are busy at a decision point is essentially the only item of information that is needed; this list also acts to define the state of the EDP system. Most variations on this basic problem do not require much more information than this. For instance, in models where vehicles busy on low-priority tasks may be preempted, the priorities of both the incoming call and the activities of the busy vehicles accompany the busy vehicle list.

Just as deliverer dispatch problem constraints resemble characteristics of constraints in the VRP and the EDP, so, too, will the former's feasibility information requirements resemble those in the latter two. Again, as in the VRP, information classes arise covering the domains of the vehicle fleet, the customers, and the transportation network. Here it must be noted, though, that the DDP explicitly models a multi-decision point, dynamic dispatching environment. Therefore, it must be expected that many information elements will vary in value over time, unlike the static information elements of the VRP. For example, the number of vehicles available for dispatch will vary according to the number not yet returned from their previous assignment; in this way, the DDP looks more like the EDP. Other time-varying information elements, such as customer inventory information, may or may not play a role in assessing feasibility, although this information would certainly be found in any plausible representation of the system state in the DDP.

In general, it should be recognized that system state information possesses different behavioral traits which affect the way that information can be utilized in formulating and solving a vehicle dispatching problem. The enumeration and analysis of these traits remains a topic for further research.

2.4.3 System Evolution

System evolution refers to the process of change of the state of the system over time. Besides the state, the dispatch decision and the action of uncontrollable or external factors are identified as evolution influences in the dynamic decision problem paradigm. In many vehicle dispatching problems, the behavior of the uncontrollable factors is known a priori in only a stochastic sense. Yet the prediction of future evolutionary paths guides the selection of the current dispatch decision in most instances. The proper characterization of a VDP must therefore provide a thorough treatment of system evolution, with emphasis on the behavior of external factors.

Because evolution may depend on the system state, it is necessary for us to widen our conception of the system state and state information to include not just information for determining decision feasibility, but also information that is required to explain how the system evolves. We must also examine how decisions affect evolution, and how external factors influence it, if at all. The three VDP prototypes that we have heretofore studied serve to focus this discussion as well.

The vehicle routing problem does not model a dynamic system at all. At its sole decision point, the decision-maker is not concerned with future evolution, but only the performance of the system until the end of the planning period. Also, external factors are absent in the deterministic case, and only affect performance in stochastic variations of the VRP (where customer demands are probabilistically distributed). System performance is treated as a separate problem component in the next section, so there is no point of further discussion here.

The emergency dispatch problem, on the other hand, models a dynamic system strongly influenced by external factors. As related in Section 2.4.2.3, the basic ingredients of state information are the characterization of each vehicle as being either busy or free, and the identity of the decision-precipitating event. Dispatch decision points are generated by a call for service or by the completion of a task by a vehicle (see Section 2.4.1). While the state of the system immediately after a decision follows directly from the previous state and the decision (which vehicle, if any, to dispatch), both the event precipitating the next decision point and the time that that next event will occur are known only probabilistically. Two types of external factors may be identified: the phenomena in the environment that generate calls for service, and the application of service to the phenomenon being addressed. In model terminology, the external factors manifest themselves as call rates and service times, quantities whose behavior can be conveyed only in probabilistic terms.

The more complex model of the deliverer dispatch problem may incorporate external factors that impinge on different components of the system. First, consider the set of customers. Each customer may be perceived as a consumer who consumes the delivered good in a probabilistic and uncontrollable fashion, or a retailer who experiences random demands for the good that must either be filled out of inventory or be lost or backordered. The evolution of these geographically distributed inventories usually determines in great measure the quality of system performance (see Section 2.4.4). Hence, information about inventory levels, although possibly not required to determine solution feasibility, certainly assumes a prominent part in defining the state of the distribution system.

In many cases, the inventory position of a particular customer at the next decision point in time may be given as the sum of the current inventory level and the quantity to be delivered according to the current decision, minus the demand experienced from the current decision point to the next. Each of these three components may be rendered uncertain in magnitude by the action of external factors. First and foremost, many of the imaginable cases of DDPs would model inventory depletion processes as stochastic, pure death processes, often assumed to obey a known probability function. Second, a dispatch decision may make delivery quantities contingent on inventory or other information obtained after the current decision point (e.g., upon arrival at the customer location). Third, at a given decision point, the inventory levels themselves may be treated as random variables for the purpose of modeling imperfect information about customer inventories. The characterization of a DDP must provide an explicit formulation of the system's inventory state transition behavior, including the identification of external factors and their effects.

Uncertainty may also be present in the evolution of the vehicle availability portion of state information. The availability of a vehicle can be represented in the system state by the number of time units remaining until the vehicle may next be dispatched (zero if currently ready for dispatch). This measure is best treated as a random variable due to the many sources of availability uncertainty. For instance, suppose a dispatch decision is of the following format:

> Visit these customers in the given sequence. Make the delivery to each visited customer that brings the inventory level to the given quantity. Return when supply is exhausted.

If external factors are active in customer inventory depletion, then inventory levels, and therefore deliveries, are uncertain. This uncertainty carries over to the duration of the assignment, which depends on the inventory levels of the visited customers. Other factors contributing to future vehicle availability uncertainty which may be present in particular DDPs include stochastic vehicle travel times, stochastic lengths of stay at delivery points, and random vehicle outage effects. To allow the tracking of the evolution of the "time until next dispatch" random variables, a DDP statement must detail the effects of all factors that may interact with vehicle availability.

The foregoing examples highlight the need for a complete account of state transition behavior in vehicle dispatching problem characterizations. It is particularly crucial to focus on the effects of external factors. "State information" may comprise a truly extensive body of information for some instances of the vehicle dispatching problem. In these instances, progress in problem characterization can be achieved by assembling reference sets of VDP archetypes. The construction of such sets remains an undertaking for future characterization efforts.

2.4.4 System Performance

The ultimate criterion of the quality of a dispatch decision lies in the performance of the system following the execution of the decision. Predicted performance acts as a driving force in the decision-making component of a vehicle dispatching system. Hence, no characterization of a problem abstracted from a vehicle dispatching operation can be considered complete until the mechanism that interprets and grades system performance is described. This section discusses the system performance function in its various forms.

System performance can be related as the degree to which the evolution of the system corresponds to certain idealized patterns of evolution. Like system evolution, performance must be expressed as a function of three system elements: the state of the system, the dispatch decision, and external factors. In practical terms, it is often more useful to limit the appraisal of system performance to performance in the interval between dispatch decision points. Under this qualification, performance results from the state and decision at the first decision point, plus external factors active from then until the second decision point. Because performance is actually a perception of evolution, we may yet need to extend our descriptions of state, decision, and external factors to include the extra information needed to interpret evolution. This idea will be reinforced below as we return to our illustrative VDP examples.

We begin once again with the vehicle routing problem. Performance in this context means the closeness of the solution's objective value to the optimal. One common objective is the minimization of total travel cost, where cost may be expressed in terms of money, distance, or time. Suppose, in a particular case, the unit of cost is monetary. Then additional information must be supplied in the state of the system if, for instance, the network arc lengths and travel range limits are given in units of time or distance. This shows that the list of information elements embodied in the system state may need to be augmented in order to contain all information relevant to performance.

Many plausible performance measures have been identified in the literature for emergency dispatch problems. Perhaps the most intuitively appealing and easy to calculate performance measure is the mean response time, i.e. the mean time elapsed from the arrival of a call for service until a vehicle first reaches the point of origin of the call and begins administering service.

Components of the response time include dispatcher reaction time, queueing delay, and travel time. The average measure of response time may be superseded in preferability by the probability that the response time exceeds a certain threshold value, although the latter measure is generally more difficult to evaluate. Subsidiary goals in emergency dispatch systems include the balancing of workloads among the fleet of vehicles, and the maintenance of customer-primary server relationships by dispatching non-primary servers to customers as infrequently as possible. In fact, the ideal performance function may be multidimensional in many cases, and would require the imposition of a multiattribute utility function for the determination of the most preferred dispatch decision, or to identify a class of "good" decisions. These considerations may lead to historical and preferential information being added to the system state, and/or a finer detailing of external factor behavior.

The deliverer dispatch problem can incorporate both deliverer and customer objectives. Deliverer objectives, and therefore supplemental state information, will follow in form what has been given for the VRP. Customer objectives, often expressed in inventory terms, may be included on a per customer basis or amalgamated into a quantity for the system as a whole. If both deliverer and customer performance measures are related in monetary terms, overall system performance may be expressed in a single monetary figure. If, instead, customer performance is assessed by frequency and duration of stockouts while deliverer performance is measured monetarily, or if each customer is monitored separately, the resulting performance function will be multidimensional, and decision analysis techniques must be employed to select a dispatch action.

Several other important issues readily arise in conjunction with the DDP. In tracking system evolution, the only concern is the state of the system at dispatch decision points; in gauging system performance, what happens between decision points makes a difference. For instance, consider a DDP where customers backorder demands when stocked out, and where a vehicle dispatched at one decision point always returns before the next. Then if a certain customer is to receive a delivery during the upcoming decision point interval, the customer's inventory level at the next decision point does not depend on when the vehicle dispatched at the current decision point reaches that customer. However, system performance may depend on time of delivery, because holding and backorder costs may reflect the moment-to-moment inventory/backorder level. This idea carries important consequences for the contents of the state, decision, and external factors components. Inventory depletion rates (external factors), in the form of probability distributions, need only be based on the timespan of a decision point interval for evolution-tracking purposes; the timespan might require refinement for assessing performance over the interval, especially if depletion rates vary significantly over time. Further, the targeted time of arrival must also be included in the decision. And in the case where vehicles do not always return by the next dispatch decision point, state information should include some account of the current degree of fulfillment of each outstanding vehicle's delivery assignment.

Performance objectives may also influence feasibility in some problems. A specific case of this behavior occurs when some system performance criteria are expressed as constraints on dispatch decisions. For instance, in the DDP, a constraint limiting the probability of stockout of each customer inventory may be included in the problem constraint set. Then, although the resulting system objective may simply be expressed as minimization of transportation costs, the implicit inventory objective must not be overlooked. Such constraints may be better characterized as elements of the system performance function than of system structure.

It should be clear from the foregoing discussion that the characterization of the system performance function in the general VDP may be a difficult task. Once again, we can be successful in the VDP characterization endeavor provided we can assemble a reference set of performance functions of sufficient scope to cover most types of VDPs, and can develop a concise means of describing those functions that lie outside the set.

2.5 Further Examples

In this section, some of the vehicle dispatching problems that have appeared in the operations research literature are described and characterized according to the core components we have identified.

2.5.1 Inventory Routing Problem

Dror [8] defines the inventory routing problem (IRP) in this manner:

> "We are given two subsets of customers, M and M', together with their corresponding storage capacities. We are also given their inventory status for each day of a specified planning period. The customers' inventory is to be replenished from a central facility using a fleet of vehicles with a known capacity. The fleet size and the maximal number of routes per vehicle per day is given for each day of the period. During the planning period, no customer is replenished more than once. Customers that belong to the subset M' may not be replenished during the period but the customers from M must be. We are also given the time limit on the drivers' work day. In addition, the cost coefficients for replenishing the customers are given for each day of the planning period. Travel costs and travel times between customers are given together with the unloading times on each day for each customer. The objective is to find the routing sequence for each vehicle on each route for each day that minimizes total distribution cost for the planning period as expressed by the cost coefficients, without violating any service, capacity, or time constraints." (p. 6)

Decision-precipitating events: This problem is an extension of the vehicle routing problem to cover a multi-day planning period and inventory considerations. We therefore identify only one dispatch decision-precipitating event, the start of the planning period; it occurs exactly once in any potential evolutionary path.

System structure: Many elements of the IRP system structure can be found in the structure of the VRP. A solution consists of a set of routes for each vehicle, for each day of the planning period. To be feasible, a solution must provide for exactly one delivery to each customer in set M, for no customer to receive more than one delivery, for restrictions on vehicle capacity and driver time to be observed, for all routes to be valid in the given network, and for each route to be handled by exactly one vehicle. The information required to determine feasibility can be inferred from the above constraints.

System evolution: Since there is only one dispatch decision point, no evolution takes place; no external factors need be considered.

System performance: Performance is measured in total costs accruing from both transportation and inventory components. The inventory component allows for an "opportunity saving" to be achieved by servicing customers who do not absolutely require service during the planning period.

The IRP and its solution are actually intended to be components in a computerized system for supporting dispatching to minimize long-run (specifically, annual) distribution costs [8]. In principle, the annual dispatching problem could be solved in its entirety prior to the start of the first planning period, assuming that all demand information is known at this point, since all IRPs solved are transformed into deterministic problems. However, it may be possible to retain both demand uncertainty and computational feasibility by construing the problem as an infinite horizon dynamic decision problem. Here, the dynamically-changing elements of the system state are the sets M and M'. To be specific, the inclusion of customers in the set M' on routes for one planning period affects the constitution of the sets M and M' in the next period. The IRP then satisfies the description of a solution as a reference to another problem as was discussed in Section 2.4.2.1, and the greater problem that the IRP is embedded in is a variety of the deliverer dispatch problem. The opportunity savings mentioned above can be interpreted as assisting in the valuation of alternative future M-M' states of the system, while the other transportation and inventory cost factors measure performance in the upcoming decision point interval.

2.5.2 EDP With A Cutoff-Priority Queue

Schaack and Larson [24] studied an EDP with N servers (vehicles) and multiple call priorities, in which calls of priority i are placed in queue unless and until fewer than N_i servers are busy. The point of this type of queueing discipline is that it grants a higher probability of high priority calls being dispatched to immediately. It does induce longer delays in attending to lower priority calls, though.

Decision-precipitating events: Dispatches may be occasioned by call arrivals or service completions. No dispatch will occur in the former case when a priority i call arrives with at least N_i vehicles busy, and in the latter case when no calls are in queue (or the highest priority call in queue is of priority i and at least N_i vehicles are still busy).

System structure: A solution consists of the dispatch decision to make (vehicle to assign to a call) in any realizable system state. The only infeasible dispatch decisions are those which assign a busy vehicle, or a free one to a priority i call when at least N_i vehicles are busy. Dynamic state information includes the status of each vehicle, the decision-precipitating event, and the number of calls of each priority in queue. In addition, the cutoff N_i is an item of information required to determine feasibility.

System evolution: Once a dispatch decision is made, the queue state at the next decision point is known with certainty. Also, at most one vehicle's status may change from busy to free, and only if the next event is a service completion; none may change in the other direction. (Of course, if the current decision involves a dispatch, that vehicle's status immediately changes from free to busy.) Given that interarrival times of calls and service durations are negative exponentially distributed, the next event type and the time of its occurrence are discretely and negative exponentially distributed, respectively.

System performance: Relevant system performance measures include the mean response times to priority i calls for each priority level i, and the probabilities that calls of each type are placed into queue.

It is somewhat ambiguous whether the cutoff-priority queue scheme defines a distinct vehicle dispatching problem or simply serves as a plausible decision rule for ordinary priority queues. One could argue the question either way, both here and in other VDPs as well.

2.5.3 Coordinated Inventory Replenishment

Although not designed to be a vehicle dispatching problem, the problem of coordinated inventory replenishment (see Silver [26]) can be construed as one. The basic idea is to devise policies for periodically replenishing a set of (in our case, geographically distributed) inventories, where it is more economical transportation-wise to replenish several customers on one dispatch than it is to dispatch to each individually. The main difference between this problem and the DDP is that the former simplifies several problem components so as to facilitate the derivation of a certain type of replenishment policy.

Decision-precipitating events: If the inventory monitoring system operates under continuous review, each and every moment in system evolution is a dispatch decision point. Under the assumption of negative exponentially distributed intervals between customer inventory depletions, it is possible to show that one need only consider dispatching when a depletion event has just occurred. If periodic review is used, the review points are the dispatch decision points, although actual dispatch may occur between review points.

One convention often employed is to restrict replenishment policies to those of the (S,c,s) variety. An (S,c,s) policy is one in which customer i places an order when his inventory position falls to or below s_i, and all other customers j whose inventory positions are lower than c_j are replenished on the same dispatch: the delivery to each customer j in the dispatch is S_j minus the current inventory level. By restricting our attention to such policies, we may reduce the set of decision-precipitating events to customer order arrivals.

System structure: A solution, in the general case, consists of a decision on whether or not to dispatch, and if so, of how much to deliver to whom, for each possible combination of customer inventory positions. If we only consider (S,c,s) policies, decisions only occur when orders arrive, so a dispatch is assured at each decision point. Under the (S,c,s) policy restriction, a feasible decision must include a delivery to the orderer, but otherwise, no restrictions on total delivery size is enforced. The idea that vehicles carry the replenishment orders is incidental to these problems, although one could certainly introduce them by appending constraints on total order size and travel distance. These measures severely complicate the process of determining good (S,c,s) policies and have not been treated *per se* in coordinated replenishment research efforts. Requisite state information consists of inventory levels, and for the (S,c,s) qualification, the set of (S,c,s) policies.

System evolution: In all cases studied in the literature, a customer's new inventory position at the next decision point is found by adding the inventory position at the current decision point and the delivery quantity (if the customer is to receive a delivery), and subtracting quantity demanded through the next decision point (inventory position may or may not be bounded below by zero). Demand is usually assumed to follow a Poisson arrival law, and the size of each demand may be one unit or randomly distributed.

System performance: In Silver [26], the objective is to minimize transportation and inventory holding costs (the relevant performance measure) subject to certain stockout constraints. The transportation cost function is of a particularly simple form: to replenish k customers on one dispatch costs $A+ka$ units, where $A>a$. This cost structure poorly models transportation resource expenditure over a geographic region, but once again, more realistic structures tend to render the problem intractable. The fact that leadtimes may be positive mandates performance to be analyzed throughout each decision point interval.

2.6 Summary

Using a dynamic decision paradigm, we have partitioned the set of vehicle dispatching problem characteristics into four groups: the set of decision-precipitating events, the system structure (including the nature of a solution and the feasibility discriminating mechanism), system evolution characteristics (including the activity of external factors), and criteria of system performance. We sought to illustrate characterization and highlight important associated issues through the analysis of three prototypical vehicle dispatching problems: the vehicle routing problem, the emergency dispatch problem, and the deliverer dispatch problem. Since the number of existing and potential VDPs is so great, progress toward the objective of routine and concise VDP characterization requires in part the construction of reference sets of VDP elements. It is here that further VDP characterization efforts should be focused.

CHAPTER III

FORMULATION OF THE DELIVERER DISPATCH PROBLEM

3.0 Introduction

This chapter describes various formulations of deliverer dispatch problems. Initially, a general formulation is presented. The general DDP is intended as a foundation for the development of more useful DDPs; in its detail, it will almost surely prove impervious to analysis for optimal or near-optimal dispatching procedures, and may only serve practically as a realistic model for simulation work. Following the general formulation, restrictions are introduced to obtain interesting and more tractable specialized DDPs. Some of these special cases are shown to be time-homogeneous Markov decision problems of a certain form.

3.1 The General Deliverer Dispatch Problem

The basic objective of the deliverer dispatch problem is to find the dispatch policy that minimizes expected cost per dispatch decision. A dispatch policy returns a dispatch decision or vehicle assignment when supplied with the information profile of the system at any point in time. The information profile contains information about all system characteristics that may change in value over time, and that are relevant to dispatch decision-making. Costs are generated by transportation of the delivered good, by holding of the good by customers in their inventories, and by demands arriving to stocked-out customers. The terminology presented thus far will be clarified in meaning in the problem formulation below. A summary of the symbols employed in this chapter appears in Exhibit 3.1.

Dispatch decisions are made at *decision points*. These decision points are regarded as instants in time and extend over an infinite planning horizon. The interval between two successive decision points is termed a *period*. All periods are taken here to be equal in duration. Decision

Exhibit 3.1: Symbols Used in Chapter III

C_{jt}:	transportation cost of itinerary j at decision point t
d_{ij}:	delivery to customer i in itinerary j
e_{it}:	lost demand for customer i from t to $t+1$
g^{δ}:	cost rate of policy δ
H_{it}:	inventory holding cost of customer i at t
i:	customer index
I_t:	information profile at t
j:	itinerary index
k:	vehicle index
K:	number of vehicles
l_{ij}:	leadtime to customer i in itinerary j
l^+:	maximum leadtime
L_{it}:	lost demand cost for customer i at t
m:	number of customers
n:	number of itineraries
$p_i(x_{it})$:	probability of customer i experiencing demand x_{it} at t
$p_{ss'}{}^{\gamma}$:	transition probability from s to s' under decision γ
P^{δ}:	transition probability matrix for rule δ
$q_s{}^{\gamma}$:	cost of decision γ in state s
s:	state of DDP
S:	number of states in DDP
t:	decision point index
u_i:	holding capacity of customer i
w_{kt}:	number of periods until vehicle k ready for dispatch (at decision point t)
x_{it}:	number of units demanded from customer i between t and $t+1$
y_{jkt}:	dispatch decision variable (1 if vehicle k assigned itinerary j at t)
z_{it}:	inventory level of customer i at t
δ:	dispatch rule (or stationary dispatch policy)
Δ:	dispatch policy
$\pi_s{}^{\delta}$:	steady-state probability of occupying state s under stationary policy δ
τ_j:	duration of itinerary j
Ω_k:	capacity of vehicle k

points are indexed by t, but the assignment of indices to decision points changes in such a way that the current decision point is marked by $t=0$.

Customers are indexed by i, $i=1, \ldots, m$, m being the total number of customers in the system. Each customer has an inventory of the delivered good whose level varies over time. Let the inventory level of customer i at decision point t be z_{it} (an integer). Inventory levels diminish due to demands (orders placed with the customer that the customer either fills out of inventory or, in the current formulation, loses altogether) and increase by deliveries. Customer i's inventory level may never be less than 0 (no backorders) nor more than its capacity u_i. Let the joint probability that x_{it} (an integer) units of the good are demanded from customer i during period t (the period from decision point t to decision point $t+1$), for each customer i, be written

$$p_t(x_{1t}, x_{2t}, \ldots, x_{mt} \mid I_t),$$

where I_t is the historical information profile (record of all previous inventory levels, demands, and dispatches) at decision point t. Hence, there may exist correlations among demands between and within periods.

The vehicle fleet consists of K vehicles, each having a certain maximum carrying capacity. Let the carrying capacity of vehicle k be denoted Ω_k. Vehicle availability is represented by the time to availability (TTA) quantity w_{kt}. If w_{kt} equals 0, then vehicle k is available for dispatch at decision point t. Otherwise, w_{kt} represents the number of periods remaining until vehicle k first becomes available. (One could generalize the TTA reduction process to account for stochastic travel times, and also to depend on the decision point t and factors relating to presumed load and location chracteristics of vehicle k at decision point t. We avoid this generalization here.)

An *itinerary* is a subset of customers that a vehicle is to visit between its departure from a depot until its next return to that (or another) depot, and a description of the activities it is to perform at each visit. Specifically, since the primary vehicle activity is delivery of physical goods in the systems we are studying, an itinerary consists basically of a list of customer-delivery quantity pairs. Itineraries are indexed by $j, j=0, 1, \ldots, n$, where $j=0$ refers to a vehicle not being dispatched at a decision point. In our DDP, any itinerary j is fully defined by specifying the following quantities:

1) the (integral) number of units d_{ij} delivered to each customer i in itinerary j (0 if customer i is not visited);

2) the transportation cost C_{jt} of each dispatch of itinerary j at decision point t;

3) the duration τ_j (an integral number of periods) of itinerary j (i.e., a vehicle dispatched on itinerary j at decision point t is next available for dispatch at decision point $t+\tau_j$); and

4) the leadtimes l_{ij} between the dispatch of itinerary j and the arrival of the delivery to each customer i on that itinerary (for i with $d_{ij}>0$). (Leadtime works as follows: if the leadtime for customer i on itinerary j is l_{ij}, and if itinerary j is dispatched at decision point t, then the delivery to i has not occurred by decision point $t+l_{ij}$, but has by point $t+l_{ij}+1$.)

It is presumed that the itinerary set is *supplied to* the DDP rather than being *created within* it. A practical implementation of a CAD system configuring its prescriptive dispatch capability around the DDP would require an itinerary generator to set up the itinerary set in a preliminary phase of the implementation. Itineraries may be designed on the basis of expert testimony and/or cost and travel time approximation functions utilizing the solutions of traveling salesman problems. Note that two itineraries can visit the same subset of customers yet be distinguished by the deliveries to those customers, or have the same customer-delivery quantity characteristics yet differ on, say, leadtime characteristics (because the customers are visited in different sequences). In order for vehicle k to be dispatched on itinerary j, it must be true that

$$\sum_{i=1}^{m} d_{ij} \leq \Omega_k .$$

Also denote the maximum leadtime among all i-j pairs by l^{*}.

The *dispatch decision* at decision point t can be expressed as the set of decision variables $\{y_{jkt}\}$, where y_{jkt} has value one if and only if vehicle k is dispatched on itinerary j at decision point t. Since only available vehicles may be dispatched, then feasible dispatch decisions must have

$$\sum_{j=1}^{n} y_{jkt} \leq 1 - sgn(w_{kt}), \tag{3.1}$$

where $sgn(x)$ is 1, 0, or -1 when x is positive, zero, or negative, respectively.

The transition function for customer i's inventory level from decision point t to decision point $t+1$, given the demand x_{it} and the dispatch decisions $\{y_{jk,t-\tau}\}$ for $\tau = 0, \ldots, \ell^{+}$, can be written

$$z_{i,t+1} = min\left[u_i, max\left(0, z_{it} + \sum_{k=1}^{K}\sum_{j=1}^{n} d_{ij} y_{jk,t-l_{ij}} - x_{it}\right)\right]. \tag{3.2}$$

Correspondingly, the number of units demanded from and lost by customer i during period t, which we signify by e_{it}, is found by

$$e_{it} = max\left(0, x_{it} - z_{it} - \sum_{k=1}^{K}\sum_{j=1}^{n} d_{ij} y_{jk,t-l_{ij}}\right). \tag{3.3}$$

One implication of the inventory transition function is that demands encountered during a period may be satisfied by units held in inventory any time during the period, as if all deliveries occur before any demands are registered. Another is that units that cannot be held in inventory due to customer capacity limitations disappear at the end of each period. The inventory costs incurred by customer i from decision point t to decision point $t+1$ are the holding cost $H_{i,t+1} \cdot z_{i,t+1}$ and the lost demand cost $L_{it} \cdot e_{it}$, where H_{it} and L_{it} are, respectively, the unit holding cost at decision point t and the lost demand cost in period t.

The objective function for the deliverer dispatch problem may be expressed, at least conceptually, as

$$min_{\{y_{jkt}\}} \lim_{T\to\infty} \left\{\frac{1}{T} E_{\{x_{it}\}} \sum_{t=0}^{T-1}\left[\sum_{j=0}^{n} C_{jt} \sum_{k=1}^{K} y_{jkt} + \sum_{i=1}^{m}\left(H_{i,t+1} z_{i,t+1} + L_{it} e_{it}\right)\right]\right\}, \tag{3.4}$$

with $z_{i,t+1}$ and e_{it} defined in terms of z_{it}, $y_{jk,t-l_{ij}}$, and x_{it} as above. Depending on structures of demands and costs over time, the limit in expression (3.4), and therefore the entire objective function, may or may not be well-defined. The specializations we introduce next allow us to surmount this difficulty while still retaining most of the general DDP's desirable characteristics.

3.2 Specializations of the General Problem

This section describes a series of specializations or refinements to the general deliverer dispatch problem. Each specialization here builds on the previous ones, although any subset may be taken for modeling particular operations. We target one specialized version of the general problem as the DDP we direct our algorithmic development toward.

3.2.1 Time-Homogeneity of Costs

This specialization says that, for every decision point t,

1) $C_{jt} = C_j$;

2) $H_{it} = H_i$; and

3) $L_{it} = L_i$.

Also, these costs are all finite.

3.2.2 Demand Independence From Historical Information

Under this refinement, demand during each period does not depend on the demands of previous periods, nor does it depend on previous decisions. Hence,

$$p_t(x_{1t}, \ldots, x_{mt} \mid I_t) = p_t(x_{1t}, \ldots, x_{mt})$$

for any decision point t and historical information profile I_t.

3.2.3 Cyclic Demand

Here we constrain the demand process to follow the cyclic pattern

$$p_t(x_{1t}, \ldots, x_{mt}) = p_\tau(x_{1t}, \ldots, x_{mt}),$$

where $\tau = (t+\theta)\ mod\ N$, θ is the phase of the cycle, and N is the number of periods in a cycle. For example, if $N=4$ and $\theta=3$, then $\tau=3$ for decision points 0, 4, 8, etc. Inclusion of the phase is necessary because the t index always changes in meaning with respect to absolute time. The problem formulation must include the transformation $\theta=(\theta+1)\ mod\ N$ for when the current decision point advances one period in time.

Through specializations 3.2.1-3.2.3, we have refined the original, nonstationary (with respect to time) DDP into a stationary one, which renders the objective function of Section 3.1 well-defined.

3.2.4 Steady Demand

This specialization corresponds to specialization 3.2.3, when N, the cycle length, is one. In other words,

$$p_t(x_{1t}, \ldots, x_{mt}) = p(x_{1t}, \ldots, x_{mt}),$$

for each period t.

3.2.5 Independent Demands Among Customers

If each customer's demand probability law does not depend on other customers' demands during the same period, then we may write the joint demand probability law in product form; i.e.,

$$p(x_{1t}, \ldots, x_{mt}) = \prod_{i=1}^{m} p_i(x_{it})$$

where here $p_i(x_{it})$ is the demand probability mass function of customer i (for all t).

3.2.6 Identical Vehicles

This specialization says that all vehicles are identical in their carrying capacities. In the given terminology, this translates to

$$\Omega_k = \Omega$$

for every vehicle k.

The general deliverer dispatch problem with specializations 3.2.1 through 3.2.6 constitutes the main DDP to be studied in this dissertation. Two more specializations are listed below.

3.2.7 No Leadtimes

In this specialization, all leadtimes l_{ij} are taken to be zero. Hence, all deliveries from itineraries dispatched at decision point t enter their respective inventories by decision point $t+1$, so that, for instance,

$$z_{i,t+1} = min\left[u_i, max\left(0, z_{it} + \sum_{k=1}^{K} \sum_{j=1}^{n} d_{ij} y_{jkt} - x_{it}\right)\right]$$

for all customers i and decision points t. This specialization renders previous dispatch information useless, since all previously dispatched vehicles will have made their deliveries by the next decision point.

3.2.8 Phased Dispatching

This final specialization permits the removal of vehicle TTA measures from the information profile. We first specify that the duration τ_j of each itinerary j is a constant τ (even for $j = 0$, which is difficult to justify on realistic grounds unless $\tau = 1$). We further stipulate that $K = K'\tau$, where K is the number of vehicles and K' a positive integer. Finally, letting t_k represent the last decision point when vehicle k was dispatched ($t_k < 0$ because $t = 0$ always represents the current decision point), we specify that vehicle k can only be dispatched at current and future decision points t when

$$(t - t_k)\, mod \tau = 0.$$

For example, with a vehicle fleet size of 10 and $\tau = 5$, if vehicle 2 was last dispatched at decision point -3, then it can be dispatched at decision points 2, 7, 12, and so on. Note that the interdispatch time for a vehicle equals the uniform itinerary duration τ, and that exactly K'

vehicles are available at every dispatch (hence the term "phased dispatching"). There is thus no need to track vehicle availability under this specialization. The DDP incorporating all these specializations was studied in [19].

3.3 The Deliverer Dispatch Problem as a Markov Decision Problem

In this section, the deliverer dispatch problem specialized in Sections 3.2.1-3.2.6 (referred to in this section as "the" DDP) is shown to be a time-homogeneous Markov decision problem (MDP). This relationship enables a much simpler conceptualization of the objective in the DDP.

The first concept to be handled in establishing the relationship between DDPs and MDPs is that of state. The *state* of a stochastic process or system is the set of values, at a given point in time, of the minimal set of system characteristics that contribute to the prediction of both future evolution and performance (in the terminology of Chapter II) of the system. Because the decision processes of our interest are Markovian, we may discard knowledge of any previous state and still retain the same degree of accuracy in projecting future evolution and performance solely from information contained in the current state. The existence of events occurring in the past and having future relevance can be accommodated in the Markov decision problem framework by embedding this information in the state.

What constitutes a state in the DDP? The relevant time-varying characteristics are the inventory levels of the customers, the time to availability measures of the vehicles, and the l^+ previous dispatches. Hence, the state of the system at decision point t may be represented by the vector

$$(z_{1t}, \ldots, z_{mt}, w_{1t}, \ldots, w_{Kt}, \{y_{jk,t-1}\}, \ldots, \{y_{jk,t-l^+}\}).$$

By assuring that itinerary durations are bounded above, the state space of the DDP is finite, if combinatorially large. Let the cardinality of the state space be S. We may then associate a unique positive integer with each state, so that the state space may also be expressed as $\{s\}$, where s ranges from 1 to S.

Let us restrict the set of types of dispatch decision-making processes we may consider in a search for an optimal dispatching procedure to those which associate exactly one dispatch decision with each potential state of the system. (It is proven in [12] that this restriction never removes all optimal procedures from consideration among the DDPs we address.) We define a *dispatch rule* δ to be an S-vector whose s-th element $\delta(s)$ is the dispatch decision $\{y_{jk}\}$ taken when the state of the system is s. A *dispatch policy* Δ assigns a dispatch rule δ_t to each decision point t; a *stationary* dispatch policy has $\delta_t = \delta$ for all t, and may be alternately identified as "the stationary dispatch policy δ" or just "the dispatch policy δ."

We now claim that, for a given stationary dispatch policy δ, the stochastic process governing the state transition behavior of the deliverer dispatch model is a discrete-state, discrete-time, time-homogeneous Markov chain. To see that this is so, we must examine the model and the way each state element changes from one decision point to the next.

First, that the stochastic process is discrete-state is clear by definition of the state. The process is discrete-time, because decisions are only made, and therefore the state is only observed, at distinct instants in time. Neither costs nor demand probability laws change over time, and only a constant, finite number of previous dispatches are retained in the state space, so the process is time-homogeneous. All that remains to demonstrate is that all of the available information that may aid the prediction of future process evolution is contained in the current state of the system.

Let the state of the system at decision point t be s, which corresponds to the state

$$(z_{1t}, \ldots, z_{mt}, w_{1t}, \ldots, w_{Kt}, \{y_{jk,t-1}\}, \ldots, \{y_{jk,t-l^-}\}),$$

and consider the transition probability to the state s' equivalent to

$$(z_{1,t+1}, \ldots, z_{m,t+1}, w_{1,t+1}, \ldots, w_{K,t+1}, \{y_{jk,t}\}, \ldots, \{y_{jk,t+1-l^-}\})$$

when the dispatch decision $\{y_{jkt}\}$ is made at decision point t. Let σ be the set of all m-vectors $(x_1,\ldots,x_m)$ such that the following relationships hold:

$$z_{i,t+1} = min\left[u_i, max\left(0, z_{it} + \sum_{k=1}^{K}\sum_{j=1}^{n} d_{ij} y_{jk,t-l_{ij}} - x_i\right)\right] \quad \forall\, i, \tag{3.5}$$

and

$$w_{k,t+1} = max\left(0, w_{kt} + \sum_{j=1}^{n} \tau_j y_{jkt} - 1\right) \quad \forall\, k. \tag{3.6}$$

The first set of relationships indicate that all inventory levels embedded in state s have increased or decreased to their respective levels of state s'. The second set assures that vehicle TTA measures have made the appropriate transformation from their levels in state s at t to theirs in s' at $t+1$; unavailable vehicles at decision point t ($w_{kt}>0$, therefore $\Sigma_j\, y_{jkt}=0$) have had their availability measures reduced, available vehicles that have been dispatched ($w_{kt}=0$ and $y_{jkt}=1$ for the itinerary j dispatched with vehicle k) become unavailable for the duration of the dispatch assignment, and available vehicles not dispatched ($w_{kt}=0$ and $\Sigma_j\, y_{jkt}=0$) remain available. The probability of occurrence of the random event $(x_1, \ldots, x_m)$, denoted $p(x_1, \ldots, x_m)$, is equal to

$$p(x_1, \ldots, x_m) = \prod_{i=1}^{m} p_i(x_i),$$

due to the independence of demands. Let

$$p_{ss'}^{\delta(s)} = \sum_{(x_1, \ldots, x_m) \in \sigma} p(x_1, \ldots, x_m).$$

Then, provided that the previous dispatch state elements of state s' are the same as those of state s with $\{y_{jk,t-l^+}\}$ deleted, the remaining previous dispatches shifted one decision point into the past, and the decision $\{y_{jkt}\}$ at t added as the most recent dispatch, it may be verified that:

1) $p_{ss'}{}^{\delta(s)}$ is the probability of a transition from state s to state s' in one period, and

2) $p_{ss'}{}^{\delta(s)}$ can be determined solely from the elements of state s and the dispatch policy δ.

Thus, the state transition process of the deliverer dispatch model obeys all the conditions for qualification as a time-homogeneous Markov chain.

It remains to be demonstrated that the decision and performance structures of the DDP conform to Markov decision problem specifications. In the standard MDP, a cost $q^{\gamma}{}_s$ is incurred when the chain occupies state s at a decision point and decision γ is taken. These state occupancy costs may be expectations of cost outcomes over probability distributions. Performance over any horizon is judged according to the sum of occupancy costs incurred during that horizon. The decision objective of the MDP is to select the policy δ which yields the minimum expected occupancy cost per decision point over an infinite horizon. The length of a period is likely to be too short to permit the advantageous use of Markov decision theory for the discounted net present cost case, since the discount factor would be so close to 1 that the usual methodology would be very slow to converge (see [12, p. 171]).

From the DDP's objective function given in (3.4), we can infer that performance over any finite horizon does relate to the sum of state occupancy costs of the form

$$q_s^{\gamma} = \sum_{j=1}^{n} C_j \sum_{k=1}^{K} y_{jk,0} + E_{\{x_i\}} \sum_{i=1}^{m} \left(H_i z'_i + L_i e_i \right), \tag{3.7}$$

provided that

$$z'_i = min\left[u_i, max\left(0, z_i + \sum_{k=1}^{K} \sum_{j=1}^{n} d_{ij} y_{jk,-l_{ij}} - x_i \right) \right], \tag{3.8}$$

$$e_i = max\left(0, x_i - z_i - \sum_{k=1}^{K} \sum_{j=1}^{n} d_{ij} y_{jk,-l_{ij}} \right), \tag{3.9}$$

s corresponds to the state

$$(z_1, \ldots, z_m, w_1, \ldots, w_K, \{y_{jk,-1}\}, \ldots, \{y_{jk,-l^+}\}),$$

and the decision $\gamma = \{y_{jk,0}\}$ is feasible in state s. Note that all information needed to determine each occupancy cost is included in either the state s or the decision γ. The MDP decision structure thus applies fully to the deliverer dispatch problem, and we may conclude that the DDP is a Markov decision problem of the form detailed above.

Given further assumptions about the types of Markov chains yielded by potential dispatch policies for the deliverer dispatch problem, we may prepare a more succinct statement of its objective function. Suppose, again, that the only allowable type of dispatching procedure for the

DDP is of the dispatch policy type, and that only stationary dispatch policies are considered. Let P^δ be the SxS matrix whose element in row s and column s' is $p_{ss'}^{\delta(s)}$; P^δ is the state transition matrix of the Markov chain induced by policy δ. Further, require the combination of probabilistic system components and feasible dispatch policies to only produce Markov chains with exactly one recurrent class which is aperiodic--this is the *unichain* assumption. Then for each dispatch policy δ, there exists a single set of stationary unconditional state occupancy probabilities $\{\pi^\delta_s\}$. An unconditional state occupancy probability is the probability that a Markov chain occupies a certain state at a certain future point in time, not conditioned on any previous state occupancies. The likelihood is that, for any point in time within some finite horizon, these probabilities cannot be calculated without knowledge of the unconditional state occupancy probabilities at the beginning of the horizon. Under the unichain assumption, however, the unconditional state occupancy probabilities will converge to the stationary values as projections further and further into the future are made. The stationary values correspond to the unique solution to the set of simultaneous linear equations

$$\pi^\delta P^\delta = \pi^\delta \quad and \quad \sum_{s=1}^{S} \pi_s^\delta = 1, \tag{3.10}$$

where $\pi^\delta = (\pi^\delta_1, \ldots, \pi^\delta_S)$. Likewise, averaging costs over the infinite horizon causes expected cost per decision point to converge to the stationary expected cost or cost rate

$$g^\delta = \sum_{s=1}^{S} \pi_s^\delta q_s^{\delta(s)}. \tag{3.11}$$

A more useful form for the DDP objective is then

$$min_\delta \; g^\delta = \sum_{s=1}^{S} \pi_s^\delta q_s^{\delta(s)}, \tag{3.12}$$

the minimization of the average cost per decision point or *cost rate*.

CHAPTER IV

SOLUTION PROCEDURES FOR THE DELIVERER DISPATCH PROBLEM

4.0 Introduction

To find the optimal dispatch policy for an instance of the deliverer dispatch problem, we may attempt to use any procedure for solving a Markov decision problem. However, we shall find that it is impractical to solve any but the smallest problems exactly, since the numbers of states and decisions grow exponentially in the number of customers and vehicles. Our hope for constructing a viable dispatch selection methodology for the type of computer-aided dispatching system that we visualize rests on our being able to devise good heuristic algorithms for handling the DDP. This chapter covers the development of such algorithms. Since some of the heuristic notions that we propose in this chapter are based on approximations to the policy iteration method for solving MDPs (see, e.g., Howard [13]), we initially describe policy iteration. (Note: new symbols used in this chapter are described in Exhibit 4.1.)

4.1 The Policy Iteration Algorithm

The policy iteration algorithm for the Markov decision problem is an iterative algorithm that is guaranteed to find an optimal policy in a finite number of iterations, provided that the unichain assumption holds. The algorithm starts with an initial policy and progressively updates the decision for each state if that yields improvement in the objective function. When no further improvement can be achieved, the algorithm terminates with the solution to the problem being the policy that could not be improved upon. This algorithm is also known as policy improvement and, in the name of its originator, Howard's algorithm.

