The Spares Provisioning Problem with Parts Inventory

by

Nadim E. Abboud

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in
Industrial Engineering and Operations Research

APPROVED:

Dr. J. N. Daigle, Co-Chairman

Dr. W. J. Fabrycky, Co-Chairman

Dr. S. C. Sarin

Dr. A. G. Hobeika

Dr. J. D. Tew

August 2, 1990
Blacksburg, Virginia
The Spares Provisioning Problem with Parts Inventory

by

Nadim E. Abboud

Dr. J. N. Daigle, Co-Chairman

Dr. W. J. Fabrycky, Co-Chairman

Industrial Engineering and Operations Research

(ABSTRACT)

In this research, we consider the spares provisioning problem, where a finite population of homogeneous machines are being deployed to meet a constant demand. While a machine is operating, it could become inoperable due to the failure of a critical built-in part in the machine. Before repairs on the machine can be initiated, however, a replacement part must be obtained. If a replacement part is available from stock, the machine is immediately transferred to the repair subsystem, in which one or more repair stations operate in parallel. If the replacement part is not in stock, then the machine waits in the ordering subsystem for the arrival of a new part. Once a machine is repaired, it is immediately deployed to meet demand if needed, else it joins a queue of standby machines. The spare machines have zero probability of failure and, if available, a spare replaces a deployed machine immediately upon the latter's failure. The machine operating time, repair time, and ordering time of the parts are assumed to be exponentially distributed.

The ordering subsystem for the parts brings a new aspect to the spares provisioning problem, and dramatically increases its difficulty. This is because the queueing network model which describes the system is a non-product-form network in the case of finite nonzero stocking policy, and specification of closed-form solutions is highly unlikely for such networks. In this dissertation, we present efficient algorithms through which the optimal number of machines, repair stations and stocking level of the parts that minimize total operations costs subject to a service-level constraint can be obtained. The algorithms, which based on Little's result from queueing theory and some
approximate models used for bounding, have proven to be extremely efficient in terms of computer storage and execution time, even for large problems.
Acknowledgements

I would like to express my appreciation to Dr. W. J. Fabrycky for introducing me to the area of repairable item inventory systems, which I find to be extremely interesting and challenging. I also appreciate and thank the committee members: Drs. A. Hobeika, S. Sarin, and J. Tew for their valuable comments and suggestions that improved on the overall quality of this research. I could not have asked for a better committee.

There are not enough words to express my sincere gratitude and deep appreciation to Dr. John Daigle. His role as co-chairman was instrumental in bringing the best out of me, and making me think like a "true" engineer. This work would have never been possible if it were not for his constructive criticism, continual guidance, and support that was extended far beyond academic boundaries.

I have been fortunate enough to be constantly surrounded by good friends. I especially want to mention Fadi Haddad and Jihad Saade whom I consider as brothers, and they treat me as such. I also want to express my thanks to the "Blacksburg Gang", which made my stay at Virginia Tech bearable.
I want to extend a million thanks to my mother, Therese, and my brother, Walid, who provided unlimited moral support and were extremely patient with me. Thanks for always believing in me.

No person has actually suffered through this ordeal more than Jihane Hanania. Her love and support were the constant source of motivation for me. She is a reminder that life is worth living. Thanks, Jihane for all that you have given me.
Dedication

For the man who made it all possible, my father, Elias Khalil Abboud, who passed away before this work came to its fruition. I hope I will never disappoint you.
Table of Contents

1. Introduction ............................................................................. 1

2. Literature Review ...................................................................... 8
   2.1 The Queueing Theory Approach ............................................ 9
      2.1.1 Models with Homogeneous Populations ...................... 10
      2.1.2 Models with Heterogeneous Populations ................. 22
      2.1.3 Models with Life Cycle Cost Analysis .................... 25
      2.1.4 Computational Complexity .................................... 27
      2.2 The Inventory Theory Approach .................................. 29
      2.2.1 Ramifications of the (S-1,S) Inventory Policy .......... 29
      2.2.2 Some Inventory Models Utilizing the (S-1,S) Policy ... 31
      2.2.3 Models with Batch Ordering Policies ..................... 37
      2.3 Summary and Conclusion ............................................ 42

3. The Spares Provisioning Problem ........................................... 44
   3.1 Solution Overview .......................................................... 47
   3.2 Convergence of Iterative Procedure ................................ 51
   3.3 Computation of Aggregate Failure Rate ......................... 53
   3.4 Numerical Examples ..................................................... 62

4. The Spares Provisioning Problem with Zero Parts Inventory .... 67
   4.1 Solution Overview .......................................................... 70
   4.2 Computation of Potentials .............................................. 72
      4.2.1 Algorithmic Solution for Potentials ...................... 78
   4.3 Computation of Aggregate Failure Rate ......................... 81
   4.4 Numerical Examples ..................................................... 85
   4.5 Approximate Model ...................................................... 88

5. The Spares Provisioning Problem with Parts Inventory .......... 95
   5.1 Time Reversibility and Product-Form .............................. 97
   5.2 Numerical Methods ....................................................... 100
      5.2.1 Gauss-Seidel and Overrelaxation Methods ............. 101
      5.2.2 Aggregation-Disaggregation Methods .................. 102
      5.2.3 Carvalho's Method .............................................. 105
      5.2.4 Uniformization .................................................. 106
   5.3 Monotonicity Properties ................................................. 108
   5.4 Solution Methodology .................................................... 113
      5.4.1 Direct Solution Method ......................................... 114
      5.4.2 A Lower Bound .................................................. 115
      5.4.3 Probability of Stockout ......................................... 116
      5.4.4 An Upper Bound ................................................. 118
List of Illustrations

Figure 1. Hierarchy of Assemblies in an End Item. ........................................... 3
Figure 2. Block Diagram for the Spares Provisioning Problem. .......................... 45
Figure 3. The Aggregate Failure Rate as a Function of the Number of Machines in the System. ............................................................. 54
Figure 4. The Aggregate Failure Rate as a Function of the Number of Machines in the System. ............................................................. 54
Figure 5. Flow Diagram for the Spares Provisioning Problem with Zero Parts Inventory. ............................................................. 68
Figure 6. The Modified Markov Chain ............................................................... 74
Figure 7. Transition Flow Diagram for state ......................................................... 76
Figure 8. The Modified Markov Chain with Self Loops. ..................................... 80
Figure 9. Block Diagram for the Spares Provisioning Problem with Parts Inventory. ............................................................. 96
Figure 10. Aggregate Failure Rate as a Function of the Stocking Level. ............... 110
Figure 11. Increase in Availability with Respect to N and S when Ordering Rate is 5 ............................................................. 112
Figure 12. Increase in Availability with Respect to N and S when Ordering Rate is 2 ............................................................. 112
Figure 13. One-to-One Relationship Between Stocking Level and Ordering Rate ............................................................. 116
Figure 14. The Linear Interpolation Method ......................................................... 122
List of Tables

Table 1. Estimates of the Cost of Some Repairable Item Inventory Systems. 2
Table 2. Performance of the Algorithm. 65
Table 3. Performance of the Algorithm with Zero Parts Inventory. 89
Table 4. Performance of the Approximate Model. 93
Table 5. Performance of the Algorithm with Parts Inventory 126
1. Introduction

Inventory systems can be classified into two categories: consumable and repairable. In the consumable case, an item is discarded when it fails. In the repairable case, once an item fails it can be repaired and used again to meet demand. For a variety of reasons that will not be discussed here, attention was focused on consumable item inventory systems for many years, mainly because they constitute the majority of the items that are kept in stock. The value of consumable items in the United States alone, could easily run into the hundreds of billions of dollars.

More recently, attention has shifted towards repairable items. Although there are fewer repairable items in any particular system, the repairable items are usually of high value, and they account for a considerable portion of the inventory investment. Estimates relating to the percentage and/or monetary value of total inventory attributable to repairable items have been made by several researchers. Some of these estimates are summarized in Table 1, where the amounts refer only to the acquisition of the equipment. The maintenance portion of these systems adds significantly to the total system life cycle cost.
### Table 1. Estimates of the Cost of Some Repairable Item Inventory Systems.

<table>
<thead>
<tr>
<th>Researcher</th>
<th>Year</th>
<th>Environment</th>
<th>% of Total Investment</th>
<th>Monetary Value in Billions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schrady [77]</td>
<td>1967</td>
<td>Navy</td>
<td>58</td>
<td>N/A</td>
</tr>
<tr>
<td>Sherbrooke [81]</td>
<td>1968</td>
<td>Air Force</td>
<td>52</td>
<td>10.00</td>
</tr>
<tr>
<td>Muckstadt [67]</td>
<td>1975</td>
<td>Air Force</td>
<td>65</td>
<td>N/A</td>
</tr>
<tr>
<td>DOD [86]</td>
<td>1981</td>
<td>US Military:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Aircraft</td>
<td>N/A</td>
<td>75.77</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ships</td>
<td>-</td>
<td>63.76</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vehicles</td>
<td>-</td>
<td>17.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Communications &amp;</td>
<td>-</td>
<td>3.82</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Electronics Eqnt.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Moore [66]</td>
<td>1983</td>
<td>Non-Farm Business:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Electrical Mach.</td>
<td>N/A</td>
<td>12.87</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Motor Vehicles</td>
<td>-</td>
<td>9.77</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Aircraft</td>
<td>-</td>
<td>5.58</td>
</tr>
</tbody>
</table>

Whether it is the airlines industry, the military, ground transit, the reprographic industry, or any organization that uses units or equipments that breakdown and can be repaired, rented and returned when the lease expires, or need to be overhauled and/or modified after being in operation for a certain period of time, deals with repairable item inventory systems. Therefore, it is paramount that these organizations manage those systems well if they want to continue to survive. Given the significance and the wide applicability of the repairable item inventory systems, the primary objective of this research is to study the operating characteristics of these systems, in order to find ways to design better and more efficient systems.
In this dissertation, we consider a repairable system in which there is a finite number of units that are homogeneous and independent of each other, and deployed to meet a deterministic and constant demand. The units are an assembly of components; these in turn are assemblies of still smaller components, and so on, until eventually one reaches the point where the components are no longer decomposable into finer structure. An illustration of this hierarchy of assemblies is shown in Figure 1.

![Diagram of assembly hierarchy]

**Figure 1.** Hierarchy of Assemblies in an End Item.
In the problem under consideration, the unit (end item) fails when and only when at least one of its subunits fails, and failed units can always be repaired. Once a failure occurs, the unit is immediately transported to the repair facility, where there is at least one repair channel. If more than one repair channel exists, then it is assumed that they are parallel and independent. Before repairs can be initiated, however, the failed component or subunit needs to be identified, removed, and replaced by another component.

There are numerous strategies that can be used in obtaining replacement parts. The general strategy considered here is to inventory $S$ spare parts, $S \in \{0, 1, 2, \ldots\}$, and to order additional parts to replenish stock each time a part is needed. That is, if a unit fails, a part is taken from inventory if any are available and simultaneously an order is placed to replenish stock. If a part is not available from stock, then the unit cannot be repaired until a part arrives from the ordering system. Limiting cases of the stocking policy are $S = 0$ and $S = \infty$. In the former case there is never a part available and the unit always waits for a part to arrive. In the latter case a part is always available and the unit does not wait for a part.

It is of general interest to minimize the total cost of operations, which can be represented by an effectiveness function $f(x_1, \ldots, x_n; a_1, \ldots, a_n)$ that is expressed in terms of the system parameters. These parameters are typically categorized as either controllable variables or fixed parameters. The controllable variables $x_1, \ldots, x_n$ are those over which the system manager has direct control. For example, the number of machines and repair stations the system must own. The fixed parameters $a_1, \ldots, a_n$ are those parameters which cannot be controlled or influenced by the system manager; such as the failure and repair rates of a machine. We should mention, however, that in the stages of the design of the machine, a designer may have available a variety of failure and repair rates from which the parameters of the system may be determined, and minimization of total acquisition and operations costs is possible.

Since a finite population of units is being deployed to meet demand, one of the pertinent performance measures of the system would be the availability of the units, i.e., the number of units
operating out of the total population. How critical this measure is depends solely on the costs of not being able to meet demand. If shortage costs are sufficiently high, then there are various methods one could think of in order to improve availability. The most obvious would be to increase the size of the population, forcing the utilization level of the units to decrease. Another alternative would be to improve on the design of the equipment or unit so that its inherent reliability would be increased, and hence, fails less frequently. These two alternatives would increase the uptime of the units. To reduce the downtime of the units, improvements should be made in the maintenance and supply activities. That is, increase maintainability, increase the stock of spare parts on hand, or increase the capacity of the repair facility by increasing the number of repair channels.

Analyzing logistics problems in a piecemeal manner can lead to some serious consequences. The danger stems from the fact that tradeoffs and interactions exist among the different logistics activities and design characteristics. For instance, if availability happens to be a critical measure, i.e., it is important to meet demand as much as possible, then by just increasing the number of units in the system, does not necessarily increase availability. Intuitively, the higher the number of units, the higher the breakdown rate (assuming the breakdown rate is equal to the breakdown rate of one unit, times the number of units operating). As a result, there is more demand for spare parts. If the stock on hand is insufficient, then more and more units will be waiting for the parts to arrive; otherwise, the failed units will be queued in front of the repair facility to await service. Therefore, it is important to consider all three stages of the system simultaneously.

Chapter 2 presents a literature review on some of the research done on repairable item inventory systems. Given the extensive number of research papers published in this area, we briefly discuss some of the solution methods used to solve these systems under different environments.

In Chapter 3, we consider the spares provisioning problem, which is also referred to as the machine repair problem, subject to an availability or a service-level constraint. Using Little's result and some elementary renewal arguments, an iterative scheme is developed in order to generate the
boundary points of the feasible solution space. A feasible solution point in this case would be some combination of machines and repair stations that meets the availability requirement. From the set of boundary points we then select the point that minimizes operation costs. Results show that the exact algorithm is extremely efficient in terms of computer storage and CPU time, and can handle problems of almost any size.

The spares provisioning system discussed in Chapter 3, probably represents the simplest class of repairable item inventory systems. The problem is analogous to the case where the end item is made up of one critical part which fails at random points in time and needs to be replaced by a new one from inventory. Furthermore, it is assumed that there is an unlimited stock of spare parts (i.e., \( S = \infty \)), and the end item never waits for a part.

With the assumption that a unit has only one critical part, in Chapter 4 we consider the other extreme stocking policy where \( S = 0 \). That is, no spare parts are kept in stock, and every time a unit becomes inoperable, it must wait for a replacement part to arrive before repairs can be initiated. With the use of potentials from Markov chain theory and the extension of the analysis from Chapter 3, we show that this model can be handled just as efficiently as the model with unlimited stock of parts.

In Chapter 5, we relax the restriction on the stocking level of the parts, so that \( S \) can be greater than zero and finite. The model with finite nonzero stocking is considerably harder to analyze than the models discussed in the previous chapters. This is due to the fact that the queueing network model which describes the finite stocking case is a non-product-form network. A brief explanation of the theory behind product-form and non-product-form networks is given, and a discussion of some of the numerical methods we have used to solve for the computationally intractable model is presented. We then analyze the monotonicity properties of the model with respect to the stocking level of the parts, and discuss an approximate model that provides reasonably good lower and upper bounds for the optimal solution. Numerous numerical results are reported to demonstrate the efficiency of the approximate model.
Finally, in Chapter 6, we summarize all the results obtained in the previous chapters, and discuss some areas of research that would complement and extend the results obtained in this dissertation.
2. Literature Review

Since the early 1900's, researchers have been dealing with the problem of repairable equipment systems and ways to optimize their logistic decisions. Logistics deals with the functions and activities associated with the procurement and operational support of equipment systems, and these normally include such activities as maintenance, supply, reliability improvement, and transportation.

The title of the problem has varied from period to period, as more and more logisticians investigated it. In the early stages it was commonly referred to as the machine repairman problem. Then, to include the study of spares equipment, it was enlarged to the machine repairman problem with spares, which was later shortened to the spares provisioning problem. The terms recoverable item inventory control, and repairable equipment system with spares are also common, among others. What has become obvious though, is that over the years, two distinct approaches to the problem have emerged; the queuing theory approach with the "easy" adaptation of the birth-death processes to the machine repairman problem, and later, the inventory theory approach to determine the optimum stocking levels of spares.
Each approach has its advantages and disadvantages, but there is no doubt, that the queueing theory approach is more popular among researchers. This is probably due to the fact that queueing models do not require the usual assumption that there are infinitely many servers at the repair facility which is accurate only when utilization rates are relatively low. Also, they include the additional dimension that one can treat the number of service channels as a decision variable. The shortcomings of this approach, however, is that the computations become too extensive to be implemented in a large-scale system, and the failure and repair times are often assumed to be exponential to alleviate the burden on computations.

The queueing models in this survey have been subdivided into three classes; the models with only one type of units, models with different types of units, and models with aging units. The inventory models were also subdivided based on their ordering policies. Nahmias [68] discusses some of the inventory and queueing models of these systems and presents an excellent review up to 1981. Finally, to avoid any confusion, it should be understood that a unit means the end item (e.g., machine) and a part is what goes into the unit. Also, when discussing the “spares provisioning” problem, the word “spares” refers to stand-by units and not to the inventory of spare parts.

2.1 The Queueing Theory Approach

One popular approach for dealing with the spares provisioning problem is through the use of the machine repair model. This model, which is also referred to as the machine interference problem, can be briefly described as follows. There is a finite population of \( m \) identical machines independent of one another and supported by \( n \) spare machines. Each machine could fail according to
some probability distribution. Furthermore, there is a repair facility which is capable of repairing $r$ machines simultaneously. If all repairmen are busy, the arriving failed machine would join a queue in front of the repair facility until a repairman is freed. It is assumed that repair times are also independent, identically distributed random variables with some probability distribution.

Barlow [7], Cox [24], Page [69], Carmichael [16], and Saaty [75], among others, have investigated the machine repair model for the cases of exponential failures and arbitrary service time distributions, multi-repair channels and exponential service time, single repair channel and constant service time, two types of breakdowns with a single repair channel, each repair channel is dedicated to one type of repair, general repair and running times, and constant repair times by a walking repairman. Explicit results for the problem, however, appear to be known only for the case where the times between failures and repair times are both exponential. Stecke and Aronson [82] survey the literature and classify various interference problems, models and associated assumptions, and solution techniques up to 1985.

Since the machine repairman problem is essentially a special case of a queueing network, a theoretical knowledge of the random flows in networks would be helpful in analyzing the spares provisioning problem. An excellent review of queueing networks is provided by Disney [26], and Disney and Konig [28]. An introduction to the topic by Walrand [88] is also helpful.

2.1.1 Models with Homogeneous Populations

One of the earliest papers to apply the birth-death processes to the spares provisioning problem is Taylor and Jackson [84]. In fact, many of the papers presented later dealing with the above topic, use the same model and modify it by relaxing some of the assumptions that would restrict the early model. The problem can be basically described as a closed queueing network with three nodes. It starts by having $N$ units operating at say node 1, and these units fail probabilistically where the
times between failures are exponential and all the units have the same failure rate (i.e., they are identical). Once a unit fails, it is instantaneously transferred to node 2 to be serviced where it is possible to service \( r \) units concurrently. The service or repair times are also assumed to be exponential, and all servers are identical and parallel to each other. The failed unit is instantaneously replaced by a spare from node 3 (provided, of course, that node 3 is not empty). If the total number of spare units available is \( n \), then at any time \( t \) when the \( N \) units are operating, there are \( n \) units distributed between nodes 2 and 3. Provided that not more than \( n \) units are held at node 2, \( N \) units can continue to operate. However, once \( n + 1 \) units are at node 2, then the units cease operating until a serviceable replacement becomes available from node 2 to restore the operational number to \( N \).

A comparison between the number of spare units available and the corresponding probability of an operational emergency, i.e. the probability that there are \((n + 1)\) units in the servicing node 2 and therefore only \(N - 1\) units available for operations, provides, as the authors suggest, a useful criterion when attempting to solve this problem. After deriving the marginal probabilities of unserviceable units at node 2 at steady-state, the authors have investigated two special cases of the problem. The first case is where the output rate or the service rate at node 2 is equal to the input rate or the failure rate of the units. For the problem when there is an infinite supply of units, the queue length tends to infinity. For the present problem, however, there is a finite number of units, and at most \( n + 1 - r \) units will be waiting for service. Under this circumstance, the authors show that the probability distribution of the number of unserviceable units is a monotonic increasing sequence. Furthermore, the rate of increase of operational efficiency is \( o\left(\frac{1}{n^2}\right) \) so that increasing the number of spare units, \( n \), rapidly cease to have material effect beyond a certain fairly low limit, and as \( n \) increases so does the number of units waiting for service. A more economical alternative is investigated in case 2, where the output rate is greater than the input rate. It was shown that the average number of units waiting for service does not exceed an upper bound which happens to be independent of \( n \). Therefore, increasing the number of spare units does not, in contrast to the
previous case, increase the number of units waiting for service. Their effect is directed only towards increasing the operational efficiency.

Another major contribution to the field of spares provisioning, is presented by Mirasol [65]. Using the queueing theory approach, Mirasol tackles the same problem given in [84] with some important generalizations. The units flowing in the circular queue become more like equipment systems that are merely an assembly of subsystems; these in turn are assemblies of still smaller systems, and so on, until eventually one reaches the point where the components are no longer decomposable into finer structure; i.e., basic parts. Given the hierarchy of assemblies in an equipment system, the author refers to the term unit to any one of the assemblies, regardless of level; and the term subunit will designate those in the next lower level that make up the unit. It is assumed that a unit fails when and only when at least one of its subunits fails. Furthermore, a unit can always be repaired, and repair will consist of identifying, removing, and replacing the malfunctioning subunit. Because of this characteristic of repairability, any unit in the system will be in one of the following states at any point in time.

1. Operable, in inventory
2. Assembled into the next larger unit
3. Failed, waiting for transportation
4. In transit to repair shop
5. Waiting for repair
6. Diagnosis and removal of failed subunit
7. Waiting for serviceable subunit
8. Replace failed subunit
9. Operable, waiting for transportation
10. In transit to inventory location

The evolution of a unit through the above states can be represented by its flow through a multi-stage, circular queueing system. The flow through the first stage (labeled 'operating') represents the
transition of a spare from state 1 to state 3. Stages 2 and 5 are the two transportation stages and they represent states 3, 4, 9, and 10, where states 3 and 4 correspond to stage 2, and states 9 and 10 correspond to stage 5. These stages can be multiple server facilities, and may also involve bulk types of service. The flow time through stage 4, which is the repair stage, is the time that a unit spends in states 5, 6, and 8, where as mentioned earlier, the time in state 6 and 8 is the repair time for the unit. The time spent in state 7, which is the time spent waiting for subunits, is represented by parallel facilities, which will be collectively referred to as the waiting stage (the number of parallel facilities corresponds to the number of subunits in the unit). This stage can be bypassed by the flowing units if the subunit required is actually available.

It should be pointed out that the waiting stage provides a method of linking the different units in the equipment system. The linking process is briefly described by Mirasol as follows: "...the unavailability and duration of unavailability of the system depend on the unavailabilities and durations of unavailability of the subsystems, which in turn depend on the unavailabilities and durations of unavailability of the assemblies, etc. These unavailabilities and durations of unavailability naturally depend on the logistics decisions that are taken."

After making the critical assumption of Poisson failures and exponential service times, the birth-death process was applied to the finite queue described earlier, and steady-state probabilities for the queue lengths at each stage were obtained. An optimization procedure was also presented, and the only decisions included were those concerning the number of spares and the number of repair channels to provide for each of the units in the system. The objective function is cost based associated with a unit and is linear in the number of spares and the repair channels.

Defining strategic unavailability, \( \psi \), as the product of the system unavailability rate and the mean duration of the system unavailability, Mirasol, in searching for the local minimum, had to balance the total system cost and \( \psi \) which is assumed to be a monotone function of cost. His conclusion was that local optimality can never be reached unless there is a balanced workload among all the parallel facilities in the waiting stage.
Mirasol’s results would certainly agree with those of Gordon and Newell [36] which appeared four years later. The classic work presented in [36], is the study of closed queueing networks with exponential servers. Gordon and Newell have found that as the number of units or customers in the network gets larger and larger, the distribution of customers in the system is regulated by the stage with the slowest effective service rate. Assuming, of course, that the system is at steady-state. This result, however, raises another point which could severely limit the applicability of not only the model presented in [65], but any model that assumes the service times at any node in the network, are independent identically distributed according to the negative exponential distribution. Since the exponential distribution has the “memoryless property”, this guarantees the independence of the time intervals between failures of the units, and that the intervals are identically distributed. This also implies, the process which measures the number of failures of the unit is a Poisson process. For a closed network with a finite number of units, however, the Poisson process does not hold since there is a dependency on the number of units operating, i.e., the failure rate is equal to the failure rate of one unit times the number of units operating. It has already been proven by Burke [14], Disney et al. [27], and Melamed [64] among others, that the flow in queues which allow for the customer to return to a node that he has previously visited is not only not Poisson, but under certain conditions, it is not even renewal. This whole problem of “Poissonness” can be avoided, of course, if we let the number of units in the network tends to infinity, or at least large enough with low failure rates, so that the overall failure rate would practically remain constant. How large is large enough, would be a little difficult to assess.

An interesting variation of the spares provisioning problem is given by Gross et al. [43]. The objective of their research is to determine an adequate number of spares and repair channels for replacing and repairing units which fail randomly. Unlike the other “static” models, which treat the problem in steady-state and allows no variations in the population sizes of the units and repair channels, the model in [43] considers a multiyear planning horizon, allowing for growth both in population sizes and unit reliability. A service level constraint of 10% is imposed on spares stockouts; that is, at least 90% of the requests for spares are immediately filled from on-hand spares.
inventory. They refer to this availability criterion as the fill rate. Therefore, instead of determining what the fill rate will be, given the number of spares and repair channels (such as in the classical machine repairman problem), the objective in [43] then becomes one of minimizing expenditures for spares and repair channels subject to the fill-rate constraint. Some of the assumptions made pertaining to the model are the following: The population is in steady-state at its average size and reliability for an entire year, changing instantaneously to a new steady-state position at new average values at the beginning of each new year, and, the times between failures of the units is exponential, and the service times which include removal, transportation, and repair times, are also exponential.

