STRATEGIC PLANNING FOR FIELD SERVICE SUPPORT SYSTEMS

by

Alisha A. Weathers Waller"
STRATEGIC PLANNING FOR
FIELD SERVICE SUPPORT SYSTEMS

A Dissertation
Presented to the Faculty of the Graduate School
of Cornell University
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

by
Alisha Adrian Weathers Waller
August 1991
For many industrial products, such as computers, copiers, and communication equipment, the after-sale service of the equipment is an integral part of the product life cycle. This dissertation presents models to aid in strategic planning for companies which provide after-sale service at the customer’s site. These service companies have a limited set of resources to be allocated in such a way as to provide high quality service.

From the customer’s perspective, the best measure of the quality of the service is the total downtime. We develop models to minimize the expected customer downtime, subject to a budget constraint. These models reveal interesting interactions between spare parts inventory levels and field engineer staffing levels. Focusing on customer downtime requires a detailed model of the sequence of events and activities necessary to complete a repair. This detailed model includes information about the spare parts inventory at the customer’s site and in the field engineer’s kit, the available emergency supply sources for spare parts, the policy for breaking repairs, and the dispatching policy.

The first model developed is a single echelon model which assumes a simple dispatch policy and no spare parts inventory at the customer’s site. This model does allow a repair to be interrupted if the required spare part is
not in the spare parts kit. The model is solved using dynamic programming and therefore is applicable only if there are a relatively few number of spare part types. This model is then extended to a two echelon version which incorporates the tradeoff between stocking at the depot level and stocking at the spare parts level. Finally, the two echelon model is expanded to allow a large number of part types. Our final model is tractable with large numbers of customers, field engineers, and part types. Numerical investigations illustrating the investment tradeoffs and the effect of different environments are also presented.
Biographical Sketch

Alisha Adrian Weathers was born in Aiken, South Carolina on March 12, 1965. She grew up in St. George, South Carolina and Augusta, Georgia. In March of 1987, she graduated with highest honors with a Bachelor of Industrial Engineering degree from the Georgia Institute of Technology in Atlanta, Georgia. She received a Master of Science degree from Cornell University in January of 1990. In June of 1990, she married Lance Allyn Waller. Alisha received her Ph.D. in Operations Research from Cornell University in August of 1991.
This dissertation is dedicated to Lance Allyn Waller, my husband and my best friend.
Acknowledgements

I would like to thank the following people, without whom this work would never have been completed: Lance, for believing in me every day; my daddy, David C. Weathers, for his support through the darkest days; my mom, Mary J. Weathers, and sisters, Wendy D. Weathers and Rachel L. Weathers, for their confidence in me; my sister, Stacy L. Murray, for the many e-mail updates on Scott, exercise, and life; Steve Eddins, for the long discussions about real issues; Marty Taylor, for reminding me why this work was started; Kelly Payne, for understanding my passion for teaching; Chuck and Kay Sox, for reminding me that God is in control; St. Paul's United Methodist Church, for helping me to be more than a student; Bunny and Sally's Wednesday night crew, for validating my ontology; my committee members, David Heath and William Carlsen, for their sound advice; my advisor, Peter Jackson, for his consistent support; and most importantly, my students, for giving me the motivation to continue working for my dreams.
# TABLE OF CONTENTS

## CHAPTER 1. INTRODUCTION

- General Background .............................................. 1
- Classification of Field Support Environments .............. 4
  - Customer Base Category ................................... 5
  - Field Engineer Staff Category .......................... 7
  - Inventory System Category ................................. 11
- Literature Review .................................................. 13
  - Related problems ............................................. 13
  - Evaluating Field Service Support Systems .............. 16
  - Staffing Levels .............................................. 18
  - Spare Parts Kitting Problem ................................ 19
- General Modeling Perspectives ................................ 21

## CHAPTER 2. SINGLE ECHELON MODELS

- Introduction ..................................................... 23
- Realizations of a Service Request .......................... 23
- Notation .......................................................... 27
- Assumptions ...................................................... 28
- Model Development ............................................. 30
  - Development of the Objective Function .................. 30
  - Development of the Budget Constraint .................. 32
- Model Solution .................................................. 33
  - Initial Model .................................................. 33
  - Calculation of the Kit Non-Fill Rate .................... 37
- Model Analysis .................................................. 39
  - Customer Service Time Plot ................................ 40
  - Inventory Budget Allocation Plot ......................... 44
  - Kit Composition Plot ........................................ 48
  - Comparing Rural and Urban Environments .............. 53
- Limitations of the Single Echelon Model .................. 54

## CHAPTER 3. TWO ECHELON MODEL

- Introduction to Two Echelon Systems ........................ 55
- Realizations of a Service Request .......................... 56
- Notation .......................................................... 58
3.4. Assumptions ........................................ 58
3.5. Model Development ............................... 59
   3.5.1. Development of the Objective Function ...... 59
   3.5.2. Development of the Budget Constraint ...... 60
3.6. Solution Method .................................. 60
3.7. Model Analysis ................................... 69
   3.7.1. Customer Service Time Plot ................ 69
   3.7.2. Inventory Budget Allocation Plot .......... 73
   3.7.3. Kit Composition Plot ....................... 76
   3.7.4. Depot Composition Plot ..................... 80

CHAPTER 4. EXTENSION TO A LARGE NUMBER OF PARTS ........................................ 85
4.1. Introduction ....................................... 85
4.2. Part Grouping .................................... 85
4.3. Model Development ............................... 86
4.4. Output Analysis of General Model Behavior .... 89
   4.4.1. Increasing the Budget Increases the Optimal
          Number of Field Engineers .................... 89
   4.4.2. Inventory Levels Vary as Budget Increases .. 92
   4.4.3. Urban and Rural Environments Have Different
          Optimal Solutions ............................. 95
   4.4.4. Different Part Groupings Yield Different
          Optimal Solutions .............................. 99
   4.4.5. Duality Gaps May Cause Strange Behavior ... 103
4.5. Strategic Planning Questions Addressed by the Model . 107
   4.5.1. Small Budget Increases May Yield Large Benefits 107
   4.5.2. Effect of the Breaking Repairs Policy Parameter . 109
   4.5.3. Effect of the Emergency and Quick Delivery Costs in Urban Environments .. 109

CHAPTER 5. ADDITIONAL MODELS ................. 113
5.1. No Emergency Supply ............................ 113
5.2. Job Completion Rate ............................. 116

CHAPTER 6. DISSERTATION SUMMARY AND AREAS FOR FURTHER RESEARCH ............. 122
6.1. Dissertation Summary ............................ 122
6.2. Areas For Further Research .................... 124
LIST OF FIGURES

Figure 1: Realizations of a Service Request – Single Echelon System ........................................ 24
Figure 2: Customer Service Time Plot for K1–R1: A Rural Environment ................................. 42
Figure 3: Customer Service Time Plot for K1–U1: An Urban Environment .................................. 43
Figure 4: Inventory Budget Allocation Plot for K1–R1 ................................................................. 46
Figure 5: Inventory Budget Allocation Plot for K1–U1 ................................................................. 47
Figure 6: Kit Composition Plot for K1–R1 ................................................................................. 51
Figure 7: Kit Composition Plot for K1–U1 ................................................................................. 52
Figure 8: Realizations of a Service Request – Two Echelon Systems ........................................... 57
Figure 9: Two Echelon Model (M4) ......................................................................................... 61
Figure 10: Model (M5) ........................................................................................................... 64
Figure 11: Customer Service Time Plot for K2–U1 ................................................................. 71
Figure 12: Customer Service Time Plot for K2–R1 ................................................................. 72
Figure 13: Inventory Budget Allocation Plot for K2–U1 ........................................................... 74
Figure 14: Inventory Budget Allocation Plot for K2–R1 ........................................................... 75
Figure 15: Kit Composition Plot for K2–U1 ............................................................................. 78
Figure 16: Kit Composition Plot for K2–R1 ............................................................................. 79
Figure 17: Depot Composition Plot for K2–U1 ....................................................................... 83
Figure 18: Depot Composition Plot for K2–R1 ....................................................................... 84
Figure 19: Controllable Time versus Total Cost for K3–U1 ......................................................... 90
Figure 20: Optimal Number of Field Engineers versus Total Cost for K3–U1 ............................. 91
Figure 21: Kit Coverage and Depot Coverage versus Total Cost for K3–U1 .............................. 94
Figure 22: Optimal Number of Field Engineers for K4–U2 and K4–R2 ..................................... 96
Figure 23: Kit Coverage versus Total Cost for K4–U2 and K4–R2 ............................................. 97
Figure 24: Depot Coverage versus Total Cost for K4–U2 and K4–R2 ....................................... 98
Figure 25: Optimal Number of Field Engineers versus Total Cost for K4–U2 and K5–U2 ........ 100
Figure 26: Optimal Kit Coverage versus Total Cost for K4–U2 and K5–U2 .......................... 101
Figure 27: Optimal Depot Coverage versus Total Cost for K4–U2 and K5–U2 ............................................ 102
Figure 28: Controllable Time Plot for K6–U1 .................. 104
Figure 29: Optimal Number of Field Engineers for K6–U1 . 105
Figure 30: Kit Coverage and Depot Coverage for K6–U1 .. 106
Figure 31: Controllable Time Plot for K3–U1 ................. 108
Figure 32: Controllable Time Plot for Various Delivery Cost Parameters .................................................. 112
Figure 33: Benefit / Cost Ratio Model – Implicit Enumeration Example ....................................................... 120
# LIST OF TABLES

Table 1. Part Profile for Kit Type 1 .......................... 40  
Table 2. Fill Rates for K1–R1 and K1–U1 ..................... 45  
Table 3. Kit Composition for K1–R1 : Rural Environment . 49  
Table 4. Kit Composition for K1–U1 : Urban Environment 50  
Table 5. Field Engineer Utilization ............................ 70  
Table 6. Fill Rates ............................................... 73  
Table 7. Kit Composition for K2–U1 : Urban Environment 77  
Table 8. Kit Composition for K2–R1 : Rural Environment . 77  
Table 9. Depot Composition for K2–U1 :  
    Urban Environment ........................................... 81  
Table 10. Depot Composition for K2–R1 :  
    Rural Environment ........................................... 82  
Table 11. Part Delivery Times and Costs ....................... 110  
Table 12. Kit Composition for Benefit / Cost Model ......... 119  
Table 13. Benefit / Cost Ratios for Implicit Enumeration  
    Example ....................................................... 121
CHAPTER 1. INTRODUCTION

1.1. General Background

The rapid rise in the technological complexity of manufacturing environments during the past several decades has shifted the burden of equipment repair from the user to the original manufacturer. No longer is a general "handy man" a sufficient maintenance and repair crew even for a small manufacturer. Furthermore, the cost of training a crew to repair all of the diverse pieces of equipment found in a modern factory is prohibitive. As a result, equipment manufacturers are receiving increased pressure to provide repair services as part of the lease agreement, as part of the sale contract, or as a distinct, marketable service. There are some companies that exist solely to provide this service and have no manufacturing capacity.

In general, manufacturing equipment is too large, too heavy, and too expensive to ship to a repair facility. Consequently, the original manufacturer or a service agency must provide this maintenance and repair at the permanent location of the equipment. This on-site repair is called field service. The repair technician, or field engineer, must travel to the customer's location to perform the field service. Spare parts needed in the repair must either be pre-positioned at the customer site or brought to the customer site at the time of the repair, typically by the field engineer.

In addition to the pressure from customers, equipment manufacturers also are pressured by competitors who have realized the impact which quality field service has on market share. Blumberg [5] has shown that
after-sale service is one of the most important criteria by which companies choose among competing equipment manufacturers. He suggests that improving field service can increase market share significantly. Blumberg concludes that among the most important field service improvements are reductions in response time and reductions in the total customer downtime.

There are several key features to field support that drive the economics of such systems. The first feature is that demand for the service, including the demand for the field engineer’s time and the demand for spare parts, is random. Furthermore, the extent of the repair time and the specific spare parts required often cannot be determined until the field engineer actually visits the customer site and diagnoses the repair problem. The second feature is that the essential performance measure is time-based: how long a customer must wait, on average, until the repair is completed. Acceptable levels of this performance measure, average customer downtime, are far shorter than the manufacturing lead time for spare parts. This forces the system to carry large safety stocks of spare parts. Tight targets for average customer downtime can also force the system to have a large staff of underutilized field engineers.

The essential problem of field support is to provide acceptable customer service with the most economical use of resources (inventory, staff, and transportation). There are many opportunities to optimize the use of resources in such systems and many tradeoffs to explore. For example, a request for service can sometimes be diagnosed over the telephone at the
time of the request and the repair effected by the customer. Such a telephone diagnosis saves the non–value–added cost of the field engineer’s travel and achieves a short customer downtime. Hence, the substitution of telephone diagnosticians for field engineers can be advantageous. However, the substitution has diminishing returns, because some service requests must be resolved with a field engineer visit and extensive telephone diagnosis can prove fruitless.

Another substitution that can be effective is to substitute inventory for field engineers. Time is wasted if a field engineer discovers that a spare part is needed to complete a repair and the spare part is not immediately available. The field engineer must either wait, idle, until the part is delivered or leave the customer and return when the part is delivered. In either case, the field engineer’s time is not being effectively utilized. Inventory, in the form of spare parts stockpiled at the customer site or in the form of a spare parts kit carried by the field engineer, can increase the frequency of immediate repairs. This has the effect of both cutting the customer downtime and increasing the field engineer utilization in value added activities. There are diminishing returns to this substitution as well. The focus of this dissertation is on understanding this tradeoff between field engineers and spare parts inventory.

The rest of this chapter is organized as follows. Section 1.2 describes the salient features of field service support systems. Section 1.3 reviews the current literature. Section 1.4 discusses the fundamental approach to
the modeling. Then Chapter 2 develops the single echelon models while
Chapter 3 develops two echelon models. Chapter 4 extends these models to
accommodate a realistic number of part types. Chapter 5 discusses
observations from the model results and Chapter 6 gives a summary and
directions for further research.

1.2. Classification of Field Support Environments

We need to recognize the diversity of environments in which field
service occurs. One way to recognize this diversity is to develop a
classification scheme based on salient criteria. This scheme will allow us to
identify instances of field service that are similar enough to be analyzed
with the same type of model, and to identify instances that are different
enough as to require different models or assumptions. For example,
servicing mainframe computers has much in common with servicing
complex photocopying equipment. However, a John Deere field engineer
servicing tractors and combines in Kansas is certainly in a different
environment than a Xerox field engineer in the John Hancock building in
Chicago. Focusing on a particular environment will allow the development
of more detailed and accurate models. In addition, using this classification
scheme will enable field service managers from different companies to
quickly understand the basic properties of other environments.

The attributes of this classification scheme can be grouped into three
categories: customer base, field engineer staff, and inventory system. The
attributes of each category are described below.
1.2.1. Customer Base Category

This group of attributes describes the customer base characteristics and the company policies that directly affect the customer.

A. Topography of Customer Base

The topography of the customer base refers to the geographical dispersion of the customers. The topography can be described qualitatively as urban, suburban, or rural. An urban customer base is one in which the customers are spread uniformly across the region and in which the average travel time between any two customers is small relative to the average repair time. As the name suggests, this topography occurs most often in large cities or industrial centers. For example, an IBM field engineer in Manhattan works in an urban environment. A suburban customer base is one in which the customers are grouped in clusters. The travel time between clusters is large relative to the repair time while the travel time within a cluster is relatively small. For example, a Maytag repairperson in south Georgia works in a suburban environment. Finally, a rural customer base is one in which the customers are spread uniformly across the region, but the travel time between any two customers is relatively large. For example, a John Deere field engineer in Kansas would probably service a rural customer base.

B. Service Call Types

In field service support systems, the customer usually initiates the request for a field engineer visit. Such service calls are called request calls.
However, in some systems, the company providing the service may initiate a visit in order to do a physical inventory count, to do periodic inspection and maintenance, or to simply keep the customer’s good will. These visits, called voluntary visits, are usually initiated when a field engineer is already near a customer’s geographic location. Voluntary calls may be especially important in rural or suburban environments because the field engineer may be able to prevent a future request call and thereby save the return travel to the customer.

C. Service Requirements

Two quantities are important in describing the service requirements: equipment reliability and equipment repairability. Equipment reliability of a particular machine is measured by the mean time between a customer’s request calls for service on that machine. It is related to, but not necessarily equal to, the equipment’s mean time to fail. If the customer attempts to repair the equipment alone, before calling for field support, then the mean time to call is longer than the mean time to fail. Also, the reliability factor given in the equipment specifications is the mean time to fail under continuous use. Since the customer may not use the equipment continuously, the calendar time between calls for service may be several orders of magnitude above the specified mean time to fail. Equipment repairability is measured by the mean time to repair, which is affected by the type of repair, the field engineer’s training level, the degree of modularity of machine design, and so on.
D. Priority Customer Classes

Some field service companies offer a variety of support contracts. These contracts may specify the priority class to which the customer is assigned. Typically, higher priority contracts cost the customer more. The high priority designation indicates that the customers will not be served in a strictly FIFO (first-come first-served) manner, but instead will be served in order of their priority classes and in FIFO order within a priority class. Some companies offer a guaranteed response time as well, with different maximum response times for different priority classes. Offering priority customer classes may increase the market share of the service company, but it also increases the complexity of the dispatch policy and the difficulty of planning the workforce. A priority customer policy will also increase the response time to lower priority customers.

1.2.2. Field Engineer Staff Category

The criteria in this category are those which describe the field engineers and their interactions with each other.

A. Homogeneity of Field Engineers

The homogeneity of the field engineers refers to the degree of specialization of the field engineers’ skills. At one end of the spectrum, each class of field engineers is specialized to repair a particular set of equipment and these sets are disjoint. At the other end, each field engineer has the same level of expertise for each piece of equipment as every other field engineer. Most companies operate somewhere in the middle, with the
field engineers cross-trained for several product lines, but emphasizing one line in particular. In a real field support system, other factors also contribute to the heterogeneity of the field engineers, including experience, intuition, and customer relations. However, these factors are very hard to quantify.

B. Travel Patterns

The travel pattern of the field engineer describes the consistent characteristics of the field engineer’s travel. For example, an "out-and-back" policy is one in which the field engineer must return to a given location, usually an inventory supply point, after each service call. This travel pattern may be used when the inventory items are too large or too valuable to carry multiple parts at one time or when the field engineer can service at most one customer per day. Another travel pattern is the "dispatch" pattern, which is characterized by the field engineer calling the dispatching center after completing each repair to find out which customer to visit next. A "deterministic" travel pattern refers to the situation in which all of the field engineer’s visits for one period of time, say a day or a week, are planned in advance. Deterministic travel patterns are advantageous because they permit optimization of travel time. However, to implement such patterns, a large portion of the service calls must be voluntary calls or low priority calls.
C. Dispatch Policy

The dispatch policy is the set of rules for assigning to each field engineer the next call to service. One example is the "nearest call" policy, in which when the field engineer completes a service call, the next call assigned to him or her is the one nearest geographically. This policy minimizes the travel time between each call, but is not guaranteed to minimize the total travel time. Another common example is the "first-in-first-out" (FIFO) policy. This policy handles the request calls strictly in the order in which they are placed. If the company has multiple priority classes of customers, then the FIFO policy may be adapted to the "FIFO with priorities" policy. This policy first handles all of the highest priority calls in a FIFO manner, next handles all of the second highest priority calls in a FIFO manner, and so on. Therefore, the lowest priority calls are serviced only if there are no other calls in the queue. A third type of dispatch policy assigns the calls in order of the severity of the customer's situation. For example, a customer whose manufacturing plant must be completely shut down due to the needed service would be handled before a customer who had a spare machine or alternative equipment. A robust dispatch policy is usually a combination of the above policies.

D. Field Engineer Pooling Level

The field engineer pooling level describes the mapping of customers to field engineers. The relationship between a field engineer and his or her customers is an important one. The customer's trust of the field engineer is
vital since the repair must be done at the customer’s site. The access to the customer site gives the field engineer an opportunity to see many facets of the customer’s business. In many cases, the field engineer has the opportunity to observe confidential processes of the customer. To protect these industrial secrets, the customer may insist that only one field engineer from a particular equipment manufacturer have access to an equipment site. This situation, in which only one field engineer is assigned to each customer, has a zero pooling level. On the other hand, if several field engineers are jointly responsible for the customer, there is a positive pooling level. The pooled allocation generally allows better utilization of the field engineering capacity since the probability of having an available field engineer is higher with a positive pooling level. Balanced against the efficiency of a positive pooling level are the intangible benefits of the close relationship that can develop between the customer and a field engineer dedicated to that customer. For example, such dedicated field engineers often alert the equipment sales representatives of the customer needs which they have noticed.

E. Territory Boundaries

Most field service companies divide the customer base into a set of disjoint territories which are serviced by different teams of field engineers. A territory boundary is designated as a "soft" boundary if a team of field engineers is allowed to cross the boundary to provide extra manpower to another territory which has an unusually large queue of request calls. On
the other hand, a "hard" boundary does not allow these temporary adjustments in manpower. For example, after hurricane Hugo hit the South Carolina coast in 1989, many insurance companies temporarily reassigned additional insurance adjustors to the area. When the majority of the work was completed, these adjustors returned to their original territories. For many companies, especially disaster relief companies, the manpower flexibility given by soft boundaries is a crucial part of the strategic planning. Very few companies truly operate with hard boundaries, but this criterion is useful in classifying the detail level of field service planning models.

1.2.3. Inventory System Category

A. Spare Parts Description

The general nature of the spare parts has many implications for the field service support system. Important characteristics include size, weight, cost, shape, and perishability. These characteristics may necessitate additional constraints on the spare parts kit composition. Other characteristics which affect the handling of the parts, such as flammability and toxicity, must also be included in the description.

B. Inventory Supply Structure

The inventory supply structure includes all inventory storage locations and the supply routes among them. In practice, there may be thousands of locations and hundreds of thousands of supply routes. However, in order to develop a tractable planning model, the real structure
must be approximated by a simple structure. For example, we examine models that include only two stocking locations: the spare parts kit of the field engineer and the depot that supplies the field engineer.

C. Replenishment Policies

The replenishment policies specify the normal restocking of each type of spare part at each of the inventory storage locations. For example, a local depot may use a continuous review \((s,S)\) policy. In this policy, the current inventory level is continuously monitored. When the current level falls below \(s\), an order is placed to bring the on-hand plus on-order level up to \(S\). A common variant of the continuous review is the \((s-1,s)\) policy, or one-for-one replenishment. Under this policy, a replenishment order is issued as soon as a demand for the part occurs. A description of the replenishment policy should include the review policy, the costs involved, and the lead time to deliver.

D. Emergency Supply Sources

In the event that a stocking location has no stock on hand when a demand occurs, the location may request stock from an emergency source at additional cost. For example, a field engineer may have the local depot deliver directly to the customer a needed part which is not in his or her kit. A local depot may place an emergency order with the central warehouse. For each stocking location and each spare part type, the emergency supply policy must specify the possible sources of emergency supply, the cost of
supplying one unit of the part from the source, and the lead time to deliver the part to the location.

The classification scheme described above is a framework for understanding diverse field support environments. It also provides a framework and terminology for contrasting the mathematical models in the research literature related to field service.

1.3. Literature Review

Although strategic planning for field service support systems is an important issue for many manufacturers, there are surprisingly few models in the research literature on this topic. The models which have been published fall into three categories: models to evaluate a particular field service support system, models to optimize the field engineer staff size, and models to evaluate inventory options. In addition to these three categories, field service support systems are strongly related to two other areas in the research literature: emergency service systems and the machine interference problem.

1.3.1. Related problems

Emergency Service

Field service support shares some characteristics with emergency service support such as fire fighting systems and ambulance support systems. In all three of these systems, the requests for service occur randomly. Also, each system has a large customer base per trained professional. On the surface, it may seem that these systems could be
analyzed with the same model. However, these systems have some major differences also. First, since human life is often at stake, emergency system planning considers the response time to be its primary objective. Field service systems, however, are primarily concerned with the total response time, that is, the time from when the request is made until when the repair is completed. Second, for the emergency systems, the potential of one request for service causing many requests, (e.g. a fire spreads to neighboring houses) and the potential loss of property and human life causes emergency system planners to accept low utilization rates for the system. For example, a fire station may spend over fifty percent of the time with the truck parked inside. However, a field service manager could not afford to have the field engineers idle that often. Also, the demands for emergency systems have seasonal trends, with large demands typically occurring during holiday seasons. In addition, spare parts are not an issue with typical emergency service, but are essential in field support. Finally, emergency service systems are public services. Therefore, strategic decisions such as fire station location are subject to a political process, involving the general public. The operation of fire fighting systems are particularly vulnerable to public outcry if a decision is made to close stations for financial reasons. On the other hand, the customers of field service systems are generally not as aware of the operational decisions made by the field service managers. These differences are some of the reasons why field service support systems cannot be analyzed with models developed for emergency support systems.
For an extensive review of emergency systems literature see Kolesar and Swersey [14].