Before we formally present the policy iteration algorithm, we must develop a new concept, that of a state's "relative value." Define the function $J_{s_0}{}^{\delta}(t)$ iteratively as follows:

Exhibit 4.1: New Symbols Used in Chapter IV

A:	number of vehicles available at the decision point (MSP)
C_{id}:	replenishment cost to i of delivery d (approximated for DDSP)
C_j^+:	dispatch cost of itinerary j (MSP)
d_a:	actual delivery (DDSP)
d_d:	desired delivery (DDSP)
d_{ij}^1:	$=1$ if customer i visited in itinerary j
d_i:	total delivery to customer i in MSP dispatch
F:	dispatch failure probability (approximated for DDSP)
${}_ig^{\delta_i}$:	cost rate in DDSP for customer i
$J_s^\delta(t)$:	expected cost over next t decision points, starting from s with policy δ
J_{id}:	set of itineraries j with $d_{ij}=d$
l_{id}:	leadtime to i of delivery d (approximated for DDSP)
${}_i^Fp_{s_is'_i}^{d_d}$:	transition probability in DDSP for i from s_i to s'_i choosing d_d with failure probability F
${}_i^Fq_{s_i}^{d_d}$:	cost in DDSP for i choosing d_d in s_i with failure probability F
${}_iQ_{s_i}^{d_d}$:	immediate inventory costs in DDSP when choosing d_d in s_i with $F=0$
s_i:	state in DDSP for customer i
S_i:	set of states in DDSP for customer i
S_{iz}:	states s where inventory level of customer i is z
t_{yi}:	time to arrival of outstanding delivery (DDSP)
v_s^δ:	relative value of state s under dispatch policy δ
${}_iv_{s_i}^{\delta_i}$:	relative value of s_i under policy δ_i in DDSP for customer i
y_i:	outstanding delivery (DDSP)
y_j:	$=1$ if route j dispatched (MSP)
γ:	dispatch decision
δ_i:	replenishment policy in DDSP for customer i
$\pi_{s\mid s_0}^\delta(t)$:	Pr{state at t is s | state at 0 is s_0, policy δ in use}
${}_i\pi_{z'_i\mid z_i}^{\delta_i}(t)$:	Pr{state at t is z'_i | state at 0 is z_i, policy δ_i in use} (DDSP)
${}_i\pi_{s_i}^{\delta_i}$:	steady-state probability of occupying s_i under policy δ_i (DDSP)

$$J_{s_0}^{\delta}(1) = q_{s_0}^{\delta(s_0)}; \qquad (4.1a)$$

$$J^{\delta}_{s_0}(t) = J^{\delta}_{s_0}(t-1) + \sum_{s=1}^{S} \pi^{\delta}_{s|s_0}(t-1)\, q_s^{\delta(s)}, \quad t=2,3,\ldots \tag{4.1b}$$

where

$$\pi^{\delta}_{s|s_0}(0) = 1, \quad s=s_0,$$

$$= 0 \ \textit{otherwise}, \tag{4.2a}$$

and

$$\pi^{\delta}_{s|s_0}(t) = \sum_{s'=1}^{S} \pi^{\delta}_{s'|s_0}(t-1)\, p^{\delta(s')}_{s's} \quad t \geq 1. \tag{4.2b}$$

It can be verified that $J_{s_0}{}^{\delta}(t)$ is the expected total cost incurred over a planning horizon consisting of the next t decision points when the current state is s_0 and the stationary policy δ is in effect. The average expected cost incurred per decision point over this finite horizon is

$$\frac{J^{\delta}_{s_0}(t)}{t}.$$

We claim that the limit of this average, as $t \to \infty$, is g^{δ}. To see this, consider the difference

$$J^{\delta}_{s_0}(t) - J^{\delta}_{s_0}(t-1).$$

This difference can be reexpressed, using the steady-state occupancy probabilities $\pi_s{}^{\delta}$ for policy δ, as

$$J^{\delta}_{s_0}(t) - J^{\delta}_{s_0}(t-1) = \sum_{s=1}^{S} \pi^{\delta}_{s|s_0}(t-1)\, q_s^{\delta(s)}$$

$$= \sum_{s=1}^{S} \pi^{\delta}_{s|s_0}(t-1) q_s^{\delta(s)} + \sum_{s=1}^{S} \pi^{\delta}_s q_s^{\delta(s)} - \sum_{s=1}^{S} \pi^{\delta}_s q_s^{\delta(s)}$$

$$= \sum_{s=1}^{S} \pi^{\delta}_s q_s^{\delta(s)} + \sum_{s=1}^{S} \left(\pi^{\delta}_{s|s_0}(t-1) - \pi^{\delta}_s \right) q_s^{\delta(s)}$$

$$= g^{\delta} + \sum_{s=1}^{S} \left(\pi^{\delta}_{s|s_0}(t-1) - \pi^{\delta}_s \right) q_s^{\delta(s)}. \tag{4.3}$$

Now, as $t \to \infty$, $\pi_{s|s_0}{}^{\delta}(t-1) \to \pi_s{}^{\delta}$, so that the limit of the difference is g^{δ}. Hence, for each additional decision point appended to the decision horizon, total expected cost incurred will increase by a factor growing closer and closer to g^{δ}. In the limit, this average increment will dominate the average cost incurred per decision point during the transient portion of the system's evolution to the steady state. A more rigorous argument for the convergence of average expected cost to g^{δ} can be found in [12].

Now consider the quantity $v_{s_0}{}^{\delta}(t)$ defined as

$$v_{s_0}^{\delta}(t) \equiv \sum_{\tau=1}^{t} \sum_{s=1}^{S} \left(\pi_{s|s_0}^{\delta}(\tau-1) - \pi_s^{\delta} \right) q_s^{\delta(s)}. \tag{4.4}$$

One can easily check that

$$J_{s_0}^{\delta}(t) = tg^{\delta} + v_{s_0}^{\delta}(t). \tag{4.5}$$

Define

$$v_{s_0}^{\delta} \equiv \lim_{t \to \infty} v_{s_0}^{\delta}(t). \tag{4.6}$$

Under the assumptions we have made thus far, $v_{s_0}{}^{\delta}$ is always finite (see [12], Section 4-6, for a detailed proof of this statement). We call $v_{s_0}{}^{\delta}$ the *relative value* of state s_0 under stationary decision policy δ, although a more descriptive term might be the "transient cost difference," since it represents the difference between actual and steady-state expected costs accruing from transient activity in the Markov chain. No matter the terminology, this quantity figures prominently in the development of the policy iteration algorithm.

Suppose that, in the course of our search for the best policy for our Markov decision problem, we currently have the best policy δ found thus far. Let us consider the nonstationary policy (meaning that different decision rules may be used at different decision points) of using decision rule δ' at the current decision point and δ at every future decision point. Bellman's *principle of optimality* (see [3]), when applied to this problem, indicates that if this nonstationary policy is better, in some sense, than the stationary policy of using δ always, then a better stationary policy is to use δ' always. The nonstationary policy is judged better if the difference between the old stationary and the new nonstationary policies' performances is in favor of the latter; i.e., if

$$lim_{t\to\infty} J^{\delta}_{s}(t+1) - \left[q_{s}^{\delta'(s)} + \sum_{s'=1}^{S} p_{ss'}^{\delta'(s)} \left(lim_{t\to\infty} J^{\delta}_{s'}(t) \right) \right] \geq 0, \ s=1,2,\ldots,S, \quad (4.7)$$

with strict inequality for at least one state s. Substitute for the total expected cost terms their equivalents in terms of t, g^{δ}, and $v_s{}^{\delta}$:

$$\left(lim_{t\to\infty}(t+1)g^{\delta} + v_{s}^{\delta} \right) - \left[q_{s}^{\delta'(s)} + \sum_{s'=1}^{S} p_{ss'}^{\delta'(s)} \left(lim_{t\to\infty} tg^{\delta} + v_{s'}^{\delta} \right) \right] \geq 0 \quad (4.8)$$

or

$$g^{\delta} + v_{s}^{\delta} - q_{s}^{\delta'(s)} - \sum_{s'=1}^{S} p_{ss'}^{\delta'} v_{s'}^{\delta} \geq 0, \ s=1,\ldots,S. \quad (4.9)$$

Relation (4.9) serves as the foundation of the policy iteration algorithm. Policy iteration consists of alternatively finding the cost rate g^{δ} and the relative values $\{v_s{}^{\delta}\}$ of the current policy δ (the "value determination" phase), and searching for a better policy by testing changes in the decision at each state (the "policy improvement" phase).

First, observe that every policy can be no worse or no better than itself, so that

$$g^{\delta} + v_{s}^{\delta} - q_{s}^{\delta(s)} - \sum_{s'=1}^{S} p_{ss'}^{\delta(s')} v_{s'}^{\delta} = 0, \ s=1,\ldots,S. \quad (4.10)$$

(4.10) is a set of S simultaneous linear equations in the $S+1$ unknown quantities $\{v_s{}^{\delta}\}$, $s=1,\ldots,S$, and g^{δ}. Value determination is achieved by arbitrarily setting one $v_s{}^{\delta}$ (say, $v_S{}^{\delta}$) to zero and solving the S resulting equations for the average expected cost and the relative values. (It may be observed that the policy improvement criterion is unaffected by the addition of a constant to all $v^{\delta}(s)$, hence the use of the word "relative" in "relative value.") Given these quantities, the policy improvement phase involves, for each state s, finding the decision γ that maximizes

$$g^{\delta} + v_{s}^{\delta} - q_{s}^{\gamma} - \sum_{s'=1}^{S} p_{ss'}^{\gamma} v_{s'}^{\delta}, \quad (4.11)$$

or equivalently, that minimizes the *improvement criterion*

$$q_s^{\gamma} + \sum_{s'=1}^{S} p_{ss'}^{\gamma} v_{s'}^{\delta} \; . \tag{4.12}$$

If $\gamma \neq \delta(s)$, then the policy δ', where $\delta'(s) = \gamma$ and $\delta'(s') = \delta(s')$, $s' \neq s$, improves upon δ. In practice, several policy improvements are instituted simultaneously on each policy iteration.

The improvement criterion offers the following insight: The immediate effect of any decision γ when the system is in state s is the incurring of the cost q_s^{γ}--call this the *immediate cost.* If we were solving a finite horizon problem, consisting only of the current decision point, with the same cost and decision structures as the original MDP, the decision γ^1 minimizing the objective function q_s^{γ} would be the optimal decision. But if the problem were in fact an infinite horizon problem, γ^1 would be considered myopic and would probably be suboptimal. Suppose that an additional term were added to the objective function that brought additional cost consequences depending on which states were accessible from s under decision γ. If this term penalized future states according to how undesirable, in some sense, they were, then the decisions obtained from solutions to these modified single-decision point problems might perform better in the infinite horizon problem than myopic decisions. The terms $\Sigma_{s'}\ p_{ss'}^{\gamma}\ v_s^{\delta}$, which we denote as *expected future values,* serve ideally as penalty functions. assuming that the stationary policy δ is used at all future decision points. This is because the expected future value relates in a single quantity the expected sum of costs incurred (relative to what would be incurred at the steady-state cost rate for policy δ), starting from state s' with probability $p_{ss'}^{\gamma}$, until the steady-state is reached. The objective function of the augmented single decision point problem is then the improvement criterion (4.12). The augmentation of a single-period problem with a penalty function is a major feature of the heuristic algorithm for the deliverer dispatch problem that we propose later in this chapter.

The policy iteration algorithm is summarized as follows:

0. Begin with an initial policy δ.

1. Solve the simultaneous linear equations

$$g^{\delta} + v_s^{\delta} - \sum_{s'=1}^{S} p_{ss'}^{\delta(s)} v_{s'}^{\delta} = q_s^{\delta(s)}, \quad s = 1, \ldots, S$$

for $g^{\delta}, v_1^{\delta}, \ldots, v_{S-1}^{\delta}$, with $v_S^{\delta} \equiv 0$.

2. For each s, find γ^* such that

$$q_s^{\gamma^*} + \sum_{s'=1}^{S} p_{ss'}^{\gamma^*} v_{s'}^{\delta} < q_s^{\gamma} + \sum_{s'=1}^{S} p_{ss'}^{\gamma} v_{s'}^{\delta}, \ \forall \gamma$$

and set $\delta'(s) = \gamma^*$.

3. If $\delta' \neq \delta$, set $\delta = \delta'$ and go to step 1. Otherwise, terminate with the optimal policy δ.

4.2 The Infeasibility of Exact Solution of the Deliverer Dispatch Problem

At each iteration, the policy iteration algorithm requires the solution of a set of S simultaneous linear equations, and the calculation of the improvement criterion for each state and decision at that state. In terms of computational complexity, for a Markov decision problem with S states and A alternatives at each state, the first step takes $O(S^3)$ and the second $O(AS^2)$ operations per iteration. Since the number of states of the deliverer dispatch problem grows exponentially in the numbers of customers and vehicles, it quickly becomes clear that solving the deliverer dispatch problem exactly through policy iteration (or any other method) is feasible only for very small problems. It is not even feasible to calculate and store the quantities q_s^{γ} and $p_{ss'}^{\gamma}$ for any but the most trivial instances. Hence, progress toward dealing with DDPs can only be achieved by resorting to approximations of various sorts. The rest of this chapter explores ideas for DDP heuristics. These heuristics are generally predicated on the basic concepts of policy iteration. (A more detailed comparison of computational complexity between policy iteration and the heuristic developed next appears later in this chapter.)

4.3 A Decomposition Heuristic for the DDP

This section develops a Markov decision theory-based heuristic algorithm for the deliverer dispatch problem. The key concept of the algorithm is the utilization of **decomposition by customer** of the original problem for the purpose of estimating the expected future values that comprise a penalty function for a single-period dispatching problem.

4.3.1 The Difficulties of Decomposition

It was intimated in Chapter II that the DDP could be viewed as a problem of coordinated inventory replenishment, but that research aimed at the latter problem has tended to sacrifice model realism (at least in the domain of multilocation inventory theory) for improved analytical capability. Chief among the simplifications invoked are the implied special network structure yielding transportation costs that are linear in the number of customers visited per dispatch, and a vehicle fleet of unlimited capacity. These simplifications allow the development of a decomposition scheme for deriving (S,c,s) policies by solving inventory subproblems for each customer. Recall that in the (S,c,s) inventory policy, customer i triggers its replenishment to the inventory level S_i whenever its inventory level drops to or below s_i, and receives a replenishment to S_i whenever another customer triggers a replenishment and i's inventory level is no greater than c_i.

By omitting the above simplifications, as we do in the DDP, one might expect to encounter the following difficulties, were one to try to establish good customer (S,c,s) policies with a decomposition approach:

1) The replenishment cost share borne by a customer in the single-customer subproblems would be impossible to assess accurately, because it depends not only on the number of customers in a replenishment, but also on their identities, which change from replenishment to replenishment.

2) There are no guarantees that there will be sufficient fleet capacity available to deliver to every customer the order dictated by its optimal inventory policy.

On the other side of the coin, it is doubtful that pure (S,c,s) policies would serve well the physical distribution operation as we have modeled it, because who gets combined on a single replenishment dispatch should be a function of customer location as well as of inventory levels. Even when the role of decomposition is limited to the task of estimating the future consequences of current dispatches, the complications cited above will hamper any effort to apply decomposition in an intelligent way. We try to proceed as best we can. Our efforts are documented below.

4.3.2 Overview of the FVD Algorithm

In Section 4.1, the search for the best decision in a particular state of a Markov decision problem was likened to solving an "augmented" finite horizon problem (the end of the horizon being the next decision point). The augmentation consisted of a term in the objective function to penalize the tactic of minimizing costs during the horizon at the expense of leaving the system in a bind at the end of the horizon. The consistent application of such penalty functions yields decisions which benefit the system over the long run, even though it is only finite horizon problems that are being solved. This penalty function device drives the heuristic algorithm we propose for the deliverer dispatch problem.

If we were in possession of them, we would surely use the expected future values derived from the optimal solution of the corresponding Markov decision problem as penalty functions, for reasons stated near the end of Section 4.1. But Section 4.2 makes clear the infeasibility of obtaining an exact set of expected future values. We pursue instead an approximate penalty determination method, where the exact expected future values are the quantities we attempt to approximate. The fundamental approximating assumption of this method is that all decisions and costs are separable by customer. In other words, we assume that the future performance of the system as a whole can be forecasted accurately by studying customers individually. It is true that the inventory costs a customer endures, given its inventory state and the delivery quantity commissioned at a decision point, may be calculated independently of inventory costs for other customers. But the transportation cost of a dispatch decision for the entire system cannot usually be inferred by considering each customer's delivery quantity in isolation of what the other customers are having delivered. Moreover, it may be impossible to locate a feasible dispatch that meets the delivery quantities set for each customer at a decision point, due to vehicle availability factors. It seems inevitable, then, that systemwide cost forecasts obtained by decomposition of the original system into single-customer systems will prove inaccurate. Our hope is that, if the expectation of costs incurred in the finite horizon is always calculated exactly and approximation by decomposition is only utilized for the forecast of future activity required for penalization purposes, the quality of guidance supplied by the resultant penalty factors will diminish only slightly with respect to that offered by the optimal penalties. In recognition of the role that decomposition plays in the proposed heuristic, we label our algorithm the "Future Value Decomposition" (FVD) algorithm.

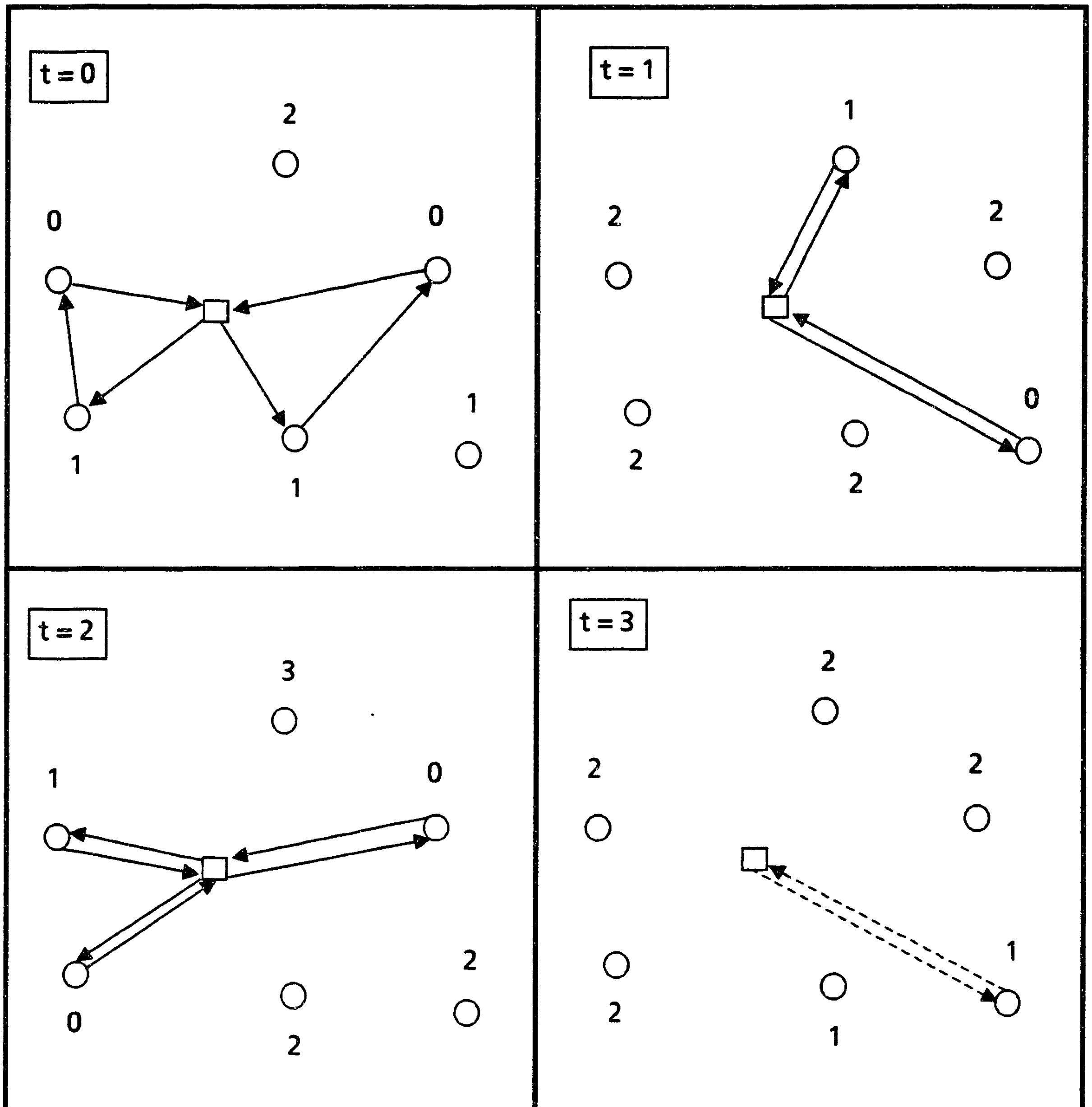
The implementation of the decomposition principle in an heuristic solution scheme for the DDP transmutes the formulation of the DDP, from the perspective of the algorithm, at future decision points. Dispatches at all decision points beyond the current one draw from a different set of itineraries. This new set is composed of itineraries which send vehicles to one customer only before they return to the depot. The characteristics of each itinerary in the new set "resemble" those of itineraries in the original set that make the same delivery to the same customer. Leadtimes are averaged over this subset of itineraries. The resemblance with respect to transportation costs is one based on separability; the cost of a multi-customer itinerary in the original problem is as near as possible to the total cost of the subset of itineraries in the new set producing the same aggregate delivery. Additionally, all future dispatches are feasible regardless of the state of vehicle availability as it would be determined under the original DDP.

However, a dispatch failure phenomenon may be instituted, which may cause any future itinerary intended for execution to be canceled with some probability. Vehicle unavailability is represented indirectly by this phenomenon. Separability (with respect to costs and decisions) of the transmuted DDP is insured in the presence of the foregoing mutations. Yet when each problem starts with the same customer inventory levels, the transient cost differences incurred in each DDP *may* agree approximately if the forms of the mutations are chosen judiciously. The viability of the FVD algorithm hinges on such agreement.

To further cultivate the understanding of the approximation mechanism that our algorithm employs, we refer to the illustrations of Exhibit 4.2. The exhibit consists of four "pictures" of the distribution region, the first taken at the current decision point and the others at the next three decision points in the future. These pictures are meant to convey some sense of what the algorithm may "see" as it peers into the future to judge what a good dispatch would be now. Overlayed in each picture are routes corresponding to itineraries permissible in whichever DDP is presumed to be in effect at that decision point (original for the current decision point, transmuted for the future ones), and inventory levels for each customer arising from the initial set of inventory levels, the displayed itineraries, and one set of future customer demands. A route composed of a dashed line corresponds to an itinerary that has been canceled. All intended replenishment routes for the transmuted DDP are of the "hub-and-spoke" variety--vehicles visit one customer and return. A complete pictorial representation of what the algorithm "looks at" in obtaining the current decision would need to show all possible combinations of current and future decisions and customer demands, and would have to extend toward $t=\infty$ rather than $t=3$. But the exhibit does serve to illustrate what the FVD algorithm perceives to be possible evolutionary paths for the system.

Exhibit 4.3 displays two evolutionary paths for the inventory level of a selected customer. The first path may be potentially observed in actual operations. The second one shows how the inventory level would respond to the same set of demands under the assumptions of the FVD

Exhibit 4.2: Potential System Evolution in the Mind of the FVD Algorithm

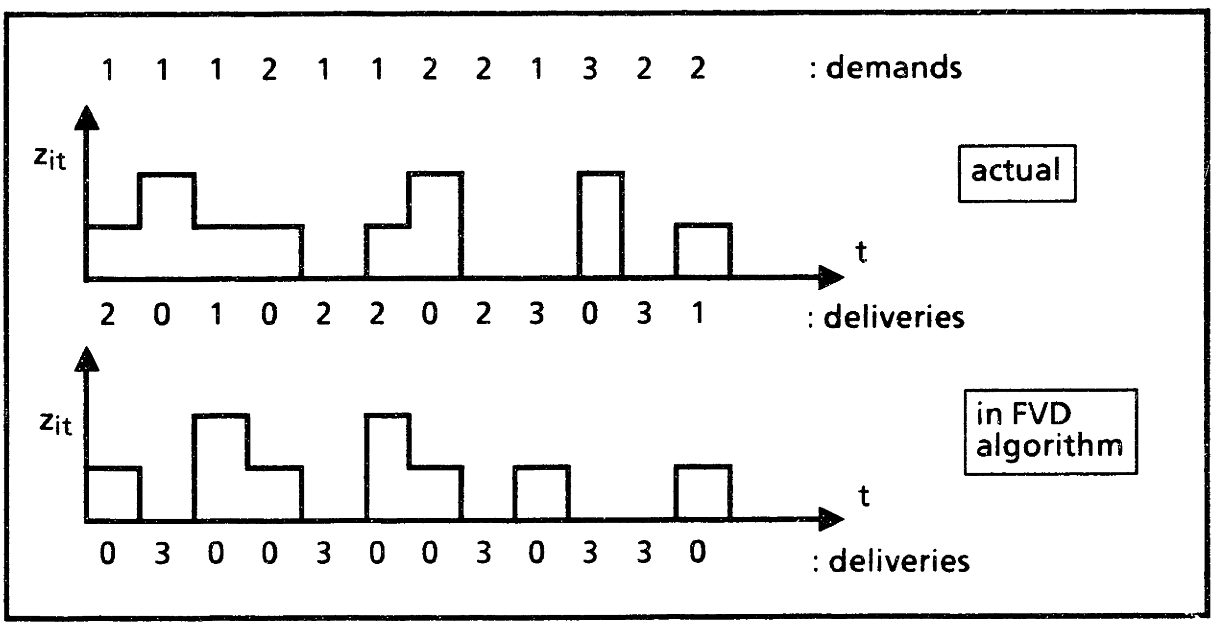


Nodes labeled by inventory level (actual or result of assumed demand quantities)
at decision point
solid line = executed itinerary
dashed line = canceled itinerary

algorithm. One implication of the separability property is that each customer operates under a replenishment policy depending only on that customer's inventory state. Assuming that the customer's optimal replenishment policy is to receive 3 units when the inventory level is zero and to receive nothing at any other level, then all delivery quantities except possibly the first must be either 0 or 3. The first quantity is determined by the solution of the augmented finite horizon problem, when any of the original itineraries may be executed. Since itineraries from the original set are executed at every decision point in the actual operation, the delivery to the customer in the first path may be any quantity, provided that there exists an itinerary directing that delivery to the customer in question.

The prediction of future system activity in the transmuted DDP can be achieved with little difficulty, owing to the separability property. What each customer has delivered to it, and what the systemwide cost consequences are, can be determined on a customer-by-customer basis in the transmuted DDP. This determination is accomplished through the formulation and solution of single-customer DDP-like subproblems, which we label the "customer DDSPs" (DDSP=deliverer dispatch subproblem). The subproblems do not strictly conform to the DDP mold established through the specializations of Sections 3.2.1-3.2.6, but their structure is similar enough and their state spaces small enough so that they all can be solved to optimality using Markov decision-theoretic techniques. The MDP solutions provide the approximate expected future values for the

Exhibit 4.3: Potential Evolutionary Paths in Actuality and in Heuristic, One Customer's Inventory Process



penalty function, since the approximations are sums over expected future values by customer, which are generated in the course of solving the DDSPs. The finite horizon dispatch selection problem augmented by the penalty function is named the "master scheduling problem"; its solution determines the final dispatch.

The rest of Section 4.3 describes the FVD algorithm in more detail. The reader might want to skim this material initially, study the implementation of the algorithm on a small numerical example in the next chapter, then come back and read this material carefully.

4.3.3 The Single-Customer Deliverer Dispatch Subproblems

A customer's deliverer dispatch subproblem is intended to represent that portion of the entire system's delivery operation affecting the customer. In the DDSP, customer i is entitled to order whatever quantity of goods d_d it wants, provided that some itinerary j exists in the original DDP that makes that delivery ($d_{ij} = d_d$) to that customer, and that no previous order has not been delivered (time since last order is greater than leadtime of replenishment); the latter measure is taken to reduce the complexity of the problem. However, the customer may find the replenishment canceled immediately with the dispatch failure probability F.

The attributes of a replenishment to a customer are determined by averaging the respective attributes of certain itineraries in the original problem. This is done in an attempt to approximate some of the effects of joint replenishment. Let J_{id} be the set of itineraries j delivering d units to customer i:

$$J_{id} = \left\{ j \mid d_{ij} = d \right\}. \tag{4.13}$$

The leadtime for the delivery of d units in customer i's DDSP is taken as the average of leadtimes to i of itineraries in J_{id}:

$$l_{id} = \left\lceil \frac{\sum_{j \in J_{id}} l_{ij}}{|J_{id}|} \right\rceil, \tag{4.14}$$

where $[x]$ is the nearest integer to x. The replenishment cost C_{id} of such a delivery is based on an average of "cost shares," or transportation costs prorated according to the proportion of the total delivery in itinerary j going to customer i:

$$C_{id} = \frac{\sum_{j \in J_{id}} \frac{d_{ij}}{\sum_{i'=1}^{m} d_{i'j}} C_j}{|J_{id}|}. \tag{4.15}$$

We formulate the single-customer deliverer dispatch subproblem for customer i as follows:

State: $s_i = (z_i,\ y_i,\ t_{yi})$, where z_i is the customer's current inventory level, y_i the delivery outstanding, and t_{yi} the number of periods until the arrival of the outstanding delivery. S_i is the state space for the problem.

Decision: d_d, the delivery quantity desired to be dispatched at the decision point. Note: d_d may be chosen positive only when $y_i = 0$. The actual delivery d_a will equal d_d with probability $1 - F$ and 0 otherwise.

Evolution: Let n' denote the value of quantity n at the next decision point. Then:

$$z'_i = min\left[u_i,\ max\left(0, z_i - x_i + y_i\right)\right]\ if\ t_{yi} = 0\ and\ y_i > 0, \tag{4.16a}$$

$$= min\left[u_i,\ max\left(0, z_i - x_i + d_a\right)\right]\ if\ d_a > 0\ and\ l_{id_a} = 0, \tag{4.16b}$$

$$= min\left[u_i,\ max\left(0, z_i - x_i\right)\right]\ otherwise; \tag{4.16c}$$

$$y'_i = y_i \quad if\ t_{yi} > 0, \tag{4.17a}$$

$$= d_a \quad otherwise; \tag{4.17b}$$

$$t'_{yi} = t_{yi} - 1 \quad if\ t_{yi} > 0, \tag{4.18a}$$

$$= l_{id_a} - 1 \quad if\, d_a > 0 \; and \; l_{id_a} > 0, \qquad (4.18b)$$

$$= 0 \quad otherwise; \qquad (4.18c)$$

where $x_i = x_0$ (demand) with probability $p_i(x_0)$, $x_0 = 0,1,2,\ldots$. Let

$$_i^F p_{s_i s'_i}^{d_a}$$

be the probability of moving from $s_i = (z_i, y_i, t_{yi})$ to $s'_i = (z'_i, y'_i, t'_{yi})$ over one period, given decision d_a and failure probability F.

Performance: Let the number of lost demands e_i be found by

$$e_i = max\left(0, x_i - z_i - y_i\right) \quad if\, t_{yi} = 0 \; and \; y_i > 0, \qquad (4.19a)$$

$$= max\left(0, x_i - z_i - d_a\right) \quad if\, d_a > 0 \; and \; l_{id_a} = 0, \qquad (4.19b)$$

$$= max\left(0, x_i - z_i\right) \quad otherwise; \qquad (4.19c)$$

then the performance of decision d_d in state s_i is the immediate cost

$$_i^F q_{s_i}^{d_d} = E_{x_i, d_a}\left[C_{id_a} + H_i z'_i + L_i e_i\right]. \qquad (4.20)$$

Objective: Select dispatch policy δ_i to minimize

$$_i g^{\delta_i} = \sum_{s_i \in S_i} {}_i\pi_{s_i}^{\delta_i} \times {}_i^F q_{s_i}^{\delta_i(s_i)}, \qquad (4.21)$$

where

$$_i\pi_{s_i}^{\delta_i}$$

is the steady-state probability of being in state s_i under dispatch policy δ_i, given by

$$\sum_{s_i \in S_i} {}_i\pi_{s_i}^{\delta_i} \times {}_i^F p_{s_i s'_i}^{\delta_i(s_i)} = {}_i\pi_{s'_i}^{\delta_i}, \quad s'_i \in S_i, and \qquad (4.22a)$$

$$\sum_{s_i \in S_i} {}_i\pi_{s_i}^{\delta_i} = 1. \qquad \textbf{(4.22b)}$$

Using the policy iteration algorithm to solve the resulting Markov decision problems, we will obtain for later use the following quantities for each customer i:

* the optimal policy δ_i* providing the replenishment order or delivery target for each inventory level;

* the single-customer expected future values

$$\sum_{s'_i \in S_i} {}^0_i p^d_{s_i s'_i} \times {}_i v_{s'_i}^{\delta^*_i}. \qquad \textbf{(4.23)}$$

We can also reduce some of the computational effort of the master scheduling problem if we save the readily available quantities

$$_iQ^d_{s_i} = {}^0_i q^d_{s_i} - C_{id}, \qquad \textbf{(4.24)}$$

i.e., the inventory cost incurred in the upcoming period, if dispatches cannot fail. Call (4.24) the *immediate failsafe inventory cost.*

We justify our approximations as follows: Averaging seems to us a reasonable basis, in lieu of auxiliary itinerary dispatch frequency information, for allocating the cost of an itinerary to the customers it services. C_{id} is simply the average of this cost "share" over itineraries delivering d to customer i. (Dror and Ball [9, p. 22] describe a different method for a related cost allocation task.) Likewise, leadtime is averaged over the itineraries in J_{id}. We presume that there is always one vehicle available for dispatch; finite vehicle availability is represented via the dispatch failure probability F. The failure probability should be set to the systemwide proportion of delivery targets missed, although this quantity is unknown *a priori*. Iterative refinement or calibration of an initial estimate of the failure probability may be nonetheless possible by either simulation or queueing-type analysis. It may improve systemwide dispatching performance to condition the failure probability on the identity of the customer and on the delivery size, but we do not do so here. Also, the question of what constitutes a dispatch "failure" should be addressed. It is possible for a customer to receive a dispatch of a positive quantity that is not its delivery target; this will not be regarded as a "failure."

4.3.4 The Master Scheduling Problem

The *master scheduling problem* (MSP) for state s is to find the immediate dispatch $\gamma = \{y_{jk}\}$ that minimizes the sum of the immediate cost and the expected penalty for the state entered at the next decision point. The penalty function used is the expected future value in the transmuted DDP (see Section 4.3.2). By construction of the transmuted DDP, this penalty is equivalent to the sum of expected future values for each customer from its DDSP when the replenishment to the customer equals the total delivery to it over itineraries selected in γ. This function is intended to approximate the actual expected value for dispatch γ in state s. The validity of the penalty function is predicated on the assumption that good future dispatches for the system as a whole will yield total costs close to the sum of those each customer would endure using its optimal replenishment policy from the single-customer operation modeled by its DDSP. The MSP receives treatment in greater depth below.

The proper objective function for the MSP would be the improvement criterion of relation (4.12). While the immediate cost term of the objective function can be calculated readily, the expected future value term cannot. Recall from Section 3.3 that the immediate cost $q_s{}^\gamma$ in the original DDP for decision γ in state s is given by

$$q_s^\gamma = \sum_{j=1}^{n} C_j \sum_{k=1}^{K} y_{jk,0} + E_{\{x_i\}} \sum_{i=1}^{m} \left(H_i z_i + L_i e_i \right),$$

with all terms as defined there. It can be easily verified, using the notation in the last section, that

$$q_s^\gamma = \sum_{j=1}^{n} C_j \sum_{k=1}^{K} y_{jk,0} + \sum_{i=1}^{m} {}_iQ_{s_i}^{d_i}, \tag{4.25}$$

provided that γ is feasible and s attainable vis-a-vis the single-customer DDSP stipulation that no more than one order per customer may remain outstanding at any point in time, the customer states s_ι are consistent with the master state s, and the delivery to customer i

$$d_i = \sum_{k=1}^{K} \sum_{j=1}^{n} d_{ij} y_{jk,0}.$$

Our formulation of the master scheduling problem addresses the infeasibility of evaluating the expected future value term exactly. Let δ^+ be the dispatch policy obtained by use of the FVD algorithm, and $\{v_s^{\delta^+}\}$ its relative values. Then consider the expected future value for any state s and decision γ in the original problem, under the condition that dispatch policy δ^+ is used at all future decision points. This is the second term in (4.12), the expression we take as the MSP's objective function. Our fundamental assumption is that this expected future value can be approximated by

$$\sum_{s'=1}^{S} p^{\gamma}_{ss'} v^{\delta^+}_{s'} \approx \sum_{\iota=1}^{m} \left[\sum_{s'_i \in S_i} {}^{0}_{\iota}p^{d_i}_{s_i s'_i} \; {}_{\iota}v^{\delta^*_i}_{s'_i} \right]. \tag{4.26}$$

This is the mathematical translation of the assumption made at the beginning of this subsection for the validity of the decomposition approach. Therefore, δ^+ is the dispatch policy with the characteristic that the dispatch $\delta^+(s)$ minimizes the MSP's objective function

$$q_s^{\delta^+(s)} + \sum_{\iota=1}^{m} \left[\sum_{s'_\iota \in S_\iota} {}^{0}_{\iota}p^{d_\iota}_{s_\iota s'_\iota} \; {}_{\iota}v^{\delta^*_\iota}_{s'_\iota} \right] \tag{4.27}$$

($\{d_i\}$ derived from $\delta^+(s)$) for every state s. The master scheduling problem seeks, for any given s, the best feasible dispatch $\delta^+(s)$.

Let the decision variable y_{jk} equal one if and only if vehicle k is dispatched on itinerary j at the current decision point. We can express the master scheduling problem as:

$$min \sum_{j=1}^{n} C_j \sum_{k=1}^{K} y_{jk} + \sum_{\iota=1}^{m} \left[{}_{\iota}Q^{d_i}_{s_\iota} + \sum_{s'_\iota \in S_\iota} {}^{0}_{\iota}p^{d_i}_{s_\iota s'_\iota} \; {}_{\iota}v^{\delta^*_\iota}_{s'_\iota} \right] \tag{4.28}$$

$$s.t. \sum_{j=1}^{n} y_{jk} \le 1 - sgn(w_k) \quad \forall\, k \tag{4.29}$$

$$\sum_{j=1}^{n} \sum_{k=1}^{K} d_{ij}^{1} y_{jk} \leq 1 - \sum_{k=1}^{K} \sum_{j=1}^{n} \sum_{l=-l_{ij}}^{-1} d_{ij}^{1} y_{jkl} \quad \forall\, i \tag{4.30}$$

$$d_i - \sum_{k=1}^{K} \sum_{j=1}^{n} d_{ij} y_{jk} = 0 \quad \forall\, i \tag{4.31}$$

$$y_{jk} = 0 \; or \; 1. \tag{4.32}$$

where

$$d_{ij}^{1} = 1 \quad if\, d_{ij} > 0, \tag{4.33a}$$

$$= 0 \; otherwise. \tag{4.33b}$$

Constraints (4.29) permit only available vehicles to be dispatched. Constraints (4.30) further restrict dispatches to only those customers with no outstanding deliveries, and assure that this condition will be maintained in the upcoming period by allowing each customer to fall on at most one itinerary in the current dispatch. This condition may detract from the quality of the resultant dispatches in some applications, but it allows certain key simplifications in the structure of the MSP (see below). The total delivery d_i to customer i is expressed as a function of the dispatch in (4.31).

Several simplifications can be introduced to the MSP (4.28)-(4.32). Since all vehicles are specified identical in Section 3.2.6, we may replace the dispatch decision $\{y_{jk}\}$ with $\{y_j\}$, where

$$y_j = \sum_{k=1}^{K} y_{jk}. \tag{4.34}$$

Constraints (4.29) can be summed over k, and substitutions of y_j and y_{jl} can be conducted in constraints (4.30) and the objective function to simplify the MSP statement. Also consider the objective function (4.28). If customer i is not in any itinerary j where $y_j = 1$, then $d_i = 0$ and the term for i in the sum over customers in the (4.28) is

$${}_tQ^0_{s_i} + \sum_{s'_t \in S_t} {}_tp^0_{s_t s'_t} \, {}_tv^{\delta^*_t}_{s'_t}.$$

If customer i is in dispatched itinerary j (and therefore, by constraint (4.30), not in any other one), then the corresponding term is

$$ {}_iQ_{s_i}^{d_{ij}} + \sum_{s'_i \in S_i} {}_ip_{s_i s'_i}^{0\,d_{ij}}\, {}_iv_{s'_i}^{\delta^*_i}. $$

Now we introduce some notation to simplify the form of the MSP. First, suppose we were to start with the solution $y_j = 0$ for all j (a solution that is always feasible). Then the solution $y_j = 1, y_{j'} = 0$ for $j' \neq j$ improves on the no-dispatch solution if and only if

$$ C_j + \sum_{i=1}^{m} \left\{ {}_iQ_{s_i}^{d_{ij}} + \sum_{s'_i \in S_i} {}_ip_{s_i s'_i}^{0\,d_{ij}}\, {}_iv_{s'_i}^{\delta^*_i} \right\} < \sum_{i=1}^{m} \left\{ {}_iQ_{s_i}^{0} + \sum_{s'_i \in S_i} {}_ip_{s_i s'_i}^{0\,0}\, {}_iv_{s'_i}^{\delta^*_i} \right\}. \tag{4.35} $$

We define the *dispatch cost* C_j^+ of itinerary j by

$$ C_j^+ = C_j + \sum_{i=1}^{m} \left\{ {}_iQ_{s_i}^{d_{ij}} + \sum_{s'_i \in S_i} {}_ip_{s_i s'_i}^{0\,d_{ij}}\, {}_iv_{s'_i}^{\delta^*_i} - {}_iQ_{s_i}^{0} - \sum_{s'_i \in S_i} {}_ip_{s_i s'_i}^{0\,0}\, {}_iv_{s'_i}^{\delta^*_i} \right\}. \tag{4.36} $$

Then dispatching j improves upon not dispatching at all if $C_j^+ < 0$, and the more that C_j^+ is negative, the greater the improvement.

If we reexpress the transportation cost C_j in terms of C_j^+, substitute into the objective function, and rearrange terms, we get the equivalent objective function

$$ \min \sum_{j=1}^{n} C_j^+ y_j + \sum_{i=1}^{m} \left\{ \left[{}_iQ_{s_i}^{d_i} + \sum_{s'_i \in S_i} {}_ip_{s_i s'_i}^{0\,d_i}\, {}_iv_{s'_i}^{\delta^*_i} \right] \right. $$

$$ \left. - \sum_{j=1}^{n} \left[{}_iQ_{s_i}^{d_{ij}} + \sum_{s'_i \in S_i} {}_ip_{s_i s'_i}^{0\,d_{ij}}\, {}_iv_{s'_i}^{\delta^*_i} - {}_iQ_{s_i}^{0} - \sum_{s'_i \in S_i} {}_ip_{s_i s'_i}^{0\,0}\, {}_iv_{s'_i}^{\delta^*_i} \right] y_j \right\}. \tag{4.37} $$

Remember that y_j may equal one for at most one j for which $d_{ij}1 = 1$. Consider the expression summed over i in the objective function. If, for customer i, no itinerary j with $y_j = 1$ visits i, then $d_i = 0$, the coefficients of all y_j's equaling 1 are zero, and the expression has value

$$ {}_iQ_{s_i}^{0} + \sum_{s'_i \in S_i} {}_ip_{s_i s'_i}^{0\,0}\, {}_iv_{s'_i}^{\delta^*_i}. $$

Otherwise, let j be the itinerary with $y_j = 1$ and $d_{ij}1 = 1$. Then $d_i = d_{ij}$ and it can be verified once again that the expression in the sum over i reduces to

$${}_iQ^0_{s_i} + \sum_{s'_i \in S_i} {}^0_i p^0_{s_i s'_i} \, {}_i v^{\delta^*_i}_{s'_i}.$$

The objective function thus is equivalent to

$$\min \sum_{j=1}^{n} C_j^+ y_j + \sum_{i=1}^{m} \left[{}_iQ^0_{s_i} + \sum_{s'_i \in S_i} {}^0_i p^0_{s_i s'_i} \, {}_i v^{\delta^*_i}_{s'_i} \right]. \quad (4.38)$$

Since the latter term in (4.38) does not depend on y_j, it is equivalent to minimize

$$\sum_{j=1}^{n} C_j^+ y_j.$$

We utilize these simplifications to form the simpler MSP

$$\min \sum_{j=1}^{n} C_j^+ y_j \quad (4.39)$$

$$s.t. \sum_{j=1}^{n} y_j \le A \quad (4.40)$$

$$\sum_{j=1}^{n} d^1_{ij} y_j \le 1 - \sum_{j=1}^{n} \sum_{l=-l_{ij}}^{-1} d^1_{ij} y_{jl} \quad \forall i \quad (4.41)$$

$$y_j = 0 \, or \, 1, \quad (4.42)$$

where the number of available vehicles

$$A \equiv \sum_{k=1}^{K} 1 - sgn(w_k). \quad (4.43)$$

The MSP (4.39)-(4.42) is observed to be equivalent to an $(m+1)$-dimensional (0,1)-knapsack problem. With constraint (4.40) omitted, the MSP is of the form of a set packing problem ([22, p. 407]). The MSP is also similar to certain routing problems studied by Cullen, Jarvis, and Ratliff [7]. Several classes of procedures (e.g., branch and bound) have been developed to find optimal solutions for these types of problems. If the MSP is large or a rapid and not necessarily optimal solution is desired, one may solve the MSP heuristicaly. Heuristic methods that have traditionally been applied to problems resembling the MSP include linear programming and

Lagrangian relaxation-based methods and other types of procedures. In our computational work, we have used a different type of heuristic, which we describe later.