The assumption that all components operate at an average failure rate was investigated by developing an exact model for one specific example. It was shown that with respect to availability, the average failure rate yielded more conservative results, and even for sizable differences in the failure rates of different components, the approximate availability was close to the actual. The authors, by using intuitive reasoning, explain the small difference in availability as follows: "... since components having higher failure rates would be expected to fail (and hence be in repair) more often. Thus, the more reliable components operate a larger proportion of the time, so that the actual failure rate for the system would tend to be lower than the arithmetic average of the failure rates over all components." It should also be noted that by imposing a fill-rate constraint of 90% and thus, causing "light traffic" in the network, is essential in preserving the Poisson process. The constraint was equally important when the authors later modified the model by separating removal, transportation, and repair into three service stations and assumed there is an infinite supply of units. As a result, the problem was transformed into a series of M/M/c queues with a solution that is easily tractable.

No integer programming algorithm has been successful in solving the integer-nonlinear programming problem when the model in [43] was reported. A heuristic method was utilized instead. The algorithm considers explicitly only two cost categories; purchase cost of spares and purchase cost of service channels. Furthermore, the heuristic considers only one year at a time, using the
"best" combination it finds for the current year as the starting point for its search for the best combination for the following year.

Under the same assumptions as in [43], that is, exponential failure and service times, Gross and Ince [41] formulate the problem as a cyclic queue for which an exact solution is tractable. The objective in [41] is mainly to compare the accuracy of the approximate method vs. the exact one, and to see if there are any significant reductions in computing time. The cyclic queue representing the system includes four stages, where the first stage represents the operating machines with \( M \) servers. When machines fail, they go directly to stage 2, which is a removal phase. After removal, machines go to stage 3, labeled the transportation phase, then to stage 4 which is the repair phase. After a machine is repaired, it returns to stage 1 ready for service. If less than \( M \) machines are operating, the newly repaired machine goes directly into service. If all \( M \) machines are "up", the newly arrived machine either starts a queue or joins an existing one in front of stage 1. This queue represents the spare machines on hand. It is assumed that there are ample servers at stages 2 and 3, so no queue will build up in front of either stage.

The system's performance measure will again be the availability of spares, which is basically, the probability that there is a queue in front of stage 1. It should be noted here, that in closed queueing networks, one must be careful when computing the steady-state probability of availability and differentiate between general time probabilities (at any arbitrary point in time), and conditional probabilities (embedded at the times of failure). This distinction must be made, since it is the failure of a unit in the system that generates a request for a spare. Let \( P(n_t) \) be the general time probability of \( n_t \) customers at stage 1, and \( Q(n_t) \) be the probability of \( n_t \) customers at stage 1 given a failure is about to occur. Thus,

\[
AVAILABILITY = \sum_{n_t = M+1}^{N} Q(n_t),
\]

[2.1]

2. Literature Review
where \( N \) is the total number of machines in the system, and \( M \) is as defined earlier. An algebraic expression for \( Q(n_i) \) in terms of \( P(n_i) \) can be derived using Bayes' theorem (see [43], p. 524), and \( P(n_i) \) can be obtained from the theory of cyclic queues.

Given an \( M/M/c \) queue with input parameter \( \lambda \), Burke [13] was able to prove that the output from that queue was Poisson with parameter \( \lambda \) independent of the service rate, as long as the service time is exponentially distributed. This result can be extended to a series of \( M/M/c \) queues with \( \lambda \) in at the first stage and \( \lambda \) out at the last stage. Therefore, at steady-state the stages would behave as if they are independent of each other. To have a good approximate model based on Burke's result, Gross and Ince assumed that the first stage is almost always operating at full capacity, so that it acts like an infinite source input to the rest of the system with an output process that is Poisson. This assumption hinges on the constraint which imposes high availability of spares. In comparing the exact and approximate models, after the marginal probabilities were obtained, the authors were able to show mathematically that

\[
Q_E(n_i) \geq P_E(n_i) \geq Q_A(n_i),
\]

where the subscripts "E" and "A" are added to indicate the exact and approximate models respectively (note that there is no need to look at \( P_A(n_i) \) since for \( M/M/c \) queues, it is well known that \( P_A(n_i) = Q_A(n_i) \), see for example [40]). Therefore,

\[
\sum_{n_i = M+1}^{N} Q_E(n_i) \geq \sum_{n_i = M+1}^{N} P_E(n_i) \geq \sum_{n_i = M+1}^{N} Q_A(n_i).
\]

The inequalities above show that if one uses the approximate or the general exact approach to determine the system availability, the true system availability will actually be greater than the original target. The authors also showed, that as long as the availability constraint is satisfied, the approximate model should be used because it is quite accurate and computationally more efficient than the exact model. Even if the constraint is not satisfied, the exact model with general time proba-

2. Literature Review
bilities is accurate with a sizable reduction in computing time, if the population sizes are not very small. Otherwise, the exact model with failure point probabilities is required.

An extension of Mirasol's type of model has appeared in the literature almost twenty years later. The work presented by Gross et al. [45] extends the model in two directions: (1) a finite population of operating times is treated and (2) a two-echelon repair system (field and depot) is considered. The system can be viewed as a network with three nodes, where the first node is referred to as node U (for "up" or operating). It is desired to have $M$ units operating at all times, and if a unit fails, then there is a probability of $\alpha$ that it will be diagnosed as base repairable and sent directly to base repair, or node B. Otherwise, with a probability of $1 - \alpha$ the failed unit will be sent directly to depot repair which is denoted by node D. Of those sent to base repair, a further fraction $\beta$, after undergoing service, cannot be fixed and are sent to depot repair. Finally, holding times at all nodes are assumed to be independent exponentially distributed random variables. The reader should note that if $\alpha$ and/or $\beta$ are strictly less than one, then the resulting network will not be a cyclic queue.

An optimization procedure is presented in order to minimize the total costs subject to an availability constraint. The objective function consists of three cost components; the cost of the spares (when more than $M$ units are purchased), the cost of the base repair channels and the depot repair channels. The unit costs include annual operating costs and capital investment amortization of a spare or a repair channel. The authors have determined the optimal spares level and repair capacities as efficiently as possible by using Buzen's algorithm [15] to compute the normalizing constant, which is needed in order to obtain the joint queue length probabilities at steady-state.

An implicit enumeration scheme introduced by Lawler and Bell [58] (L-B) was also used since the decision variables are integer-valued. The L-B algorithm, however, requires that the objective and constraint functions each be expressible as the difference of two monotonic functions of the decision variables, and the authors were able to verify that property.

A simplified version of the model given in [45] was analyzed by Madu [63]. Madu's model differs basically in three parts. First, when a unit fails, it is sent to node B (i.e., the base) where a
diagnostic check is made, and if it requires a major repair, then it will be transferred to node D (i.e., the depot). As a result, direct transition from node U to node D is not allowed. Second, without using any implicit enumeration schemes, Madu was able to determine the optimal number of spares using Buzen's algorithm only. The optimization problem, however, may not be too stimulating, because it did not incorporate any constraints, such as availability or budgetary constraints. The objective function then, was reduced to balancing the linear cost of spares with the linear cost of lost production, much like the simple EOQ problem.

Albright and Soni [3] studied a system very similar to the one presented by Gross et al. [45]. The emphasis in their work, however, is on the solution procedures for the steady-state probabilities of the system, rather than on optimization. Two models are considered in [3], where the first model is basically the same as the one in [45]. The only difference is that they do not allow units to go from base repair facility to the depot. The system was modeled as a continuous-time Markov process with a multi-dimensional state space and developed an exact solution procedure based on aggregate/disaggregate methods. The algorithm requires only the solution of a series of one-dimensional birth-death systems. As a result, the algorithm is very easy to solve and computationally very efficient.

To show briefly how the aggregate/disaggregate method works, let \( N \) be the number of identical units at the base where as many as \( J \) of them can be working at any given time. As in [45], if more than \( J \) units are at the base, then the rest of the units act as spares. Each unit is assumed to have an exponential failure rate of \( \lambda \). When a unit fails, it is classified as base repairable with probability \( q \). Otherwise, with probability \( 1 - q \), it goes to the depot for repair. The number of repairmen at the base and depot are \( R \) and \( R_d \) respectively, with exponential repair times of rates \( \mu \) per base repairman, and \( \mu_d \) per depot repairman. The state space of the Markovian system is of the form \((k,m), 0 \leq k \leq N - m, 0 \leq m \leq N\), where \( k \) is the number of units currently in the base repair facility, and \( m \) is the number of units in the depot.

2. Literature Review
Define the set \( I_m = \{(k,m) | 0 \leq k \leq N - m\} \) for any fixed \( m \). If transitions out of this set are ignored, that is, transitions where \( m \) increases or decreases by 1, the system acts as a one-dimensional birth-death process on \( k \). In state \( (k,m) \) the birth rate is \( \min(J,N - k - m)\lambda q \) and the death rate is \( \min(R,k)\mu \). Let \( p_{k|m} \) be the steady-state probabilities for this birth-death process, and define \( L_m \) as

\[
L_m = \sum_{k=0}^{N-m} \left[ \min(J,N - m - k)\right] p_{k|m}.
\]  

[2.4]

Now consider the one-dimensional birth-death process on \( m \) with birth rates \( L_m\lambda(1 - q) \) and death rates \( \min(R,m)\mu \). If \( p_m, 0 \leq m \leq N \), are the steady-state probabilities for this process, then it can be shown that \( p_{k,m} = p_{k|m}p_m \) is the steady-state probability of being in state \( (k,m) \) for the single-base problem. Furthermore, \( p_{k|m} \) is the conditional steady-state probability of having \( k \) units in base repair given that \( m \) are in the depot, and \( p_m \) is the marginal probability of having \( m \) units in the depot.

The second model that was studied, allows for more than one base with a base repair facility for each base, and a central repair depot. Even though the aggregate/disaggregate method used to develop a solution procedure provides very good results, it is not an exact procedure. This type of approximation was also used on another model by Albright and Soni [4], where they have allowed for the possibility of a failed unit to be classified as irreparable, resulting in the depletion of overall inventory. The main interest in [4] was the interaction between the repairable inventory system and the procurement of new units to replace the condemned ones. It was already determined in [2] as to what the optimal procurement policy should be for the units.

Carvalho [17] considered a system of homogeneous population of \( N \) repairable units deployed to meet a deterministic and constant demand, \( D \). If \( n \) units are available or "up", and \( n \) is greater than the demand, then only \( D \) units will be operating, and the remainder \( (n - D) \) units will be kept in standby. If an operating unit fails, it will be immediately replaced by one of the standbys if one is available; otherwise, a shortage situation will occur. A failed unit will require a spare part before

2. Literature Review
it goes to repair. If one is available in stock, then repairs will immediately start provided one of the 
$M$ parallel repair channels is available; otherwise, the unit waits in queue for service. If a stockout 
of spare parts exists, then the unit waits until the next order arrives. Upon completion of repair, 
the unit is instantaneously transferred back to the first stage, where it starts operating if the number 
of units up is less than $D$; otherwise, it joins the pool of spare units in standby. The times between 
failure of an operating unit, and repair times are assumed to be exponential. Also, a one-for-one 
ordering policy, designated as an $(S - 1, S)$ policy, was chosen for the spare parts inventory policy 
with exponential lead times. This is equivalent to a queueing system with ample servers.

Carvalho modeled the system as a two-dimensional Markov process. With the 2-tuple $(n,s)$ 
ordering system, where $n$ is the number of units available to meet demand and $s$ is the number of 
orders outstanding, it is sufficient to represent all the possible states that the system could be in. Neither the marginal process that describes the number of units available, nor the marginal process 
that describes the number of orders outstanding is Markovian. Only the joint two-dimensional 
process has the characteristics needed to be Markovian.

Optimization of the system involves a balance between the number of units in the population 
($N$), the number of repair channels ($M$), and the level of spare parts to maintain in stock ($S$). A 
computer algorithm was developed which uses a simple search technique in the tridimensional 
space $(N,M,S)$ in order to minimize total cost. The algorithm, however, does not guarantee global optimality.

Finally, it is interesting to note, that by modifying the state space, Carvalho modeled the system 
as a quasi-birth death process that produces a block tri-diagonal transition matrix $P$. Therefore, 
by inverting a certain diagonal submatrix from $P$, the whole system of equations can be solved in 
one pass. Unbeknown to Carvalho, the technique turned out to be a special case of the 
aggregation/disaggregation method; see for example Takahashi and Takami [83]. For more on 
aggregation/disaggregation methods, the reader should refer to Schweitzer [78,79].
2.1.2 Models with Heterogeneous Populations

As is evident from the review, so far, there is a sizable literature on the spares provisioning problem with homogeneous units. This is probably due to the fact that when considering different types of units in a first come, first served discipline, it is necessary to keep track of where each individual unit is in the system. This would suggest that even with a moderately sized problem, the state space describing all the possible states of the system, would simply be unmanageable. Furthermore, none of the models presented below, dealing with heterogeneous populations, considered the hierarchical structure given in [65]. They treat different classes of customers or machines as separate entities.

Posner and Bernholtz [71] analyze a model, which is not directly related to the spares provisioning problem, but does have some attractive properties that would be useful for the problem in interest. The model can be briefly described as a closed, finite, queueing network having several classes of units, that is, the units are assigned to various groups, the units within a group are identical, and those in different groups having possibly different service parameters, travel-time distributions, and routing procedures. For convenience, the authors assumed that each service station is represented by a single exponential channel.

Posner and Bernholtz avoided the task of keeping track of the specific order of units of different classes in a queue, by assuming that the service rate for a particular class was dependent on the total number of units of that class, and the total number of units of all classes at that stage. The service rate then, would be obtained by using the mean of a hypergeometric distribution. The trade-off, of course, is that the solution is approximate rather than exact.

Motivated by the conception of a computer system as a network of processors and a collection of jobs, Baskett et al. [8] considered the problem of heterogeneous units, and obtained solutions for four service disciplines: first-come, first-served; processor sharing, where each customer gets a share
of and is serviced simultaneously by a single server; no queueing (ample server); and last-come, first-served. With the FCFS discipline, the authors could not handle the problem without considering the ordering of jobs, and assumed instead that all jobs have the same service time distribution.

Kelly [55,56], is credited for setting up a notational structure for, and possibly formulating the most general network of queues, which takes into consideration different types of customers. His formulation can handle the FCFS discipline, however, the notation is brutal, and a numerical application of the results would still involve the generation of and accounting for the vast state space of all the possible orderings in queues.

Gross and Ince [42] had some success in limiting the state space when a heterogeneous population exists. They were able to develop an algorithm for the generation of states which uses a "semi-brute force" approach that has the advantage of exhaustiveness, but can also take into account the structure of the system to cut down on cases to check. Gross and Ince considered only the two-stage system (operating and repair-the classic machine repair problem) with \( N \) of type \( l \) and \( M \) of type 2 with \( c_1, c_2 \) servers at stages 1 and 2, respectively. With the exponential servers, a stationary Markov process, \( X(t) \), can be constructed whose transition probability matrix has elements:

\[
P_{ij}(t) = Pr(X(t) = j \mid X(0) = i).
\]  \[2.5\]

Kelly in [55] has already proven that this Markov process is irreducible and that if an equilibrium distribution can be found, then it is unique. Letting \( \Pi \) denote the steady state probability vector with elements \( \pi_i \), being the steady state probability that the system is in state \( i \), \( i = 1,2,...,I \), where \( I \) is the total number of unique and feasible states, the \( \pi_i \)'s can be found by solving the system of equations

\[
\Omega = \Pi Q,
\]  \[2.6\]
where $O$ is a row vector of 1's, and $Q$ is the $I \times I$ infinitesimal generator matrix which can be easily obtained once the transition probability matrix is found.

Since one of the equations in the above system is redundant, due to the constraint that all the $\pi_i$ elements must sum to unity, an equivalent system of equations would be

$$B\pi' = C,$$  \hspace{1cm} [2.7]

where $B$ is $I \times I$ and is the transpose of $Q$ with the last row replaced by 1's, $\Pi'$ is the transpose of $\Pi$, and $C$ is an $I \times 1$ column vector and consists of all zeroes, except for the last element, which is one. To get the steady state probabilities then, the system of equations should be multiplied by $B^{-1}$ inverse on both sides, yielding

$$\Pi' = B^{-1}C.$$  \hspace{1cm} [2.8]

Since the vector $C$ is all zeroes, except for the last entry which is 1, the solution would be the last column of $B^{-1}$.

To cut down on the computations, two approximate models were considered. In the first approximate model, for each stage, a weighted average of service rates of the two classes was taken, so the problem would be reduced to the classic machine repair with spares. To justify this approximation, consider a system with a large number of servers, each always busy, outputting one class of customers, either type 1 or type 2. The overall output rate from the multiserver stage is equivalent to the overall output rate of a system where each server has the average service rate of the original system.

The second approximate model averages holding times rather than service rates. The rationale behind this approximation is as follows. Consider an infinite queue in front of a single-server, one-stage system, where the queue consists of alternating type 1 and type 2 customers. The average output rate from the single server would then be the inverse of the average service times for the two types, where each service time itself is the inverse of that type's service rate. Since averaging service

2. Literature Review
times seems to imply a single server, and averaging service rates seems to imply infinite servers, the two approximate models represent the extremes of the machine repair problem, and it is best to consider both.

After running numerous cases, the authors have concluded that the approximation using rate averaging underestimates machine availability, as it tends to behave as though all machines are running simultaneously with regard to failure, while the time averaging approximation overestimates availability, as it tends to behave as though only one machine is operating. The authors further state that the approximations for the most part, gave tolerable errors.

2.1.3 Models with Life Cycle Cost Analysis

When analyzing the spares provisioning problem, researchers often assume that the units in the system have an infinite life and invariant failure rates. Not much attention has been paid to the life cycle cost analysis of the problem, even though, this could have a major economic impact in the airlines industry, the military, ground transit, and virtually any system that operates units which deteriorate over time and would eventually have to be thrown away after they become costly and inefficient.

Hart [47], and Fabrycky and Hart [30] analyzed a model with a finite population of units deployed to meet a constant demand. As the units fail, they are repaired and returned to service, and as they age, they become less reliable and their maintenance cost increase. Therefore, it is important to determine the optimum replacement or throw-away age. It is assumed that the number of new units procured each year is constant and that the number of units in each age group is equal to the ratio of the total number required in the population and the desired number of age groups.

Two major design problems were treated. In the first design problem, the design process consists of specifying a population of units, a number of maintenance channels, and a replacement
schedule for bringing new units into the system. In the second design problem, the design process is extended to include selecting the unit's reliability and maintainability characteristics. A computer algorithm was developed to perform the calculations and to determine optimum values for system design variables. Since the objective is to minimize total system cost, a search technique was used to locate that minimum. By assuming that total system cost as a function of the number of channels is unimodal, a search technique was used to find the number of channels giving the minimum total cost for a particular population size and number of age groups. The total system cost is the sum of three cost components; the annual equivalent cost of the units, the annual equivalent cost of the repair facility, and the cost of not meeting demand.

Hinger [48] extended Hart's model to include two repair levels. The severity of the failure will determine which level of maintenance (intermediate or depot repair level) the unit will enter. The primary objective of the model is to determine the optimum throw-away age, the total number of units in the system, and the number of repair channels at each level in order to minimize total system annual equivalent cost. A computer algorithm was developed to determine the optimal solution, and sensitivity analysis was performed on the reliability and maintainability characteristics of the units, the percent of the total population assigned to each repair level, the depreciation method used in determining the book value of the units, the interest rate, and the repair time at one of the repair levels. A constrained version of the model was also investigated. Hinger allowed for the possibility of replacing units at an early age, and limiting the population size due to scarcity of space.

Banks and Fabrycky [6] present a number of models that are similar to the one given in [47]. Models are first derived for the assumption of deterministic age-invariant failure rate and deterministic age-invariant repair rate for the single-source system. These deterministic models are then extended to the multisource case. The probabilistic failure rate and repair rate cases are then treated for both the single-source and multisource systems. The policy variables are the procurement source for the units that constitute the population, the number of units, the retirement age, and the number of repair channels. The models in [6], and all the previous models, were treated
at steady-state. Repairable item inventories normally come into being over a non-steady-state buildup phase. They then operate over a steady-state interval of years, after which a phase-out period is entered.

Moore [66] has extended the work in [6] to include more than a single repairable equipment population. Moore refers to the models in [6], [30], [47] and [48] as a single repairable equipment and logistic systems (REAL), as supposed to the multiple repairable equipment and logistic systems (MREAL), which are basically a collection of more than one REAL system, each held to meet a different demand. For each REAL system in the MREAL system, the model attempts to answer the following questions simultaneously.

1. How much capital should be invested in each REAL system?
2. How many units of repairable equipment should be procured in each REAL system?
3. How many repair channels should be provided for each REAL system?
4. At what age should the equipment in each REAL system be routinely retired?
5. From what source should the equipment be procured for each REAL system?

Moore attempts to answer all the above questions with respect to two MREAL performance measures; the system-wide equipment availability achieved in each individual REAL system, and the total annual equivalent costs of the MREAL system, which include procurement and operation of the repairable equipment, procurement and operation of the repair facilities, and shortage costs.

2.1.4 Computational Complexity

Before proceeding to the next section, it is worth mentioning that efficient algorithms have been developed to compute different performance measures for closed queueing networks. To illustrate the complexity involved in computing the mean queue lengths, consider a network with $M$ nodes and $N$ indistinguishable items. The state of such a network can be described by a vector
\( \tilde{n} = (n_1, n_2, \ldots, n_M) \) where \( n_i \) is the number of items present at the \( i \)th node \( (\sum_{i=1}^{M} n_i = N) \). Therefore, the total number of states is the number of permutations of \( N \) indistinguishable objects and \( M - 1 \) indistinguishable objects, namely, \( \frac{(N + M - 1)!}{N!(M - 1)!} \). Even for the relatively modest case in which \( M = 8 \) and \( N = 20 \), the calculation of the normalization constant, needed to compute the probabilities, requires the summation of 888030 terms, each of which is the product of 8 factors. This is assuming the nodes have exponential service times, and so the solution is a product form solution. See [36] and [73].

A well known algorithm, known as Buzen's algorithm [15], has been developed in the early 1970's for evaluating the normalization constant and the marginal distribution of queue lengths. The algorithm assumes that the routing probabilities, that is the probability that an item leaves a node and goes to a certain node, remains constant and is known in advance, all service times are exponential, and no spares needed to initiate the repair on an item are kept in stock.

Bruell and Balbo [12] extended Buzen's algorithm to include more than one class of items. Each class could have its own routing probabilities, and it is possible for an item to change classes. Besides discussing the normalization constant approach, Bruell and Balbo also discuss the mean value analysis approach, which is easier to understand from an intuitive point of view than the former approach. The mean value analysis algorithms are slightly more computationally efficient than the normalization constant algorithms, and they require less storage provided that the implementation strategies given in [12] are followed. (See Reiser and Lavenberg [72] for more details on the mean value analysis approach.)
2.2 The Inventory Theory Approach

Many of the repairable item models presented below are descriptions of a two-echelon system which operates in the following manner. Items fail at the base level (echelon 2). They are either repaired or shipped to a central depot (echelon 1) if the repair is too complex to be carried out at the base. The base replaces the failed item from base-level stock (if it is available) and when shipping an item to the depot immediately places an order for a replacement from the depot. Define the inventory position at the base as the total number of units on hand plus units due in from base and depot repair minus backorders. The base maintains its inventory position at a fixed level, $S$, operating according to an $(S - 1,S)$ policy. This policy can be briefly explained as follows: each time one or more units is demanded, the inventory position drops below $S$ and an order for an equal number of units is placed. When the net inventory (on-hand minus backorders) becomes negative, a backorder condition exists. The policy is usually implemented in systems with very expensive items that have a low demand rate. (See Hadley and Whitin [46] for further discussion on the $(S - 1,S)$ inventory policy.)

2.2.1 Ramifications of the $(S-1,S)$ Inventory Policy

An analogy can be drawn between the $(S - 1,S)$ model and the theory of infinite server queues. At each placement of an order, one may think of a customer entering service in a system with an infinite number of servers. The number of busy servers corresponds to the number of outstanding orders. In his classic paper, Palm [70] has shown that if customers arrive according to a stationary Poisson process and service times are independent identically distributed random variables with a finite mean, then the steady-state probability distribution of the number of busy servers is Poisson, independent of the form of the service distribution. In the inventory problem this implies that if
customer demands are generated by a stationary Poisson process, the steady-state number of outstanding orders is a random variable having a Poisson distribution, regardless of lead times as long as lead times are independent. That is,

\[
Pr(x \text{ orders outstanding}) = Pr(\text{net inventory} = S - x) = e^{-\lambda \tau} (\lambda \tau)^x / x!, \quad x = 0, 1, 2, \ldots
\]

where \( \lambda = \) mean number of units demanded per unit time and \( \tau = \) expected lead time.

This result is based on the assumption that excess demand is backordered. In the case of lost sales, a similar result holds except that the Poisson distribution is truncated at \( S \).

An important generalization of Palm’s theorem was obtained by Feeney and Sherbrooke [31] who showed that the distribution of outstanding orders is still Poisson when demands are generated by a stationary compound Poisson process. That is, the times of demand occurrences still follows a Poisson process with rate \( \lambda \), but at each demand occurrence the number of units demanded is a random variable with an arbitrary discrete distribution.

The compound Poisson has been described by many authors such as Feller [32] with the following as its most salient features:

1. Any compound Poisson distribution has a variance that exceeds or equals its mean. When the variance equals the mean the compound Poisson reduces to the simple Poisson.
2. The compound Poisson distributions are the most general class of “memoryless” discrete distributions, i.e., the number of demands occurring in a time period does not influence the probabilities of demand in any other non-overlapping time period.

One of the reasons Feeney and Sherbrooke were interested in generalizing the assumption of Poisson demand to distributions with a larger variance is that, as Sherbrooke [81] explains, the de-
mand data usually produces variances that exceed the means. Furthermore, the physical model of the customers who can order several units appear to be a reasonable description of many supply processes. One of the drawbacks of the generalization, however, is that there is only one resupply time for a customer's demands. In the spares provisioning problem this translates to many units being repaired at the same time.

Gross and Harris [39] describe several one-for-one-ordering inventory models in which the time required for order replenishment, or leadtime, depends on the number of orders outstanding. Demand is assumed to be a Poisson random variable with a constant mean $\lambda$, and leadtime is assumed to be state-dependent in either one of two ways: (1) the service-time contribution to the leadtime is an exponentially distributed random variable with mean $\mu(m)$, where $m$ is the number of orders outstanding embedded at the times of fulfillment of orders, and (2) the instantaneous probability at an arbitrary point in time of an order being filled in an infinitesimal interval of time $\delta t$ is $\mu(n)\delta t$, where $n$ represents the number of outstanding orders. The models in which $\mu(m)$ is used are analyzed using an imbedded Markov chain approach, while the models utilizing $\mu(n)$ are treated by a birth-death analysis. After determining the optimal stocking level $S$, which minimizes holding and backordering costs, it was found that the two types of analysis were very similar.