Machine Interference Problem

The classic machine interference problem, also called the machine repairman problem, arises when a single repairman is assigned to service a collection of machines. If a machine fails and the repairman is busy working on another machine, then the machine must wait for service. Thus, each machine is always in one of three states: working, idle being repaired, or idle waiting for repair. The classic machine interference problem is to determine how many machines to assign to one repairman, assuming homogeneous, independent machines. The most common modeling approach is Markov chain analysis with the evaluation criterion being the average number of machines working. For a review of the classic problem and extensions, see Stecke and Aronson [22].

This problem is similar to a field support system with an urban environment and a staff size of one. If the travel time, compared to the repair time, is insignificant and the number of customers is small, then the field support system can be analyzed as a machine interference problem. Most field service support systems have significant travel times or a large staff size; therefore, the machine interference literature is insufficient for general field service support systems.

Repairable Spare Parts Planning

Another area of the research literature which is related to field
service planning is the repairable spare parts planning literature. Nahmias [17] gives a comprehensive review of this area. The basic repairable spare parts system has a depot echelon and a retail echelon. When parts fail at the retail level, they are sent to the depot to be repaired. If the depot has any repaired units in stock, a replacement unit is sent to the retail echelon. The decision variables for this problem are the total numbers of units of each part type to have in the system.

1.3.2. Evaluating Field Service Support Systems

The papers in this group assume the field service support system is completely specified and take as their goal the evaluation of the effectiveness of the system. In these models, the spare parts inventory is not explicitly considered; instead, the necessary parts are assumed to be stocked at the customer’s location.

Agnihothri [1 & 2] and Agnihothri and Karmarkar [4] develop models to evaluate performance characteristics of given field service systems. They assume a small number of customers, a single server, and a small number of customers. In [2], Agnihothri shows that the M/G_R/G_T/1//N model, where G_R and G_T represent general repair time and travel time distributions, can be reduced to a machine interference problem with the first machine to be repaired in each busy period having a different repair time distribution. The different repair time is a result of the assumption that an idle repairman waits at the last repair location until another request arrives. With this assumption there is a positive probability
that the request will come from the current location, hence a travel time may not be needed. Therefore, the first service of a busy period may or may not require a travel time, but each successive service of the same busy period will require a travel time.

In [4], Agnihothri and Karmarkar develop a travel time distribution model. They consider each machine location as a node on a complete network with the travel time between locations as the length of the connecting arc. Then they develop an approximation for the long run proportion of times the repairman travels between each pair of nodes and therefore, an overall travel time distribution. This approximation is used in their other papers to model the travel time distribution.

In [3], Agnihothri considers the machine interference problem with non-identical machines and multiple repairmen. He develops various performance measures for the aggregate system and establishes their interrelationships.

Agnihothri’s work assumes that the field engineers are each completely dedicated to a small number of customers. This prevents the application of his models to environments which have pooled field engineers or large numbers of customers.

Another interesting paper is by Reynolds [18]. Instead of assuming a first-in-first-out service discipline, he assumes that after completing a repair, a field engineer chooses the nearest customer in need of a repair. Reynolds models service time as being exponentially distributed with a
service rate dependent on the number of customers waiting to be served, the expected repair time, and the expected travel time. He measures system performance by the steady state number of operating customers. This model, like Agnihothri’s models, is difficult to apply if the number of customers is large. It also assumes that every repair is completed on the first visit.

1.3.3. Staffing Levels

In his 1982 paper, Hambleton [12] addresses the question of setting the appropriate field engineer staff size. He assumes the customers are spread uniformly across the region (e.g. an urban environment), and calculates an average travel time based on the area of the region. Hambleton uses initial response time as his criteria and assumes that all calls are completed on the first visit. He finds the optimal staff size by equating the expected total hours work with the total labor hours provided by N field engineers, then solving for N. His model considers random travel, two priority classes of customers, number of machines per area, number of calls per area, and field engineer utilization.

Smith [20] also considers the issue of setting staffing levels for field service systems. He assumes one field engineer is completely responsible for all calls in the region and a rural customer base with FIFO dispatching. Given these assumptions, Smith develops a model to calculate the area per server as a function of the call arrival rate, the desired average response time, and the service time parameters. The service time parameters
incorporate the sequential dependence of the travel time by means of an approximation. Smith then extends the model to consider multiple priority customer classes.

Neither Hambleton nor Smith consider the spare parts inventory investment. Therefore, they also fail to consider the possibility that a repair may be broken; hence requiring two travel times.

1.3.4. Spare Parts Kitting Problem

In their 1982 paper, Mamer and Smith [15] develop a spare parts kitting model. They assume the kit will be restocked between each service call, but do not require independence between demands for various part types for one job. They identify each different job type which is possible. Then, to each of these job types they assign a cost of breaking the job due to part unavailability and assign an arrival rate of requests. Next, they form a min cut - max flow network in which the minimum cut is the sum of the kit holding cost and the expected broken job penalty. Although this model may be solved exactly, for many field service support systems, identifying all possible job types would be a formidable task. In their 1985 paper, Mamer and Smith [16] consider a more general spare parts kitting situation in which spare machines are also stocked to be used to satisfy demands which can not be met with the spare parts kit.

Smith, Chambers, and Shlifer [21] also address the spare parts kitting problem by assuming kits are restocked between jobs and a penalty cost is incurred for each broken job. In addition, they assume independent
part failures. Their model chooses the subset of parts which minimizes the sum of the holding costs and the penalty costs. Under their assumptions, they show that the set of policies which need to be considered is well-defined and has cardinality equal to the number of part types plus one.

Graves [8] considers the same problem and assumptions as Smith, Chambers, and Shlifer, but he is unwilling to assign a penalty cost for a broken repair. Instead, he minimizes the holding cost subject to a job completion criterion, using a binary knapsack model. This model has the advantage that it can be modified easily to include additional constraints, such as limited space or weight. However, there is no polynomial time solution procedure, although many good heuristics are available.

Hausman [13] comments on the dominance of Graves approach over Smith, Chambers, and Shlifer’s approach. He then introduces the idea of a mixed strategy in which field engineers do not carry identical kits.

In all of the above papers, the interaction between the field engineers and the inventory was ignored. The only paper which we found that does consider these interactions is Graves’ 1988 paper [9]. He considers a repair depot that repairs failed units by replacing components and also repairs the failed components to return them to stock. He develops a discrete-time, linear-systems model that characterizes the trade-off between labor requirements and inventory requirements.
1.4. General Modeling Perspectives

As summarized above, the current literature dealing with field service support system models specific parts of the system. These models miss considering important interactions among system components. For example, the number of field engineers assigned to a territory and the spare parts kits affect the level of customer service which the system is able to provide. Since the field support company pays for both of these components, the decision of each affects the other component.

To capture these interactions, we will take a corporate view that there is a fixed budget to provide all of the resources required by the field support system. The operating costs of providing the field engineers as well as the inventory costs must be considered in the budget. In addition, since the company has the option of not providing this service, the working capital needed to finance the system must also be considered. Hence, the term "budget" in this dissertation refers to the money available to pay operating costs and to finance the inventory.

Many different criteria can be used to judge the effectiveness of a field service support system, including response time, percentage of calls completed with one visit, field engineer utilization, and total customer service time. Based on conversations with field engineers, their customers, and field engineer managers, we have decided to evaluate the systems by total customer service time, that is, the time from the customer’s initial request for service to the completion of the service. This criterion is
documented by Blumberg [ 5 ] as an important aspect which customers use to choose among competing field service companies.
CHAPTER 2. SINGLE ECHELON MODELS

2.1. Introduction

In this chapter we consider single echelon field support systems. In these systems, each field engineer carries a kit of spare parts for use in the repairs. If the repair requires a part that is not in the kit, then it is ordered from the central warehouse for emergency delivery. The normal replenishment of the kit is also supported by the central warehouse, which is assumed to have unlimited supply. There are costs and part delivery times associated with both the normal kit replenishment and the emergency supply.

2.2. Realizations of a Service Request

In order to understand the expected total customer service time, let us first examine the possible realizations of a service request. These realizations are depicted in Figure 1. Horizontal lines in the figure represent activities whose duration is measured by the associated variable. Diagonal lines represent probabilistic outcomes whose conditional probability is measured by the associated variable. The first activity initiated by a customer’s call is a phone diagnosis by the hot-line attendant with duration $D_1$. This phone diagnosis may be as simple as asking questions such as “Is the machine plugged into the wall socket securely?” or it may be as complex as guiding the customer through a series of tests and actions based on an expert system using a detailed knowledge base of the machine. Regardless of its complexity, the phone diagnosis may be
REALIZATIONS OF A SERVICE REQUEST

$p_1 = \text{Prob\{phone diagnosis not sufficient\}}$

$p_2 = \text{Prob\{repair needs a part\}}$

$p_3(K) = \text{Prob\{needed part is not in kit\}}$

$p_4 = \text{Prob\{repair is broken when needed part is not in kit\}}$

---

Figure 1: Realizations of a Service Request – Single Echelon System
sufficient to resolve the customer’s problem, which would complete the service, denoted on the figure by outcome A. Let $p_1$ be the probability that phone diagnosis is not sufficient.

If the phone diagnosis is not sufficient, then the customer’s request is added to a queue of requests and the customer must wait for a field engineer to become available. The time spent in the field engineer request queue is denoted by $Q_L$. After a field engineer is assigned to this call, a further delay occurs as the field engineer travels to the customer site (T).

Upon arrival, the field engineer begins the on-site diagnosis ($D_2$). The repair may or may not include replacing a failed part with a spare part. With probability $p_2$ the field engineer finds that a replacement part is required. With probability $(1-p_2)$, the field engineer finds that no part is needed and only machine adjustments are required ($R_A$), denoted in the figure by outcome B. If a spare part is needed, the field engineer may obtain the part from his or her spare parts kit or the local depot. If the part is in the kit, then the repair may start immediately ($R_P$), indicated in the figure by outcome C.

When a part is required, let $p_3(K)$ be the probability that the needed part is not in the kit, where K is a vector describing the contents of the spare parts kit. If the part is not in the kit, it is ordered from the central warehouse for emergency delivery and the field engineer must decide whether or not to “break” the repair. A “broken” repair means that the field engineer will leave the current customer to respond to another
customer’s call. If the repair is not broken, then the field engineer waits at the customer site for the emergency delivery ($\tau_{EW}$) and then completes the repair using this part ($R_P$). The field engineer may use the emergency part delivery time to conduct preventive maintenance, employee training, or customer relations. In other circumstances, the emergency delivery time may consist of the time required for the field engineer to return to the central warehouse, get the part, and return to the customer. A repair which is not broken is indicated by outcome D in the figure.

If the repair is broken, then the customer must wait for the part to be delivered ($\tau_{EB}$). If the field service organization has several delivery mechanisms, such as a taxi or their own delivery van, the emergency delivery time when the field engineer is waiting may be different from the emergency delivery time when the repair is broken. For example, it may be advantageous to combine several emergency deliveries to the same area when the repairs are broken. After the part has been delivered to the customer, the customer is put into the revisit queue to wait for a field engineer to become available to complete the repair. The revisit queue may be assigned a higher priority than the first visit queue, so we may have $Q_H < Q_L$. Once a field engineer has been assigned to revisit the customer, the customer experiences the travel time delay ($T$), and the repair time delay ($R_P$) before the service is finally performed. The repair time is assumed to be the same whether or not the repair is broken. A broken repair is indicated in the figure by outcome E.
Outcomes A through E represent the possible service request realizations in a single echelon system. In Chapter 3, we consider an expanded set of realizations for a two echelon stocking system.

2.3. Notation

In this section, we define the notation which is used in this and subsequent chapters. The quantities fall into three groups: model parameters, decision variables, and calculated values.

Model Parameters:

\( \lambda \) : the expected arrival rate into the system of requests for service;

\( D_1 \) : the expected phone diagnosis time;

\( T \) : the expected travel time between customers;

\( D_2 \) : the expected on-site diagnosis time;

\( R_A \) : the expected adjustment repair time when no part is required;

\( R_P \) : the expected repair time when a part is required;

\( \tau_{EB} \) : the expected emergency part delivery time for a broken repair;

\( \tau_{EW} \) : the expected emergency part delivery time when the FE waits;

\( p_1 \) : the probability that phone diagnosis is not sufficient;

\( p_2 \) : the probability that some part will be needed for the repair;

\( p_4 \) : the probability of breaking the repair if the part is not in kit;

\( c_{FE} \) : the annual cost of one field engineer;

\( c_{N1} \) : the normal replenishment cost for one unit of a part;

\( c_E \) : the emergency delivery cost for one unit of a part;

\( N \) : the number of different part types;
\( G_i \): the probability that part \( i \) is the needed part; and
\( H_{i1} \): the holding cost per unit year at the kit level for part \( i \).

Decision Variables:

\( X \): the number of field engineers;
\( K_i \): the number of units of part \( i \) carried in the kit, \( i = 1, \ldots, N \); and
\( K \): ( \( K_i \)).

Calculated Values:

\( Q_L \): the expected initial wait time in queue for a FE to be assigned;
\( Q_H \): the expected wait time in the revisit queue for a FE to be assigned; and
\( p_3(K) \): the probability that the needed part is not in the kit, given the kit composition \( K \).

2.4. Assumptions

We make the following assumptions for all models developed in this dissertation:

(A1) The field engineers are homogeneous in skill level and carry identical kits.

(A2) The field engineers are completely pooled; that is, each field engineer is equally likely to serve a customer request.

(A3) The spare parts kits are not pooled; that is, if a field engineer does not have a particular part in his or her kit, he or she may not try to obtain the part from another field engineer.

(A4) The dispatch policy is first-in-first-out (FIFO) within priority classes.
The revisit requests have higher priority than the first visit requests.

(A5) Travel time (T) between any two customers is constant and independent of the sequence of customers visited.

(A6) Repair time (R_p) is constant and equal for all parts.

(A7) The calls arrive to the dispatch system according to a Poisson process with rate \( \lambda \).

(A8) The part delivery times (\( \tau_{EB} \) and \( \tau_{EW} \)) are constant and equal for all parts.

Note that A1 and A4 imply that a customer is equally likely to be served by any of the field engineers. Therefore, the initial request call arrival rate that a particular field engineer experiences is \( \lambda p_1 / X \).

Assumption A5 is certainly not true if we examine the system over time, but it is appropriate for our goal of developing a planning tool based on average performance.

Assumptions A6 and A8 could be relaxed by allowing the repair times and delivery times to depend on the part type. This relaxation would only increase the notational complexity of the model without revealing additional insight.

The conditional probability, \( p_3(K) \), requires a further detailed model to develop its relationship to the model input parameters. We defer presentation of this model to section 2.6.2.
2.5. Model Development

2.5.1. Development of the Objective Function

As discussed in Chapter 1, the objective of our model is to maximize the customer service as measured by the expected total customer service time, that is, the time from when the customer first calls the dispatching system until the repair is completed. This total service time can be calculated by taking a weighted average of the expected customer service times conditioned on the service call outcomes, A through E, as depicted in Figure 1. Let C.S.T. denote the customer service time.

\[
E[\text{Customer Service Time}] = E[C.S.T(X, K)]
\]

(1) 

\[
= D_1 + p_1(Q_L + T + D_2) + p_1(1-p_2)R_A + p_1 p_2 R_P
\]

\[
+ p_1 p_2 p_3(K) (p_4(\tau_{EB} + Q_H + T) + (1-p_4)\tau_{EW})
\]

To estimate \( Q_L \) and \( Q_H \), the expected queue time for an initial visit and a return visit, respectively, we model the system of field engineers as an M/M/X queue with two priority classes \([10]\). The number of servers in the queueing system is the number of field engineers, \( X \). Let \( \lambda_L \) denote the expected arrival rate of low priority customers (those requiring initial visits) and let \( \lambda_H \) denote the expected arrival rate of high priority customers (those requiring return visits). By assumption, the arrivals of these classes are thinned Poisson processes of the arrivals of service requests.
Therefore,

\begin{align*}
(2) \quad \lambda_L &= \lambda \cdot p_1 \quad \text{and} \\
(3) \quad \lambda_H &= \lambda \cdot p_1 \cdot p_2 \cdot p_3(K) \cdot p_4 .
\end{align*}

Note that the high priority arrival rate is dependent on the spare parts kit composition.

In order for the mathematical analysis of this queueing model to be tractable, we must have a common service rate for both priority classes. The mean service rate, \( \mu \), is given by the inverse of the expected time a field engineer spends servicing a request:

\begin{align*}
(4) \quad \mu &= \frac{1}{\{p_1(T + D_2) + p_1(1-p_2)R_A + p_1p_2R_P + p_1p_2p_3(K)(p_4T + (1-p_4)\tau_{EW})\}} .
\end{align*}

By Gross and Harris [10], we can compute the expected time that a customer spends in each queue by the following equations:

\begin{align*}
(5) \quad Q_L(X, \lambda_L, \lambda_H, \mu) &= \left[ X! \left( 1 - \frac{\lambda_L + \lambda_H}{X\mu} \right) \sum_{n=0}^{X-1} \left( \frac{\lambda_L + \lambda_H}{\mu} \right)^{n-X} \frac{1}{n!} + X\mu \right]^{-1} \\
&\quad \times \left\{ 1 - \frac{\lambda_H}{X\mu} \right\} \left[ 1 - \frac{\lambda_L + \lambda_H}{X\mu} \right] ;
\end{align*}

and

\begin{align*}
(6) \quad Q_H(X, \lambda_L, \lambda_H, \mu) &= Q_L(X, \lambda_L, \lambda_H, \mu) \left( 1 - \frac{\lambda_H}{X\mu} \right) .
\end{align*}
Equations (1) – (6) represent a complete computationally tractable approach to estimating the expected customer service time as a function of the number of field engineers and $p_3(K)$, the probability that a needed part is not in the kit.

2.5.2. Development of the Budget Constraint

Next, let us consider the budget constraint. In this model, we consider four types of costs. The equation for each type is given below. The first is the annual cost of providing the field engineers, including salary, benefits, travel expenses, and vehicle costs. We assume that total cost is a linear function of the number of field engineers. The second cost is the annual normal replenishment cost of the field engineers’ kits. Normal replenishment costs are incurred each time a demand for a part is satisfied from the spare parts kit. If the demand can not be satisfied from the kit, then an emergency delivery cost is incurred. The third cost is the annual emergency delivery costs. The final cost is the holding cost for the spare parts kit. These four costs are given by the following formulas.

Field Engineer Cost:  $c_{FE} \times X$

Normal Replenishment Cost:  $\lambda \times p_1 \times p_2 \times (1 - p_3(K)) \times c_{N1}$

Emergency Cost:  $\lambda \times p_1 \times p_2 \times p_3(K) \times c_E$

Holding Cost:  $\sum_{i=1}^{N} (X \times K_i \times H_{ii})$
The total annual cost is the sum of these four costs. We assume that the system planners must choose the number of field engineers, $X$, and the kit stocking levels, $K$, to ensure that this total cost does not exceed a given annual corporate budget, $B$, for field support.

2.6. Model Solution

2.6.1. Initial Model

Our goal, then, is to minimize the expected customer service time subject to a budget constraint. We can express this model in the following manner.

(M1)

$$\begin{align*}
\text{min} & \quad \{ D_1 + \mu^{-1} + p_1 Q_L(X, \lambda_L, \lambda_H, \mu) \\
& \quad + p_1 p_2 p_3(K) p_4 \left[ \tau_{EB} + Q_H(X, \lambda_L, \lambda_H, \mu) \right] \\
X, K_1, \ldots, K_N \\
\text{subject to} & \\
& \quad c_{FE} X + \sum_{i=1}^{N} (X K_i H_{i1}) + \lambda p_1 p_2 (1 - p_3(K)) c_{N1} \\
& \quad + \lambda p_1 p_2 p_3(K) c_E \leq B \\
X & \in \{1, 2, 3, \ldots\} \\
K_i & \in \{0, 1, 2, \ldots\} , \quad i = 1, 2, \ldots, N
\end{align*}$$

where $\mu$, $Q_L$, and $Q_H$ are given by equations (4), (5), and (6), respectively.
In order to solve this model, let us first note that the optimal number of field engineers is easily bounded. A lower bound on the optimal number of filed engineers is the number needed to keep the queue–wait–time finite and an upper bound is the maximum number which the budget will allow. Denote these bounds by $X_L$ and $X_U$, respectively:

\[ (8) \quad X_L = \frac{\lambda_L}{\mu} \; ; \; \text{and} \]
\[ (9) \quad X_H = \frac{B}{C_{FE}} . \]

Since these bounds will be relatively close for realistic instances, we propose enumerating all values of $X$ within the range, solving the resulting spare parts kitting problem for each value of $X$, and selecting the optimal value of $X$. However, even if the number of field engineers is set, the objective function is not separable by part type due to the dependence of $Q_L$ and $Q_H$ on the entire vector $K$. Nevertheless, a different decomposition is possible.

**Theorem [1]:** For single echelon field support systems (M1) with a fixed number of field engineers, the minimum expected total customer service time is achieved by the kit composition which minimizes $p_3(K)$ subject to the budget constraint.

**Proof:** To prove this theorem, we will show that $E[CST]$ is a monotone increasing function of $p_3(K)$. For ease of notation, let $p = p_3(K)$. 
Let \( t = \mu^{-1} \). Then by (4)

\[
t = c_1 + c_2 \ p,
\]

where

\[
c_1 = p_1 \ (D_1 + T + D_2) + p_1 \ (1 - p_2) \ R_A + p_1 \ p_2 \ R_P \quad \text{and}
\]

\[
c_2 = p_1 \ p_2 \ (p_4 \ T + (1 - p_4) \ \tau_{EW}).
\]

Observe that \( c_2 > 0 \).

\[
E[C.S.T.] = D_1 + c_1 + c_2 \ p + p_1 \ Q_L + p_1 \ p_2 \ p_4 (\tau_{EB} + Q_{H}) .
\]

To show that \( \frac{\partial E[C.S.T.]}{\partial p} > 0 \), it suffices to show that

\[
\frac{\partial Q_L}{\partial p} > 0 \quad \text{and} \quad \frac{\partial Q_H}{\partial p} > 0 .
\]

Equivalently, it suffices to show that

\[
\frac{\partial Q_L^{-1}}{\partial p} < 0 \quad \text{and} \quad \frac{\partial Q_H^{-1}}{\partial p} > 0 .
\]

Let

\[
a = X! \sum_{n=0}^{X-1} \frac{(\lambda_L + \lambda_H)^n t^{n-X}}{n!} ;
\]

\[
b = \frac{X}{t} ;
\]

\[
c = 1 - \frac{\lambda_L t}{X} ; \quad \text{and}
\]

\[
d = 1 - \frac{(\lambda_L + \lambda_H) t}{X} .
\]
Then \( Q_L^{-1} = (ad + b)cd \) and \( Q_H^{-1} = (ad + b)d \).

It can be shown that

\[
\frac{\partial a}{\partial p} = X! \sum_{n=0}^{X-1} \frac{(n-X)(\lambda_L + \lambda_H)^{n-X}}{n!} t^{n-X} \left[ \frac{\lambda p_1 p_2 p_4}{\lambda_L + \lambda_H} + \frac{c_2}{t} \right] < 0,
\]

since \( (n-X) < 0 \) for all \( n \leq X - 1 \).

Similarly,

\[
\frac{\partial b}{\partial p} = \frac{-c_2 X}{t^2} < 0;
\]

\[
\frac{\partial c}{\partial p} = \frac{-c_2}{X} < 0; \quad \text{and}
\]

\[
\frac{\partial d}{\partial p} = \frac{-1}{X} \left[ t \frac{\partial \lambda_H}{\partial p} + (\lambda_L + \lambda_H) \frac{\partial t}{\partial p} \right]
\]

\[
= \frac{-1}{X} \left( \lambda p_1 p_2 p_4 t + (\lambda_L + \lambda_H) c_2 \right) < 0.
\]

Hence, \( \frac{\partial Q_L^{-1}}{\partial p} < 0 \) and \( \frac{\partial Q_H^{-1}}{\partial p} < 0 \).