The set of itineraries to consider may be reduced to further simplify the MSP. Let I be a subset of the customer set $\{1,\ldots,m\}$ and let $J(I)$ be the set of itineraries j such that $d_{ij}^1 = 1$ if $i \in I$ and $d_{ij}^1 = 0$ if $i \notin I$. $J(I)$ is thus the subset of itineraries visiting exactly the customer subset I. The itinerary j_I^* with the minimum dispatch cost among all itineraries in $J(I)$ *dominates* all other itineraries in $J(I)$ in this sense: Suppose we switch from dispatching j_I^* to dispatching some other itinerary j in $J(I)$ in some MSP solution . All constraints are still respected but the objective cost increases (gets worse). Also, no improvement in the solution can be made by exchanging dispatched and undispatched itineraries that could not be made with j_I^* in the solution. Hence, we need only consider in the MSP itineraries j that are undominated among all itineraries delivering to the same customer subset.

An heuristic algorithm for the solution of the master scheduling problem is offered through Exhibits 4.4, 4.5, and 4.6. The BEST_ITIN procedure (Exhibit 4.4) takes each undominated feasible itinerary j as a seed itinerary and approximates the objective cost of the best dispatch that includes the seed itinerary via an embedded greedy heuristic (GREEDY_ASSIGN in Exhibit 4.5) for dispatching the unassigned vehicles. The seed yielding the lowest objective cost, provided that it is negative (represents an improvement over not dispatching), is then fixed for dispatch, and if any vehicles remain, the process is repeated. The main procedure given in Exhibit 4.6 controls the execution of this heuristic. Since all feasible itinerary singletons and pairs are examined in the algorithm, the MSP is solved optimally via the heuristic for $A \leq 2$. The quality of MSP solutions for $A > 2$ is unknown as yet. Further research into MSP solution procedures is called for.

Exhibit 4.4: BEST_ITIN Procedure from MSP Heuristic

```
    procedure BEST_ITIN(I_O, k)
    begin
1.    MIN_OBJECT_COST←0
2.    MIN_ITIN←0
3.    for each undominated itinerary j do
        begin
4.        if {customers in j} ∩ I_O = ∅ then
            begin
5.            OBJECT_COST←C_j^+
6.            for each customer i do
                I_T ← I_O ∪ {customers in j}
7.            GREEDY_ASSIGN(I_T, OBJECT_COST, k)
8.            if OBJECT_COST < MIN_OBJECT_COST then
                begin
9.                MIN_OBJECT_COST←OBJECT_COST
10.               MIN_ITIN←j
                end
            end
        end
11.   return MIN_ITIN
    end
```

4.3.5 Summary of the FVD Algorithm

The FVD algorithm may be considered to consist of a "preprocessor" and a "dispatcher." The preprocessor solves a set of DDP-like Markov decision problems, one for each customer, in advance of the onset of system evolution. The key information derived from these solutions is not the set of single-customer dispatch policies, but the cost information generated as a byproduct of the solution methodology. The dispatcher solves a master scheduling problem at each decision point to obtain the dispatch for that decision point. The MSP is a multidimensional knapsack problem, and the instance examined at any particular decision point is dependent on the state of

Exhibit 4.5: GREEDY_ASSIGN Procedure from MSP Heuristic

```
     procedure GREEDY_ASSIGN(I_T, OBJECT_COST, k)
     begin
1.     for k'←k+1 until A do
         begin
2.         find itinerary j* with minimum C_j'^+ among itineraries j' with
             {customers in j'} ∩ I_T = ∅
3.         if no such itinerary j* exists then return
4.         OBJECT_COST←OBJECT_COST + C_j*^+
5.         I_T ← I_T ∪ {customers in j*}
         end
6.     return
     end
```

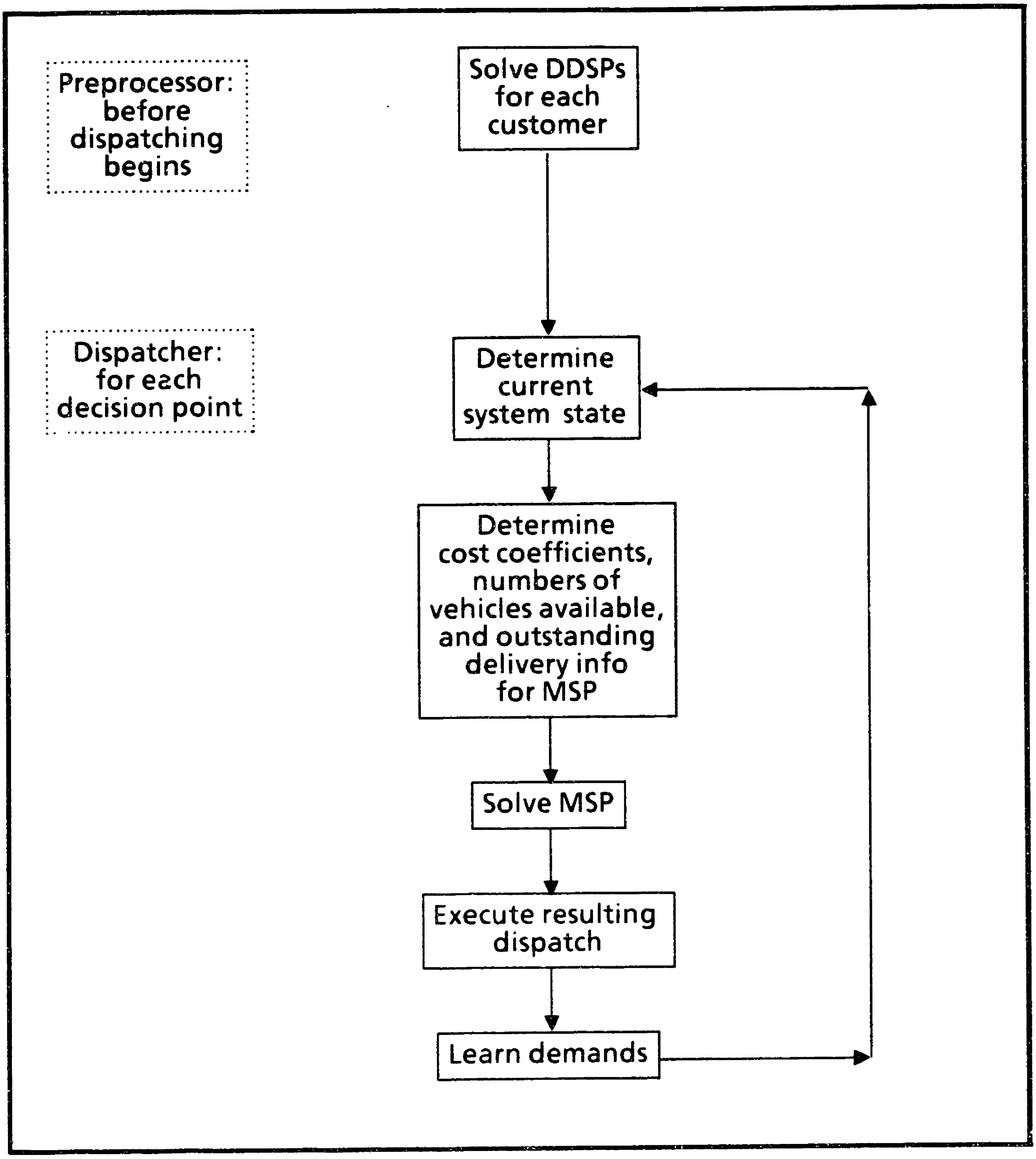
Exhibit 4.6: Main Procedure from MSP Heuristic

```
     begin
1.     I_O ← set of all customers with outstanding deliveries
2.     for each available vehicle k do DISPATCH(k)←0
3.     k←0
4.     repeat
         begin
5.         k←k+1
6.         DISPATCH(k)←BEST_ITIN(I_O, k)
7.         I_O ← I_O ∪ {customers in DISPATCH(k)}
         end
8.     until k=A or DISPATCH(k)=0
     end
```

the system at that point. Exhibit 4.7 presents a flowchart that shows how the delivery system functions when using the FVD algorithm to generate dispatches.

We may gain some insight about how the FVD algorithm approximates the policy iteration algorithm by seeking some form of correspondence in their respective phases. The FVD

Exhibit 4.7: Implementation of FVD Algorithm in Delivery System



algorithm may be considered a one-iteration policy iteration algorithm. Value determination in the FVD algorithm is performed by the preprocessor, although it is expectations of the relative values, rather than the relative values themselves, which are determined. The value determination process has embedded within it the exact solution of small Markov decision problems, so that by using the policy iteration algorithm to solve them, iteration goes on ***within*** value determination for the original DDP. Policy improvement in the FVD algorithm works much as does its counterpart in policy iteration, with approximations employed for the expected future values. Because there is no iteration in the FVD algorithm, the search for the best dispatch at a given state may be postponed until that state is entered during the evolution of the vehicle dispatching system.

4.3.6 Properties of the FVD Algorithm

In this section we establish a potentially useful lower bound property of the decomposition algorithm applied to a slightly more specialized version of the DDP. We also investigate the computational complexity of the algorithm.

Consider the DDP we have been studying with the additional stipulation that all leadtimes are zero. Further, restrict the maximum total delivery allowable to any customer during a period to be one full vehicle load Ω.

Proposition 1. Suppose that the number of vehicles K equals or exceeds the number of customers times the maximum itinerary duration, and that the itinerary set $\{j\}$ has the properties that there exists a itinerary j_{id} that delivers only d units to customer i, $d=1,\dots,\Omega$ and $i=1,\dots,m$, and that

$$C_j = \sum_{i=1}^{m} C_{i,d_{ij}} \quad \forall\, j, \tag{4.44}$$

where the replenishment cost shares C_{id} are strictly convex in the sense that

$$C_{id_1} + C_{id_2} > C_{i,d_1+d_2} \quad \forall\, d_1, d_2 > 0.$$

Then the policy δ^D obtained via the decomposition algorithm using cost shares C_{id} and zero dispatch failure probability is optimal for the original DDP.

Proof. Basically, the foregoing assumptions cause any resource constraints and economies of scale to be removed, so each customer's replenishment problem can be treated separately, as we do in solving the DDSPs. In case this argument is not satisfactory to the reader, a more rigorous demonstration of the validity of the proposition is given below. It expresses the same argument mathematically. Also, the proofs of two further propositions extend in part from this proof.

We need to show that

$$q_s^{\delta^D(s)} + \sum_{s'=1}^{S} p_{ss'}^{\delta^D(s)} v_{s'}^{\delta^D} \le q_s^{\gamma} + \sum_{s'=1}^{S} p_{ss'}^{\gamma} v_{s'}^{\delta^D} \tag{4.45}$$

for every state s and decision γ feasible when the system is in state s. We may reduce the information contained in state s to the set of customer inventory levels, since the zero leadtimes assumption makes previous dispatch information valueless, and the large vehicle fleet size and the customer delivery limit assure that no desired and permissible dispatch will fail due to vehicle unavailability. So let

$$s=(z_1, \ldots, z_m)$$

and

$$s'=(z'_1, \ldots, z'_m).$$

First we demonstrate that

$$q_s^{\delta^D(s)} + \sum_{s'=1}^{S} p_{ss'}^{\delta^D(s)} v_{s'}^{\delta^D} = \sum_{i=1}^{m} \left[{}_iq_{z_i}^{\delta^*_i(z_i)} + \sum_{z'_i=0}^{u_i} {}_ip_{z_i z'_i}^{\delta^*_i(z_i)} \, {}_iv_{z'_i}^{\delta^*_i} \right], \tag{4.46}$$

where δ_i^* is the optimal policy from customer i's DDSP. (Since the dispatch failure probability F will always be zero, the F superscripts are dropped from the notation.) Keep in mind that the decomposition policy δ^D is obtained via application of the master scheduling problem. For the given specialized DDP, the MSP becomes.

$$min \sum_{j=1}^{n} \left\{ \sum_{i=1}^{m} \left[C_{i,d_{ij}} + {}_iQ_{z_i}^{d_{ij}} + \sum_{z'_i=0}^{u_i} {}_ip_{z_i z'_i}^{d_{ij}} \, {}_iv_{z'_i}^{\delta^*_i} - {}_iQ_{z_i}^{0} - \sum_{z'_i=0}^{u_i} {}_ip_{z_i z'_i}^{0} \, {}_iv_{z'_i}^{\delta^*_i} \right] \right\} y_j \tag{4.47}$$

$$s.t. \sum_{j=1}^{n} d^{1}_{ij} y_j \leq 1 \quad \forall i \tag{4.48}$$

$$y_j = 0 \; or \; 1 \tag{4.49}$$

The objective function is obtained primarily by the substitution of (4.44) into (4.36). The constraint (4.40) disappears due to the ample vehicle supply.

Lemma. There exists an optimal solution $\{y_j^*\}$ to the MSP (4.47) - (4.49) with

$$\sum_{i=1}^{m} d^{1}_{ij} = 1$$

for all j with $y_j^* = 1$ (i.e., all selected itineraries visit one customer only).

Proof. Suppose one optimal solution $\{y_j^*\}$ for the MSP has some itinerary j' for which $y_{j'}^* = 1$ and $\Sigma_i \, d_{ij'}^{1} = M > 1$. Consider the solution $\{y_j^{**}\}$ identical to this $\{y_j^*\}$ except that itinerary j' is replaced in the dispatch by single-customer itineraries making the same aggregate delivery. Let i_k, $k = 1,\ldots,M$, be the customers in itinerary j' and $\{j_k\}$ the single-customer itineraries where j_k delivers $d_{i_k j'}$ to i_k. The itineraries $\{j_k\}$ exist by assumption, and

$$C_{j'} \cdot y^*_{j'} = \sum_{i=1}^{m} C_{i,d_{ij'}} = \sum_{k=1}^{M} \sum_{i=1}^{m} C_{i,d_{ij_k}} \, y^{**}_{j_k} = \sum_{k=1}^{M} C_{j_k} \, y^{**}_{j_k} .$$

The objective value remains the same after this alteration to the solution, and constraint (4.48) is still respected, hence $\{y_j^{**}\}$ is also optimal. If we continue to make these substitutions involving the $\{y_j\}$, we will eventually arrive at an optimal solution in which all executed itineraries visit single customers only.□

By the preceding lemma, the following MSP variant has the same objective value as the original:

$$min_{\{d_i\}} \sum_{i=1}^{m} \left[C_{i,d_i} + {}_iQ^{d_i}_{z_i} + \sum_{z'_i=0}^{u_i} {}_iP^{d_i}_{z_i z'_i} \, {}_iv^{\delta^*_i}_{z'_i} - {}_iQ^{0}_{z_i} - \sum_{z'_i=0}^{u_i} {}_iP^{0}_{z_i z'_i} \, {}_iv^{\delta^*_i}_{z'_i} \right] . \tag{4.50}$$

The optimal solution of delivery quantities $\{d_i^*\}$ to this problem yields one optimal solution of dispatches $\{y_j^*\}$ to the preceding MSP, when $y_j^* = 1$ if and only if $d_{ij} = d_i^*$ and $d_{i'j} = 0$ for $i' \neq i$

(assuming there is only one single-customer itinerary j for which $d_{ij}=d_i^*$). Problem (4.50) can be solved for each i independently. The objective function component for customer i is its improvement criterion minus a constant, hence $d_i^*=\delta_i^*(z_i)$, and the relationship between δ^D and $\{\delta_j^*\}$ is established.

To verify that relationship (4.46) holds for all s, we first show that

$$q_s^{\delta^D(s)}=\sum_{i=1}^{m} {}_iq_{z_i}^{\delta^*_i(z_i)}. \tag{4.51}$$

From (3.7),

$$q_s^{\delta^D(s)}=\sum_{j=1}^{n} C_j y_j+E_{\{x_i\}}\sum_{i=1}^{m}\left(H_i z_i'+L_i e_i\right), \tag{4.52}$$

where $\delta^D(s)=\{y_j\}$ and z_i' and e_i are as defined in (3.8) and (3.9), respectively (with all $l_{ij}=0$). By application of the lemma, we have

$$\sum_{j=1}^{n} d_{ij}y_j=\delta^*_i(z_i)$$

and

$$\sum_{j=1}^{n} C_j y_j=\sum_{i=1}^{m} C_{i,\delta^*_i(z_i)}.$$

(4.52) may then be reexpressed as

$$q_s^{\delta^D(s)}=\sum_{i=1}^{m}\left[C_{i,\delta^*_i(z_i)}+E_{x_i}\left(H_i z_i'+L_i e_i\right)\right]$$

$$=\sum_{i=1}^{m} {}_iq_{z_i}^{\delta^*_i(z_i)}.$$

Now we would like to equate the expected future value terms of (4.46). To do this, we establish and employ the relationship

$$\sum_{s'\in S_{iz_i'}} p_{ss'}^{\delta^D(s)}={}_ip_{z_iz_i'}^{\delta^*_i(z_i)}, \tag{4.53}$$

where

$$S_{iz}=\left\{s \mid z_i=z\right\}.$$

(4.53) holds since the inventory state of customer i in s is z_i, each customer's inventory process behaves independently of the others (since there is never vehicle unavailability), and for i's inventory to be z_i' at the next decision point, the system must enter a state s' where i's inventory level is z_i'. This independence relationship can be extended to show that

$$\sum_{s' \in S_{\iota z'_\iota}} \pi^{\delta^D}_{s'|s}(t) = {}_\iota\pi^{\delta^*_\iota}_{z'_\iota|z_\iota}(t). \tag{4.54}$$

The relative value is defined through (4.4) and (4.6) as

$$v^{\delta^D}_s = \lim_{t \to \infty} \sum_{\tau=1}^{t} \sum_{s'=1}^{S} \left(\pi^{\delta^D}_{s'|s}(\tau - 1) - \pi^{\delta^D}_{s'} \right) q^{\delta^D(s')}_{s'}. \tag{4.55}$$

Using (4.51),

$$v^{\delta^D}_s = \lim_{t \to \infty} \sum_{\iota=1}^{m} \sum_{\tau=1}^{t} \sum_{s'=1}^{S} \left(\pi^{\delta^D}_{s'|s}(\tau - 1) - \pi^{\delta^D}_{s'} \right) {}_\iota q^{\delta^*_\iota(z_i')}_{z_i'}. \tag{4.56}$$

Observe that a sum over s' from 1 to S is equivalent to a double sum, the inside sum being over states s' in $S_{\iota z_\iota'}$, and the outside sum over z_ι' from 0 to u_i, for any chosen i. Hence,

$$v^{\delta^D}_s = \lim_{t \to \infty} \sum_{\iota=1}^{m} \sum_{\tau=1}^{t} \sum_{z'_\iota=0}^{u_\iota} \sum_{s' \in S_{\iota z'_\iota}} \left(\pi^{\delta^D}_{s'|s}(\tau - 1) - \pi^{\delta^D}_{s'} \right) {}_i q^{\delta^*_\iota(z_\iota')}_{z_i'}$$

$$= \lim_{t \to \infty} \sum_{\iota=1}^{m} \sum_{\tau=1}^{t} \sum_{z'_\iota=0}^{u_\iota} \left({}_i\pi^{\delta^*_i}_{z_i'|z_i}(t-1) - {}_i\pi^{\delta^*_\iota}_{z_i'} \right) {}_i q^{\delta^*_\iota(z_i')}_{z_i'}$$

$$= \sum_{\iota=1}^{m} {}_\iota v^{\delta^*_\iota}_{z_i}, \tag{4.57}$$

invoking the corresponding definition of the relative value in i's DDSP of state z_ι under policy δ_ι. From here, the expected future value terms of (4.46) are shown to be equivalent by the following:

$$\sum_{s'=1}^{S} p^{\delta^D(s)}_{ss'} v^{\delta^D}_{s'} = \sum_{s'=1}^{S} p^{\delta^D(s)}_{ss'} \sum_{\iota=1}^{m} {}_\iota v^{\delta^*_\iota}_{z'_\iota}$$

$$= \sum_{i=1}^{m} \sum_{s'=1}^{S} p_{ss'}^{\delta^{D}(s)} \, {}_{i}v_{z'_i}^{\delta^*_i}$$

$$= \sum_{i=1}^{m} \sum_{z'_i=0}^{u_i} \sum_{s' \in S_{iz'_i}} p_{ss'}^{\delta^{D}(s)} \, {}_{i}v_{z'_i}^{\delta^*_i}$$

$$= \sum_{i=1}^{m} \sum_{z'_i=0}^{u_i} {}_{i}p_{z_i z'_i}^{\delta^*_i(z_i)} \, {}_{i}v_{z'_i}^{\delta^*_i} . \tag{4.58}$$

To complete the proof, we show that, by contradiction, no dispatch better than $\delta^D(s)$ exists in any state s. Suppose that, indeed, in state s, it would be better to make dispatch decision γ than $\delta^D(s)$. This implies

$$q_s^{\gamma} + \sum_{s'=1}^{S} p_{ss'}^{\gamma} \, v_{s'}^{\delta^D} < q_s^{\delta^D(s)} + \sum_{s'=1}^{S} p_{ss'}^{\delta^D(s)} \, v_{s'}^{\delta^D} . \tag{4.59}$$

If $\gamma = \{y_j\}$ does not have the property that, for every itinerary j with $y_j > 0$,

$$\sum_{i=1}^{m} d_{ij}^{1} = 1,$$

then there must exist a decision $\gamma' = \{y_j'\}$ which does have that property and has an improvement criterion value equal to the one for γ. This is so because each itinerary visiting multiple customers can be replaced by single-customer itineraries yielding the same transportation costs and inventory effects. So consider γ to just dispatch single-customer itineraries. Let $d_i = \Sigma_j \, d_{ij} \, y_j$. Note that

$$\sum_{j=1}^{n} C_j y_j = \sum_{j=1}^{n} \sum_{i=1}^{m} C_{j,d_{ij}} y_j = \sum_{i=1}^{m} \sum_{j=1}^{n} C_{j,d_{ij}} y_j \geq \sum_{i=1}^{m} C_{i,d_i}$$

by the convexity of C_{id}, so

$$\sum_{i=1}^{m} \left[{}_{i}q_{z_i}^{d_i} + \sum_{z'_i=0}^{u_i} {}_{i}p_{z_i z'_i}^{d_i} \, {}_{i}v_{z'_i}^{\delta^*_i} \right] \leq q_s^{\gamma} + \sum_{s'=1}^{S} p_{ss'}^{\gamma} \, v_{s'}^{\delta^D} . \tag{4.60}$$

Together with (4.59) and (4.46), this inequality means that

$$\sum_{i=1}^{m}\left[{}_iq_{z_i}^{d_i}+\sum_{z'_i=0}^{u_i}{}_ip_{z_iz'_i}^{d_i}\,{}_iv_{z'_i}^{\delta^*_i}\right]<\sum_{i=1}^{m}\left[{}_iq_{z_i}^{\delta^*_i(z_i)}+\sum_{z'_i=0}^{u_i}{}_ip_{z_iz'_i}^{\delta^*_i(z_i)}\,{}_iv_{z'_i}^{\delta^*_i}\right],$$

so

$${}_iq_{z_i}^{d_i}+\sum_{z'_i=0}^{u_i}{}_ip_{z_iz'_i}^{d_i}\,{}_iv_{z'_i}^{\delta^*_i}<{}_iq_{z_i}^{\delta^*_i(z_i)}+\sum_{z'_i=0}^{u_i}{}_ip_{z_iz'_i}^{\delta^*_i(z_i)}\,{}_iv_{z'_i}^{\delta^*_i}$$

must hold for at least one i. But this would contradict the meaning of the optimal replenishment policy δ_i^* for customer i. Therefore, $\delta^D(s)$ is optimal in state s, and the decomposition algorithm finds the optimal dispatch policy.□

As a consequence of this property, we can also establish the following:

Proposition 2. The cost rate g^{δ^D} for the decomposition policy δ^D is equivalent to the sum of the cost rates from the customer DDSPs.

Proof.

$$g^{\delta^D}=\sum_{s=1}^{S}\pi_s^{\delta^D}q_s^{\delta^D(s)}$$

$$=\sum_{s=1}^{S}\pi_s^{\delta^D}\sum_{i=1}^{m}{}_iq_{z_i}^{\delta^*_i(z_i)}$$

$$=\sum_{i=1}^{m}\sum_{z_i=0}^{u_i}\sum_{s\in S_{iz_i}}\pi_s^{\delta^D}\,{}_iq_{z_i}^{\delta^*_i(z_i)}$$

$$=\sum_{i=1}^{m}\sum_{z_i=0}^{u_i}{}_i\pi_{z_i}^{\delta^*_i}\,{}_iq_{z_i}^{\delta^*_i(z_i)}=\sum_{i=1}^{m}{}_ig^{\delta^*_i}.\square \tag{4.61}$$

Now we may establish a lower bound for the optimal cost rate in the restricted DDP of this section.

Proposition 3. The optimal cost rate $g^{\delta *}$ for the DDP with zero leadtimes and maximum customer delivery Ω per period is greater than or equal to the sum of the cost rates ${}_{i}g^{\delta_i *}$ for the optimal solutions of single-customer DDSPs solved with $F=0$ and C_{id} chosen such that

$$\sum_{i=1}^{m} C_{i,d_{ij}} \leq C_j \quad \forall j$$

and C_{id} is strictly convex with respect to d for $d>0$.

Proof. Consider the following two DDPs obtained through modification of the original DDP:

1) $DDP_1 = DDP$ with $C_j{}^1 = \Sigma_i C_{i,d_{ij}}$ for every j and number of vehicles $K^1 = K$.

2) $DDP_2 = DDP_1$ with $C_j{}^2 = C_j{}^1$ for every j and $K^2 = m\tau$, where maximum itinerary duration $\tau = max_j \tau_j$.

Let $g_1{}^{\delta *}$ and $g_2{}^{\delta *}$ be the optimal cost rates for DDP_1 and DDP_2, respectively. It is clear that $g^{\delta *} \geq g_1{}^{\delta *}$, since the only difference between the two problems is that the transportation costs in DDP_1 are smaller. Also, $g_1{}^{\delta *} \geq g_2{}^{\delta *}$, since the only operational difference between DDP_1 and DDP_2 is that the dispatch $\delta_2{}^*(s)$ may be infeasible in DDP_1 for state s due to vehicle unavailability. But $\delta_2{}^* = \delta^D$, therefore

$$g^{\delta *} \geq g^{\delta^D} = \sum_{i=1}^{m} {}_{i}g^{\delta *_i}, \tag{4.62}$$

by Proposition 2.□

It is reasonable to expect that, the closer on the whole that the sums $\Sigma_i C_{i,d_{ij}}$ are to C_j, the tighter the lower bound on $g^{\delta *}$ will be. The effect of vehicle availability on the quality of the lower bound is generally more difficult to gauge Customers may commonly receive a positive delivery quantity other than the delivery target, but how close the decomposition policy's cost rate comes to $\Sigma_i \ {}_{i}g^{\delta_i *}$ is uncertain under these circumstances. Also, we have not attempted to answer questions about the validity of these lower bounds for the more general DDP studied in the rest of this chapter.

In Section 4.2, the computational complexity of each iteration of the policy iteration algorithm with S states and A actions in each state was shown to be $O(S^3 + AS^2)$ operations. If ifnding the optimal solution takes I iterations, then the complexity of the policy iteration algorithm is $O(I[S^3 + AS^2])$ (the substantial calculations needed to determine $p_{ss'}{}^{\gamma}$ and $q_s{}^{\gamma}$ are not even included here). Let us estimate the quantities S and A for the deliverer dispatch problem with zero leadtimes.

If each customer has capacity for $u-1$ units of the delivered good, and if each itinerary takes a maximum of τ units to complete, then the number of states is $u^m\tau^K$, where m is the number of customers and K the number of vehicles. (Since leadtimes are all zero, no previous dispatch information need be included in the system state description.) The number of dispatches varies by state. First let us approximate the number of itineraries. Suppose vehicles travel with full loads (Ω units), and visit two customers per itinerary (in the applications we envision, the number of visits per itinerary will usually be three or less). There are approximately m^2 different ways to pair customers in an itinerary, and for each pair, Ω-1 ways to split the load so that each customer receives one unit. Add to this the m single-customer itineraries, and it seems reasonable to estimate the total number of itineraries conservatively as being $O(\Omega m^2)$. Suupose now that the average itinerary duration is τ_a periods. Then a reasonable guess for the average number of vehicles available at a decision point is K/τ_a. Working from these estimates, the average number of actions available in a state is judged to be

$$O\left([\Omega m^2]^{K/\tau_a} \right).$$

We therefore place the complexity of the policy iteration algorithm for solving the DDP at

$$O\left(I \left[u^{3m}\tau^{3K} + u^{2m}\tau^{2K}(\Omega m^2)^{K/\tau_a} \right] \right), \tag{4.63}$$

acknowledging that this may underestimate the true complexity to some extent.

Comparing the computational complexity of the FVD algorithm to that of policy iteration is hard to do in one sense. The policy iteration algorithm finds the policy for every state as a matter of course, whereas the FVD algorithm only determines them as needed. We will calculate the complexity of the solution of the DDSPs and the per decision point complexity of the MSP to gain some conception of the magnitude of the computational difference between the exact method and our heuristic.

We use the same information about the structure of the DDSP as above. Each DDSP is a Markov decision problem which is solved exactly. Each has only u different states (when leadtimes are zero), and $\Omega + 1$ actions (replenishment quantities) per state. If it takes I' iterations to solve each DDSP by the policy iteration algorithm, then the total computational complexity of the preprocessing phase of the FVD algorithm is $O(mI' [u^3 + u^2\Omega])$. If the MSP is solved via the algorithm given in Section 4.3.4, then the number of operations can be inferred from what amounts to a nested do-loop structure. The outermost loop, in the main procedure, ranges over the number of available vehicles, which we guess to be K/τ_a in the average case. The loop in the BEST_ITIN procedure runs over the set of undominated itineraries. This set varies roughly in size according to the square of the number of customers, when itineraries visit two customers at most. GREEDY_ASSIGN adds two more nested do-loops, one over vehicles and one over itineraries. Hence, the complexity of determining a dispatch by heuristically solving the MSP is $O([K/\tau_a]^2 [m^2]^2)$ or $O(K^2 m^4 / \tau_a^2)$. To determine the dispatches for D decision points via the FVD algorithm requires a total number of operations of

$$O\left(mI' \left[u^3 + u^2\Omega \right] + \frac{DK^2m^4}{\tau_a^2} \right). \tag{4.64}$$

The FVD algorithm is thus observed to be polynomial, with repect to the numbers of customers and of vehicles, in its order of complexity, while the policy iteration algorithm is exponential.

4.3.7 Extensions

The approximations that have been utilized to separate the original DDP into single-customer problems in the preprocessing phase limit the quality of dispatching through the decomposition algorithm. That is, misrepresentation of joint replenishment effects in the customer subproblems may inflict serious damage to the effectiveness of our proposed heuristic. We prescribe in this section several extensions to the basic algorithm that may help prevent algorithm-generated schedules from going awry.

4.3.7.1 Determination of F

One important input to the single-customer DDSPs is the dispatch failure probability F. The probability of a dispatch failure depends upon demand rates, itinerary durations, and the dispatch policy being used. If the decomposition algorithm described in this section directs dispatching, then the dispatch policy, in turn, depends upon the failure probability. Although these forms of mutual dependence relationships often render it impossible to derive closed-form problem solutions, they do sometimes lend themselves to iterative solution schemes where the outputs of succeeding iterations converge to the quantities desired. In our case, even the development of an iterative scheme proceeds only with difficulty. The root of the problem lies with the activity of joint replenishment. Vehicles frequently replenish two or more customers on a single trip in our models. One might be inclined to depict vehicles as "servers" and customers' optimal replenishment quantities (from the DDSPs) as "customers" and to derive a dispatch failure probability via queueing analysis. But the service characteristic that possibly more than one customer may be served during a single service administration, depending on who the customers in queue are, severely complicates the analysis. One can choose to rely on approximations of the queue's behavior to derive the required measures, but only at some risk of accuracy.

The ready alternative to analytic determination of the dispatch failure probability is simulation. Using the decomposition algorithm for dispatching with F set at an initial value, one may simulate the delivery system for a given number of periods. The value of F updated through the simulation would correspond to the number of customer-dispatches executed (incremented by one for each customer on each dispatch), divided by the number of customer-periods in which the customer's DDSP policy dictates it receives a dispatch that period. The simulation process may then be repeated, each time taking the updated value of F from the last round as the initial guess for the next, until the last updated value of the dispatch failure probability is judged adequate for dispatching needs. We note that it would be just as easy to obtain from simulation distinct failure probabilities F_i for each customer i, possibly at the expense of longer simulation trials for the same degree of accuracy as with determining F alone.

When simulating the delivery system in the manner described above, one must negotiate the simulation hazards of start-up bias and autocorrelation of observed dispatch success and failure counts. These hazards may be potentially avoided using a slightly different scheme. From the solutions of the DDSPs, it is easy to obtain the steady-state occupancy probabilites for the states of the customer subproblems. A general idea for randomly drawing a state for the entire system at the steady state is as follows: Draw a customer state randomly for each customer. These draws supply directly the current inventory levels of the system state. The outstanding deliveries of the random customer states, together with the vehicle fleet size and itinerary durations, then will either indicate current vehicle availability, at which point the MSP may be solved to determine customer-dispatches executed and failed, or be inconsistent with available vehicle resources, causing the draw to be discarded. Each observation is generated in this fashion. Using this procedure, we sample directly from the steady-state (or approximation thereof), and the observations are uncorrelated with one another. The one loose thread here is how to determine what the set of outstanding deliveries implies for vehicle availability in a given draw, particularly when it is only possible to have arrived at the outstanding delivery state by packing several customer deliveries into single itineraries. Further investigation of this approach is warranted for future research efforts.

4.3.7.2 DDSP Itinerary Attribute Estimation

If the dispatch failure probability is to be iteratively refined and simulation or random sampling is the method of choice for this task, refining the cost and leadtime of replenishment for each customer and delivery quantity can be accomplished simultaneously. The procedure is straightforward: For each customer i, let the updated cost share C_{id} and leadtime l_{id} for a replenishment of size d be the averages of their respective quantities in the simulation. Determination of the simulation duration for a given refinement iteration should take into account the requirements for refining the estimates of replenishment cost shares and leadtimes.

4.3.7.3 Real-Time Updates

The principles of input parameter refinement can also be utilized to adjust the components of a decomposition-based dispatching mechanism once it is installed and running in a physical distribution environment. Provided that the mechanism has access to an historical system information bank, the same iterative process as outlined above can be executed in real-time to generate updated penalty functions for use by the master scheduler at subsequent decision points. Since the computational effort for an iteration may be rather extensive, and the rate of parameter change slow, updating is probably advisable only on an infrequent basis.

Periodic parameter refinement carries an additional benefit in real-world systems: it enables the dispatch mechanism to adapt to changes in the physical environment. Just as replenishment cost and leadtime information for the customer DDSPs is garnered from the recent historical profile, so too should the inventory cost and demand rate inputs be drawn from this database. The CAD system whose dispatch generator is implemented with this feature may be qualified to handle real-world operations corresponding to less-specialized DDPs, such as those in which the specializations of Sections 3.2.3 and 3.2.4 do not hold. Exhibit 4.8 indicates how a delivery system would operate when utilizing a CAD system with real-time updates.

4.3.7.4 Extended Scheduling Horizon

A further extension of the FVD algorithm is intended to address two of its perceptible shortcomings, namely: the potential inadequacy of the expected future value approximations, and the omission of future vehicle availability considerations in the master scheduling problem. The first shortcoming noted was discussed in more depth in the algorithm's development above. The second refers to the algorithm's characteristic that the impacts of vehicle availability with respect to dispatch decisions are only felt in the limit on the number of dispatches that can be issued at the current decision point and indirectly in the expected future values arising from the DDSPs. The effect of vehicle availability on the valuation of states attainable at the next decision point is lost. To illustrate, consider a DDP with a two-vehicle fleet, and suppose that at a particular decision point, only one vehicle is ready to be dispatched. The master scheduler would return the same dispatch whether the unavailable vehicle were ready one decision point or four decision points hence. If the latter were the case, the current dispatch may be better directed toward heavy demand customers, who may experience severe stockout consequences regardless of their current inventories if their deliveries are postponed until the next dispatch. The current inventory state of the system probably will have more influence in the former case, because the customers with heavy demand may be served equally well between dispatch now and at the next decision point.

The extension to the FVD algorithm we propose converts the master scheduling problem from a one-period to a multi-period scheduling horizon. Exhibit 4.9 describes this extension graphically. The *scheduling horizon* refers to that interval over which immediate costs are assessed; expected future values accrue over the *future planning horizon*. The original MSP implicitly utilizes a scheduling horizon one period in duration. The extended master scheduling

Exhibit 4.8: Implementation of FVD Algorithm with Real-Time Updating in Delivery System

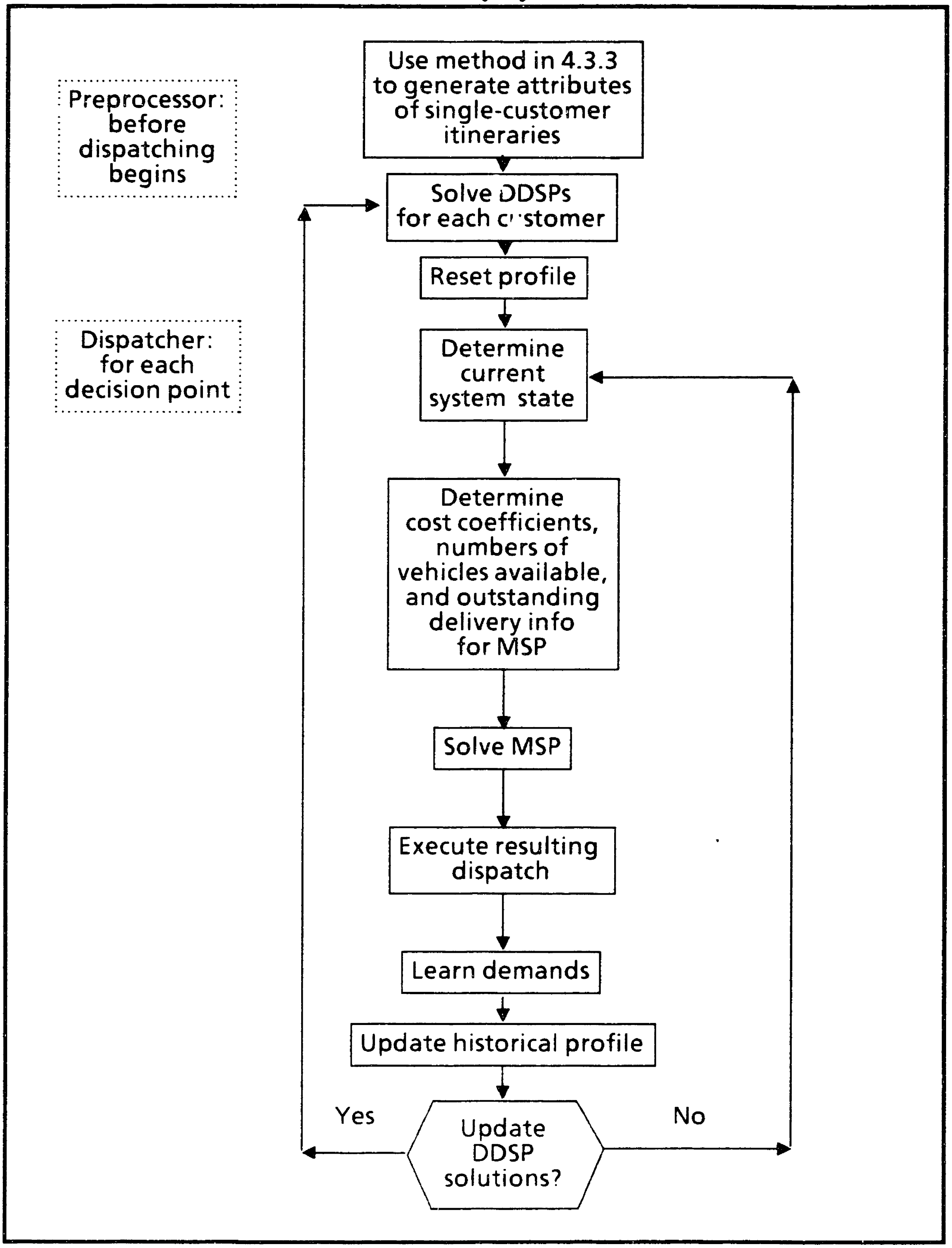
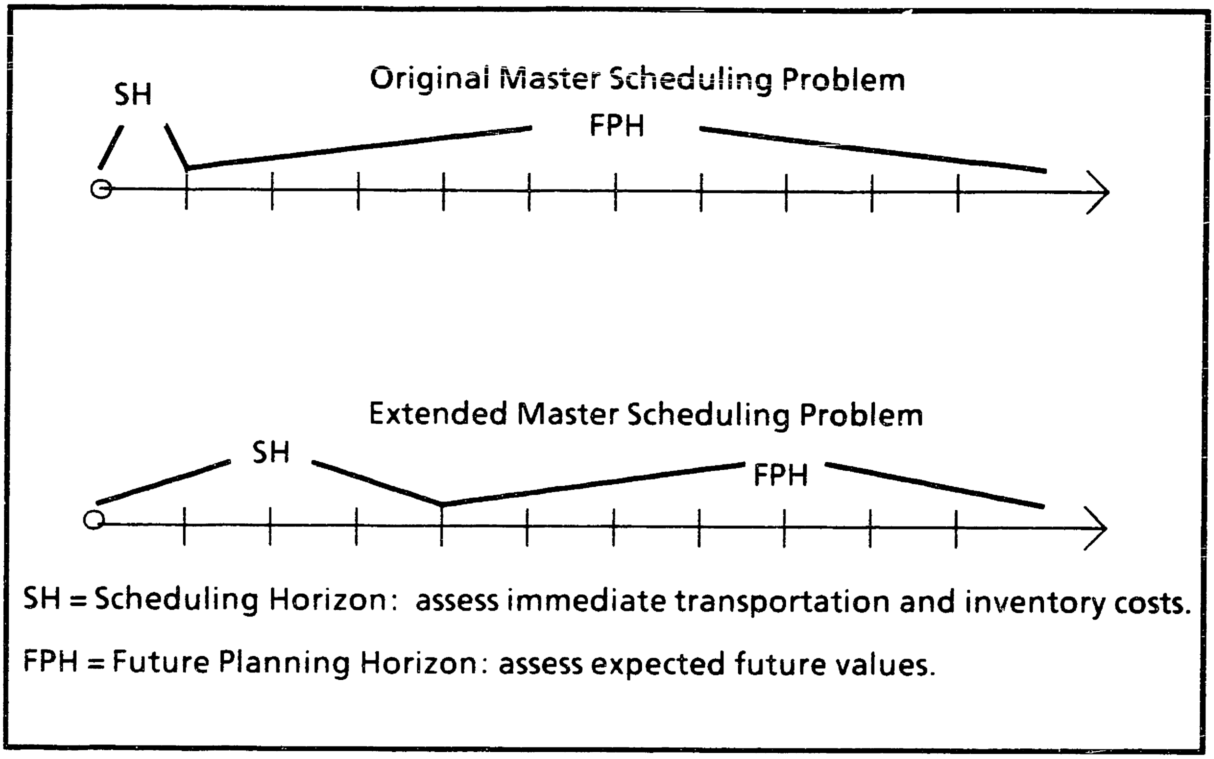


Exhibit 4.9: Extended Master Scheduling Problem



problem (EMSP) involves the selection of dispatches over a multi-period scheduling horizon to minimize immediate costs sustained during the horizon plus the approximated expected future value registered afterwards. However, commitment is only made to the dispatches scheduled for the current decision point. In order for the EMSP to produce meaningful results, the DDSPs that are solved to generate some of its coefficients must also be extended to multi-period problems.