2.2.2 Some Inventory Models Utilizing the (S-1,S) Policy

Sherbrooke [81] reported the METRIC model which was developed at the RAND Corporation. METRIC became a classic because it appears to capture many of the significant features of the problem of determining suitable spares levels in a large-scale repairable item inventory system, and it is probably the first and one of the few multi-echelon, multi-item models to be implemented.

METRIC, is basically a mathematical model translated into a computer program, with its major purpose is to determine the optimal base and depot stock levels for each item, subject to a
constraint on system investment or system performance. The objective will be to minimize the sum of expected backorders on all recoverable items at all bases. It is assumed that the penalty depends linearly on the length of the backorder and the number of backorders. Some of the additional assumptions made are the following: 1) demands for item \( i \) at base \( j \) are generated by a stationary compound Poisson process with rate \( \lambda_j \) and a logarithmic Poisson process being the compounding distribution, so that the number of demands in a given interval is negative binomially distributed, 2) with probability \( r_j \) a failed item of type \( i \) at base \( j \) can be repaired at the base. With probability \( (1 - r_j) \) the item must be repaired at the depot, and 3) there is no lateral resupply or transshipment among bases.

The assumption of no transshipment among bases, guarantees the independence of each base-to-depot demand process. Furthermore, given that each failed unit either goes to the depot or remains at the base for repairs is described by a Bernoulli process, it is safe to assume that the demand at the depot level is also compound Poisson. This is due to the superposition principle of Poisson processes, which is explained by Çinlar [20] as follows. Given two Poisson processes \( L \) and \( M \), independent of each other, with respective rates \( \lambda \) and \( \mu \), the superposition of the processes is a Poisson process \( N \) with rate \( v = \lambda + \mu \). Conversely, with a Bernoulli process, a Poisson process can be decomposed into two independent Poisson processes. The two principles can be easily extended to any number of processes and could be applied to the more general compound Poisson process.

Muckstadt [67] presented a model which he called MOD-METRIC, and it is basically an extension of the METRIC model given in [81]. MOD-METRIC permits the explicit consideration of a hierarchical parts structure. For example, a major assembly may consist of a casing and several components. The components are subordinate to the assembly in the parts hierarchy. The objectives of the model are to describe the logistics relationship between the components and the final assembly, and to compute base and depot spare stock levels for all items with explicit consideration of this logistics relationship.
The significant difference between METRIC and MOD-METRIC is in the manner in which one computes the mean base repair cycle time for the end item at base $j$. Muckstadt expressed this as $R_j + \Delta_j$, where $R_j$ is the average repair time when items are available and $\Delta_j$ is the average delay due to the unavailability of items at base $j$. By utilizing the classic Little's formula $L = \lambda W$ from queueing theory (which says that the expected queue length is the product of the arrival rate and the expected waiting time of an entering customer), the expected delay or waiting time at base $j$ due specifically to the unavailability of item $i$, say $\Delta_{ij}$, is given by

$$\Delta_{ij} = \frac{\beta_i(S_{Sj}, S_{ij})}{\lambda_i},$$  \hspace{1cm} [2.10]

where $\beta_i(S_{Sj}, S_{ij})$ is the expected number of backorders of item $i$ at base $j$ at a random point in time when depot stock level for item $i$ is $S_{Sj}$ and base stock level is $S_{ij}$, and $\lambda_i$ is the failure rate of item $i$ at base $j$. The total expected delay per demand at base $j$ due to shortages in all items is then

$$\Delta_j = \frac{1}{r_j \lambda_j} \sum_{i=1}^{l} \lambda_i \Delta_{ij},$$  \hspace{1cm} [2.11]

where $\lambda_j$ is the failure rate of the end items at base $j$, $r_j$ is the likelihood that an end item can be repaired at base $j$, and $l$ is the number of items.

Muckstadt discusses the optimization problem, which is considerably more complex than with METRIC due to the incorporation of the multi-indenture aspect of the problem. This model has been implemented by the Air Force as the method for computing recoverable spare stock levels for the F-15 weapon system.

The models in [67] and [81], like many others, assume that repair times are independent, that is, there is no waiting or batching before failed items begin the repair process. Scudder and Hausman [80] have simulated a hypothetical repair shop with limited or shared capacity which forces the repair times to be dependent. The end item analyzed is similar to the one in [67] (i.e., a hierarchical (indentured) product structure) with an additional level. An example would be a jet
engine which is composed of several subassemblies called modules, which in turn are composed of a number of parts called components. An engine failure then could be caused by the malfunction of one or more modules, which are in turn caused by one or more component failures. Prioritized scheduling is used for repairs which gives the highest priority to the component with the lowest value of net inventory. The performance of the system then, is a function of three major factors: 1) the target spares inventory level for the end items, modules and components, 2) the capacity to repair components and to perform inspection, assembly, and testing on modules and engines, and 3) the priority scheduling system used in the repair shop.

A variety of heuristic approaches to the spares stocking decision were explored, and they were also compared with use of a model requiring independent repair times (even though that assumption is not valid here). The results show that even when repair time dependencies are present, the performance of a model which assumes independent repair times is quite good.

Schaefer [76] treated the problem of determining optimal inventories for a maintenance center, servicing a fleet of \( M \) identical and highly reliable machines which experience equal workloads. Each machine is assumed to contain a single part of type \( i (i = 1,2,\ldots,n) \), and essential functions of the machine can be carried out by many different subsets of operating parts. Each machine is sent to the maintenance center every \( M \) days and thoroughly inspected. It is possible that none of the parts have failed. All failed parts for which replacements are in stock are removed and sent immediately to the repair shop, from which part \( i \) returns according to a repair time distribution with an item-dependent mean. If a replacement is not available, the failed part must either be exchanged for a good part from some exogeneous source, or, repaired immediately under some emergency procedure which does not affect the routine repair time distributions to keep repair times independent. Assuming the failure rates of the items is Poisson, the objective of the model is to maximize job-completion rate subject to a holding cost constraint. The problem is reduced to a nonlinear knapsack problem which can be solved using a dynamic programming algorithm.

2. Literature Review
Graves [38] considered a two-echelon inventory system for a repairable item where the system consists of a repair depot and \( N \) operating sites. Each site requires a set of working items and maintains an inventory of spare items. All failed items are repaired at the repair depot, which also maintains an inventory of spare items. When an item fails at a site, three events occur simultaneously. First, the failed item is replaced with a spare item from the site's inventory, if one is available; otherwise, there is a shortage at the site that will last until a replacement can arrive from the repair depot. Second, the failed item is sent to the depot for repair. Third, the depot ships a replacement item if it has available inventory; otherwise, the depot backorders the replacement request and will fill it when stock is available. When the failed item arrives at the repair depot, it enters the repair process; upon completion of the repair process, the item goes into the depot inventory or fills a backorder if any exit. Graves assumed that at each site the failure process is a compound Poisson process that may depend upon the required number of working items, but does not depend on the actual number of working items. Another assumption made is that the shipment time from the repair depot to each site is deterministic. For convenience, the depot-to-site shipment time is considered the same for all sites and given by \( T_1 \).

Defining \( Q_i(t) \) to be the outstanding orders at site \( i \) at time \( t \), where an unfilled request could be either in transit from the depot to the site or backordered at the depot, and \( s_i \) is the number of spare items stocked at the site, then \( s_i - Q_i(t) \) is the net spare inventory on hand at time \( t \), where a negative value denotes a shortage level. Note that because the failure process is a Poisson process that depends on the required number of working items, \( Q_i(t) \) depends only on the number of spares stocked at the depot, denoted by \( s_0 \), and not on \( s_i \). In fact, it is shown by Graves that the aggregate outstanding orders at the sites, \( Q(t + T_1) \), is given as

\[
Q(t + T_1) \equiv \sum_{i=1}^{N} Q_i(t + T_1) = B(t|s_0) + D(t, t + T_1),
\]

where \( B(t|s_0) \) is the backorders at time \( t \) at the depot assuming the depot stocks \( s_0 \) items, and \( D(t, t + T_1) \) is the aggregate failures at all sites over the time interval \( (t, t + T_1) \). Graves gave an in-
tuitive explanation to the equation above which goes as follows. “At time \( t \) the outstanding orders at the sites either are in-transit to the sites or are backordered at the depot. All items that were in-transit at time \( t \) will arrive at the sites by time \( t + T_1 \). However, none of the depot backorders at time \( t \) can arrive at the sites by time \( t + T_1 \), since the shipping time from depot to site is \( T_1 \). For the same reason, any failure in the time interval \((t, t + T_1]\) generates an outstanding order that cannot be filled by time \( t + T_1 \). Hence, at time \( t + T_1 \), the depot backorders from time \( t \), \( B(t|s_0) \), and the aggregate failures, \( D(t,t + T_1) \), must be outstanding.” He further states, that since \( B(t|s_0) \) and \( D(t,t + T_1) \) are independent, they do not double-count any orders.

Since Graves’ objective is the characterization of system performance for a given specification of the inventory stockage levels, it was necessary to determine the distribution of \( Q(t) \) from which the best site stockage level \( s_i \) that optimizes some criterion can be found, given \( s_0 \). But first, two issues had to be resolved: 1) the convolution of \( B(t|s_0) \) and \( D(t,t + T_1) \) to obtain \( Q(t) \), and 2) the disaggregation of \( Q(t) \) into \( Q_i(t) \) for \( i = 1,2,...,N \). The distribution for \( D(t,t + T_1) \) is assumed to be known, and the distribution for depot backorders is

\[
B(t|s_0) = [Q_0(t) - s_0]^+,
\]

where \( Q_0(t) \) is the number of failed items in the system (in-transit to the depot and in the repair process at the depot) at time \( t \), and \([x]^+\) denotes the nonnegative part of \( x \). By assuming ample repair capacity, \( Q_0(t) \) is equivalent to the occupancy level in an \( M/G/\infty \) queue. Therefore, due to Plam’s theorem in [70], the steady-state distribution of \( Q_0(t) \) is Poisson, and hence, the steady-state distribution of \( B(t|s_0) \) can be obtained using the relationship above.

Assuming that backorders are filled on a first-come, first-serve basis, Graves argues, that this implies that the disaggregation of \( Q(t) \) is equivalent to a random disaggregation across the sites. That is the likelihood that any outstanding order is from a particular site \( i \) is directly proportional to the site’s failure rate \( \lambda_i \). Using the fact that the conditional distribution of \( Q_i(t) \) on \( Q(t) \) is a binomial distribution, it was determined that

2. Literature Review
\[
Pr(Q(t) = j) = \sum_{k=j}^{\infty} Pr(Q(t) = k) Pr(Q(t) = j | Q(t) = k) = \sum_{k=j}^{\infty} \binom{k}{j} \rho^j (1 - \rho)^{k-j},
\]

[2.14]

where \( \lambda = \sum_{i=1}^{N} \lambda_i \) is the aggregate failure rate, and \( \rho = \frac{\lambda_i}{\lambda} \).

2.2.3 Models with Batch Ordering Policies

As the majority of repairable items are of high value, most of the studies in the literature of multi-echelon repairable systems have employed an order-for-order \((S - i, S)\) inventory policy. The policy is a reasonable one when the item is expensive and the holding cost is high, and when demand rate for the item (failure rate) is low. Nevertheless, there could exist cases, in general, where the batch ordering policies are more cost-effective than the order-for-order policies. The study of multi-echelon repairable models with batch ordering policies is motivated by the following factors: 1) when failures of the item and hence demands are more frequent and/or the set up cost of placing an order is relatively high, it may be more cost-effective to employ the batch ordering policy. Note that the high demand rate of the item may not be a result of the item being less reliable, but could be due to the fact that there is more of the product in operation in the market; and 2) the batch model can be viewed as a generalization of the previous studies on \((S - 1, S)\) policies, which becomes a special case of the batch ordering models.

The drawback for not using the \((S - 1, S)\) policy, however, is the possibility of losing some of the attractive properties, such as those related to the birth-death process, which greatly simplify the analysis and reduce computations. This could explain why the literature is scarce on multi-echelon inventory systems that do not employ the \((S - 1, S)\) policy.
Lee and Moinzadeh [59] considered a system with one depot and \( N \) bases. Failed items are brought to one of the bases. If the base has stock on hand it will replace the failed items; otherwise a backorder is incurred. The stocking level at each base is \( s_i \), and when there are \( Q \) failed items in a base, they are sent to the depot for repair and an order is placed by the base to the depot simultaneously to replenish the stock. In conventional continuous review inventory systems, each base operates in a similar fashion as a \((s_i, Q)\) policy. Upon receiving orders, if the depot has sufficient inventory on hand, it will send the items down to the bases in batches of \( Q \), otherwise a delay will be incurred.

The authors' approach to the problem follows closely that of Graves [38], with the objective of studying the underlying operating characteristics of the inventory system under batch ordering policy. They also consider the more general case of finite servers, as opposed to infinite servers, at the repair center. The assumptions made here are basically the same as those made in [38], with the additional assumption that the depot will always fill orders from the bases in complete batches, i.e., no partial fill is allowed. With the assumption that the demand process at each base is Poisson with mean rate \( \lambda_i \) \((i = 1, 2, ..., N)\), the inter-arrival time between orders from each base to the depot follows an Erlang distribution of order \( Q \). The process describing the arrivals of orders at the depot is the superposition of \( N \) arrival processes each having an inter-arrival time that is Erlang distributed. Lee and Moinzadeh state, that when the number of bases is large, the superposition process can be approximated well by a Poisson process with mean rate \( \lambda_0 = \sum_{i=1}^{N} \lambda_i/Q \). It is this approximation that is used throughout their research. (See Çinlar [21] and Feller [33] for further discussion on the superposition process.)

Using Grave's notation, except for the number of orders outstanding at base \( i \) at time \( t \), denoted by \( w_i(t) \). Lee and and Moinzadeh use the familiar expression for the aggregate number of orders outstanding which is

\[
w(t + T_i) = B(t|s_0) + D(t, t + T_i).
\]

[2.15]
Here again, the authors are faced with the two issues of convolving the two distributions to get \( w(t + T_i) \), and then disaggregating \( w(t + T_i) \) to obtain \( w_i(t) \) for \( i = 1,2,\ldots,N \). For the case of \( Q = 1 \), based on [38], the disaggregation of \( w \) is equivalent to a random disaggregation across the bases. For \( Q > 1 \), the authors still use the random disaggregation even though it does no apply here. Nevertheless, the method yields an excellent approximation when compared to simulation tests.

The steady state distribution of \( w_i \) then is approximated as follows:

\[
Pr(w_i = j) = \sum_{k=j}^{\infty} Pr(w = k) Pr(w_i = j|w = k) \\
= \sum_{k=j}^{\infty} Pr(w = k) \binom{k}{j} \rho^j (1 - \rho)^{k-j},
\]

where \( \rho = (\lambda_i/\lambda_d Q) \).

The number of units backordered at base \( i \), \( B_i(t) \), can be obtained by

\[
B_i(t) = \left[ Qw_i(t) + Y_i(t) - s_i \right]^+, \tag{2.17}
\]

where \( Y_i(t) \) is the number of failed units waiting at base \( i \) to be sent to the depot repair. For Poisson arrivals at the bases, Hadley and Whitin [46] have shown that \( Y_i(t) \) is uniformly distributed between 0 and \( Q - 1 \). After performing many simulation runs, the authors claim that the model gave accurate results for cases with \( Q \leq 7 \).

In [60], Lee and Moinzadeh have presented a model which is an extension of the one given in [59]. In addition to all the assumptions made in the previous model, the authors also assumed that the shipping time from the base to the depot is negligible; and for each failed item arriving at a base, there is a probability \( p \) that it can be repairable at the depot and a probability of \( 1 - p \) it would be condemned. The decision to repair or condemn an item is made at the depot, i.e., all failed items have to be sent to the depot before the repair/condemnation decisions can be made. The time in making such a decision is negligible.

2. Literature Review
The procurement policy at the depot for the condemned items is assumed to be that of a continuous review \((s_0 - Q_0, nQ_0)\) procurement policy, where \(nQ_0\) is the batch size for the depot's order to the outside supplier, and \(s_0\) is the maximum inventory position at the depot. Whenever the inventory position drops below \(s_0 - Q_0\), an amount \(nQ_0\) is ordered, where \(n\) denotes the positive integer that will bring the inventory position at the depot back to the set \(\{s_0 - Q_0 + 1, s_0 - Q_0 + 2, \ldots, s_0\}\). The inventory position is computed as on-hand inventory + on-order inventory + items in repair - number backordered.

In studying the operating characteristics of the model, Lee and Moinzadeh have approached the problem the same way they have approached the model in [59]. They have also considered the cases where the repair times are deterministic and general. Again comparing their approximate model with many simulation runs, it was shown that their model was very accurate in about 90% of the cases.

In a recent paper, Cohen et al. [22] studied a problem which is a generalization of the well-known tool-kit problem. The analysis considers a facility that stocks various parts in support of repairs for a set of products in some customer region. The aim is to determine base stock policies for each part to minimize expected inventory costs across all parts while satisfying some service constraint on total customer repair services completed. Some of the important assumptions which characterize the model developed are the following: 1) the repair network is single-echelon, i.e., the parts that are not stocked at the facility are obtained from independent suppliers; 2) the stocking policy used by the facility is a periodic review base stock or order-up-to policy. At the beginning of each period, each part \(i\) is stocked to the level \(S_i\) by ordering the usage recorded from the previous period, and 3) the lead times to restock failed parts are zero.

Denote the parts that make up a product by the subscripts \(i \in N = \{1, 2, \ldots, n\}\), and let \(\{D_i | i \in N\}\) be the random variables that represent the failure of parts in any given period with marginal pmfs and cdfs denoted, respectively, \(f_i\) and \(F_i\), \(i \in N\); and with joint pmf and cdf denoted \(f\) and
Mathematically, the problem can be stated as follows. The objective is to minimize expected costs per period, i.e.,

$$\text{minimize } G(\mathcal{S}) = \sum_{i \in N} G_i(\mathcal{S}_i)$$  \hspace{1cm} [2.18]

where $\mathcal{S} = (\mathcal{S}_1, ..., \mathcal{S}_n)$ and $G_i(\mathcal{S}_i)$ are the expected ordering, holding, transport, and shortage costs per period associated with part $i \in N$. Eq. [2.18] is to be minimized subject to service constraints, and the authors consider two forms of service level specification.

The first specification is of the form:

$$\Pr\{\sum_{i \in N} (D_i - S_i)^+ > 0 | \sum_{i \in N} D_i > 0\} \leq \beta,$$  \hspace{1cm} [2.19]

which says that in the long run, excess demand (for any part in the product of interest) should be greater than zero for at most a fraction $\beta$ of those periods for which demand is nonzero. Knowing that the event $\{\sum_{i \in N} (D_i - S_i)^+ > 0\}$ is equivalent to the event $\{\bigcup_{i \in N} [D_i > S_i]\}$ and $\Pr(\sum_{i \in N} D_i > 0) = 1 - F(\emptyset)$, where $\emptyset$ is the $n$-dimensional null vector, Eq. [2.19] can be expressed as

$$1 - F(\mathcal{S}) \leq \beta[1 - F(\emptyset)].$$  \hspace{1cm} [2.20]

The second specification takes the form

$$E\left\{\frac{\sum_{i \in N} \min[D_i, S_i]}{\sum_{i \in N} D_i} \left| \sum_{i \in N} D_i > 0\right\} \geq PAL,$$  \hspace{1cm} [2.21]

The second specification takes the form

2. Literature Review
where PAL is the required part availability level for parts in the product as a whole. Because [2.21] is not tractable, the authors have used the following approximation:

\[
\frac{\sum_{i \in N} E\left(\min[D_i, S_i]\right)}{\sum_{i \in N} E[D_i]} \geq PAL.
\]  

[2.22]

Restricting their attention to the case where part failures are independent, Cohen et al. [22] developed two near-optimal procedures to minimize [2.18] subject to [2.20] or [2.22]: the dual method which is based on the Lagrange multipliers technique, and the greedy heuristic which is based on incremental benefit/cost analysis. Conditions were given for the equivalence of the two methods and comments were made on their relative computational merits. Extensions to deal with two priority classes and with parts commonality are also discussed.

### 2.3 Summary and Conclusion

The survey presented in this chapter, is by no means an exhaustive review of the research done on repairable item problems; but that is not the purpose here either. Besides giving a general idea of what has already been investigated, the purpose of the survey is to give a clear idea of what the spares provisioning or repairable item inventory problem entails, what are the difficulties that one could encounter, and what is the best feasible approach to the problem. Often, deriving the exact solution method would be too involved mathematically, and so researchers tend to simplify the model by adding intelligent assumptions, which would justify using approximate solution methods.
that are practical and accurate. The survey discusses, in detail, some of these assumptions, their validity, and a number of the approximate methods used, in order to facilitate future research.

Some of the important conclusions derived from the survey are the following. The demand or failure process is assumed to be Poisson, and any other process is analyzed almost exclusively through simulation. The vast majority of the models, have been studied only at steady state; the literature on transient state analysis is very rare. Some of the research on transient analysis that could be cited are those of Brammer [10], Frisch [34], Gross et al. [44], and Balana et al. [5]. No study has been successful in deriving an exact solution method for the multi-item case, and as a result few papers deal with hierarchical equipment systems such as those discussed in [65].
3. The Spares Provisioning Problem

In this chapter, we consider a spares provisioning system which consists of two subsystems: the production floor, and the repair facility. A total of $N$ machines are owned by the system, and a maximum of $D$ machines may be deployed simultaneously towards meeting demand. A machine that is in use operates for $\bar{t}$ time units before failure. Once a machine fails it instantaneously enters the repair facility, in which $M$ identical and parallel repair stations are retained to service the machines. If an arriving machine finds all repair stations busy, the machine joins the repair queue and waits its turn for service. The time required to repair a machine is $\bar{x}$.

Once a machine is repaired, it is either immediately deployed to meet demand or it joins the queue of stand-by machines if the number of operational machines exceeds $D$. The spare machines have no probability of failure and, if available, a spare replaces a deployed machine immediately upon the latter's failure. Figure 2 shows a block diagram of a typical spares provisioning system.
It is assumed that $\bar{r}$ and $\bar{x}$ are both drawn from exponential distributions with respective rates $\lambda$ and $\mu$. We denote $\bar{n}_d$ as the number of machines operating, $\bar{n}_r$ as the number of machines being repaired, $\bar{n}_q$ as the number of machines waiting in the repair queue, and $\bar{n}_s$ as the number of spare machines. The machines in the various parts of the system sum to the total number of machines owned by the system as follows:

$$N = \bar{n}_d + \bar{n}_r + \bar{n}_q + \bar{n}_s.$$

Given the inherent probabilistic behavior of the system, the objective is to determine the $N$ and $M$, given $D$, which minimizes cost while maintaining an average of at least $n^*_r$ deployed machines; $n^*_r \leq D$, where $D$ is only met under very restricted conditions. The objective function of such a

Figure 2. Block Diagram for the Spares Provisioning Problem.
system typically reflects total system costs, including the costs of the machines and the service
channels. Shortage costs are incurred whenever demand is not met, and are also included in the
objective function; however, they can be eliminated if an availability or a service level constraint is
imposed on the system, which is what we have chosen to do here. The total expected system cost
$TC(N,M)$, is then just a function of the number of machines and repair stations, and the optim-
mization problem can be stated mathematically as follows:

$$\text{Min: } TC(N,M) \quad s.t. \quad \sum_{i=1}^{D} ip_i \geq n_g^*,$$

where $p_i$ is defined as the marginal steady-state probability of $i$ machines operating. A formula for
the steady-state probability distribution of the spares provisioning problem is given in Appendix
A, which can be used to determine $p_i$.

Gross et al. [45] have examined some of the mathematical properties of a system similar to the
one described above. With a linear cost function and a linear constraint, they showed that the av-
erage cost is a monotonically nondecreasing function of the number of machines and repair stations.
We show in Appendix B that the cost function for our system has the same property. Hence, the
minimum cost is achieved on the boundary of the feasible solution space. Keeping this in mind,
our approach then is to simply generate the set of boundary points efficiently, and then select the
one that minimizes $TC(N,M)$. As will be shown later, the set of boundary points is usually very
small, and the selection of the minimizing point can be easily done through enumeration.

3. The Spares Provisioning Problem
3.1 Solution Overview

Our approach to specifying the set of boundary points for the solution space depends heavily upon Little's result [62]. In broad terms, Little's result states that, under very general condition, the expected number of entities in a system, \( L \), is equal to the product of the average arrival rate of entities to the system, \( \lambda \), and the expected amount of time spent by the entities in the system, \( W \); that is, \( L = \lambda W \). The key idea involved in our approach is to apply Little's result to both subsystems of the overall system considered in isolation in order to generate lower bounds for \( N \) and \( M \). These lower bounds are then used as input to an iterative procedure which yields the boundary points of the feasible solution space.

Recall that our objective is to maintain an average of at least \( n^*_g \) deployed machines. Therefore, as a starting point, we consider the set of deployed machines as a system. Since the working time of a deployed machine is drawn from an exponential distribution with parameter \( \lambda \), the average amount of time a machine spends in the deployed machine subsystem is \( \frac{1}{\lambda} \). Now, define \( \lambda_f \) to be the aggregate rate at which working machines fail. Then, we have from Little's result

\[
E[\tilde{n}_d] = \frac{\lambda_f}{\lambda} \quad [3.1]
\]

and, since our goal is to have \( E[\tilde{n}_d] \geq n^*_g \), a necessary condition for optimality is

\[
n^*_g \leq \frac{\lambda_f}{\lambda} \quad [3.2]
\]

or equivalently,

\[
\lambda_f \geq \lambda n^*_g
\]
Since we have a finite number of machines in a closed system, \( \lambda_f \) is the average arrival rate to every subsystem in the entire system. Therefore, from Little's result, we find that the expected number of machines in repair subsystem is given by

\[
E[\tilde{n}_s] = \lambda_f E[\tilde{x}] = \frac{\lambda_f}{\mu} \geq \frac{\lambda}{\mu} n^*_s,
\]

[3.3]

where we have used the fact that \( E[\tilde{x}] = 1/\mu \) in the last equation.