Therefore, \( \text{E[C.S.T.]} \) is a monotone increasing function of \( p \) and is

minimized by the minimum feasible \( p \).
This theorem simplifies our optimization problem to the following:

(M2)

\[
\begin{align*}
\min & \quad p_3(K) \\
\text{subject to} & \quad \sum_{i=1}^{N} (X_i K_i H_{i1}) + \lambda p_1 p_2 (1 - p_3(K)) c_{N1} + \lambda p_1 p_2 p_3(K) c_E \leq B \\
& \quad X \in \{1, 2, 3, \ldots\} \\
& \quad K_i \in \{0, 1, 2, \ldots\} \text{ for } i = 1, 2, \ldots, N
\end{align*}
\]

This model is similar to the kitting problem in the literature (Graves [8]) with the exception that the budget constraint includes the cost of normal and emergency replenishments in addition to the inventory holding costs.

### 2.6.2. Calculation of the Kit Non-Fill Rate

At this point we must address the calculation of \( p_3(K) \). We assume that each part fails independently. For a field service support system in which the part failures are dependent, we could form "superparts" as collections of the dependent parts and calculate the combined costs.

If we assume that more than one part may be required for a repair, as Graves [8] does, then \( p_3(K) \) has a product form. However, in this case, unless \( c_{N1} \) is equal to \( c_E \), (M2) is not separable by part and enumeration
appears to be the only way to solve it to optimality. For this section, let us assume that at most one part is needed per repair. By this assumption, \( p_3(K) \) is the weighted sum of the individual part non-fill rates, which are denoted \( p_{3i}(K_i) \) for \( i = 1, \ldots, N \). Let \( G_i \) denote the probability that part \( i \) is the needed part. Therefore, the objective function and budget constraint are additive by part as follows:

\[
(\text{M3})
\]

\[
\begin{align*}
\min & \quad \sum_{i=1}^{N} G_i \ p_{3i}(K_i) \\
\text{subject to} & \quad \sum_{i=1}^{N} (X \ K_i \ H_{i1} + \lambda \ p_1 \ p_2 \ p_{3i}(K_i) \ G_i \ c_E) \leq B - \lambda \ p_1 \ p_2 \ c_{N1} \\
\end{align*}
\]

\[
X \in \{1, 2, \ldots\} \\
K_i \in \{0, 1, 2, \ldots\} \text{ for } i = 1, 2, \ldots, N
\]

The calculation of \( p_{3i}(K_i) \) depends on the specific inventory policy being used for the kit's normal replenishment. We shall assume that a \((K_i-1,K_i)\) policy is used for the kit restocking. In this policy, also called a one-for-one ordering policy, a replacement part is ordered each time a part is used from the kit. In Smith [19], this spare parts system is analyzed as an M/M/K/K queue. We note that an M/M/K/K queue is equivalent to an M/G/K/K queue [10]. Therefore, whatever distribution the normal
replenishment lead time has, this analysis is appropriate. He shows that the number of busy servers, that is, the number of empty spots in the kit, has a truncated Poisson distribution:

\[
\frac{r^K_i}{K!}, \quad \text{where} \quad r = \frac{\lambda p_1 p_2 \tau_{n1}}{X}.
\]

(10) \( p_{3i}(K_i) = \frac{r^K_i}{K!} \), where \( r = \frac{\lambda p_1 p_2 \tau_{n1}}{X} \).

For fixed \( X \), the problem (M3) can be solved by dynamic programming.

The general solution method is to iterate over the number of field engineers from \( X_L \) to \( X_U \), solving the resulting kit composition problem by dynamic programming, and to select the optimal number of field engineers.

2.7. Model Analysis

The purpose of this section is to exercise the model developed in the previous sections and to introduce the types of plots that are helpful in making observations about policy decisions. We will begin to explore the model by analyzing two parameter sets. The parameters set are described in detail in Appendix A. The first parameter set, K1–R1, represents a rural environment, with a travel time to repair time ratio of 2.0. The second set, K1–U1, represents an urban environment, with a travel time to repair time ratio of 0.25. Both environments have the same part type profile of nine different part types. Each part type has two distinguishing characteristics:
the kit holding cost and the probability of being the required part. Table 1 below indicates these characteristics for each part.

Table 1. Part Profile for Kit Type 1

<table>
<thead>
<tr>
<th>Probability</th>
<th>Low Cost</th>
<th>Medium Cost</th>
<th>High Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low Probability</td>
<td>K1</td>
<td>K2</td>
<td>K3</td>
</tr>
<tr>
<td>Medium Probability</td>
<td>K4</td>
<td>K5</td>
<td>K6</td>
</tr>
<tr>
<td>High Probability</td>
<td>K7</td>
<td>K8</td>
<td>K9</td>
</tr>
</tbody>
</table>

In this section, we focus on a fixed budget level and vary the number of field engineers to establish the tradeoff between field engineer investment and spare part investment. The lowest value of the field engineers is the minimum number required to keep the queue from exploding. In Chapter 5 we explore the effect of varying the budget with the two echelon model.

2.7.1. Customer Service Time Plot

The Customer Service Time Plot demonstrates the effect of staff size on the customer service time. Total customer service time can be broken into three parts: the expected first queue wait time, the expected part acquisition and return visit time, and a remaining constant. This constant includes the initial travel time, diagnostic time, etc. Since the constant time is independent of the staff size and the budget, we ignore it. We graph three quantities in this plot, the expected first queue time, the expected part
acquisition and return visit time, and the sum of these two quantities, each as a function of the field engineer staff size.

Figures 2 and 3 are the Customer Service Time Plots for the rural and urban environments, respectively. In both plots, we see that the part acquisition time increases as we increase the number of field engineers. This increase is due to the decrease in the money available to provide spare parts and the resulting increase in the probability of needing an emergency delivery. This also results in an increase in the expected time which the field engineer must spend. We notice that the queue time decreases, then increases. In any queueing system, an increase in the number of servers in a queue will decrease the wait time; while an increase in the expected service time will increase the wait time. In the urban environment, as the number of field engineers changes from 9 to 10, the effect of the servers is stronger than the effect of the increased service time; but as the number increases from 10 to 11, the effect of the service time is stronger than the effect of the servers. Similar effects are seen in the rural environment. Note that the reallocation of the total budget to employ 10 instead of 9 field engineers will cause a 65% improvement in the total controllable time for the urban environment. The rural environment would experience a 64% improvement in the total controllable time by employing 11 field engineers instead of 10.
Figure 2: Customer Service Time Plot for K1–R1: A Rural Environment
Figure 3: Customer Service Time Plot for K1-U1 : An Urban Environment
2.7.2. Inventory Budget Allocation Plot

The inventory allocation plot is a bar chart showing the amount of money allocated to emergency delivery costs, to normal replenishment costs, and to kit holding costs. The height of each bar indicates the number of dollars invested and the percentages of the inventory budget invested are given by the numbers to the left of the bar.

Figures 4 and 5 present the Inventory Budget Allocation Plots for the rural and urban environments, respectively. We see in Figure 4 for the rural environment that although the total amount of inventory budget decreased, the allocation between the types of costs remained essentially constant. This is another indication that the change in the kits corresponding to changing the field engineers from 10 to 11 is not significant. This insignificance is also reflected in Figure 3 by the essentially zero part acquisition time for both levels of the field engineer staff. In Figure 5 for the urban environment, the allocation changes slightly and is also reflected in the slight increase in the part acquisition time in Figure 2.

For both environments, a dramatic change in allocation occurs when the number of field engineers is increased to the highest level. This indicates that the available inventory budget is not sufficient to provide a kit fill rate of 100% and therefore some of the inventory budget must be spent on emergency orders. The actual fill rates are given in the table below.
Table 2. Fill Rates for K1–R1 and K1–U1

<table>
<thead>
<tr>
<th></th>
<th>K1–R1 : Rural</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of FEs</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>Fill Rate = 1–p3</td>
<td>99.99 %</td>
<td>99.81 %</td>
<td>86.32 %</td>
</tr>
<tr>
<td></td>
<td>K1–U1 : Urban</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. of FEs</td>
<td>9</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>Fill Rate = 1–p3</td>
<td>99.96 %</td>
<td>98.88 %</td>
<td>77.96 %</td>
</tr>
</tbody>
</table>
Figure 4: Inventory Budget Allocation Plot for K1–R1
Figure 5: Inventory Budget Allocation Plot for K1–U1
2.7.3. Kit Composition Plot

The Kit Composition Plot is a bar chart showing the fraction of the holding cost allocated to each part type. The height of the block indicates the fraction of the holding cost allocated to the corresponding part. On the top of each bar is the total holding cost in dollars. Each part type block is indicated by a combination of shading density and shading angle. The high, medium, and low cost parts are indicated by high, medium, and low shading densities respectively. The probability that a part is the needed part is indicated by the shading angle. The low probability parts are K1, K2, and K3; the medium probability parts are K4, K5, and K6; and the high probability parts are K7, K8, and K9. The quantity of each part type and the holding cost fraction are also given in Tables 3 and 4 at the end of this section.

Figure 6 is the Kit Composition Plot for the rural environment, K1–R1. We observe that the greatest changes between kit compositions corresponding to different field engineer levels occur in the high cost parts (the densest shadings). For example, the percentage of the holding cost assigned to part type 3 decreases from 15% to 0% which is reasonable since this part has a high cost and a low probability of being needed. We may also observe that over 50% of the budget is spent on the three high probability parts, part types 7, 8, and 9.

Figure 7 is the Kit Composition Plot for the urban environment K1–U1. Again, the greatest changes occur in the high cost parts, with parts
3 and 6 decreasing to 0%. This plot reveals that the optimal kit compositions for 10 versus 11 field engineers are quite different. This difference further emphasizes the need to consider the interactions between field engineer staff level and kit composition. In addition, we observe the high probability parts account for 60 to 75% of the budget.

Table 3. Kit Composition for K1–R1: Rural Environment

<table>
<thead>
<tr>
<th>Part Type</th>
<th>Quantity</th>
<th>10 F.E.s Holding Cost</th>
<th>10 F.E.s Quantity</th>
<th>11 F.E.s Holding Cost</th>
<th>11 F.E.s Quantity</th>
<th>12 F.E.s Holding Cost</th>
<th>12 F.E.s Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>0.011</td>
<td></td>
<td>2</td>
<td>0.010</td>
<td>2</td>
<td>0.029</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.042</td>
<td></td>
<td>2</td>
<td>0.048</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0.113</td>
<td></td>
<td>1</td>
<td>0.095</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0.017</td>
<td></td>
<td>6</td>
<td>0.029</td>
<td>3</td>
<td>0.043</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>0.071</td>
<td></td>
<td>3</td>
<td>0.071</td>
<td>1</td>
<td>0.071</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>0.226</td>
<td></td>
<td>2</td>
<td>0.190</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>0.025</td>
<td></td>
<td>7</td>
<td>0.033</td>
<td>5</td>
<td>0.071</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>0.099</td>
<td></td>
<td>6</td>
<td>0.143</td>
<td>3</td>
<td>0.214</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>0.395</td>
<td></td>
<td>4</td>
<td>0.381</td>
<td>2</td>
<td>0.571</td>
</tr>
</tbody>
</table>
Table 4. Kit Composition for K1–U1: Urban Environment

<table>
<thead>
<tr>
<th>Part Type</th>
<th>Quantity</th>
<th>Holding Cost</th>
<th>Quantity</th>
<th>Holding Cost</th>
<th>Quantity</th>
<th>Holding Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>0.006</td>
<td>3</td>
<td>0.010</td>
<td>3</td>
<td>0.031</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.031</td>
<td>2</td>
<td>0.033</td>
<td>1</td>
<td>0.052</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0.083</td>
<td>1</td>
<td>0.066</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>0.014</td>
<td>5</td>
<td>0.016</td>
<td>5</td>
<td>0.052</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>0.062</td>
<td>3</td>
<td>0.049</td>
<td>2</td>
<td>0.103</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>0.207</td>
<td>3</td>
<td>0.197</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>7</td>
<td>13</td>
<td>0.027</td>
<td>12</td>
<td>0.039</td>
<td>9</td>
<td>0.093</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>0.114</td>
<td>8</td>
<td>0.131</td>
<td>5</td>
<td>0.258</td>
</tr>
<tr>
<td>9</td>
<td>11</td>
<td>0.455</td>
<td>7</td>
<td>0.459</td>
<td>2</td>
<td>0.412</td>
</tr>
</tbody>
</table>
Figure 6: Kit Composition Plot for K1–R1
Figure 7: Kit Composition Plot for K1-U1
2.7.4. Comparing Rural and Urban Environments

In this section, we compare the output results for the rural and urban environments. The first difference between the two environments is the minimum number of field engineers needed to keep the queues from exploding. In the rural environment, 10 field engineers must be hired; while in the urban environment, 9 field engineers are sufficient. Also, the optimal number of field engineers is different: for the rural, 11 is optimal and for the urban, 10 is optimal. However, both environments do have the same general shape in the Customer Service Time Plots. We also note that the urban environment achieves a smaller customer service time for all levels of field engineers.

In the Inventory Budget Allocation Plots, we see that the environments again have the same basic features when we compare them without regard to the number of field engineers. However, when we compare the budget allocation for the two environments for 11 field engineers, they are very different. In the rural environment, 92% of the inventory costs are allocated to holding costs; while in the urban environment, only 42% of the inventory costs are allocated to holding costs. This reflects the differences in the emergency delivery time and in the travel time.

The Kit Composition Plots for the environments are also similar when compared without regard for the number of field engineers. Once again focusing on the kit compositions corresponding to 11 field engineers,
we note two major differences. First, the urban environment has dropped parts K3 and K6 from the kit; while the rural environment keeps them in the kit. Second, the urban environment increases the number of parts K1 and K7 which are carried. These differences in the number of parts carried are reflected in the holding cost percentages for the various parts.

These differences in the rural and urban environments emphasize the need to determine the important characteristics of the field service environment and to develop models which incorporate these characteristics.

2.8. Limitations of the Single Echelon Model

The model which we have developed and analyzed in this chapter has two important limitations. First, it only considers the inventory in the field engineers' kits and assumes that the emergency supply source has an infinite supply. Second, since the model is solved using dynamic programming, the number of part types must be relatively small in order for the model to be solved in a reasonable amount of time. These two limitations will be overcome by expanded models in the next two chapters.
CHAPTER 3. TWO ECHELON MODEL

3.1. Introduction to Two Echelon Systems

In this chapter we develop a model for two echelon field service support systems analogous to the single echelon model of the previous chapter. In a two echelon system, inventory is stocked at a local depot as well as in the field engineers' kits. Normal replenishment for the kit is assumed to come directly from the central warehouse. However, the existence of the local depot means that there are two sources of supply available if the field engineer does not have the part in his kit to complete a repair. We assume that the field engineer will always try first to obtain the part from the local depot if the part is not in his or her kit. If the local depot does not have the part, then it will be ordered from the central warehouse. A delivery from the local depot will be called a "quick" delivery; while a delivery from the central warehouse to satisfy a stockout will be called an "emergency" delivery.

The interactions between the two echelons are reflected in the budget allocation and in the objective function, as follows. We consider the holding costs, the replenishment costs, and the delivery costs of both echelons, depot and kit level, in the budget allocation. The objective of customer downtime depends upon whether spare parts are available immediately by quick delivery or by emergency delivery. The probabilities of these events are interrelated by the stocking levels at the two echelons. In particular, a demand for a part at the local depot is realized only when
the field engineer's kit is out of stock. Consequently, the stocking level at
the kit echelon will affect the stockout probability at the depot.

This chapter is organized in the same manner as Chapter 2. Section
3.2 examines the realizations of a service request. Sections 3.3 and 3.4
review the notation and assumptions, respectively. The model is developed
in Section 3.5, solved in Section 3.6, and analyzed in Section 3.7.

3.2. Realizations of a Service Request

Figure 8 presents the possible realizations of a service request for a
two echelon system. The figure is the same as the single echelon figure up
to the p3(K) junction. In the two echelon system, if the field engineer does
not have the part, then he or she tries to order it from the local depot. If the
depot has the part in stock, then it will ship the part directly to the customer
location. The field engineer must decide whether to break the repair. This
decision results in outcome D or outcome E, as in the single echelon case.
Unlike the single echelon case, however, we assume that the part
acquisition time is the same whether or not the field engineer breaks the
repair. This time is called the quick delivery time (τQ). Let f(K,S)
denote the probability that the depot will be out of stock. In this case,
denoted as outcome F on the figure, the part is ordered from the central
warehouse and we assume that the repair is always broken. The customer
then experiences the emergency delivery time (τE), a revisit queue wait
time (QH), a travel time (T), and, finally, the repair time (Rp).
Figure 8: Realizations of a Service Request – Two Echelon Systems
3.3. Notation

In addition to the notation introduced in Chapter 2, we define the following variables.

Model Parameters:

$\tau_Q$ : the quick part delivery time from the local depot; and

$\tau_E$ : the emergency part delivery time from the central warehouse.

Decision Variables:

$S_i$ : the number of units of part type $i$ stocked at the local depot.

Calculated Values:

$f(K,S)$ : the probability that the local depot is out of stock.

3.4. Assumptions

In addition to assumptions (A1) through (A8) in Chapter 2, we have also assumed the following:

(A9) Normal replenishments for the kits and the local depot come from the central warehouse.

For notational convenience, we will assume:

(A10) Part delivery time from the local depot is independent of whether the field engineer has broken the repair. If an emergency delivery is required, then the repair is always broken.

Assumption (A10) is reasonable in the two echelon system since if a local depot exists close enough to provide quick deliveries, then the central warehouse is probably far away.
3.5. Model Development

3.5.1. Development of the Objective Function

As in Chapter 2, the object is to maximize customer service by minimizing the expected total customer service time, which is calculated with the aid of Figure 8.

\[
E(C.S.T.) = D_1 + p_1 (Q_L + T + D_2) + p_1 (1-p_2) R_A + p_1 p_2 R_P \\
+ p_1 p_2 p_3(K) (1-f(K,S)) [p_4 (\tau_Q + Q_H + T) + (1-p_4) \tau_Q] \\
+ p_1 p_2 p_3(K) f(K,S) [\tau_E + Q_H + T].
\]

(11)

Again, we use the M/M/X with two priority classes queueing model to calculate \(Q_L\) and \(Q_H\). In the two echelon system, we define

\[
\frac{1}{\mu} = \{ p_1 (T + D_2) + p_1 (1-p_2) R_A + p_1 p_2 R_P \\
+ p_1 p_2 p_3(K) (1-f(K,S)) [(1-p_4) \tau_Q + p_4 T] \\
+ p_1 p_2 p_3(K) f(K,S) T \}
\]

(12)

and

\[
\lambda_H = \lambda p_1 p_2 p_3(K) [f(K,S) + (1-f(K,S)) p_4].
\]

(13)

The expression for \(\lambda_H\) is derived by considering the two ways in which a revisit could occur, the local depot is out of stock and the local depot has the part, but the field engineer decides to break the repair. Then \(Q_L\) and \(Q_H\) are calculated as before using equations (5) and (6).

(5) and (6)
3.5.2. Development of the Budget Constraint

In the two echelon system, we consider for each echelon the same types of costs as in the single echelon system. Therefore we have:

Field Engineer Cost: \( X c_{FE} \);

Kit Holding Cost: \( \sum_{i=1}^{N} X H_{i1} K_i \);

Depot Holding Cost: \( \sum_{i=1}^{N} H_{i2} S_i \);

Kit Replenishment Cost: \( \lambda p_1 p_2 (1 - p_3(K)) c_{N1} \);

Depot Replenishment Cost: \( \lambda p_1 p_2 p_3(K) (1 - f(K, S)) c_{N2} \);

Quick Delivery Cost: \( \lambda p_1 p_2 p_3(K) (1 - f(K, S)) c_Q \) and

Emergency Delivery Cost: \( \lambda p_1 p_2 p_3(K) f(K, S) c_E \).

We constrain the inventory problem by requiring that the sum of the above costs be less than or equal to the budget.

3.6. Solution Method

Using the equations above, we can express our two echelon model as indicated in Figure 9 on the following page.
\( \text{(M4)} \)

\[
\min \quad \{ D_1 + p_1(T + D_2) + p_2(1 - p_3)R_A + p_1p_2R_P + p_1p_2p_3(K)[\tau_Q + T] \\
X, K_1, \ldots, K_N, \\
S_1, \ldots, S_N \}
\]

subject to

\[
Xc_{FE} + \sum_{i=1}^{N} \left[ XH_1K_i + H_2S_i \right] + \lambda p_1p_2(1 - p_3(K))c_{N1} \\
+ \lambda p_1p_2p_3(K)(1 - f(K, S))[c_{N2} + c_Q] + \lambda p_1p_2p_3(K)f(K, S)c_E \leq B; 
\]

\[
\mu^{-1} = \{ p_1(T + D_2) + p_2(1 - p_3)R_A + p_1p_2R_P + p_1p_2p_3(K)f(K, S)T + p_1p_2p_3(K)(1 - f(K, S))[(1 - p_4)p_4 + p_4T] \}; 
\]

\[
\lambda_H = \lambda p_1p_2p_3(K)[f(K, S) + (1 - f(K, S))p_4]; 
\]

\[
Q_L = \frac{\left[ X! \left(1 - \frac{\lambda_L + \lambda_H}{X\mu}\right)^{X-1} \sum_{n=0}^{X-1} \left(\frac{\lambda_L + \lambda_H}{\mu}\right)^{n-X} \frac{1}{n!} + X\mu \right]^{-1}}{\left(1 - \frac{\lambda_H}{X\mu}\right) \left(1 - \frac{\lambda_L + \lambda_H}{X\mu}\right)}; 
\]

\[
Q_H = Q_L \left(1 - \frac{\lambda_H}{X\mu}\right); 
\]

\( X \in \{1, 2, 3, \ldots\}; \)

\( K_i \in \{0, 1, 2, \ldots\}, \quad \text{for} \quad i = 1, 2, \ldots N; \quad \text{and} \)

\( S_i \in \{0, 1, 2, \ldots\}, \quad \text{for} \quad i = 1, 2, \ldots N. \)

Figure 9: Two Echelon Model (M4)
As in the single echelon case, the optimal number of field engineers can be bounded. We iterate over the number of field engineers and solve the resulting inventory problem for a fixed number of field engineers at each iteration. For the single echelon model, we were able to show that the original problem was equivalent to an easier problem. Unfortunately, the interactions between echelons prevent us from forming a similar theorem for this model. We again assume that at most one part is needed for a repair, which implies that \( p_3(K) \) and \( f(K,S) \) have summation forms.

In order to solve the model, we first simplify the model by defining new constants.

\[
\begin{align*}
  d_1 &= D_1 + p_1(T + D_2) + p_1(1 - p_2)R_A + p_1p_2R_P ; \\
  d_2 &= p_1p_2(p_4T + (1 - p_4)\tau_Q) ; \\
  d_3 &= p_1p_2[(1 - p_4)T - (1 - p_4)\tau_Q] ; \\
  d_4 &= \lambda p_1p_2p_4 ; \\
  d_5 &= \lambda p_1p_2(1 - p_4) ; \\
  d_6 &= p_1p_2(T + \tau_Q) ; \quad \text{and} \\
  d_7 &= p_1p_2(\tau_E - \tau_Q + (1 + p_4)T) .
\end{align*}
\]

Using these constants, we can write the following:

\[
\begin{align*}
  (14) & \quad \mu^{-1} = \{d_1 - D_1 + d_2p_3(K) + d_3p_3(K)f(K,S)\} ; \\
  (15) & \quad \lambda_H = d_4p_3(K) + d_5p_3(K)f(K,S) ; \quad \text{and} \\
  (16) & \quad E(C.S.T.) = d_1 + d_6p_3(K) + d_7p_3(K)f(K,S) + p_1Q_L \\
  & \quad + p_1p_2p_3(K)p_4Q_H + p_1p_2p_3(K)f(K,S)(1 - p_4)Q_H .
\end{align*}
\]
We also define the following constants to simplify the budget constraint.

\[ d_8 = X \ c_{FE} + \lambda \ p_1 \ p_2 \ c_{N1} ; \]

\[ d_9 = \lambda \ p_1 \ p_2 \ (c_{N2} + c_Q - c_{N1}) ; \quad \text{and} \]

\[ d_{10} = \lambda \ p_1 \ p_2 \ (c_E - c_{N2} - c_Q) . \]

Using these constants, we rewrite (M4) as shown in Figure 10 on the following page.
(M5)

\[
\begin{align*}
\min & \quad \{ d_1 + d_6 p_3(K) + d_7 p_3(K) f(K, S) + p_1 Q_L \\
& \quad + p_1 p_2 p_3(K) p_4 Q_H + p_1 p_2 p_3(K) f(K, S) (1 - p_4) Q_H \} \\
\text{subject to} & \quad \sum_{i=1}^{N} (X_i H_{i1} K_i + H_{i2} S_i) + d_9 p_3(K) + d_{10} p_3(K) f(K, S) \leq B - d_8 ; \\
& \quad d_1 - D_1 + d_2 p_3(K) + d_3 p_3(K) f(K, S) = \mu^{-1} ; \\
& \quad d_4 p_3(K) + d_5 p_3(K) f(K, S) = \lambda_H ; \\
& \quad \frac{X! \left(1 - \frac{\lambda_L + \lambda_H}{X\mu} \right)^{X-1} \sum_{n=0}^{X} \left(\frac{\lambda_L + \lambda_H}{\mu} \right)^{n-X} \frac{1}{n! + X\mu}}{\left(1 - \frac{\lambda_H}{X\mu} \right) \left(1 - \frac{\lambda_L + \lambda_H}{X\mu} \right)} = Q_L ; \\
& \quad Q_L \left(1 - \frac{\lambda_H}{X\mu} \right) = Q_H ;
\end{align*}
\]

\[K_i \in \{0, 1, 2, \ldots\}, \quad \text{for } i = 1, 2, \ldots N ; \quad \text{and} \]

\[S_i \in \{0, 1, 2, \ldots\}, \quad \text{for } i = 1, 2, \ldots N. \]

Figure 10: Model (M5)
Next, we associate the Lagrangian multiplier, $\gamma$, with the budget constraint and bring it into the objective. Now, we need to solve model (M6) below for different values of $\gamma$ to get the optimal solution for a range of budget levels. We also decompose $p_3(K)$ and $f(K,S)$ into their summation forms, where the subscripts refer to part types.