There are several advantages to be realized from this extension to the FVD algorithm. One is that vehicle availability effects are better represented when scheduling over a multi-period horizon: the EMSP would probably be more sensitive to situations such as the one in the two-vehicle example given above. Multi-period DDSPs must have extended decision spaces indicating the timing as well as quantity of the replenishment order, and state transition probabilities and inventory costs may be more difficult to handle for an extended planning horizon. Yet it should prove less difficult to approximate accurately, if that is indeed necessary, immediate costs than expected future values in the multi-period DDSP. That being the case, the objective function of the EMSP will resemble more precisely the improvement criterion for the current state, because the immediate cost component comprises a greater proportion of the improvement criterion.

To confirm the supposition that extending the scheduling horizon "improves" the improvement criterion, we refer back to the development of the policy iteration algorithm in Section 4.1. Extending from the principle of optimality, if the nonstationary policy of using rule δ' for the first n decision points (the "immediate horizon") and rule δ thereafter outperforms the stationary policy δ, then among stationary policies, δ' is better than δ. In terms of improvement criteria, δ' improves on δ if and only if

$$\lim_{t\to\infty}\left(t+n\right)g^{\delta}+v^{\delta}_{s_0}-\left[\sum_{s=1}^{S} q_s^{\delta'(s)}\sum_{m=0}^{n-1}\pi^{\delta'}_{s|s_0}(m)+\sum_{s=1}^{S}\pi^{\delta'}_{s|s_0}(n)\left(\lim_{t\to\infty} tg^{\delta}+v^{\delta}_{s}\right)\right]\geq 0 \quad \textbf{(4.65)}$$

for all s_0, with strict inequality for at least one s_0. As in policy iteration for $n=1$, policy improvement proceeds here by maximizing the left-hand side of (4.65), which is equivalent to

$$ng^{\delta}+v^{\delta}_{s_0}-\sum_{s=1}^{S} q_s^{\delta'(s)}\sum_{m=0}^{n-1}\pi^{\delta'}_{s|s_0}(m)-\sum_{s=1}^{S}\pi^{\delta'}_{s|s_0}(n)\,v^{\delta}_{s} \quad \textbf{(4.66)}$$

or by minimizing

$$\sum_{s=1}^{S} q_s^{\delta'(s)}\sum_{m=0}^{n-1}\pi^{\delta'}_{s|s_0}(m)+\sum_{s=1}^{S}\pi^{\delta'}_{s|s_0}(n)\,v^{\delta}_{s} \quad \textbf{(4.67)}$$

for each state s_0. This criterion reduces to (4.12) for $n=1$, as expected. Observe that, as n increases, the immediate cost term of (4.67) gets larger, whereas the expected future value term converges to the constant

$$\sum_{s=1}^{S}\pi^{\delta'}_{s}v^{\delta}_{s}.$$

Hence, the immediate cost term asymptotically assumes a greater proportion of the improvement criterion. Under the assumption that approximations of the immediate costs are more accurate than those of the expected future values, the improvement criteria become in some sense more reliable as n (the scheduling horizon in our application) grows. Of course, it also becomes more and more burdensome to estimate immediate costs with increasing n; tradeoffs must be made when deciding upon the length of the immediate horizon.

Some real-world systems that we might want to apply our dispatch decision support methodology to exhibit, in our model's terminology, interperiod demand correlation. Consider a continuous-time system in which, each occasion on which a customer receives a demand, the size of the demand is exactly one unit. Moreover, these demands may occur at any instant in time. If we choose to fit this system within our deliverer dispatch framework as it has been established to this point, full interperiod demand independence only occurs when the probability distribution of interdemand intervals is negative exponential. If the intervals follow some other probability distribution, then information at a decision point about demand in the preceding period allows a more accurate forecast of demand for the upcoming period. The presence of interperiod demand correlation violates the assumptions necessary to establish that the DDP derived from the system at hand is a Markov decision problem, and therefore dispatch policies generated via Markov decision theory may be suboptimal, perhaps severely so. With an extended scheduling horizon, the impact on predictive accuracy of prior demand information decreases in magnitude, so that the degree of suboptimality in Markov decision-theoretic solutions may be assumed to decline as well. Additionally, expected immediate cost calculations can be easier to perform over an extended horizon for interdemand interval distributions with low coefficients of variation than for those with high ones (such as the negative exponential), since the range of total demand quantities per period of the scheduling horizon may well be smaller (refer to the case study in Part II of this dissertation).

One other potential drawback of multi-period scheduling horizons concerns feasible dispatch decisions, feasible both in the sense of legitimacy and in the sense of computability. In the original DDSP, we permitted only one delivery to remain outstanding in order to restrict the state space, although multiple outstanding deliveries do not pose any inordinate difficulties toward the calculation of immediate costs and transition probabilities. The implication of single outstanding deliveries is that a customer receives a maximum of one dispatch per decision point. The DDSPs would sustain a combinatorial increase in the size of the decision space, and the master scheduling problem in the number of decision variables, were multiple dispatches to a customer permitted. While the restriction to single dispatches might not ruin the quality of dispatching with the one-period scheduling horizon procedure, it will tend to relegate multi-period schedules to greater and greater degrees of suboptimality as the length of the horizon increases, simply because an increasing proportion of customers will experience demand for over one vehicle-load of the delivered good. This is not to say that the quality of dispatching for long scheduling horizons necessarily suffers, since only the current decision point's dispatches are adhered to. But there is the danger that customers will want their single delivery to cover demand for the entire scheduling horizon, when near-optimal policies dictate more frequent visits with smaller delivery quantities. We warn that the information contained in the solution to EMSPs with long horizons is, at best, misleading. The impetus to keep the EMSP manageable contributes to the forces limiting the length of the scheduling horizon.

The formulation and solution of multi-period deliverer dispatch subproblems and master scheduling problems will be treated in conjunction with the computational and applied work of Chapters V, VII, and VIII.

4.4 Other Heuristic Approaches

Other avenues can be explored in the quest for a viable DDP solution procedure. This section briefly describes alternative heuristic approaches. Some of these ideas have been pursued in [19].

4.4.1 Value Approximation Via Weighted Functions of the State

An alternative method for approximating the expected future value of a given state, decision, and policy set is suggested by the recent work of Schweitzer and Seidmann [25]. Their main idea is to approximate the cost rate g^{δ} and relative values v_s^{δ} by weighted sums of functions of the state. Exploiting this assumption, the value determination phase reduces to the determination of the weights of the functions (which are ideally far less in number than states are) and the cost rate. Policy improvement remains about as computationally burdensome as in exact policy iteration.

The foregoing approach is appealing as a basis for a DDP heuristic because the state contains extensive inventory level and vehicle availability information in vector form. A condensed expression of the state of the system for value assessment purposes may well suffice in some circumstances. However, because the policy improvement step of Schweitzer and Seidmann's approximation algorithm is still computationally infeasible in problems with large state spaces, their ideas cannot be applied directly in CAD systems for operations commonly encountered. The approximation idea of weighted sums of state functions may prove useful as a component in some larger heuristic scheme.

4.4.2 Aggregation/Disaggregation

Aggregation/disaggregation is a general idea for reducing the computational load of a DDP solution procedure to a manageable size. The intent is to develop a hierarchical dispatch policy via Markov decision theory. The initial aggregation phase of the prototypical procedure formulates and solves a small MDP designed to "resemble" the original problem. Geographic clustering of customers and aggregations of units of the delivered good are some means that may be employed in this endeavor. Preferably, the resemblance between original and aggregate problems will be such that good dispatches in each problem will usually be aimed at the same geographic region. At the next level, the dispatch is further refined by referring the state of the targeted region to the solution of a somewhat less aggregate MDP in which only the customers of that region are considered. This level may not provide the final dispatch if there are still too many customers in the region for feasible solution of a modified DDP over the subset of customers in the region. The disaggregation process continues until a dispatch relevant to the original DDP is produced. How to perform the aggregations and disaggregations well is a topic for future research.

4.4.3 Modified Coordinated Inventory Replenishment

Section 4.3.1 presented the obstacles impeding the direct application of coordinated inventory replenishment theory to the DDP we have formulated. However, it may still be possible to modify the function of the (S,c,s) policies (say, limiting add-on orders to only those customers in the same geographic region) and the procedures for determining them so that they respect the distinguishing characteristics of our DDP.

CHAPTER V

COMPUTATIONAL STUDIES OF THE DELIVERER DISPATCH PROBLEM

5.0 Introduction

This chapter studies the deliverer dispatch problem from a computational perspective. The Future Value Decomposition algorithm developed in Chapter IV is applied to instances of the DDP conforming to the specializations set out in Sections 3.2.1-3.2.6. The main vehicle for evaluating the FVD heuristic is simulation. Initially, we illustrate the basic concepts of our computational analysis through their application to a small, in fact optimally solvable, instance of a DDP. This preliminary investigation also serves to provide insight into how the FVD algorithm operates and how well it performs, at least on small problems.

5.1 Decomposition Applied to a Small DDP: Illustration of Computational Analysis

Consider a distribution operation in which two vehicles regularly replenish three customers. Exhibit 5.1 displays such an operation. Customers are identical with respect to the following characteristics:

Inventory holding capacity: 3 units of the delivered good.

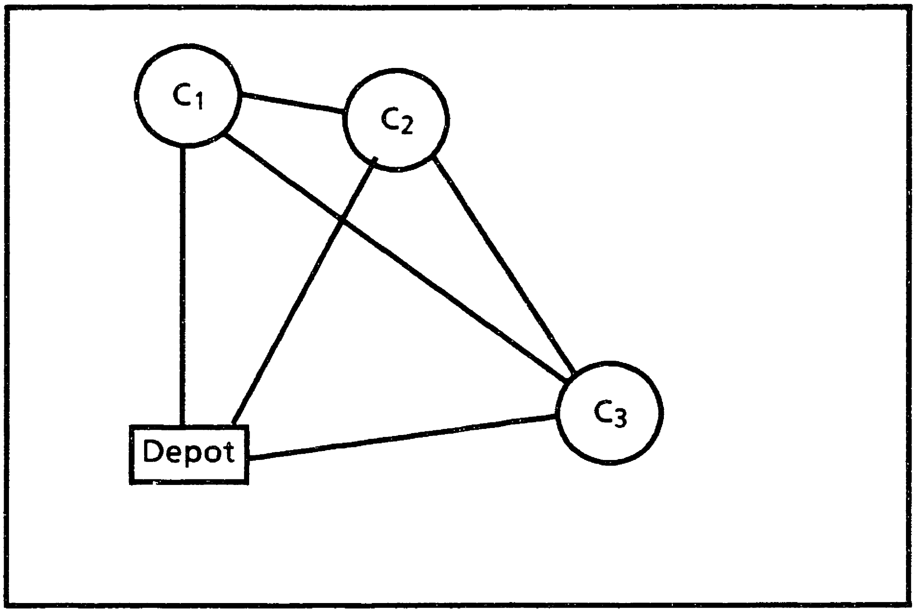
Demand: 1 unit with probability 0.91;

0 and 2 units with probability 0.045 each (the coefficient of variation of demand per customer per period is 0.3).

Holding cost: 4 cost units per unit of the delivered good in stock per period.

Lost demand cost: 30 cost units per unit of lost demand.

Exhibit 5.1: Transportation Network for a Small DDP



The two vehicles in the fleet have the same carrying capacity of 3 units per itinerary. Information about the itinerary set is supplied in Exhibit 5.2. All leadtimes are zero, but the itinerary durations may be one or two periods in length. Since itinerary durations do not exceed two periods, then any vehicle not available at one decision point will definitely be ready by the next one. The leadtime and itinerary duration specifications reduce the set of non-inventory information elements that must be included in the system state to the number of vehicles available for dispatch at a decision point.

5.1.1 Optimal Solution

Since each of the three customers' inventory levels can assume four different values (0, 1, 2, and 3), and there are three potential vehicle states (0, 1, and 2 vehicles available), there are $4 \times 4 \times 4 \times 3 = 192$ states present in this "small" DDP. When no vehicles are available, the dispatch decision must be automatically "do not dispatch any vehicles." With one vehicle ready, 19 different decisions (the number of itineraries in the itinerary set) may be made. The number of

Exhibit 5.2: Route Set for a Small DDP

Itinerary #	Delivery			Duration	Transportation Cost
	C_1	C_2	C_3		
0	0	0	0	1	0
1	1	0	0	1	6
2	0	1	0	1	8
3	0	0	1	1	8
4	2	0	0	1	6
5	0	2	0	1	8
6	0	0	2	1	8
7	1	1	0	2	11
8	1	0	1	2	14
9	0	1	1	2	13
10	3	0	0	1	6
11	0	3	0	1	8
12	0	0	3	1	8
13	2	1	0	2	11
14	2	0	1	2	14
15	0	2	1	2	13
16	1	2	0	2	11
17	1	0	2	2	14
18	0	1	2	2	13

distinct dispatch decisions jumps to 190 when both vehicles may be dispatched immediately ($19 \times 18 \div 2$ ways of dispatching two vehicles on different itineraries, plus 19 decisions sending both vehicles on the same itinerary). Nevertheless, this example may be solved optimally. Exhibit 5.3 presents the optimal solution. The cost rate for the best dispatch policy is approximately 19.1. (Note: the Markov decision problem algorithm used was a modified version of the policy iteration algorithm known as a "k-th order" algorithm--see [12, pp. 257-266].)

Exhibit 5.3: Optimal Solution for DDP Example

States		$z_3 = 0$			$z_3 = 1$			$z_3 = 2$			$z_3 = 3$		
z_1	z_2	2	1	0	2	1	0	2	1	0	2	1	0
0	0	16,6	13,0	0,0	5,4	13,0	0,0	16,0	16,0	0,0	16,0	16,0	0,0
0	1	6,4	17,0	0,0	4,0	4,0	0,0	4,0	4,0	0,0	4,0	4,0	0,0
0	2	6,4	17,0	0,0	4,0	4,0	0,0	4,0	4,0	0,0	4,0	4,0	0,0
0	3	6,4	17,0	0,0	4,0	4,0	0,0	4,0	4,0	0,0	4,0	4,0	0,0
1	0	6,5	18,0	0,0	5,0	5,0	0,0	5,0	5,0	0,0	5,0	5,0	0,0
1	1	6,0	6,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
1	2	6,0	6,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
1	3	6,0	6,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
2	0	6,5	18,0	0,0	5,0	5,0	0,0	5,0	5,0	0,0	5,0	5,0	0,0
2	1	6,0	6,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
2	2	6,0	6,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
2	3	6,0	6,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
3	0	18,0	18,0	0,0	5,0	5,0	0,0	5,0	5,0	0,0	5,0	5,0	0,0
3	1	6,0	6,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
3	2	6,0	6,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
3	3	6,0	6,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0

z_i = inventory level of customer i;
subcolumn headings are numbers of vehicles available;
table entries are the itineraries to select for each state

Exhibit 5.4 reveals some other interesting information associated with the optimal solution of this DDP. The steady-state occupancy probabilities yielded by the optimal dispatch policy indicate that no customer will ever hold three units of inventory (except perhaps initially), and that both vehicles are never simultaneously unavailable at any decision point. The cost and demand structures of the DDP produce the former condition. Prior to solving the problem, one might have anticipated that dispatch decision (13,18) (sending two units to each customer) might be useful in the state with no customers holding any inventory and both vehicles available, even though no vehicles would be available at the next decision point. But even in that state, the

Exhibit 5.4: Steady-State Probabilities for DDP Example

States		$z_3 = 0$			$z_3 = 1$			$z_3 = 2$			$z_3 = 3$		
z_1	z_2	2	1	0	2	1	0	2	1	0	2	1	0
0	0	.045	.002	0	.108	.016	0	.005	.001	0	0	0	0
0	1	.110	.002	0	.160	.041	0	.008	.002	0	0	0	0
0	2	.005	0	0	.008	.002	0	0	0	0	0	0	0
0	3	0	0	0	0	0	0	0	0	0	0	0	0
1	0	.180	.014	0	.113	.002	0	.005	0	0	0	0	0
1	1	.111	.001	0	.029	.002	0	.001	0	0	0	0	0
1	2	.005	0	0	.001	0	0	0	0	0	0	0	0
1	3	0	0	0	0	0	0	0	0	0	0	0	0
2	0	.009	.001	0	.005	0	0	0	0	0	0	0	0
2	1	.005	0	0	.001	0	0	0	0	0	0	0	0
2	2	0	0	0	0	0	0	0	0	0	0	0	0
2	3	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0	0	0	0
3	2	0	0	0	0	0	0	0	0	0	0	0	0
3	3	0	0	0	0	0	0	0	0	0	0	0	0

z_i = inventory level of customer i
subcolumn headings are numbers of vehicles available

optimal dispatch assures that one vehicle can be dispatched at the next decision point, if necessary. This decision risks customer 1 sustaining a lost demand in the immediate period; evidently, the benefits of maintaining vehicle availability and reducing transportation cost outweigh the added lost demand risk.

The relative values for the deliverer dispatch problem example also emphasize the importance of having vehicles available. Exhibit 5.5 lists the relative values of all the states. We have calculated and displayed in Exhibit 5.6 the differences in relative value between the states

Exhibit 5.5: Relative Values for DDP Example

States		$z_3 = 0$			$z_3 = 1$			$z_3 = 2$			$z_3 = 3$		
z_1	z_2	2	1	0	2	1	0	2	1	0	2	1	0
0	0	0	28	71	-5	-1	42	-8	-8	41	-7	-7	41
0	1	-5	1	42	-10	-10	14	-13	-13	12	-1	-1	12
0	2	-8	-5	41	-13	-13	12	-16	-16	9	-14	-14	11
0	3	-6	-4	42	-11	-11	13	-14	-14	11	-12	-12	12
1	0	-3	0	42	-9	-9	14	-11	-11	12	-10	-10	13
1	1	-9	-9	14	-15	-15	-15	-17	-17	-17	-16	-16	-16
1	2	-11	-11	12	-17	-17	-17	-19	-19	-19	-18	-18	-18
1	3	-10	-10	13	-16	-16	-16	-18	-18	-18	-17	-17	-17
2	0	-6	-5	42	-11	-11	13	-14	-14	10	-12	-12	12
2	1	-11	-11	13	-16	-16	-16	-19	-19	-19	-17	-17	-17
2	2	-14	-14	10	-19	-19	-19	-22	-22	-22	-20	-20	-20
2	3	-12	-12	12	-17	-17	-17	-20	-20	-20	-18	-18	-18
3	0	-3	-3	44	-8	-8	15	-11	-11	13	-9	-9	14
3	1	-8	-8	15	-14	-14	-14	-16	-16	-16	-15	-15	-15
3	2	-11	-11	13	-16	-16	-16	-19	-19	-19	-17	-17	-17
3	3	-9	-9	14	-15	-15	-15	-17	-17	-17	-16	-16	-16

state of reference: (0,0,0;2)
values rounded to nearest integer

$$s = \left(z_1, z_2, z_3 : a \right) \tag{5.1}$$

and

$$s_0 = \left(z_1, z_2, z_3 ; 0 \right)$$

(where z_i is customer i's inventory level and a is the number of vehicles available) for the seven states s most likely to be occupied in the steady-state under the optimal dispatch policy. Recall that the difference between two states' relative values provides a measure of the preferability of occupying one state versus the other, or, conceptually, what one would "pay" to occupy the more preferred state. The magnitudes of these differences, in comparison with, say, the cost rate or the

Exhibit 5.6: The Value of Vehicle Availability

s	$\pi_s{}^{\delta*}$	$v_{s_0}{}^{\delta*} - v_s{}^{\delta*}$
(1,0,0;2)	0.1797	45.7
(0,1,1;2)	0.1595	23.5
(1,0,1;2)	0.1127	22.4
(1,1,0;2)	0.1112	22.5
(0,1,0;2)	0.1099	46.7
(0,0,1;2)	0.1078	46.7
(0,0,0;2)	0.0451	70.9

$s=(z_1, z_2, z_3; a)$
$s_0=(z_1, z_2, z_3; 0)$

relative value differences between two vehicles and one vehicle available, clearly indicate how critically the best performing dispatching procedures depend on vehicle availability.

5.1.2 FVD Algorithm Solutions

Now we demonstrate the application of the FVD algorithm developed in the previous chapter to this example. First we notice that, because all leadtimes are zero, the state of a single-customer deliverer dispatch subproblem is simply the customer's inventory level (no deliveries can be outstanding). Initially we solve the DDSPs with the dispatch failure probability $F=0$. The cost rates and replenishment policies generated by the optimal solutions to the DDSPs are given in Exhibit 5.7. More relevant to establishing a dispatch policy for the original problem are the dispatch savings quantities

Exhibit 5.7: Solutions of the DDSPs for the Example

i (customer)	${}_ig^{\delta_i*}$	$\delta_i*(0)$	$\delta_i*(1)$	$\delta_i*(2)$	$\delta_i*(3)$
1	6.5	2	0	0	0
2	6.7	2	0	0	0
3	7.0	2	0	0	0

$$ {}_iD^d_{z_i} \equiv {}_iQ^d_{z_i} - {}_iQ^0_{z_i} + \sum_{z'_i=0}^{3} \left({}^0_ip^d_{z_iz'_i} - {}^0_ip^0_{z_iz'_i} \right) {}_iv^{\delta*_i}_{z'_i} , $$

components of the dispatch costs C_j^+ as defined in (4.36). The dispatch savings is the difference between the sums of immediate failsafe inventory cost and expected future value for dispatches of d and zero units to customer i, and is employed in the calculations of the modified cost coefficients for the master scheduling problem. Exhibit 5.8 provides their values extracted from the DDSP solutions.

Exhibit 5.8: Dispatch Savings Values for the Example

i	z_i	${}_iD_{z_i}{}^0$	${}_iD_{z_i}{}^1$	${}_iD_{z_i}{}^2$	${}_iD_{z_i}{}^3$
1	0	0	-28.7	-31.0	-29.4
	1	0	-2.3	-0.7	4.6
	2	0	1.7	6.9	7.2
	3	0	5.3	5.5	5.5
2	0	0	-28.7	-31.3	-29.8
	1	0	-2.5	-1.1	4.0
	2	0	1.4	6.5	6.8
	3	0	5.1	5.3	5.3
3	0	0	-28.7	-31.7	-30.5
	1	0	-2.9	-1.8	2.9
	2	0	1.2	5.9	6.1
	3	0	4.7	4.9	4.9

After the DDSPs have been solved, the solution of a master scheduling problem produces a dispatch decision for any state of the original system. For instance, suppose the inventory portion of the state is $(z_1, z_2, z_3) = (0, 1, 0)$. Exhibit 5.9 contains the modified costs C_j^+ for each itinerary j. If only one vehicle is available (state of the system $s = (0, 1, 0; 1)$, in the format of (5.1)), then we would naturally select the itinerary j having the minimum C_j^+, itinerary 17. If two vehicles are available, though, we can dispatch both itineraries 4 and 6 without dispatching twice to one

Exhibit 5.9: Itinerary Dispatch Costs for Inventory State (0,1,0)

j (itinerary)	Delivery			C_j+
	d_{1j}	d_{2j}	d_{3j}	
0	0	0	0	0.0
1	1	0	0	-22.7
2	0	1	0	5.5
3	0	0	1	-20.7
4	2	0	0	-25.0
5	0	2	0	6.9
6	0	0	2	-23.7
7	1	1	0	-20.2
8	1	0	1	-43.4
9	0	1	1	-18.2
10	3	0	0	-23.4
11	0	3	0	12.0
12	0	0	3	-22.5
13	2	1	0	-22.5
14	2	0	1	-45.7
15	0	2	1	-16.8
16	1	2	0	-18.8
17	1	0	2	-46.4
18	0	1	2	-21.2

customer, and the objective value of this combination is -48.7, which would be a little bit better than the objective value -46.4 for itinerary 17 alone. The dispatches for all states of the system, as generated by the heuristic algorithm for the MSP described in Section 4.3.4, are supplied in Exhibit 5.10.

Exhibit 5.10: FVD Algorithm Solution for DDP Example

States		$z_3 = 0$			$z_3 = 1$			$z_3 = 2$			$z_3 = 3$		
z_1	z_2	2	1	0	2	1	0	2	1	0	2	1	0
0	0	16,6	16,0	0,0	16,0	16,0	0,0	16,0	16,0	0,0	16,0	16,0	0,0
0	1	6,4	17,0	0,0	4,0	4,0	0,0	4,0	4,0	0,0	4,0	4,0	0,0
0	2	6,4	17,0	0,0	4,0	4,0	0,0	4,0	4,0	0,0	4,0	4,0	0,0
0	3	6,4	17,0	0,0	4,0	4,0	0,0	4,0	4,0	0,0	4,0	4,0	0,0
1	0	18,0	18,0	0,0	5,0	5,0	0,0	5,0	5,0	0,0	5,0	5,0	0,0
1	1	6,0	6,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
1	2	6,0	6,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
1	3	6,0	6,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
2	0	18,0	18,0	0,0	5,0	5,0	0,0	5,0	5,0	0,0	5,0	5,0	0,0
2	1	6,0	6,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
2	2	6,0	6,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
2	3	6,0	6,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
3	0	18,0	18,0	0,0	5,0	5,0	0,0	5,0	5,0	0,0	5,0	5,0	0,0
3	1	6,0	6,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
3	2	6,0	6,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
3	3	6,0	6,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0

z_i = inventory level of customer i;
subcolumn headings are numbers of vehicles available;
table entries are the itineraries to select for each state

Because the example is not large, we may calculate the true steady-state occupancy probabilities and cost rate for the dispatch policy that the FVD algorithm finds, and compare them to the optimal policy's properties. The average cost per decision point for the FVD heuristic's solution is 19.8, a 3.7% increase above the optimal cost rate. Exhibit 5.11 has the steady-state probabilities for the FVD policy. As with the optimal policy, no customer ever has three units of inventory, and there is always one vehicle available at a decision point. The latter

Exhibit 5.11: Steady-State Probabilities for DDP Example, FVD Solution

States		$z_3 = 0$			$z_3 = 1$			$z_3 = 2$			$z_3 = 3$		
z_1	z_2	2	1	0	2	1	0	2	1	0	2	1	0
0	0	.014	.029	0	.023	.299	0	.001	.015	0	0	0	0
0	1	.058	.295	0	.007	.062	0	0	.002	0	0	0	0
0	2	.003	.015	0	.003	.002	0	0	0	0	0	0	0
0	3	0	0	0	0	0	0	0	0	0	0	0	0
1	0	.061	.002	0	.052	.017	0	.002	.001	0	0	0	0
1	1	.010	.014	0	.005	.003	0	0	0	0	0	0	0
1	2	0	.001	0	0	0	0	0	0	0	0	0	0
1	3	0	0	0	0	0	0	0	0	0	0	0	0
2	0	.003	0	0	.003	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0
2	2	0	0	0	0	0	0	0	0	0	0	0	0
2	3	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0	0	0	0
3	2	0	0	0	0	0	0	0	0	0	0	0	0
3	3	0	0	0	0	0	0	0	0	0	0	0	0

z_i = inventory level of customer i
subcolumn headings are numbers of vehicles available

occurrence arises from the structures of the itinerary set and costs, and not from any conscious effort on the algorithm's part to maintain vehicle availability, since future vehicle availability has no representation in the MSP's objective function. The FVD policy is nonetheless effective, and requires a fraction of the computer time to generate (for the example, the modified policy iteration algorithm took 291 seconds of CPU time on an Apollo DN660 workstation, while the FVD algorithm found its solution in 2 seconds).

Exhibit 5.7 gave the average cost per period incurred by each customer under the optimal replenishment policy. That sum for the example comes to 20.2 cost units per period, a relative error of only 2% from the actual rate of 19.8. Nothing can be generalized from the closeness of this approximation, but its occurrence is encouraging. However, we can apply the results of Section 4.3.6 to determine a lower bound for the optimal cost rate. Using

$$C_{id} = \min_{j \in J_{id}} \frac{d_{ij}}{\sum_{i'=1}^{m} d_{i'j}} C_j ,$$

satisfies in this example both convexity of C_{id} and the condition that the sum of cost shares along any itinerary is less than or equal to the cost of the itinerary. The DDSP results are given in Exhibit 5.12. The sum of the DDSP cost rates comes to 15.7, a figure 18% lower than the known optimal cost rate, so the bound is not very tight in this circumstance. It is noteworthy that the FVD algorithm using C_{id} as above (and $F = 0$) achieved an actual cost rate of 19.4, a mark better than the FVD with averaged cost shares. This observation may indicate that the best cost shares to use in the FVD algorithm are closer to the minimum ones than the averages in this example, or that some itineraries present in the itinerary set but never used in good dispatch policies may be boosting the cost share averages to the detriment of systemwide dispatch selection.

One source of discrepancy in behavior between actual and FVD-transmuted (see Section 4.3.2) dispatching systems is associated with the phenomenon of dispatch failure. The optimal DDSP policies imply that the only time dispatch failure occurs is when each customer has no inventory, because that is the one situation in which each customer wants a replenishment. The steady-state occupancy probabilities for inventory states in each of the customer subproblems are

$$({}_i\pi_0^{\delta,*}, {}_i\pi_1^{\delta,*}, {}_i\pi_2^{\delta,*}, {}_i\pi_3^{\delta,*}) = (0.489, 0.488, 0.023, 0.000)$$

for $i = 1,2,3$; they are the same for each customer since each faces the same demand distribution and uses the same replenishment policy. We may be able to effect an improvement in systemwide

Exhibit 5.12: Solutions of the DDSPs to Obtain a Lower Bound on $g^{\delta*}$

i (customer)	$_ig^{\delta_i*}$	$\delta_i*(0)$	$\delta_i*(1)$	$\delta_i*(2)$	$\delta_i*(3)$
1	5.0	1	0	0	0
2	5.0	1	0	0	0
3	5.7	1	0	0	0

dispatching by running the FVD algorithm with a non-zero failure probability F. Let us approximate F as follows. Suppose all customer inventories vary independently of one another. Then the probability that both customers i', $i' \neq i$, want a replenishment when i wants one is $(0.489)^2 = 0.239$. Further, suppose that at most two customers of the three may receive replenishment, since to replenish all three would leave no vehicles available at the next decision point, which we have already seen is an event that is avoided. Each customer has a 1/3 chance of being left out of the replenishment, so there may be some justification for using an overall failure probability of $(0.239)/3 \approx 0.08$. Exhibit 5.13 shows what happens to the DDSP solutions when $F = 0.08$ and C_{id} is obtained by averaging. The sum of DDSP cost rates here is 22.7, yet the actual cost rate for the resulting systemwide dispatch policy is 19.3, the best FVD solution seen thus far.

We do not aim to try all imaginable combinations of F and $\{C_{id}\}$ in a quest for the best FVD-derived policy, but to gain an idea of how well the FVD algorithm may be able to perform, and of

Exhibit 5.13: Solutions of the DDSPs with F = 0.08

i (customer)	$_ig^{\delta_i*}$	$\delta_i*(0)$	$\delta_i*(1)$	$\delta_i*(2)$	$\delta_i*(3)$
1	7.2	3	0	0	0
2	7.7	2	0	0	0
3	7.8	3	0	0	0

how sensitive it might be to its joint replenishment-simulating factors. The next section provides us with a little more insight into the workings of the proposed heuristic.

5.1.3 Variations to the Example

Let us see how the performance of the various solution approaches compare on two variations to the original instance of the DDP.

The first variation changes the customer demand probability distributions so that the coefficients of variation (CV) are doubled to 0.6. This means that each customer experiences a demand for one unit with probability 0.64, and for zero or two units with probability 0.18 each. We may conjecture that the customers will tend to hold larger inventories for buffering against the greater demand uncertainty, while still suffering from more frequent stocking out. In other words, we should see an increase in average cost per decision point in all disptach policies.

The main results for this example, presented in Exhibit 5.14, confirm these suspicions. On this occasion, the FVD heuristic with C_{id} obtained by averaging and $F=0$ performed best, with cost rate only slightly more than 2% above optimal. The lower bound on the optimal cost rate attainable from the sum of customer cost rates when the minimumn cost share is used for C_{id} is 23.3, only 7% lower than the true optimal in this case. A FVD algorithm policy utilizing a non-zero dispatch failure probability (0.13 here, calculated in much the same manner as in the previous section; no other positive value of F did much better) did not improve upon the respective policy with $F=0$. Recall that a policy of this last type worked best on the original example. Already we observe that the best FVD variation to use on a particular problem depends on the problem characteristics, probably in some unpredictable way.

In the second variation, the coefficient of variation is the same as in the original problem, but the cost of holding a unit of good in inventory has been marked down to one cost unit. Since it has become cheaper to hold inventory, or relatively more expensive to replenish inventory and to lose

Exhibit 5.14: Results for Variation with CV = 0.6

Alg Type	C_{id}	F	Sum of Customer Cost Rates	Overall Cost Rate
Optimal	--	--	--	25.1
FVD	avg	0	25.4	25.7
FVD	min	0	23.3	26.5
FVD	avg	0.13	28.3	26.3

demands, we would expect average inventory levels to rise. Good policies must achieve lower average costs than in the original problem, since the optimal policy from the original problem necessarily would incur a lower average cost here, and may not be optimal.

Exhibit 5.15 provides the alternative policies' results on this second problem variation. Cost rates are lower, as expected. What might not have been anticipated is how close the performances of the policies are to one another. The averaged C_{id}, $F = 0$ FVD policy's cost rate is less than 0.2% worse than the optimal, and the other heuristics about 0.3% worse. Moreover, the lower bound is only 0.4% away from the optimal cost rate. Why the FVD does so well in this instance can be answered best by considering the optimal policy (not shown). It so happens that, in the steady state, most dispatches send vehicles on single-customer itineraries fully loaded. Hence, systemwide replenishment activity corresponds quite closely to what is represented in the single-customer DDSPs, resulting in good approximations for expected future value in the master scheduling problem.

5.1.4 Performance Evaluation by Simulation

Before we proceed to the bigger deliverer dispatch problems, we need to find some way of evaluating the performance of alternative dispatching policies on them. Just as the size of the state space will grow beyond our means for determining the optimal policy, so will it be infeasible

Exhibit 5.15: Results for Variation with $H_i = 1$

Alg Type	C_{id}	F	Sum of Customer Cost Rates	Overall Cost Rate
Optimal	--	--	--	11.68
FVD	avg	0	11.63	11.70
FVD	min	0	11.63	11.72
FVD	avg	0.036	12.58	11.72

to solve the MSP for each state of the DDP and calculate the cost rate of the resultant policy. Our only recourse in this situation, apart from the determination of lower bounds, is to simulate the delivery operation as modeled by the DDP, apply the heuristics, and collect results. In this section, we illustrate, with our original example, the simulation methods to be employed for studying heuristic performance in larger DDP instances.

The design of the simulation model requires no discussion here, because its structure is drawn directly from the dynamic properties of the system inherent in the formulation of the relevant DDP. The potential difficulties we must deal with concern the analysis of the simulation output. Specifically, how long must we run the simulation in order to be fairly confident that the cost averages we obtain reflect the true cost rates? We have chosen the sequential procedure of Law and Carson [16] to assist us in this determination. The procedure is described briefly below.

Two factors that plague simulation analysis are start-up bias and autocorrelation. If our aim is to estimate the steady-state attributes of a system, we must recognize that the initial measurements of the attributes obtained from the simulation may depend more upon the initial system state than on the system's steady-state characteristics. *Start-up bias* refers to the contaminating influence that the simulation's start-up configuration exerts on attribute measurements. This bias tends to diminish as the length of the simulation increases, but should not be assumed to disappear at any particular point in the simulation unless adequate statistical tests are made.

The application of classical statistical techniques is impeded by the usual presence of autocorrelation in the output measures of the simulation. *Autocorrelation* means that statistical correlations exist among values of the output measures taken from near points in time. The estimate of variance of a random variable under the assumption of independence among measurements will be biased if, in fact, the measurements are correlated. Since confidence intervals are constructed with these variance estimates, this is an important problem to be aware of. The Law and Carson procedure is geared primarily toward obtaining unbiased estimates of variance for establishing confidence intervals of a prescribed relative width, presumably under the assumption that any simulation long enough to generate a proper confidence interval will have eliminated any serious start-up bias.

Law and Carson utilize the method of batch means in their procedure. The observations of a single simulation run are grouped into batches, and the means of the batches, rather than the original observations themselves, are tested for autocorrelation. When no substantial positive autocorrelation is found, a confidence interval is calculated. If the first test detects autocorrelation, the data is grouped into larger batches, and another test judges whether to collect more observations or not. If the current dataset is deemed satisfactory, a re-batching takes place and the resulting confidence interval is determined. The simulation is stopped and the most recent point and interval estimates for the attribute used if the relative half-width (ratio of half-width to mean) of the confidence interval falls within some prescribed tolerance. Otherwise, the simulation continues until the total number of observations reaches some new threshold, and the tests are performed again. This process repeats until the confidence interval shrinks to the desired relative width.

We tried this procedure in a simulation of the FVD heuristic with averaged C_{id} and $F=0$ on the original DDP. First we conducted 200 simulation runs, invoking the Law and Carson procedure on each to obtain a 90% confidence interval (CI) for the average cost per decision point. Each run simulated the delivery operation for 800 decision points, ran the tests, and continued as necessary. The initial run length and other parameters (including a batch size of 40 for determining the CI and a relative half-width tolerance of 0.075) were set to the values suggested by Law and Carson. Of the 200 runs, the 90% CI covered the actual cost rate 180 times. Although this result might not seem unusual (180 *is* 90% of 200), empirical studies of sequential procedures often find that CIs fail to cover with frequencies exceeding their intended confidence levels. Inventory systems do seem the best-behaving of the types of systems generally experimented with, so our results might not be too surprising.

Acting on the suspicion that perhaps we were being too cautious in running the simulations initially for 800 periods, we pared down the initial run length to 200 and reduced the final batch size to 20 and looked at another 200 simulation runs. This time, 177 of the runs covered the cost rate, although the CI half-widths were typically a little over 1 cost unit, as compared to the first set's being in the neighborhood of 0.5. Since we will be moving to larger problems, we will adhere to the larger initial run length, since systemwide cost variation per period is likely to be greater.

5.1.5 Partially Adaptive Heuristics

One of the fundamental goals of this research is to demonstrate that basing each dispatch on the state of the system at the time of dispatch (or being "fully adaptive" to state information) works substantially better than establishing schedules that fix some dispatches well in advance of their execution (the "partially adaptive" approach that is the one most commonly adopted). To support this claim we must be able to compare the performances of both the fully and the partially adaptive approaches on specific DDP instances. But we have been looking at only fully adaptive techniques thus far. Below, we propose a class of partially adaptive dispatching techniques and present simulation results pitting members of this class against the fully adaptive methods.

The general method we suggest can be easily applied to the examples we have seen thus far. It is denoted PAD-T; "PAD" stands for "partially adpative decomposition," and the "T" refers to the number of dispatches prepared at a decision point. (T may also be described as the "scheduling horizon," using the terminology of Section 4.3.7.) The first version of PAD-T, outlined below, requires each customer's mean demand per period to be integral (non-integral mean demand is treated later).

0. Obtain the FVD policy. Set the schedule counter $t=T$. Then, for every decision point:

1. If $t<T$, the dispatch $\{y_{jt}\}$ for this decision point has been fixed in advance; execute $\{y_{jt}\}$, set $t=t+1$, and stop. Otherwise, solve the MSP for the current state s_0 to get the dispatch $\{y_{j0}\}$ for the current decision point (indexed 0), execute it, set the schedule counter t to 1, and proceed to step 2.

2. Obtain the projected state of the system s_t at decision point t from s_{t-1} and $\{y_{jt-1}\}$, assuming that the number of units $x_{i,t-1}$ demanded of customer i from $t-1$ to t is equal to customer i's demand mean.

3. Solve the MSP for s_t. Save the resulting dispatch $\{y_{jt}\}$. Set $t=t+1$. If $t<T$, go to step 2. Otherwise, set $t=1$ and stop.

Customer demands in the examples studied thus far are all one, so there is no difficulty in using the first version of PAD-T as is. Exhibit 5.16 and 5.17 compare cost rates from simulation runs (exact evaluation of PAD-T performance would be hard to do) on which the FVD algorithm and PAD-2, PAD-3, and PAD-4 handled the same demand flow. Each run lasted 800 periods. The first four runs were based on the original problem presented in this section, and the second four used the 0.6 coefficient of variation modification. Averages of the cost rates for each set of four

Exhibit 5.16: Simulation Results for Fully vs. Partially Adaptive Decomposition with CV = 0.3

Heuristic	Simulation Run*				Avg	% over FVD
	1	2	3	4		
FVD	19.10	19.36	19.70	19.98	19.54	--
PAD-2	19.75	19.75	20.19	20.48	20.04	2.5
PAD-3	20.12	20.18	20.66	20.75	20.43	4.6
PAD-4	20.23	20.09	20.86	20.97	20.54	5.1

* Table entries are average cost per period based on 800 simulated periods

Exhibit 5.17: Simulation Results for Fully vs. Partially Adaptive Decomposition with CV = 0.6

Heuristic	Simulation Run*				Avg	% over FVD
	1	2	3	4		
FVD	26.26	25.33	26.27	26.99	26.21	--
PAD-2	28.97	28.63	29.49	29.43	29.13	11
PAD-3	29.94	29.14	30.14	30.36	29.90	14
PAD-4	30.51	29.89	30.41	30.49	30.33	16

* Table entries are average cost per period based on 800 simulated periods

runs are also shown, as well as the percentage that average lies above the FVD policy's average. Two trends may be detected in the results:

1) As T, the length of the scheduling horizon, increases, the dispatch performance of PAD-T deteriorates.

2) As the degree of demand uncertainty grows, PAD-T's performance level decreases.