Since each machine in repair is being repaired in a repair station, there must be at least as many repair stations as there are machines being repaired. Thus, from [3.3], we see immediately that

\[
M \geq \frac{\lambda}{\mu} n^*_s.
\]

[3.4]

In addition, since \( E[\tilde{n}_s] \geq n_s^* \), and the total number of machines in the system must be at least as large as the sum of the average numbers of machines in each of the subsystems, we find that

\[
N \geq n^*_s + \frac{\lambda}{\mu} n^*_s.
\]

[3.5]

From [3.4] and [3.5] then, we may obtain lower bounds for the set of feasible solutions. Specifically, we take the point \((M_0, N_0(M_0))\), where

\[
M_0 = \left[ \frac{\lambda}{\mu} n^*_s \right]^+,
\]

[3.6]

and

\[
N_0(M_0) = \left[ n^*_s + \frac{\lambda}{\mu} n^*_s \right]^+.
\]

[3.7]

as a starting point in our iterative procedure which yields the boundary points.

We now turn our attention to the description of our iterative procedure for obtaining the boundary points of the feasible solution space. Before describing the procedure, we point out that
the set of feasible boundary values for \( M^* \) is \( \hat{M} = \{ M_0, M_0 + 1, \ldots, N(N_0(M_0)) \} \). The reasoning for the upper limit is as follows. First, note that \( N_0(M_0) \) machines will meet the demand if the machines never have to wait for repair, and if there are no spares. Now, if there are \( N_0(M_0) \) repair stations and \( N_0(M_0) \) machines, although machines will never wait for repair, demand may not be met because the average number of spare machines will exceed zero. In fact, demand will be met only if the rounding up process provides enough machines to compensate for the average number of machines in the spares queue. Thus, \( N(N_0(M_0)) \geq N_0(M_0) \) machines will be required to meet demand. If equality is not satisfied, then it may be possible to satisfy requirements with fewer machines by increasing the number of repair stations. However, since \( N(M) \) is nonincreasing in \( M \), any increase in \( M \) beyond \( N(N_0(M_0)) \) will result in more than one repair station per machine, a situation which is obviously not beneficial. The boundary region is therefore the set of points \( \{(M_0, N(M_0)), (M_0 + 1, N(M_0 + 1)), \ldots, (N(N_0(M_0)), N(N_0(M_0)))\} \). That is, for each feasible boundary value, \( m \in \hat{M} \), there is a corresponding value, \( N(m) \), that is the minimum value of \( N \) which satisfies the production constraint when there are \( m \) repair channels. The idea is to generate a sequence of values \( \{N_0(m), N_1(m), \ldots\} \) that converges to \( N(m) \) for each \( m \in \hat{M} \).

We note from [3.2] and [3.3] that

\[
\lambda_f = \mu E[\tilde{n}_s] \geq \lambda n^*_s. \tag{3.8}
\]

If we denote the aggregate machine failure rate required to maintain \( n^*_s \) deployed machines by \( \lambda^*_s \), then \( \lambda^*_s = \lambda n^*_s \). Our approach is, given \( \{(m, N_0(m))\} \), to generate an increasing sequence \( \{N_0(m), N_1(m), \ldots\} \) that produces an increasing sequence of aggregate failure rates \( \{\lambda_0, \lambda_1, \ldots, \lambda_f\} \), stopping at the minimum value of \( N \) that yields an aggregate failure rate of at least \( \lambda^*_s \). This failure rate would then guarantee that the average number of deployed machines exceeds the average demand \( n^*_s \). That is, starting with \( m \) repair stations, we find the minimum \( N, N(m) \), which yields \( \lambda_f \geq \lambda^*_s \).

3. The Spares Provisioning Problem
A more formal statement of the iterative procedure is now provided. First, obtain the initial solution $M_0$ and $N_0(M_0)$ from [3.6] and [3.7], and then set $m = M_0$ and $i = 0$. For $m = M_0$ to $N_0(M_0)$, do the following:

Step 1: Based on $(m, N_i(m))$ compute $\lambda_i$, the aggregate failure rate for a system with $m$ repair stations and $N_i(m)$ machines.

Step 2: If $\lambda_i \geq \lambda^*$, then set $N(m) = N_i(m)$ and stop; the feasible boundary point $(m, N(m))$ has been reached. Otherwise, go to Step 3.

Step 3: Increase the total number of machines to

$$N_{i+1}(m) = \left[ \frac{\lambda^*}{\lambda_i} N_i(m) \right]^+. \tag{3.9}$$

Step 4: Set $i = i + 1$, and go to Step 1.

We argue in the next section, that the above iterative procedure always yields $N(m)$.

The procedure just described requires the computation of the failure rate $\lambda_i$, given $(m, N_i(m))$. There are many ways to compute $\lambda_i$ for our system; these methods include mean value analysis and computing the equilibrium state probabilities of the system and then finding the mean values. Numerous algorithms have been presented to solve such a problem; however, we chose to develop an alternate technique based on some elementary results from renewal theory. The new technique, which is presented in Section 3.3, computes the expected number of busy repair stations, $E[\bar{n}_i]$, given $(m, N_i(m))$, from the analysis of the busy periods. Knowing $E[\bar{n}_i]$, we can then easily obtain $\lambda_i$ from Little’s result as shown in [3.3].
3.2 Convergence of Iterative Procedure

In this section, we argue that our iterative procedure generates the minimum number of machines $N(m)$ for a given number of repair stations $m$. As described earlier, a typical machine in the system cycles through the following set of phases: operating, waiting in the repair queue, being repaired, and finally, acting as a spare until it is needed again for deployment. Passage through each of these phases one time constitutes a cycle, the length of which we denote by $\tilde{C}$. Let $\tilde{w}$ denote the total time that a typical machine spends both in the repair queue and acting as a spare. Then the average cycle length is given by

$$E[\tilde{C}] = E[\tilde{t}] + E[\tilde{x}] + E[\tilde{w}] = \frac{1}{\lambda} + \frac{1}{\mu} + E[\tilde{w}].$$ \hspace{1cm} [3.10]

For fixed $M = m$, the proportion of time that a typical machine is operating is a function of $N$ and is equal to the ratio of the expected operating time to the expected cycle length. That is,

$$\pi_N = \frac{E[\tilde{t}]}{E[\tilde{t}] + E[\tilde{x}] + E[\tilde{w}]},$$ \hspace{1cm} [3.11]

where $\pi_N$ denotes the proportion of time a typical machine spends operating in a system having $m$ repair stations and $N$ machines. Once $\pi_N$ is known, we can easily compute the aggregate failure rate. Since each machine generates failures at a rate $\lambda$ whenever it is operating, the aggregate failure rate is readily found as

$$\lambda_f = N\lambda \pi_N.$$ \hspace{1cm} [3.12]

Since $E[\tilde{t}]$ and $E[\tilde{x}]$ are both known constants which are independent of $N$, $E[\tilde{w}]$ is the only term in the expression for $\pi_N$ which is a function of $N$. Clearly, $E[\tilde{w}]$ is a nondecreasing function.
of $N$. In particular, for $N \leq \min(D,M)$, a machine in the system will never have to wait for an idle repair station since $M \geq N$; and of course, will never act as a spare since $D \geq N$, and as a result $E[\tilde{w}] = 0$. Therefore, for $N \leq \min(D,M)$, $\pi_N$ is a constant, and the aggregate failure rate increases linearly with $N$. Specifically,

$$
\lambda_f = N\lambda \left( \frac{1}{1/\lambda + 1/\mu} \right). \tag{3.13}
$$

For $N > \min(D,M)$, there is a positive probability that a machine will wait in one of the two queues in the system, and thus, $E[\tilde{w}] > 0$. In fact, since $\tilde{w}$ arises from mutual interference among the machines, we know intuitively that increasing the total number of machines in the system beyond $\min(D,M)$, will result in a corresponding increase in $E[\tilde{w}]$. Therefore, from [3.10] and [3.11], we see that the expected cycle length is an increasing function of $N$, and thus, $\pi_N$ is a decreasing function of $N$.

To see more clearly how $E[\tilde{w}]$ affects the aggregate failure rate in the system as the number of machines increases, we examine the first derivative of $\lambda_f$ with respect to $N$. The derivative, which expresses the rate of change of $\lambda_f$ as $N$ increases, is obtained as follows:

$$
\frac{d}{dN} \lambda_f = \frac{d}{dN} (N\lambda \pi_N) = \lambda \pi_N + N\lambda \frac{d}{dN} \frac{d\pi_N}{dN}
$$

$$
= \frac{1}{1/\lambda + 1/\mu + E[\tilde{w}]} - N \frac{(d/dN)E[\tilde{w}]}{(1/\lambda + 1/\mu + E[\tilde{w}])^2} \tag{3.14}
$$

where $(d/dN)E[\tilde{w}]$ is nonnegative, since $E[\tilde{w}]$ is a nondecreasing function of $N$.

The first term of the right hand side of [3.14] is the slope of the straight line connecting the origin and $(N, \lambda_f(N))$. Since the second term on the right hand side of [3.14] is never positive, then for any value of $N$, the slope of $\lambda_f(N)$ cannot exceed the slope of the line connecting $(0,0)$ and $(N, \lambda_f(N))$. This means that $\lambda_f(N)$ has a nonincreasing derivative and is, therefore, concave. Figure

3. The Spares Provisioning Problem
3 illustrates the shape of the curve which is required to have these properties. From the discussion above, we can conclude that the slope of the function \( \lambda_f(N) \) is strictly less than the slope \( \frac{\lambda_f(N)}{N} \), except in the region where \( N \leq \min(D,M) \), where \( \frac{d}{dN} \lambda_f(N) = \frac{\lambda_f(N)}{N} \).

As a final note on Figure 3, we see that as \( N \) increases, \( \lambda_f \) approaches a limit that is set by \( \min(\lambda D, \mu M) \). In other words, the subsystem with the lowest output rate acts as a bottleneck and sets the maximum traffic rate for the system. It is then clear that if \( \lambda^* = \lambda n^* > \min(\lambda D, \mu M) \), the service-level constraint cannot be met.

Figure 4 will be used as an aid to understanding why our iterative procedure converges to \( \lambda^*_f \). We first obtain the initial number of machines \( N_0(m) \), given a certain value of the number of repair stations, \( m \). The point \((m, N_0(m))\) produces an aggregate failure rate \( \lambda_f \). If we extend the line which passes through \((0,0)\) and \((N_0, \lambda_f)\), the value of \( N \) at which the line crosses \( \lambda = \lambda^*_f \) is given by \( \frac{\lambda^*_f}{\lambda_f} N_0 \). Due to our earlier argument, the failure rate resulting from \( \left[ \frac{\lambda^*_f}{\lambda_f} N_0 \right] \) will be below \( \lambda^*_f \). It may turn out that the failure rate resulting from \( \left[ \frac{\lambda^*_f}{\lambda_f} N_0 \right] \) will be above \( \lambda^*_f \). If so, then we have reached the feasible boundary point \((m, N(m))\); otherwise, the procedure needs to be repeated. Note that so long as \( \lambda^*_f \) has not been reached \( \frac{\lambda^*_f}{\lambda_f} > 1 \) so that the sequence of \( N \)'s is strictly increasing. The resulting progression is illustrated in Figure 4.

### 3.3 Computation of Aggregate Failure Rate

The iterative procedure presented earlier in this chapter depends on the computation of the average failure rate, \( \lambda_f \), given an operating point \((m, N(m))\). Although many existing algorithms can

3. The Spares Provisioning Problem
Figure 3. The Aggregate Failure Rate as a Function of the Number of Machines in the System.

Figure 4. The Aggregate Failure Rate as a Function of the Number of Machines in the System.

3. The Spares Provisioning Problem
be used to obtain \( \lambda_f \) for a system such as ours, most are inefficient in terms of computer storage and CPU time, especially for large machine repair systems. An alternate technique, based upon first passage time analysis, is presented in this section. This alternate technique requires very little computer storage and negligible CPU time.

To begin our development, we note that from Little's result,

\[
\lambda_f = \mu E[\tilde{n}_f].
\]

The expectation of the previous equation can be obtained using the well known result that the expected value of a nonnegative random variable is equal to the integral of the complementary distribution. Since our random variable is integer valued, we find

\[
E[\tilde{n}_f] = \sum_{i=1}^{M} P\{\text{at least } i \text{ repair stations busy}\}. \tag{3.15}
\]

We shall compute the required probabilities by using first passage time analysis.

Define \( \tilde{y}_N \) to be the random variable which represents the length of an ordinary busy period when \( N \) machines are in the system. That is, if all repair stations are idle, \( \tilde{y}_N \) is the length of time starting with a machine failure, and ending when all repair stations are idle again. Since the system alternates between a busy period and an idle period, the times to successive periods of either type constitutes a renewal interval, and the process which counts the number of renewals is an alternating renewal process (Ross [74]). Thus, from classical results on alternating renewal processes, we find that

\[
P\{\text{all repair stations are idle}\} = \frac{E[\tilde{i}]}{E[\tilde{y}_N] + E[\tilde{i}]}, \tag{3.16}
\]

where \( E[\tilde{i}] \) denote the expected length of an idle period, and
\[ P(\text{at least 1 repair station busy}) = \frac{E[\tilde{Y}_N]}{E[\tilde{Y}_N] + E[\tilde{I}]} . \]  

[3.17]

Now, define \( \tilde{s}_i \) as the aggregate amount of time in a busy period during which there are at least \( i \) machines demanding repair; for example, \( \tilde{s}_i \) is the length of the busy period itself. Then, analogous to [3.17],

\[ P(\text{at least } i \text{ machines demanding repair}) = \frac{E[\tilde{s}_i]}{E[\tilde{Y}_N] + E[\tilde{I}]} \quad \text{for } 1 \leq i \leq N. \]  

[3.18]

Since

\[ P(\text{at least } i \text{ repair stations busy}) = P(\text{at least } i \text{ machines demanding repair}) \quad \text{for } 1 \leq i \leq M, \]

we see that

\[ P(\text{at least } i \text{ repair stations busy}) = \frac{E[\tilde{s}_i]}{E[\tilde{Y}_N] + E[\tilde{I}]} \quad \text{for } 1 \leq i \leq \min\{M, N\}. \]  

[3.19]

Thus, in order to compute \( E[\tilde{n}_i] \), we need simply to determine \( E[\tilde{s}_i] \) for \( 1 \leq i \leq \min\{M, N\} \).

Define the state of the system by the total number of failed machines, and define \( \tilde{F}_{i-1}^{\mu} \) to be the random variable representing the first passage time from state \( i \) to state \( i - 1 \) for a system which owns a total of \( N \) machines. In other words, \( \tilde{F}_{i-1}^{\mu} \) is the length of a period which starts with \( i \) failed machines and \( N - i \) operational machines, and ends at the instant the number of failed machines is equal to \( i - 1 \). For example, the length of an ordinary busy period is simply \( \tilde{F}_{10}^{\mu} \). Now, define \( \tilde{v}_{ij} \) as the number of visits to state \( j \) from state \( i \) during the first passage time from state \( i \) to \( i - 1 \).

Then, we find that

3. The Spares Provisioning Problem
\[ \tilde{v}_i = \tilde{\gamma}_i^N \quad \text{for } i = 1, \]
\[ = \sum_{n=0}^{\infty} f_{i-1,n}^{\tilde{\gamma}_i} \quad \text{for } 2 \leq i \leq N, \]  

[3.20]

where \( f_{i-1,n}^{\tilde{\gamma}_i} \) denotes the first passage time from state \( i \) to \( i-1 \) on the \( n \)-th visit to state \( i \) during a busy period. Then, since \( f_{i-1,n}^{\tilde{\gamma}_i} \) is independent of the number of visits to state \( i \), we find from Wald’s equation that

\[ E[\tilde{v}_i] = E[\tilde{v}_{i-1}] E[\tilde{\gamma}_i]. \]  

[3.21]

It remains to specify computational formulae for \( E[\tilde{v}_{i,n}] \) and \( E[\tilde{\gamma}_i] \) for \( 1 \leq i \leq N \), to which we now turn our attention.

Define \( \tilde{\tau}_i \) to be the time to first failure during a period in which there are \( i \) operational machines (including the stand-by machines), \( \tilde{\tau}_j \) as the time to first repair during a period in which there are \( j \) machines demanding repair, and their corresponding failure and repair rates by \( \lambda_i \) and \( \mu_j \), respectively. Then,

\[ \lambda_i = \begin{cases} i\lambda, & \text{if } i = 0,1,...,D-1, \\ D\lambda, & \text{if } i = D,D+1,...,N, \end{cases} \]

and

\[ \mu_j = \begin{cases} j\mu, & \text{if } j = 0,1,...,M-1, \\ M\mu, & \text{if } j = M,M+1,...,N. \end{cases} \]

Given the properties of the exponential distribution, we can compute the expected length of an ordinary busy period, \( E[\tilde{\gamma}_n] \), by conditioning on whether or not the initial repair is completed before the next failure occurs. That is

3. The Spares Provisioning Problem
\[ E[\bar{\gamma}_N] = E[\bar{\gamma}_N | \bar{x}_1 < \bar{t}_{N-1}] P(\bar{x}_1 < \bar{t}_{N-1}) + E[\bar{\gamma}_N | \bar{x}_1 > \bar{t}_{N-1}] P(\bar{x}_1 > \bar{t}_{N-1}). \]  \[3.22\]

where \( \{\bar{x}_1 < \bar{t}_{N-1}\} \) represents the event that a repair is completed before the next failure occurs, and \( \{\bar{x}_1 > \bar{t}_{N-1}\} \) is the complementary event. Now, due to the properties of the exponential distribution,

\[ E[\bar{\gamma}_N | \bar{x}_1 < \bar{t}_{N-1}] = E[\min(\bar{x}_1, \bar{t}_{N-1}) | \bar{x}_1 < \bar{t}_{N-1}] = E[\min(\bar{x}_1, \bar{t}_{N-1})] = \frac{1}{\lambda_{N-1} + \mu_1}. \]

Also,

\[ P(\bar{x}_1 < \bar{t}_{N-1}) = \left( \frac{\mu_1}{\lambda_{N-1} + \mu_1} \right). \]

Under the condition that a second failure occurs before the first repair of the busy period is completed, the remainder of the busy period can be separated into two periods: the first passage time from state 2 to state 1, and the first passage time from state 1 to state 0. But the latter period is simply the length of an ordinary busy period. Thus, we find

\[ E[\bar{\gamma}_N] = \frac{1}{\lambda_{N-1} + \mu_1} \left( \frac{\mu_1}{\lambda_{N-1} + \mu_1} \right) + \left[ \frac{1}{\lambda_{N-1} + \mu_1} + E[\bar{\gamma}_{N-1}] + E[\bar{\gamma}_{N-1}] \right] \left( \frac{\lambda_{N-1}}{\lambda_{N-1} + \mu_1} \right), \]  \[3.23\]

where the first term in the square brackets, \( \frac{1}{\lambda_{N-1} + \mu_1} \), is just the average length of time before the first repair or failure during a busy period. Simplification of [3.23] yields

\[ E[\bar{\gamma}_N] = \frac{1}{\mu_1} + \frac{\lambda_{N-1}}{\mu_1} E[\bar{\gamma}_{N-1}], \]  \[3.24\]

The result in [3.24] is central to the remainder of our development, and further explanation is warranted. It can be easily shown that our system can be described by a birth-death stochastic process, and hence, during a busy period, if state 2 has been visited \( n \) times, then state 1 must have been visited exactly \( (n + 1) \) times before the busy period is terminated. Furthermore, the length of a busy period is the sum of the times the system spends in states 1 through \( N \). Therefore, if we
define $\tilde{T}_{1,n}$ as the length of time the system spends in state $j$ on the $n$-th visit during the busy period. Then,

$$\tilde{y}_N = \sum_{n=1}^{\tilde{v}_{1,2}+1} \tilde{T}_{1,n} + \sum_{n=0}^{\tilde{v}_{1,2}} \tilde{f}_{2,1}^N,$$  \hspace{1cm} [3.25]

where the latter sum is equivalent to the total amount of time the system spends in states $2,...,N$ during a busy period. Since $\tilde{T}_{1,n}$ and $\tilde{f}_{2,1}^N$ are each independent of $\tilde{v}_{1,2}$, by Wald's equation [3.25] reduces to

$$E[\tilde{y}_N] = E[\tilde{T}_{1,0}^N] = (E[\tilde{v}_{1,2}] + 1)E[\tilde{T}_{1,1}] + E[\tilde{v}_{1,2}]E[\tilde{f}_{2,1}^N].$$ \hspace{1cm} [3.26]

Now, the probability of a transition to state $i+1$ from state $i$ on each visit to state $i$ is independent of the number of times state $i$ has been visited. Therefore, $\tilde{v}_{i,i+1}$, the number of visits to state $i+1$ from state $i$ during $\tilde{f}_{i-1}^i$, the first passage time from $i$ to $i-1$, is governed by a geometric distribution. In particular,

$$P(\tilde{v}_{i,i+1} = n) = \left( \frac{\lambda_{N-i}}{\lambda_{N-i} + \mu_i} \right)^n \left( \frac{\mu_i}{\lambda_{N-i} + \mu_i} \right) \text{ for } n = 0, 1, ..., 1 \leq i \leq N - 1. \hspace{1cm} [3.27]$$

Thus, in general, if we define $\alpha_i = E[\tilde{v}_{i,i+1}]$, we find

$$\alpha_i = \frac{\lambda_{N-i}}{\mu_i}, \hspace{1cm} \text{for } 1 \leq i \leq N - 1, \hspace{1cm} [3.28]$$

and in particular,

$$E[\tilde{v}_{1,2}] = \frac{\lambda_{N-1}}{\mu_1}.$$

Finally, substituting [3.28] into [3.26], and using the fact that $E[\tilde{T}_{1,n}] = \frac{1}{\lambda_{N-1} + \mu_1}$, we get the expression in [3.24].

3. The Spares Provisioning Problem
Following arguments similar to those presented above, we obtain the general result

\[
\tilde{f}^{\mathcal{N}}_{i,-1} = \sum_{n=1}^{\tilde{v}_{i+1}} \tilde{T}_{i,n} + \sum_{n=0}^{\tilde{v}_{i+1}} \tilde{f}^{\mathcal{N}}_{i+1,n},
\]

[3.29]

\[
= \frac{1}{\mu_i} + \frac{\lambda_{i-1}}{\mu_i} \tilde{E}[\tilde{U}_{i+1,i}] \quad \text{for } 1 \leq i \leq N,
\]

where \(\lambda_0 = 0\) and \(\tilde{f}^{\mathcal{N}}_{0+1,0} = 0\). This expression reduces to the computational formula

\[
\tilde{E}[\tilde{U}_{i,-1}] = \begin{cases} 
\mu_i^{-1} + \alpha_i \tilde{E}[\tilde{U}_{i+1,i}], & \text{for } 1 \leq i \leq N - 1, \\
\mu_N^{-1}, & \text{for } i = N.
\end{cases}
\]

[3.30]

Upon recursively solving [3.30] for \(\tilde{E}[\tilde{U}_{N,N}] = \tilde{E}[\tilde{v}_N]\) starting with \(i = N\) and working backwards, we can easily obtain a closed form solution for the general case \(N \geq D\) and \(M \leq N\).

With respect to \(\tilde{v}_{i,l}\) for \(2 \leq i \leq N\), it is easy to see that

\[
\tilde{v}_{i,l} = \sum_{n=0}^{\tilde{v}_{i+1}} \tilde{v}_{i,l,n} \quad \text{for } 2 \leq i \leq N,
\]

where \(\tilde{v}_{2,n}\) denotes the number of visits from state 2 to state \(i\) following the \(n - 1\)th visit to state 2 from state 1, and before returning to state 1, during the busy period. Since \(\tilde{v}_{2,n}\) is independent of the number of visits to state 2, we find from Wald's equation that

\[
\tilde{E}[\tilde{v}_{i,l}] = \tilde{E}[\tilde{v}_{i,2}] \tilde{E}[\tilde{v}_{2,i}] \quad \text{for } 2 \leq i \leq N.
\]

[3.31]

Following similar reasoning, it is straightforward to establish by induction that

3. The Spares Provisioning Problem
\[ E[\tilde{\gamma}_i] = \prod_{j=2}^{l} E[\tilde{\gamma}_{j-1,i}] \quad \text{for } 2 \leq i \leq N. \] [3.32]


\[ E[\tilde{\gamma}_i] = E[U_{i-1}^{\tilde{\gamma}}] \prod_{j=2}^{l} E[\tilde{\gamma}_{j-1,i}] \quad \text{for } 2 \leq i \leq N. \] [3.33]

Upon combining [3.18] and [3.33], we find

\[ P(\text{at least } i \text{ machines demanding repair}) = \frac{E[U_{i-1}^{\tilde{\gamma}}] \prod_{j=1}^{i-1} \alpha_j}{E[\tilde{\gamma}_N] + E[\tilde{\gamma}]} \quad ; \quad i = 2, 3, \ldots, N. \] [3.34]

Thus, from [3.15] and [3.34], we obtain

\[ E[\tilde{\gamma}_i] = \frac{E[U_{i-1}^{\tilde{\gamma}}] \prod_{j=1}^{i-1} \alpha_j + \alpha_1 E[U_{i-1}^{\tilde{\gamma}}] + \alpha_2 E[U_{i-1}^{\tilde{\gamma}}] + \cdots + \alpha_{M-1} E[U_{i-1}^{\tilde{\gamma}}]}{E[\tilde{\gamma}_N] + E[\tilde{\gamma}]} \cdot \] [3.35]

With the above expression, we can finally determine \( \lambda \), with the use of Little's result as shown in [3.8]. Algorithmically, we simply use [3.30] recursively, starting with \( i = N \), to obtain the first passage times required in [3.35], accumulating the sum in the numerator along the way while retaining only the most recently computed value for \( E[U_{i-1}^{\tilde{\gamma}}] \). The recursion terminates when \( i = 0 \), and at this point, we have \( E[\tilde{\gamma}_0] = E[\tilde{\gamma}_N] \) which is needed as the final term in the numerator and in the denominator.

3. The Spares Provisioning Problem
3.4 Numerical Examples

In this section, we present a detailed numerical example which illustrates use of our algorithm. Numerous additional numerical results are presented to show the performance of the algorithm. It will be seen that our Little's result approach generates excellent initial guesses and that the algorithm described earlier generally yields the required boundary points within a few iterations.