(M6)

$$\min_{K_1, \ldots, K_N, S_1, \ldots, S_N} \left\{ \sum_{i=1}^{N} \left[ G_i \left[p_{3i}(K_i)(d_6 + d_9\gamma) + p_{3i}(K_i) f_i(K_i, S_i)(d_7 + d_{10}\gamma) \right] + p_1 p_2 p_{3i}(K_i) Q_H + p_1 p_2 (1 - p_4) p_{3i}(K_i) f_i(K_i, S_i) Q_H \right] \right\} + \gamma(XH_{i1}K_i + H_{i2}S_i) - \gamma(B - d_8) + p_1 Q_L$$

subject to

$$d_1 - D_1 + d_2 p_3(K) + d_3 p_3(K) f(K, S) = \mu;$$

$$d_4 p_3(K) + d_5 p_3(K) f(K, S) = \lambda_H;$$

$$\left[ X! \left( 1 - \frac{\lambda_L + \lambda_H}{X\mu} \right) \sum_{n=0}^{X-1} \left( \frac{\lambda_L + \lambda_H}{\mu} \right)^n \frac{1}{n!} + X\mu \right]^{-1} = Q_L;$$

$$\left( 1 - \frac{\lambda_H}{X\mu} \right) \left( 1 - \frac{\lambda_L + \lambda_H}{X\mu} \right) Q_L \left( 1 - \frac{\lambda_H}{X\mu} \right) = Q_H;$$

$$K_i \in \{0, 1, 2, \ldots\}, \quad i = 1, 2, \ldots, N; \quad \text{and} \quad$$

$$S_i \in \{0, 1, 2, \ldots\}, \quad i = 1, 2, \ldots, N.$$
We note that interactions between the parts are captured by the values of $\mu$ and $\lambda_H$. Therefore, if we knew the optimal values of $\mu$ and $\lambda_H$, then the values of $Q_L$ and $Q_H$ could be considered constant. The resulting problem would be separable by part type and could be solved as a series of two-variable optimizations. This problem would have the following form:

\[(M7)\]

\[
\min_{K_1, \ldots, K_N, S_1, \ldots, S_N} \quad d_1 + p_1 Q_L + \sum_{i=1}^{N} \{ \begin{array}{l}
G_i (p_3(K_i) (d_6 + dsy + p_1 p_2 p_4 Q_H) \\
+ p_3(K_i) f_i(K_i, S_i) (d_7 + \gamma d_{10} + p_1 p_2 (1 - p_4) Q_H) \\
+ \gamma (XH_iK_i + H_{i2}S_i) \end{array} \} - \gamma (B - d_8)
\]

subject to

\[
K_i \in \{0, 1, 2, \ldots\}, \quad i = 1, 2, \ldots, N; \quad \text{and}
\]

\[
S_i \in \{0, 1, 2, \ldots\}, \quad i = 1, 2, \ldots, N.
\]

This problem can be solved as a series of two-variable optimization problems. The first two-variable problem would consider only $K_1$ and $S_1$; the second would consider $K_2$ and $S_2$; and so on. There would be a total of $N$ two-variable problems to solve.

Although we do not know the optimal values of $\lambda_H$ and $\mu$ exactly, we can find bounds on them as follows.
Define \[ \mu_L = \frac{1}{d_1 - D_1 + p_1 p_2 [(1 + p_4)T + (1 - p_4)\pi_Q]} \]
and \[ \mu_U = \frac{1}{d_1 - D_1} . \]

Lemma 1: \( \mu_L \leq \mu \leq \mu_U \).

Proof: The lower bound can be derived as follows:

\[ \mu^{-1} = d_1 - D_1 + p_3(K)p_1 p_2[f(K,S)T + (1 - f(K,S))(p_4T + (1 - p_4)\pi_Q)] \]

\[ \leq d_1 - D_1 + p_1 p_2 [(1 + p_4)T + (1 - p_4)\pi_Q] ; \]

since \( 0 \leq f(K,S) \leq 1 \) and \( p_3(K) \leq 1 \).

This implies \( \mu_L \leq \mu \).

The upper bound can be derived in a similar manner:

\[ \mu^{-1} = d_1 - D_1 + p_3(K)p_1 p_2[f(K,S)T + (1 - f(K,S))(p_4T + (1 - p_4)\pi_Q)] \]

\[ \geq d_1 - D_1 , \]

since \( 0 \leq f(K,S) \leq 1 \) and \( 0 \leq p_3(K) \).

This implies \( \mu \leq \mu_U \).
Lemma 2: \[ 0 \leq \lambda_H \leq p_1 p_2 \lambda. \]

*Proof:* \[ \lambda_H = \lambda_1 p_1 p_2 p_3(K) \left[ f(K, S) + (1 - f(K, S)) p_4 \right] . \]

\[ 0 \leq p_3(K) \Rightarrow 0 \leq \lambda_H . \]

\[ p_3(K) \leq 1 \Rightarrow \lambda_H \leq \lambda_1 p_1 p_2 \left[ p_4 + (1 - p_4) f(K, S) \right] . \]

\[ f(K, S) \leq 1 \Rightarrow \lambda_H \leq \lambda_1 p_1 p_2 . \]

These bounds limit the search for the optimal values of the high priority arrival rate and the service rate to a small range.

Unfortunately, (M6) is a difficult non-linear, non-differentiable integer programming problem. Therefore, we will not find a provable optimal solution, but will use an optimization-based algorithm to find a good solution:

Step 1. Set \( \lambda_H = 0 \) and \( \mu = \mu_U \).

Step 2. Calculate \( Q_L, Q_H, d_1, \ldots, d_{10} \) as functions of \( \lambda_H \) and \( \mu \).

Step 3. Solve (M7) as a series of \( N \) two variable optimization problems.

Step 4. Set \( \lambda_T = \) true arrival rate of solution found in Step 3, using equation (13).

Step 5. Set \( \mu_T = \) true service rate of solution found in Step 4, using equation (12).

Step 6. If \( \lambda_H = \lambda_T \) and \( \mu = \mu_T \), then stop with a good solution,

else set \( \lambda_H := \lambda_T \) and \( \mu := \mu_T \) and go to Step 2.

This algorithm does not always converge to a feasible solution. An
example where it does not converge, due to a duality gap, is presented in section 4.4.6. In most other instances, the algorithm converged in two steps. The best solution is found by varying $\gamma$ and choosing among the good solutions found by the algorithm above.

3.7. Model Analysis

3.7.1. Customer Service Time Plot

Figures 11 and 12 are the customer service time plots for the two echelon data sets K2-U1 and K2-R1 respectively. These data sets are the same ones used in Chapter 2, with the depot holding cost parameters added. In Figure 11, we see the part acquisition time increasing dramatically when the number of field engineers increases from 11 to 12. This increase corresponds to a fill rate decrease resulting from the lower inventory budget associated with larger staff sizes. However, the initial queue continues to decrease, which indicates that the effect of increasing the number of field engineers overwhelms the increase in the expected service time. On the other hand, in Figure 12, we see that the initial queue wait time drastically increases as the field engineers increase from 12 to 13. In this case, the effect of the increase in the expected service time is much greater than the effect of adding a field engineer. These changes in queue time correspond to the changes in field engineer utilization as given in Table 5.
Table 5. Field Engineer Utilization

<table>
<thead>
<tr>
<th>Number of FEs</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>K2–U1 : Urban FE Utilization</td>
<td>80.5%</td>
<td>73.2%</td>
<td>69.2%</td>
</tr>
<tr>
<td>Number of FEs</td>
<td>11</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>K2–R1 : Rural FE Utilization</td>
<td>80.0%</td>
<td>73.3%</td>
<td>78.5%</td>
</tr>
</tbody>
</table>
Time versus Field Engineers for Constant Budget

Data Set K2-U1: Urban Environment

Figure 11: Customer Service Time Plot for K2-U1
Time versus Field Engineers for Constant Budget

Data Set K2-R1: Rural Environment

- First Queue Wait
- Part Acquisition
- Total Time

Figure 12: Customer Service Time Plot for K2-R1
3.7.2. Inventory Budget Allocation Plot

The inventory budget allocation plots for the two echelon data sets K2–U1 and K2–R1 are Figures 13 and 14 respectively. In both environments, the lower two values of staff size cause a budget allocation in which the majority of the budget is spent on the kit holding cost and a significant portion on the kit replenishment. This allocation corresponds to the very high fill rates given below in Table 6.

Table 6. Fill Rates

<table>
<thead>
<tr>
<th>Number of FEs</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>K2–U1 Kit</td>
<td>100%</td>
<td>100%</td>
<td>87.4%</td>
</tr>
<tr>
<td>K2–U1 Depot</td>
<td>100%</td>
<td>99.8%</td>
<td>84.8%</td>
</tr>
<tr>
<td>Number of FEs</td>
<td>11</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>K2–R1 Kit</td>
<td>100%</td>
<td>100%</td>
<td>77.1%</td>
</tr>
<tr>
<td>K2–R1 Depot</td>
<td>91.0%</td>
<td>89.6%</td>
<td>87.8%</td>
</tr>
</tbody>
</table>

By examining the kit and depot composition (sections 3.7.3 and 3.7.4 below) with the fill rates, we note that when the kit carries enough inventory to provide a fill rate of almost 100%, the depot only needs to stock one of each part type in order to provide very high fill rates for the kit requests. Note that the numbers in the table have been rounded to the nearest tenth of a percent.
Inventory Costs
Data Set K2-U1: Urban Environment

Figure 13: Inventory Budget Allocation Plot for K2-U1
Figure 14: Inventory Budget Allocation Plot for K2–R1
3.7.3. Kit Composition Plot

Figures 15 and 16 are the kit composition plots for the two echelon data sets K2–U1 and K2–R1 respectively. The quantity and holding cost fraction of each part is also given in Tables 7 and 8 at the end of this section. As in Chapter 2, the heights of the bars represent the percentage of the holding cost allocated to each part type. We observe in Figure 15 that the major changes between the kits as the number of field engineers is increased to 12 are the decrease of part K3 to zero and the corresponding increase in parts K8 and K7. This tradeoff is made because K3 is a low probability, high cost part while K7 and K8 are medium probability, low and medium cost parts. In Figure 16, the changes are more drastic, with parts K2, K3, and K6 decreasing to zero. At 13 field engineers, over 80% of the inventory budget is invested in the three high probability parts, with the remaining 20% being invested in the less expensive, lower probability parts. Also note that for each level of field engineers, not only is there a different quantity of each part, but there is also a different percentage of the budget invested in each part. This implies that simple heuristics which allocate the same proportion of the budget to each part type would not be optimal.
### Table 7. Kit Composition for K2–U1: Urban Environment

<table>
<thead>
<tr>
<th>Part Type</th>
<th>Quantity</th>
<th>10 F.E.s</th>
<th>10 F.E.s</th>
<th>11 F.E.s</th>
<th>11 F.E.s</th>
<th>12 F.E.s</th>
<th>12 F.E.s</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Holding Cost</td>
<td>Quantity</td>
<td>Holding Cost</td>
<td>Quantity</td>
<td>Holding Cost</td>
<td>Quantity</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>0.008</td>
<td>4</td>
<td>0.009</td>
<td>2</td>
<td>0.015</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>0.041</td>
<td>3</td>
<td>0.034</td>
<td>1</td>
<td>0.038</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0.110</td>
<td>2</td>
<td>0.090</td>
<td>0</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>0.014</td>
<td>6</td>
<td>0.014</td>
<td>3</td>
<td>0.023</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>0.061</td>
<td>6</td>
<td>0.068</td>
<td>2</td>
<td>0.076</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>0.220</td>
<td>4</td>
<td>0.181</td>
<td>1</td>
<td>0.152</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>17</td>
<td>0.023</td>
<td>12</td>
<td>0.027</td>
<td>7</td>
<td>0.053</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>0.110</td>
<td>11</td>
<td>0.124</td>
<td>5</td>
<td>0.189</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>15</td>
<td>0.412</td>
<td>10</td>
<td>0.452</td>
<td>3</td>
<td>0.455</td>
<td></td>
</tr>
</tbody>
</table>

### Table 8. Kit Composition for K2–R1: Rural Environment

<table>
<thead>
<tr>
<th>Part Type</th>
<th>11 F.E.s</th>
<th>11 F.E.s</th>
<th>12 F.E.s</th>
<th>12 F.E.s</th>
<th>13 F.E.s</th>
<th>13 F.E.s</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Quantity</td>
<td>Holding Cost</td>
<td>Quantity</td>
<td>Holding Cost</td>
<td>Quantity</td>
<td>Holding Cost</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>0.011</td>
<td>3</td>
<td>0.010</td>
<td>1</td>
<td>0.024</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>0.046</td>
<td>2</td>
<td>0.034</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0.148</td>
<td>2</td>
<td>0.136</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0.015</td>
<td>4</td>
<td>0.014</td>
<td>2</td>
<td>0.049</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>0.065</td>
<td>4</td>
<td>0.068</td>
<td>1</td>
<td>0.122</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>0.222</td>
<td>3</td>
<td>0.204</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>0.020</td>
<td>7</td>
<td>0.024</td>
<td>3</td>
<td>0.073</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>0.102</td>
<td>6</td>
<td>0.102</td>
<td>2</td>
<td>0.244</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>0.370</td>
<td>6</td>
<td>0.408</td>
<td>1</td>
<td>0.488</td>
</tr>
</tbody>
</table>
Figure 15: Kit Composition Plot for K2-U1
Figure 16: Kit Composition Plot for K2–R1
3.7.4. Depot Composition Plot

For the two echelon models, we also explore the composition of the depot inventory. Figures 17 and 18 give the depot composition plots for the two echelon data sets K2–U1 and K2–R1 respectively, while Tables 9 and 10 give the numerical values at the end of this section. In Figure 17 we first notice that the total depot holding cost, shown on top of each bar, increases as the number of field engineers increases. This was reflected in the inventory budget allocation plot of Figure 13 in Section 3.7.2. when the percentage of the budget dedicated to depot holding costs increased from 1% to 4%. As the number of field engineers increases from 11 to 12, the depot holding cost more than doubles. However, the quantity of each part type did not double, nor did each part’s percentage of the holding cost remain constant. Instead, for example, part S3 was decreased to zero, while part S9 was increased from 1 to 7.

It is interesting to compare the quantities of each part held at the kit and at the depot. Part 6, the high probability and low cost part, is stocked to a level of 7 in the kit and 2 in the depot; while part 9, the high probability and high cost part, is stocked to a level of 3 in the kit and 7 at the depot. This confirms the advantage of carrying the high cost parts at the depot, which can be used to support all of the field engineers. However, this allocation is also a consequence of the integrality restrictions and our assumption that each field engineer carry the same kit. This assumption implies that the total quantity of a part type in the kit echelon is an integer
multiple of the number of field engineers. On the other hand, any integral quantity is feasible for the depot echelon.

In the depot composition for K2–R1, Figure 18, we observe the same types of changes as in K2–U1, such as K3 decreasing to zero and K9 increasing. In this data set, we see the total depot holding cost increases threefold as the field engineers increase from 12 to 13.

Table 9. Depot Composition for K2–U1: Urban Environment

<table>
<thead>
<tr>
<th>Part Type</th>
<th>Quantity</th>
<th>Holding Cost</th>
<th>Quantity</th>
<th>Holding Cost</th>
<th>Quantity</th>
<th>Holding Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.013</td>
<td>1</td>
<td>0.010</td>
<td>1</td>
<td>0.005</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.064</td>
<td>1</td>
<td>0.051</td>
<td>1</td>
<td>0.023</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.256</td>
<td>1</td>
<td>0.204</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0.013</td>
<td>1</td>
<td>0.010</td>
<td>1</td>
<td>0.005</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0.064</td>
<td>1</td>
<td>0.051</td>
<td>2</td>
<td>0.046</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0.256</td>
<td>2</td>
<td>0.408</td>
<td>2</td>
<td>0.183</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0.013</td>
<td>1</td>
<td>0.010</td>
<td>2</td>
<td>0.009</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0.064</td>
<td>1</td>
<td>0.051</td>
<td>4</td>
<td>0.091</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>0.256</td>
<td>1</td>
<td>0.204</td>
<td>7</td>
<td>0.639</td>
</tr>
</tbody>
</table>
Table 10. Depot Composition for K2–R1: Rural Environment

<table>
<thead>
<tr>
<th>Part Type</th>
<th>Quantity</th>
<th>11 F.E.s Holding Cost</th>
<th>Quantity</th>
<th>12 F.E.s Holding Cost</th>
<th>Quantity</th>
<th>12 F.E.s Holding Cost</th>
<th>Quantity</th>
<th>13 F.E.s Holding Cost</th>
<th>Quantity</th>
<th>13 F.E.s Holding Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td>1</td>
<td>0.017</td>
<td>1</td>
<td>0.006</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.066</td>
<td>1</td>
<td>0.086</td>
<td>2</td>
<td>0.065</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.263</td>
<td>0</td>
<td>0.000</td>
<td>0</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0.000</td>
<td>1</td>
<td>0.017</td>
<td>1</td>
<td>0.006</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0.066</td>
<td>1</td>
<td>0.086</td>
<td>1</td>
<td>0.032</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0.263</td>
<td>1</td>
<td>0.345</td>
<td>2</td>
<td>0.260</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0.013</td>
<td>1</td>
<td>0.017</td>
<td>2</td>
<td>0.013</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0.066</td>
<td>1</td>
<td>0.086</td>
<td>3</td>
<td>0.097</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>0.263</td>
<td>1</td>
<td>0.345</td>
<td>4</td>
<td>0.519</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 17: Depot Composition Plot for K2–U1
Figure 18: Depot Composition Plot for K2–R1
CHAPTER 4. EXTENSION TO A LARGE NUMBER OF PARTS

4.1. Introduction

Each of the models developed in the previous chapters represents each of the part types individually. Realistic field service support systems have thousands of different part types and using these models would require unacceptably large amounts of computer time. To alleviate this problem, we develop a model based on grouping similar parts. We analyze several instances and explore the use of this model more fully.

4.2. Part Grouping

The important features of a part type for the model are the probability that the part is needed and the relevant costs. As previously discussed, for notational simplicity we assume that each part has the same replenishment, quick delivery, and emergency delivery costs. If this were not the case, then the model can be easily extended to incorporate different cost levels for each group of parts.

To develop the part groupings, we first sort the parts according to historical part usage. Then we group the parts at natural breakpoints so that all of the parts in a group have roughly the same usage. Next, within each group, order the parts according to kit holding cost, then further divide the group into cost subgroups. If a two echelon system is being modeled, then these subgroups must be divided further by depot holding cost. For each group, define the following parameters:
\( n_i \): number of different part types in group \( i \);

\( G_i \): probability that the needed part type is in group \( i \);

\( H_{i1} \): holding cost per unit year of group \( i \) at the kit level;

\( H_{i2} \): holding cost per unit year of group \( i \) at the depot level;

and the following decision variables:

\( K_i \): number of different group \( i \) part types to be carried in the kit; and

\( S_i \): number of different group \( i \) part types to be stocked in the depot.

We assume that at most one unit of each part type is carried at each location.

For mathematical tractability, we assume that each part within a group is equally likely to be needed in a repair. Therefore, if \( K_i \) parts are randomly chosen from among the \( n_i \) part types in group \( i \) then,

\[
P(\text{part } j \text{ is in kit} / \ K_i \text{ group } i \text{ parts are in the kit, part } j \text{ is in group } i ) = \frac{K_i}{n_i}.
\]

Although we are assuming the \( K_i \) parts are chosen randomly, in the actual implementation, additional information about the particular customers being supported from the kit may be available to choose the parts in a better way and achieve a higher fill rate. Therefore, we can use \( K_i/n_i \) as a lower bound on the achievable fill rate.

### 4.3. Model Development

Given this new definition of \( K_i \) and \( S_i \), we must now develop a new formula for \( p_3(K) \). We consider group \( i \) of the kit as a queueing system with \( K_i \) servers. The arrivals are again Poisson with rate \( \lambda p_1 p_2 G_i \).

The service times are deterministic and equal to \( \tau_{N1} \), the normal
replenishment time for the parts. The difference between this system and
the previous ones is the presence of balking. Balking occurs when arrivals
to the system do not enter the system for some reason. Balking occurs in
two ways in this system. First, if the needed part is not one of the $K_i$ parts
which are normally stocked in the kit, then the part must be ordered for
quick delivery or emergency delivery. The probability of this event is

$$(1 - \frac{K_i}{n_i})$$

The other way in which balking can happen is if the needed part
is usually carried in the kit, but another demand has used the part and the
replenishment part has not arrived. This probability is dependent on the
number of busy servers. Therefore, the probability of balking, given that
there are $s$ servers busy is $b_{i,s} = (1 - \frac{K_i - s}{n_i}), \quad 0 \leq s \leq K_i$. This

M/D/K_i/K_i system with balking is very difficult to analyze, so we will
approximate it with an M/M/K_i/K_i system with balking, with $\mu_B = (1/\tau_{n1})$.
The memoryless property of the exponential service times allows us to
analyze the system as follows.

Define $\pi_s$ as the long run probability that $s$ servers are busy,

$$0 \leq s \leq K_i.$$ 

Also define $\lambda_B = \frac{\lambda \cdot p_1 \cdot p_2 \cdot G_i \cdot \tau_{N1}}{X},$ the average demand
over the replenishment lead time that a field engineer would see for group $i.$
Then the flow balance equations for this system are:

\[ 0 = -\frac{\lambda_B}{n_i} K_i \pi_0 + \mu_B \pi_1 ; \]

for \( 1 \leq s \leq K_i - 1 \)

\[ 0 = -\left(\frac{\lambda_B(K_i-s)}{n_i} + s\mu_B\right)\pi_s + \mu_B(s+1)\pi_{s+1} + \frac{\lambda_B(K_i-s+1)}{n_i} \pi_{s-1} ; \]

\[ 0 = -(K_i \mu_B)\pi_{K_i} + \frac{\lambda_B}{n_i} \pi_{K_i-1} ; \quad \text{and} \]

\[ 1 = \pi_0 + \pi_1 + \ldots + \pi_{K_i} . \]

The solution to these equations is:

\[ \pi_0 = \left(1 + \frac{\lambda_B}{\mu_B n_i}\right)^{-K_i} ; \quad \text{and} \]

\[ \pi_s = \frac{K_i!}{s!(K_i-s)!} \left(\frac{\lambda_B}{\mu_B n_i}\right)^s \pi_0, \quad 1 \leq s \leq K_i . \]

Now we calculate \( p_{3i}(K_i) \), the probability of not having the part required, given the part is in group \( i \) and \( K_i \) parts from group \( i \) are stocked.

\[ p_{3i}(K_i) = \text{Prob(balking)} \]

\[ = 1 - \sum_{s=0}^{K_i} \left(\frac{K_i-s}{n_i}\right) \pi_s . \]

Similarly, we can calculate \( f(K_i,S_i) \). If we redefine

\[ \lambda_B = \lambda \ p_1 \ p_2 \ G_i \ p_{3i}(K_i) \ \tau_{N2} \] and \[ \mu_B = \frac{1}{\tau_{N2}} , \] then \( f(K_i,S_i) \) has the same
form as $p_3(K_1)$ with new values of the $\pi$ variables calculated using the new values of $\lambda_B$ and $\mu_B$. We solve this expanded model using the Lagrangian–based approach described earlier in this chapter.