These trends could well have been anticipated before examining the simulation results. However, it is interesting to note the magnitudes of the performance deterioration. For the case where demand variation is lower, the average cost rates are all close, with PAD-4 only 5% above the FVD average. On the other hand, even PAD-2 scores over 10% worse than the fully adaptive FVD when demand uncertainty is set at the higher level. When one considers that, in this scenario, planning four rather than one dispatch decisions at a time brings a 15% reduction in dispatching quality, and that that result was obtained with a relatively good partially adaptive technique (since it is based on the FVD algorithm), one begins to realize what kind of cost savings may be achieved with the implementation of a CAD system equipped with a fully adaptive dispatching algorithm, when the deliverer must cope with demand uncertainty. We should therefore be sensitive to these trends as we analyze dispatching performance in larger problems.

An extreme case of the partially adaptive approach is an algorithm that follows a fixed, cyclic schedule, where dispatches are made without regard to any state information. This would be termed a "non-adaptive" approach. Two cyclic schedules (labeled CYCLE-1 and CYCLE-2) for the original DDP example are given in Exhibit 5.18. They were drawn from good fully-adaptive dispatch policies by looking for cycling state occupancy patterns under the assumption of deterministic demand. We start, for instance, with the information about the FVD policy for averaged C_{id} and $F=0$ contained in Exhibits 5.10 and 5.11. The state occupied most frequently is state (0,0,1;1). The policy says to dispatch the available vehicle on itinerary 16 here. Under that dispatch and one unit demanded from each customer, the next state entered is (0,1,0;1), not coincidentally the second most occupied state. Now the policy has itinerary 17 dispatched, and if demand is one unit per customer, the system winds up in state (0,0,1;1) again. Hence, one cyclic schedule that may work well is to alternately dispatch itineraries 16 and 17. This is the policy CYCLE-1. CYCLE-2 was derived from another FVD-based policy. The simulation results for these and the averaged C_{id}, $F=0$ fully adaptive FVD algorithm are displayed in Exhibit 5.19. It is evident that these non-adaptive policies are far inferior to the fully adaptive heuristic policy for this problem. We suspect that the same will be true for all problems except perhaps those in

Exhibit 5.18: Two Cyclic Schedules

Schedule	Period 1	Period 2	Period 3	Period 4
CYCLE-1	16,0	17,0	(cycle is 2	periods)
CYCLE-2	0,0	6,16	4,0	6,16

Exhibit 5.19: Simulation Results for Cyclic Schedules

Heuristic	Simulation Run*				Avg	% over FVD
	1	2	3	4		
FVD	20.17	19.66	19.45	19.94	19.81	--
CYCLE-1	31.10	33.03	31.07	32.34	31.89	61
CYCLE-2	29.87	31.58	29.70	30.71	30.47	54

* Table entries are average cost per period based on 2000 simulated periods

which demand is nearly or totally deterministic. If demand were totally deterministic, then a method such as the one described in [23] could be employed to generate cyclic schedules.

Lastly we must deal with the case where demand means are not integral. We suggest two approaches. The first one amends the algorithm in the following ways:

Step 1. Also set for each customer i the demand counter X_i to $T\mu_i$, where μ_i is the mean of demand for customer i.

Step 2. Determine demand $x_{i,t-1}$ from

$$x_{i,t-1} = \left[\frac{X_i}{T-(t-1)} \right],$$

where $[z]$ is the closest integer to z.

Step 3. Also update X_i via $X_i = X_i - x_{i,t-1}$.

Note that when μ_i is an integer, this algorithm works exactly as does the original PAD-T.

By projecting demands in the same way for every scheduling horizon, as the first approach does, one may introduce some form of bias in the dispatching procedure. The second approach attempts to mitigate this bias by randomizing the demand projection method. It works the same way as does the first approach, except for the projection of x_{it} in step 2. Letting

$$e = \left\{ \frac{X_i}{T-(t-1)} \right\},$$

and

$$f = \frac{X_i}{T-(t-1)} - e,$$

where $\{z\}$ is the greatest integer less than or equal to z, x_{it} is found as follows:

$$\begin{aligned} x_{it} &= e \text{ with probability } 1\text{-}f, \\ &= e+1 \text{ with probability } f. \end{aligned}$$

Computational results using these methods will be presented in Section 5.4.

5.2 Aims of the Computational Experiments

The goals of the first part of this dissertation are twofold: to demonstrate the computational viability of fully adaptive heuristics for the DDP, and to convey a sense of the potential cost savings achievable by switching from partially to fully adaptive dispatch support technology. The first goal has largely been achieved by the formulation of the FVD algorithm. However, we have yet to confirm that the method performs well, or at least significantly better than naive methods, in more complex settings. The remainder of this chapter documents computational experiments involving the FVD algorithm in application to larger DDPs (more customers, more vehicles). One aim of these experiments is to evaluate the algorithm's performance, both in quality and in computational speed, when dealing with the larger problems. We also pursue our second goal here by comparing the dispatching performance of the PAD-T heuristics on the same problem instances. The PAD-T heuristics serve as surrogates for the various techniques that are typically employed to set up delivery schedules in the class of operations modeled suitably by the DDP. By measuring the differences in quality between the FVD algorithm and the PAD-T heuristics over the set of DDP instances studied, we hope to gain some notion about both the magnitude of savings afforded by the fully adaptive approach, and the factors which contribute to greater or smaller savings than expected in particular problems.

Results for a myopic algorithm corresponding to the minimization of expected costs during the upcoming period, with no penalties factored into the objective function, are included for further comparison and discussion.

5.3 The Set of Larger Problem Instances

Fourteen instances of the deliverer dispatch problem containing more customers and more vehicles than did the example studied earlier provide the basis for our computational assessment of the methodology we have developed. These instances were generated largely at random, although the author needed to intervene in the establishment of certain problem characteristics. This section relates the manner in which the test instances were generated. The data for each instance appear in Appendix B.

Instance sizes: Initially, seven instances of six customers each, and labeled 1 through 7, were produced. A 12-customer instance *I*x was adapted from each 6-customer instance *I* by changing the customer locations, the vehicle fleet size, and the customer demand distributions. Further discussion of the selection of the latter two problem characteristics appears later.

Holding capacities: In each instance, all customers have the same inventory holding capacity. The capacities vary from problem to problem, but all were picked in advance by the author.

Demand distributions: In every instance generated, customer demands are assumed independent of one another. Also, each customer's demands are taken as independent from one period to another. In a given instance, customer demand distributions are such that each distribution has the same coefficient of variation (CV) and the same maximum number of units that can be demanded per period. The CVs and the maximum demand quantities of each problem were selected by the author to yield a good mix of problem characteristics. Each customer's mean demand was drawn initially at random. Sometimes this mean level was adjusted to achieve pre-selected targets of aggregate mean demand in a scenario, and sometimes the mean level was rejected for another because no demand distribution having the necessary mean and CV attributes could be found in a limited search. The CVs of some of the distributions do not match exactly the CV target of the instances they are used in, owing to the same difficulty in locating a distribution that matched both qualities exactly. Demand distributions were unimodal for the most part.

For the 12-customer problem associated with a given 6-customer problem, the demand distribution for each customer was selected at random from distributions used in any of the 6-customer problems having the same CV and maximum demand. In this way, we avoided having to search for new demand distributions.

Vehicle fleet characteristics: The order of establishment of vehicle fleet characteristics ran as follows: Vehicle transport capacities, identical for each vehicle in the fleet of a given instance, were assigned by the author for each instance in the set of 6-customer problems. Then, a "utilization factor" was generated randomly for each instance. The utilization factor was intended to measure the ratio of systemwide mean demand per period to systemwide mean quantity deliverable per period. With the utilization factor and mean demand having been determined, a target was established for the mean quantity deliverable. Expressing this quantity as

$$\frac{K\Omega}{\tau_a},$$

where K is the vehicle fleet size, Ω the vehicle capacity, and τ_a some targeted average itinerary duration, the integral vehicle fleet size K was obtained that allowed the average itinerary duration and utilization factor to best meet their targets. The utilization factors were drawn from the range (0.5, 0.95) in such a way that their logarithms were distributed uniformly (the utilization factors are referred to as being distributed "log-uniformly"). The purpose of having the logarithm distributed uniformly was to skew the distribution of the factors themselves toward the values which we were most interested in, while allowing a couple of instances with other values to be simulated, too. (Note: the actual utilization rates of vehicles differed significantly from the factors employed to select the fleet sizes.)

The actual utilization rates from the set of 6-customer instances were examined for signs of over- or under-utilization of vehicles. Vehicle fleet sizes for the corresponding 12-customer problems were chosen intuitively with the intent to avoid extremes in vehicle utilization (less than 50% or nearly 100% of the fleet transport capacity used over the simulation). Our intuition proved faulty in many instances, as vehicle fleets tended to be over-utilized with respect to our intentions. In these instances, we looked at alternate scenarios where more vehicles were available, in part to test the sensitivity of algorithmic performance to changes in the amount of delivery resources.

Itineraries: In each instance (both 6-customer and 12-customer), customers were distributed uniformly over the unit square, and the depot location was drawn from a uniform distribution over the 0.5x0.5 square concentric to the unit square. Customers were clustered visually into groups of two to five, with some customers belonging to more than one group. All one-, two-, and three-customer routes ("route" just referring to the subset of customers visited in an itinerary) were generated such that all customers in a route belonged to a common group. All itineraries delivering from 2 to Ω units to the customers in a route were included in the itinerary set for a particular problem. The cost of each itinerary was determined by the function

$$C_j = c\left(E_j + 0.25\, m_j\right), \tag{5.2}$$

where E_j is the Euclidean distance of the optimal traveling salesman tour through the depot and the customers in itinerary j, m_j the number of customers in the itinerary plus one, and c a randomly selected (log-uniformly distributed) unit cost of transportation. The cost function models costs as accruing linearly over the time taken to execute an itinerary, where time is spent both in travel and in delivery activities. In the problems studied, c ranged from 0.8 to 21. Transportation cost was used as the basis of itinerary duration determination. Thresholds for itinerary duration in terms of cost were chosen in an effort to produce an average itinerary duration that would come close to the value targeted in selecting the vehicle fleet size. All leadtimes were set to zero.

Inventory costs: Inventory holding costs were one cost unit per unit of inventory held per period in every scenario. The cost per unit of lost demand was drawn from a log-uniform distribution. The values of this cost appearing in this study range from 6 to 75, skewed toward the lower end.

5.4 Presentation of the Results

For each instance designed using the methods of the previous section, we ran at least one simulation in which these five heuristic algorithms were used for dispatching in response to a common demand pattern:

1) the FVD algorithm with cost shares C_{id} obtained by averaging over equally-weighted itineraries and dispatch failure probability $F=0$;
2) a PAD-2 heuristic based on the penalties used in the FVD algorithm (1) above;
3) a PAD-4 heuristic based on the penalties used in the FVD algorithm (1) above;
4) a fully-adaptive (dispatch decided at each decision point) myopic heuristic to minimize expected costs in the upcoming period with no penalties attached to states likely to be entered at the next decision point;

5) an updated FVD algorithm, where cost shares are obtained by weighting the relevant itineraries by their frequencies of dispatch using algorithm (1) in a previous simulation run of the same instance, and the dispatch failure probability is set to the frequency with which a customer who would receive a replenishment at a decision point according to its DDSP was not included in any dispatched itinerary then (refer to Sections 4.3.7.1-4.3.7.3 for a fuller discussion of updating).

The six-customer instances ran anywhere from 800 to 3200 periods, and the twelve-customer ones from 200 to 800. Each simulation was terminated according to the Law and Carson procedure discussed in [16] and Section 5.1.4 of this dissertation. But 6-customer and 12-customer simulations that had not terminated according to this procedure by 3200 or 800 periods, respectively, were stopped and statistics calculated. The simulation control parameters are listed in Exhibit 5.20. Uniformly distributed random numbers were obtained using the routine listed in [27]. The complete computer program appears in [20].

Exhibit 5.21 summarizes the simulation results for the fourteen instances. This table shows the average cost per period using each of the five algorithms. (In some of the twelve-customer instances, when several vehicle fleet sizes were tried, one representative run was chosen for presentation; the other results are presented when we study sensitivity to vehicle fleet sizes below.) It is clear that the fully adaptive FVD algorithm (initial or updated) is superior to the partially adaptive and myopic heuristics. The degree of this superiority varies from problem to problem, though. The rest of this chapter compares and contrasts the simulation results across a number of different dimensions.

5.4.1 Fully vs. Partially Adaptive Heuristics

The fully adaptive FVD heuristic outperformed the PAD-4 algorithm in every instance, and the PAD-2 in every instance but one. Even in the case where PAD-2 had lower average simulated cost, the difference was not significant; in fact, on the run used to generate itinerary weights for

Exhibit 5.20: Simulation Control Parameters

Parameter*	Size of problem (# of customers)	
	6	12
final number of batches (k)	40	20
correlation test initial batch size (m)	2	2
initial simulation run length (n_1)	800	200
relative half-width tolerance (γ)	.075	.15

* symbols in parentheses correspond to those used in [16]

Exhibit 5.21: Average Simulated Costs of DDP Heuristics

Instance	Initial FVD	PAD-2	PAD-4	Myopic	Updated FVD
1	59.4	58.4	66.4	70.0	44.0
2	34.6	35.3	36.1	40.4	34.6
3	81.0	82.3	85.2	131.9	80.7
4	10.9	13.6	16.7	10.4	10.3
5	29.0	33.0	41.2	63.5	29.0
6	12.0	14.0	16.6	12.5	12.2
7	70.9	74.0	75.8	89.8	70.9
1x*	99.5	101.8	104.9	140.5	97.5
2x	108.2	108.6	109.4	118.5	108.1
3x	179.3	180.5	183.6	246.1	171.0
4x	19.7	27.4	32.0	19.7	18.7
5x	49.9	60.2	75.4	69.6	51.2
6x	37.2	38.0	41.3	40.1	31.9
7x	135.2	138.5	143.4	167.8	134.6

* CV not same for Instance 1x and Instance 1

the updated FVD algorithm, the initial FVD algorithm did better than PAD-2, 56.7 to 57.6, on a common demand pattern. We therefore judge the fully adaptive FVD algorithm to be at least marginally, if not substantially, better than good adaptive techniques.

Still unanswered is the question of whether the benefit of improved dispatching performance achievable by switching from partially to fully adaptive dispatching methods outweighs the costs of more frequent information gathering and acquisition of the means by which to make good fully adaptive dispatches. The answer will vary from situation to situation, since it depends on a number of factors. One important factor brought out by the simulation results is the degree of demand uncertainty in the system, as measured by the coefficient of variation of demand. Exhibit 5.22 shows for each instance the decline in dispatching quality of the PAD-2 and PAD-4 heuristics relative to the FVD algorithm. The decline is expressed as a percentage increase in average simulated cost. Also provided are the prevailing CVs in each instance. For PAD-2, which represents updating the information base at every other decision point, the percentage increases above the FVD cost range from -1.7% to 16.7% for instances with CV=0.3, and from 2.3% to 39.1% when CV=0.6. More to the point, the median increase is 1.6% over DDP instances with CV=0.3, and 13.8% with CV=0.6. The respective ranges and medians are even more pronounced for the PAD-4 heuristic. For CV=0.3, percentage increases extend from 1.1% to 38.3%, with median 5.2%, and for CV=0.6, from 5.4% to 62.4%, with median 42.1%. Dispatch performance is thus observed to deteriorate rapidly as the frequency of information base update decreases.

We presume that in real-world operations, if an inventory database is not updated at every dispatch decision point (or whatever this corresponds to in the actual operation), it seems likelier that it would be updated every fourth decision point or more, rather than every other one. Also, we have reason to believe that partially adaptive dispatching configured around the FVD algorithm performs better than whatever partially adaptive methods are currently employed in practice, since allowing the system to enter undesirable future states is still explicitly penalized in the algorithm's objective function. We may venture to say that a fourfold increase in

Exhibit 5.22: Increase in Average Simulated Costs for Partially Adaptive Heuristics

Instance	PAD-2	PAD-4	CV
1	-1.7	11.8	0.3
2	2.0	4.3	0.3
3	1.6	5.2	0.3
4	24.8	53.2	0.6
5	13.8	42.1	0.6
6	16.7	38.3	0.3
7	4.4	6.9	0.6
1x	2.3	5.4	0.6
2x	0.4	1.1	0.3
3x	0.7	2.4	0.3
4x	39.1	62.4	0.6
5x	20.6	51.1	0.6
6x	2.2	11.0	0.3
7x	2.4	6.1	0.6

Table figures are percentage increases in average simulated cost of the partially adaptive heuristics relative to the (initial) FVD algorithm for the same demand flow.

information base update frequency may be counted on to induce a savings of 5% in distribution and inventory costs unless future demand is nearly or entirely known beforehand. The savings rate may approach 10%, 20%, or more when demand uncertainty is moderate to high. Of course, information collection frequency is not the sole determinant of dispatching quality; that information must be acted upon by an effective dispatching procedure.

5.4.2 Myopic vs. Long-Range Planning

We acknowledge that the representation of future dispatching activity embodied in the deliverer dispatch subproblems, from which we derive the FVD algorithm's penalty function, tends to be flawed. This prompts the question: Is it worth going to the bother of calculating penalties? In other words, does the FVD algorithm improve upon fully adaptive, myopic dispatching? A comparison of the "initial FVD" and "myopic" columns of Exhibit 5.21 leads us to answer, with firm conviction, in the affirmative.

Exhibit 5.23 highlights the loss of dispatch quality after an exchange of the initial FVD algorithm for the myopic algorithm for dispatching. The median percentage increase in simulated per period cost of the myopic against the FVD algorithm is 21%. The spectrum of actual dispatching quality declines is very wide. In one scenario (Instance 4), the myopic algorithm actually outperformed the initial FVD (although not the updated FVD), and in that scenario's extended version (Instance 4x), performances were about equal. Yet in another, (Instance 5), the myopic algorithm suffered twice the costs of the FVD algorithm. Demand coefficients of variation do not seem to explain why the myopic algorithm does well in several instances and poorly in most of the others. Added to Exhibit 5.23 are other potential explanatory factors, the cost of a lost demand L and the transportation cost multiplier c (as employed in expression (5.2)). The best myopic performances occurred when c was at its lowest values (Instances 4 and 6 and extensions), and when L attained its lowest value (Instance 2). On the other side of the coin, the worst performances relative to the FVD algorithm took place when either L or c was at its highest point (Instances 3 and 5).

The foregoing results seem to indicate that when delivering can be done cheaply, or when the penalty for not filling a demand is relatively low, myopic procedures may be adequate for dispatching purposes. We offer the following rationale for this conjecture. By dispatching a vehicle, the system immediately incurs transportation costs. Further, under the zero leadtime assumption, inventories at the end of the period are higher when several vehicles are dispatched than they would be if no vehicles were. Therefore, the only gain to be realized in the immediate period by dispatching is the reduction of expected lost demand costs. But dispatching is only likely to make a significant impact on lost demand for the immediate period when inventories are very low, lower than might instigate dispatching in near-optimal policies. Hence, using myopic

Exhibit 5.23: Comparison of FVD and Myopic Algorithms

Instance	% increase in total cost from FVD to Myopic	Cost per unit of lost demand	Transport cost multiplier	% decrease in holding cost	% increase in lost demand cost
1	17.8	15	7.8	48.1	83.2
2	16.8	6	10.0	753.2	89.0
3	62.8	30	21.0	1364.5	91.1
4	-4.6	20	0.8	-20.9	-5.3
5	119.0	75	4.8	929.4	53.9
6	4.2	20	1.3	173.6	16.6
7	26.7	10	13.5	524.5	64.3
1x	41.2	15	7.8	307.2	77.1
2x	9.5	6	10.0	80.0	91.5
3x	37.3	30	21.0	377.5	85.3
4x	0	20	0.8	-0.7	-1.6
5x	39.5	75	4.8	500.0	46.9
6x	7.8	20	1.3	22.6	26.8
7x	24.1	10	13.5	242.2	58.9

dispatching, we might expect to obtain lower inventory holding costs and higher lost demand costs than we would by using the FVD algorithm. The overall difference between myopic and long-range planning may be small under these circumstances: If the cost of a lost demand is especially small, the impact of the higher lost demand risk is less than the norm, perhaps small enough to balance savings in holding and possibly transportation costs. If transportation costs are expecially low, the return a dispatch must show in terms of lost demand risk reduction does not need to be so great, so that dispatching at higher inventory levels may occur.

The preceding suppositions are supported by the simulation results. Exhibit 5.23 has columns for the simulated average inventory holding and lost demand costs per period for the myopic heuristic, in terms of their percentage decrease and increase, respectively, relative to the corresponding cost using the initial FVD algorithm. When transportation costs were their smallest, holding and lost demand costs went opposite to their overall trend of being substantially lower and higher, respectively, than their FVD counterparts. Although the trend was upheld when the cost of a lost demand was small, the net effect on total costs was not as great as in some of the other scenarios. The difficulty with applying the above conjectures about when myopic dispatching is not much worse than dispatching via the FVD heuristic is that it is hard to judge *a priori* when transportation or lost demand costs are low enough for the conjectures to hold. Therefore, we wholeheartedly endorse the type of long-range planning that the FVD algorithm performs.

5.4.3 Initial vs. Updated FVD Algorithms

In Section 4.3.7.3, we suggested updating the cost shares C_{id} and dispatch failure probability F of the DDSPs once better information about the relative frequency with which each itinerary is dispatched comes available, namely after the algorithm is implemented and has been used for a while. The pertinent question here is, does updating make a difference? Preliminary indications read from our simulation experiments are that is does to some extent, but rarely does it turn a bad procedure into a good one, or a good one into a great one (see Instances 1 and 6x). Exhibit 5.24 recapitulates the outcomes of our experiments.

Two simulation outputs are enlisted to help explain the observed updating effects. The values of these outputs for each instance are included in Exhibit 5.24. These particular outputs were selected because they indicate to what degree the perceptions of the FVD algorithm about future dispatches are misguided. Recall that the FVD algorithm works from the assumption that in the future, each customer always receives its optimal replenishment quantity as specified in the solution to its DDSP (since F is set to zero on the preliminary simulation run to get the itinerary dispatch frequencies and dispatch failure proportion), and that it is visited in a single-customer

Exhibit 5.24: Comparison of Initial and Updated FVD Algorithms

Instance	% decrease in total cost from Initial to Updated	Dispatch failure proportion*	Proportion of single-customer itineraries*
1	25.9	.20	.42
2	0	0	1.00
3	0.4	.09	.96
4	5.5	.15	.47
5	0	.03	.84
6	-1.7	0	.05
7	0	.03	1.00
1x	2.0	.18	.80
2x	0.1	.36	.99
3x	4.6	.11	.83
4x	5.1	.12	.14
5x	-2.6	0	.41
6x	14.2	.04	.13
7x	0.4	.18	.90

* as measured on preliminary initial FVD run

itinerary. The second column of Exhibit 5.24 tells what fraction of the time in each instance (in the preliminary runs) a customer anticipated a replenishment being sent and none was. We have termed this fraction the "dispatch failure proportion." It quantifies the degree of violation of the first FVD assumption listed above. The last column of the exhibit shows what fraction of dispatched itineraries were single-customer itineraries. A customer can receive its optimal (with respect to its DDSP) delivery quantity on a multiple customer route, if the delivery is less than a vehicle's capacity. However, it may also receive less than the target quantity. This event has not been accredited as a dispatch failure; nevertheless, it breaks the correspondence between actual operations and those in the perception of the FVD algorithm.

Our search for relationships between the improvement percentage of updating and the DDSP violation measures can be counted only as partly successful. In the six instances where updating has the most positive impact, the fraction of single-customer itineraries dispatched is greater than 0.8 in only one of them. Yet the two cases where the updated FVD algorithm performed worse than the original had this fraction well below 0.8 also. There appeared no clear relationship between the measure of fraction of replenishments not to arrive and the level of improvement due to updating. We did not try additional updates to seek further dispatching improvement. We stop here by intimating that occasional updating may be valuable in finetuning dispatch performance, but no quantum leaps in performance should be expected through updating We may add that the "quantum leap" has in most cases already been made by moving to the FVD algorithm, that is to say, by appending realistic penalty functions to the finite horizon dispatching problem objective functions.

5.4.4 Uses of DDSP Cost Rates

The cost rates generated in the solutions of the customer DDSPs can be put to several uses, in ways described earlier. The primary use is in constructing a lower bound for the average cost per period incurred with the FVD algorithm; refer to Section 4.3.6 to review how. Exhibit 5.25 supplies the lower bound for each instance, and marks the percentage decrease to it from the average cost of the initial FVD algorithm. The DDSP violation measures from Exhibit 5.24 are repeated here for explanatory purposes. A third factor is appended to the table. It is the mean vehicle loading percentage, an average of the proportion of vehicle capacity utilized over all dispatches, each dispatched itinerary weighted by its duration.

The correlation between percentage of single-customer itineraries dispatched and percentage gap between the actual cost rate of the FVD algorithm and its lower bound is evident. The same is not true of the percentage of desired replenishments not delivered, though. For each instance in which the mean vehicle loading proportion is less than one, the lower bound was markedly

Exhibit 5.25: Comparison of Initial FVD Performance with Lower Bound

Instance	Lower Bound (LB)	% decrease from FVD cost to LB	Dispatch failure prop.	Prop. of single-customer itin's	Mean vehicle loading %	% decrease from updated FVD to LB
1	39.6	33.3	.20	.42	100	10.0
2	33.9	2.0	0	1.00	100	2.0
3	79.7	1.6	.09	.96	100	1.2
4	6.1	44.0	.15	.47	76	40.8
5	24.9	14.1	.03	.84	100	14.1
6	10.2	15.0	0	.05	85	16.4
7	69.6	1.8	.03	1.00	100	1.8
1x	92.8	6.7	.18	.80	100	4.8
2x	107.4	0.7	.36	.99	100	0.6
3x	167.9	6.4	.11	.83	100	1.8
4x	12.6	36.0	.12	.14	89	32.6
5x	45.3	9.2	0	.41	100	11.5
6x	20.1	46.0	.04	.13	92	37.0
7x	126.0	6.8	.18	.90	100	6.4

lower than the average cost of the FVD algorithm; the causes behind this observation are unclear. The disparity between the average cost of a heuristic and the lower bound is made up of two sub-disparities: the degree of suboptimality of the heuristic, and the underestimate of the optimal value by the lower bound. Updating makes up some of the total disparity in the instances with the most widely separated average cost/lower bound pairs, but much remains. Further investigation is necessary to explain the relationships among the contributory factors cited here and the lower bound/actual performance disparity.

The customer cost rates from the DDSPs can be summed to provide a guess of the average cost per period of the resultant algorithm. We compare predicted and actual costs for both the initial and the updated FVD algorithms in Exhibit 5.26. The predictions for the initial FVD's performance lie closer to the actual values than do the lower bounds thereof, but not that much closer. Almost all predictions here were underestimates (understandable, since no dispatch failures are predicted). Contrast this with predicted vs. actual average costs for the updated FVD. About half the predictions underestimate average costs, and half overestimate it. Also, the predictions lie much closer to the simulated values. The median absolute percentage deviation of the predicted from the actual value in the updated case is 3.2%, as opposed to 6.0% for the initial FVD. Especially when dealing with the updated FVD heuristic, then, the quality of cost rate predictions seems to be very good. The capability of predicting average costs may come to be quite useful for strategic planning purposes.

5.4.5 Sensitivity to Vehicle Fleet Capacity

Misjudgments in the appraisal of vehicle fleet sizes sufficient for delivery needs in the twelve-customer problems led us to compare alternative fleet sizes for some instances. Average cost results for the updated FVD algorithm with the different fleet sizes examined are presented in Exhibit 5.27. Also printed for each scenario is the capacity utilization rate of the fleet, i.e. the mean proportion of vehicle transport capacity in use over the simulation run. In these scenarios, utilization for the first fleet size value tried was at or near 100%, and lower bounds were substantially lower than the average cost (except in Instance 2x). In Instance 5x, increasing the number of vehicles from 4 to 6 brought nearly a six-fold reduction in average costs. Yet in Instances 2x and 3x, boosting the fleet size actually induced an *increase* in average costs, though not necessarily a statistically significant one. Perhaps with more vehicles available, the algorithm becomes "dispatch-happy," not realizing that the incremental inventory cost savings of an extra dispatch does not outweigh the added transportation cost. Utilization always dropped as

Exhibit 5.26: Comparison of Predicted and Actual Cost Rates in the FVD Algorithms

Instance	Cost rate predicted from DDSPs, initial FVD	% decrease from actual to predicted cost rate	Cost rate predicted from DDSPs, updated FVD	% decrease from actual to predicted cost rate
1	39.8	33.0	46.9	-6.6
2	34.6	0	34.6	0
3	79.7	1.6	82.8	-2.6
4	6.6	39.4	12.0	-16.5
5	26.9	7.2	28.1	3.1
6	11.8	1.7	11.7	4.1
7	69.9	1.4	70.5	0.6
1x	92.8	6.7	98.3	-0.8
2x	107.7	0.5	111.4	-3.1
3x	167.9	6.4	176.5	-3.2
4x	13.8	29.9	22.7	-21.4
5x	51.1	-2.4	48.1	6.1
6x	23.7	36.3	30.5	4.4
7x	127.6	5.6	135.1	-0.4

more vehicles were added, but the rate of the drop varied in unpredictable ways from instance to instance. If we were to keep adding vehicles, the cost rate could approach its lower bound, but this depends on the itinerary cost structure. Vehicle fleet capacity effects on FVD dispatching is an interesting topic for further research.

5.4.6 Extended Scheduling Horizon

In Section 4.3.7.4, the idea of employing an extended scheduling horizon in the FVD algorithm was discussed, but technical details were for the most part avoided in order to limit the scope of the algorithmic development. The absence of details notwithstanding, FVD algorithm

Exhibit 5.27: Sensitivity of the Updated FVD Algorithm to Vehicle Fleet Size, Selected Instances

Instance	Fleet size	Simulated average cost	Vehicle capacity utilization %
2x	6	108.1	99.8
	8	108.2	95.8
3x	7	234.9	100.0
	8	171.0	93.6
	9	175.6	86.7
5x	4	326.4	100.0
	5	129.5	100.0
	6	51.2	78.3
7x	6	134.6	97.8
	7	129.3	89.1

prototypes operating with an extended scheduling horizon were applied to a few of the instances studied heretofore. In Instance 1, average costs for the initial FVD algorithm were reduced from 59.4 to 48.8, albeit on different simulation runs. In Instance 5x with a vehicle fleet size of 4, the average cost rate shrank from 326.4 to 212.5 for the updated FVD algorithm when the scheduling horizon was extended. The horizon extension in each case was to two periods. We infer that considering the vehicle availability consequences of decisions even just one period into the future can have a tremendous impact on dispatching performance when vehicle utilization is heavy. Continued study of extended scheduling horizons is highly recommended. (Note: extended scheduling horizons are used extensively in Part II of this dissertation.)

5.4.7 Computation Time

The final operating characteristic of the FVD algorithm that we need to examine is its speed of computation. No matter how good the dispatches suggested by the algorithm are, we will not have achieved our objective of demonstrating the feasibility of implementing the algorithm in a CAD system if it takes too long to generate dispatches. The CAD system must be able to respond quickly to a dispatcher's command in order for the dispatcher to consider it a useful decision support tool. This section relates some of our findings about the computation times of the FVD algorithm.

For each scenario, we ran a simulation 100 periods long for the purpose of measuring computation time. The sole algorithm active in this simulation was the initial FVD algorithm. We timed both the DDSP solution phase and the elapsed time of the actual simulation. The latter quantity included statistics gathering and state update operations as well as the solution of the MSP to obtain the dispatch at each decision point. The computation times of each of these phases are listed in Exhibit 5.28. As with the small example studied in the beginning of this chapter, all computations were performed on an Apollo DN660 computer. The times in the exhibit are in units of CPU seconds.

The first column of Exhibit 5.27 shows the time taken to solve the DDSPs. The DDSPs tend to be solved very quickly, somewhere on the order of 0.1 CPU seconds per customer. This being the case, frequent updating of the penalty functions may be quite feasible, especially when the time to do an update is compared to the time to make a dispatch decision. While in the 6-customer case, each decision was generated in a little over 0.03 CPU seconds, on average, the mean elapsed time per decision for twelve customers was on the order of 0.75 CPU seconds. This is a 25-fold increase in response to a doubling of customers (with associated doubling or near-doubling of vehicles). It may also be an indication that, if the FVD algorithm is to be applied to larger problems, a faster heuristic must be implemented to solve the MSP. The reader may consult Section 4.3.4 to review the classes of heuristics that have been studied in the operations research literature to solve problems of a form similar to the MSP. Customer and itinerary elimination procedures may also help reduce the computation time for a dispatch decision in the FVD algorithm. More research is called for in the development of efficient MSP heuristics.

Exhibit 5.28: Computation Times for the Initial FVD Algorithm

Instance	DDSP solutions	Simulation, 100 periods long
1	1.07	2.12
2	1.02	3.40
3	.74	3.20
4	.45	1.23
5	.68	1.96
6	.47	3.25
7	.48	3.18
1x	.81	89.91
2x	1.24	59.82
3x	1.20	75.75
4x	.88	46.61
5x	1.50	43.50
6x	.95	139.64
7x	1.08	107.20

Table entries are in CPU seconds on an Apollo DN660 computer.

PART II

CASE STUDY:

THE NEW YORK CITY DEPARTMENT OF SANITATION'S

MARINE WASTE TRANSPORT SYSTEM

CHAPTER VI

A MODEL OF THE MARINE WASTE TRANSPORT SYSTEM

6.0 Introduction

This chapter presents a mathematical model of the New York City Department of Sanitation's marine waste transport system (MWTS). The model simplifies many of the MWTS's features. We believe that, even in its streamlined form, the model depicts marine waste transport operations realistically enough to serve as a key component in a viable decision support system for tug and refuse barge dispatching. Significant departures from reality in the model are nonetheless duly noted in the course of the presentation.

6.1. Model Terminology

The marine waste transport system model we construct is in essence a stochastic decision process. To wit, many components of the model evolve over time, as a result of both exogeneous stochastic factors and endogeneous decisions. We also permit certain key parameters affecting the evolution of the system to change in value with time, hence the process is not time-homogeneous. For the sake of convenience, the time dimension is discretized into units we shall term *pulses*.

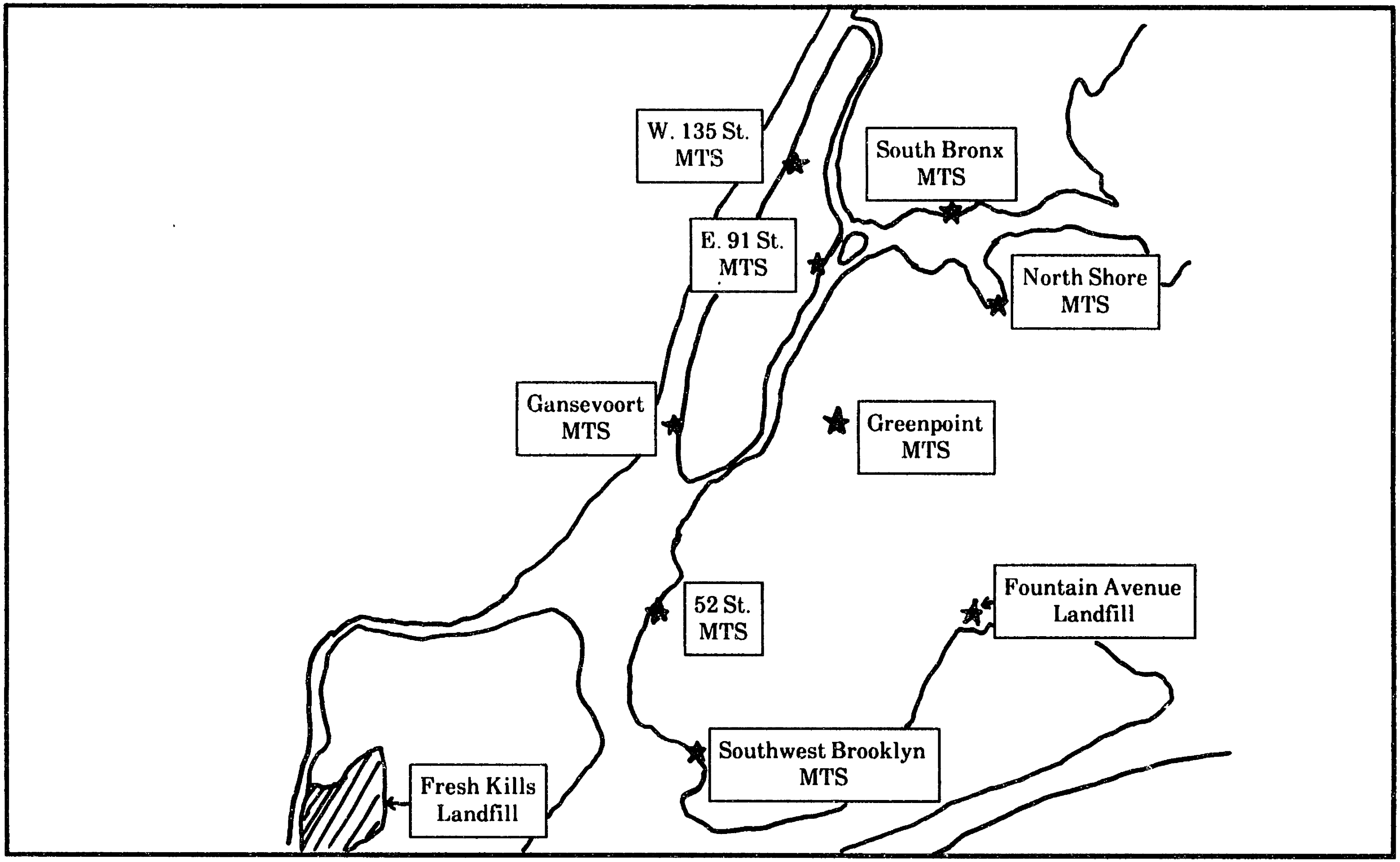
Implicit in the notion of a decision process is that somewhere embedded in the process is a decision-maker who is trying to make good decisions. At any given point in time when a decision may be made, the decision-maker takes all the information about the system that he finds useful and comes up with a decision. The effect of that decision unfolds thereafter, although this effect may be confounded with the effects of other decisions and with stochastic phenomena.

We mark the progress of the evolution of the system at any point in time as the system *state*. The state is basically a body of time-varying information about the system. To assume its place in the decision-maker paradigm, the state must contain all the kinds of information relevant to decision-making and system performance evaluation. We need only represent dynamic system components in the system state; static components may be regarded as fixtures of the decision mechanism. The state of the process will actually be a vector of integer, real, and perhaps other descriptive values, each value informing of some component of the system. It is the object of this chapter to represent verbally and notationally the interrelationships among the various dynamic state components as they evolve from pulse to pulse.

6.2. The MWTS Model

The purpose of the New York City Department of Sanitation's marine waste transport system is to convey refuse from various collection points throughout New York City to Fresh Kills Landfill on Staten Island, and thereby to facilitate the disposal of much of New York City's refuse. A map of New York City noting the Department of Sanitation's marine and landfill facilities appears in Exhibit 6.1. To review, here is a brief description of how the system functions: Many sanitation trucks, after completing their refuse pick-up assignments, dispose of their loads at marine transfer stations (MTS). An MTS is a two-story structure located on New York City's shoreline. A sanitation truck assigned to an MTS drives onto the second story of the MTS, backs up to one of several apertures in the platform, and dumps its load through the aperture onto a refuse barge moored in the slip directly below (the first story of the MTS). Tugboats circulate throughout New York City's waterways, dropping off empty refuse barges and picking up loaded ones. Loaded barges are assembled into barge trains at MTSs or at staging areas (SA). The tugs tow the trains to Fresh Kills Landfill (FKL), where the barges are unloaded with large cranes called "diggers." Refuse dug out of the barges is deposited in large wagons. Tractors pull the wagons out to the active "face" of the landfill, the location where the contents of the wagon are

Exhibit 6.1: Map of New York City with Department of Sanitation Facilities



dumped. Dispatchers send tugs back to the SAs and MTSs with newly dug-out barges.

While the entire journey of an item of refuse from the trash can to the landfill is not accounted for within our MWTS model, the model does cover a major segment of that journey. Our model formulation is organized by the various locales (MTS, FKL, waterway) where system activity occurs. The activity at each locale is first described verbally, then translated into algebraic relationships. Points where the model departs significantly from reality are identified.

6.2.1 Marine Transfer Stations

At each MTS, a random amount of refuse arrives every pulse. This amount depends on the MTS and the pulse index, but not on what arrived in previous pulses or at other MTSs during the same pulse. If the MTS is open (meaning that there is empty barge space on barges there), the refuse is loaded onto the barge. Otherwise, the refuse is weighed and then lost to the system. A running count is kept of total lost refuse (also referred to as "deferred refuse," as in [21]) at each MTS. Systemwide lost refuse is the primary performance measure in our model of the MWTS. (Discussion of performance measures for the actual system appears in Chapter VIII.)

Each barge has loading criteria assigned to it before it begins receiving refuse. The loading criteria are random variables in the general case. The function of these criteria is to determine when, in terms of tons of refuse loaded aboard, the barge may and must stop receiving refuse. The criteria consist of a maximum and a minimum tonnage level (i.e., the tug must stop taking on refuse once it reaches its maximum level, but may be towed away by a tug if its loading exceeds the minimum level). After taking care of incoming refuse, if the MTS is open, the tonnage level aboard the loading barge is referred to these criteria and the tug presence situation to see whether it has become full. If so, it is moved out to the queue of loaded barges awaiting tow to FKL. If any empty barges are present, one empty barge leaves the empty queue and becomes active. If not, the MTS enters the blocked state. Exhibit 6.2 explains the notation used in depicting these relationships in our model. Exhibit 6.3 provides the model's state transition equations that portray MTS operations.

Actual normal operations depart from their portrayal in the model in the following ways:

* the operations work over continuous time;
* refuse arrival quantities may exhibit dependence among MTSs and/or over time;
* lost refuse queues up at the MTS or is rerouted--it is never "lost to the system";
* most MTSs have two locations at which barges can load;
* barge transfers ("shifts") from empty queue to active and from active to full queue are not instantaneous (although there is no break in refuse transfer when another location with an empty barge exists at that MTS);
* the ability to shift barges depends on the MTS's structure, location, staffing level and whether or not a tug is present.

6.2.2 Fresh Kills Landfill

Barges are dug out at one location at FKL. When the digger is active, a random amount of refuse is dug out of the unloading barge every pulse. In any pulse, if the amount of refuse remaining on the barge drops to zero, that barge moves to the FKL empty queue. Then if any loaded barges are available, one becomes active; otherwise, FKL becomes idle. Exhibit 6.4 explains the notation used in depicting these relationships in our model. Exhibit 6.5 provides the model's state transition equations that portray FKL operations.