For our illustrative example, we choose the system parameters as follows: $\lambda = 1$, $\mu = 2$, $D = 4$ and $n^*_i = 3.5$. From [3.6] and [3.7], we obtain the initial solution $M_0 = 2$ and $N_0 = 6$. With two repair stations, we must compute the average length of the ordinary busy period $E[\tilde{\nu}_i]$ and the average first passage time from state 2 to state 1, $E[\tilde{\nu}_{2,1}]$. From the previous section, by applying [3.30] recursively starting with $i = 6$ and decreasing, we find

\[
E[\tilde{\nu}_{6,3}] = \frac{1}{\mu_6} = \frac{1}{2\mu} = 0.25000,
\]

\[
E[\tilde{\nu}_{5,4}] = \frac{1}{\mu_5} + \alpha_5 E[\tilde{\nu}_{6,3}] = \frac{1}{2\mu} + \frac{\lambda}{2\mu} E[\tilde{\nu}_{6,5}] = 0.31250,
\]

\[
E[\tilde{\nu}_{4,3}] = \frac{1}{\mu_4} + \alpha_4 E[\tilde{\nu}_{5,4}] = \frac{1}{2\mu} + \frac{2\lambda}{2\mu} E[\tilde{\nu}_{5,4}] = 0.40625,
\]

\[
E[\tilde{\nu}_{3,2}] = \frac{1}{\mu_3} + \alpha_3 E[\tilde{\nu}_{4,3}] = \frac{1}{2\mu} + \frac{3\lambda}{2\mu} E[\tilde{\nu}_{4,3}] = 0.55469,
\]

\[
E[\tilde{\nu}_{2,1}] = \frac{1}{\mu_2} + \alpha_2 E[\tilde{\nu}_{3,2}] = \frac{1}{2\mu} + \frac{4\lambda}{2\mu} E[\tilde{\nu}_{3,2}] = 0.80469,
\]

and

\[
E[\tilde{\nu}_6] = \frac{1}{\mu} + \frac{4\lambda}{\mu} E[\tilde{\nu}_{2,1}] = 2.10938.
\]

From [3.35], we compute the expected number of busy repair stations. We find
\[ E[\tilde{n}_0] = \frac{E[\tilde{y}_0] + \alpha_1 E[\tilde{y}_{2,1}]}{E[\tilde{y}_0] + E[\tilde{y}]} = 1.57616, \]

where \( E[\tilde{y}] = \frac{1}{4\lambda} = 0.25 \). From [3.8], we determine the aggregate failure rate to be \( \lambda_0 = \mu E[\tilde{n}_0] = 3.15232 \), which is less than our target of 3.5. Thus, we increase the number of machines in the system according to step 3 of our iterative procedure. From [3.9], we find

\[
N_1(2) = \begin{bmatrix} \frac{\lambda^*}{\lambda_0} & N_0(2) \end{bmatrix}^+ \\
\begin{bmatrix} 3.5 \\ 3.15232 \end{bmatrix}^+ = 7.
\]

With the current solution point (2,7), we compute \( E[\tilde{n}_{1,2}], E[\tilde{n}_{1,3}], ..., E[\tilde{n}_{1,6}] \) according to [3.30] as illustrated above. Then, from [3.33], we find \( E[\tilde{n}]_0 = 1.65027 \), which yields an aggregate failure rate of \( \lambda_0 = 3.30054 \). Since \( \lambda_0 \) does not meet our requirement, we obtain \( N_0(2) = 8 \) from [3.9] and iterate. The iterative procedure generates the sequence \( \{N_0(2), N_1(2), N_2(2), N_3(2), N_4(2)\} = \{6, 7, 8, 9, 10\} \) and corresponding aggregate failure rates \( \{\lambda_0, \lambda_1, \lambda_2, \lambda_3, \lambda_4\} = \{3.15232, 3.30054, 3.40466, 3.48178, 3.54122\} \). Since \( N_4(2) = 10 \) yields \( \lambda_4 = 3.541 > 3.5 \), the boundary point is \( (M_0, N(M_0)) = (2,10) \).

Since \( N(M_0) > N_0(M_0) \), we start with another initial solution (3,6). The expected number of busy repair stations corresponding to this solution is \( E[\tilde{n}]_0 = 1.7602 \), which yields \( \lambda_0 = 3.5204 > 3.5 \). Since \( N(m) = N_0(m) \), there is no need to increase \( m \) further. Depending on the objective cost function, we will choose either \( (N,M) = (2,10) \) or \( (3,6) \) as the optimal solution. This concludes the detailed example.

Numerous examples have been solved using a simple FORTRAN program. Some of the results are reported in Table 2. We looked at the different cases where the utilization rate \( \rho = \frac{\lambda}{\mu} \) was varied from 10\% to 100\%, and as \( \rho \) increases, the number of feasible boundary points would tend to increase. The increase, however, was not dramatic, and we can easily enumerate all the

3. The Spares Provisioning Problem
points to determine the optimum. The increase in boundary points was also evident as the difference between \( n_i \) and \( D \) becomes smaller. It is interesting to note that even if \( D = 1000 \) with high utilization rates, convergence to the boundary of the feasible solution space would rarely require more than six iterations. This appears to indicate that Little's result generates excellent initial solution.

We note in passing that there are numerous possible variations on the iterative procedure discussed above. For example, we implemented an alternate procedure in which successive values of \( N_{i+1}(m) \) were computed by using the slope of the line that connects the last two points generated by the iterative procedure. Mathematically, at the \((i+1)st\) iteration, a new point is obtained as follows:

\[
N_{i+1}(m) = \left[ N_i(m) + (\lambda_g - \lambda_f) \frac{N_i(m) - N_{i-1}(m)}{\lambda_f - \lambda_{f-1}} \right]^+. \tag{3.36}
\]

From our previous discussion on convergence, it can be easily shown that the slope \( \frac{\lambda_f - \lambda_{f-1}}{N_i(m) - N_{i-1}(m)} \) is always less than the slope of the line that passes through the origin, \( \frac{\lambda_f}{N_i(m)} \), when \( i = 1, 2, \ldots \). Furthermore, due to [3.14], the new slope will never be less than \( \frac{d\lambda_f}{dN} \) evaluated at \( N = N_i(m) \). As a result, the sequence \( \{N_{i+1}(m), i \geq 0\} \) generated by [3.36] is guaranteed to converge to the boundary of the feasible solution space at least as fast as the alternate sequence generated by [3.9]. For example, for the initial solution (91,180) from Table 2, the boundary point (91,191) was reached in six iterations when the sequence \( \{N_{i+1}(m), i \geq 0\} \) was generated by [3.9] while only four iterations were required when the sequence was generated by [3.36]. Judging from the computational experience of the procedure illustrated in this section, however, the computation time for a problem of almost any size is negligible. On the other hand, the iterative idea is valid in many situations, some of which would require extensive computation at each iteration. In that case, or in case the current solution technique is implemented on a nonprogrammable hand calculator, the advantages of saving iterations are obvious.

3. The Spares Provisioning Problem
| $D$ | $n^*_D$ | $ho$ | Initial Solution $(M,N_0(M))$ | Boundary Point $(M,N(M))$ | Number of Iterations |
|-----|-----|-----|-----------------|-------------------|------------------|
| 10  | 8   | 0.1 | (1,9)           | (1,11)            | 2                |
|     |     |     | (2,9)           | (2,9)             | 0                |
|     |     | 0.2 | (2,10)          | (2,11)            | 1                |
|     |     |     | (3,10)          | (3,10)            | 0                |
|     |     | 0.5 | (5,12)          | (5,13)            | 1                |
|     |     |     | (12,12)         | (12,13)           | 1                |
|     |     | 1.0 | (9,16)          | (9,18)            | 2                |
|     |     |     | (10,16)         | (10,17)           | 1                |
|     |     |     | (16,16)         | (16,17)           | 1                |
| 50  | 40  | 0.1 | (5,44)          | (5,46)            | 1                |
|     |     |     | (6,44)          | (6,45)            | 1                |
|     |     | 0.2 | (8,44)          | (8,44)            | 0                |
|     |     |     | (9,48)          | (9,51)            | 2                |
|     |     | 0.5 | (10,48)         | (10,49)           | 1                |
|     |     |     | (13,48)         | (13,48)           | 0                |
|     |     | 1.0 | (21,60)         | (21,64)           | 2                |
|     |     |     | (22,60)         | (22,62)           | 1                |
|     |     |     | (23,60)         | (23,61)           | 1                |
|     |     |     | (28,60)         | (28,66)           | 0                |
| 100 | 90  | 0.1 | (10,99)         | (10,103)          | 2                |
|     |     |     | (11,99)         | (11,101)          | 1                |
|     |     |     | (12,99)         | (12,100)          | 1                |
|     |     | 0.2 | (14,99)         | (14,99)           | 0                |
|     |     |     | (19,108)        | (19,114)          | 3                |
|     |     |     | (20,108)        | (20,111)          | 2                |
|     |     |     | (21,108)        | (21,110)          | 1                |
|     |     |     | (22,108)        | (22,109)          | 1                |
|     |     |     | (26,108)        | (26,108)          | 0                |
|     |     | 0.5 | (46,135)        | (46,143)          | 4                |
|     |     |     | (47,135)        | (47,140)          | 3                |
|     |     |     | (48,135)        | (48,138)          | 2                |
|     |     |     | (49,135)        | (49,137)          | 1                |
|     |     |     | (51,135)        | (51,136)          | 1                |
|     |     | 1.0 | (135,135)       | (135,136)         | 1                |
|     |     |     | (91,180)        | (91,191)          | 6                |
|     |     |     | (92,180)        | (92,186)          | 3                |
|     |     |     | (93,180)        | (93,184)          | 2                |
|     |     |     | (94,180)        | (94,183)          | 2                |
|     |     |     | (96,180)        | (96,182)          | 2                |
|     |     |     | (98,180)        | (98,181)          | 1                |
|     |     |     | (180,180)       | (180,181)         | 1                |
Thus, the real computational advantage is noticeable only when the size of the population of machines is in the thousands, and only when the number of repair stations is much less than the number of machines. Under these situations, the alternate procedure could reach the boundary in less than half the number of iterations required by the existing procedure.
4. The Spares Provisioning Problem with Zero Parts Inventory

In this chapter, we consider the spares provisioning problem where machines become inoperable due to the failure of a critical built-in part. Before the repair activity can start on a machine, however, a consumable replacement part to replace the failed one must be obtained from the parts (ordering) subsystem. We assume that there is a permanent stockout of replacement parts in the ordering subsystem, that is, $S = 0$, and each time a machine fails it must wait for the arrival of a replacement part before proceeding to the repair subsystem. This is in contrast to the model discussed in the previous chapter, where upon failure, a machine is immediately transferred to the repair subsystem as if there are unlimited parts in stock. Thus, by applying the other extreme inventory policy where $S = \infty$, the machines will completely ignore the ordering subsystem. Figure 5 shows a block diagram for the spares provisioning system with zero parts inventory.
Figure 5. Flow Diagram for the Spares Provisioning Problem with Zero Parts Inventory.

The ordering policy for the parts is the lot-for-lot or \((S - 1, S)\) inventory policy, where again, we are considering the special case \(S = 0\). The maximum inventory level is zero, and whenever a failure occurs, an order is placed via the ordering subsystem to eliminate backorders and bring the inventory level back to zero. This ordering policy with \(S = 0\), in effect, reduces the ordering subsystem to an ample servers system, since all the machines in the ordering subsystem will receive immediate attention and orders of parts could cross.
The time required to obtain the part from the instant it is ordered is \( \bar{u} \). Thus, the time a machine spends in the parts subsystem is upper bounded by \( \bar{u} \), unless \( \bar{u} \) has an increasing failure rate in which case the time spent in this subsystem may be drawn from a distribution which dominates that of \( \bar{u} \). It is assumed that \( \bar{u} \) is drawn from the exponential distribution with rate \( \tau \).

Since \( S \) is fixed with the zero parts inventory policy, this will eliminate \( S \) as a decision variable. Therefore, when considering the operations costs of the overall system, we ignore the costs associated with the inventory of the parts. Our optimization problem then, is essentially the same as that given in Chapter 3, where \( TC \) is only a function of \( N \) and \( M \).

In the next section, a solution overview will be presented and how Little's result can be extended to the present model. In section 2 we use potentials from Markov chain theory to determine the expected number of visits to each state of the system during a typical "cycle". Using potentials and some elementary arguments from regenerative processes, in section 3 we develop a computationally efficient formula for the aggregate failure rate as has been done in the previous chapter. A detailed numerical example is given in section 4, with numerous of other numerical results, to show the efficiency of the optimizing routine. Finally, a good approximate model is presented in the last section which could be useful when solving extremely large problems.
4.1 Solution Overview

Let $\tilde{n}_i$ denote the number of machines waiting for parts in the ordering subsystem. Using the same notation as before, the machines in the various subsystems must sum to the total number of machines in the system as shown in Figure 6, so that

$$N = \tilde{n}_d + \tilde{n}_o + \tilde{n}_r + \tilde{n}_q + \tilde{n}_s.$$  

As before, we let $\lambda_j$ denote the average arrival rate to every subsystem in the entire system. If we consider the ordering subsystem in isolation and apply Little’s result in a manner similar to that of Chapter 3, we obtain an additional necessary condition for optimality which is

$$E[\tilde{w}_j] = \frac{\lambda_j}{\tilde{n}_j} \geq \lambda n_j^*.$$  \hspace{1cm} [4.1]

We can now readily obtain the lower bounds for the number of machines and repair stations that the system must own. In particular, the lower bound for $M$ is given by [3.4] since the expected number of busy repair stations that satisfies the availability constraint is the same, whether an ordering subsystem exists or not. A lower bound for $N$ would have to take the ordering subsystem into consideration since with probability one a machine will wait for a part. Therefore,

$$N \geq n_j^* + \frac{\lambda_j}{\mu} n_j^* + \frac{\lambda_j}{\mu} n_j^*.$$  \hspace{1cm} [4.2]

The monotonicity properties of the problem described above are not affected by the inclusion of the ordering subsystem. That is, the objective function and the availability constraint are still monotonically nondecreasing functions of $N$ and $M$. Therefore, the minimizing solution point lies on the boundary of the feasible solution space. Given this fact, our approach to solving this system...
is the same as the approach taken previously, which is to generate the set of points that comprise the boundary and choose the best one.

As before, each boundary point will be generated via an iterative routine. Let \((M_0, N_0(M_0))\) denote the initial value for the iterative routine. Then we may choose \(M_0\) as given in [3.6], and \(N_0\) as the lower bound specified by the right hand side of [4.2]. That is

\[
N_0(M_0) = \left[ n^*_g + \frac{\lambda}{\tau} n^*_g + \frac{\lambda}{\mu} n^*_g \right]^+.
\]  

[4.3]

By substituting [3.36] for [3.9] in step 3 of the algorithm in Chapter 3, the modified algorithm can be used to generate the boundary points that will result in an average of at least \(n^*_g\) deployed machines. By following the same logic as before, it can be shown that the convergence of the algorithm is guaranteed in a finite number of iterations. The only difference between the current model and the previous one is the cycle length for a typical machine. In the present model, a typical machine is either operating, waiting in the ordering subsystem, waiting in the repair queue, being repaired, or acting as a spare until it is needed again for deployment. The average cycle length then, is given by

\[
E[\tilde{C}] = E[\tilde{t}] + E[\tilde{u}] + E[\tilde{x}] + E[\tilde{w}]
\]

\[
= \frac{1}{\lambda} + \frac{1}{\tau} + \frac{1}{\mu} + E[\tilde{w}],
\]

[4.4]

where \(E[\tilde{w}]\) includes the total time that a typical machine spends both in the repair queue and acting as a spare.

Intuitively, with the insertion of the ordering subsystem, it would seem that the expected cycle length would be longer. But, since \(E[\tilde{w}]\) could possibly decrease as machines are held in the ordering subsystem, the cycle length would not necessarily have to be longer. However, it can be shown mathematically that the expected cycle length is indeed longer. As a result of this fact and in consideration of [3.11] and [3.12], for fixed \(M\) and \(N\), the proportion of time a typical machine spends operating and the aggregate failure rate will decrease when the ordering system is inserted.

4. The Spares Provisioning Problem with Zero Parts Inventory
Thus, for fixed $M$, more machines will be needed to maintain the same level of service provided with the unlimited stocking case.

Next, we show the computation of the potentials of the Markov chain that describes our system. The treatment of potentials follows that of Çinlar [20], p. 144.

### 4.2 Computation of Potentials

Each state of the system can be uniquely represented by the pair $(s,m)$, where $s$ is the number of machines waiting for parts, or equivalently, the number of orders of parts outstanding, and $m$ is the number of machines either being repaired or waiting for repair. Therefore, the number of machines operating or acting as spares is $N - s - m$. Thus, we can model the system as a Markov chain with state space $\mathcal{L} = \{(s,m): s,m = 0,1, \ldots , N \text{ and } s + m \leq N\}$. There is a finite number of states in the system, and every state can be reached from any other state; therefore, the Markov chain embedded at points of state transition, denoted by $X$, with state space $\mathcal{L}$ and one-step transition matrix $P = [P_{i,k,m,n}]$, is an irreducible chain. Hence, the embedded Markov chain is ergodic and all states are recurrent nonnull. In the limit, all states will be visited an infinite number of times.

For a system such as ours, we define a cycle to be the time starting at the instant the system is in state $(0,0)$, and terminates at the instant the system goes back to state $(0,0)$. Furthermore, every time the system is in state $(0,0)$, the whole process renews itself so that different cycles will be probabilistically identical. As discussed in Ross [74], such a process is referred to as a regenerative
process, where the renewal epochs are embedded at the times when the system makes a transition to state (0,0).

To study the behavior of the system during a typical cycle, we modify the embedded Markov chain \( X \), so that the state (0,0) is an absorbing state and the rest of the states are transient, as shown in Figure 6. If the system is initially in state (1,0), then it is of interest to know the expected number of times the transient states will be visited before reaching state (0,0). Knowing that, and the fact that the times the process spends in each state are independent from the number of visits to the states, we use Wald's equation to determine the expected amount of time the system spends at each state and the expected length of a cycle. With the modified Markov chain, the expected cycle length is identical to the expected first passage time from state (0,1) to state (0,0). Finally, using the expected cycle length, we can determine the proportion of time the system spends in a state or class of states, which is necessary for the computation of the aggregate failure rate as will be shown in the next section.

Determining the expected number of visits to each state amounts to obtaining the potential matrix of the modified chain, which will be denoted by \( R \). \( R(u,v) \), which is the \((u,v)\)-entry of the matrix \( R \), is defined as the expected number of visits to \( v \) starting at \( u \). Let \( Q \) and \( T \) be the matrices obtained from \( P \) and \( R \) respectively, by deleting all the rows and columns corresponding to the recurrent states, and rearranging the states so that the recurrent states precede the transient ones, it can be shown that

\[
R = \sum_{n=0}^{\infty} P^n = \begin{bmatrix} \sum_{n} k^n & 0 \\ \sum_{n} L_n & \sum_{n} Q^n \end{bmatrix},
\]

[4.5]
Figure 6. The Modified Markov Chain $X$

where the submatrix $K$ contains the transition probabilities within the set of recurrent states. Since, in our case we only have one recurrent state $(0,0)$, $K = [1]$. From the relationship above we can also conclude that $T = \sum_{n=0}^{\infty} Q^n$, from which we can obtain the computational formula

$$T(I - Q) = I. \quad [4.6]$$

With a finite number of transient states, the unique solution for $T$ is $(I - Q)^{-1}$. 

4. The Spares Provisioning Problem with Zero Parts Inventory
By imposing the condition that the initial state of the system is \((1,0)\), and letting 
\(\tilde{r}_{ij} = R((1,0),(i,j))\) where \((i,j) \neq (0,0)\), then from the above discussion we can determine the expected number of visits to any of the transient states by solving the following system of equations:

\[
\tilde{r}^N = \tilde{r}^N Q_N + e_1,
\]

where \(\tilde{r}^N = [\tilde{r}_{i1}^N, \tilde{r}_{i2}^N, \ldots, \tilde{r}_{iN}^N]\), \(Q_N\) is the reduced one-step transition matrix for the embedded Markov chain as described earlier, and \(e_1\) is the vector \([1,0,\ldots,0]\), where the first entry corresponds to the state \((1,0)\). \(e_1\) represents the fact that the initial state for the transient chain is \((1,0)\), since the cycle starts at the instant the system makes a transition from state \((0,0)\) to state \((1,0)\). The following theorem provides an explicit solution to \([4.7]\).

**Theorem 4.1:** For \(N \geq 2\)

\[
\tilde{r}_{ij}^N = \prod_{m=1}^{j-1} \left( \frac{\lambda_{N-m} \tau_m}{\mu_n \tau_m} \right) \left( \frac{\lambda_{N-j} + \mu_j}{\mu_j} \right); \quad i = 0; j = 1, \ldots, N - 1
\]

\[
= \prod_{m=1}^{j-1} \left( \frac{\lambda_{N-m}}{\mu_n \tau_m} \right) \left( \frac{\lambda_{N-j} + \tau_l}{\tau_l} \right); \quad j = 0; i = 1, \ldots, N - 1
\]

\[
= \prod_{n=1}^{N-1} \left( \frac{\lambda_{N-n}}{\mu_n} \right); \quad i = 0; j = N
\]

\[
= \prod_{m=1}^{i-1} \left( \frac{\lambda_{N-m}}{\mu_n \tau_m} \right) \left( \frac{\lambda_{N-i} \tau_m + \mu_j}{\tau_l} \right); \quad i + j < N; i,j \neq 0
\]

\[
= \prod_{n=1}^{j-1} \left( \frac{\lambda_{N-j} \tau_m + \mu_j}{\tau_l} \right); \quad i + j = N; i,j \neq N
\]
where $\frac{\lambda_n}{\mu_n} = 1$ if $n < 1$, $\frac{\lambda_m}{\mu_2} = 1$ if $m_1 < 1$ or $m_2 < 1$, and $\tau_m = m\tau$.

**Proof:** The proof of the theorem amounts to showing that the potential equations are satisfied. This can be accomplished by straightforward substitution. We will demonstrate our technique with only one specific case; the remaining cases can be proved in a similar manner.

Suppose the system is in state $(i,j)$, where $i + j < N$ and $ij \neq 0$ as shown in Figure 7. The potential equation for state $(i,j)$ is

$$
\bar{n}_{ij}^N = \frac{\lambda_{N-i-j+1}}{\lambda_{N-i-j+1} + \tau_{i-1} + \mu_j} \bar{n}_{i-1,j+1}^N + \frac{\tau_{i+1}}{\lambda_{N-i-j} + \tau_{i+1} + \mu_{j-1}} \bar{n}_{i+1,j}^N + \\
\frac{\mu_{j+1}}{\lambda_{N-i-j-1} + \tau_{i} + \mu_{j+1}} \bar{n}_{i,j+1}^N
$$

Figure 7. Transition Flow Diagram for state $(i,j)$.
Without loss of generality, we assume that \((i + j + 1) < N\), \(i - 1 > 0\), and \(j - 1 > 0\). Substituting the expressions for \(\bar{r}_{i-1,j}^N\), \(\bar{r}_{i+1,j-1}^N\) and \(\bar{r}_{i+1,j}^N\) from the theorem, we have

\[
\bar{r}_{i,j}^N = \frac{\lambda_{N-i-j+1}}{\lambda_{N-i-j+1} + \tau_{i-1} + \mu_j} \frac{\lambda_{N-j}}{\mu_j} \frac{\lambda_{N-j-1}}{\tau_1} \cdots \frac{\lambda_{N-i-j+2}}{\tau_{i-2}} \frac{\lambda_{N-i-j+1} + \tau_{i-1} + \mu_j}{\tau_{i-1}}
+ \frac{\tau_{i+1}}{\lambda_{N-i-j} + \lambda_{N-j+1} + \mu_{j-1}} \frac{\lambda_{N-j}}{\mu_j} \frac{\lambda_{N-j-1}}{\tau_1} \cdots \frac{\lambda_{N-i-j+1} + \tau_{i+1} + \mu_{j-1}}{\tau_{i+1}}
+ \frac{\mu_{j+1}}{\lambda_{N-i-j-1} + \tau_i + \mu_{j+1}} \frac{\lambda_{N-j}}{\mu_j} \frac{\lambda_{N-j-1}}{\tau_1} \cdots \frac{\lambda_{N-i-j} + \tau_i + \mu_j + \mu_{j+1}}{\tau_i}.
\]

Simplifying, we get

\[
\bar{r}_{i,j}^N = \frac{\lambda_{N-i-j}}{\lambda_{N-i-j} + \tau_i + \mu_j} \frac{\lambda_{N-j}}{\mu_j} \frac{\lambda_{N-j-1}}{\tau_1} \cdots \frac{\lambda_{N-i-j+1}}{\tau_{i-1}}
+ \frac{\lambda_{N-i-j+1}}{\mu_j} \frac{\lambda_{N-j}}{\mu_j} \frac{\lambda_{N-j-1}}{\tau_1} \cdots \frac{\lambda_{N-i-j+1}}{\tau_{i}}
+ \frac{\lambda_{N-i-j+1}}{\mu_j} \frac{\lambda_{N-j}}{\mu_j} \frac{\lambda_{N-j-1}}{\tau_1} \cdots \frac{\lambda_{N-i-j+1}}{\tau_{i}}
= \frac{\lambda_{N-i-j}}{\mu_j} \frac{\lambda_{N-j}}{\mu_j} \frac{\lambda_{N-j-1}}{\tau_1} \cdots \frac{\lambda_{N-i-j+1}}{\tau_{i-1}} \left( \frac{1}{\tau_i} + \frac{\lambda_{N-j}}{\mu_j} \frac{1}{\mu_j} \right)
= \frac{\lambda_{N-i-j}}{\mu_j} \frac{\lambda_{N-j}}{\mu_j} \frac{\lambda_{N-j-1}}{\tau_1} \cdots \frac{\lambda_{N-i-j+1}}{\tau_{i-1}} \left( \frac{1}{\tau_i} + \frac{\lambda_{N-j}}{\mu_j} \frac{1}{\mu_j} \frac{1}{\tau_i} \right)
= \frac{\lambda_{N-i-j}}{\mu_j} \frac{\lambda_{N-j}}{\mu_j} \frac{\lambda_{N-j-1}}{\tau_1} \cdots \frac{\lambda_{N-i-j+1}}{\tau_{i-1}} \left( \frac{\lambda_{N-i-j} + \tau_i + \mu_j}{\tau_i} \right).
\]

The above expression agrees with that of the theorem. Following the same procedure, the rest of the cases can be proven just as easily.

An observation that is useful from a computational point of view is summarized in the following corollary, the proof of which is readily obtained by direct substitution.

**Corollary 4.1:** Let \(\sigma_{i,j}^k\) be the total departure rate from state \((i,j)\) with \(k\) machines in the system.