4.4. Output Analysis of General Model Behavior

This section explores the model developed in Section 4.1. The motivation behind this section is to demonstrate that the model does behave in certain ways that match our intuitive understanding of field service support systems.

4.4.1. Increasing the Budget Increases the Optimal Number of Field Engineers

Figure 19 is a plot of the "controllable time" versus the total cost of the system and Figure 20 is a plot of the optimal number of field engineers versus the total cost of the system. The controllable time is the total customer service time minus the constant portions of the customer service time. In other words, the controllable time is the random part of the total service time which is directly affected by the number of field engineers, the kit compositions, and the depot composition. In this plot, we see that as the budget is increased, parts are added to the kits and to the depot stock until a critical budget level is reached. At this critical level, the inventory is reduced and another field engineer is added. In Figures 19 and 20, the critical levels are $605,217$ and $671,836$. 
Controllable Time versus Total Cost

Data Set K3-U1: Urban Environment

Total Time
First Queue Time
Part Acquisition Time

Figure 19: Controllable Time versus Total Cost for K3-U1
Figure 20: Optimal Number of Field Engineers versus Total Cost for K3-U1
When the field engineer is added, the first queue time decreases since there are more engineers to respond to the request calls. Also, the part acquisition time increases due to the reduced inventory carried in the kit. However, the optimal number of field engineers is not always monotonically increasing as demonstrated in Section 4.4.5.

The clustering of data points in the neighborhood of the critical points is a consequence of our Lagrangian multiplier technique. We varied the multiplier in an attempt to identify the critical budget levels. An additional reason for the clustering is the decision to output only those points whose objective function value was different than the previous point’s objective function value.

4.4.2. Inventory Levels Vary as Budget Increases

In Figure 21, we plot the optimal kit coverage and depot coverage, respectively, versus the total cost. The kit (depot) coverage for part group j is defined as the number of parts in group j carried in the kit (depot) divided by the total number of parts in group j. Each of the nine part groups is plotted, but only groups K3, K6, and K9 have a kit coverage that is significantly less than 100% kit coverage. When the number of field engineers increases, the kit coverage drops and the depot coverage drops. Note, however, that different parts are dropped from the kits than those dropped from the dept. This corresponds to our intuitive understanding that if a part is not carried in the kit, then the depot should be fully stocked with that part. Also note that the expensive parts are the ones dropped from the
kits, which frees a large portion of the holding and replenishment costs to be reallocated to field engineer costs.
Figure 21: Kit Coverage and Depot Coverage versus Total Cost for K3–U1
4.4.3. Urban and Rural Environments Have Different Optimal Solutions

In this section, we compare the output analysis for a rural environment and an urban environment with the same part group data. In Figure 22, the optimal number of field engineers is plotted for data set K4-U2 and data set K4-R2. We can see that the urban environment always has a greater or equal number of field engineers than the rural environment. It would seem that the rural environment would need more field engineers to keep the queue stable since the expected service time is longer. However, the call arrival rates were adjusted in both data sets to reduce this tendency by setting the arrival rates such that the minimum number of field engineers to keep each system stable is nine.

Figures 23 and 24 are the kit and depot coverage plots for the data sets K4-R2 and K4-U2, respectively. We see in Figure 23 that the rural environment always has near 100% coverage for each part. This is due to the long travel times and delivery times incurred if the part is not in the kit. In Figure 24, we see that the rural environment has more inventory at the depot than the urban environment for a large set of budget levels. Overall, these three figures demonstrate the relative advantage of investing in inventory in rural environments and in field engineers in urban environments, provided sufficient engineers are in place to keep the request queues stable.
Figure 22: Optimal Number of Field Engineers for K4–U2 and K4–R2
Figure 23: Kit Coverage versus Total Cost for K4-U2 and K4-R2
Figure 24: Depot Coverage versus Total Cost for K4–U2 and K4–R2
4.4.4. Different Part Groupings Yield Different Optimal Solutions

The next group of plots compare two urban environments with different part groupings. All of the input parameters are the same except for the number of parts in each group. The input data for the plots is given in Appendix A.

Figure 25 shows the optimal number of field engineers, Figure 26 shows the optimal kit coverage, and Figure 27 shows the optimal depot coverage. From these plots, we see that K5–U2 has more field engineers and less inventory than K4–U2 for all budget levels. The reason that K5–U2 can achieve high customer service with fewer parts in the inventory is because in kit type 5, 100 parts account for 75% of the demand; while in kit type 4, it takes 750 parts to account for 75% of the demand. These plots illustrate the importance of carefully creating the part groupings, since the different groupings may yield different optimal solutions.
Figure 25: Optimal Number of Field Engineers versus Total Cost for K4–U2 and K5–U2
Figure 26: Optimal Kit Coverage versus Total Cost for K4–U2 and K5–U2
Figure 27: Optimal Depot Coverage versus Total Cost for K4-U2 and K5-U2
4.4.5. Duality Gaps May Cause Strange Behavior

We now demonstrate one of the inherent problems in solving integer, non-linear programs with approximate Lagrangian relaxation techniques. Figure 28 is the controllable time for data set K6–U1. Note that there are no data points between $700,000 and $850,000. Now consider Figure 29, the optimal number of field engineers plot. Notice the decrease in the optimal number of field engineers corresponding to this range of total cost. For budget levels up to $863,774, the budget is too small to stock part K9 in each field engineer’s kit. Once this level is reached, however, we can afford to add this part group to the kits if we drop one field engineer also. For budget levels even higher, the optimal strategy is to have as many field engineers for whom we can afford to provide full kits. Figure 30 gives the kit coverage and depot coverage plots for this data set.

When solving the two echelon, large number of parts model for this data set, the gamma values, that is, the Lagrangian multipliers for the budget constraint, in the range $[0.0001000, 0.0001030]$ produced the full kits; while values in the range $[0.0001083, 0.0001090]$ produced the kits without part group K9. When gamma was in the range $[0.0001031, 0.0001082]$, however, the true arrival rate and the estimated arrival rate did not converge. This strange behavior is caused by the duality gap in the integer program.
Controllable Time versus Total Cost
Data Set K6-U1 : Urban Environment

Figure 28: Controllable Time Plot for K6–U1
Figure 29: Optimal Number of Field Engineers for K6–U1
Figure 30: Kit Coverage and Depot Coverage for K6–U1
4.5. Strategic Planning Questions Addressed by the Model

This section explores the use of the two echelon, large number of parts model in strategic planning. First, we explore the effect of changing the available budget level. Next, we analyze the breaking repairs parameter, \( p_4 \). Finally, we compare the output from several different delivery costs and delivery times.

4.5.1. Small Budget Increases May Yield Large Benefits

Field service support managers are very interested in the question: "What change in customer service time would result from a small increase in the budget?" This question can be answered easily by the Controllable Time Plot. For example, Figure 31 is the Controllable Time Plot for data set K3–U1. We see that increasing the budget from $605,217 to $620,112 would result in the controllable customer service time decreasing from 1.93 hours to 0.63 hours, which is a large benefit for a small increase. This model's output could be a strong argument for changes in the current budget level. On the other hand, we also observe that increasing the budget from $564,401 to $605,217 would decrease the controllable customer service time from 1.99 hours to 1.93 hours, which is virtually no benefit for a large increase in the budget level. We conclude that the benefit of increasing the budget is highly dependent on the current budget level.
Controllable Time versus Total Cost

Data Set K3-U1: Urban Environment

- Total Time
- First Queue Time
- Part Acquisition Time

Figure 31: Controllable Time Plot for K3-U1
4.5.2. Effect of the Breaking Repairs Policy Parameter

The field service support planning model which we have developed has many input parameters, most of which can be estimated from historical data. The probability of breaking a repair, given that the field engineer does not have the needed part, $p_4$, is a management policy parameter. We experimented with a range of values from 0.1 to 0.9 for this parameter with the data set K5–U2. The value of $p_4$ affected the part acquisition time and the initial queue wait time; specifically, higher values of $p_4$ resulted in higher times. However, the optimal number of field engineers, the optimal kit composition, and the optimal depot composition did not change. We conclude that for this data set it is important to estimate $p_4$ close to the true value in order to know the true value of the controllable time; yet, even if the estimate is not exact, then the optimal solution will remain the same. The effect of the $p_4$ parameter for a particular data set depends on the utilization level of the field engineers and the relative values of the travel time, the repair time, and the quick delivery time.

4.5.3. Effect of the Emergency and Quick Delivery Costs in Urban Environments

Field service managers have many options to explore in their efforts to improve their field service support system. One option is to decrease the part delivery times from the local depot and the central warehouse. This may be accomplished in some cases by replacing a manual part picking system with an automated storage and retrieval system (ASRS). Although the ASRS would require a large initial capital outlay, the annual operating
costs would be much lower, decreasing the cost of a quick delivery or an emergency delivery. In addition, an ASRS would take less time to retrieve the needed parts, thereby reducing the part delivery times. The ASRS would also be used to pick the normal replenishment shipments, reducing the cost and the time associated with them.

We experimented with 4 sets of part delivery costs and times, given below in Table 11. The other parameters were from data set K5–U2.

<table>
<thead>
<tr>
<th>Data Set #</th>
<th>( c_q )</th>
<th>( c_e )</th>
<th>( \tau_q )</th>
<th>( \tau_e )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.16</td>
<td>0.30</td>
<td>2.0</td>
<td>12.0</td>
</tr>
<tr>
<td>2</td>
<td>0.16</td>
<td>0.18</td>
<td>2.0</td>
<td>8.0</td>
</tr>
<tr>
<td>3</td>
<td>0.08</td>
<td>0.12</td>
<td>1.0</td>
<td>6.0</td>
</tr>
<tr>
<td>4</td>
<td>0.04</td>
<td>0.12</td>
<td>0.5</td>
<td>6.0</td>
</tr>
</tbody>
</table>

Figure 32 is a plot of the total controllable time versus the total cost for each of the above data sets. We first note that each data set has the same basic shape for its plot. Therefore, the main effect of these parameters is to shift the location of the plot with respect to the x axis and the y axis. We also note that due to the step function nature of the plots, there are several ranges of total cost (for example, $640,000 to $675,000) which do not change the customer service time. If a field service manager were currently operating at a total cost of $675,000 with data set 2, this manager could consider two different options for improving the customer service level. The first option is to increase the budget to step down on the data set 2
curve, thereby improving the total controllable time from 0.941 hours to 0.810 hours at a cost of $6,155. The second option is to improve the part delivery costs and times by upgrading the central warehouse and local depot. Although this may require a large initial outlay, the total annual system costs could be cut by $42,530 and achieve a reduction in the controllable time from 0.941 hours to 0.554 hours. Since the number of field engineers is not different for these options, the difficulties of altering the size of the field engineer staff would not be encountered. The actual choice of the first or second option would depend on the amount of initial capital available for investment, the cost of capital, the priorities of the management, and other tangible and intangible factors. This model can be very useful in evaluating these kinds of strategic planning options.
Figure 32: Controllable Time Plot for Various Delivery Cost Parameters
CHAPTER 5. ADDITIONAL MODELS

5.1. No Emergency Supply

In this section, we consider modifying the model of Chapter 3 to eliminate emergency deliveries. It is independent of the large number of parts model developed in Chapter 4. For two echelon field support systems with no emergency deliveries, when the local depot runs out of stock, the demands are backordered. Since the demand rates for spare parts are very low and yet we wish to maintain high fill rates, we assume that the depot uses a one-for-one replenishment policy. We continue to assume that the field engineers do not pool their inventories. The individual field engineer kit can still be analyzed as an M/D/K/K queue as in section 3.6; therefore, $p_{3i}(K_i)$ has a truncated Poisson distribution.

The depot fill rate must be analyzed differently. To begin, we let

$$\bar{\lambda}_i = \lambda_1 p_1 p_2 p_{3i}(K_i) G_i.$$ 

That is, $\bar{\lambda}_i$ is the arrival rate of demands at the local depot for part i. These demands form a Poisson process. Since there is no emergency supply, an incoming demand may see an out of stock situation and be backordered. An arriving demand for part i which sees b backorders for part i must wait for b+1 parts to be delivered. Since each part delivery was triggered by a demand arrival, the interarrival times of part deliveries are independent, exponential random variables. In addition, the time which a demand must wait until a part is available, $W(K_i,S_i)$, is always less than or equal to $\tau_{N2}$. Therefore, the conditional distribution of $W(K_i,S_i)$, given b backorders is a truncated gamma distribution. Denote
the gamma density with parameters $\bar{\lambda}$ and $b+1$ by $f_G(x;\bar{\lambda},b+1)$. We calculate the expected value of $W(K_i,S_i)$ as follows, dropping the subscript $i$ for convenience.

$$E[W(K,S)/b] = \frac{\int_0^{\tau_{n2}} x f_G(x;\bar{\lambda},b+1) \, dx}{\int_0^{\tau_{n2}} f_G(x;\bar{\lambda},b+1) \, dx}.$$

Evaluating the numerator, we have:

$$\int_0^{\tau_{n2}} x f_G(x;\bar{\lambda},b+1) \, dx = \int_0^{\tau_{n2}} \bar{\lambda}^{-(b+1)} x^{b+2} e^{\frac{-x}{\bar{\lambda}}} \frac{1}{b!} \, dx$$

$$= \bar{\lambda}^{-(b+1)} \left[ \frac{b+2}{b!} \sum_{r=0}^{b+2} (-1)^r \frac{(b+2)!}{(b+2-r)!} x^{b+2-r} (-\bar{\lambda})^{r+1} \right]_{0}^{\tau_{n2}}$$

$$= \bar{\lambda}^2(b+1)(b+2) - \bar{\lambda}^{-(b+1)}(b+1)(b+2) e^{\frac{-\tau_{n2}}{\bar{\lambda}}} \sum_{r=0}^{b+2} (\bar{\lambda})^r (\tau_{n2})^{b+2-r} \frac{1}{(b+2-r)!}.$$
Evaluating the denominator, we have:

\[
\int_0^{\tau_{n2}} f_G(x) \, dx = \int_0^{\tau_{n2}} \chi^{-b+1} e^{\frac{x}{\lambda}} \frac{1}{b!} \, dx
\]

\[
= \frac{1}{b!} \left[ \sum_{r=0}^{b+1} \frac{(b+1)! \chi^{b+1-r}}{(b+1-r)!} (-\chi)^{r+1} e^{\frac{\tau_{n2}}{\lambda}} \frac{1}{b+1-r} \right]_{0}^{\tau_{n2}}
\]

\[
= \lambda(b+1) - \lambda^{-b+1} (b+1)e^{-\frac{\tau_{n2}}{\lambda}} \sum_{r=0}^{b+1} \left[ \frac{1}{(b+1-r)!} \right].
\]

Therefore,

\[
\lambda(b+2) - \lambda^{-b+1}(b+2) e^{-\frac{\tau_{n2}}{\lambda}} \sum_{r=0}^{b+2} \frac{(\lambda)^{r+1}(\lambda_{n2})^{b+2-r}}{(b+2-r)!} \frac{1}{(b+1-r)!}
\]

\[
E[W(K,S) / b] = \frac{1 - \lambda^{-b+1} e^{-\frac{\tau_{n2}}{\lambda}} \sum_{r=0}^{b+1} (\lambda)^{r+1}(\lambda_{n2})^{b+1-r} \frac{1}{(b+1-r)!}}{1 - \lambda^{-b+1} e^{-\frac{\tau_{n2}}{\lambda}} \sum_{r=0}^{b+1} (\lambda)^{r+1}(\lambda_{n2})^{b+1-r} \frac{1}{(b+1-r)!}}.
\]

Since the depot uses a one-for-one ordering policy, the number of backorders has a Poisson distribution [11]:

\[
\text{Prob} \{ \text{b backorders} \} = \frac{e^{-\lambda_{n2}} (\lambda_{n2})^{b+S}}{(b+S)!}.
\]

Therefore,

\[
E[W(K,S)] = \sum_{b=0}^{\infty} \frac{e^{-\lambda_{n2}} (\lambda_{n2})^{b+S}}{(b+S)!} E[W(K,S) / b].
\]
In our expression for the expected total customer service time from section 3.5.1, we now substitute \( E[W(K,S)] + \tau_Q \) for the emergency delivery time.

\[
E[C.S.T.] = D_1 + p_1 (Q_L + T + D_2) + p_1 (1 - p_2) R_A + p_1 p_2 R_P
+ p_1 p_2 p_3(K) f(K,S) [E[W(K,S)] + \tau_Q + Q_H + T]
+ p_1 p_2 p_3(K) (1 - f(K,S)) [\tau_Q + p_4(Q_H + T)] .
\]

Although this model has a nice theoretical development, the dependence of \( W(K,S) \) on \( K \) and \( S \) makes it very difficult to solve optimally. We propose a heuristic search with Lagrangian relaxation. A solution algorithm for this problem has not been implemented and we do not pursue this model any further.

5.2. Job Completion Rate

In Chapter 1, with reference to the spare parts kitting problem, we mentioned several papers which concentrate on the job completion rate instead of the customer service time and which consider only the holding cost of the kits. In Chapter 2, we showed that for a single echelon system with a fixed number of field engineers, maximizing the job completion rate is equivalent to minimizing the total customer service time. In this section, we will expand the model of Graves to include the cost of not having the part as well as the holding cost. First, we explain Graves' model in more detail and then we shall explore our expanded model.
Graves’ model minimizes the holding cost of the kit, subject to achieving a minimum job completion rate, \( \rho \). He assumes that the spare parts kits are restocked between jobs and that at most one unit of each part is needed per repair. He does allow multiple parts to be required for a repair. Therefore, the probability of having the needed parts in the kit has a product form. The model is a zero–one integer program with a linear objective function and a non–linear constraint.

\[
(M8) \quad \min_{K_1, \ldots, K_N} \sum_{i=1}^{N} K_i H_{i1}
\]

subject to:

\[
\prod_{i=1}^{N} (1 - G_i)^{1-K_i} \geq Q
\]

\[
K_i \in \{0, 1\} \quad i = 1, \ldots, N
\]

In order to solve this model, Graves takes the logarithm of the job completion rate to transform it into a linear constraint. This reduces the problem to a binary knapsack problem which is easy to solve, although there are no polynomial time algorithms.

Graves’ model fails to directly capture the fact that if the parts are not available for the field engineer, the company incurs additional cost in terms of field engineer time, emergency shipment cost, customer good will, etc. This interaction must be evaluated through different levels of \( \rho \).

We will make the same assumptions on the restocking between jobs and the multiple parts per repair. However, we will directly account for the
additional cost of not having the needed parts. Let $\theta$ denote this cost per broken job. Following the approach established in Chapters 2 and 3, we will maximize the job fill rate subject to a budget constraint.

(M9) \[
\min_{K_1, \ldots, K_N} \prod_{i=1}^{N} (1 - G_i)^{1-K_i} \\
\text{subject to: } \sum_{i=1}^{N} K_i H_{i1} + (1 - \prod_{i=1}^{N} (1 - G_i)^{1-K_i}) \lambda \theta \leq B; \\
K_i \in \{0, 1\}, \quad i = 1, \ldots, N.
\]

The specification of $\theta$ is a difficult task. It should incorporate several types of costs. The field engineers will spend more time travelling if the parts are not available, which increases the annual cost of a field engineer. The parts must be picked from inventory at the central warehouse and shipped to the customer’s location. In addition, breaking the repairs may affect the customer’s good will.

Solving this non-linear integer program is more difficult than solving Graves’ model, since the constraint cannot be linearized. We use implicit enumeration to solve it. We begin the algorithm with a full kit. If the full kit is feasible, then it is optimal; otherwise we must decide on which part type to branch. This decision is made based on the benefit–cost ratio of the parts, calculated as follows, where $K$ is the solution at the current branch:

(17) \[
\frac{\text{Incremental Benefit of Part } j}{\text{Incremental Cost of Part } j} = \frac{G_j \prod_{i \neq j} (1 - G_i)^{1-K_i}}{H_j - \lambda \theta G_j \prod_{i \neq j} (1 - G_i)^{1-K_i}}
\]
As each part is branched upon, the resulting leaves may be fathomed by: 1) feasibility, 2) infeasibility, or 3) domination. A depth-first search will yield a feasible solution quickly, which can then be used to prune additional branches by domination. In addition, each time a variable is set to one, the new node has the same cost, value, and benefit/cost ratios as the node from which we are branching. Therefore, we can collapse nodes and set two variable values on such branches. This solution method is guaranteed to find the optimal solution.

We now solve an example of this model to demonstrate the method. We have 9 different part types with the probabilities and costs given in the following table. Complete enumeration of all possible kit compositions would require evaluating $2^9$ solutions.

<table>
<thead>
<tr>
<th>Part</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_i$</td>
<td>0.02</td>
<td>0.02</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.30</td>
</tr>
<tr>
<td>$H_i$</td>
<td>5</td>
<td>10</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>5</td>
</tr>
</tbody>
</table>
Figure 33: Benefit / Cost Ratio Model – Implicit Enumeration Example
The following table gives the benefit/cost ratios for each node in the example. Once the value of a variable has been set by branching, then the value of the ratio is no longer important. We also note that a negative ratio indicates that if that part is removed from the kit, then the cost of the new kit will be higher than the kit with the part and the value will be lower. Therefore, we do not branch on variables with negative ratios.

Table 13. Benefit / Cost Ratios for Implicit Enumeration Example

<table>
<thead>
<tr>
<th>Node</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>K1</td>
<td>0.004</td>
<td>0.004</td>
<td>—</td>
<td>0.004</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>K2</td>
<td>0.002</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>K3</td>
<td>-0.200</td>
<td>-0.21</td>
<td>-0.218</td>
<td>-0.273</td>
<td>-0.209</td>
<td>-0.273</td>
<td>-0.273</td>
</tr>
<tr>
<td>K4</td>
<td>0.029</td>
<td>0.028</td>
<td>0.027</td>
<td>0.024</td>
<td>0.028</td>
<td>0.024</td>
<td>—</td>
</tr>
<tr>
<td>K5</td>
<td>0.012</td>
<td>0.012</td>
<td>0.011</td>
<td>—</td>
<td>0.012</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>K6</td>
<td>-0.100</td>
<td>-0.101</td>
<td>-0.102</td>
<td>-0.107</td>
<td>-0.101</td>
<td>-0.107</td>
<td>-0.115</td>
</tr>
<tr>
<td>K7</td>
<td>0.100</td>
<td>0.095</td>
<td>0.091</td>
<td>0.075</td>
<td>0.095</td>
<td>0.075</td>
<td>0.0606</td>
</tr>
<tr>
<td>K8</td>
<td>0.029</td>
<td>0.028</td>
<td>0.027</td>
<td>0.024</td>
<td>0.028</td>
<td>0.024</td>
<td>0.021</td>
</tr>
<tr>
<td>K9</td>
<td>0.600</td>
<td>0.498</td>
<td>0.425</td>
<td>0.257</td>
<td>0.498</td>
<td>0.257</td>
<td>0.167</td>
</tr>
</tbody>
</table>

This benefit / cost ratio model has the advantage over Graves' model in that it incorporates the cost of not having the part. However, some of the assumptions which are made in both of these models, such as the kits being restocked between jobs, are inappropriate for many field support systems. We believe that the models considered in the earlier chapters are more appropriate for most field support systems.
CHAPTER 6. DISSERTATION SUMMARY AND AREAS FOR FURTHER RESEARCH

6.1. Dissertation Summary

In this dissertation, we have accomplished three primary goals: 1) the development of a classification scheme for field service support systems; 2) the construction of a set of mathematical models which capture the tradeoffs between investment in field engineering staff and investment in spare parts inventory; and 3) the demonstration of how these models can aid in answering strategic management questions.

The classification scheme for field service support systems identifies the distinguishing features of a support system for planning purposes with respect to three main categories: the customer base; the field engineer staff; and the spare parts inventory system. This classification scheme provides a framework for organizing the available research literature on field service support systems. It also provides a way to systematically compare two instances of field support in order to determine how similar they are and consequently, whether they may be analyzed with the same type of model.