Actual normal operations depart from their portrayal in the model in the following ways:

* the operation works over continuous time;

Exhibit 6.2: MTS State Components

ME:	number of empty barges at MTS
MA:	amount of refuse on loading barge at MTS
MF:	number of full barges at MTS
MW:	amount of refuse on each full barge at MTS
MS:	status of MTS (1 = open, 0 = blocked)
MD:	amount of refuse deferred at MTS
MT:	number of tugs at MTS
$f_{i0}(x)$:	probability that loading barge has capacity x of refuse
$f_{i1}(x)$:	probability that loading barge loaded with up to x less than capacity may be towed away by tug
$w_i(t)$:	amount of refuse delivered to MTS i in pulse t (random variable)

Exhibit 6.3: MTS State Transition Equations

$MA_i(t) \leftarrow MA_i(t\text{-}1) + w_i(t) \cdot MS_i(t)$

$MD_i(t) \leftarrow MD_i(t\text{-}1) + w_i(t) \cdot (1 - MS_i(t))$

if $MA_i(t) \geqq ML_i(0)$ **then**

  begin

    $MF_i(t) \leftarrow MF_i(t) + 1$

    $MW_i(MF_i(t),t) \leftarrow ML_i(0)$

    if $ME_i(t) > 0$ **then**

      begin

        $ME_i(t) \leftarrow ME_i(t) - 1$

        $MA_i(t) \leftarrow MA_i(t) - ML_i(0)$

        $ML_i(0) \leftarrow x$, x drawn from $f_{i0}(x)$

        $ML_i(1) \leftarrow ML_i(0) - x$, x drawn from $f_{i1}(x)$

      end

    else

      begin

        $MA_i(t) \leftarrow 0$

        $MS_i(t) \leftarrow 0$

      end

  end

Exhibit 6.4: FKL State Components

FE:	number of empty barges at FKL
FA:	amount of refuse on unloading barge at FKL
FF:	number of full barges at FKL
FW:	amount of refuse on each full barge at FKL
FS:	status of FKL (1 = active, 0 = idle)
FT:	number of tugs at FKL
v(*t*):	maximum amount of refuse unloaded at FKL in pulse *t* (random variable)

Exhibit 6.5: FKL State Transition Equations

```
FA(t) ← max( 0, FA(t-1) – v(t)·FS(t) )
if FA(t) = 0 and FS(t) = 1 then
    begin
        FE(t) ← FE(t) + 1
        if FF(t) > 0 then
            begin
                FA(t) ← FW(FF(t),t)
                FF(t) ← FF(t) – 1
            end
        else FS(t) ← 0
    end
```

* there are usually several diggers working simultaneously at FKL, each with its own loaded barge queue;
* the number of barges that are unloaded per digger per eight-hour shift is often bounded above by work rules;
* barge transfers from full queue to active and from active to empty queue are not instantaneous;
* there may be some dependence among quantities unloaded from pulse to pulse.

6.2.3 Waterways

A tugboat that is en route to a facility (MTS, SA, or FKL) in the marine waste transport system travels a certain distance through the waterways each pulse. The travel distance depends on the tug and the composition of its barge train. Upon its arrival, the tug transfers its train to the destination's facilities. It waits a certain number of pulses at this location and then either departs or begins another waiting period at the location. A tug's arrival and/or presence at a facility may affect other activities (in particular, whether a loading barge is considered full enough to be towed away or not). The tug's new destination and the size and makeup of its barge train is decided upon its departure. It is here that dispatch decisions impinge on the modeled operation. Exhibit 6.6 explains the notation used in depicting these relationships in our model. Exhibit 6.7 provides the model's state transition equations that portray waterway (tug) operations.

Actual normal operations depart from their portrayal in the model in the following ways:

Exhibit 6.6: SA, Tug, and Other State Components

SB:	number of barges at SA
SW:	amount of refuse on each barge at SA (0 if empty)
TT:	number of barges towed by tug
TW:	amount of refuse on each barge towed by tug (0 if empty)
TZ:	status of tug (1 = on assignment, 0 = awaiting dispatch)
TD:	if positive, destination of tug (tug is en route); if negative, location of tug (tug is docked)
TP:	progress of tug (if en route, distance remaining to destination; if docked, number of pulses until next dispatch attempt)
M:	set of facility identification numbers corresponding to MTSs
S:	set of facility identification numbers corresponding to SAs
D:	dwell time at a facility, in pulses (function of facility and tug status TZ)
$u_j(t)$:	maximum distance traveled by tug j in pulse t (random variable)

Exhibit 6.7: Waterway State Transition Equations

```
if TD_j(t-1) < 0 then      "docked
   begin
      TP_j(t) <- TP_j(t-1) - 1
      if TP_j(t) = 0 then      "dispatch attempt
         begin
            i <- -TD_j(t-1)
            if i ∈ M then
               begin
                  if MA_i(t-1) ≧ ML_i(1) then
                     begin
                        MF_i(t-1) <- MF_i(t-1) + 1
                        MW_i(MF_i(t-1),t-1) <- MA_i(t-1)
                        if ME_i(t-1) > 0 then
                           begin
                              ME_i(t-1) <- ME_i(t-1) - 1
                              MA_i(t-1) <- 0
                              ML_i(0) <- x, x drawn from f_i0(x)
                              ML_i(1) <- ML_i(0) - x, x drawn from f_i1(x)
                           end
                        else
                           begin
                              MA_i(t-1) <- 0
                              MS_i(t-1) <- 0
                           end
                     end
                  DISPATCH (TD_j(t-1), MF_i(t-1), MW_i(·, t-1), TZ_j(t-1),
                              TD_j(t), TP_j(t), TT_j(t), TW_j(·, t), TZ_j(t) )
                  if TD_j(t-1) > 0 then MT_i(t) <- MT_i(t-1) - 1
               end
            else if i ∈ S then
               DISPATCH (TD_j(t-1), SB_i(t-1), SW_i(·, t-1), TZ_j(t-1),
                           TD_j(t), TP_j(t), TT_j(t), TW_j(·, t), TZ_j(t) )
(continued)
```

Exhibit 6.7: Waterway State Transition Equations (continued)

```
            else
               begin
                  DISPATCH (TDj(t-1), FE(t-1), - - , TZj(t-1),
                                   TDj(t), TPj(t), TTj(t), TWj(·, t), TZj(t) )
                  if TDj(t)>0 then FT(t) <- FT(t-1) - 1
               end
         end
   end
else      "en route
   begin
      TPj(t) <- max ( 0, TPj(t-1) - uj(t) )
      if TPj(t) = 0 then      "arrival
         begin
            i <- TDj(t-1)
            TPj(t) <- Di( 0 )
            TDj(t) <- -TDj(t-1)
            if i ∈ M then
               begin
                  MTi(t-1) <- MTi(t-1) + 1
                  for n <- 1 to TTj(t-1) do      "unload barges
                     if TWj(n,t-1) = 0 then
                        MEi(t-1) <- MEi(t-1) + 1
                     else
                        begin
                           MFi(t-1) <- MFi(t-1) + 1
                           MWi( MFi(t-1), t-1) <- TWj(n, t-1)
                        end
                  TTj(t) <- 0
                  if MEi(t-1) > 0 and MSi(t-1) = 0 then
                     begin
                        MSi(t-1) <- 1
                        MEi(t-1) <- MEi(t-1) - 1
                     end
               end
```

(continued)

Exhibit 6.7: Waterway State Transition Equations (continued)

```
            else if i ∈ S then
                begin
                    for n ← 1 to TT_j(t−1) do
                        begin
                            SB_i(t−1) ← SB_i(t−1) + 1
                            SW_i( SB_i(t−1), t−1) ← TW_j(n,t−1)
                        end
                    TT_j(t) ← 0
                end
            else
                begin
                    FT(t−1) ← FT(t−1) + 1
                    for n ← 1 to TT_j(t−1) do
                        begin
                            FF(t−1) ← FF(t−1) + 1
                            FW( FF(t−1), t−1) ← TW_j(n,t−1)
                        end
                    TT_j(t) ← 0
                    if FF(t−1) > 0 and FS(t−1) = 0 then
                        begin
                            FS(t−1) ← 1
                            FF(t−1) ← FF(t−1) − 1
                        end
                end
        end
    end
```

* the operation works over continuous time;
* there may be statistical dependence among the travel distance random variables;
* barge transfers between tugs and stationary facilities are not instantaneous;
* the dwell times vary stochastically.

6.2.4 Additional Notes

Unless indicated otherwise, each quantity Y has the property that

$$Y(t) = Y(t\text{-}1)$$

The only significant sequence rule in this model is that, in every pulse, waterway activity occurs prior to MTS and FKL activity. No dependence relationships exist among any of the random variables. Random variables of the same type and facility need not be identically distributed over pulses. The one model component yet to be described, the dispatch procedure, is treated in the next section.

6.3. Dispatching

In the marine waste transport system, to *dispatch* a tug means to decide:

* when the tug is to depart from the facility it resides at;
* what its new destination will be; and
* which barges at the current facility it is to tow (if any),

and to execute this decision. In practice, a tug can be "dispatched" at any time, even when it is en route to some facility in the MWTS, because the tug can be rerouted via radio command to some other destination. In our model, we constrain dispatches to occur only when the tug in question resides at some fixed location in the system. This model feature is not as restrictive as it may seem, because we may locate artificial facilities (adding them to the set of staging areas) within the waterway network so as to allow some tug rerouting. This is accomplished as follows: add artificial staging areas with no barge holding capabilities and zero dwell times to the network such that every path between real facilities passes through at least one artificial staging area. The ideal positioning of the artificial staging areas is where a rerouted tug would normally break off from its former path.

A dispatch is normally executed immediately after it has been decided. In some cases, particularly when a tug awaits dispatch at FKL, it may be advantageous to hold the tug at the facility for a period of time before executing the dispatch. Examples where a dispatch delay is sometimes called for include waiting for barges to be emptied at FKL and waiting until barges become fully loaded at MTSs. The dwell time function allows us to incorporate dispatch delays into our model by making dwell times at facilities on the second and subsequent dispatch attempts short enough to allow frequent monitoring of system activity.

From a macroscopic perspective, our MWTS model's dispatch mechanism may be construed thusly: Initially, a decision is made about where to send the tug that is awaiting dispatch. This decision may alternately specify that the dispatch be delayed. If the dispatch is to be made immediately, then the composition of the barge train is decided, and the tug departs for its first destination. Otherwise, a new dispatch attempt is scheduled some number of pulses into the future. The algorithm given in Exhibit 6.8 portrays the dispatch procedure from this perspective. (Note: the line

$$TP(t) \leftarrow DISTANCE\ (TD(t-1), TD(t))$$

in Exhibit 6.8 causes the progress of the tug to be set to the travel distance from origin to destination.)

We recognize three possible dispatching modes in the MWTS model. The first mode may be termed "all points dispatching." In this mode, a dispatch procedure to determine the next destination is executed whenever a tug is ready to leave any facility, real or artificial, in the system. In the second or "fixed route" mode, the sequence of system points that a tug visits and the numbers of empty barges delivered to each MTS visited between stays at FKL is fixed upon departure from FKL. The third mode, called "fixed MTS" mode, is similar to fixed route dispatching in that the set of MTSs to be visited is fixed in advance. but the sequence of the visits is a matter left undecided until the tug has to depart for one of the MTSs in its assignment that it

Exhibit 6.8: Structure of the Dispatch Procedure

```
procedure DISPATCH (TD(t−1), BA(t−1), BW(·,t−1), TZ(t−1),
                    TD(t), TP(t), TT(t), TW(·,t), TZ(t) )
DESTINATION ( TD(t−1), TD(t), TZ(t) )
if TZ(t) = 1 then
   begin
      TRAIN ( BA(t−1), BW(·,t−1), TT(t), TW(·, t) )
      TP(t) ← DISTANCE ( TD(t−1), TD(t) )
   end
else TP(t) ← D_−TD(t)(1)
return
```

has not yet visited. With the last two modes, compositions of barge trains may remain undecided until departure from each facility, but changing the set of system points to visit is prohibited except possibly when, due to stochastic effects, certain visits in the sequence become entirely unproductive. Clearly, the first dispatching mode will, if implemented properly, produce more effective decisions, but the dispatch mechanism used here must be more sophisticated. The second and third modes are considered because they may be more practical to implement. Our algorithms for dispatching in the MWTS model employ the "fixed MTS" dispatching mode. Chapter VII provides more detail about procedures DESTINATION and TRAIN, the procedures making up the heart of any dispatching algorithm.

CHAPTER VII

DISPATCHING IN THE MWTS MODEL

7.0 Introduction

To complete the description of our model of the marine waste transport system, we must specify how dispatching is accomplished, or how the dispatcher decides to move tugs and barges throughout the system. We have isolated two procedures the dispatcher must perform for each decision: a DESTINATION procedure which determines the system facility a tug is to next travel to (or signals that the disptach be delayed), and a TRAIN procedure which selects the locally-held barges to take along. In this chapter, we adapt the MWTS model structure, insofar as the dispatching algorithm is aware of it, so that the methods of Chapter IV may be applied for MWTS dispatching. We also assemble another algorithm to serve as a basis for performance comparisons.

The task of dispatching in the MWTS environment conforms closely to the deliverer dispatch problem paradigm. The dispatcher tries to deploy scarce delivery resources to avoid undesirable customer inventory events (demands while stocked out, holding excessive inventory). The delivery resources are the tugs and the delivered good is the empty space on barges. Tugs operate around the clock, and often start out on a new assignment shortly after finishing the old one. Barge usage (demand) is uncertain to some extent. But the fit between the MWTS model and the DDP that the Future Value Decomposition algorithm was designed for is not close enough to allow the straightforward application of the methods being developed in this dissertation. We therefore need to study the discrepancies between the MWTS model and the DDP. This chapter proceeds by alternatingly identifying a key discrepancy and describing how the resulting difficulties are overcome in our case study. The final assessment of the FVD algorithm in MWTS dispatching is performed in Chapter VIII.

7.1 Time Scales

The first notable discrepancy between the MWTS model and the DDP that the decomposition algorithm was developed for is one of time scale. While both models operate in discrete time with constant intervals between event or decision points, the most practical length of a pulse to use in the MWTS model is a great deal shorter than the most practical length of a period in the DDP. In the MWTS model, decisions and events only occur at the start of each pulse. From this it can be reasoned that, for instance, all trip times from facility to facility in the MWTS model must take an integral number of pulses. Since some trips may take half an hour or less, choosing a pulse length of more than 15 minutes might distort the representation of travel activity in the model. Now suppose a *period* were 15 minutes long. Then in the DDP representation of the system, the probability of demanding one unit in a period, when the unit equals the empty space of one barge, would be very low at any marine transfer station. Since the FVD algorithm is based in part on the assumption of demand being independent from period to period, we would need to model barge demand as having a Bernoulli distribution with a very low probability of "success" (success = barge demand). The time between barge demands would as a result have far greater variance than occurs in reality, another serious distortion of MWTS activity.

The resolution to the period-pulse discrepancy is to allow a period to span multiple pulses. In our later computational work, we fix a pulse at 15 minutes of real time, and use a period length of 15 pulses (3.75 hours). This resolution will require the FVD algorithm to reinterpret some elements of system behavior. Characteristics of refuse arrival, expressed in tons per pulse in the MWTS model, are translated into barges per period. Itinerary leadtimes and duration attributes, which also must be related in terms of periods, are inferred from travel distances and tug travel rates. When these time intervals work out to non-integral numbers of periods, the values are generally rounded upwards; it is safer to assume a tug arrives at a destination later than it does, than to assume it arrives earlier. More information about pulse-period transformations is provided below.

7.2 Barge Demand over an Extended Scheduling Horizon

The demand for the delivered good is continuous rather than discrete. The quantity of the good delivered to an MTS is generally expressed as an integral number of barges, but the rate at which the space on the barges is used is more closely reflected by a statement such as "X number of tons per pulse." In other words, we have fractional demand for barges per pulse. An added complication pertains to the assumption of demand independence between periods. Exhibit 7.1 indicates the nature of the problem. Suppose, at a certain MTS, refuse arrives at a constant rate so that each barge takes exactly eight pulses to fill, while a period lasts six pulses (these figures are for illustrative purposes only). Even though the variance of refuse inflow per pulse is zero, the barge demand per period (demand being equated to wanting to start loading another barge) is not constant, a phenomenon known as the "integer round-off effect". Furthermore, knowledge of when the last barge was filled in the preceding period enables the more accurate prediction of barge demand in the upcoming period, so that barge demands are not truly independent.

To deal with barge demand dependence, we switch to an extended scheduling horizon version of the FVD algorithm (refer back to Section 4.3.7.4 for an introductory discussion). Let the scheduling horizon be T periods long. To solve the DDSP for customer i, we need the quantities

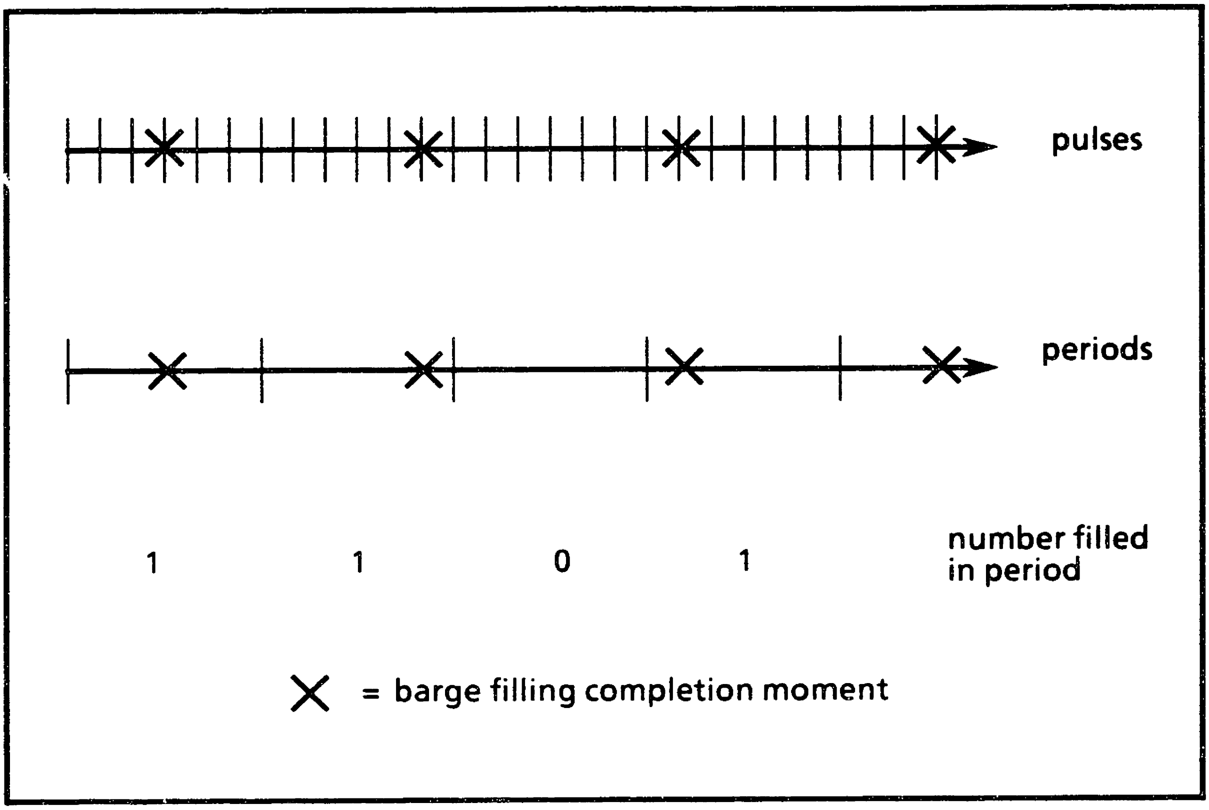
$$p_i(x_{i0}, x_{i1}, \ldots, x_{i,T-1}),$$

the probabilities that the barge demand in period t is x_{it}, $t=0,1,\ldots,T-1$, for each realizable demand T-tuple $(x_{i0},x_{i1},\ldots,x_{i,T-1})$. If interperiod demands are independent and identically distributed according to $p_i(x_t)$, then the above probability equals

$$\prod_{t=0}^{T-1} p_i(x_{it}),$$

and there are $(X+1)^T$ possible T-tuples, where X is the maximum demand per period. But if the

Exhibit 7.1: Effects of Time-Discretization on Barge Demand



demands are not independent, then the number of demand T-tuples that occur with positive probability may be somewhat less than $(X+1)^T$, and we may have direct methods of assessing their probabilities. For instance, in Exhibit 7.1, when the scheduling horizon is two periods, the demand 2-tuple (0,0) will never occur, even though it is possible for no barges to become full in a single period.

We now outline a method for establishing demand probabilities over a horizon; an example follows. We return to the original MWTS model to obtain, for a given MTS, the distribution of the quantity of refuse arriving per pulse. This information is used to calculate the mean and standard deviation of refuse arriving per scheduling horizon. Invoking the Central Limit Theorem, total refuse per scheduling horizon is taken to be distributed normally (but truncated below zero so that no negative inflows are possible). We assume that the amount of refuse residing in the barge loading at the start of the horizon is evenly distributed between zero and the full capacity of a barge, a random incidence assumption. (In practice, we would know the time this barge began loading, and would therefore be able to restrict the range of initial barge loading; this information has no role in the solution of the deliverer dispatch subproblems, though.) The sum of this holdover amount and the total arrival in the scheduling horizon indicates how many barges are demanded (filled or lost) in the scheduling horizon. We numerically determine the probability p_N, for each N, that N barges are demanded in the scheduling horizon. For each N, we find the a_N distinct allotments of the N barges to the T periods of the scheduling horizon such that no more than one barge is demanded per period (which we can certify with a high degree of confidence if we make the period length small enough) and assume each allotment is equally likely to occur. The probability of any demand T-tuple of total demand N occurring is then

$$\frac{p_N}{a_N}.$$

When these probability assessments are incorporated into the solution of the DDSPs, we must still assume these demand pattern probabilities are independent between scheduling horizons. This will not be the case in the MWTS model, although the degree of correlation of patterns between extended scheduling horizons may be less than that between periods.

An example may help to clarify this procedure. Using Exhibit 7.1 as a basis, we consider an MTS which receives exactly one-eighth of a barge worth of refuse each pulse, a period length of six pulses, and a scheduling horizon length of two periods (twelve pulses). Exactly one and one-half bargeloads of refuse arrives during each scheduling horizon (this can be considered normally distributed with zero variance). The number of barges actually filled in the current scheduling horizon depends on the time the currently loading barge began loading, meaning how much refuse is in it at the start of the horizon (we assume the MTS is not stocked out). Employing the principle of random incidence, we consider the number of eighth-barges of refuse already on board to be evenly distributed from zero to seven. If the number of eighths on board is not more than three, then only the currently loading barge will be completed during the scheduling horizon. Two will be finished if four eighths or more have already been loaded. By our assumption, it is equally likely that the initial barge loading be three-eighths or less, or four or more. Hence, for this example, the horizon demand probabilities p_1 and p_2 both equal 1/2.

To complete the estimation of the probability of occurrence of each demand 2-tuple, we must generate the barge allotments to periods for $N=1$ and $N=2$ barges demanded. If $N=1$, we consider the two allotments (1,0) and (0,1) to be equally likely. For $N=2$, only one allotment exists with a maximum of one barge demanded per period, (1,1). The total number of possible 2-tuples is found here to be 3, less than the the $(X+1)^T=4$ combinations of 0 or 1 barges demanded per period. Exhibit 7.2 summarizes the horizon demand probability assessment process and presents final results. Exhibit 7.3 shows each of the eight possible barge completion patterns over a scheduling horizon, one for each of the eight possible pulses when the currently loading barge began loading. Assuming each pattern is equally likely, then the demand 2-tuple probabilities given in Exhibit 7.2 are corroborated by their frequencies of occurrence as indicated in Exhibit 7.3.

We observe that, for any particular scheduling horizon, *if* we have some knowledge about the time of start-up of the currently loading barge, then our forecast of horizon demand is liable to change. For instance, if we know that the current barge began loading *at least* two pulses ago, the first two patterns of Exhibit 7.3 must be eliminated, and the assessment of 2-tuple probabilities (assuming all remaining patterns equally likely) would become

(1,0) with probability 1/3,

(1,1) with probability 2/3.

Incorporating this type of information within the FVD algorithm would be infeasible to do, though, because new DDSPs would have to be solved for each potential information profile. Some consideration might be lent to this information when solving the master scheduling problem,

Exhibit 7.2: Horizon Demand Probability Assessment Process

N	p_N	a_N	Allotment	Probability of demand 2-tuple
1	0.5	2	(1,0)	0.25
1	0.5	2	(0,1)	0.25
2	0.5	1	(1,1)	0.50

perhaps for selecting the scheduling horizon dispatch decision among the ones with the lowest objective value.

In the DDP we have developed the FVD algorithm for, it is assumed that the demand distributions hold steady over the infinite planning horizon. Our computational experiments treat only the situation in which, for each MTS, refuse inflow is independent and identically distributed across all pulses. The random amount of refuse x that arrives in a pulse at an MTS is taken to have the gamma distribution

$$\frac{1}{\Gamma(\alpha)}\beta^{\alpha}x^{\alpha-1}e^{-\beta x},\ x>0,$$

for given parameters α and β. The mean of the gamma distribution is α/β, and the standard deviation is $(\sqrt{\alpha})/\beta$. In each scenario we simulate in Chapter VIII, each MTS has its own mean rate of refuse inflow per pulse, while the scenario has a daily coefficient of variation (0.1 or 0.2) assumed the same for each MTS. This information provides us with all we need to calculate α and β for a given MTS. Let P be the number of pulses in a day. If refuse flows in at the rate of μ per pulse, and the daily coefficient of variation is c, then:

$$\frac{\alpha}{\beta}=\mu \ \ or \ \ \alpha=\mu\beta;$$

$$standard\ deviation\ of\ inflow\ per\ day = c\mu = \frac{\sqrt{\alpha}}{\beta}\sqrt{P} \ \ or \ \ \alpha=\frac{(c\mu\beta)^2}{P};$$

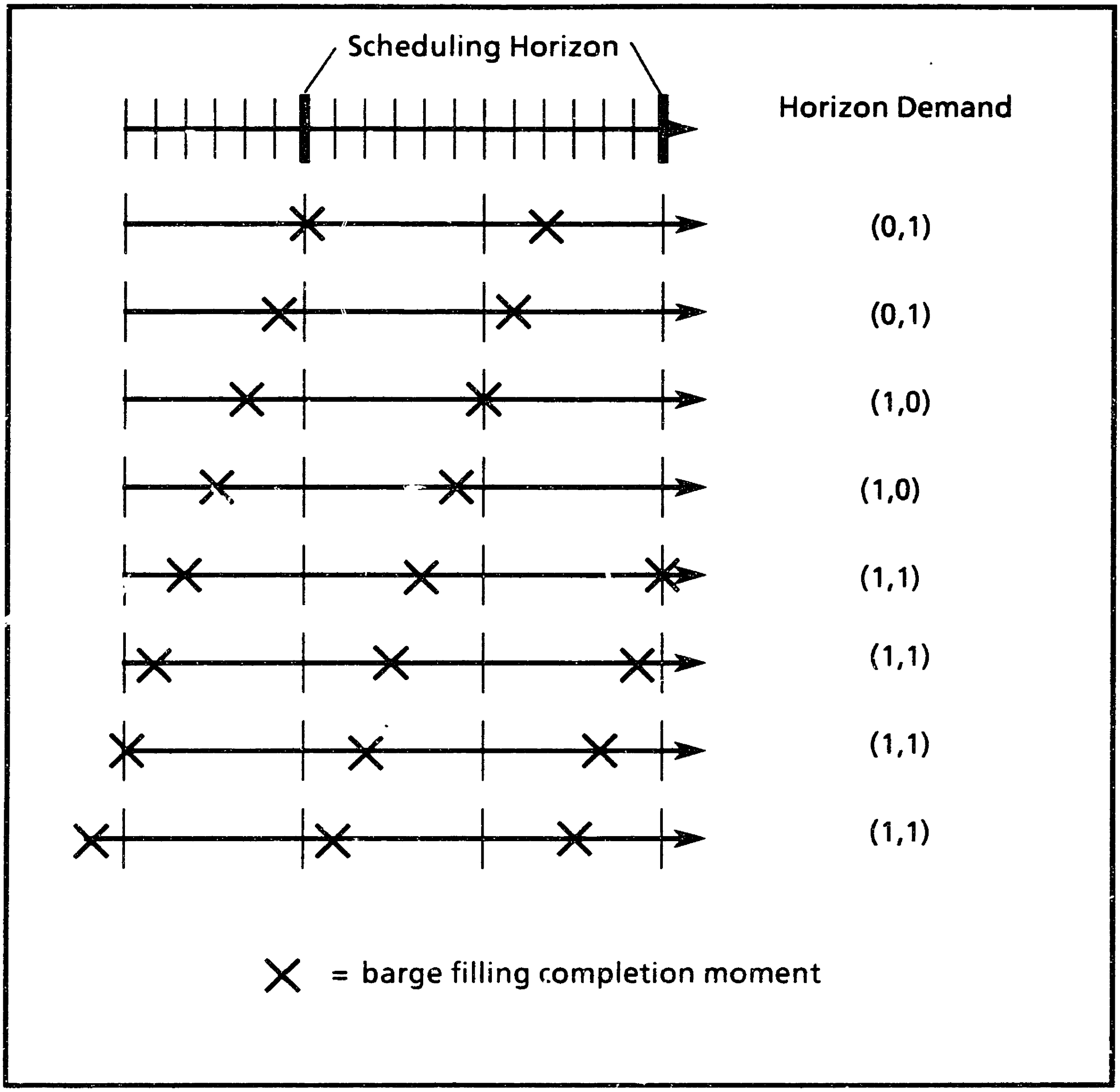
therefore,

$$\beta = \frac{P}{c^2\mu} \ and \ \alpha = \frac{P}{c^2}.$$

If, for example, $c=0.1$ and $P=100$ ($P=96$ when a pulse is 15 minutes long), the standard deviation of refuse inflow per *pulse* would equal μ. An assumption of normality about refuse inflow per pulse would cause a sizable proportion of refuse inflows to be negative, if no truncation of the distribution were used. This is why we have opted for the gamma distribution. Unfortunately, the data do not currently exist to verify or reject the supposition of gamma-distributed refuse inflows per pulse.

The way in which refuse arrival is modeled in the previous chapter allows for the arrival distribution to vary over time. Data analyses (see [15]) have shown that refuse arrival means vary by time of year, day of week, and time of day. The only means visible to us for incorporating demand distribution variability into the FVD algorithm in a natural way is by augmentation of the state spaces of the DDSPs. Assuming cyclic demand, as it is defined in Section 3.2.3, the state space for each DDSP may be augmented with an information element that indicates which phase the demand cycle will be in in the upcoming period. If we assume the phases to cycle in a given sequence, then the resultant DDSP, while larger, can be solved in the same way as before. We use the output from the DDSPs to construct cost coefficients $C_{jt\theta}{}^{+}$ for the extended master scheduling problem (see Section 7.3), since now future expected values depend on which phases come soonest in the future. The maximum number of phases that can be accommodated in a cycle is limited by the computational burden one is willing to endure in the execution of the FVD algorithm, and by the data storage one has at hand. In application to the MWTS case, there will be the additional difficulty that dispatches sometimes occur in the middle of a phase. A round-off technique should be applied to position the dispatch point squarely at the beginning of one phase or another, so that the cost coefficients of the MSP may be determined in the usual manner. We have no computational experience for the method just described. We also remark that we know of no feasible way to formally handle the case where demand varies acyclically.

Exhibit 7.3: Barge Completion Patterns



7.3 Other Adaptations for an Extended Scheduling Horizon

When an extended scheduling horizon is used in the FVD algorithm, several modifications to the algorithm's components must be introduced. In each deliverer dispatch subproblem, the optimal replenishment policy and expected future values must make reference to the period in which the order is dispatched, not just the quantity of the order. The decision variables of the master scheduling problem must specify the period of the dispatch as well as the itinerary, and vehicle availability constraints must be broken down by period (with undispatched vehicles in one period made available in the next). We stipulate that in both the DDSPs and the MSP we restrict each customer to be included in at most one dispatched itinerary per scheduling horizon, in order to avoid serious complications of the formulation and/or solution of the problems. Remember, though, that only the dispatches of the first decision point in the horizon are executed immediately; the rest may be changed as new information becomes available at the next decision point and the extended problem is solved once again.

Another important difference between the MWTS model and the DDP assumptions is that the ability to dispatch a tug from Fresh Kills Landfill does not depend alone on the tug being ready, but also on whether there are enough barges available at FKL to send with the tug. A finite number of barges circulate throughout the system. Empty barge availability at FKL is directly related to previous full barge deliveries to FKL and to FKL's unloading capabilities. In order to utilize the extended scheduling horizon enhancement, one must obtain forecasts of barge availability in future periods of the scheduling horizon. Generally, the degree of variability of barge unloading times will be small, so that a deterministic approximation of barge unloading activity at FKL will suffice for projecting the number of barges newly coming available at each decision point. It is important, for algorithmic purposes, not to overlook that barges delivered to FKL early in the scheduling horizon may be unloaded before the last decision point in the horizon, so that barge availability forecasts should not depend solely on how many full barges are present at FKL at the start of the horizon. Also, barges available at future decision points must include barges available but not assigned to a tug earlier in the horizon.

Let us rephrase the master scheduling problem (4.39)-(4.42) for dispatching in the MWTS model under an extended scheduling horizon. The new decision variable y_{jt} has value 1 if and only if itinerary j is to be dispatched at decision point t of the scheduling horizon. The dispatch cost C_{jt}^{-} in the objective function (7.1) becomes dependent on the decision point of dispatch as well. The number of tugs ready for dispatch at decision point t is denoted A_t, and the number of barges ready then is written B_t. We introduce the number of delayed tugs variable r_t to allow tugs not dispatched at one decision point to be available at the next (constraints (7.2)), and the number of barges held over variable s_t to do the same for barges (constraints (7.3)). Let d_j in constraints (7.3) be the total number of barges delivered in itinerary j. Constraints (7.4) state that no MTS may have more than one delivery outstanding at the end of a period. Finally, in this formulation we restrict each vehicle to be dispatched at most once in the scheduling horizon (constraints (7.5)). Then the extended master scheduling problem is:

$$min \sum_{j=1}^{n} \sum_{t=0}^{T-1} C_{jt}^{+} y_{jt} \tag{7.1}$$

$$s.t. \sum_{j=1}^{n} y_{j0} + r_0 = A_0 \tag{7.2a}$$

$$\sum_{j=1}^{n} y_{jt} - r_{t-1} + r_t = A_t \ , \ t=1,\ldots,T-1 \tag{7.2b}$$

$$s.t. \sum_{j=1}^{n} d_j y_{j0} + s_0 = B_0 \tag{7.3a}$$

$$\sum_{j=1}^{n} d_j y_{jt} - s_{t-1} + s_t = B_t \ , \ t=1,\ldots,T-1 \tag{7.3b}$$

$$\sum_{j=1}^{n} d_{ij}^{1} y_{jt} \leq 1 - \sum_{j=1}^{n} \sum_{l=t-l_{ij}}^{t-1} d_{ij}^{1} y_{jl} \quad \forall i, t=0,\ldots,T-1 \tag{7.4}$$

$$\sum_{j=1}^{n} \sum_{t=0}^{T-1} d_{ij}^{1} y_{jt} \leq 1 \quad \forall i \tag{7.5}$$

$$y_{jt} = 0 \ or \ 1 . \tag{7.6}$$

The heuristic algorithm given in Section 4.3.4 for solving the MSP can also be used to solve the EMSP, if it is modified in the following ways:

Procedure BEST_ITIN, step 5: Substitute C_{j0}^+ for C_j^+.

Procedure GREEDY_ASSIGN, step 1: Understand A to equal

$$\sum_{t=0}^{T-1} A_t.$$

Step 2: Substitute $C_{j't}^+$ for $C_{j'}^+$, where t is the period of dispatch of k'. Also consider only j' with total delivery less than or equal to the number of barges available.

Step 3: Only return if $t=T-1$; otherwise, move all unassigned tugs and barges to $t+1$, and start again from step 2.

Step 4: Substitute $C_{j^*t}^+$ for $C_{j^*}^+$.

Step 5: Remove from I_T all customers who had outstanding deliveries at $t=0$ that will have arrived by the end of period t.

Main Procedure: Substitute A_0 for A.

Because the MSP heuristic considers all itineraries for the imminent dispatch, the tendency to go for the "biggest bang for the tug" is diminished. However, we may need to try as seeds those itineraries regarded as "dominated," since domination in the sense that it is defined in Section 4.3.4 pays no regard to constraints on the total delivery of vehicles dispatched at a decision point.

One could allow vehicles dispatched during the scheduling horizon that return before the end of the horizon to be dispatched again. In addition, one could implement procedures at the end of GREEDY_ASSIGN to seek improvement in the final vehicle-itinerary assignments. Two types of improvement procedures are dispatch delays and assignment exchanges. A dispatch delay improvement procedure tests whether delaying the dispatch by one period of a vehicle on its assigned itinerary decreases the value of the objective function. If so (and if no other dispatches are affected, as could happen if the vehicle is scheduled for a later dispatch in the scheduling horizon), the horizon schedule is modified to dispatch the vehicle one period later, and the objective value OBJECT_COST of this assignment is reduced accordingly. If any exchange of vehicle assignments proves beneficial, an assignment exchange is carried out and the new, improved value of OBJECT_COST calculated.

7.4 The TRAIN Procedure

MTS barge holding capacities were not modeled in Chapter VI; they are better represented in the TRAIN procedure in the general dispatching procedure presented there for deciding which barges to take with a tug on the trip to its next destination. This section discusses the TRAIN procedure.

Tugs leaving FKL simply take the number of empty barges equal to the total delivery on its assigned itinerary. The dispatching procedure, in assigning an itinerary, must verify that sufficient empty barges are currently available for the assignment. There are no barge capacity restrictions at FKL. Similarly, tugs leaving an MTS, whether for a staging area or for FKL, take away as many full barges as there are at the MTS up to the tug's towing limit of four barges. Tugs leaving staging areas for FKL do the same thing. There is no point to leaving full barges anywhere other than FKL if the tug has room to move them along to FKL.

The determination of the composition of the barge train for tugs departing staging areas for MTSs is more complex. The complicating factor is the general inability of tugs to tow full and empty barges in the same train. This restriction requires the TRAIN procedure for a tug heading from a staging area to an MTS to be a little more sophisticated than "tow the number of empty barges corresponding to the delivery to the MTS in the currently executing itinerary for that tug," if it is to prevent an MTS from storing more barges than its holding capacity allows. The TRAIN procedure should estimate its future time of departure from the MTS it is preparing to visit, and forecast the numbers of empty and full barges that will be there at that time. The anticipated delivery from the itinerary assigned to the tug may have to be adjusted to respect the train composition and holding capacity constraints. Since refuse inflow and tug travel times are stochastic, the forecasts are susceptible to some error. When the tug is scheduled to depart from the MTS, if the MTS still has more empty barges than it can store, the tug should wait until it can remove enough full barges to put the MTS's barge storage level at its given capacity; i.e., it should wait for one or more barges to fill up. These considerations will have a further impact on the choice of MTS to visit next from an SA. (Note: We do not employ this sort of forecasting technique in our procedure for dispatching from SAs to MTSs in the dispatching algorithms tested in Chapter VIII. In the implementation of the FVD algorithm there, MTS holding capacities are represented only within the MSP. Some potential itineraries are by-passed if their dispatch at a given time is judged likely to induce an MTS capacity violation in the future.)

7.5 The Itinerary Set

Tugs generally visit one or two MTSs before returning to FKL. Since barges cannot tow empty and full barges in the same train, routes for itineraries including more than one MTS visit must be of the "hub-and-spoke" variety. The hub is typically a staging area (SA), where the tug travels to once it has left FKL with empty barges in tow. The spokes of the route are the trips from SA to MTS with empty barges, and from MTS back to SA with full ones. The itinerary set we use is given in Appendix C. It is based on the MWTS facility configuration of Exhibit 6.1, and includes most of the conceivably useful one- and two-MTS itineraries. A potential itinerary was regarded as useful if the MTSs to visit were all relatively close to the hub of the route (termed the "focal facility"), if the total delivery was at least two barges, and if no MTS delivery quantity exceeded that MTS's storage capacity. Rather than further distinguishing itineraries by leadtimes, all MTSs were assigned a common leadtime of about one-half the duration of the itinerary. Itinerary durations can be estimated quite accurately in advance, because in our computational study, we assume that travel distances per tug per pulse are deterministic. Since the number of full barges towed on any execution of an itinerary is uncertain, so are actual itinerary durations, to some degree. The transportation cost of an itinerary is the product of its duration in pulses and the transporation cost rate per pulse (see Section 7.7).

7.6 The DESTINATION Procedure at MTSs and SAs

When a tug prepares to leave FKL, the modified FVD algorithm is executed to assign the tug an itinerary, and the tug heads for the focal facility of that itinerary. Tugs leaving an MTS head for their focal facility, or FKL if the focal facility is the MTS it is leaving. But when a tug leaves an SA for an MTS, the question of which MTS to visit may not be so clearcut, if more than one MTS remains to be visited. The sequencing of the visits in multiple-MTS itineraries is handled by the DESTINATION procedure for tugs leaving SAs. This is in accordance with our selection of the "fixed MTS" mode of dispatching. The DESTINATION procedure tested sent the tug to the MTS left on the itinerary that was estimated to be stocking out the soonest. If more than one were already stocked out of empty barges, the MTS with the highest mean rate of refuse arrival became the destination.

A more sophisticated procedure for choosing which MTS to next visit would consider each possible sequence of the remaining MTSs. It would determine the expected tug arrival and departure times for each facility under each sequence. Information about these times would serve two purposes. First, it would enable the forecast of barge configurations at MTSs at the time of MTS departure, as was discussed in conjunction with the TRAIN procedure. Second, it would support a more accurate estimation of the total expected lost tonnage among the MTSs under consideration in each sequence. Lost tonnage is the measure of sequence quality that we would most like to minimize. The procedure would choose the sequence minimizing expected lost tonnage. To the extent that MTS visit order decisions are in the hands of the dispatcher, a sophisticated CAD system should incorporate a DESTINATION procedure of this type.

7.7 Performance Measures

The chief objective in the MWTS model is to minimize lost refuse throughout the system. Drawing a parallel to the DDP, this would seem to imply that all transportation costs C_j and inventory holding costs H_i should be set to zero. Indeed, since the barges are owned by New York City no matter where the barges are located, and the tug services rented at a cost independent of the proportion of the time they are in use (except for the occasional hiring of an extra tug on a temporary basis), perhaps zero *is* the relevant cost for these resources. However, setting the costs to zero for solving a DDP based on this operation may be hazardous, due to the finite availability of barges and tugs. For instance, if barge holding costs were zero, DDP-generated dispatches would keep all MTSs nearly fully stocked at all times, if possible. This tendency may put a greater strain on barge and tug resources than might otherwise be applied, leaving the system more vulnerable (without buffer) to heavier than normal demand.

We test two alternatives to establishing transportation and holding costs, in order to learn about the sensitivity of the FVD algorithm to their values. One examines the consequences of attaching nominal costs to these activities. In the other method, the capitalized cost of a barge is used as a basis for calculating holding costs, and the hourly contracted rate at which a tug is rented for transportation costs. Let us take a look at some cost information for the MWTS in the year 1982. The following data are drawn from work done by the author on the project described in [21], and are rough rather than precise figures:

Barges: Capital and maintenance costs: $150,000 per barge per year.

Tugs: Rented at the rate of $200 per tug per hour.

Lost refuse: Costed at about $20 per ton lost.