For \(N \geq 2\)

4. The Spares Provisioning Problem with Zero Parts Inventory 77
\[ \tilde{r}_{ij}^N = \tilde{r}_{ij}^{N-1} ; \quad i+j < N - 1 \]
\[ = \tilde{r}_{ij}^{N-1} \frac{\sigma_{ij}^N}{\sigma_{ij}^{N-1}} ; \quad i+j = N - 1 \]
\[ = \tilde{r}_{i-1,j}^{N-1} \frac{\sigma_{i-1,j}^N - \sigma_{i-1,j}^{N-1}}{\sigma_{i-1,j}^{N-1}} + \tilde{r}_{ij}^{N-1} \frac{\sigma_{ij}^{N-1} - \sigma_{ij}^N}{\sigma_{i,j}^{N-1}} ; \quad i+j = N \]

where \( \tilde{r}_{ij} = 0 \) if \( i+j < 0 \) and \( \tilde{r}_{ij} = 1 \).

### 4.2.1 Algorithmic Solution for Potentials

Before concluding this section, it is worth mentioning that the closed queueing network which describes our system is a cyclic network. That is, each machine in the system passes, in order, through the production floor, ordering and repair subsystems respectively, and no subsystem can be bypassed. It is well known that cyclic queues possess product-form solutions for the steady-state probabilities (see Appendix A), and more importantly, the detailed or local balance conditions at each state are satisfied. See Kelly [56] and Tijms [85] for further discussion.

Knowing that our system satisfies local balance is extremely helpful when we have to deal with the computation of potentials or the steady-state probabilities. For instance, if we need to obtain the potentials for a system that owns 50 machines, we need to solve 1325 equations simultaneously as shown in [4.7]. However, given local balance, we can easily solve for [4.7] no matter how many machines are in the system!

It is possible for the system to be in state \((i,j)\) as shown in Figure 7. To compute the potential for state \((i,j)\), we first set up the global balance condition for that state. That is

4. The Spares Provisioning Problem with Zero Parts Inventory
\[
\bar{N}_{i,j}^N = \frac{\lambda_{N-i-j+1}}{\lambda_{N-i-j+1} + \tau_{i-1} + \mu_j} \bar{N}_{i-1,j}^N + \frac{\tau_{i+1}}{\lambda_{N-i-j} + \tau_{i+1} + \mu_{j-1}} \bar{N}_{i+1,j-1}^N + \frac{\mu_{j+1}}{\lambda_{N-i-j-1} + \tau_i + \mu_{j+1}} \bar{N}_{i,j+1}^N.
\] [4.8]

From the local balance conditions we know that the rate out of state \((i,j)\) due to failures, must equal the rate into state \((i,j)\) due to repairs, since traffic in the system is cyclic. Thus,

\[
\frac{\mu_{j+1}}{\lambda_{N-i-j-1} + \tau_i + \mu_{j+1}} \bar{N}_{i,j+1}^N = \frac{\lambda_{N-i-j}}{\lambda_{N-i-j} + \tau_{i} + \mu_{j}} \bar{N}_{i,j}^N.
\] [4.9]

Substituting [4.9] into [4.8], yields

\[
\bar{N}_{i,j}^N = \frac{\lambda_{N-i-j+1}}{\lambda_{N-i-j+1} + \tau_{i-1} + \mu_j} \bar{N}_{i-1,j}^N + \frac{\tau_{i+1}}{\lambda_{N-i-j} + \tau_{i+1} + \mu_{j-1}} \bar{N}_{i+1,j-1}^N + \frac{\lambda_{N-i-j}}{\lambda_{N-i-j} + \tau_i + \mu_{j}} \bar{N}_{i,j}^N.
\] [4.10]

Obviously, [4.8] and [4.10] will give the same value for \(\bar{N}_{i,j}^N\), where [4.10] completely ignores the state \((i,j+1)\) by introducing a self loop for state \((i,j)\) at a rate \(\frac{\lambda_{N-i-j}}{\lambda_{N-i-j} + \tau_i + \mu_{j}}\). Since the state \((i,j)\) could be any state in the system, with the exception of \((0,0)\) and the outer layer of states; i.e., \(\{(i,j) : i + j = N\}\), the implication is that the potentials for any subset of states, \(C_n = \{(i,j) : i + j \leq n \text{ and } 0 < n < N\}\), can be obtained by solving a reduced system. Consequently, the potentials can be obtained iteratively as shown below.

Algorithmically, to obtain \(\bar{N}_N^N\) for any \(N \geq 2\), we start by solving for the most inner states in the two-dimensional Markov chain; \((1,0)\) and \((0,1)\) as shown in Figure 8. Therefore, we have
Figure 8. The Modified Markov Chain with Self Loops.

\[
\begin{align*}
\tilde{N}_{1,0}^N &= 1 + \frac{\lambda_{N-1}}{\lambda_{N-1} + \tau_1} \tilde{N}_{1,0}^N, \\
\tilde{N}_{0,1}^N &= \frac{\tau_1}{\lambda_{N-1} + \tau_1} \tilde{N}_{1,0}^N + \frac{\lambda_{N-1}}{\lambda_{N-1} + \mu_1} \tilde{N}_{0,1}^N.
\end{align*}
\]

The system of equations in [4.11] yields the solution

\[
\tilde{N}_{1,0}^N = \frac{\lambda_{N-1} + \tau_1}{\tau_1}, \quad \text{and} \quad \tilde{N}_{0,1}^N = \frac{\lambda_{N-1} + \mu_1}{\mu_1},
\]

which is the same as that given by Theorem 4.1.

Using [4.12], we can immediately solve for the next layer of states: (2,0), (1,1) and (0,2), which will in turn solve for the next layer and so on. We follow the same procedure until we reach the outer layer of states where, of course, no self loops are introduced.
4.3 Computation of Aggregate Failure Rate

The number of visits to a transient state \((i,j)\) is independent of the time the system spends in that state during each visit, and all the times in the system are exponentially distributed. From Wald's equation then, we find that the total expected time the system spends in state \((i,j)\) before absorption into state \((0,0)\) is

\[
E[\tilde{t}_{ij}^N] = \tilde{n}_{ij}^N \frac{1}{\rho_{ij}^N},
\]

where \(\tilde{t}_{ij}^N\) represents the total time spent in \((i,j)\) by a system having \(N\) machines during the first passage time from \((0,1)\) to \((0,0)\).

Let \(\rho(i,j)\) denote the steady-state probability that the system is in state \((i,j)\), then from [4.13] and the theory of regenerative processes (Ross [74]),

\[
\rho(i,j) = \frac{E[\text{amount of time in state } (i,j) \text{ during a cycle}]}{E[\text{time of a cycle}]} \quad \text{[4.14]}
\]

where a cycle is as defined in the previous section.

If we define an idle state to be any of the states \((i,0); \ i = 0,1,...,N\), then it is possible that the process will alternate between idle states and busy states during a cycle, where a busy state is considered to be any state such that at least one machine is demanding repair. We define \(E[\tilde{t}]\) as the total expected amount of time spent in the idle states during a cycle. Then,
\[ E[\tilde{t}] = \sum_{i=0}^{N} E[\tilde{t}_{i,0}^N] = \frac{1}{\lambda_N} + \sum_{i=1}^{N} E[\tilde{t}_{i,0}^N], \]  

[4.15]

where \( E[\tilde{t}_{i,0}^N] \) is as defined in [4.13], and \( E[\tilde{t}_{0,0}^N] = \frac{1}{\lambda_N} \). Similarly, we define \( E[\tilde{y}_N] \) to be the total expected amount of time spent in the busy states during a cycle. So that,

\[ E[\tilde{y}_N] = \sum_{(i,j), j>0} E[\tilde{t}_{i,j}^N]. \]  

[4.16]

Thus, from the theory of regenerative processes, we find that

\[ P(\text{at least 1 repair station busy}) = \frac{E[\tilde{y}_N]}{E[\tilde{y}_N] + E[\tilde{t}]}, \]  

[4.17]

and

\[ P(\text{at least } j \text{ repair stations busy}) = \frac{E[\tilde{s}_j]}{E[\tilde{y}_N] + E[\tilde{t}]} \quad \text{for } 1 \leq j \leq M, \]  

[4.18]

where \( \tilde{s}_j \) denotes the aggregate amount of time during a cycle where at least \( j \) machines are demanding repair. That is,

\[ E[\tilde{s}_j] = \sum_{(i,k), k \geq j} E[\tilde{t}_{i,k}^N] \quad \text{for } 1 \leq j \leq N; \ 0 \leq i \leq N - 1. \]  

[4.19]

We can now compute the expected number of busy repair stations by summing [4.19] over the appropriate values of \( j \), and dividing the sum by the expected cycle length. That is,
\[ E[\tilde{n}_r] = \sum_{j=1}^{M} P(\text{at least } j \text{ repair stations busy}) \]
\[ = \sum_{j=1}^{M} E[\tilde{\gamma}_j] \]
\[ = \frac{E[\tilde{\gamma}_N]}{E[\tilde{\gamma}_N] + E[\tilde{\gamma}]} \quad \text{[4.20]} \]

Once \( E[\tilde{n}_r] \) is computed, \( \lambda_r \) is readily obtained from Little's result. However, we will now use the principle of local balance to obtain a more computationally efficient formula for \( E[\tilde{n}_r] \) as has been done in Chapter 3.

Suppose that the system owns \( N \) machines and we need to compute \( E[\tilde{T}] \). We already know that \( E[\tilde{t}_{0,0}] = \frac{1}{\lambda_N} \) and from [4.12] and [4.13], \( E[\tilde{t}_{0,0}] = \frac{1}{\tau_1} \). We also know from the principle of local balance that \( \lambda_{N-1} p(i-1,0) = \tau_i p(i,0) \), for \( i = 2, \ldots, N \). Hence,

\[ p(i,0) = \frac{\lambda_{N-1}}{\tau_i} p(i-1,0) \quad \text{for } i = 2, \ldots, N \quad \text{[4.21]} \]

and

\[ p(1,0) = \frac{1}{\tau_1} \]
\[ = \frac{1}{E[\tilde{\gamma}_N] + E[\tilde{T}]} \quad \text{[4.22]} \]

From the relationship in [4.21], we find the computational formula for \( E[\tilde{T}] \) to be

\[ E[\tilde{T}] = \frac{1}{\lambda_N} + E[\tilde{\gamma}] \quad \text{[4.23]} \]

where

\[ E[\tilde{T}_{i,0}] = \begin{cases} \frac{1}{\tau_i} + \frac{\lambda_{N-1}}{\tau_i} E[\tilde{T}_{i+1,0}] ; & \text{for } i = 1, \ldots, N - 1 \\ \frac{1}{\tau_N} ; & \text{for } i = N \end{cases} \quad \text{[4.24]} \]
Applying the local balance principle on the busy states, we find that for \( j = 1, \ldots, N \), \( \tau_j p(1, j - 1) = \mu_j p(0, j) \); and for \( j = 1, \ldots, N - 1 \) and \( i = j + 1, \ldots, N - j \), \( \lambda_{N-i-j+1} p(i - 1, j) = \tau_i p(i, j) \).

Therefore,

\[
p(i, j) = \frac{\lambda_{N-i-j+1}}{\tau_i} p(i - 1, j) \quad \text{for } j = 1, \ldots, N - 1; \ i = j + 1, \ldots, N - j \tag{4.25}
\]

and

\[
p(0, j) = \frac{\tau_j}{\mu_j} p(1, j - 1) \quad \text{for } j = 1, \ldots, N. \tag{4.26}
\]

Let \( E[\tilde{\tau}_{j,j-1}] \) be the expected time spent in the states for which at least \( j \) machines are demanding repair during the first passage time from state \( (0, j) \) to \( (0, j - 1) \). Then, knowing that \( E[\tilde{\tau}_N] = E[\tilde{\tau}_{0,0}] \), from [4.25] and [4.26], we get

\[
E[\tilde{\tau}_{j,j-1}^N] = \begin{cases} 
\frac{1}{\mu_j} + \frac{\lambda_{N-j}}{\mu_j} E[\tilde{T}_i] + E[\tilde{\tau}_{i+1,j}], & \text{for } j = 1, \ldots, N - 1 \\
\frac{1}{\mu_N}, & \text{for } j = N
\end{cases} \tag{4.27}
\]

where \( E[\tilde{T}_i] \) is similar to [4.24], such that

\[
E[\tilde{T}_i] = \begin{cases} 
\frac{1}{\tau_i} + \frac{\lambda_{N-i-j}}{\tau_i} E[\tilde{T}_{i+1}^j], & \text{for } i = 1, \ldots, N - j - 1 \\
\frac{1}{\tau_{N-j}}, & \text{for } i = N - j.
\end{cases} \tag{4.28}
\]

As before, if we let \( \alpha_j = \frac{\lambda_{N-j}}{\mu_j} \) for \( j = 1, \ldots, N - 1 \), then from [4.18] and [4.27], we find that

\[
P(\text{at least } j \text{ repair stations busy}) = \frac{\prod_{i=1}^{j-1} \alpha_i}{E[\tilde{\tau}_N] + E[\tilde{\tau}_j]} \quad \text{for } j = 2, \ldots, M. \tag{4.29}
\]

Thus, from [3.15] and [4.29], we obtain the computational formula.
\[ E(\tilde{n}_i) = \frac{E(\tilde{v}_N) + \alpha_1(E(\tilde{v}_{x,1}) + \alpha_2(E(\tilde{v}_{x,2}) + \cdots + \alpha_{M-1}E(\tilde{v}_{x,M-1}) \cdots))}{E(\tilde{v}_N) + E(\tilde{v})} \]  \hspace{2cm} [4.30]

By using Little's result with [4.30] we can finally determine the aggregate failure rate \( \lambda_r \). Algorithmically, we start with \( j = N \) and use the recursive relationship in [4.27] in conjunction with [4.28], and obtain the first passage times required in [4.30]. As before, we accumulate the sum in the numerator of [4.30] while retaining only the most recent value of \( E(\tilde{v}_{j,j-1}) \). The recursion terminates when \( j = 1 \), where the final value computed is \( E(\tilde{v}_{1,x}) = E(\tilde{v}_N) \), which is needed both in the numerator and denominator. To obtain \( E(\tilde{v}) \), we simply use the recursion in [4.24] starting with \( i = N \) and terminating when \( i = 1 \), at which point we use [4.23].

### 4.4 Numerical Examples

A detailed numerical example is now presented to demonstrate how [4.30] is computed. The algorithm we use to generate the boundary points is the modified procedure from Chapter 3. Numerous additional numerical results are then given in Table 3 to show how well the algorithm performs under different conditions.

For our illustrative example, to compare the case of zero stock to the case of unlimited stock of parts, we choose the same system parameters we chose earlier; that is, \( \lambda = 1, \mu = 2, D = 4 \) and \( n'_1 = 3.5 \), with the additional parameter \( \tau = 3 \). From [3.6] and [4.3], we obtain the starting solution point \( (M_0, N_0(M_0)) = (2,7) \). By applying [4.27] and [4.28] recursively starting with \( j = 7 \) and decreasing, we get
\[ E[\tilde{\gamma}_{0,3}] = \frac{1}{\mu_{7}} = \frac{1}{2\mu} = 0.25000, \]
\[ E[\tilde{\gamma}_{0,5}] = \frac{1}{\mu_{6}} + \alpha_{5} \left[ E[\tilde{T}_{1,0}] + E[\tilde{\gamma}_{0,7}] \right] = \frac{1}{2\mu} + \frac{\lambda_{5}}{2\mu} \left[ 0.3333 + 0.25000 \right] = 0.39583, \]
\[ E[\tilde{\gamma}_{0,4}] = \frac{1}{\mu_{5}} + \alpha_{5} \left[ E[\tilde{T}_{1,3}] + E[\tilde{\gamma}_{0,5}] \right] = \frac{1}{2\mu} + \frac{2\lambda_{5}}{2\mu} \left[ 0.38889 + 0.39583 \right] = 0.64236, \]
\[ E[\tilde{\gamma}_{0,2}] = \frac{1}{\mu_{3}} + \alpha_{4} \left[ E[\tilde{T}_{1,2}] + E[\tilde{\gamma}_{0,5}] \right] = \frac{1}{2\mu} + \frac{3\lambda_{4}}{2\mu} \left[ 0.45679 + 0.64236 \right] = 1.07436, \]
\[ E[\tilde{\gamma}_{0,1}] = \frac{1}{\mu_{2}} + \alpha_{2} \left[ E[\tilde{T}_{1,2}] + E[\tilde{\gamma}_{0,5}] \right] = \frac{1}{2\mu} + \frac{4\lambda_{2}}{2\mu} \left[ 0.68249 + 2.75728 \right] = 2.75728, \]

and

\[ E[\tilde{\gamma}_{0,7}] = \frac{1}{\mu_{1}} + \alpha_{1} \left[ E[\tilde{T}_{1,1}] + E[\tilde{\gamma}_{0,5}] \right] = \frac{1}{\mu} + \frac{4\lambda_{1}}{\mu} \left[ 0.68249 + 2.75728 \right] = 7.37954. \]

To illustrate how the terms \( E[\tilde{T}_{1,i}] \) are computed, we show the detailed computation of \( E[\tilde{T}_{1,4}] \), and the rest of the terms will follow in a similar manner. From [4.28], starting with \( i = N - 4 = 3 \) and decreasing, we have

\[ E[\tilde{T}_{3,4}] = \frac{1}{\tau_{3}} + \frac{1}{3\tau_{3}} = 0.11111, \]
\[ E[\tilde{T}_{2,4}] = \frac{1}{\tau_{2}} + \frac{\lambda_{1}}{\tau_{2}} E[\tilde{T}_{3,4}] = \frac{1}{2\tau} + \frac{2\lambda_{1}}{2\tau} E[\tilde{T}_{3,4}] = 0.18519, \]

and

\[ E[\tilde{T}_{2,1}] = \frac{1}{\tau_{1}} + \frac{\lambda_{2}}{\tau_{1}} E[\tilde{T}_{2,4}] = \frac{1}{\tau} + \frac{2\lambda_{2}}{\tau} E[\tilde{T}_{2,4}] = 0.45679. \]

To determine \( E[\tilde{n}_{i}] \), we need to compute the remaining term in [4.30], which is \( E[\tilde{i}] \). From [4.23],

\[ E[\tilde{i}] = \frac{1}{\lambda_{7}} + E[\tilde{T}_{1,0}] = \frac{1}{4\lambda} + E[\tilde{T}_{1,0}] = 0.94459. \]

4. The Spares Provisioning Problem with Zero Parts Inventory
$E[\tilde{T}_{1,0}]$, which is equal to 0.69459, is obtained from [4.24] which is applied recursively starting with $i = 7$ and decreasing. Finally, from [4.30], the expected number of busy repair stations is

$$E[\tilde{n}_{i,0}] = \frac{E[\tilde{\gamma}_2] + \alpha_1 E[\tilde{\gamma}_{2,1}]}{E[\tilde{\gamma}_2] + E[\tilde{\gamma}_1]} = 1.54901.$$

From [3.8], we determine the aggregate failure rate to be $\lambda_{f_0} = \mu E[\tilde{n}_{i,0}] = 3.09802$, which is less than our target of 3.5. Thus, based on [3.36], we increase the number of machines in the system, so that

$$N_1(2) = \left[ N_0(2) + (\lambda_{f_0} - \lambda_{f_2}) \frac{N_0(2) - 0}{\lambda_{f_0} - 0} \right]^+ = \left[ 7 + \frac{3.5 - 3.09802}{3.09802} \right]^+ = 8.$$

Given the new solution point (2,8), we follow the same procedure as above and compute $E[\tilde{n}_{1,0}]$, $E[\tilde{n}_{2,0}]$, ..., $E[\tilde{n}_{i,0}]$ according to [4.27] and [4.28], and $E[\tilde{T}]$ according to [4.23]. Then, from [4.30], we find that $E[\tilde{n}_{i,0}] = 1.62957$ which yields an aggregate failure rate of $\lambda_{f_1} = 3.25914$. Since the current solution still does not meet our target, we increase the number of machines according to [3.36] and iterate. The iterative procedure generates the sequence \{N_0(2), N_1(2), N_2(2), N_3(2)\} = \{7,8,10,11\} and corresponding aggregate failure rates \{\lambda_{f_0}, \lambda_{f_1}, \lambda_{f_2}, \lambda_{f_3}\} = \{3.09802, 3.25914, 3.45844, 3.52299\}. Since $N_3(2) = 11$ yields $\lambda_{f_3} = 3.52299 > 3.5$, the first boundary point to be generated is $(M_0, N(M_0)) = (2,11)$.

Since $N(M_0) > N_0(M_0)$, we need to start with another initial solution (3,7). Based on the new starting point, the expected number of busy repair stations is $E[\tilde{n}_{i,0}] = 1.70120$, which yields $\lambda_{f_0} = 3.40240 < 3.5$. Increasing the number of machines based on [3.36], we get the new point (3,8) which yields $\lambda_{f_1} = 3.61435 > 3.5$. Thus, the second boundary point $(M_1, N(M_1)) = (3,8)$ has been generated. However, $N(M_1) > N_0(M_1)$, therefore, we need to start with another initial solution (4,7). We keep generating new boundary points until $N(M_i) = N_0(M_i)$, or the number of repair stations reaches $N_0(M_0)$. For our example, the set of boundary points will be

4. The Spares Provisioning Problem with Zero Parts Inventory 87
{(2,11),(3,8),(4,8),(5,8),(6,8),(7,8)} and based on our objective function, we choose the point which minimizes our costs. Note that the set of boundary points can be reduced to {(2,11),(3,8)} since the rest of the points are obviously not economically feasible, because they require more repair stations for the same number of machines. Finally, as we have conjectured earlier, when compared with the case of unlimited stock of parts, the case of zero stock will most likely require more machines to provide the same level of service. For instance, in our detailed example, when $S = \infty$, the set of boundary points is {(2,10),(3,6)} as compared to {(2,11),(3,8)} when $S = 0$.

Numerous examples have been solved, and some of the results are reported in Table 3. With the ordering system, there are two types of utilization rates that we have to deal with: $\rho_1 = \frac{\lambda}{\bar{r}}$ and $\rho_2 = \frac{\lambda}{\bar{r}}$. Both rates were varied from 10\% to 100\%, and as expected, the system behaved in the same manner as with the previous model. That is, more boundary points were generated as $\rho_1$ and $\rho_2$ approached 1. The most likely explanation for this behavior is that as the repair rate and the ordering rate approach the failure rate, this would make it necessary to increase the number of machines in the system. Since the number of machines tends to decrease with an additional repair station, as the number of machines increases, so does the number of tradeoffs between machines and repair stations. Finally, as Table 3 shows, convergence to the boundary of the feasible solution space is very fast, which strengthens our belief that Little's result produces excellent initial solutions.

4.5 Approximate Model

While more computations are required to solve for the aggregate failure rate when $S = 0$, the algorithm which generates the boundary points is extremely efficient in terms of computer storage.
Table 3. Performance of the Algorithm with Zero Parts Inventory.

<table>
<thead>
<tr>
<th>$D$</th>
<th>$n_s^-$</th>
<th>$\rho_1$</th>
<th>$\rho_2$</th>
<th>Initial Solution $(M,N_0(M))$</th>
<th>Boundary Point $(M,N(M))$</th>
<th>Number of Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>4</td>
<td>0.1</td>
<td>0.1</td>
<td>(1.5) (1.5)</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2</td>
<td>0.2</td>
<td>(1.6) (1.7)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(2.6) (2.6)</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5</td>
<td>0.5</td>
<td>(3.7) (3.7)</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.0</td>
<td>1.0</td>
<td>(5.9) (5.10)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(6.9) (6.9)</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1</td>
<td></td>
<td></td>
<td>(1.6) (1.6)</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td></td>
<td></td>
<td>(1.6) (1.8)</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td></td>
<td></td>
<td>(2.6) (2.6)</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td></td>
<td></td>
<td>(3.7) (3.8)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(4.7) (4.8)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(5.9) (5.10)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(9.9) (9.10)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>0.5</td>
<td>0.1</td>
<td></td>
<td></td>
<td>(1.7) (1.7)</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td></td>
<td></td>
<td>(1.7) (1.9)</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td></td>
<td></td>
<td>(2.7) (2.8)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td></td>
<td></td>
<td>(3.8) (3.9)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(4.8) (4.9)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(8.8) (8.9)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(5.10) (5.12)</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(6.10) (6.11)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(10.10) (10.11)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1.0</td>
<td>0.1</td>
<td></td>
<td></td>
<td>(1.9) (1.9)</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td></td>
<td></td>
<td>(1.9) (1.12)</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td></td>
<td></td>
<td>(2.9) (2.10)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(3.9) (3.10)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(9.9) (9.10)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(3.10) (3.11)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(4.10) (4.11)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(10.10) (10.11)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1.0</td>
<td>0.1</td>
<td></td>
<td></td>
<td>(5.12) (5.14)</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td></td>
<td></td>
<td>(6.12) (6.13)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td></td>
<td></td>
<td>(12.12) (12.13)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>17</td>
<td>0.1</td>
<td>0.1</td>
<td>(2.21) (2.23)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2</td>
<td>0.2</td>
<td>(3.21) (3.21)</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5</td>
<td>0.5</td>
<td>(4.23) (4.25)</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(5.23) (5.23)</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(9.28) (9.32)</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(10.28) (10.29)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(11.28) (11.28)</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

4. The Spares Provisioning Problem with Zero Parts Inventory
<table>
<thead>
<tr>
<th>$D$</th>
<th>$n^*_i$</th>
<th>$\rho_1$</th>
<th>$\rho_2$</th>
<th>Initial Solution $(M,N_6(M))$</th>
<th>Boundary Point $(M,N(M))$</th>
<th>Number of Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td>(18,36)</td>
<td>(18,39)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(19,36)</td>
<td>(19,38)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(20,36)</td>
<td>(20,37)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(36,36)</td>
<td>(36,37)</td>
<td>1</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1</td>
<td></td>
<td></td>
<td>(2,23)</td>
<td>(2,25)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(3,23)</td>
<td>(3,23)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(4,24)</td>
<td>(4,26)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(5,24)</td>
<td>(5,25)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(6,24)</td>
<td>(6,24)</td>
<td>0</td>
</tr>
<tr>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
<td>(9,29)</td>
<td>(9,34)</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(10,29)</td>
<td>(10,31)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(11,29)</td>
<td>(11,30)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(29,29)</td>
<td>(29,30)</td>
<td>1</td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td>(18,38)</td>
<td>(18,41)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(19,38)</td>
<td>(19,40)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(20,38)</td>
<td>(20,39)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(21,38)</td>
<td>(21,39)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(22,38)</td>
<td>(22,38)</td>
<td>0</td>
</tr>
<tr>
<td>0.5</td>
<td>0.1</td>
<td></td>
<td></td>
<td>(2,28)</td>
<td>(2,31)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(3,28)</td>
<td>(3,28)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(4,29)</td>
<td>(4,32)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(5,29)</td>
<td>(5,30)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(29,29)</td>
<td>(29,30)</td>
<td>1</td>
</tr>
<tr>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
<td>(9,34)</td>
<td>(9,39)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(10,34)</td>
<td>(10,36)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(11,34)</td>
<td>(11,35)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(34,34)</td>
<td>(34,35)</td>
<td>1</td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td>(18,43)</td>
<td>(18,47)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(19,43)</td>
<td>(19,45)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(20,43)</td>
<td>(20,44)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(43,43)</td>
<td>(43,44)</td>
<td>1</td>
</tr>
<tr>
<td>1.0</td>
<td>0.1</td>
<td></td>
<td></td>
<td>(2,36)</td>
<td>(2,40)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(3,36)</td>
<td>(3,37)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(36,36)</td>
<td>(36,37)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(4,38)</td>
<td>(4,41)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(5,38)</td>
<td>(5,39)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(6,38)</td>
<td>(6,38)</td>
<td>0</td>
</tr>
<tr>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
<td>(9,43)</td>
<td>(9,48)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(10,43)</td>
<td>(10,45)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(11,43)</td>
<td>(11,44)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(43,43)</td>
<td>(43,44)</td>
<td>1</td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td>(18,51)</td>
<td>(18,56)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(19,51)</td>
<td>(19,54)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(20,51)</td>
<td>(20,53)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(21,51)</td>
<td>(21,53)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(22,51)</td>
<td>(22,52)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(51,51)</td>
<td>(51,52)</td>
<td>1</td>
</tr>
</tbody>
</table>
and CPU time, and performs just as efficiently as when \( S = \infty \). However, before we conclude this chapter, we would like to present an approximate model to the case \( S = 0 \), so that for problems of moderate size, computation of optimal solutions via a nonprogrammable pocket calculator is feasible.