We argued that the substitution of investment in inventory for the hiring of field support staff is an important but neglected planning issue in managing a field support system. One reason for its neglect is the complexity of the interactions between inventory stocking levels, field engineer utilization, and customer downtime. To demonstrate that these interactions can be presented quantitatively with tractable mathematical
models we developed a series of models of field support that focus on the tradeoffs between investment in staff and investment in spare parts inventory. We also developed algorithms to find near-optimal staff and inventory levels. Our goal was to carry this modeling activity to the point of providing tractable models of realistic complexity that could be used by field support planners to explore the inventory–staff tradeoff. To this end, the most sophisticated of the models we developed considers two echelons of spare parts inventory: the field engineer spare parts kit and the local spare parts depot. It also allows for a realistically large number of part types by grouping the parts based on cost and the probability of being needed for a repair.

The model could be used to assist planners in forming a general allocation of their field support budget between inventory and personnel. To illustrate its potential use, we applied the model in a number of hypothetical field service environments to answer a variety of strategic management questions. In particular, we examined the benefit, in terms of customer service, of increasing the total budget of the field service function. We considered the optimal allocation of the inventory budget among the different part types and stocking echelons. We also considered the effect of automation in the spare parts delivery system on overall system performance. For each question, we noted how the answer was dependent upon the particular field service environment being studied. We believe that the results of this dissertation can be used by field support system
planners to improve the quality of their economic decision making. In the process of developing these models, we have identified many areas of further research on this topic, which are described in the next section.

6.2. Areas For Further Research

Field support systems are rich with possibilities for further research. In this section we briefly mention several of these areas. The following areas could be explored using the basic approach of this dissertation of examining the various realizations for a field service request. First, modeling the variance of the customer service time would give the field service manager more information about the impact of his or her decisions on customer service. This information would allow the manager to choose among several options which yield the same average performance, but have different variances of performance. Second, modeling the travel times in detail would remove the constant time assumption and would result in a model with higher accuracy. Next, a model which would allow the field engineers to pool their spare parts kits would reflect the great flexibility available in many field support systems with multiple field engineers. The pooling of parts would reduce the necessary investment in spare parts inventory at the field engineer level and simultaneously increase the customer service level. The effect of this flexibility would be especially strong in urban environments. Finally, a model which could analyze alternative dispatch policies, instead of assuming a first-in-first-out policy,
would connect the minute-by-minute dispatch decisions with the strategic planning decisions, providing a more complete picture of the system.

Other areas of further research would probably require a completely new approach to modeling. First, many service companies offer a range of service contracts, which guarantee different levels of customer service, usually measured in terms of response time. The pricing of these service level contracts is difficult in the competitive market of field service. Research in this area would include integrating customer service models, marketing studies, and cost accounting methodologies. Second, many types of products which are supported by field engineers are sophisticated equipment which have some level of self-diagnosis when a failure occurs. This advance information can reveal which spare parts are likely to be needed and how severe the failure is. Field service managers could benefit from having a model which can incorporate this type of advance information. Next, the assignment of customers to field service territories is an area of research which has not been explored fully. Given a map which locates each customer by a pinpoint, the field service manager must divide the customers into field service regions. This division affects both the customer service level which may be achieved and the management structure necessary to support the regions. Therefore, a model which gives a systematic way to assign customers to regions would be a useful tool for managers. Finally, field support can be considered as a hierarchy of models from strategic planning to real time dispatching. A hierarchy of models
which work together could provide a comprehensive view of field service. As the competitive nature of field service intensifies, the need grows to extend the science base of the strategic, planning, and operational decisions made to provide this service.
APPENDIX A: INPUT DATA FOR MODELS
Environment Parameters

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>U1</th>
<th>U2</th>
<th>R2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_{n1}$</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>$\tau_{n2}$</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>$\tau_e$</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>$\tau_q$</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>T</td>
<td>6</td>
<td>0.5</td>
<td>0.5</td>
<td>5</td>
</tr>
<tr>
<td>$R_A$</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>$R_p$</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>1</td>
<td>3.5</td>
<td>3.5</td>
<td>1</td>
</tr>
<tr>
<td>$c_{FE}$</td>
<td>50k</td>
<td>50k</td>
<td>50k</td>
<td>50k</td>
</tr>
<tr>
<td>$c_{n1}$</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>$c_{n2}$</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>$c_e$</td>
<td>0.40</td>
<td>0.3</td>
<td>0.12</td>
<td>0.12</td>
</tr>
<tr>
<td>$c_q$</td>
<td>0.20</td>
<td>0.15</td>
<td>0.08</td>
<td>0.08</td>
</tr>
<tr>
<td>$p_2$</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
<td>0.8</td>
</tr>
</tbody>
</table>
Kit Type 1: Pareto Cost and Probability

<table>
<thead>
<tr>
<th>Part Number</th>
<th>$G_i$</th>
<th>$H_{i1}$</th>
<th>$H_{i2}$</th>
<th>$n_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.02</td>
<td>0.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.02</td>
<td>2.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.01</td>
<td>8.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.07</td>
<td>0.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.07</td>
<td>2.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.06</td>
<td>8.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.25</td>
<td>0.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.25</td>
<td>2.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.25</td>
<td>8.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Kit Type 2: Pareto Cost and Probability for Two Echelons

<table>
<thead>
<tr>
<th>Part Number</th>
<th>$G_i$</th>
<th>$H_{i1}$</th>
<th>$H_{i2}$</th>
<th>$n_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.02</td>
<td>0.2</td>
<td>0.15</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.02</td>
<td>1.0</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.01</td>
<td>4.0</td>
<td>3.00</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.07</td>
<td>0.2</td>
<td>0.15</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.07</td>
<td>1.0</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.06</td>
<td>4.0</td>
<td>3.00</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.25</td>
<td>0.2</td>
<td>0.15</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.25</td>
<td>1.0</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.25</td>
<td>4.0</td>
<td>3.00</td>
<td></td>
</tr>
</tbody>
</table>
Kit Type 3: Pareto Cost and Probability for Two Echelons and Large Number of Parts

<table>
<thead>
<tr>
<th>Part Number</th>
<th>$G_i$</th>
<th>$H_{i1}$</th>
<th>$H_{i2}$</th>
<th>$n_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.02</td>
<td>0.04</td>
<td>0.030</td>
<td>35</td>
</tr>
<tr>
<td>2</td>
<td>0.02</td>
<td>0.10</td>
<td>0.075</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>0.01</td>
<td>0.40</td>
<td>0.300</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>0.07</td>
<td>0.04</td>
<td>0.030</td>
<td>140</td>
</tr>
<tr>
<td>5</td>
<td>0.07</td>
<td>0.10</td>
<td>0.075</td>
<td>40</td>
</tr>
<tr>
<td>6</td>
<td>0.06</td>
<td>0.40</td>
<td>0.300</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>0.25</td>
<td>0.04</td>
<td>0.030</td>
<td>525</td>
</tr>
<tr>
<td>8</td>
<td>0.25</td>
<td>0.10</td>
<td>0.075</td>
<td>150</td>
</tr>
<tr>
<td>9</td>
<td>0.25</td>
<td>0.40</td>
<td>0.300</td>
<td>75</td>
</tr>
</tbody>
</table>
Kit Type 4: Normal Cost and Probability for Two Echelons and Large Number of Parts

<table>
<thead>
<tr>
<th>Part Number</th>
<th>$G_i$</th>
<th>$H_{i1}$</th>
<th>$H_{i2}$</th>
<th>$n_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.02</td>
<td>0.04</td>
<td>0.030</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>0.02</td>
<td>0.10</td>
<td>0.075</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>0.01</td>
<td>0.40</td>
<td>0.300</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>0.07</td>
<td>0.04</td>
<td>0.030</td>
<td>40</td>
</tr>
<tr>
<td>5</td>
<td>0.07</td>
<td>0.10</td>
<td>0.075</td>
<td>140</td>
</tr>
<tr>
<td>6</td>
<td>0.06</td>
<td>0.40</td>
<td>0.300</td>
<td>40</td>
</tr>
<tr>
<td>7</td>
<td>0.25</td>
<td>0.04</td>
<td>0.030</td>
<td>150</td>
</tr>
<tr>
<td>8</td>
<td>0.25</td>
<td>0.10</td>
<td>0.075</td>
<td>450</td>
</tr>
<tr>
<td>9</td>
<td>0.25</td>
<td>0.40</td>
<td>0.300</td>
<td>150</td>
</tr>
</tbody>
</table>
Kit Type 5: Normal Cost, Low Probability for Two Echelons with Large Number of Parts

<table>
<thead>
<tr>
<th>Part Number</th>
<th>$G_i$</th>
<th>$H_{i1}$</th>
<th>$H_{i2}$</th>
<th>$n_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.02</td>
<td>0.04</td>
<td>0.030</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td>0.02</td>
<td>0.10</td>
<td>0.075</td>
<td>400</td>
</tr>
<tr>
<td>3</td>
<td>0.01</td>
<td>0.40</td>
<td>0.300</td>
<td>200</td>
</tr>
<tr>
<td>4</td>
<td>0.07</td>
<td>0.04</td>
<td>0.030</td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td>0.07</td>
<td>0.10</td>
<td>0.075</td>
<td>50</td>
</tr>
<tr>
<td>6</td>
<td>0.06</td>
<td>0.40</td>
<td>0.300</td>
<td>25</td>
</tr>
<tr>
<td>7</td>
<td>0.25</td>
<td>0.04</td>
<td>0.030</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>0.25</td>
<td>0.10</td>
<td>0.075</td>
<td>80</td>
</tr>
<tr>
<td>9</td>
<td>0.25</td>
<td>0.40</td>
<td>0.300</td>
<td>10</td>
</tr>
</tbody>
</table>
Kit Type 6: Normal Cost, High Probability for Two Echelons with Large Number of Parts

<table>
<thead>
<tr>
<th>Part Number</th>
<th>$G_i$</th>
<th>$H_{i1}$</th>
<th>$H_{i2}$</th>
<th>$n_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.02</td>
<td>0.04</td>
<td>0.030</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>0.02</td>
<td>0.10</td>
<td>0.075</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>0.01</td>
<td>0.40</td>
<td>0.300</td>
<td>40</td>
</tr>
<tr>
<td>4</td>
<td>0.07</td>
<td>0.04</td>
<td>0.030</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>0.07</td>
<td>0.10</td>
<td>0.075</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>0.06</td>
<td>0.40</td>
<td>0.300</td>
<td>120</td>
</tr>
<tr>
<td>7</td>
<td>0.25</td>
<td>0.04</td>
<td>0.030</td>
<td>80</td>
</tr>
<tr>
<td>8</td>
<td>0.25</td>
<td>0.10</td>
<td>0.075</td>
<td>80</td>
</tr>
<tr>
<td>9</td>
<td>0.25</td>
<td>0.40</td>
<td>0.300</td>
<td>640</td>
</tr>
</tbody>
</table>
(* SINGLE  - Model of Chapter 2 *)
(* Created by: Alisha Waller *)
(* Purpose: Find the optimal # FEs and optimal inventory *)
(* p4 <> 0 and single echelon inventory *)
(* M/M/x with multiple priority classes *)
(* Last Altered: 3/28/91 *)

(* ------------------------------------------------------------------------- *)
program SINGLE(input,output,Outfile1,Outfile2,Infile);
const
  NumParts = 9;
  NumClasses = 2;
  epsilon = 1.0E-10;
  LargeNum = 9999999999;
  HoursPerYear = 2000;
  MaxK = 150;
  MaxInvBud = 3000;

type
  creal = array[1..NumParts,0..MaxK] of real;
  aint = array[1..NumParts] of integer;

var
  UBudget, LBudget, SBudget, Budget, cn1, cn2, ce, cq, taun1, taun2,
  tau, tauq, FECost : real;
  LambdaInitial, Mu, Travel, Adjust, Repair,
  p2, p4, Up2, Up4, Lp2, Lp4, Sp2, Sp4 : real;
  X : integer;
  Outfile1, Outfile2, Infile : text;
  ID : array[1..3] of char;
  Rep : integer;
  G : array[1..NumParts] of real;
  H1, H2 : array[1..NumParts] of real;
  p3i : creal;
  p3, util : real;
  PA, BestPA, Bestp3, Bestutil, BestQ1, BestQ2 : real;
  lx, ux, BestX : integer;
  Lambda, Rho : array[1..NumClasses] of real;
  p1 : real;
  Sigma : array[0..NumClasses] of real;
  BigRho : real;
  BestKit, Kit : aint;
  Bestholdcost, Besttotalcost, Bestemergcost, Bestreplencost : real;

(* ------------------------------------------------------------------------- *)
function Power(Mantissa, exponent : real) : real;
begin
  if Mantissa < 0 then
  begin
    writeln('Mantissa < 0');
    writeln('Mantissa = ', Mantissa, ' exp = ', exponent);
    readln;
  end; (* if Mantissa < 0 *)
  Power := exp(ln(Mantissa)*exponent);
end; (* Power *)

procedure InputData;
var
  Count : integer;
begin
  readln(Infile,ID);
  readln(Infile,taun1);
  readln(Infile,taun2);
  readln(Infile,taue);
  readln(Infile,tauq);
  readln(Infile,Travel);
  readln(Infile,Adjust);
  readln(Infile,Repair);
  readln(Infile,LambdaInitial);
  readln(Infile,FECost);
  readln(Infile,cn1);
  readln(Infile,cn2);
  readln(Infile,ce);
  readln(Infile,cq);
  readln(Infile,LBudget,UBudget,SBudget);
  readln(Infile,Up2,Lp2,Sp2);
  readln(Infile,Up4,Lp4,Sp4);
  for Count := 1 to NumParts do
    readln(Infile,G[Count],H1[Count],H2[Count]);
end; (* InputData *)

procedure SetUpOutfiles;
var
  FN1, FN2 : array[1..20] of char;
begin
  writeln('Setupoutfiles');
  FN1 := 'all' + ID + '.s';
  rewrite(Outfile1,FN1); (* output for reading *)
  FN2 := 'best' + ID + '.s';
  rewrite(Outfile2,FN2); (* output for graphical display *)
end; (* SetUpOutfiles *)
procedure Outresultsall(q1, pa, q2, tc, ec, rc, hc, p3, u: real; x: integer; k: aint);
var
  i: integer;
begin
  writeln(Outfile1,q1:6:3,' ',pa:6:3,' ',q2:6:3,' ',tc:7:0,' ',ec:6:0,
  write(Outfile1,x:2,' ');
  for i := 1 to NumParts do
    write(Outfile1,k[i]:3:0,' ');
  writeln(Outfile1,' ');
end; (* OutResults *)

procedure Outresultsbest(q1, pa, q2, tc, ec, rc, hc, p3, u: real;
                          x: integer; k: aint);
var
  i: integer;
begin
  writeln(Outfile2,q1:6:3,' ',pa:6:3,' ',q2:6:3,' ',tc:7:0,' ',ec:6:0,
  write(Outfile2,x:2,' ');
  for i := 1 to NumParts do
    write(Outfile2,k[i]:3:0,' ');
  writeln(Outfile2,' ');
end; (* OutResults *)

function Factorial(Num: integer): real;
var
  Count: integer;
  Accum: real;
begin
  Accum := 1.0;
  for Count := 1 to Num do
    Accum := Accum * Count;
  Factorial := 1.0 * Accum;
end; (* Factorial *)

function CalcMu: real;
var
  sum: real;
begin
  (*writeln('begin CalcMu');*)
  sum := p1*(Travel) + p1*(1-p2)*Adjust + p1*p2*Repair;
  sum := sum + p1*p2*p3*((p4*Travel) + (1-p4)*tauq);
  CalcMu := (1/sum);
procedure Calcp3;
var
  i, Count : integer;
  t, sum, part : real;
  test : real;
begin
  i := 1 to NumParts do
    begin
      p3[i,0] := 1.0;
      t := (p1*p2*(LambdaInitial)*G[i]*taun1)/X;
      sum := 1.0;
      part := 1.0;
      for Count := 1 to MaxK do
        begin
          part := (part*t)/Count;
          if (part<epsilon) then part := 0.0;
          sum := sum + part;
          test := part/sum;
          if (test < epsilon) then p3[i,Count] := 0.0
            else p3[i,Count] := test;
          if (test > 1.0) then
            begin
              writeln('Calc3 = ', test);
              writeln('part = ',part,' sum = ',sum);
              writeln('t = ',t, ' LambdaInitial = ',LambdaInitial);
              halt;
            end; (* if test *)
        end; (* for Count *)
    end; (* for i *)
end; (* Calcp3 *)

(* Calcp3 *)

function CalcQWait (i:integer; m:real) : real;
var
  Temp1, Temp2, Temp3 : real;
  Count : integer;
begin
  (*writeln('begin CalcQWait BigRho = ',BigRho:8:6);*)
  Temp1 := 0;
  for Count := 0 to (X-1) do
Temp1 := Temp1 + (Power(X*BigRho,Count−X)/Factorial(Count));
Temp2 := Factorial(X)*(1−BigRho)*X*m*Temp1;
Temp3 := 1/(Temp2 + (X*m));
CalcQWait := Temp3/((1−Sigma[I−1])*(1−Sigma[I]));
(* writeln(‘end CalcQWait’); *)
end; (* function CalcQWait *)

procedure AllocInvBud(B:integer; var flag:boolean);
var
 v : array[0..NumParts,0..MaxInvBud] of real;
cost : array[0..NumParts,0..MaxK] of real;
i1,i2, i, k, b : integer;
kstar : array[0..NumParts,0..MaxInvBud] of integer;
bestval, t1, t2 : real;
begin
 t1 := LambdaInitial*HoursPerYear*p1*p2*cn1/X;
t2 := LambdaInitial*HoursPerYear*p1*p2*ce/X;
for i1 := 1 to NumParts do
for i2 := 0 to MaxK do
 cost[i1,i2] := (H1[i1]*i2) + (t1*G[i1]*(1−p3[i1,i2]))
 + (t2*G[i1]*p3[i1,i2]);
for i := 0 to NumParts do
 begin
 for k := 0 to MaxInvBud do
 begin
 v[i,k] := 0.0;
kstar[i,k] := 0;
 end;
end;
for i := 1 to NumParts do
 for b := 0 to B do
 begin
 bestval := 2.0;
 for k := 0 to MaxK do
 if (b−cost[i,k] >= 0) then
 if ((v[i−1, trunc(b−cost[i,k]))+(G[i]*p3[i,k])) <= bestval) then
 begin
 bestval := v[i−1, trunc(b−cost[i,k]))+(G[i]*p3[i,k]);
kstar[i,b] := k;
 end;
v[i,b] := bestval;
 end; (* for b * )
(* backtrack to find opt *)
b := B;
for i := NumParts downto 1 do
begin
    Kit[i] := kstar[i,b];
    b := trunc(b-cost[i,Kit[i]]);
    if ( b < 0 ) then
        begin
            b := 0;
            flag := true;
        end; (* if b < 0 *)
    end;
    p3 := v[NumParts,B];
end; (* AllocInvBud *)

********************************************************************************
procedure FindOpt;
var
    i : integer;
    Q1, Q2 : real;
    BudgetPrime : integer;
    emergcost, replencost, holdcost, totalcost : real;
    BestTotalTime, TotalTime : real;
    flag : boolean;
begin (* 1 *)
    BestTotalTime := LargeNum;
    BestX := 0;
    (* for x := lower to upper *)
    for X := lx to ux do
        begin (* 2 *)
            CalcP3;
            BudgetPrime := trunc((Budget - (X*FECost))/(X*100));
            if (BudgetPrime > MaxInvBud) then
                writeln('Warning : Available Inventory Budget exceeds MaxInvBud');
            if (0 < BudgetPrime) then
                begin
                    flag := false;
                    AllocInvBud(BudgetPrime,flag);
                    if (flag=false) then
                        begin
                            Mu := CalcMu;
                            Lambda[1] := LambdaInitial*p1*p2*p3*p4;
                            Lambda[2] := LambdaInitial*p1;
                            (* calculate Q1(mu,Lambda[1]) & Q2(mu,Lambda[1]) *)
                            for i := 1 to NumClasses do
                                Rho[i] := Lambda[i]/(X*Mu);
                            Sigma[0] := 0.0;
                            for i := 1 to NumClasses do
Sigma[i] := Sigma[i-1] + Rho[i];
BigRho := Sigma[NumClasses];
if (BigRho > 1.0) then (* queue explodes *)
  begin (* 6 *)
    writeln('Queue explodes in FindOpt ');
    writeln('BigRho = ',BigRho);
  end (* 6 if BigRho *)
else
begin (* 7 *)
    (* class 1 is higher priority – return visits *)
    Q1 := CalcQWait(2,Mu);
    Q2 := CalcQWait(1,Mu);
    util := (LambdaInitial*p1*(Travel+(1-p2)*Adjust+
               p2*Repair+p2*p3*((p4*Travel)+(1-p4)*tauq)))/X;
    PA := p1*p2*p3*(p4*(Travel+Q2+tauq))
         + (1-p4)*tauq);
    TotalTime := p1*Q1 + PA;
    (* Note ce, cn1, and h1 are in hundreds of dollars *)
    emergcost := 100*ce*Lambdaintial*HoursPerYear*p1*p2*p3;
    replencost := 100*cn1*Lambdaintial*HoursPerYear*p1*p2*(1-p3);
    holdcost := 0.0;
    for i := 1 to NumParts do
      holdcost := holdcost + (100*X*H1[i]*Kit[i]);
    totalcost := emergcost + replencost + holdcost + (X*FECost);
Outresultslall(Q1,PA,Q2,totalcost,emergcost,replencost,holdcost,
                p3,util,X,Kit);
if (TotalTime<=BestTotalTime) then
  begin
    BestTotalTime := TotalTime;
    BestX := X;
    Bestp3 := p3;
    for i := 1 to NumParts do
      BestKit[i] := Kit[i];
    BestQ1 := Q1;
    BestQ2 := Q2;
    BestPA := PA;
    Bestutil := util;
    Bestemergcost := emergcost;
    Bestreplencost := replencost;
    Bestholdcost := holdcost;
    Besttotalcost := totalcost;
end; (* if better solution *)
  end; (* 7 if queue does not explode *)
end; (* flag = false *)
end;
end; (* 2 for x *)
if (BestX <> 0) then
  Outresultsbest(BestQ1,BestPA,BestQ2,Besttotalcost,Bestemergcost,
  Bestreplencost,Bestholdcost,Bestp3,Bestutil,BestX,BestKit);
end; (* 1 FindOpt *)
(*==============================================================================*)

begin
  Rep := 0;
  reset(Infie,'fs.dat');
  InputData;
  SetUpOutfiles;
  Budget := LBudget;
  p2 := Lp2;
  p4 := Lp4;
  p1 := 1;
  writeln('');
  write('Enter lower bound on FEs : '); readln(lx);
  write('Enter upper bound on FEs : '); readln(ux);
  repeat
    repeat
      Rep := Rep + 1;
      writeln('Rep = ',Rep:3);
      FindOpt;
      Budget := Budget + SBudget;
      until (Budget > UBudget);
    Budget := LBudget;
    p2 := p2 + Sp2;
    until (p2 > Up2);
    p2 := Lp2;
    p4 := p4 + Sp4;
    until (p4 > Up4);
  end.
program TWO(input, output, Outfile1, Infile);
const
   NumParts = 9;
   NumClasses = 2;
   epsilon = 1.0E-10;
   LargeNum = 999999999;
   HoursPerYear = 2000;
   MaxK = 20;
   MaxS = 20;

var
   prob : array[1..NumParts,0..MaxK,1..20] of real;
   k1, j2, lx, ux : integer;
   combo : array[1..MaxK,1..MaxK] of real;
   UBudget, LBudget, SBudget, Budget,
   cn1, cn2, ce, cq, taun1, taun2, taue, tauq,
   FECost : real;
   LambdaInitial, Mu, Travel, Adjust, Repair,
   p2, p4 : real;
   X : integer;
   Outfile1, Infile : text;
   ID : array[1..3] of char;
   Rep : integer;
   G : array[1..NumParts] of real;
   H1, H2 : array[1..NumParts] of real;
   p3i : creal;
   fsi : ereal;
Lambda, Rho : array[1..NumClasses] of real;
p1 : real;
Sigma : array[0..NumClasses] of real;
BigRho : real;
Depot, Kit : aint;
DepotNonFill, KitNonFill : areal;
d1, d2, d3, d4, d5, d6 : real;
n : aint;
BestX : integer;
pa, Bestpa, Bestp3, Bestfs, Bestutil, BestQ1, BestQ2 : real;
BestKit, BestDepot : aint;
Bestholdcost, Besttotalcost, Bestemergcost, Bestreplencost :real;
Up2, Lp2, Up4, Lp4, Sp2, Sp4 : real;
p3, fs : real;