If a pulse lasts a quarter of an hour and there are 15 pulses in a period, then the figures above imply the following activity and lost demand costs for use by the FVD algorithm:

Inventory holding: To hold one unit (barge) in inventory costs $65 per unit (barge) per period.

Itineraries: Tugs cost $50 per pulse--multiply by the duration of an itinerary in pulses to find the cost of the itinerary.

Lost Demand: Each barge holds around 600 tons of refuse. At $20 per ton, a barge-full of lost refuse would cost $12,000.

In every simulation run in Chapter VIII, the cost of a lost demand L, assumed equal among all MTSs, will be set at $12,000. We will try some runs with the holding and transportation per pulse costs at $40 and $50, respectively, and others at $0 and $1. (The transportation cost in the latter approach is not set to $0 due to computer programming considerations; $1 is still an almost trivial cost, when compared with the cost of a lost demand.)

The optimal activity costs to feed into the FVD algorithm for purposes of minimizing the rate of lost refuse may not be either of the cost combinations mentioned thus far. The resource base is considered fixed while the algorithm operates--i.e., the algorithm cannot issue the decision "rent another tug" if it finds none available when it wants to dispatch one. This suggests that the *shadow price* of a resource might prove the best way to cost out its usage. The shadow price of a unit of resource is essentially the maximum price one would pay to acquire an additional unit of the resource. Although shadow prices are a natural by-product of such constrained optimization procedures as the simplex method for linear programming, the relationships among constraints, objectives, and decision variables are much more opaque in this problem. We feel that the assessment of shadow prices in models such as this one is an excellent topic for future research.

7.8 A Competing Dispatch Algorithm

The chief competition for the FVD algorithm adapted to the MWTS model is a procedure based on the actual methods used to route tugs and barges. Here is a brief recap of what goes on in dispatching in the real-world operation:

Every morning, officials in Operations at the Bureau of Waste Disposal estimate how many barges each MTS will require to handle the *following* day's refuse inflow. On the basis of these forecasts, the officials determine how many empty barges, including fractions thereof, must be present at each MTS by the start of the next day. Taking into account the amount of empty barge space that may be left over from the current day's activity, barge delivery requirements for over the course of the current day are established. These requirements are radioed in the morning to the dispatchers (who are situated at FKL). The dispatchers then plan the day's tug schedule, earmarking empty and soon-to-be-emptied barges for the different MTSs. Dispatchers may intercede at various times during the day and alter the schedule to respond to updated forecasts. Barge requirements forecasts and tug schedules are not based on explicit formulas and solutions to routing problems, but on human judgment, so it is difficult to translate verbatim the actual procedure into a computer algorithm.

The algorithm we offer below is intended to carry the flavor of current dispatching methods. It is named the Actual Dispatching Emulator (ADE) algorithm. Exhibits 7.4 and 7.5 give flowcharts of the algorithm. The ADE algorithm consists of two phases. In the first phase (Exhibit 7.4), all the itineraries that will be dispatched during the next 24 hours are identified. Itineraries are selected to meet targets for barges-on-hand at each MTS one day hence. These itineraries are sequenced in the second phase of the algorithm (Exhibit 7.5) with the objective of minimizing expected lost tonnage during the day.

The ADE algorithm is executed once each day, say, at midnight. The first step in the algorithm is to determine a barge capacity holding target t_i for each MTS i. This is the number of tons of barge capacity we want to be at each MTS or en route there 24 hours hence (midnight tomorrow). Let P be the number of pulses in a day, and μ_i the mean rate of refuse arriving at MTS i per pulse. $P\mu_i$ is the expected tonnage inflow to MTS i for one day. Also let ω_i be the mean capacity of a barge at MTS i (in tons), and u_i the maximum number of barges that MTS i may

Exhibit 7.4: Itinerary Selection Phase of ADE Algorithm

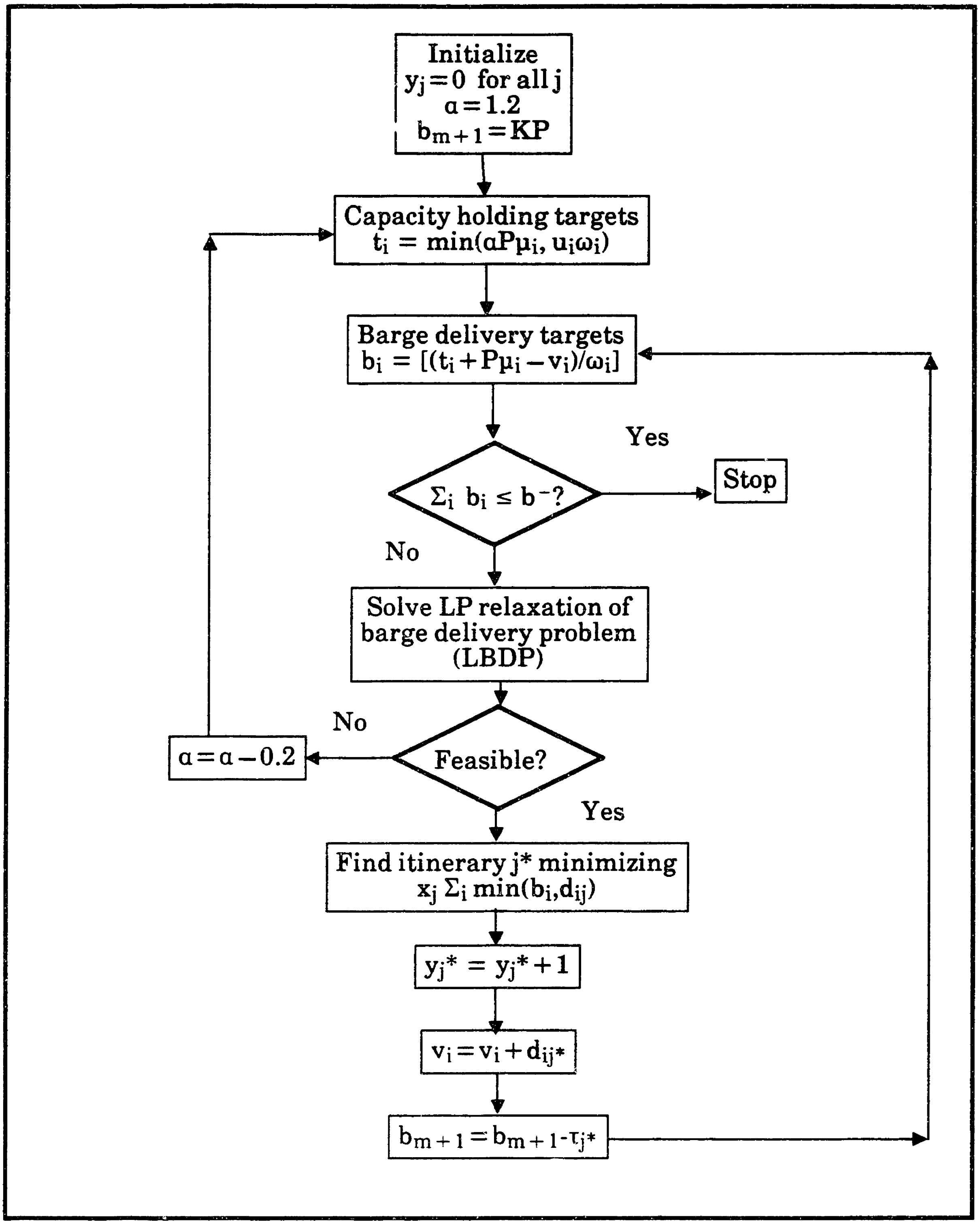
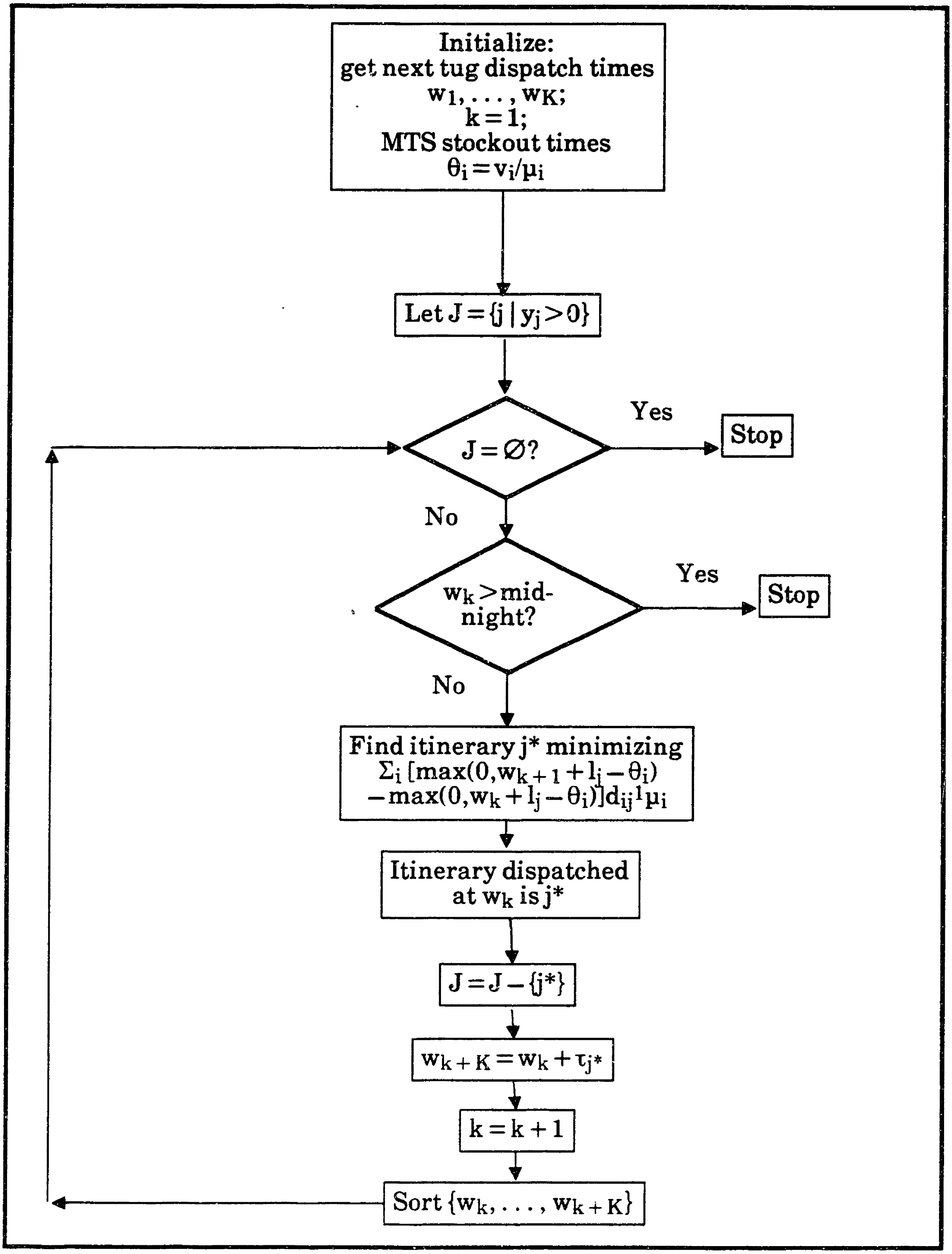


Exhibit 7.5: Itinerary Sequencing Phase of ADE Algorithm



store (aside from the loading barge). Finally, let α be some constant (α has been initialized at 1.2 in our computational experiments). Then we set

$$t_i = min\left(\alpha P\mu_i , u_i\omega_i\right).$$

The number of barges we initially seek to deliver to each MTS is the number that causes its holding target to be equaled or exceeded, taking into account how much barge space is currently on hand and how much is likely to be consumed over the day. Let v_i be the number of tons of refuse that can be accommodated solely with the barges currently at MTS i plus any barges en route to MTS i. If b_i additional barges are dispatched to MTS i over the course of the day, then the targets for next midnight will be met provided that

$$v_i + \omega_i b_i - P\mu_i \geq t_i \quad \forall\ i.$$

Hence, the initial barge delivery targets $\{b_i\}$ are determined by

$$b_i = \left\lceil \frac{t_i + P\mu_i - v_i}{\omega_i} \right\rceil, \ i=1,\ldots,m,$$

where $\lceil x \rceil$ here is the smallest integer greater than or equal to x, and m is the number of MTSs.

The next step consists of seeking a collection of itineraries J whose aggregate delivery meets or exceeds the delivery targets, while being feasible to execute given the present tug fleet. To accomplish this, we attempt to solve an integer program called the barge delivery problem (BDP). The constraints of the BDP reflect the delivery targets and tug availability. Our objective in solving the BDP is simply to find a feasible solution, or ascertain that none exists. We are therefore quite flexible as to what objective function we may use in the definition of the problem. Our choice of objective function is based on our use of a linear programming relaxation heuristic to find a feasible solution. We seek cost coefficients for the itineraries that tend to yield solutions from which feasible integral solutions are most easily derived. After testing several functions of itinerary properties, it was found that the cost coefficient which worked best for our needs was

$$c_j = (4 - m_j) \times d_j,$$

with m_j being the total number of MTSs visited in itinerary j, and d_j the total delivery in the itinerary. The $4 - m_j$ term prevented only fractions of single-customer itineraries from being included in the optimal solution, and the d_j term discouraged delivering more to MTSs than was necessary (tendencies which were observed in the optimal solution under other objective functions). The complete barge delivery problem is:

$$min \sum_{j=1}^{n} c_j y_j \tag{7.7}$$

$$s.t. \sum_{j=1}^{n} d_{ij} y_j \geq b_i \quad \forall\, i \tag{7.8}$$

$$\sum_{j=1}^{n} \tau_j y_j \leq b_{m+1} = KP \tag{7.9}$$

$$y_j = 0 \ or \ 1. \tag{7.10}$$

In the BDP, τ_j is the duration of itinerary j in pulses and K is the number of tugs. Its LP relaxation, denoted LBDP, is obtained by dropping the integrality constraint (7.10) and substituting the decision variables x_j for y_j.

Our LP relaxation heuristic works as follows: We start with a solution $y_j = 0$ for all j. Then we solve the LBDP. Depending upon whether it has a feasible solution or not, one of two actions is taken. If the problem proves infeasible, we reduce the multiplier α by 0.2 (or some other increment). Then we generate new b_i's and re-solve the LBDP. Once we find an α which yields an optimal solution, we look at all itineraries j for which $x_j > 0$ in the solution. We have found that by finding the j^* which minimizes

$$x_j \sum_{i=1}^{m} min\left(b_i , d_{ij} \right),$$

and setting $y_{j*} = y_{j*} + 1$, we move toward a good integral solution. We add itinerary $j*$ to the set of itineraries J to dispatch, update b_i to $max\ (0, b_i - d_{ij*})$, and update b_{m+1} to $b_{m+1} - \tau_j$. We then start the process all over again. The process continues until the sum of remaining delivery targets falls below some criterion.

Having the itineraries to dispatch over the course of the day in the set J, the question becomes, In what order are they sequenced? We try to sequence the itineraries to minimize lost refuse, in the following way:

Let θ_i be the time MTS i is expected to stock out of barge space:

$$\theta_i = \frac{v_i}{\mu_i}.$$

Also let the times of the first two dispatches of the day be approximated as w_1 and w_2. These times are both midnight if two tugs currently wait at FKL; otherwise, times corresponding to tugs not at FKL are obtained from estimating how long it will take them to return to FKL from whatever they are doing now. The itinerary assigned to the first dispatch is the one that produces the greatest savings in lost refuse if dispatched at w_1 instead of at w_2. If l_j is the average leadtime in itinerary j, then the savings for itinerary j is computed as

$$\sum_{i=1}^{m} \left[max\left(0, w_2 + l_j - \theta_i\right) - max\left(0, w_1 + l_j - \theta_i\right)\right] d_{ij}^1 \mu_i$$

($d_{ij}^1 = 1$ if MTS i receives a delivery in itinerary j, $=0$ otherwise). Suppose itinerary $j*$ has maximum savings. Then $j*$ is assigned to the first tug to leave FKL. The θ_i for all customers i in itinerary $j*$ are updated to show the time following their delivery on $j*$ that they run out of barge space, and the time of dispatch w_1 of the tug running $j*$ and the itinerary duration τ_{j*} are used to determine when this tug is next dispatched during the day. This process resumes with departure times w_2 and w_3 to select the itinerary to dispatch at w_2, and so on until all itineraries from J are assigned or the next midnight passes. Of course, the actual time of dispatch of a tug, when using the ADE algorithm, depends both on the exact time it becomes ready and on barge availability then and in the near future.

Obviously, many features of the ADE algorithm are *ad hoc*, and some could probably be improved upon. Nevertheless, we feel that the algorithm as it was implemented for our case study suited its purpose well, that of affording us some idea of how a partially adaptive dispatching algorithm performs relative to the adapted FVD algorithm. The performance of the ADE algorithm could have been possibly enhanced in some scenarios were we to allow rescheduling during the day. Again, our intention was not to devise the best algorithm reminiscent of actual dispatching practices, but to suggest and to test one such algorithm against the performance of the technique we have devoted this dissertation to.

Other procedures can be contemplated for aiding tug and barge dispatching in the MWTS. The dispatching algorithm used in the Barge Operations System Simulator (or BOSS, described in [21]), though not strictly a variant of either the FVD or the ADE algorithm, bears resemblances to both. The BOSS dispatching algorithm features dispatch-by-dispatch scheduling, and tries to optimize dispatches over a finite time horizon, like the adapted FVD. However, no penalties are assessed on the state of the system at the end of the horizon, so the resemblance is more to the myopic algorithm described in Chapter VIII. Each dispatch is decided by considering sequences of current and future itinerary dispatches, with the objective of minimizing the expected amount of lost refuse. This procedure is similar to the second phase of the ADE algorithm. The BOSS algorithm was designed for dispatching within a more complex MWTS model; it was not adapted to our version of the model for our computational experiments in the next chapter.

CHAPTER VIII

COMPUTATIONAL RESULTS FOR MWTS DISPATCHING

8.0 Introduction

In this chapter, the adapted Future Value Decomposition algorithm and the Actual Dispatching Emulator algorithm intended to emulate present dispatching practices are run side-by-side on a series of scenarios deriving from actual or potential situations arising in New York City's marine waste transport system.

8.1 Aims of the Computational Experiments

It will be recalled that the FVD algorithm addresses a variant of the deliverer dispatch problem. Dispatching in the MWTS differs in many ways from dispatching in the deliverer dispatch model. Hence, some doubt arises as to whether an FVD-type algorithm is suited to the demands of MWTS dispatching, even after certain adaptations are instituted. The adapted FVD algorithm is put on trial in this chapter. The charge is that it does not improve much upon current dispatching practices, as represented by the ADE algorithm. In a moment, the results of that trial. . .

8.2 The Scenarios

FVD performance is compared to ADE performance across a number of scenarios. The scenarios vary in the numbers of barges and tugs available, in Fresh Kills Landfill unloading rates, in the coefficient of variation of daily demand, and in marine transfer station refuse inflow rates. Exhibit 8.1 lists most of the details of how the scenarios differ. All share the same itinerary set, and system activity in each of the scenarios is governed by the set of dynamics described in Chapter VI.

Exhibit 8.1: MWTS Scenarios

Scenario	# Barges	#Tugs	Digger speed (tons/pulse)	Coefficient of variation of daily demand
1	45	3	112.5	0.1
2	40	3	112.5	0.1
3	35	3	112.5	0.1
4	35	3	112.5	0.2
5	50	4	170.0	0.1
6	45	4	170.0	0.1
7	50	3	170.0	0.1
8	50	4	170.0	0.2

Exhibit 8.2 provides the mean refuse inflow for each MTS for the scenarios. The mean inflows in the second four scenarios are those of the first, multiplied by 1.5. The first set corresponds to mean system workloads in 1982, at the time that work on building a simulation of the MWTS for barge purchasing decisions (documented in [21]) was being performed. The future event motivating the MWTS simulation project was the shutdown of a major truck-based landfill, the Fountain Avenue Landfill (shown on the map in Exhibit 6.1), scheduled to take effect in 1985. The result of this shutdown, as far as the MWTS was concerned, was a 50% jump in refuse inflow to MTSs. This is the reason why inflow rates are increased 50% in the second set of scenarios.

Each scenario was simulated for two weeks without gathering statistics (to reduce the serious start-up biases that were observed in practice runs) and ten weeks with statistics gathering. Hence, scenarios were of fixed length. The Law and Carson method for terminating simulations, described in Section 5.1.4, was not used; however, their method for calculating the final confidence interval was used here. Let the reader beware that any stated confidence interval may possibly be based on a biased estimate of variance. The simulation was run for ten weeks in the

Exhibit 8.2: MTS Refuse Inflow Rates

MTS	Scenario range	
	1-4	5-8
1	13.13	19.70
2	7.00	10.50
3	8.53	12.80
4	14.89	22.34
5	10.00	15.00
6	5.73	8.60
7	21.82	32.73
8	18.91	28.37

Table entries are in tons per pulse

hope that such biases would be minimal. The generators for the gamma-distributed random numbers are the ones given in [1] (Johnk's algorithm) and [6]. Uniform random numbers to feed these generators were obtained from the generator listed in [27]. Programs are listed in [20].

8.3 The Algorithms

The ADE algorithm was executed in each scenario, but on a different refuse inflow pattern in Scenarios 5-8, due to a late change installed in the algorithm. The FVD algorithm actually becomes a whole class of algorithms when applied to MWTS dispatching. Not only can the replenishment cost determination method and dispatch failure probability be twiddled with, but there also arise choices for the following quantities:

* the inventory holding cost:
* the transportation cost per pulse:
* the number of pulses in a period.
* the number of periods in the scheduling horizon

Also, a myopic algorithm derives from the FVD procedure when no penalties are assessed on the final state entered. We test a number of FVD variants along with the ADE algorithm on each of the scenarios. The details of the FVD variants used, and the labels according to which they are referred to in the discussion below, are supplied in Exhibit 8.3.

8.4 The Results

Exhibit 8.4 carries the dispatch performance results for each of the eight simulated scenarios. The figures in each cell of the table are, on the top, the estimated mean percentage of lost refuse, and below in parentheses, the half-width of a 90% confidence interval for the mean percentage of lost refuse. The mean percentage given is not the measured mean from the simulation, but rather a mean inflow-weighted average of the fraction of time each MTS is stocked out of barges. Since, in many cases, stockouts occur very infrequently, the total number of tons lost measured in the simulation is the sum of a small number of random variables, each with a very high standard deviation (it equals or exceeds the mean). More of interest at a particular MTS is the mean amount of refuse that would arrive there during the fraction of time that it is out of barge capacity (fraction of time out of stock is measured empirically within the simulation). Our measure is the ratio of the sum of the mean amounts of lost refuse divided by the total mean arrival rate. The

Exhibit 8.3: MWTS Dispatching Algorithms Tested

FVD variant label	Penalty derivation method	Holding cost	Transp-ortation cost per pulse	Periods per horizon	Hours per horizon
Standard	initial	40	50	4	15
0/1	initial	0	1	4	15
2:450	initial	40	50	2	7.5
Updated	updated	40	50	4	15
M4:900	myopic	40	50	4	15
M2:450	myopic	40	50	2	7.5

confidence intervals shown are derived from observed percentages of lost refuse, however. The lowest mean percentage of refuse lost measure in each scenario is printed in bold in Exhibit 8.4.

We now compare results of the simulations along several dimensions.

8.4.1 FVD vs. ADE

Scenario 1 is the scenario that most closely resembles the actual MWTS in 1982, in terms of number of barges, tugs, and diggers operated. The ADE algorithm performed respectably in this scenario, where less than 3% of the refuse delivered to MTSs was lost. When viewed along with the performance of the FVD algorithms, however, ADE's performance pales by comparison. The same holds true in all the scenarios. The *ratio* of lost tonnage of ADE to that of the Standard FVD algorithm was never less than 2.6, and usually much more. Hence, to the extent that the ADE implementation in this dissertation reflects current dispatching practice, we conclude that dispatching performance can be greatly improved by employing an FVD-type algorithm for dispatch support.

The advantage of using the FVD algorithm instead of the ADE may be exhibited in another way. The base scenario provides 45 barges to do marine waste transport work. Suppose we regard the 3% lost refuse rate as the maximum tolerable rate. The Standard FVD algorithm, as well as most of the other ones tested, produces a more tolerable lost refuse rate with *ten fewer* barges. With refuse inflow increased 50%, several additional simulation runs of the ADE algorithm showed that the lost refuse rate could be cut to about 2% if 4 tugs and 60 barges were available; the FVDs could do that with 45 barges (the base size of the barge fleet). The point here is that new barge acquisitions, for replacement or for accommodating greater demand, can be forestalled or eliminated by implementation of an FVD algorithm in a CAD system. The same can undoubtedly be said for additional tug hires and digger acquisitions.

Exhibit 8.4: MWTS Simulation Results

Scen-ario	ADE	Standard	0/1	2:450	Updated	M4:900	M2:450
1	2.9	0.1	0.1	**0.0**	0.2	0.2	7.1
	(0.8)	(0.1)	(0.1)	(0.0)	(0.1)	(0.1)	(1.0)
2	13.2	0.3	0.3	**0.1**	0.6	0.2	8.6
	(2.1)	(0.1)	(0.1)	(0.1)	(0.2)	(0.1)	(1.2)
3	37.5	1.9	3.1	**1.3**	1.8	1.5	8.8
	(2.5)	(0.6)	(0.5)	(0.8)	(0.4)	(0.4)	(0.8)
4	36.1	4.5	5.2	4.2	**3.2**	3.6	12.2
	(3.6)	(0.9)	(0.9)	(1.0)	(0.7)	(0.9)	(1.7)
5	13.6	0.6	0.5	**0.2**	0.4	0.3	21.3
	(1.6)	(0.2)	(0.2)	(0.1)	(0.2)	(0.2)	(1.3)
6	22.1	1.7	1.3	1.3	1.9	**0.5**	19.9
	(2.2)	(0.4)	(0.3)	(0.3)	(0.6)	(0.2)	(1.8)
7	26.0	10.0	9.5	11.0	**8.2**	16.0	46.5
	(1.6)	(1.0)	(1.4)	(1.2)	(1.0)	(0.8)	(0.6)
8	16.7	1.1	1.5	1.5	1.4	**0.7**	13.5
	(2.2)	(0.3)	(0.5)	(0.6)	(0.5)	(0.3)	(0.2)

Table entries are simulated mean percentage of lost refuse in top of each cell and half-width of confidence interval of this measure in bottom.

8.4.2 Initial vs. Updated FVD

The updated FVD algorithm used here applied updates only to the determination of replenishment costs. The fact that the MWTS model operates on close to continuous time made it difficult, though not impossible, to quantify a dispatch failure in this circumstance. The dispatch failure probability was left at zero in all runs. This may explain in part why the updated FVD did not show consistent improvement over the Standard FVD, from which it differs by only its updating procedure. It is true that even in the simulations of Chapter V, updating did not uniformly improve dispatching performance in every instance. One cause for occasional updating ineffectiveness cited there was that several more update iterations may have been necessary to allow the procedure to converge to the best set of replenishment costs and failure probabilities. The same may be true in this case. More extensive testing of updating in the FVD algorithm is called for here.

8.4.3 Resource Costs

The 0/1 FVD algorithm is configured as the Standard FVD algorithm except that it charges nothing to hold inventory and costs tug transport at the exceedingly cheap rate of $1 per pulse. No consistent dominance relationship between the 0/1 and Standard FVDs materializes in the simulations. One might think that under tight resource constraints, the Standard FVD would perform better. This was observed in the base inflow rate Scenario 3, but not in the added inflow rate Scenarios 6 and 7, each scenario of which was somewhat insufficient in the number of barges or tugs supplied. There did seem to be some consistent difference in performance in the higher inflow variation Scenarios 4 and 8. We conjectured in Section 7.7 that under-costing inventory holding and transportation may leave the system vulnerable to heavier barge demands than usual. This conjecture may be construed to be supported by the lower proportions of lost refuse for the Standard FVD algorithm in the high demand variability scenarios. Still, the improvement may not be significant.

8.4.4 Scheduling Horizons

Consider the performance of the 2:450 FVD algorithm. The 2:450 FVD is the Standard FVD with half the extended scheduling horizon. One would think that an algorithm which "sees" two periods into the future would not perform as well as one which looks ahead four periods. On the contrary, the 2:450 FVD outperformed the Standard in all but the last two scenarios. Actually, the point about one algorithm seeing further into the future is not accurate, since both algorithms use penalty structures based on infinite horizon performance. The difference is that the 2:450 calculates expectations of actual dispatch decisions over a shorter time period than the other.

How can the Standard FVD's inferiority to techniques based on shorter scheduling horizons be explained? Random statistical fluctuation cannot account entirely for this behavior, because the methods were tested on identical refuse arrival patterns. The only explanation conceivable at this time is that the accuracy of the estimates of costs incurred during the scheduling horizon and/or future planning horizon deteriorate somewhat as the length of the scheduling horizon increases, thereby causing some misguidance as the master scheduling problem is solved from dispatch to dispatch. We suggest that the FVD algorithm feature most likely to be the main source of this inaccuracy is the restrictions on inclusion of MTSs in itineraries over the scheduling horizon.

Recall that customers are entitled to have at most one delivery outstanding at any time in the FVD algorithm, may receive no more than one replenishment (of up to a vehicle load) per period in the deliverer dispatch subproblems, and may not be included in more than one itinerary in feasible solutions to the master scheduling problem. In the MWTS implementation of the extended scheduling horizon version of the FVD algorithm, these restrictions are interpreted to mean that no more than one delivery may be outstanding at the end of any period in the scheduling horizon, and at most one replenishment per scheduling horizon may be planned for in the DDSPs and the MSP. Dispatch schedules are updated with each dispatch, though, so the only restriction that binds actual dispatching is that no MTS is included in an itinerary whenever it is due to receive a delivery more than one period into the future from a previously dispatched tug. In other words, MTSs may have received in retrospect two dispatches in a given scheduling horizon. The likelihood of this event, and therefore the degree of misrepresentation of MWTS operations within the FVD algorithm, increases as the scheduling horizon lengthens. The inevitable misrepresentation of MWTS activity in the future planning horizon, which constitutes a greater proportion of the entire horizon in the 2:450 FVD algorithm, may have less effect than the misrepresentation within the longer scheduling horizon of the Standard FVD. This is the only interpretation we can offer for the observed behavior.

8.4.5 FVD vs. Myopic Algorithms

One of the major surprises of the computational tests was the performance of the myopic algorithm based on a 15-hour scheduling horizon. Percentage of tons lost for the M4:900 algorithm exceeded the best observed percentage by less than 0.4% in all scenarios except Scenario 7, where tug resources were unrealistically tight. M4:900 even "won" two of the scenarios. We are even more hard-pressed to rationalize these findings than we were in the last section.

Let us first review the results for the M2:450 algorithm. It has half the scheduling horizon of the M4:900 algorithm, and performs far worse. In fact, the ADE algorithm outperforms M2:450 in several scenarios. These observations jibe with the inconsistent and usually poor results obtained with myopic dispatching in Chapter V. M2:450 looks ahead 7.5 hours, long enough to foresee what happens to MTSs included and not included on the current dispatch, but often not long enough to also consider the consequences of the dispatch of the next available tug, which

M4:900 usually can do. Herein may lie part of the reason for M4:900's surprising dispatching ability, but this alone does not explain all.

The disturbing thing about the M4:900 algorithm, if one can find disturbing things about quality dispatching, is that it makes do without any consideration of what happens after 15 hours. The conclusion being thrust upon us is that it usually suffices in this system to dispatch to prevent near-term stockout, when "near-term" extends several dispatches into the future. We draw a parallel here to the theoretical foundations of the FVD penalty functions. Penalties are intended to approximate expected future values, in the sense that the term is used in the description of the policy iteration algorithm in Section 4.1. The expected future values, in turn, are composed of relative values, which measure the difference in costs over an infinite horizon between starting in a given state and starting in the steady state, using a particular stationary dispatch policy. The contribution made in each period to the relative value diminishes with each successive period, because the system moves closer and closer to the steady state. What the M4:900 results may be saying is that the point at which contributions to the dispatching objective function become superfluous occurs less than 15 hours into the future, in most situations. Only further research can help us fully explain this intriguing phenomenon.

8.5 A Note on Comparisons with the Actual MWTS

This section reviews the relationship between the model and reality, particularly with regard to performance measures. The performance measure we have been relying upon is the percentage of lost refuse, meaning the percentage of refuse delivered to MTSs that could not immediately be dumped onto a barge. In reality, refuse is never "lost." When an MTS stockout occurs, one of two things happens:

(1) sanitation trucks wait in a queue at the MTS for an empty barge to be placed in loading position;

(2) the trucks are rerouted to another, less convenient disposal facility.

The way MTS stockout is treated by DOS personnel makes comparisons between model and reality difficult for a number of reasons. The first difficulty is associated with the source of the MTS refuse inflow characteristics in our computational experiments. Refuse inflow data was generated originally by consulting a set of scale tapes. Just before a sanitation truck dumps its load, it is weighed on a large scale and the weight printed on a paper tape connected to that scale. Obviously, then, our refuse inflow data is tainted to some degree if we want it to represent refuse originally destined for each MTS. This is because the trucks that have been rerouted do not cross the scales of their primary dumping facility. Unfortunately, no records are available of the frequency with which sanitation trucks are diverted to secondary and tertiary facilities. Similarly unavailable are records of temporary MTS shutdowns while waiting for empty barges. Were either or both of these types of data extant, we would be better able to evaluate acutal system performance, and to thereby compare the MWTS model with the actual operation.

A host of other performance measures can be extracted from MWTS operations and our model of them. However, these other measures only tell us indirectly and incompletely about how effectively the system is working, and hence must be considered auxiliary performance measures. Some of them are the following:

* mean amount of refuse (or number of barges) unloaded per day;

* mean cycle time of a barge (time between successive arrivals at FKL);

* mean time spent by barges at system facilities (MTS, SA, FKL, in tow) in various states (empty, full, loading, unloading);

* mean cycle time of tug;

* mean fraction of cycle tug spends waiting at FKL;

* mean time FKL diggers are idle;

* fraction of tug arrivals to FKL with less than four barges in tow (this is one of the few measures actually used by management to evaluate dispatching performance).

These quantities generally measure the utilization of the various system resources. Read in certain combinations, the auxiliary performance measures may describe how well the MWTS is executing its primary function. They may further serve as points of comparison between FVD algorithm performance within the model and actual dispatching performance. Even these measures would be difficult to acquire at present from the actual system, because the data exist only in raw and incomplete form. The Department of Sanitation is moving in the direction of increased computerization of its record-keeping, though, so some performance measures may become routinely available in the next few years.

CHAPTER IX

SUMMARY, CONCLUSIONS, AND DIRECTIONS FOR FUTURE RESEARCH

9.1 General Summary

We begin this last chapter with a restatement of the goal of this dissertation: to demonstrate the feasibility of a prescriptive capability for a computer-aided dispatching system supporting delivery dispatch decisions in an environment of demand uncertainty and supplier responsibility for customer inventories. This section summarizes the novel achievements of this dissertation.

The first step in attacking the problem of dispatch decision-making, when the supplier strives to balance customer inventory as well as transportation concerns, is to broaden the conception of the problem. We have described a class of vehicle dispatching problems, of which the types of problems we study in this dissertation constitute only a subclass. In the description of this class, we detect common components in all manner of vehicle dispatching problems. Further, the list of problem components provides a foundation for a formal classification scheme.

We have constructed a new model of a physical goods delivery operation and defined an associated optimization problem, the deliverer dispatch problem (DDP). The objective of the deliverer dispatch problem is to find a method of dispatching a set of vehicles to minimize a sum of transportation and inventory costs incurred per unit time over an infinite time horizon. The DDP is designed expressly to model those operations in which vehicles are sent out on new assignments shortly after they have returned from previous ones, and in which the decision of how much to deliver to a customer is left with the supplier. The model naturally incorporates random demand for the delivered good. We show that some variants of the DDP are instances of Markov decision problems.

The version of the DDP that we select for closer study is shown to be infeasible to solve exactly. To continue the pursuit of our goal, we have developed an heuristic algorithm for solving the DDP, which we have called the Future Value Decomposition (FVD) algorithm. This heuristic returns a dispatch decision in response to the current status of the delivery system being supplied to it. The FVD algorithm is based loosely on Markov decision theory, but is better explained thusly: It solves a finite horizon dispatching problem, named the "master scheduling problem" (MSP), with penalties of various sizes imposed according to what condition the system is in at the end of the horizon. The penalties are determined in a preliminary phase of the algorithm (the solution of a set of single-customer deliverer dispatch "subproblems," or DDSPs), and correspond in rough fashion to what is perceived as likely to happen beyond the end of the horizon. The roughness arises from a decomposition, necessarily imperfect, by customer of system activity, meaning that what happens to each customer in the future is forecast in isolation of what happens to the others. A more detailed summary of the FVD algorithm appears in the next section.

We have studied certain aspects of the FVD algorithm analytically, but our knowledge of how well it performs has been acquired by and large through simulation. The algorithm behaves well under a variety of conditions, certainly with respect to other procedures which may potentially be applied to the task of dispatching under the same conditions. It also executes very rapidly on a computer. The newness of the model actually made seeking relevant competitors difficult. The computational tests are more a comparison of dispatching *concepts* than of actual procedures in use today.

We have demonstrated the adaptability of the FVD algorithm by applying it to a real-world operation that does not strictly conform to the DDP model underlying the algorithm. This operation is the New York City Department of Sanitation's marine waste transport system (MWTS). A mathematical and somewhat streamlined model of the MWTS was designed. This model served as the basis for a simulation of the MWTS.

The performance of the adapted FVD algorithm in the simulated MWTS was excellent across a number of relevant scenarios, and clearly superior to an algorithm we developed that emulated current dispatching practices. Projected savings by switching to the FVD procedure, expressed as the number of additional barges needed in the current dispatching system to reach the same performance level as the FVD algorithm achieved, was on the order of 10 to 15 barges. Since each barge may cost somewhere between $600,000 and $1,000,000, the savings are indeed substantial.

9.2 Summary of the DDP Heuristic

This section summarizes the Future Value Decomposition algorithm in a little more depth. The basic idea of the FVD algorithm is to try to approximate the exact solution that can be obtained, but for most problems only in principle, with Markov decision theory. The stumbling block to the direct application of Markov decision theory to the DDP is the calcuation of what are known as "expected future values" that serve as penalty terms in a finite horizon dispatch optimization problem. The inclusion of penalties in the finite horizon problem's objective function permits long-term dispatching objectives to be pursued while considering short-term strategies. It is the expected future values, which are the optimal penalty terms to use, that we approximate.

The approximation of expected future values derives from a transmutation of the DDP at future decision points. The transmuted DDP is marked by the following alterations to the original DDP:

1) the itinerary set of the original DDP is replaced by a new itinerary set with all itineraries including one customer only;

2) the vehicle fleet is unlimited in size;

3) dispatches fail at random with a given probability.

Under these changes to the nature of the DDP at future decision points, the expected future value of a given decision in a given state equals a sum of expected future values by customer. The latter values are found as a set of DDP-like problems called "single-customer deliverer dispatch subproblems" is solved. The DDSPs are solved in advance of any dispatching, so that the component expected future values for each customer can be accessed by the procedure used to generate each dispatch. Since the decisions and dynamics of the transmuted DDP do not match exactly those of the original DDP, the expected future values derived by this decomposition by customer will be in error of the true values to some degree. We try to minimize that degree of error by judicious selection of the itinerary set and the dispatch failure probability for the transmuted DDP.

For each dispatch, a finite horizon dispatch optimization problem is solved. This problem is called the "master scheduling problem." The MSP's objective function consists of terms representing transporatation and expected inventory costs incurred during the horizon, plus penalties incurred at the end of the horizon, as functions of the itineraries dispatched during the horizon. Again, the penalty terms are approximated using the expected future values arising from the solutions of the DDSPs in the preliminary phase of the algorithm. Under the additional assumption that each customer may appear in at most one itinerary dispatched during the horizon, the MSP attains a relatively simple form and can be solved using integer programming methods or one of various heuristics.

9.3 Conclusions

The two main criteria for the evaluation of any proposed dispatch determination scheme, as cited at the beginning of this dissertation, are that it supply good dispatches, and that it not take too long to find them. The former criterion relates to the effectiveness of the scheme, and the latter to the potential for its being accepted by those who have the choice whether to use or not to use it. Judging from the amount of time the FVD algorithm needs to calculate a dispatch and from the quality of the resultant dispatches observed in all our computational experiments, we conclude that computer-aided dispatching systems possessing prescriptive capabilities are viable here and now for a broader class of delivery operations than may have been previously supposed.

It is true that the algorithm never was applied in a situation where a customer could hold more than five units of the delivered good, or where a vehicle could transport more than four. Also, the greatest number of customers the dispatcher ever had to deal with was twelve. Yet we feel that the procedures we have developed, or at least their driving concepts, may be incorporated in CAD systems for operations larger in one or more of the relevant dimensions (see the next section for suggestions of how this may be done). The basis for this conjecture lies in the robustness we have observed in the performances of different variants of the FVD algorithm in many problems. It seems that any concerted effort to instill long-term objectives in short-term dispatching is rewarded to some degree. The main contribution of this work may then be viewed as the provision of a consistent framework for making tradeoffs between short-term and long-term objectives in the types of problems we have examined.

9.4 Directions for Future Research

This dissertation represents the first formal study of the deliverer dispatch problem as we have defined it. Because the subject is so new, many interesting and important questions have yet to be answered. Some of them have been brought up during the course of this work, while others have not even been conceived yet. This section summarizes the questions raised in this thesis.

Some of the puzzling numerical results from the case study in Part II may indicate that the constraints on dispatching in the FVD algorithm are too restrictive in certain situations. Can it be made feasible computationally to allow multiple dispatches to a customer in the deliverer dispatch subproblems or the master scheduling problem?

More efficient heuristics to solve the MSP seem to be required if the FVD algorithm is to be applied to larger problems. Perhaps MSP heuristics can take computational advantage of the slow change in system characteristics from dispatch to dispatch in large systems. Also, can the FVD algorithm deal with large vehicle transport and customer holding capacities? The sizes of the itinerary set and the DDSPs grow uncomfortably large in these circumstances. One way of dealing with these situations may be to work in aggregated units of the delivered good, in which case we can apply directly the techniques we have developed and then disaggregate to obtain the final dispatch. This is reminiscent of what we did to adapt the FVD algorithm to MWTS dispatching; although we planned deliveries in units of barges, the delivered good was consumed in fractions of a barge. An interesting and related question about MWTS dispatching is: Would working in units of fractions of a barge (say, half- or quarter-barges) improve dispatching performance? Although the complexity of the DDSPs would grow, the itinerary set would remain the same size, because only whole barges may be transported.

If the DDSPs are of managable size but the itinerary set has grown to unwieldy dimensions, we may be able to employ itinerary elimination procedures so that the number of itineraries to include in any MSP is not too great. We already investigated one such procedure in Section 4.3.4, where the MSP was first detailed, that utilized the principle of domination. There, one itinerary could be shown to dominate all others visiting the same customer subset. Perhaps the domain of domination can be extended beyond the same customer subset to, say, all itineraries visiting customers in the same geographic region. Action elimination procedures that have been developed for assisting in solving Markov decision problems may provide a basis for applying an extended domination principle to itinerary elimination. Another potential basis for eliminating itineraries from consideration are the optimal replenishment policies from the DDSPs. Maybe it makes sense to eliminate an itinerary if the total delivery made to its customers in their DDSPs, given their current inventory levels, falls below some threshold based on the optimal replenishment policies.