By fixing the stocking level at zero, we force each machine that fails to go through the ordering subsystem before requesting repair. The idea behind the approximate model is to combine the ordering subsystem with the working population (including the pool of spare machines) and choose an appropriate failure rate for the new subsystem as defined below. This, in effect, reduces the model to the spares provisioning problem with unlimited stock of parts, and there is no need for any new analysis to be performed. In addition, the two-dimensional Markov chain \( X \) which describes our system is now reduced to a one-dimensional chain which will vastly reduce the state space to \( (N + 1) \) states.

Define \( \lambda' \) to be the failure rate of a machine in the approximate model. Then, since the production floor and the ordering subsystem are in series, we set

\[
\lambda' = \left[ \frac{1}{\lambda} + \frac{1}{\tau} \right]^{-1}, \tag{4.31}
\]

which is the inverse of the average time required to traverse both systems. Now, in the actual system, a spare machine may replace a failed machine so that ordering and failure are taking place in parallel. The approximate model eliminates this parallelism so that the aggregate failure rate would be too low if shop capacity were maintained at \( D \). Thus, we must adjust \( D \) upwards to increase the aggregate failure rate. Let \( D' \) denote the maximum number of machines that can operate simultaneously. In particular, we adjust \( D \) to \( D' \) according to the following relationship,

\[
\lambda'D' = \lambda D. \tag{4.32}
\]
In computing $D'$, it should be remembered that $D'$ is an integer-valued parameter, and it should always be rounded up to the nearest integer if [4.32] yields any fractions. With $\lambda'$ and $D'$, we can determine $E[\bar{n}]$ and $\lambda_j$ in the same manner as we did before with the case $S = \infty$.

To test the approximate model, we have used the same set of examples reported in Table 3. As Table 4 shows, when comparing the approximate solutions obtained from the iterative procedure with the actual boundary points, we found that even if they are not the same, the approximate solution misses the actual by a total of not more than one machine for the extensive set of examples considered here. Furthermore, the approximate solution will never overshoot the actual one, since by combining the production floor and the ordering subsystem, as shown in [4.31], will increase the variance of the output process from the combined subsystem. With higher variability, there is a tendency for the repair queue to grow larger, which will in turn have a similar effect on $E[\bar{n}]$ and $\lambda_j$. As a result, the approximate solution will always be less than or equal to the actual solution. This indicates that for extremely large problems, an efficient technique for generating the boundary points may be to first use Little’s result as input to the approximate model, then use the approximate solutions as input to the exact model.
<table>
<thead>
<tr>
<th>$D$</th>
<th>$n^*_2$</th>
<th>$\rho_1$</th>
<th>$\rho_2$</th>
<th>Approximate Solution</th>
<th>Actual Solution</th>
<th>Number of Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>4</td>
<td>0.1</td>
<td></td>
<td>(1,5)</td>
<td>(1,5)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2</td>
<td></td>
<td>(1,7)</td>
<td>(1,7)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5</td>
<td></td>
<td>(2,6)</td>
<td>(2,6)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.0</td>
<td></td>
<td>(3,7)</td>
<td>(3,7)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(5,10)</td>
<td></td>
<td>(5,10)</td>
<td>(5,10)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(6,9)</td>
<td></td>
<td>(6,9)</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>0.2</td>
<td></td>
<td>0.1</td>
<td></td>
<td>(1,6)</td>
<td>(1,6)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2</td>
<td></td>
<td>(1,8)</td>
<td>(1,8)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5</td>
<td></td>
<td>(2,6)</td>
<td>(2,6)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3,8)</td>
<td></td>
<td>(3,8)</td>
<td>(3,8)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(4,7)</td>
<td></td>
<td>(4,7)</td>
<td>(4,7)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.0</td>
<td></td>
<td>(5,10)</td>
<td>(5,10)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(9,10)</td>
<td></td>
<td>(9,10)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>0.5</td>
<td></td>
<td>0.1</td>
<td></td>
<td>(1,7)</td>
<td>(1,7)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2</td>
<td></td>
<td>(1,9)</td>
<td>(1,9)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5</td>
<td></td>
<td>(2,7)</td>
<td>(2,7)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3,9)</td>
<td></td>
<td>(3,9)</td>
<td>(3,9)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(4,9)</td>
<td></td>
<td>(4,9)</td>
<td>(4,9)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(8,9)</td>
<td></td>
<td>(8,9)</td>
<td>(8,9)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.0</td>
<td></td>
<td>(5,11)</td>
<td>(5,12)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(6,11)</td>
<td></td>
<td>(6,11)</td>
<td>(6,11)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(10,11)</td>
<td></td>
<td>(10,11)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td>0.1</td>
<td></td>
<td>(1,9)</td>
<td>(1,9)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2</td>
<td></td>
<td>(1,11)</td>
<td>(1,12)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5</td>
<td></td>
<td>(2,9)</td>
<td>(2,10)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3,11)</td>
<td></td>
<td>(3,11)</td>
<td>(3,11)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(4,11)</td>
<td></td>
<td>(4,11)</td>
<td>(4,11)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(10,11)</td>
<td></td>
<td>(10,11)</td>
<td>(10,11)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.0</td>
<td></td>
<td>(5,13)</td>
<td>(5,14)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(6,13)</td>
<td></td>
<td>(6,13)</td>
<td>(6,13)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(12,13)</td>
<td></td>
<td>(12,13)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>17</td>
<td>0.1</td>
<td></td>
<td>(2,23)</td>
<td>(2,23)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2</td>
<td></td>
<td>(3,21)</td>
<td>(3,21)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5</td>
<td></td>
<td>(4,24)</td>
<td>(4,25)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(5,23)</td>
<td></td>
<td>(5,23)</td>
<td>(5,23)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(9,32)</td>
<td></td>
<td>(9,32)</td>
<td>(9,32)</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(10,29)</td>
<td></td>
<td>(10,29)</td>
<td>(10,29)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(11,28)</td>
<td></td>
<td>(11,28)</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>$D$</td>
<td>$n^*_i$</td>
<td>$\rho_1$</td>
<td>$\rho_2$</td>
<td>Approximate Solution</td>
<td>Actual Solution</td>
<td>Number of Iterations</td>
</tr>
<tr>
<td>-----</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------------------</td>
<td>----------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>1.0</td>
<td>(18,39)</td>
<td>(18,39)</td>
<td>2</td>
<td>(18,39)</td>
<td>(19,38)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>(19,38)</td>
<td>(19,38)</td>
<td>2</td>
<td>(18,39)</td>
<td>(20,37)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(20,37)</td>
<td>(20,37)</td>
<td>1</td>
<td>(24,36)</td>
<td>(24,36)</td>
<td>0</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1</td>
<td>(2,25)</td>
<td>(2,25)</td>
<td>2</td>
<td>(3,23)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>(3,23)</td>
<td>(3,23)</td>
<td>0</td>
<td>(4,26)</td>
<td>(4,26)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(4,26)</td>
<td>(4,26)</td>
<td>1</td>
<td>(5,25)</td>
<td>(5,25)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(5,25)</td>
<td>(5,25)</td>
<td>1</td>
<td>(6,24)</td>
<td>(6,24)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>(6,24)</td>
<td>(6,24)</td>
<td>0</td>
<td>(9,33)</td>
<td>(9,34)</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>(9,33)</td>
<td>(9,34)</td>
<td>3</td>
<td>(10,31)</td>
<td>(10,31)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>(10,31)</td>
<td>(10,31)</td>
<td>2</td>
<td>(11,30)</td>
<td>(11,30)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(11,30)</td>
<td>(11,30)</td>
<td>1</td>
<td>(14,29)</td>
<td>(14,30)</td>
<td>0</td>
</tr>
<tr>
<td>1.0</td>
<td>(18,41)</td>
<td>(18,41)</td>
<td>2</td>
<td>(19,39)</td>
<td>(19,40)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(19,39)</td>
<td>(19,40)</td>
<td>1</td>
<td>(21,38)</td>
<td>(21,39)</td>
<td>0</td>
</tr>
<tr>
<td>0.5</td>
<td>0.1</td>
<td>(2,30)</td>
<td>(2,31)</td>
<td>1</td>
<td>(3,28)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>(3,28)</td>
<td>(3,28)</td>
<td>0</td>
<td>(4,31)</td>
<td>(4,32)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(4,31)</td>
<td>(4,32)</td>
<td>1</td>
<td>(5,30)</td>
<td>(5,30)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(5,30)</td>
<td>(5,30)</td>
<td>1</td>
<td>(7,29)</td>
<td>(7,30)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>(7,29)</td>
<td>(7,30)</td>
<td>0</td>
<td>(9,38)</td>
<td>(9,39)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>(9,38)</td>
<td>(9,39)</td>
<td>2</td>
<td>(10,36)</td>
<td>(10,36)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(10,36)</td>
<td>(10,36)</td>
<td>1</td>
<td>(11,35)</td>
<td>(11,35)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(11,35)</td>
<td>(11,35)</td>
<td>1</td>
<td>(14,29)</td>
<td>(14,30)</td>
<td>0</td>
</tr>
<tr>
<td>1.0</td>
<td>(18,46)</td>
<td>(18,47)</td>
<td>2</td>
<td>(19,44)</td>
<td>(19,45)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(19,44)</td>
<td>(19,45)</td>
<td>1</td>
<td>(20,44)</td>
<td>(20,44)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(20,44)</td>
<td>(20,44)</td>
<td>1</td>
<td>(21,43)</td>
<td>(21,44)</td>
<td>0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.1</td>
<td>(2,39)</td>
<td>(2,40)</td>
<td>2</td>
<td>(3,37)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(3,37)</td>
<td>(3,37)</td>
<td>1</td>
<td>(4,36)</td>
<td>(4,37)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>(4,36)</td>
<td>(4,37)</td>
<td>0</td>
<td>(4,40)</td>
<td>(4,41)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(4,40)</td>
<td>(4,41)</td>
<td>1</td>
<td>(5,38)</td>
<td>(5,39)</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>(5,38)</td>
<td>(5,39)</td>
<td>0</td>
<td>(9,47)</td>
<td>(9,48)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>(9,47)</td>
<td>(9,48)</td>
<td>2</td>
<td>(10,44)</td>
<td>(10,45)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(10,44)</td>
<td>(10,45)</td>
<td>1</td>
<td>(11,44)</td>
<td>(11,44)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(11,44)</td>
<td>(11,44)</td>
<td>1</td>
<td>(12,43)</td>
<td>(12,44)</td>
<td>0</td>
</tr>
<tr>
<td>1.0</td>
<td>(18,55)</td>
<td>(18,56)</td>
<td>2</td>
<td>(19,53)</td>
<td>(19,54)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(19,53)</td>
<td>(19,54)</td>
<td>1</td>
<td>(20,52)</td>
<td>(20,53)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(20,52)</td>
<td>(20,53)</td>
<td>1</td>
<td>(21,52)</td>
<td>(21,53)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(21,52)</td>
<td>(21,53)</td>
<td>1</td>
<td>(22,52)</td>
<td>(22,52)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(22,52)</td>
<td>(22,52)</td>
<td>1</td>
<td>(51,52)</td>
<td>(51,52)</td>
<td>1</td>
</tr>
</tbody>
</table>
5. The Spares Provisioning Problem with Parts Inventory

So far, we have analyzed the spares provisioning problem with the two extreme inventory policies for the parts, \( S = 0 \) and \( S = \infty \). In this chapter, as Figure 9 shows, we consider the same model with the case of finite nonzero inventory stocking of the parts; i.e., \( 0 < S < \infty \). The ordering policy for the parts to be used here is the same one used in the previous chapter, which is the lot-for-lot or \((S-1, S)\) ordering policy. Therefore, the maximum inventory level of the parts is \( S \), and whenever a failure occurs, an order is placed via the ordering subsystem to bring the inventory level back to \( S \). Clearly, when \( S \in (0, \infty) \), there is a positive probability that a stockout of parts will occur, and in the event a machine becomes inoperable, it must wait in the ordering subsystem for the arrival of a new part. Similarly, a failed machine bypasses the ordering subsystem and proceeds immediately to the repair facility if parts are on hand. The proportion of machine failures that are delayed in the ordering subsystem depends on the state of the system and the system parameters: \( N, M, D, S, \lambda, \tau \) and \( \mu \). It is not yet clear how this proportion can be computed.
Figure 9. Block Diagram for the Spares Provisioning Problem with Parts Inventory.

In the present model, we have introduced an additional integer valued decision variable $S$. Hence, the total expected system cost $TC$ is now a function of $N, M$ and $S$, and as a consequence, the dimensionality and complexity of the optimization problem has increased. Furthermore, given that there is a state dependent routing of machine failures, as explained above, adds a new aspect to the spares provisioning problem and dramatically increases its difficulty. In fact, the queueing
network model that describes our new system is a non-product-form network, and specification of closed-form solutions is highly unlikely for such networks.

Organization of the current chapter is now described. In sections 1 and 2, we will briefly explain why the present model does not possess a product-form solution, and discuss some existing numerical methods that offer exact solutions for our system. In section 3, we then demonstrate via numerical examples that availability is a monotonically nondecreasing function of $S$, the stocking level of the parts. We also provide numerical examples that indicate that the marginal increase in availability with respect to $S$ is less than the marginal increase with respect to $N$. These properties are important for optimization purposes. Section 4 discusses a solution methodology for the present model. We begin with a direct solution method, which is computationally intense, and then an indirect method that drastically reduces the computational load by bounding the optimal solution. Finally, a detailed numerical example solved via both the direct and indirect methods is presented in section 5, and numerous numerical results are presented to demonstrate the efficiency of the indirect method.

5.1 Time Reversibility and Product-Form

To answer the question of what makes a queueing network a product-form or a non-product-form network, we should first understand the concept of time reversibility. Intuitively speaking, a stochastic process is reversible if it has the property that the behavior of the process remains the same when the direction of time is reversed. Formally, Kelly [56] states
Definition 5.1: A stochastic process $X(t)$ is reversible if, for $t_1 < t_2 < \cdots < t_n$,
$(X(t_1), X(t_2), \ldots, X(t_n))$ has the same distribution as $(X(T - t_1), X(T - t_2), \ldots, X(T - t_n))$ for all $t_1, t_2, \ldots, t_n, T \in \mathbb{R}$.

Based on the definition of reversibility, Kelly [56] then shows that a Markov chain or process is reversible, if and only if, there exists a collection of positive numbers summing to unity that satisfy the detailed (local) balance conditions of the chain or process.

A related concept is quasi-reversibility, which is essentially an input-output property of queues. As Walrand [88] explains, a stationary queue is said to be quasi-reversible if at any time the past departure times, the present state, and the future arrival times are independent. In the context of closed queueing networks, if a queue is, in isolation, quasi-reversible, then the network has a product-form equilibrium distribution. Another intriguing property of a closed network of quasi-reversible queues, is that a typical customer who jumps in the network would see the rest of the customers distributed in accordance with the equilibrium distribution which would obtain if they were the only customers in the network.

It is well known that cyclic queueing networks are not only networks of quasi-reversible queues, but they are also reversible. From the above discussion then, we find that the queueing networks that describe the models in Chapters 3 and 4, being cyclic queueing networks, are reversible networks. Therefore, the networks satisfy the principles of local balance, possess product-form solutions, and due to the last property stated above, we are able to derive all the recursive relationships needed to obtain the computational formulae for the aggregate failure rates. For the present model however, the cyclic property has been destroyed due to the finite inventory of parts in the ordering subsystem. Furthermore, the network is not quasi-reversible since the probability of stockout of parts is state dependent. Thus, the repair subsystem in isolation, for instance, would not be quasi-reversible. Hence, the queueing network model for the present system does not satisfy the local balance conditions, and there is a strong indication that it is a non-product-form network.
We now show that the present queueing network model does not have a product-form solution. Suppose we have a system where all the parameters \( N, M, D, \) and \( S \), are equal to to unity. Let \( p(i,j) \) denote the steady-state probability that the system is in state \((i,j)\). The global balance equations for such a system are

\[
\begin{align*}
\lambda p(0,0) &= \tau p(1,0) + \mu p(0,1) \\
(\lambda + \tau) p(1,0) &= \mu p(1,1) \\
\mu p(0,1) &= \tau p(1,1) \\
(\tau + \mu) p(1,1) &= \lambda p(0,0) + 2\tau p(2,0) \\
2\tau p(2,0) &= \lambda p(1,0)
\end{align*}
\]

and the normalizing condition is

\[
\text{s.t. } \sum_{i=0}^{1} \sum_{j=0}^{1} p(i,j) = 1.
\]

Solving the above system of equations will yield the steady-state probabilities:

\[
\begin{align*}
p(1,0) &= \left( \frac{\mu}{\lambda + \tau + \mu} \right) \left( \frac{\lambda}{\tau} \right) p(0,0) \\
p(0,1) &= \left( \frac{\lambda + \tau}{\lambda + \tau + \mu} \right) \left( \frac{\lambda}{\mu} \right) p(0,0) \\
p(1,1) &= \left( \frac{\lambda + \tau}{\lambda + \tau + \mu} \right) \left( \frac{\lambda}{\tau} \right) p(0,0)
\end{align*}
\]

and

\[
p(2,0) = \left( \frac{\mu}{\lambda + \tau + \mu} \right) \frac{1}{2!} \left( \frac{\lambda}{\tau} \right)^2 p(0,0).
\]

While there are similarities to a product-form solution, the above solution is clearly not product-form, since the probability \( p(1,1) \) does not have the term \( \frac{1}{\mu} \) that would correspond to a machine in repair. With the above example we can conclude that, in general, the network with finite nonzero stocking of parts is a non-product-form network.

5. The Spares Provisioning Problem with Parts Inventory
Before concluding this section, we note that if the probability of stockout of parts were fixed, i.e., it is not state dependent, then the network would be a network of quasi-reversible queues. For further discussion on reversibility and stochastic networks, the reader should refer to Kelly [56]. Also, Chandy et al. [18], and Chandy and Martin [19] provide some work on the intricate relationship between local balance and product-form queueing networks.

### 5.2 Numerical Methods

Knowing that the queueing network model which describes our system is a non-product-form network indicates that getting a closed-form solution for the equilibrium probabilities or any other performance measure of the system, might be difficult, if not impossible. There are, however, numerous iterative schemes which exist and to a certain extent, yield exact solutions to systems such as ours. The disadvantages of such methods is that large problems usually require excessive computer storage and CPU time, and in addition, problems suffer from numerical instability including floating point underflow and overflow and round off error. We encountered problems of this nature in the course of our numerical work and, consequently, developed alternate approaches. Before describing our approach, we briefly describe some of the iterative schemes that we have used to produce exact solutions for our system.

As mentioned earlier, there is a vast literature on numerical methods, and the list of techniques presented here is only a small fraction of what exists. Our purpose is not to discuss all the different methods, but simply to describe those methods we used, and report our experiences with them.
5.2.1 Gauss-Seidel and Overrelaxation Methods

The Gauss-Seidel method is probably one of the simplest and most reliable methods in existence. The idea behind this method is that given any set of linear equations $Ax = b$, where the square matrix $A$ and the vector $b$ are given, and $x$ is the vector of unknowns, the system of equations can be rewritten in the form

$$ (I - L - U)x = d, \tag{5.1} $$

where $I$ is the identity matrix, and $L$ and $U$ are the lower and upper triangular matrices, respectively, with zeros along their main diagonals. The system of equations in [5.1] can also be expressed in the form

$$ x = Lx + Ux + d. \tag{5.2} $$

For an iterative procedure, let $x^{(0)}$ be the vector obtained on the $n - th$ iteration, $n = 0, 1, \ldots$. Then [5.2] suggests the iteration scheme

$$ x^{(n+1)} = Lx^{(n+1)} + Ux^{(n)} + d. \tag{5.3} $$

where $x^{(0)}$ is assigned arbitrary values. Equation [5.2] reveals that each component of the vector $x^{(n+1)}$ is computed entirely from the vector $x^{(0)}$. Since the $j - th$ component $x_j^{(n+1)}$ is assumed to be closer to the true answer than $x_j^{(0)}$, the estimate for $x_j^{(n+1)}$ should be improved by replacing $x_j^{(0)}$ by $x_j^{(n+1)}$ whenever $j < i$ as shown in [5.3]. That is, we should use our most recent information as soon as it becomes available. This is the motivation for organizing the matrix $A$ in terms of $L$ and $U$.

A generalization of the Gauss-Seidel method, is the successive overrelaxation method, which weighs the most recent value of $x$ by a factor of $\omega \geq 1$. That is,

$$ x^{(n+1)} = \omega [Lx^{(n+1)} + Ux^{(n)} + d] + (1 - \omega)x^{(n)}; \quad \omega \geq 1. \tag{5.4} $$
Faster convergence might be achieved if [5.4] is used instead of [5.3], and experience shows that overrelaxation with \( \omega \approx 1.3 \) might be the fastest, as reported in [23].

If [5.1] represents the global balance equations of Markov chains, the vector \( \mathbf{x} \) represents the equilibrium probabilities, and the matrix \( \mathbf{A} \) represents the one-step transition probabilities or the infinitesimal generator. Cooper [23] explains, that a sufficient condition for convergence of the iteration scheme is that the matrix \( \mathbf{A} \) be irreducible and exhibit weak diagonal dominance. That is, the nonzero elements of \( \mathbf{A} \) should be concentrated along its main diagonal, so that faster convergence might be achieved. This is the strategy used when we needed to solve our system and, in general, convergence was achieved in a reasonable number of iterations. However, for large systems, a significant effort may be required to insure that matrix \( \mathbf{A} \) has weak diagonal dominance. In our work, we found that the Gauss-Seidel method applied directly to continuous-time Markov chains, resulted in a large variance in the number of iterations. For further discussion on the Gauss-Seidel method and its variations, the reader should refer to [11], [23], [87] and [89].

### 5.2.2 Aggregation-Disaggregation Methods

Simply stated, aggregation-disaggregation methods solve for the steady-state probabilities of Markov chains or processes by alternately solving an aggregated version and a disaggregated version of the problem. The idea is to reduce computer time and memory when the Markov chain (process) is very large. For a more formal statement on aggregation-disaggregation, we will now briefly discuss the procedure when applied to Markov chains. The treatment follows that of Schweitzer [78], where Markov processes are also considered.

Let \( \mathbf{P} = [p_{ij}] \) denote the one-step transition probability matrix of a Markov chain with finite state space \( \Omega = \{1, 2, \ldots, K\} \), and \( \pi_i \) denote the steady-state probability of the chain being in state \( i \in \Omega \). The global balance equations for the chain are
\[ \pi_i = \sum_{j \in \Omega} \pi_{j|i}, \quad i \in \Omega \]  

s.t. \( \sum_{i \in \Omega} \pi_i = 1 \). \[ 5.5 \]

To solve for [5.5], we partition \( \Omega \) into \( \bar{K} \) blocks (where typically \( \bar{K} < K \)), so that

\[ \Omega = \bigcup_{\alpha=1}^{\bar{K}} \Omega(\alpha), \] \[ 5.6 \]

and the aggregate probability of each block is now defined as

\[ \bar{\pi}_\alpha = \sum_{i \in \Omega(\alpha)} \pi_i, \quad 1 \leq \alpha \leq \bar{K}. \] \[ 5.7 \]

After partitioning \( \Omega \), we can replace the original equations in [5.5] by the pair

\[ \bar{\pi}_\alpha = \sum_{\beta=1}^{\bar{K}} \bar{p}(\pi)_{\beta\alpha}, \quad 1 \leq \alpha \leq \bar{K} \] \[ 5.8 \]

s.t. \( \sum_{\alpha=1}^{\bar{K}} \bar{\pi}_\alpha = 1 \)

and

\[ \pi_i = \sum_{j \in \Omega(\alpha)} \pi_{j|i} + \sum_{\beta=1}^{\bar{K}} \bar{\pi}_{\beta|i}, \quad i \in \Omega(\alpha), \quad 1 \leq \alpha \leq \bar{K}, \] \[ 5.9 \]

where for any \( X = (x_1, x_2, \ldots, x_K) > 0 \)

5. The Spares Provisioning Problem with Parts Inventory
\[ \bar{P}(\bar{z})_{\beta \alpha} = \frac{\sum_{j \in \Omega(\beta)} \sum_{i \in \Omega(\alpha)} x_{j \beta} P_{j \alpha}}{\sum_{k \in \Omega(\beta)} x_{k}}, \quad [5.10] \]

and

\[ P(\bar{z})_{\beta \beta}^+ = \frac{\sum_{j \in \Omega(\beta)} x_{j \beta} P_{j \beta}}{\sum_{k \in \Omega(\beta)} x_{k}}. \quad [5.11] \]

Solving the system in [5.5] is equivalent to solving the pair [5.8] and [5.9] where [5.8] is the aggregation aspect, and [5.9] is the disaggregation aspect. Algorithmically, we assign the vector \( \bar{x} \) some initial estimates and we then alternate between [5.8] and [5.9], starting with [5.8], until convergence is attained. When applied to our system, we partitioned the state space so that the resulting aggregated version of the chain looked like a simple birth-death process. Specifically, at the \( j - th \) level of repair, we lumped all the states \( \{(i,j)|i = 0, 1, \ldots, N - j\} \), so that the lumped process was similar to the case of unlimited stock of parts of Chapter 3. The number of blocks of such a partition would be \( N + 1 \) and the number of states in each block varies from 1 to \( N + S + 1 \). The chain could also be partitioned according to the level of orders of parts outstanding, or the level of machines operational.