function Power(Mantissa, exponent : real) : real;
begin
  if Mantissa < 0 then
    begin
      writeln('Mantissa < 0');
      writeln('Mantissa = ', Mantissa, ' exp = ', exponent);
      readln;
    end; (* if Mantissa < 0 *)
  Power := exp(ln(Mantissa)*exponent);
end; (* Power *)

procedure InputData;
var
  Count : integer;
begin
  readln(Infile,ID);
  readln(Infile,taun1);
  readln(Infile,taun2);
  readln(Infile,taue);
  readln(Infile,tauq);
  readln(Infile,Travel);
  readln(Infile,Adjust);
  readln(Infile,Repair);
  readln(Infile,LambdaInitial);
  readln(Infile,FECost);
  readln(Infile,cn1);
  readln(Infile,cn2);
  readln(Infile,ce);
  readln(Infile,cq);
  readln(Infile,LBudget,UBudget,SBudget);
  readln(Infile,Lp2,Up2,Sp2);
readln(Infie,Lp4,Up4,Sp4);
for Count := 1 to NumParts do
readln(Infie,G[Count],H1[Count],H2[Count],n[Count]);
end; (* InputData *)

procedure SetUpOutfiles;
var
FN1 : array[1..20] of char;
begin
writeln('Setupoutfiles');
FN1 := 'all' + ID + '.t';
rewrite(Outfile1,FN1); (* output for reading *)
end; (* SetUpOutfiles *)

procedure OutResultsAll(q1,pa,q2,tc,ec,qc,r1c,r2c,h1c,h2c,p3,fs,u: real;
x:integer; k,s:aint;g:real);
var
i : integer;
begin
writeln(Outfile1,g:10:8,' ',q1:6:3,' ',pa:6:3,' ',q2:6:3,' ',tc:7:0,' ',ec:6:0,
' ',qc:6:0,' ',r1c:6:0,' ',r2c:6:0,' ',h1c:6:0,' ',h2c:6:0,' ');
writeln(Outfile1,p3:8:6,' ',fs:8:6,' ',u:5:3,' ');
write(Outfile1,x:2,' ');
for i := 1 to NumParts do
write(Outfile1,k[i]:3:0,' ');
for i := 1 to NumParts do
write(Outfile1,s[i]:3:0,' ');
writeln(Outfile1,' ');
end; (* OutResults *)

function Factorial(Num:integer) : real;
var
Count : integer;
Accum : real;
begin
Accum := 1.0;
for Count := 1 to Num do
Accum := Accum * Count;
Factorial := 1.0 * Accum;
end; (* Factorial *)

procedure Calcds;
var
lp : real;
begin
lp := LambdaInitial*HoursPerYear*p2;
d1 := p1*Travel + p1*(1−p2)*Adjust + p1*p2*Repair;
d2 := p4*Travel + tauq;
d3 := tauq + Travel − (1−p4)*tauq;
d4 := X*FECost + p1*lp*cn1;
d5 := p1*lp*(cq−cn1+cn2);
d6 := p1*lp*(ce−cq−cn2);
end; (* Calcds *)

function CalcMu(k,d:aint) : real;
var
   i : integer;
   t1, t2, sum : real;
begin
   (*writeln('begin CalcMu');*)
   sum := 0.0;
   for i := 1 to NumParts do
      begin
         t1 := p3i[i,k[i]]*(1−fsi[i,k[i],d[i]])*(tauq*(1−p4) + p4*Travel);
         t2 := p3i[i,k[i]]*fsi[i,k[i],d[i]]*Travel;
         sum := sum + (G[i]*(t1+t2));
      end;
   sum := (sum*p2*p1) + (p1*Travel) + (p1*(1−p2)*Adjust) +
   (p1*p2*Repair);
   CalcMu := (1/sum);
   (*writeln('end CalcMu ');*)
end;

procedure Calcp3;
var
   i, j :integer;
   sum, part : real;
   t, test : real;
begin
   for i := 1 to NumParts do
      begin
         prob[i,0,X−lx+1] := 1.0;
         t := (p1*p2*LambdaInitial*G[i]*taun1)/X;
         sum := 1.0;
         part := 1.0;
         for j := 1 to MaxK do
            begin
               part := (part*t)/j;
               if (part<epsilon) then part := 0.0;
               sum := sum + part;
               test := part/sum;
            end;
      end;
end;
if (test<epsilon) then prob[i,j,X-lx+1] := 0.0
else prob[i,j,X-lx+1] := test;
if ( test > 1.0) then
  begin
    writeln('Calcp3 = ',test);
    halt;
  end; (* if test *)
end; (* for j *)
end; (* for i *)
end; (* Calcp3 *)

procedure Calcfs;
var
  i, j,k :integer;
  sum, part : real;
  t, test : real;
begin
  for i := 1 to NumParts do
    for k := 0 to MaxK do
      begin
        fsi[i,k,0] := 1.0;
        t := (p1*p2*LambdaInitial*G[i]*taun1*p3[i,k]);
        sum := 1.0;
        part := 1.0;
        for j := 1 to MaxS do
          begin
            part := (part*t)/j;
            if (part<epsilon) then part := 0.0;
            sum := sum + part;
            test := part/sum;
            if (test<epsilon) then fsi[i,k,j] := 0.0
              else fsi[i,k,j] := test;
            if ( test > 1.0) then
              begin
                writeln('Calcp3 = ',test);
                halt;
              end; (* if test *)
          end; (* for j *)
        end; (* for i *)
      end; (* Calcfs *)
  end; (* Calcp3 *)

function CalcQWait (I:integer; m:real) : real;
var
  Temp1, Temp2, Temp3 : real;
  Count : integer;
begin
  // Function implementation
end;
begin  
(* writeln('begin CalcQWait BigRho = ',BigRho:8:6); *)  
Temp1 := 0;  
for Count := 0 to (X-1) do  
  Temp1 := Temp1 + (Power(X*BigRho,Count-X)/Factorial(Count));  
Temp2 := Factorial(X)*(1-BigRho)*X*m*Temp1;  
Temp3 := 1/(Temp2 + (X*m));  
CalcQWait := Temp3/((1-Sigma[I-1])*(1-Sigma[I]));  
(* writeln('end CalcQWait'); *)  
end; (* function CalcQWait *)

**********************************************************************************

procedure CalcSubproblem(Q1,Q2,g:real; var K,D : aint);  
var  
  value, bestvalue, t1, t2, t3 : real;  
  bestK, bestS : integer;  
  i, j, k : integer;  
begin  
  for i := 1 to NumParts do  
    begin  
      bestvalue := LargeNum;  
      bestK := -1;  
      bestS := -1;  
      for j := 0 to MaxK do  
        begin  
          for k := 0 to MaxS do  
            begin  
              t1 := g*(X*H1[i]*j + H2[i]*k);  
              t2 := d1+ p1*Q1  
                  + p1*p2*p3[i,j]*d2+p1*p2*p3[i,j]*f[s][i,j,k]*(d3+Q2)  
                  + p1*p2*p3[i,j]*p4*Q2;  
              t3 := g*(d4 + p3[i,j]*d5 + p3[i,j]*f[s][i,j,k]*d6 - Budget);  
              value := t1 + G[i]*(t2+t3);  
              if (value <= bestvalue) then  
                begin  
                  bestvalue := value;  
                  bestK := j;  
                  bestS := k;  
                  end; (* if value < bestvalue *)  
            end; (* for k *)  
          end; (* for j *)  
        end; (* for i *)  
  end; (* CalcSubproblem *)

**********************************************************************************
procedure FindOpt;

var i, reps, k1, k2 : integer;
gamma, mubar : real;
ugamma, lgamma, stepgamma : real;
igamma : integer;
ap, Q1, Q2 : real;
qc, util, h1c, h2c, ec, r1c, r2c : real;
lambdabar : array[1..NumClasses] of real;
flagmu, flaglambda, flagineas : boolean;
lastTotalWait, TotalWait, TotalCost : real;
gpf, g1p, gp1f : real;
(* goon : integer ; *)
begin (* 1 *)
flagmu := false;
flaglambda := false;
Lambda[2] := p1*LambdaInitial; (* lower priority class *)
writeln(’ ’);
writeln(’Enter values for budget multiplier (gamma) : ’);
write(’Lower : ’);
readln(lgamma);
write(’Upper : ’);
readln(ugamma);
write(’Step : ’);
readln(stepgamma);
(* for x := lower to upper *)
for X := lx to ux do
begin (* 2 *)
writeln(’ ’);
writeln(’X = ’,X);
writeln(’ ’);
for k1 := 1 to NumParts do
  for k2 := 0 to MaxK do
    begin
      p3i[k1,k2] := prob[k1,k2,X–lx+1];
    end; (* for k2 *)
Calcfs;
Calcds;
for igamma := 0 to (trunc((ugamma–lgamma)/stepgamma)+1) do
begin (* 3 *)
gamma := lgamma + (igamma*stepgamma);
mubar := 1/(Travel+((1–p2)*Adjust)+(p2*Repair));
lambdabar[1] := 0.0;
reps := 0;
flagmu := false;
flaglambda := false;
flaginfeas := false;
repeat
  begin (* 4 *)
    reps := reps + 1;
    if (reps > 4) then flaginfeas := true;
    (* writeln(' mubar = ',mubar, ' reps = ',reps); *)
    if (LambdaInitial/(mubar*X) < 1.0) then
      begin (* 5 *)
        flagmu := false;
        flaglambda := false;
      (* calculate Q1(mubar, Lambda[1]) & Q2(mubar, Lambda[1]) *)
      for i := 1 to NumClasses do
        Rho[i] := lambdabar[i]/(X*mubar);
      Sigma[0] := 0;
      for i := 1 to NumClasses do
        Sigma[i] := Sigma[i-1] + Rho[i];
      BigRho := Sigma[NumClasses];
      if (BigRho > 1.0) then (* queue explodes *)
        begin (* 6 *)
          writeln('Queue explodes in FindOpt ');
          writeln('BigRho = ', BigRho);
          writeln('mu=', mubar:8:3, ' lambda=', lambdabar[1]:8:3, ' X=', X:4);
          flaginfeas := true;
        end (* 6 if BigRho *)
      else
        begin (* 7 *)
          (* class 1 is higher priority - return visits *)
          Q1 := CalcQWait(2, mubar);
          Q2 := CalcQWait(1, mubar);
          (* calculate optimal k and s *)
          CalcSubproblem(Q1, Q2, gamma, Kit, Depot);
          (* calculate true arrival rate *)
          p3 := 0.0;
          for i := 1 to NumParts do
            p3 := p3 + (G[i]*p3[i,Kit[i]]* 
                         (fsi[i,Kit[i],Depot[i]]+p4*(1-fsi[i,Kit[i],Depot[i]])));
          Lambda[1] := p1*p2*p3*LambdaInitial; (*measured in hours *)
          if (abs(Lambda[1] - lambdabar[1]) > epsilon) then
          else
            flaglambda := true;
        (* calculate true service rate *)
        Mu := CalcMu(Kit, Depot); (* measured in hours *)
        if (abs(Mu - mubar) > epsilon) then
begin (* 8 *)
    mubar := Mu;
    flagmu := false;
end (* 8 if Mu=mubar *)
else (* close enough *)
    flagmu := true;
(* $$$$$$$$$$$ *)
(* writeln(' ' );
  writeln('Mu = ',Mu:8:6,' mubar = ',mubar:8:6);
  writeln('lambda = ',Lambda[1]:8:4,' lambdabar = ',lambdabar[1]:8:4);
for i := 1 to NumParts do
  writeln('Kit = ',Kit[i]:3,' Depot = ',Depot[i]:3);
readin(goon);
if (goon = 0) then halt;(*)
(* $$$$$$$$$$$$$ $$ *)
end (* 7 if queue does not explode *)
end (* 5 if lambda/mubar*X < 1 *)
else (* lambda/(mubar*X) *)
    flaginfeas := true;
end; (* 4 repeat *)
until ((flagmu=true)&&(flaglambda=true)) ! (flaginfeas=true);
if ((flagmu=true)&&(flaglambda=true)) then
begin (* 9 *)
    for i := 1 to NumParts do
        KitNonFill[i] := p3[i,Kit[i]];
    for i := 1 to NumParts do
        DepotNonFill[i] := fsi[i,Kit[i],Depot[i]];
    TotalWait := p1*(Q1+Travel+(1-p2)*Adjust+p2*Repair);
    for i := 1 to NumParts do
        TotalWait := TotalWait + G[i]*KitNonFill[i]*
        (DepotNonFill[i]*(tau + Q2 + Travel)
        +(1-DepotNonFill[i])*(p4*(tau+Q2+Travel)+(1-p4)*tauq));
    pa := TotalWait - p1*(Q1+Travel+(1-p2)*Adjust+p2*Repair);
    p3 := 0;
    fs := 0;
    for i := 1 to NumParts do
        p3 := p3 + G[i]*p3[i,Kit[i]];
    for i := 1 to NumParts do
        fs := fs + G[i]*fsi[i,Kit[i],Depot[i]];
    r1c := 0;
    r2c := 0;
    h1c := 0;
    h2c := 0;
    ec := 0;
    qc := 0;
gpf := 0;
glp := 0;
gpf := 0;
TotalCost := 0;
for i := 1 to NumParts do
    begin
        gpf := gpf + (G[i]*p3[i,Kit[i]]*fsi[i,Kit[i],Depot[i]]);
        glp := glp + (G[i]*(1-p3[i,Kit[i]]));
        gpf1f := gpf1f + (G[i]*p3[i,Kit[i]]*(1-fsi[i,Kit[i],Depot[i]]));
    end;
ec := ce*p1*p2*LambdaInitial*HoursPerYear*gpf*100;
qc := cq*p1*p2*LambdaInitial*HoursPerYear*gpf1f*100;
rlc := cn1*p1*p2*LambdaInitial*HoursPerYear*glp*100;
r2c := cn2*p1*p2*LambdaInitial*HoursPerYear*gpf1f*100;
for i := 1 to NumParts do
    hlc := hlc + (X*H1[i]*Kit[i]*100);
for i := 1 to NumParts do
    h2c := h2c + (H2[i]*Depot[i]*100);
TotalCost := X*FECost + ec + qc + rlc + r2c + hlc + h2c;
util := LambdaInitial*(p1*Travel+p1*(1-p2)*Adjust+ p1*p2*Repair
                   + p1*p2*p3*(1-fs)*(p4*Travel + (1-p4)*tauq)
                   + p1*p2*p3*fs*Travel)/X;
if (TotalWait <> lastTotalWait) then
    OutResultsAll(Q1,pa,Q2,TotalCost,ec,qc,rlc,r2c,h1c,h2c,p3,fs,util,
                   X,Kit,Depot,gamma);
    lastTotalWait := TotalWait;
end; (* 9 *)
end; (* 3 for igamma *)
end; (* 2 for x *)
end; (* 1 FindOpt *)

(*==============================================================================*)
begin
    Rep := 0;
    reset(Infie,’fs.dat’);
    InputData;
    SetUpOutfiles;
    Budget := LBudget;
p1 := 1;
p2 := Lp2;
p4 := Lp4;
for k1 := 1 to MaxK do
    combo[k1,1] := k1;
for k1 := 1 to MaxK do
    for j2 := 2 to k1 do
        combo[k1,j2] := combo[k1,j2-1]*(k1-j2+1)/j2;
writeln(’ ’);
write(’Enter lower bound on FE’s ‘);
readln(1x);
write('Enter upper bound on FEs:');
readln(ux);
for X := 1x to ux do
begin
  writeln('Calculating probabilities — X = ',X:3);
  Calcp3;
end; (* for X *)
repeat
  repeat
    Rep := Rep + 1;
    FindOpt;
    p2 := p2 + Sp2;
    until (p2 >= Up2);
  p2 := Lp2;
  p4 := p4 + Sp4;
  until (p4 >= Up4);
end.
program SLARGE(input, output, Outfile1, Outfile2, Infile);
const
  NumParts = 9;
  NumClasses = 2;
  epsilon = 1.0E-10;
  LargeNum = 999999999;
  HoursPerYear = 2000;
  MaxK = 530;
  MaxInvBud = 5000;

type
  creal = array[1..NumParts, 0..MaxK] of real;
  aint = array[1..NumParts] of integer;

var
  prob: array[1..NumParts, 0..MaxK, 1..20] of real;
  k1, j2: integer;
  UBudget, LBudget, SBudget, Budget,
  cn1, cn2, ce, cq, taun1, taun2, taua, tauq,
  FECost: real;
  LambdaInitial, Mu, Travel, Adjust, Repair,
  p2, p4, Up2, Up4, Lp2, Lp4, Sp2, Sp4: real;
  X: integer;
  Outfile1, Outfile2, Infile: text;
  ID: array[1..3] of char;
  Rep: integer;
  n: array[1..NumParts] of integer;
  G: array[1..NumParts] of real;
  H1, H2: array[1..NumParts] of real;
  p3i: creal;
  p3, util: real;
  PA, BestPA, Bestp3, Bestutil, BestQ1, BestQ2: real;
  lx, ux, BestX: integer;
  Lambda, Rho: array[1..NumClasses] of real;
  p1: real;
  Sigma: array[0..NumClasses] of real;
  BigRho: real;
BestKit, Kit : aint;
Bestholdcost, Besttotalcost, Bestemergcost,Bestreplencost : real;
combo : array[1..MaxK,1..MaxK] of real;

function Power(Mantissa, exponent : real) : real;
begin
  if Mantissa < 0 then
    begin
      writeln('Mantissa < 0');
      writeln('Mantissa = ',Mantissa, ' exp = ',exponent);
      readln;
    end; (* if Mantissa < 0 *)
  Power := exp(ln(Mantissa)*exponent);
end; (* Power *)

procedure InputData;
var
  Count : integer;
begin
  readln(Infile,ID);
  readln(Infile,taun1);
  readln(Infile,taun2);
  readln(Infile,taue);
  readln(Infile,tauq);
  readln(Infile,Travel);
  readln(Infile,Adjust);
  readln(Infile,Repair);
  readln(Infile,Initial);
  readln(Infile,FECost);
  readln(Infile,cn1);
  readln(Infile,cn2);
  readln(Infile,ce);
  readln(Infile,cq);
  readln(Infile,UBudget,UBudget,SBudget);
  readln(Infile,Lp2,Up2,Sp2);
  readln(Infile,Lp4,Up4,Sp4);
  for Count := 1 to NumParts do
    readln(Infile,G[Count],H1[Count],H2[Count],n[Count]);
end; (* InputData *)

procedure SetUpOutfiles;
var
  FN1, FN2 : array[1..20] of char;
begin
  writeln('Setupoutfiles');
  FN1 := 'all' + ID + '.sl';
rewrite(Outfile1,FN1); (* output for reading *)
FN2 := 'best' + ID + '.sl';
rewrite(Outfile2,FN2); (* output for graphical display *)
end; (* SetUpOutfiles *)

(*------------------------------------------------------------------------)
procedure Outresultsall(q1,pa,q2,tc,ec,rc,hc,p3,u:real;
x : integer; k : aint);
var
  i : integer;
begin
  writeln(Outfile1,q1:6:3,' ',pa:6:3,' ',q2:6:3,' ',tc:7:0,' ',ec:6:0,
  write(Outfile1,x:2,' ');
  for i := 1 to NumParts do
    write(Outfile1,k[i]:3:0,' ');
  writeln(Outfile1,'
end; (* OutResults *)

(*------------------------------------------------------------------------)
procedure Outresultsbest(q1,pa,q2,tc,ec,rc,hc,p3,u:real;
x : integer; k : aint);
var
  i : integer;
begin
  writeln(Outfile2,q1:6:3,' ',pa:6:3,' ',q2:6:3,' ',tc:7:0,' ',ec:6:0,
  write(Outfile2,x:2,' ');
  for i := 1 to NumParts do
    write(Outfile2,k[i]:3:0,' ');
  writeln(Outfile2,'
end; (* OutResults *)

(*------------------------------------------------------------------------)
function Factorial(Num:integer) : real;
var
  Count : integer;
  Accum : real;
begin
  Accum := 1.0;
  for Count := 1 to Num do
    Accum := Accum * Count;
  Factorial := 1.0 * Accum;
end; (* Factorial *)

(*------------------------------------------------------------------------)
function CalcMu : real;
var
  sum : real;
begin
(*writeln('begin CalcMu');*)
sum := p1*(Travel) + p1*(1-p2)*Adjust + p1*p2*Repair;
sum := sum + p1*p2*p3*((p4*Travel) + (1-p4)*tauq);
CalcMu := (1/sum);
(*writeln('end CalcMu');*)
end;
*****************************************************************************

procedure Calcp3;
var
  k, i, j: integer;
  sum, part: real;
  test: real;
  lt, ltau: real;
begin
  for i := 1 to NumParts do
    begin
      prob[i,0,X-1x+1] := 1.0;
      for k := 1 to n[i] do
        begin
          sum := (k/n[i]);
          lt := 1;
          ltau := (LambdaInitial*p1*p2*G[i]*taun1)/(X*n[i]);
          part := 1;
          for j := 1 to k do
            begin
              lt := lt*ltau;
              part := ((k-j)/n[i])*lt*combo[k,j];
              if (part<epsilon) then part := 0.0;
              sum := sum + part;
              end; (* for j *)
          test := 1-(sum*Power(n[i]/(n[i]+(ltau*n[i])),k));
          if (test < epsilon) then prob[i,k,X-1x+1] := 0.0
          else prob[i,k,X-1x+1] := test;
          if ((test > 1.0) or (test<0.0)) then
            begin
              writeln('Calcp3 = ', test);
              writeln('part = ',part,' sum = ',sum);
              writeln(' LambdaInitial = ',LambdaInitial);
              halt;
              end; (* if test *)
          end; (* for k *)
      end; (* for i *)
(*for i := 1 to NumParts do
  for j := 1 to n[i] do
    writeln('i=',i,' j=',j,' prob[i,j]:12:10);
function CalcQWait (I:integer; m:real) : real;
var
  Temp1, Temp2, Temp3 : real;
  Count : integer;
begin
  (* writeln('begin CalcQWait BigRho = ',BigRho:8:6); *)
  Temp1 := 0;
  for Count := 0 to (X–1) do
    Temp1 := Temp1 + (Power(X*BigRho,Count–X)/Factorial(Count));
  Temp2 := Factorial(X)*(1–BigRho)*X*m*Temp1;
  Temp3 := 1/(Temp2 + (X*m));
  CalcQWait := Temp3/((1–Sigma[I–1])*(1–Sigma[I]));
  (* writeln('end CalcQWait'); *)
end; (* function CalcQWait *)

procedure AllocInvBud(B:integer; var flag:boolean);
var
  v : array[0..NumParts,0..MaxInvBud] of real;
  cost : array[0..NumParts,0..MaxK] of real;
  i1,i2, i, k, b : integer;
  kstar : array[0..NumParts,0..MaxInvBud] of integer;
  bestval, t1, t2 : real;
  found : boolean;
begin
  t1 := (LambdaInitial*HoursPerYear*p1*p2*cn1)/X;
  t2 := (LambdaInitial*HoursPerYear*p1*p2*ce)/X;
  for i1 := 1 to NumParts do
    for i2 := 0 to MaxK do
      cost[i1,i2] := (H1[i1]*i2) + (t1*G[i1]*(1–p3[i1,i2]))
                    + (t2*G[i1]*p3[i1,i2]);
  for i := 0 to NumParts do
    begin
      for k := 0 to MaxInvBud do
        begin
          v[i,k] := 0.0;
          kstar[i,k] := -1;
          end;
        end;
  for i := 1 to NumParts do
    Kit[i] := -1;
  for i := 1 to NumParts do
    for b := 0 to B do
begin
  bestval := 20.0;
  found := false;
  for k := 0 to n[i] do
    begin
      if (b–cost[i,k] >= 0) then
        begin
          if (((v[i–1, trunc(b–cost[i,k])]+(G[i]*p3i[i,k]))<=bestval) then
            begin
              bestval := v[i–1, trunc(b–cost[i,k])]+(G[i]*p3i[i,k]);
              kstar[i,b] := k;
              found := true;
            end;
          end;
        end; (* for k *)
      if (found = true) then v[i,b] := bestval
      else v[i,b] := 20.0;
    end; (* for b *)

    for b := 0 to B do
      begin
        for i := 1 to NumParts do
          write(' i=',i,'; b=' ,b:4,'; v[i,b]=' ,v[i,b]:6:4);
        writeln(' ');
        writeln(' ');
      end;
      for i := 1 to NumParts do
        for b := 0 to B do
          writeln(' i=',i,'; b=' ,b:3,'; kstar[i,b]=' ,kstar[i,b]:4);
  end;