The itinerary size problem may be attacked from the other direction, too. Suppose only a small set of "core" itineraries is considered for any dispatch. The resulting MSP would not be too difficult to solve. We may then be able to test exchanges of itineraries in the solution set with other itineraries not members of the core set to see if dispatch performance can be improved upon. The form of the dispatch cost C_j^+ given in (4.36) also gives us the opportunity to quickly test itineraries to see if they are useful or not, becuase it is never useful to dispatch an itinerary j wtih $C_j^+ > 0$. For application of the DDP to larger problems, more research into itinerary elimination procedures may be quite important.

Much work remains in the development of good methods for establishing replenishment costs and dispatch failure probabilities for the DDSPs. We have made several suggestions about what may be done, but have only had the liberty to investigate at a very superficial level. Along the same lines, can missed delivery target effects be represented more realistically in the DDSPs? We have assumed in the DDSPs that a customer receives either its optimal replenishment quantity or nothing. Can we design DDSPs in which customers may receive deliveries in between these amounts, at frequencies approaching those occurring in the actual system? Another approach is to do away with dispatch failures entirely, and to discourage over-utilization of the vehicle fleet by increasing the replenishment costs in the DDSPs.

The DDSPs as currently formulated also completely discard vehicle availability information. Perhaps dispatch failure probabilities may be made to depend on the number of vehicles available or some other quantity derived from the vehicle time-to-availability measures. The DDSP formulation must then describe the dynamic behavior of the representation of systemwide vehicle availability solely as a function of decisions made for the single customer. This poses the major obstacle for vehicle availability representation in the DDSPs.

A very interesting question is how well an implemented FVD algorithm with periodic updating would perform in a system where demand means have a tendency to drift over time. We have not had the opportunity to study this topic, but practical concerns create an interest in it. Also, can the FVD algorithm be adapted feasibly to handle the case of cyclically varying demand distributions?

It would be valuable to have some means of establishing lower bounds on cost rates in DDPs more general than the one we developed bounds for in the dissertation.

The chapter on computational experience in this dissertation (Chapter V) went a long way in answering some of the questions arising in connection with the development of a new algorithm. But more work is called for. In particular, sensitivity analyses should be conducted to study the effects of vehicle fleet capacity, demand coefficient of variation, and other factors on the effectiveness of the FVD algorithm.

Finally, the more specialized questions that could not be answered completely concerning the application of the FVD algorithm to MWTS dispatching merit further study. Why does myopic dispatching perform as well as FVD dispatching? Why do shorter horizons work better than long ones? The answers to these questions will teach us even more about the quality of the FVD algorithm for solving deliverer dispatch problems.

APPENDIX A

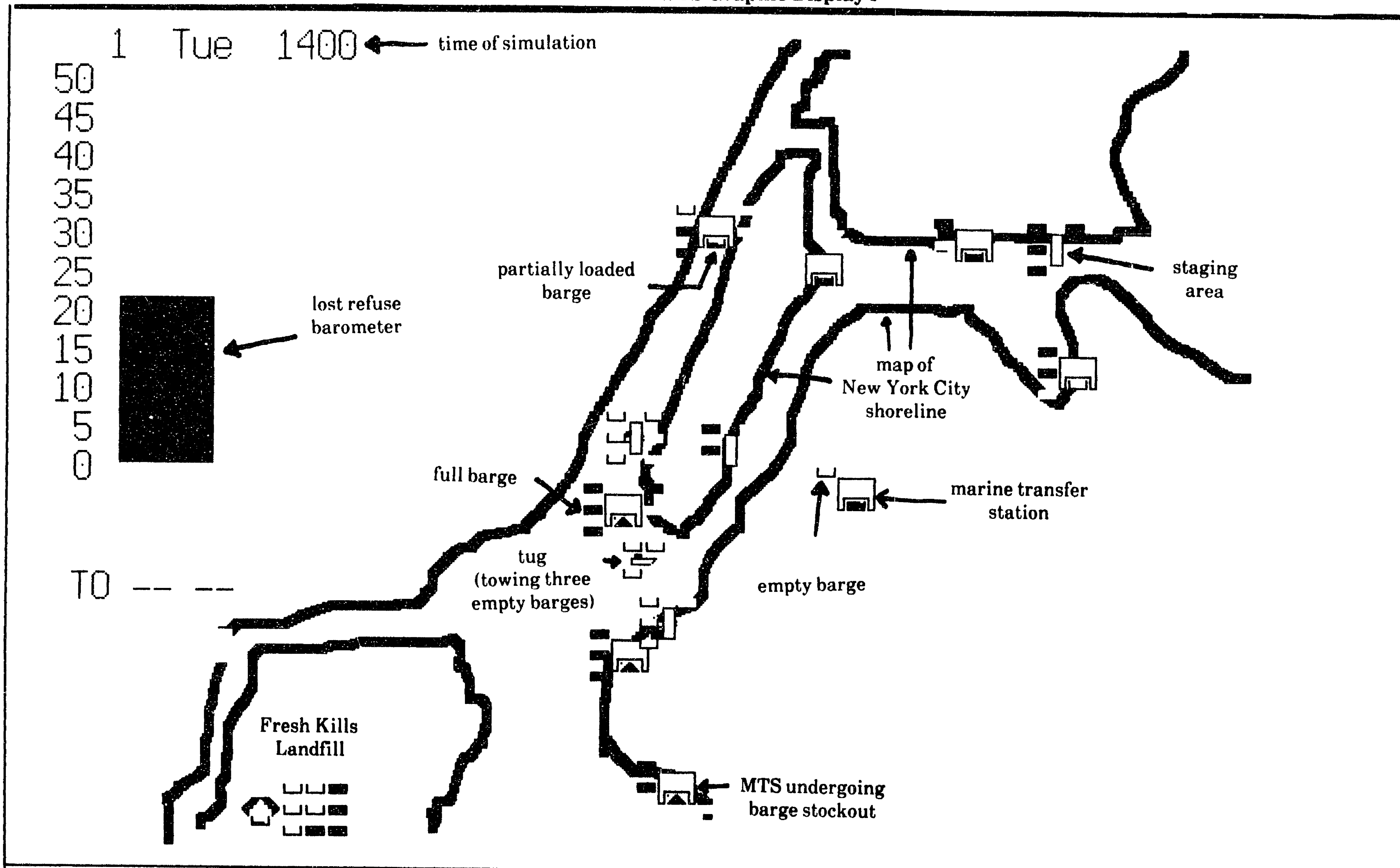
GRAPHIC DISPLAY OF THE MWTS

During the course of research on the case study of this dissertation, the author developed a graphic display of the simulated marine waste transport system on an Apollo DN660 workstation. This appendix discusses the display and considers its possible uses.

The development of the MWTS graphic display began as an exercise in computer graphics programming for the author. After the graphics routines had been debugged, though, the display supplied some unexpected dividends. Logic errors in the MWTS model (for instance, a vanishing barge phenomenon) could easily be detected by visual inspection. Also, poor dispatching practices became readily apparent. In one version of the set of dispatch procedures, full barges tended to pile up at staging areas, creating a dearth of full barges for digging out at Fresh Kills Landfill. Not only was bad dispatching noticed quickly, but little quirks in good dispatching procedures could be identified, and finetuning measures assessed in short order.

Exhibit A.1 shows what is displayed on the graphics monitor at an instant in time during the simulation. (Note: the picture was generated on a black-and-white monitor, whereas the main version tested operates on the color monitor. The color version's display is also less compact, making it easier to distinguish the various system elements visually.) Marine transfer stations, staging areas, Fresh Kills Landfill, tugs, empty barges, and full barges are each represented on the screen by a distinct icon (and, in the color graphics version, with different colors). To depict a partially loaded barge, the display fills in the empty barge icon in proportion to the fraction of total barge capacity occupied by refuse. The loading barge icon at each MTS is situated within the MTS icon in what looks like a slip When an MTS is in a stockout condition, a solid triangle replaces the barge icon in the MTS's slip. A rough map of New York City is drawn in the background. Samples of each symbol are labeled in the exhibit for reference.

Exhibit A.1: MWTS Graphic Display I



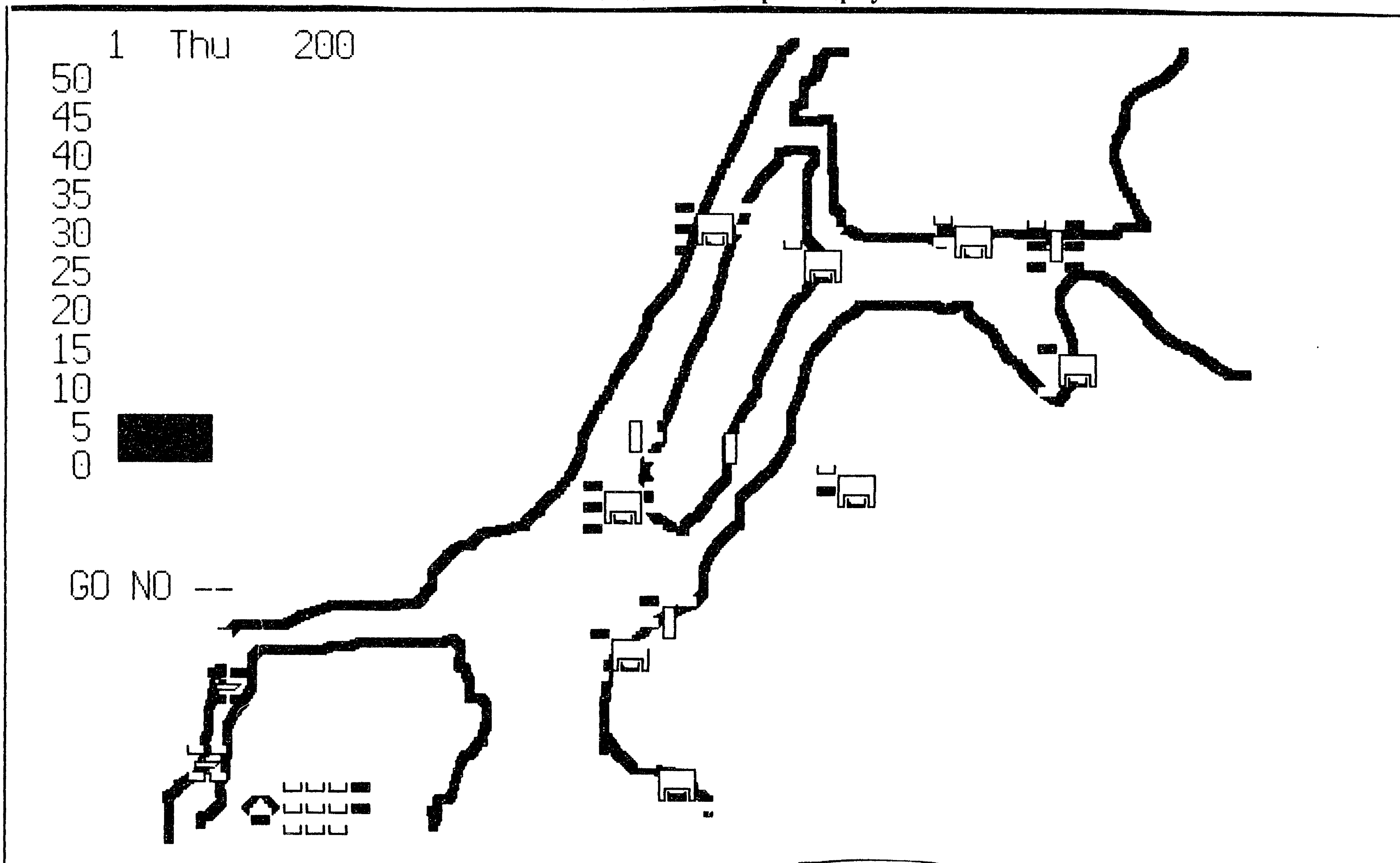
Additional information appears in the upper left-hand corner of the display. At the very top there, the week number, day, and time of day of the simulation are printed. Below this is located a graphic "barometer" of recent system performance. The numbers running down the left side refer to percentage of recently delivered refuse lost by the system. This performance measure is an exponentially smoothed average of percentage of lost refuse per pulse (one pulse equals 15 minutes here). In Exhibit A.1, this measure stands at about 22% (not very good); also, three MTSs were out of light barges at the moment this snapshot was taken.

What Exhibit A.1 cannot convey is that the display is animated. With every pulse, the entire display is updated to reflect the current status of the system. On the Apollo DN660, the rate of screen updates is quite fast, and the simulation time interval between updates short enough (15 minutes), so that tugs seem to move smoothly on their voyages, and barges gradually fill up with refuse. Depending upon which algorithm is used to dispatch tugs from FKL, the display may pause before some updates as a new dispatch is determined. A manual dispatching mode is also available. In this mode, the graphic display user instructs tugs where to go and which barges to take along by moving a mouse (the computer kind) over the adjacent tabletop and pressing buttons on top of the mouse.

Exhibit A.2 shows the same system later in the week, when it stands in much better shape.

With a couple of alterations, the graphic display of the MWTS may qualify as a *decision simulator* (DS), as it is described by Lembersky and Chi [18]. In their words, "a DS provides an interactive, visual (instead of numerical) simulation of the actual decision-making scenario, including the consequences of the decisions made" (p. 2). Lembersky and Chi offer three salient features of DSs. Below, we list these features and indicate how the MWTS display carries or can be made to carry each of them:

Exhibit A.2: MWTS Graphic Display II



1) "A DS provides a believable representation of the actual decision-making environment." Believability is ultimately in the eye of the beholder, and the designs of the icons and the map may not be to the liking of a particular user. Nevertheless, numerous studies cited by Lembersky and Chi emphasize that watching a symbolic representation of a system in action engenders more belief in a final report of system performance than simply being handed the final report would. The display also received favorable feedback from an official at the New York City Department of Sanitation after an informal demonstration.

2) "A DS is highly interactive and provides immediate feedback on the effect of decisions." When the display program is run in manual mode, the lost refuse barometer keeps the user informed about recent system performance. In color, readings below 5%, which do not represent serious system difficulties in and of themselves, are given in green on the baromenter. When the lost refuse rate surpasses 5%, the excess is displayed in red. Since refuse inflows are modeled as probabilistic, system performance for the same decisions may vary slightly from instance to instance. But the barometer does not portray performance in fine detail, so differences due to stochastic fluctuation are by and large not easily detectable. Lembersky and Chi do not discuss the case of implementing DSs in environments where some key system inputs have high degrees of variability, although this seems to us an important issue.

3) "A DS is easy to use, without special training." To use the MWTS display in manual mode, the user must simply learn what to do with the mouse in order to relay his/her decisions to the program. The user is cued on the display for what each button does at any particular point in the dispatch process. Basically, the user must decide either to delay dispatch (and if so, for how long), or to send the tug somewhere. In the latter case, the user rolls the mouse around to move a pointer on the screen to the tug's destination facility, then presses a button once for each barge to take along. It does not take long for a user to learn how to execute dispatches in the MWTS display program.

Here is one scenario for the employment of the MWTS display in real-time dispatching support (i.e., within a computer-aided dispatching system):

The computer and graphics display monitor are located in the dispatcher's office, and are on at all times. When the dispatcher is not involved with the planning of dispatches using the CAD system, the monitor carries a display of the current configuration of the system (with certain elements extrapolated from their last known values, if necessary). The dispatcher presses a key on the keyboard or a button on the mouse to switch from "display current system" to "plan dispatch" mode when he/she wants to plan for a current or future dispatch. He/she then types in the projected time that this dispatch will occur so that the CAD system can forecast what the MWTS will look like at dispatch time and display that on the monitor. What follows is a series of "what-if" investigations by the dispatcher. To relieve the burden of manually specifying each future dispatch in each "what-if" scenario, the dispatcher may set up the CAD system to automate some subset of future dispatches. Here is where the dispatching procedures studied in this thesis may be implemented. When the dispatcher settles on a dispatch, he/she enters this information into the CAD system, where it is stored and made available for future reference.

The MWTS graphic display does not require much modification to serve in the capacity outlined above. In its present form, it also can be used to aid long-term MWTS decisions, much as the Barge Operations System Simulator described in [20] did. The user may weigh alternate decisions in barge purchasing, tug hiring, MTS shutdown, and other strategic actions by visually inspecting their consequences as the MWTS is simulated on the display. The reader is invited to consult [18] and its references for a broader discussion of the potential impacts of visual feedback of decisions in a simulation environment.

APPENDIX B

DESCRIPTION OF INSTANCES SIMULATED IN CHAPTER V

In this appendix, we provide the data for each of the DDP instances simulated during the computational experiments reported on in Chapter V of this dissertation.

Parameters for each DDP instance:

instance	number of customers	number of vehicles	customer holding capacity	vehicle capacity	maximum demand	lost demand cost
1	6	4	3	3	3	15
2	6	6	4	3	3	6
3	6	4	4	3	2	30
4	6	5	4	3	2	20
5	6	3	4	4	3	75
6	6	6	5	4	3	20
7	6	4	5	4	4	10
1x	12	6	3	3	3	15
2x	12	6	4	3	3	6
3x	12	8	4	3	2	30
4x	12	8	4	3	2	20
5x	12	6	4	4	3	75
6x	12	10	5	4	3	20
7x	12	6	5	4	4	10

Demand probabilities for each customer in each instance:

Instance 1

Customer	0 units	1 units	2 units	3 units
1	0	0.60	0.40	0
2	0	0.14	0.52	0.34
3	0	0.91	0.08	0.01
4	0.03	0.09	0.73	0.15
5	0.03	0.94	0.02	0.01
6	0.05	0.91	0.04	0

Instance 1x

Customer	0 units	1 units	2 units	3 units
1	0.15	0.53	0.29	0.03
2	0.13	0.51	0.29	0.07
3	0.16	0.70	0.12	0.02
4	0.10	0.48	0.24	0.18
5	0.23	0.75	0.01	0.01
6	0.24	0.73	0.03	0
7	0.24	0.73	0.03	0
8	0.15	0.53	0.29	0.03
9	0.10	0.48	0.24	0.18
10	0.13	0.51	0.29	0.07
11	0.16	0.70	0.12	0.02
12	0.23	0.75	0.01	0.01

Instance 2

Customer	0 units	1 units	2 units	3 units
1	0.09	0.91	0	0
2	0.01	0.11	0.55	0.33
3	0.02	0.25	0.73	0
4	0	0.78	0.22	0
5	0.04	0.93	0.02	0.01
6	0.03	0.19	0.74	0.04

Instance 2x

Customer	0 units	1 units	2 units	3 units
1	0.03	0.19	0.74	0.04
2	0.02	0.12	0.70	0.16
3	0	0.78	0.22	0
4	0.01	0.11	0.55	0.33
5	0.03	0.19	0.74	0.04
6	0	0.78	0.22	0
7	0.03	0.19	0.74	0.04
8	0.02	0.25	0.73	0
9	0.04	0.93	0.02	0.01
10	0.01	0.11	0.55	0.33
11	0.01	0.11	0.55	0.33
12	0.01	0.11	0.55	0.33

Instance 3

Customer	0 units	1 units	2 units
1	0	0.80	0.20
2	0	0.70	0.30
3	0.10	0.90	0
4	0	0.70	0.30
5	0.02	0.25	0.73
6	0.01	0.88	0.11

Instance 3x

Customer	0 units	1 units	2 units
1	0.02	0.25	0.73
2	0	0.70	0.30
3	0	0.70	0.30
4	0	0.70	0.30
5	0	0.80	0.20
6	0.02	0.25	0.73
7	0.02	0.25	0.73
8	0.10	0.90	0
9	0	0.70	0.30
10	0	0.70	0.30
11	0.01	0.88	0.11
12	0	0.80	0.20

Instance 4

Customer	0 units	1 units	2 units
1	0.30	0.70	0
2	0.40	0.60	0
3	0.18	0.44	0.38
4	0.20	0.70	0.10
5	0.40	0.60	0
6	0.70	0.30	0

Instance 4x

Customer	0 units	1 units	2 units
1	0.40	0.60	0
2	0.40	0.60	0
3	0.40	0.60	0
4	0.30	0.70	0
5	0.70	0.30	0
6	0.70	0.30	0
7	0.30	0.70	0
8	0.20	0.70	0.10
9	0.20	0.70	0.10
10	0.20	0.70	0.10
11	0.18	0.44	0.38
12	0.18	0.44	0.38

Instance 5

Customer	0 units	1 units	2 units	3 units
1	0.24	0.73	0.03	0
2	0.15	0.53	0.29	0.03
3	0.10	0.48	0.24	0.18
4	0.13	0.51	0.29	0.07
5	0.23	0.75	0.01	0.01
6	0.16	0.70	0.12	0.02

Instance 5x

Customer	0 units	1 units	2 units	3 units
1	0.23	0.75	0.01	0.01
2	0.15	0.53	0.29	0.03
3	0.15	0.53	0.29	0.03
4	0.16	0.70	0.12	0.02
5	0.10	0.48	0.24	0.18
6	0.24	0.73	0.03	0
7	0.23	0.75	0.01	0.01
8	0.10	0.48	0.24	0.18
9	0.23	0.75	0.01	0.01
10	0.13	0.51	0.29	0.07
11	0.23	0.75	0.01	0.01
12	0.24	0.73	0.03	0

Instance 6

Customer	0 units	1 units	2 units	3 units
1	0	0.15	0.59	0.26
2	0.01	0.11	0.55	0.33
3	0.09	0.91	0	0
4	0.01	0.88	0.11	0
5	0.02	0.12	0.70	0.16
6	0.02	0.21	0.73	0.04

Instance 6x

Customer	0 units	1 units	2 units	3 units
1	0.01	0.88	0.11	0
2	0.01	0.11	0.55	0.33
3	0.02	0.21	0.73	0.04
4	0.03	0.19	0.74	0.04
5	0.02	0.25	0.73	0
6	0.03	0.19	0.74	0.04
7	0.01	0.11	0.55	0.33
8	0.01	0.88	0.11	0
9	0.01	0.88	0.11	0
10	0.04	0.93	0.02	0.01
11	0	0.15	0.59	0.26
12	0	0.15	0.59	0.26

Instance 7

Customer	0 units	1 units	2 units	3 units	4 units
1	0.18	0.18	0.32	0.32	0
2	0.15	0.27	0.32	0.26	0
3	0.17	0.12	0.12	0.12	0.47
4	0.13	0.20	0.20	0.27	0.20
5	0.15	0.29	0.37	0.19	0
6	0.13	0.20	0.25	0.27	0.15

Instance 7x

Customer	0 units	1 units	2 units	3 units	4 units
1	0.18	0.18	0.32	0.32	0
2	0.13	0.20	0.20	0.27	0.20
3	0.15	0.29	0.37	0.19	0
4	0.18	0.18	0.32	0.32	0
5	0.15	0.29	0.37	0.19	0
6	0.13	0.20	0.20	0.27	0.20
7	0.13	0.20	0.25	0.27	0.15
8	0.17	0.12	0.12	0.12	0.47
9	0.13	0.20	0.20	0.27	0.20
10	0.15	0.29	0.37	0.19	0
11	0.17	0.12	0.12	0.12	0.47
12	0.18	0.18	0.32	0.32	0

Below we supply itinerary information for each instance simulated in Chapter V in the form of "basic itinerary sets." The full set of itineraries for each instance is generated from the basic itinerary set in the following way: For each basic itinerary, include in the full itinerary set all itineraries delivering at least one unit to each customer in the basic itinerary, provided that the total delivery on that itinerary is at least two units but no greater than the vehicle capacity of that instance. The other characteristics of that itinerary are the ones of the corresponding basic itinerary.

Basic itinerary sets:

Itineraries for Instance 1

Customers	Duration (periods)	Transport-ation Cost
1	1	10.7
2	1	11.2
3	1	10.6
4	1	11.1
5	1	10.1
6	1	9.3
1-2	2	14.7
2-3	2	15.1
2-4	2	17.3
3-4	2	14.9
2-3-4	2	19.4
4-5	2	14.1
5-6	1	13.5

Itineraries for Instance 2

Customers	Duration (periods)	Transportation Cost
1	1	9.5
2	1	8.8
3	1	8.6
4	1	11.8
5	1	5.8
6	1	11.0
1-2	1	11.9
1-3	2	14.2
2-3	2	12.7
1-2-3	2	15.8
3-4	2	13.5
3-5	1	11.7
5-6	2	12.7

Itineraries for Instance 3

Customers	Duration (periods)	Transport-ation Cost
1	1	14.9
2	1	21.9
3	1	26.1
4	1	30.9
5	2	35.3
6	2	37.4
1-2	1	31.1
1-4	2	39.5
2-3	2	35.4
2-4	2	36.8
1-2-3	2	44.5
4-5	2	42.3
5-6	2	42.8
4-6	2	44.4
4-5-6	2	49.8

Itineraries for Instance 4

Customers	Duration (periods)	Transport-ation Cost
1	1	0.9
2	2	1.0
3	2	1.6
4	2	1.1
5	2	1.5
6	2	1.1
1-2	2	1.4
3-4	3	1.8
5-6	3	1.7

Itineraries for Instance 5

Customers	Duration (periods)	Transport-ation Cost
1	1	6.5
2	1	5.1
3	1	7.3
4	1	6.5
5	1	6.3
6	1	7.8
1-2	2	8.9
2-3	2	9.2
1-2-3	2	13.0
2-4	2	8.9
5-6	2	10.9

Itineraries for Instance 6

Customers	Duration (periods)	Transportation Cost
1	1	1.7
2	1	1.1
3	1	1.8
4	1	1.3
5	2	2.2
6	2	2.2
1-2	2	2.0
1-4	2	2.0
2-4	1	1.7
1-2-4	2	2.5
2-3	2	2.3
3-4	2	2.2
2-3-4	2	2.7
5-6	2	2.7

Itineraries for Instance 7

Customers	Duration (periods)	Transportation Cost
1	1	21.7
2	1	18.5
3	1	23.9
4	1	17.2
5	1	9.7
6	1	17.6
1-2	2	28.3
2-3	2	30.5
4-5	1	22.3
5-6	1	21.0
4-6	2	28.0
4-5-6	2	33.7

Itineraries for Instance 1x

Customers	Dura-tion	Trans. cost	Customers	Dura-tion	Trans. cost
1	1	18.7	5-8	2	25.2
2	2	20.0	6-7	2	21.9
3	1	14.7	6-8	2	21.6
4	1	13.0	7-8	2	22.8
5	1	16.8	5-6-7	2	31.6
6	1	12.0	5-6-8	2	31.4
7	1	17.1	5-7-8	2	30.1
8	1	16.6	6-7-8	2	27.5
9	2	20.6	8-9	2	25.4
10	1	16.3	8-10	2	23.6
11	1	13.5	8-11	2	22.4
12	1	11.5	8-12	2	22.0
1-2	2	24.9	9-10	2	26.4
1-11	2	23.9	9-11	2	25.9
1-12	2	23.4	9-12	2	25.8
2-11	2	25.4	10-11	2	21.0
2-12	2	24.8	10-12	2	21.1
11-12	1	18.3	11-12	1	18.3
1-2-11	2	30.3	8-9-10	2	31.2
1-2-12	2	29.7	8-9-11	2	30.7
1-11-12	2	28.6	8-9-12	2	30.6
2-11-12	2	30.1	8-10-11	2	28.3
3-4	2	20.2	8-10-12	2	28.4
3-12	2	21.0	8-11-12	2	27.1
4-12	1	18.7	9-10-11	2	31.1
3-4-12	2	25.9	9-10-12	2	31.2
5-6	2	21.7	9-11-12	2	30.7
5-7	2	24.4	10-11-12	2	25.8

Itineraries for Instance 2x

Customers	Duration	Trans. cost	Customers	Duration	Trans. cost
1	2	16.5	3-4	2	19.8
2	1	12.2	3-5	2	19.7
3	2	17.1	3-4-5	2	23.4
4	2	14.0	6-7	1	11.0
5	2	13.8	6-8	1	12.5
6	1	5.9	7-8	1	11.9
7	1	7.7	6-7-8	2	15.2
8	1	9.1	7-9	2	16.2
9	1	12.9	7-10	2	19.3
10	2	16.2	8-9	2	16.0
11	2	16.2	8-10	2	19.1
12	2	17.2	9-10	2	18.8
1-2	2	19.9	7-8-9	2	18.8
1-6	2	19.0	7-8-10	2	21.8
2-6	2	14.7	7-9-10	2	22.0
1-2-6	2	22.4	8-9-10	2	21.9
2-4	2	18.4	7-11	2	19.2
2-5	2	19.2	7-12	2	20.1
4-5	2	17.6	11-12	2	20.0
2-4-5	2	22.0	7-11-12	2	23.0

Itineraries for Instance 3x

Customers	Duration	Trans. cost	Customers	Duration	Trans. cost
1	2	34.4	4-7	2	34.6
2	1	18.4	4-8	2	41.8
3	1	12.1	5-6	1	29.8
4	1	23.9	5-7	2	34.3
5	1	20.1	5-8	2	41.9
6	1	23.9	6-7	2	35.0
7	1	29.1	6-8	2	43.3
8	2	36.5	7-8	2	42.7
9	1	30.4	4-5-6	2	38.9
10	2	32.3	4-5-7	2	43.4
11	2	40.5	4-5-8	2	5.0
12	2	37.1	4-6-7	2	42.9
1-12	2	45.1	4-6-8	2	51.2
2-3	1	25.1	4-7-8	2	48.2
2-4	2	33.3	5-6-7	2	40.8
2-5	1	29.6	5-6-8	2	49.2
3-4	1	30.4	5-7-8	2	47.9
3-5	1	26.7	6-7-8	2	48.6
4-5	1	29.2	9-10	2	40.0
2-3-4	2	43.5	9-11	2	45.9
2-3-5	2	39.8	10-11	2	46.4
2-4-5	2	38.6	9-10-11	2	54.1
3-4-5	2	35.7	9-12	2	43.9
4-6	1	31.8	--	--	--

Itineraries for Instance 4x

Customers	Duration	Trans. cost	Customers	Duration	Trans. cost
1	2	1.4	5-7	3	1.9
2	2	1.0	5-8	3	1.9
3	2	1.2	6-7	3	1.9
4	2	1.3	6-8	3	1.9
5	2	1.6	7-8	2	1.6
6	2	1.7	5-6-7	3	2.1
7	2	1.4	5-6-8	3	2.1
8	2	1.3	5-7-8	3	2.1
9	2	1.3	6-7-8	3	2.1
10	2	1.4	7-9	2	1.7
11	2	1.0	7-11	2	1.7
12	2	0.9	8-9	2	1.6
2-3	2	1.4	8-11	2	1.6
1-2	2	1.6	9-11	2	1.6
2-4	2	1.6	7-8-9	3	1.9
1-3	2	1.6	7-8-11	3	1.9
3-4	2	1.6	7-9-11	3	2.0
1-4	2	1.7	8-9-11	2	1.8
1-2-3	3	1.8	10-11	2	1.6
2-3-4	3	1.8	11-12	2	1.4
1-2-4	3	1.9	10-12	2	1.7
1-3-4	3	1.9	10-11-12	3	1.9
5-6	3	1.9	--	--	--

Itineraries for Instance 5x

Customers	Duration	Trans. cost	Customers	Duration	Trans. cost
1	1	6.0	4-5	1	7.9
2	1	4.0	5-6	1	7.3
3	1	3.7	3-4-6	2	9.7
4	1	4.9	3-4-5	2	9.1
5	1	4.7	3-5-6	2	9.1
6	1	5.8	4-5-6	2	10.0
7	2	8.9	5-7	2	10.1
8	2	9.1	6-7	2	10.3
9	2	8.7	5-6-7	2	11.5
10	1	7.3	8-9	2	12.1
11	1	6.8	8-10	2	12.0
12	1	6.8	9-10	2	10.2
1-2	1	7.4	8-9-10	2	13.6
1-4	2	8.8	9-11	2	10.7
1-3	2	8.0	9-12	2	11.5
2-4	1	6.7	10-11	2	9.1
2-3	1	5.8	10-12	2	9.9
1-2-4	2	10.1	11-12	2	8.9
1-2-3	2	9.2	9-10-11	2	11.9
1-3-4	2	10.0	9-10-12	2	12.7
2-3-4	1	7.9	9-11-12	2	12.7
3-4	1	6.1	10-11-12	2	11.1
3-6	1	7.6	1-11	2	9.7
3-5	1	6.8	1-12	2	9.0
4-6	2	8.5	1-11-12	2	11.0

Itineraries for Instance 6x

Customers	Duration	Trans. cost	Customers	Duration	Trans. cost
1	2	2.4	6-7	1	1.7
2	1	1.9	5-7	2	2.2
3	1	1.6	5-6-7	2	2.6
4	1	1.8	7-8	1	1.7
5	1	1.6	7-9	2	2.3
6	1	1.0	7-11	1	1.8
7	1	1.1	7-10	1	1.8
8	1	1.2	8-9	2	2.2
9	1	1.8	8-11	1	1.8
10	1	1.1	8-10	1	1.8
11	1	1.1	9-11	2	2.3
12	2	2.1	9-10	2	2.4
1-2	2	2.8	7-8-9	2	2.6
2-10	2	2.3	7-8-11	2	2.3
1-10	2	2.8	7-8-10	2	2.3
1-2-10	2	3.1	7-9-11	2	2.8
2-3	2	2.4	7-9-10	2	2.8
2-4	2	2.5	7-10-11	2	2.2
3-4	2	2.1	8-9-11	2	2.7
2-3-4	2	2.9	8-9-10	2	2.7
3-5	2	2.3	8-10-11	2	2.2
3-6	2	2.1	9-10-11	2	2.7
4-5	2	2.3	8-12	2	2.7
4-6	2	2.2	11-12	2	2.4
5-6	1	1.9	10-12	2	2.5
3-4-5	2	2.7	8-11-12	2	3.0
3-4-6	2	2.6	8-10-12	2	3.0
3-5-6	2	2.6	10-11-12	2	2.8
4-5-6	2	2.7	10-11	1	1.5

Itineraries for Instance 7x

Customers	Duration	Trans. cost	Customers	Duration	Trans. cost
1	1	14.0	10-11	1	16.0
2	1	17.9	1-11-12	1	25.6
3	1	10.6	1-10-12	1	25.4
4	1	19.2	1-10-11	1	24.4
5	1	25.8	10-11-12	1	21.5
6	1	8.9	7-10	1	14.2
7	1	9.6	6-10	1	14.6
8	2	26.6	7-12	1	18.7
9	1	25.3	6-12	1	19.5
10	1	9.5	7-11	1	16.5
11	1	12.5	6-11	1	17.3
12	1	14.6	6-7	1	13.9
4-5	2	30.6	7-10-12	1	22.1
2-4	2	28.9	6-10-12	1	22.9
3-4	1	23.3	7-10-11	1	19.9
2-3	1	21.6	6-10-11	1	20.8
2-3-4	2	32.7	6-7-10	1	18.5
1-3	1	19.4	7-11-12	1	22.1
1-2	1	23.5	6-11-12	1	22.9
1-2-3	2	27.3	6-7-12	1	23.0
1-12	1	22.0	6-7-11	1	20.8
1-11	1	21.0	7-9	2	28.7
1-10	1	18.6	7-8	2	30.1
11-12	1	18.1	8-9	2	30.8
10-12	1	17.9	7-8-9	2	34.2

APPENDIX C

MWTS ITINERARY SET

This appendix describes the itinerary set used in all the marine waste transport system scenarios in the computational work of Chapter VIII. Exhibit C.1 lists the facilities of the MWTS and their digital codes. Exhibit C.2 gives, for each itinerary, its focal facility (by facility code), the barge delivery to each marine transfer station, and the expected duration in pulses.

Exhibit C.1: MWTS Facilities

Name of facility	Type	Facility code
Gansevoort	MTS	1
West 135th Street	MTS	2
East 91st Street	MTS	3
Greenpoint	MTS	4
52nd Street	MTS	5
Southwest Brooklyn	MTS	6
South Bronx	MTS	7
North Shore Queens	MTS	8
Pier 56	SA	9
36th Street	SA	10
Army Pier	SA	11
South Bronx	SA	12
Fresh Kills Landfill	FKL	13

Exhibit C.2: MWTS Itinerary Set

Focal facility	Barge delivery to MTS								Approx. duration (pulses)
	1	2	3	4	5	6	7	8	
1	2	0	0	0	0	0	0	0	33.46
1	3	0	0	0	0	0	0	0	38.60
1	4	0	0	0	0	0	0	0	40.32
2	0	2	0	0	0	0	0	0	41.20
2	0	3	0	0	0	0	0	0	47.80
2	0	4	0	0	0	0	0	0	50.00
3	0	0	2	0	0	0	0	0	34.55
4	0	0	0	2	0	0	0	0	37.68
4	0	0	0	3	0	0	0	0	43.62
4	0	0	0	4	0	0	0	0	45.60
5	0	0	0	0	2	0	0	0	26.42
5	0	0	0	0	3	0	0	0	30.24
5	0	0	0	0	4	0	0	0	31.52
6	0	0	0	0	0	2	0	0	30.25
7	0	0	0	0	0	0	2	0	36.15
7	0	0	0	0	0	0	3	0	41.81
7	0	0	0	0	0	0	4	0	43.69
8	0	0	0	0	0	0	0	2	38.27
8	0	0	0	0	0	0	0	3	44.32
8	0	0	0	0	0	0	0	4	46.33
9	1	1	0	0	0	0	0	0	52.06
9	2	1	0	0	0	0	0	0	57.12
9	3	1	0	0	0	0	0	0	59.01
9	2	2	0	0	0	0	0	0	59.71
9	1	3	0	0	0	0	0	0	60.99

Exhibit C.2: MWTS Itinerary Set (continued)

Focal facility	Barge delivery to MTS								Approx. duration (pulses)
	1	2	3	4	5	6	7	8	
9	2	0	0	2	0	0	0	0	74.65
9	1	0	0	3	0	0	0	0	78.73
9	3	0	0	1	0	0	0	0	72.08
11	2	0	0	0	2	0	0	0	54.73
11	1	0	0	0	3	0	0	0	53.65
11	3	0	0	0	1	0	0	0	56.66
10	0	0	1	1	0	0	0	0	56.67
10	0	0	1	2	0	0	0	0	62.48
10	0	0	2	2	0	0	0	0	65.36
10	0	0	1	3	0	0	0	0	64.72
11	0	0	0	2	2	0	0	0	56.68
11	0	0	0	3	1	0	0	0	58.98
11	0	0	0	1	3	0	0	0	55.37
11	0	0	0	0	1	1	0	0	44.07
11	0	0	0	0	2	1	0	0	47.77
11	0	0	0	0	1	2	0	0	48.35
11	0	0	0	0	3	1	0	0	49.31
11	0	0	0	0	2	2	0	0	49.76
12	0	0	0	0	0	0	1	1	52.65
12	0	0	0	0	0	0	2	1	58.33
12	0	0	0	0	0	0	1	2	58.95
12	0	0	0	0	0	0	3	1	60.22
12	0	0	0	0	0	0	2	2	60.85
12	0	0	0	0	0	0	1	3	61.78

REFERENCES

1. A. Atkinson and M. Pearce: The computer generation of beta, gamma and normal random variables. *Journal of the Royal Statistical Society, Series A*, 139(4), 431-448 (1976).

2. W. Bell, L. Dalberto, M. Fisher, A. Greenfield, R. Jaikumar, P. Kedia, R. Mack, and P. Prutzman: Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces* 13(6), 4-23 (1983).

3. R. Bellman: *Dynamic Programming*. Princeton University Press, Princeton, New Jersey (1957).

4. L. Bodin and B. Golden: Classification in vehicle routing and scheduling. *Networks* 11(2), 97-108 (1981).

5. L. Bodin, B. Golden, A. Assad, and M. Ball: Routing and scheduling of vehicles and crews: the state of the art. *Computers and Operations Research* 10(2), 65-210 (1983).

6. R. Cheng: The generation of gamma variables with non-integral shape parameter. *Applied Statistics* 26(1), 71-75 (1977).

7. F. Cullen, J. Jarvis, and H. Ratliff: Set partitioning based heuristics for interactive routing. *Networks* 11(2), 125-143 (1981).

8. M. Dror: The inventory routing problem. Ph.D. thesis, University of Maryland, College Park, Maryland (1983).

9. M. Dror and M. Ball: Inventory/routing: reduction from an annual to a short period problem. Working Paper Series MS/S 85-001, College of Business and Management, University of Maryland, College Park, Maryland (1985).

10. A. Federgruen and P. Zipkin: A combined vehicle routing and inventory allocation problem. *Operations Research* 32(5), 1019-1037 (1984).

11. M. Fisher, R. Jaikumar, and W. Bell: The impact of advanced technologies on inventory management. Working Paper, Graduate School of Business Administration, Harvard University, Boston, Massachusetts (1982).

12. D. Heyman and M. Sobel: *Stochastic Models in Operations Research, Volume II: Stochastic Optimization.* McGraw-Hill Book Company, New York (1984).

13. R. Howard: *Dynamic Programming and Markov Processes.* MIT Press, Cambridge, Massachusetts (1960).

14. R. Larson: A hypercube queueing model for facility location and redistricting in urban emergency services. *Computers and Operations Research* 1(1), 67-95 (1974).

15. R. Larson and A. Minkoff: Task 1 report: establishing existing conditions. Report for the New York City Department of Sanitation barge fleet sizing project, ENFORTH Corporation, Cambridge, Massachusetts (1983).

16. A. Law and J. Carson: A sequential procedure for determining the length of a steady-state simulation. *Operations Research* 27(5), 1011-1025 (1979).

17. I. Lee and R. Larson: Computer-aided dispatch system as a decision making tool in public and private sectors. Working Paper 129-84, Operations Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts (1984).

18. M. Lembersky and U. Chi: "Decision Simulators" speed implementation and improve operations. *Interfaces* 14(4), 1-15 (1984).

19. A. Minkoff: Empirical comparison of heuristics for the deliverer dispatch problem, I: on optimally solvable problems. Unpublished manuscript (1985).

20. A. Minkoff: Computer programs for simulating deliverer dispatch models, with dispatch algorithms. Working paper, Operations Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts (1985).

21. A. Minkoff and R. Larson: *Barge Operations System Simulator*: a planning model developed for the New York City Department of Sanitation. Final Report, ENFORTH Corporation, Cambridge, Massachusetts (1983).

22. K. Murty: *Linear and Combinatorial Programming.* John Wiley & Sons, New York (1976).

23. M. Rosenblatt and M. Kaspi: A dynamic programming algorithm for joint replenishment under general order cost functions. *Management Science* 31(3), 369-373 (1985).

24. C. Schaack and R. Larson: An *N* server cutoff multi-priority queue. Working Paper OR 135-85, Operations Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts (1985).

25. P. Schweitzer and A. Seidmann: General polynomial approximation in Markovian decision processes. Working Paper Series No. QM8326, University of Rochester, Rochester, New York (1984).

26. E. Silver: A control system for coordinated inventory replenishment. *International Journal of Production Research* 12(6), 647-671 (1974)

27. C. Swain and M. Swain: A uniform random number generator that is reproducible, hardware-independent, and fast. *Journal of Chemical Information and Computer Sciences* 20, 56-58 (1980).

28. M. Syslo, N. Deo, and J. Kowalik: *Discrete Optimization Algorithms with Pascal Programs*. Prentice-Hall, Englewood Cliffs, New Jersey (1983).