  (* backtrack to find opt *)
  b := B;
  for i := NumParts downto 1 do
    begin
      Kit[i] := kstar[i,b];
      if (kstar[i,b] < 0) then b := -1
      else b := trunc(b–cost[i,Kit[i]]);
      if (b<0) then
        begin
          b := 0;
          flag := true;
        end; (* if b<0 *)
    end; (* for i *)
  if (flag = true) then writeln('Flag = true');
p3 := v[NumParts,B];
end; (* AllocInvBud *)

procedure FindOpt;
var
  i : integer;
  Q1, Q2 : real;
  BudgetPrime : integer;
  emergcost, replencost, holdcost, totalcost : real;
  BestTotalTime, TotalTime : real;
  flag : boolean;
  k1, k2 : integer;
begin (* 1 *)
  BestTotalTime := LargeNum;
  BestX := 0;
  (* for x := lower to upper *)
  for X := lx to ux do
    begin (* 2 *)
      writeln('X = ',X);
      for k1 := 1 to NumParts do
        for k2 := 0 to n[k1] do
          p3i[k1,k2] := prob[k1,k2,X-lx+1];
      BudgetPrime := trunc((Budget - (X*FECost))/(X*100));
      if (BudgetPrime > MaxInvBud) then
        writeln('Warning : Available Inventory Budget exceeds MaxInvBud');
      if (BudgetPrime > 0) then
        begin
          writeln('allocating budget');
          flag := false;
          AllocInvBud(BudgetPrime,flag);
          if (flag = false) then
            begin
              Mu := CalcMu;
              Lambda[1] := LambdaInitial*p1*p2*p3*p4;
              Lambda[2] := LambdaInitial*p1;
              (* calculate Q1(mu,Lambda[1]) & Q2(mu,Lambda[1]) *)
              for i := 1 to NumClasses do
                Rho[i] := Lambda[i]/(X*Mu);
              Sigma[0] := 0.0;
              for i := 1 to NumClasses do
                Sigma[i] := Sigma[i-1] + Rho[i];
              BigRho := Sigma[NumClasses];
              if (BigRho > 1.0) then (* queue explodes *)
                begin (* 6 *)
                  writeln('Queue explodes in FindOpt ');
                  writeln('BigRho = ',BigRho);
end (* 6 if BigRho *)
else
begin (* 7 *)
(* class 1 is higher priority - return visits *)
Q1 := CalcQWait(2,Mu);
Q2 := CalcQWait(1,Mu);
util := (LambdaiInitial*p1*(Travel+(1-p2)*Adjust
+(p2*Repair) + p2*p3*((p4*Travel)+(1-p4)*tauq)))/X;
PA := p1*p2*p3*(p4*(Travel+Q2+taue)+(1-p4)*tauq);
TotalTime := p1*Q1 + PA;
(* Note ce, cn1, and h1 are in hundreds of dollars *)
emergcost := 100*ce*LambdaiInitial*HoursPerYear*p1*p2*p3;
replencost := 100*cn1*LambdaiInitial*HoursPerYear*p1*p2*(1-p3);
holdcost := 0.0;
for i := 1 to NumParts do
  holdcost := holdcost + (100*X*H1[i]*Kit[i]);
totalcost := emergcost + replencost + holdcost + (X*FE Cost);
Outresultsall(Q1,PA,Q2,totalcost,emergcost,replencost,holdcost,
p3,util,X,Kit);
if (TotalTime<=BestTotalTime) then
begin
  BestTotalTime := TotalTime;
  BestX := X;
  Bestp3 := p3;
  for i := 1 to NumParts do
    BestKit[i] := Kit[i];
  BestQ1 := Q1;
  BestQ2 := Q2;
  BestPA := PA;
  Bestutil := util;
  Bestemergcost := emergcost;
  Bestreplencost := replencost;
  Bestholdcost := holdcost;
  Besttotalcost := totalcost;
end; (* if better solution *)
end; (* 7 if queue does not explode *)
end; (* flag Kit[i] < 0 *)
end; (* sum <Tab>
end; (* 2 for x *)
if (BestX <> 0) then
Outresultsbest(BestQ1,BestPA,BestQ2,Besttotalcost,Bestemergcost,
Bestreplencost,Bestholdcost,Bestp3,Bestutil,BestX,BestKit);
end; (* 1 FindOpt *)
(*=================================================================*)
begin
Rep := 0;
reset(Infile, ’fs.dat’);
InputData;
SetUpOutfiles;
Budget := LBudget;
p2 := Lp2;
p4 := Lp4;
p1 := 1;
writeln(’ ’);
write(’Enter lower bound on FEs : ’);
readln(lx);
write(’Enter upper bound on FEs : ’);
readln(ux);
for k1 := 1 to NumParts do
  for j2 := 0 to MaxK do
    for X := lx to ux do
      prob[k1,j2,X–lx+1] := 2.0;
  for k1 := 1 to MaxK do
    combo[k1,1] := k1;
  for k1 := 1 to MaxK do
    for j2 := 2 to k1 do
      combo[k1,j2] := combo[k1,j2–1] *(k1–j2+1)/j2;
  for X := lx to ux do
    begin
      writeln(’Calculating prob. X = ’,X:3);
      Calcp3;
    end;
  repeat
    repeat
      Rep := Rep + 1;
      FindOpt;
      Budget := Budget + SBudget;
      until ( Budget > UBudget );
    Budget := LBudget;
    p2 := p2 + Sp2;
    until ( p2 > Up2 );
    p2 := Lp2;
    p4 := p4 + Sp4;
    until ( p4 > Up4 );
  end.
program TWOLARGE(input,output,Outfile1,Infile,OutProbFile_InProbFile);
const
  NumParts = 9;
  NumClasses = 2;
  epsilon = 1.0E-10;
  LargeNum = 999999999;
  HoursPerYear = 2000;
  MaxK = 650;
  MaxS = 650;

var
  go_on : integer;
  prob : array[1..NumParts,0..MaxK,..20] of real;
  k1, j2, k3, k2, fileprobs : integer;
  combo : array[1..MaxK,1..MaxK] of real;
  UBudget, LBudget, SBudget, Budget, 
  cn1, cn2, ce, cq, taun1, taun2, taue, tauq, 
  FECost : real;
  LambdaInitial, Mu, Travel, Adjust, Repair,
  p2, p4 : real;
  X : integer;
  Outfile1, Infile, OutProbFile, InProbFile : text;
  ID : array[1..3] of char;
  Rep : integer;
  G : array[1..NumParts] of real;
  H1, H2 : array[1..NumParts] of real;
p3i : creal;
fsi : ereal;
Lambda, Rho : array[1..NumClasses] of real;
p1 : real;
Sigma : array[0..NumClasses] of real;
BigRho : real;
Depot, Kit : aint;
DepotNonFill, KitNonFill : areal;
d1, d2, d3, d4, d5, d6 : real;
n : aint;
BestX : integer;
p4, Bestpa, Bestp3, Bestfs, Bestutil, BestQ1, BestQ2 : real;
BestKit, BestDepot : aint;
Bestholdcost, Besttotalcost, Bestemergcost, Bestreplencost :real;
Up2, Lp2, Up4, Lp4, Sp2, Sp4 : real;
p3, fs : real;

function Power(Mantissa, exponent : real) : real;
begin
  if Mantissa < 0 then
    begin
      writeln('Mantissa < 0');
      writeln('Mantissa = ',Mantissa, ' exp = ',exponent);
      readln;
      end; (* if Mantissa < 0 *)
    Power := exp(ln(Mantissa)*exponent);
  end; (* Power *)
end;

procedure InputData;
var
  Count : integer;
begin
  readln(Infle,ID);
  readln(Infle,taun1);
  readln(Infle,taun2);
  readln(Infle,taue);
  readln(Infle,tauq);
  readln(Infle,Travel);
  readln(Infle,Adjust);
  readln(Infle,Repair);
  readln(Infle,LambdaInitial);
  readln(Infle,FECost);
  readln(Infle,cn1);
  readln(Infle,cn2);
  readln(Infle,ce);
  readln(Infle,cq);
readln(Infile,LBudget,UBudget,SBudget);
readln(Infile,Lp2,Up2,Sp2);
readln(Infile,Lp4,Up4,Sp4);
for Count := 1 to NumParts do
readln(Infile,G[Count],H1[Count],H2[Count],n[Count]);
end; (* InputData *)

procedure SetUpOutfiles;
var
FN1 : array[1..20] of char;
begin
  writeln('Setupoutfiles');
  FN1 := 'all' + ID + '.tl';
  rewrite(Outfile1,FN1);  (* output for reading *)
end; (* SetUpOutfiles *)

procedure OutResultsAll(q1,pa,q2,tc,ec,qc,r1c,r2c,h1c,h2c,p3,fs,u: real;
x:integer; k,s:aint;g:real);
var
  i: integer;
begin
  writeln(Outfile1,g:10:8,' ',q1:6:3,' ',pa:6:3,' ',q2:6:3,' ',tc:7:0,' ',ec:6:0,
    ' ',qc:6:0,' ',r1c:6:0,' ',r2c:6:0,' ',h1c:6:0,' ',h2c:6:0,' ',p3:8:6,
    ' ',fs:8:6,' ',u:5:3,' ',ce:5:2,' ',cq:5:2,' ');
  writeln(Outfile1,' ');for i := 1 to NumParts do
    write(Outfile1,k[i]:3:0, ' ');
  for i := 1 to NumParts do
    write(Outfile1,s[i]:3:0, ' ');
  writeln(Outfile1,' ');writeln(g:12:10,' ',q1:6:3,' ',pa:6:3,' ',q2:6:3,' ',tc:7:0,' ',ec:6:0,
    ' ',qc:6:0,' ',r1c:6:0,' ',r2c:6:0,' ',h1c:6:0,' ',h2c:6:0,' ',p3:8:6,
    ' ',fs:8:6,' ',u:5:3,' ',ce:5:2,' ',cq:5:2,' ');write(x:2, ' ');
  for i := 1 to NumParts do
    write(k[i]:3:0, ' ');
  for i := 1 to NumParts do
    write(s[i]:3:0, ' ');
  writeln(' ');
end; (* OutResults *)

function Factorial(Num:integer) : real;
var
  Count : integer;
  Accum : real;
begin
   Accum := 1.0;
   for Count := 1 to Num do
      Accum := Accum * Count;
      Factorial := 1.0 * Accum;
   end; (* Factorial *)
end; (* Calculates *)

procedure CalcMu(k,d:aint) : real;
begin
   (*writeln('begin CalcMu');*)
   sum := 0.0;
   for i := 1 to NumParts do
      begin
         t1 := p3i[i,k[i]]*(1-fsi[i,k[i],d[i]])*(tauq*(1-p4) + p4*Travel);
         t2 := p3i[i,k[i]]*fsi[i,k[i],d[i]]*Travel;
         sum := sum + (G[i] + (t1+t2));
      end;
   sum := (sum*p2*p1) + (p1*Travel) + (p1*(1-p2)*Adjust) +
         (p1*p2*Repair);
   CalcMu := (1/sum);
   (*writeln('end CalcMu');*)
end;

procedure Calcp3;
begin
   i, k, j :integer;
   sum, part : real;
   test : real;
   lt, ltau : real;
begin
writeln('begin Calcp3');
for i := 1 to NumParts do
begin
p3i[i,0] := 1.0;
for k := 1 to n[i] do
begin
sum := (k/n[i]);
lt := 1;
ltau := (p1*p2*(LambdaInitial)*G[i]*taun1)/(X*n[i]);
part := 1;
for j := 1 to k do
begin
lt := lt * ltau;
part := ((k-j)/n[i])*lt*combo[k,j];
if (part < epsilon) then part := 0.0;
sum := sum + part;
end; (* for j *)
test := 1-(sum*Power(n[i]/(n[i]+(ltau*n[i])),k));
if (test < epsilon) then p3i[i,k] := 0.0
else p3i[i,k] := test;
if (test > 1.0) then
begin
writeln('Calcp3 = ', test);
writeln('part = ',part,' sum = ',sum);
writeln('lt = ',lt,' LambdaInitial = ',LambdaInitial);
halt;
end; (* if test *)
end; (* for k *)
end; (* for i *)
(*for i := 0 to MaxK do
writeln(i,' ',p3i[1,i]:12:10,' ',p3i[2,i]:12:10);
*)
end; (* Calcp3 *)
(*******************************************************************************)
procedure Calcfs;
var
i, j, k : integer;
test, part, sum : real;
j3 : integer;
l0, ltau : real;
begin
writeln('begin Calcfs ');
(* calc fs *)
for i := 1 to NumParts do
begin
    writeln('i = ',i:4);
    for j:= 0 to n[i] do
        begin
            if (trunc(j/50)*50 = j) then writeln(' j = ',j:4);
            fsi[i,j,0] := 1.0;
            for k := 1 to n[i] do
                begin
                    sum := (k/n[i]);
                    lt := 1;
                    ltau := (p1*p2*(LambdaInitial)*p3[i,j]*G[i]*taun1)/(X*n[i]);
                    part := 1;
                    for j3 := 1 to k do
                        begin
                            lt := lt * ltau;
                            part := ((k–j3)/n[i])*lt*combo[k,j3];
                            if (part < epsilon) then part := 0.0;
                            sum := sum + part;
                        end; (* for j3 *)
                    test := 1–(sum*Power(n[i]/(n[i]+(ltau*n[i]),k));
                    if (test < epsilon) then fsi[i,j,k] := 0.0
                    else fsi[i,j,k] := test;
                    if (test > 1.0) then
                        begin
                            writeln('Calcfs = ',test);
                            writeln('part = ',part,' sum = ',sum);
                            writeln('lt = ',lt,' LambdaInitial = ',LambdaInitial);
                            halt;
                        end; (* if test *)
                end; (* for k *)
        end; (* for j *)
    end; (* for i *)
writeln('end Calcfs');
end; (* procedure Calcfs *)

function CalcQWait (l:integer; m:real) : real;
var
    Temp1, Temp2, Temp3 : real;
    Count : integer;
begin
    (*writeln('begin CalcQWait BigRho = ',BigRho:8:6);*)
    Temp1 := 0;
    for Count := 0 to (X–1) do
        Temp1 := Temp1 + (Power(X*BigRho,Count–X)/Factorial(Count));
    Temp2 := Factorial(X)*(1–BigRho)*X*m*Temp1;
Temp3 := 1/(Temp2 + (X*m));
CalcQWait := Temp3/((1–Sigma[I–1])*(1–Sigma[I]));
(* writeln(‘end CalcQWait’); * )
end; (* function CalcQWait *)
*********************************************************************
procedure CalcSubproblem(Q1,Q2,g:real; var K,D: aint);
var
  value, bestvalue, t1, t2, t3 : real;
  bestK, bestS : integer;
  i, j, k : integer;
begin
  writeln(‘Calling Subproblem Q1 = ’,Q1:8:6,’ Q2 = ’,Q2:8:6);
  for i := 1 to NumParts do
    begin
      bestvalue := LargeNum;
      bestK := -1;
      bestS := -1;
      for j := 0 to n[i] do
        begin
          for k := 0 to n[i] do
            begin
              t1 := g*(X*H1[i]*j + H2[i]*k);
              t2 := d1 + p1*Q1 + p1*p2*p3[i,j]*d2
                   + p1*p2*p3[i,j]*fsc[i,j,k]*(d3+Q2) +
                   p1*p2*p3[i,j]*p4*Q2;
              t3 := g*(d4 + p3[i,j]*d5 + p3[i,j]*fsc[i,j,k]*d6 – Budget);
              value := t1 + G[i]*(t2+t3);
              if (value <= bestvalue) then
                begin
                  bestvalue := value;
                  bestK := j;
                  bestS := k;
                end; (* if value < bestvalue *)
            end; (* for k *)
        end; (* for j *)
    end; (* CalcSubproblem *)
*********************************************************************
procedure FindOpt;
var
  i, reps : integer;
  gamma, mubar : real;
  ulgamma, lgamma, stepgamma : real;
igamma : integer;
apa, Q1, Q2 : real;
qc, util, h1c, h2c, ec, r1c, r2c : real;
lambdabar : array[1..NumClasses] of real;
flagmu, flaglambda, flaginfeas : boolean;
lastTotalWait, TotalWait, TotalCost : real;
gpf, glp, gp1f : real;
goan : integer;

begin (* 1 *)
flagmu := false;
flaglambda := false;
Lambda[2] := p1*LambdaInitial; (* lower priority class *)
writeln(‘X = ’,X);
writeln(’ ’);
CalcDs;
goan := 1;
repeat
writeln(’ ’);
writeln(’Enter values for budget multiplier (gmma) : ’);
write(’ Lower : ‘);
readln(lgamma);
write(’ Upper : ‘);
readln(ugamma);
write(’ Step : ‘);
readln(stepgamma);
for igamma := 0 to (trunc((ugamma-lgamma)/stepgamma)+1) do
begin (* 3 *)
gamma := lgamma + (igamma*stepgamma);
mubar := 1/(Travel+((1-p2)*Adjust)+(p2*Repair));
lambdabar[1] := 0.0;
reps := 0;
flagmu := false;
flaglambda := false;
flaginfeas := false;
repeat
begin (* 4 *)
reps := reps + 1;
if (reps > 4) then flaginfeas := true;
end
writeln(’ mubar = ’,mubar,’ reps = ’,reps); (*
if (LambdaInitial/mubar*X < 1.0) then
begin (* 5 *)
flagmu := false;
flaglambda := false;
end
(* calculate Q1(mubar,Lambda[1]) & Q2(mubar,Lambda[1]) *)
for i := 1 to NumClasses do

Rho[i] := lambdabar[i]/(X*mubar);
Sigma[0] := 0;
for i := 1 to NumClasses do
    Sigma[i] := Sigma[i-1] + Rho[i];
BigRho := Sigma[NumClasses];
if (BigRho > 1.0) then (* queue explodes *)
begin (* 6 *)
    writeln('Queue explodes in FindOpt ');
    writeln('BigRho = ',BigRho);
    writeln('mu = ',mubar:8:3,' lambda = ',lambdabar[1]:8:3,'
        X = ',X:4);
    flaginfeas := true;
end (* 6 if BigRho *)
else
begin (* 7 *)
(* class 1 is higher priority – return visits *)
Q1 := CalcQWait(2,mubar);
Q2 := CalcQWait(1,mubar);
(* calculate optimal k and s *)
CalcSubproblem(Q1,Q2,gamma, Kit, Depot);
(* calculate true arrival rate *)
p3 := 0.0;
for i := 1 to NumParts do
    p3 := p3 + (G[i]*p3[i,Kit[i]]*p4*1-fsi[i,Kit[i],Depot[i]])
if (abs(Lambda[1] – lambdabar[1]) > epsilon) then
else
    flaglambda := true;
(* calculate true service rate *)
Mu := CalcMu(Kit,Depot); (* measured in hours *)
if (abs(Mu – mubar) > epsilon) then
begin (* 8 *)
    mubar := Mu;
    flagmu := false;
end (* 8 if Mu–mubar *)
else (* close enough *)
    flagmu := true;
(* $$$$$$$$$$$$$ $$ *)
(* writeln(' ');*)
writeln('Mu = ',Mu:8:6,' mubar = ',mubar:8:6);
writeln('lambda = ',Lambda[1]:8:4,' lambdabar = ',lambdabar[1]:8:4);
for i := 1 to NumParts do
    writeln('Kit = ',Kit[i]:3,' Depot = ',Depot[i]:3);
readln(goon);
if (goon = 0) then halt;(*
(*$$$$$$$$$$$$$$*
end; (* 7 if queue does not explode *)
end (* 5 if lambda/mubar*x < 1 *)
else (* lambda/(mubar*X) *)
flagfeas := true;
end; (* 4 repeat *)
until ((flagmu=true)&&(flaglambda=true)) ! (flagfeas=true);
if ((flagmu=true)&&(flaglambda=true)) then
begin (* 9 *)
for i := 1 to NumParts do
KitNonFill[i] := p3[i,Kit[i]];
for i := 1 to NumParts do
DepotNonFill[i] := fsi[i,Kit[i],Depot[i]];
TotalWait := p1*(Q1+Travel+(1−p2)*Adjust+p2*Repair);
for i := 1 to NumParts do
TotalWait := TotalWait + G[i]*KitNonFill[i]*
(DepotNonFill[i]*(tau + Q2 + Travel)
+(1−DepotNonFill[i])*(p4*(tauq+Q2+Travel)+(1−p4)*tauq));
p3 := TotalWait − p1*(Q1+Travel+(1−p2)*Adjust+p2*Repair);
p3 := 0;
fs := 0;
for i := 1 to NumParts do
p3 := p3 + G[i]*p3[i,Kit[i]];
for i := 1 to NumParts do
fs := fs + G[i]*fsi[i,Kit[i],Depot[i]];
r1c := 0;
r2c := 0;
h1c := 0;
h2c := 0;
ec := 0;
qc := 0;
gpf := 0;
glp := 0;
gplf := 0;
TotalCost := 0;
for i := 1 to NumParts do
begin
  gpf := gpf + (G[i]*p3[i,Kit[i]]*fsi[i,Kit[i],Depot[i]]);
glp := glp + (G[i]*(1−p3[i,Kit[i]]));
gplf := gplf + (G[i]*p3[i,Kit[i]]*(1−fsi[i,Kit[i],Depot[i]]));
end;
writeln(’gpf = ’,gpf:5:3,’ glp = ’,glp:5:3,’ gplf = ’,gplf:5:3);
cec := 100*ce*p1*p2*LambdaInitial*HoursPerYear*gpf;
qc := 100*cq*p1*p2*LambdaInitial*HoursPerYear*gplf;
r1c := 100* cn1*p1*p2*LambdaInitial*HoursPerYear*glp;
r2c := 100* cn2*p1*p2*LambdaInitial*HoursPerYear*gp1f;
for i := 1 to NumParts do
  h1c := h1c + (100*X*H1[i]*Kit[i]);
for i := 1 to NumParts do
  h2c := h2c + (100*H2[i]*Depot[i]);
TotalCost := X*FECost + ec + qc + r1c + r2c + h1c + h2c;
util := LambdaInitial*(p1*Travel + p1*(1-p2)*Adjust +
  p1*p2*Repair+p1*p2*p3*(1-fs)*(p4*Travel + (1-p4)*tauq)
  + p1*p2*p3*fs*Travel)/X;
if (TotalWait <> lastTotalWait) then
  OutResults.All(Q1,pa,Q2,TotalCost,ec,qc,r1c,r2c,h1c,h2c,p3,fs,util,
    X,Kit,Depot,gamma);
lastTotalWait := TotalWait;
end; (* 9 *)
end; (* 3 for igamma *)
writeln(' ');
write('Enter 1 to try another set of gammas, 0 to stop : ');
readln(goon);
until (goon=0)
end; (* 1 FindOpt *)
(*============================================================================*)

begin
  Rep := 0;
  reset(InfFile,'fs.dat');
  InputData;
  SetUpOutfiles;
  Budget := LBudget;
  p1 := 1;
  p2 := Lp2;
  p4 := Lp4;
for k1 := 1 to MaxK do
  combo[k1,1] := k1;
for k1 := 1 to MaxK do
  for j2 := 2 to k1 do
    combo[k1,j2] := combo[k1,j2-1]*(k1-j2+1)/j2;
writeln(' ');
write('Enter number of FEs : ');
readln(X);
writeln(' ');
write('Read probs from file? 1=yes, 0=no : ');
readln(fileprobs);
if (fileprobs = 1) then
  begin
    reset(InProbFile,'prob.dat');
    for k1:= 1 to NumParts do
for k2 := 0 to n[k1] do 
  readln(InProbFile,p3i[k1,k2]);
for k1:= 1 to NumParts do
  for k2 := 0 to n[k1] do 
    for k3 := 0 to n[k1] do 
      readln(InProbFile,fsi[k1,k2,k3]);
  end
end
else if (fileprobs = 0) then
  begin
    rewrite(InProbFile,'prob.dat');
    Calcp3;
    Calcfs;
    (* for k1:=1 to NumParts do
    for k2 := 0 to n[k1] do
      writeln(InProbFile,p3i[k1,k2]);
    for k1:= 1 to NumParts do
      for k2 := 0 to n[k1] do 
        for k3 := 0 to n[k1] do 
          writeln(InProbFile,fsi[k1,k2,k3]); *)
  end
else halt;
repeat
  write('Enter the cq parameter: '); 
  readln(cq);
  write('Enter the ce parameter: '); 
  readln(ce);
  write('Enter the tau q parameter: '); 
  readln(tauq);
  write('Enter the tau e parameter: '); 
  readln(taue);
  Rep := Rep + 1;
  FindOpt;
  write('Enter another set ? 1=yes 0=no '); 
  readln(go_on);
until (go_on = 0); 
end.
REFERENCES