Performance of the aggregation-disaggregation technique was not very consistent. For some sets of parameters the algorithm would converge, while for others it did not. This is consistent with Schweitzer’s experience as reported in [78]. We chose to abandon this approach for the current problem. Schweitzer [79] presents an excellent survey of aggregation-disaggregation methods, and it would be very helpful if this research area is of interest.
5.2.3 Carvalho's Method

Carvalho's method is a special case of aggregation-disaggregation that can solve for $\pi$ in one iteration, provided that the one-step transition probability matrix or the generator matrix have exploitable structures. As discussed above, our system has the special structure needed, since by lumping the states in a certain way produces a birth-death process. However, the one-step transition matrix should be modified by adding dummy states to the Markov chain, so that the resulting one-step transition matrix is a block tri-diagonal matrix. Obviously, the rates in and out of the dummy states should be set so that their limiting probabilities would be zero.

To use Carvalho's method for the current problem, we let $\Omega(i) = \{(i,j) | j = 0, 1, \ldots, N - \max(0, i - S)\}$. That is, $\Omega(i)$ is the collection of states such that there are $i$ parts on order. We refer to this collection of states as "level $i$". The maximum number of states in the levels is $N + 1$. Therefore, to obtain a block tri-diagonal transition matrix, we add dummy states to the levels with less than $N + 1$ states to bring the number of states up to $N + 1$. Specifically, for level $S + 1 \leq i \leq N + S$, we add $i - S$ dummy states. The rates out of the dummy states are set, so that a transition is possible from $\Omega(i)$ to $\Omega(i - 1)$ at a rate of $ir$, and from $\Omega(i)$ to $\Omega(i + 1)$ at a rate of $\lambda_{N-i}$. Since no transitions are made to a dummy state due to repairs, the dummy states will act as transient states.

Let $L$ denote the submatrix from the infinitesimal generator that contains the failure rates that result in transitions from $\Omega(i)$ to $\Omega(i + 1)$, and let $M_i$ be the diagonal submatrix that contains the transition rates within $\Omega(i)$. The following set of vector equations can be derived:

$$\pi_{\Omega(i)} = -\pi_{\Omega(i)} M_0$$
$$\pi_{\Omega(i+1)}(i + 1) = -\pi_{\Omega(i)} M_i - \pi_{\Omega(i-1)} L; \text{ for } i = 1, \ldots, N + S - 1$$

and
\[ \pi_{\Omega(N+S)}(N+S) = -\pi_{\Omega(N+S-1)} \bar{L}, \]

where \( \pi_{\Omega(0)} \) is a vector that contains the steady-state probabilities of the states in the \( i-th \) level. Using the above vector equations, we can express \( \pi_{\Omega(i)} \) in terms of \( \pi_{\Omega(0)} \), so that

\[ \pi_{\Omega(i)} = -\pi_{\Omega(0)} K_i ; \quad i = 0, 1, \ldots, N + S \]  \[5.12\]

where \( K_i \) is a matrix with dimensions \((N + 1) \times (N + 1)\). To determine \( K_i \), we can use the recursive scheme

\[ K_{i+1} = \frac{1}{i+1} [K_{i-1} L + K_i M_j] ; \quad i = 0, 1, \ldots, N + S - 1, \]  \[5.13\]

where \( K_0 = I \) and \( K_1 = 0 \).

To determine the probabilities \( \pi_{\Omega(0)} \), we need to invert the matrix \( K_{N+S} \), which usually does not contain more than one zero element. As a result, solutions to problems with more than \( N + S = 60 \) was difficult when we used this method. For a detailed analysis of Carvalho's method, the reader is referred to [17].

### 5.2.4 Uniformization

Uniformization, which is also known as randomization or Jensen's method, is usually used to compute the time dependent probabilities of a continuous-time Markov chain. The main idea behind this method is to introduce self transitions into each state of the system, so that the amount of time spent in each state is the same for all states. For a more formal statement of uniformization, we follow the treatment of Daigle [25]. For the history of its development and a technical description of its computational properties, the reader is referred to Grassman [37].
Let $P(t)$ denote the time dependent transition probabilities of a continuous time Markov chain, and define the matrix $Q$ to be its infinitesimal generator. $P(t)$ can be computed by solving the system of linear differential equations:

$$\frac{d}{dt} P(t) = P(t)Q,$$  \hspace{1cm} [5.14]

which has the general solution

$$P(t) = P(0)e^{Qt},$$  \hspace{1cm} [5.15]

where $P(0)$ denotes the vector of initial state probabilities.

Suppose the time spent in each state is exponentially distributed with parameter $\nu$. Then, [5.15] can be rewritten as

$$P(t) = P(0)e^{-\nu t+\nu t+Q0},$$  \hspace{1cm} [5.16]

where $I$ is the identity matrix. With simple manipulations, we obtain an alternative expression for [5.16], which is

$$P(t) = P(0)e^{-\nu t}e^{\nu(I+\frac{1}{\nu}Q)t}.$$  \hspace{1cm} [5.17]

Using the McLaurin series expansion and regrouping, we obtain

$$P(t) = P(0)\sum_{n=0}^{\infty} \frac{(\nu t)^n}{n!} e^{-\nu t} \left[ I + \frac{1}{\nu}Q \right]^n.$$  \hspace{1cm} [5.18]

By having $\nu$ be at least as large as the magnitude of the maximal term on the diagonal of $Q$, [5.18] may be viewed as describing the dynamics of a discrete time Markov chain with one-step transition probability matrix $\left[ I + \frac{1}{\nu}Q \right]$. The transition epochs of the chain will be generated according to a Poisson process with rate $\nu$.
With our limited experience in numerical analysis, we found that uniformization in conjunction with the Gauss-Seidel method, has produced satisfactory results. For an in-depth treatment of uniformization, the reader should refer to [25], [37] and [74].

5.3 Monotonicity Properties

As we mentioned earlier, the total expected system cost, TC, and the availability constraint, are both functions of N, M and S. If TC and the availability constraint are monotonically nondecreasing functions of N, M and S, then the optimal solution (M',N',S'), which minimizes TC, will lie on the boundary of the feasible solution space. Monotonicity properties with respect to N and M have already been established, as shown in Appendix B. What remains to be proven, is that S exhibits the same monotonicity properties as N and M.

The idea is to show mathematically, that as we increase S, the expected number of machines operational would tend to increase, or at least never decreases. Let $E[\tilde{n}(S)]$ denote the expected number of busy repair stations given that the stocking level of the parts is S. For a fixed N and M, from [3.35] and [4.30] we can easily show that

$$E[\tilde{n}(S = 0)] < E[\tilde{n}(S = \infty)].$$  \[5.19\]

Using Little’s result and [5.19], we know that $\lambda(S = 0) < \lambda(S = \infty)$. Therefore,

$$E[\tilde{n}(S = 0)] < E[\tilde{n}(S = \infty)].$$  \[5.20\]
The relationship in [5.20] can be easily proven since we have the needed closed-form expressions for the cases $S = 0$ and $S = \infty$. As discussed earlier, however, for $0 < S < \infty$, the computation of the aggregate failure rate $\lambda$ is intractable. As a result, we cannot mathematically prove [5.20] for the general case, and instead make the following conjecture.

**Conjecture 5.1:** If $S_1 < S_2$, then $E[\tilde{\tau}(S = S_1)] \leq E[\tilde{\tau}(S = S_2)]$.

A very limited example is now presented that supports the above conjecture. Suppose, $N, M$ and $D$ are all equal to one. From [4.30], we find that

$$E[\tilde{\tau}(S = 0)] = \frac{1}{\frac{1}{\lambda} + \frac{1}{\tau} + \frac{1}{\mu}},$$

[5.21]

and from [3.35], we get

$$E[\tilde{\tau}(S = \infty)] = \frac{1}{\frac{1}{\lambda} + \frac{1}{\mu}},$$

[5.22]

for some fixed and positive $\lambda$, $\tau$ and $\mu$. Given the small size of the problem, it is possible to compute $E[\tilde{\tau}]$ when $S = 1$, which turns out to be

$$E[\tilde{\tau}(S = 1)] = \frac{1}{\frac{1}{\lambda} + \frac{1}{\mu} + \left(\frac{\lambda}{\lambda + \tau}\right)\left(\frac{\mu}{\tau + \mu}\right)\frac{1}{2\tau}}.$$  

[5.23]

From the denominator of [5.23], we know that $0 < \left(\frac{\lambda}{\lambda + \tau}\right)\left(\frac{\mu}{\tau + \mu}\right) < 1$ and $0 < \frac{1}{2\tau} < \frac{1}{\tau}$. Hence, the denominator of [5.21] is greater than the denominator of [5.23], which in turn is greater than the denominator of [5.22]. Thus,

$$E[\tilde{\tau}(S = 0)] < E[\tilde{\tau}(S = 1)] < E[\tilde{\tau}(S = \infty)].$$
To examine whether or not the conjecture holds in general, we ran numerous examples using uniformization in conjunction with the Gauss-Seidel method. The stopping criterion used for the iterative method is \( \sum |\pi^{(n)}_{i} - \pi^{(n-1)}_{i}| \leq 10^{-7} \), where \( \pi^{(n)}_{i} \) is the limiting probability of state \( i \) obtained in the \( n \)-th iterate. All the examples satisfied Conjecture 5.1, and some of them are reported in Figure 10, where \( N = 20, M = 5, D = 20, \lambda = 1, \mu = 3 \) and \( \tau \) was varied from 1 to 10.

![Graph showing aggregate failure rate as a function of stocking level](image)

**Figure 10. Aggregate Failure Rate as a Function of the Stocking Level.**

As Figure 10 shows, \( \lambda_{f} \) approaches a limit \( \lambda(S = \infty) \) which is set by the unlimited stocking case. The case \( S = \infty \) behaves as if there were no ordering subsystem, and obviously, no matter how many parts are kept in stock, the aggregate failure rate will not go beyond \( \lambda(S = \infty) \). Similarly, as \( \tau \) increases, \( \lambda_{f} \) approaches \( \lambda(S = \infty) \), where the case \( \tau = \infty \) would be analogous to the case of unlimited stock of parts.

For reasons that will become apparent later, we like to compare the marginal increase in availability when the stocking level is increased by one to the marginal increase in availability when the

5. The Spares Provisioning Problem with Parts Inventory
number of machines in the system is increased by one. Let \( E[\tilde{n}_d(N,S)] \) denote the expected number of machines operational when the system owns \( N \) machines and the stocking level of the parts is \( S \). The following, then, is our second conjecture.

**Conjecture 5.2:** For a fixed \( N, M, S \) and \( D \), \( E[\tilde{n}_d(N,S+1)] < E[\tilde{n}_d(N+1,S)] \).

The idea behind Conjecture 5.2, is that a part in stock will not act as a complete substitute for a machine. Using the special case \( N = M = 1 \), we know for instance that

\[
E[\tilde{n}(N = 2, S = 0)] - E[\tilde{n}(N = 1, S = 1)] = 2 \left( \frac{\tau + \mu}{\mu} \right) - \frac{\lambda}{\lambda + \tau} > 0.
\]

Hence, \( E[\tilde{n}_d(N = 1, S = 1)] < E[\tilde{n}_d(N = 2, S = 0)] \).

To support Conjecture 5.2, numerous numerical examples were run under general conditions, and all the results were as predicted. Figures 11 and 12 show some of the numerical results obtained. The examples have the same parameters as those previously used; that is, \( N = 20, M = 5, D = 20, \lambda = 1, \mu = 3, \) and \( \tau = 5 \) and 2 for Figures 11 and 12 respectively.

We note in passing that as \( \tau \) increases, adding one more part in stock will have a diminishing effect on availability since orders of parts outstanding will be filled at a faster rate. Finally, it is interesting to note that as \( S \) approaches infinity, the aggregate failure rate approaches \( \lambda_f(S = \infty) \) as discussed above, where

\[
\lambda_f(S = \infty) = \min(\lambda E[\tilde{n}_d], \mu E[\tilde{n}_s]). \quad [5.24]
\]

Since, in general, \( E[\tilde{n}_d] < D \) and \( E[\tilde{n}_s] < M \), this suggests that we can increase the aggregate failure rate beyond \( \lambda_f(S = \infty) \), by increasing the population size of the machines. That is, as \( N \) approaches infinity,

\[
\lambda_f(N = \infty) = \min(\lambda D, \mu M). \quad [5.25]
\]

5. The Spares Provisioning Problem with Parts Inventory
Figure 11. Increase in Availability with Respect to N and S when Ordering Rate is 5

Figure 12. Increase in Availability with Respect to N and S when Ordering Rate is 2

5. The Spares Provisioning Problem with Parts Inventory
Clearly, from [5.24] and [5.25], $\lambda(S = \infty) < \lambda(N = \infty)$.

5.4 Solution Methodology

If there is a nonzero cost associated with stocking parts, then the model in Chapter 3 with $S = \infty$ is, obviously, not economically feasible. Furthermore, assuming a consumable part is much cheaper than a repairable machine, the model in Chapter 4 with $S = 0$ may rarely yield the most economical solution, because more costly machines would have to be purchased to meet the service-level requirements. If we let $(M, N^0)$ and $(M, N^\infty)$ denote the feasible solution points obtained when $S = 0$ and $S = \infty$, respectively, given the number of repair stations is $M$, then the optimal solution point would have to lie in between the two points, and is obtained only if $0 < S < \infty$.

From the above discussion, we see that the boundary points for the cases $S = 0$ and $S = \infty$, when our service-level constraint is $n^*_f$, bound the optimal solution $(M^*, N^*, S^*)$. Thus, we may generate the boundary set, $\{(m, N(m), S(m))\}$, by first determining the feasible values for $m$, next determining $N^\infty(m)$ and $N^0(m)$, and then finally for each $n \in (N^\infty(m), N^0(m))$, we determine $S(m)$. All of the above is, of course, subject to the availability constraint $n^*_f$.

If machines are considerably more costly than parts, we need not vary $n$ between $N^\infty(m)$ and $N^0(m)$ since the optimal value will be $N^\infty(m)$. In this case, we may limit our computations to finding the set of points $\{(m, N^\infty(m), S(m))\}$. In effect, given $(m, N^\infty(m))$, we find the minimum stocking level of parts, $S(m)$, so that the system behaves as if there were an unlimited stock of parts.
It may, at first glance, seem impossible to meet the availability requirement with finite \( S \), since the probability of stockout of parts will then always be strictly greater than zero. But, the stockout probability is small enough so that the expected number of machines waiting for parts will always be a small fraction of a machine. Since the number of machines must be an integer, we can always find a finite \( S(m) \) that meets \( n_1^* \) with \( N^\infty(m) \) machines. In addition, based on Conjecture 5.1, the aggregate failure rate for the point \((m, N^\infty(m), S(m))\) will be strictly less than the rate for \((m, N^\infty(m), \infty)\), but will be at least \( \lambda_1^* = \lambda n_1^* \). We now turn to our solution procedure for \( S(m) \).

### 5.4.1 Direct Solution Method

Since we know that it is possible to meet the service-level constraint with \( N^\infty(m) \) machines and \( m \) repair stations, our approach is to produce an increasing sequence of stocking levels \( \{S_0(m), S_1(m), \ldots\} \) which produces an increasing sequence of aggregate failure rates \( \{\lambda_0(S_0(m)), \lambda_1(S_1(m)), \ldots\} \), stopping at the minimum value of \( S \) that yields an aggregate failure rate of at least \( \lambda_1^* \). This approach depends on the conjecture made earlier, that availability is a monotonically nondecreasing function of \( S \). If our conjecture is true, then we can easily use the iterative scheme developed in Chapter 3 to generate the minimum \( S, S(m) \), for each \( m \in \hat{M} \), where as before, \( \hat{M} \) denotes the set of feasible boundary values for \( M^* \).

With \( M = m \), suppose that at the \( i - th \) iteration the iterative scheme has yielded the solution point \( (m, N^\infty(m), S_i(m)) \). To compute \( \lambda_{i+1} \) for that point, we need to use one of the numerical methods such as those discussed in section 5.2. While it is possible to obtain the optimal solution for problems of moderate size (e.g., 100 machines), it is very difficult solving problems with more than 150 machines. Furthermore, there is always the possibility of not being able to converge to the desired residual error, even for problems of small size. Thus, there is a need to limit the number of iterations required to obtain \( S(m) \).

5. The Spares Provisioning Problem with Parts Inventory
Given the disadvantages of numerical methods, one way to limit the number of iterations required is to use simple approximate models that provide reasonably tight lower and upper bounds for each point \((m,N^\infty(m))\). Using these bounds we should be able to converge to the point \((m,N^\infty(m), S(m))\) much faster than with the direct method discussed above.

### 5.4.2 A Lower Bound

We begin our discussion by pointing out that for each \(m \in \hat{M}\), \(N^0(m) \geq N^\infty(m)\). This is due to the fact, that when \(S = 0\), machines will be held in the ordering subsystem for a finite amount of time every time they fail. Intuitively, \(E[\hat{N}(m)]\), the expected number of machines waiting for parts in the ordering subsystem when \(S = 0\), is upper bounded by \(N^0(m) - N^\infty(m)\), if \(N^0(m) > N^\infty(m)\).

Based on Conjecture 5.2, which says that if we replace \(N^0(m) - N^\infty(m)\) machines with \(N^0(m) - N^\infty(m)\) parts, then availability of the system will decrease. We conjecture that the stocking level of the parts must be

\[
S(m) \geq N^0(m) - N^\infty(m); \quad m \in \hat{M}.
\]  \hspace{1cm} \[5.26\]

That is, a lower bound for the stocking level is

\[
S_0(m) = N^0(m) - N^\infty(m); \quad m \in \hat{M}.
\]  \hspace{1cm} \[5.27\]

Note that [5.27] can be assigned to the initial value \(S_0(m)\) for the direct solution method, if the iterative scheme in Chapter 3, or any other iterative procedure is to be implemented.

---

5. The Spares Provisioning Problem with Parts Inventory
5.4.3 Probability of Stockout

To obtain an upper bound on $S(m)$, an approximate model has been developed for the finite stocking case. The model depends on the probability of stockout of parts given $(m, N^w(m), S(m))$. Therefore, we direct our attention to the computation of the stockout probability first.

Define $p'$ as the stockout probability of parts when the system owns $N^w(m)$ machines, $m$ repair stations, and the stocking level is $S(m)$. What makes the computation of $p'$ possible, is the one-to-one relationship that exists between the stocking level of the parts and the ordering rate. That is, suppose a system has the parameters $N, M, D, \lambda, \mu, S$ and $\tau$, and produces an aggregate failure rate $\lambda_p$. Then, there exists a $S' (\neq S)$ and $\tau' (\neq \tau)$, such that a system with parameters $N, M, D, \lambda, \mu, S'$ and $\tau'$ will produce the same aggregate failure rate $\lambda_p$. Furthermore, given $S'$, $\tau'$ is unique and vice versa. In other words, as Figure 13 shows, given $\tau$, there is only one $S$ that achieves a certain $\lambda_p$; and vice versa, given $S$, there is only one $\tau$ that achieves the same $\lambda_p$.

![Figure 13. One-to-One Relationship Between Stocking Level and Ordering Rate](image)

5. The Spares Provisioning Problem with Parts Inventory
Given an ordering rate of \( \tau \) and a constraint \( \lambda_* \), we know that there exists a minimum stocking level \( S(m) < \infty \), such that the point \( (m, N^{\infty}(m), S(m)) \) will yield an aggregate failure rate of \( \lambda_f \geq \lambda_* \).

From the above discussion then, we can set the stocking level at zero and find the corresponding ordering rate \( \tau_\infty \), such that the point \( (m, N^{\infty}(m), 0) \) yields an aggregate failure rate of \( \lambda_n \) where \( \tau_\infty \) is finite and it is the minimum ordering rate that makes the system behaves as if there were an unlimited stock of parts.

The system with a stocking level of \( S(m) \) is equivalent to the system with zero stocking in the sense that both systems yield the same aggregate failure rate \( \lambda_n \) and behave as though there were an unlimited stock of parts. If we define \( E[\tilde{n}_0]_\infty \) to be the expected number of machines waiting for parts when \( S = 0 \) and the ordering rate is \( \tau_\infty \), then from Little’s result we know that

\[
\lambda_f = \tau E[\tilde{n}_0] = \tau_\infty E[\tilde{n}_0]_\infty. \tag{5.28}
\]

As illustrated in Figure 13, we also know that \( \tau_\infty \) is larger than \( \tau \) in order to compensate for the decrease in the stocking level; therefore, \( E[\tilde{n}_0]_\infty < E[\tilde{n}_0] \). While \( E[\tilde{n}_0]_\infty \) is the expected number of machines waiting for parts, \( E[\tilde{n}_0] \) is the expected number of orders of parts outstanding. If \( E[\tilde{n}_{0,m}] \) denotes the expected number of machines waiting for parts when the stocking level is \( S(m) \), and the ordering rate is \( \tau \), then from Little’s result,

\[
E[\tilde{n}_{0,m}] = p \cdot \lambda_f \frac{1}{\tau}. \tag{5.29}
\]

Also, from Little’s result and [5.28], we get

\[
\lambda_f = \frac{E[\tilde{n}_{0,m}]\tau}{p} = \tau_\infty E[\tilde{n}_0]_\infty,
\]

so that

\[
p \cdot \frac{E[\tilde{n}_{0,m}]\tau}{E[\tilde{n}_0]_\infty \tau_\infty}. \tag{5.30}
\]

5. The Spares Provisioning Problem with Parts Inventory

117
While the system with ordering rate $\tau$ and $(m,N^\infty(m),S(m))$ yields the same $E[\tilde{n}_d]$ and $E[\tilde{n}_r]$ as the system with rate $\tau_\infty$ and $(m,N^\infty(m),0)$, the expected queue lengths in front of the repair subsystem and the standby machines may not be equal for both systems. However, since the average arrival rate to the queues is identical in both systems, we would expect the queue lengths to be approximately equal. As a result, $E[\tilde{n}_d]_\infty$ is only an approximation of $E[\tilde{n}_d]_\infty$. Therefore, the proportion of machines that wait for parts when the stocking level is $S(m)$, is

$$p^* \approx \frac{\tau}{\tau_\infty}. \quad \text{(5.31)}$$

The remaining question is, how do we obtain $\tau_\infty$ to compute $p^*$? Fortunately, $\tau_\infty$ is based on the stocking policy $S = 0$, and all the analysis for this policy is discussed in Chapter 4. Based on Conjecture 5.1 and the one-to-one relationship between the ordering rate and the stocking level, we believe that the ordering rate exhibits the same monotonicity properties as the stocking level. As a result, we can use any line search method, or the iterative scheme in Chapter 3, to find $\tau_\infty$, which is the minimum ordering rate that yields an aggregate failure rate of at least $\lambda^*_r$.

Finally, suppose we have a model where upon failure, there is a state independent probability $p^*$ a machine will wait in the ordering subsystem, and with probability $1 - p^*$ it proceeds immediately to the repair subsystem. With $N^\infty(m)$ machines and $m$ repair stations, we conjecture that the model will yield approximately the same aggregate failure rate as the model with the state dependent probability and stocking level $S(m)$ or ordering rate $\tau_\infty$. That is, we conjecture that the average number of machines in the subsystems and queues is relatively insensitive to the distribution of the interarrival process in the context of this particular class of models.

### 5.4.4 An Upper Bound

Even though we can approximate the stationary probability that a machine will wait for a part upon failure for the case $(m,N^\infty(m),S(m))$, we still do not have a computational formula for $S(m)$.
We will now propose an approximate model that will provide an upper bound for \( S(m) \). Using the upper bound and the lower bound from [5.27], we aim to obtain at least a reasonably good approximation for \( S(m) \) with very little computation.

The idea behind the approximate model is as follows. Suppose we ignore the repair subsystem completely, and assume that we have only the production floor and the ordering subsystem. The Markov process that describes our new system is now a simple one-dimensional birth-death process similar to that of the unlimited stocking case. If state \( i \) denotes the number of orders of parts outstanding, then with \( N \) machines and \( S \) parts in stock, \( i \in \{0, 1, \ldots, N + S\} \). Since our objective is to find the minimum \( S \) such that the aggregate failure rate of the system is at least \( \lambda'_i = \lambda n'_i \), we could set the failure rate per machine to be \( \lambda \) as in the real model. Then the rate at which we leave state \( i \) due to failures is \( \lambda_i \), where

\[
\lambda_i = \begin{cases} 
D\lambda, & i = 0, 1, \ldots, N - D + S \\
i\lambda, & i = N - D + S + 1, \ldots, N + S - 1
\end{cases}
\]  

[5.32]

and the rate at which we leave state \( i \) due to the arrival of parts is \( \tau_i \), where

\[
\tau_i = i\tau; \quad i = 0, 1, \ldots, N + S.
\]  

[5.33]

If \( S \) parts are kept in stock, then the probability of stockout of parts is the probability that at least \( S + 1 \) orders of parts are outstanding. To compute this probability, we use the outlined approach in Chapter 3. That is, we determine \( E[\tilde{\tau}_{1,S+1}] \), the expected number of visits to state \( S + 1 \) from state 1 during the first passage time from state 1 to state 0. Based on Wald's equation, we then multiply \( E[\tilde{\tau}_{1,S+1}] \) by \( E[\tilde{C}_{S+1}] \), which is the expected first passage time from state \( S + 1 \) to state \( S \), to obtain \( E[\tilde{X}_{S+1}] \), the expected aggregate amount of time spent in states \( S + 1, \ldots, S + N \) during the first passage time from state 1 to state 0. Thus,

\[
P\{\text{stockout of parts} \mid \text{the stocking level is } S\} = \frac{E[\tilde{X}_{S+1}]}{E[C]}.
\]  

[5.34]
where $E[\tilde{C}]$ is the expected cycle length and is equal to $\frac{1}{D^2} + E[\tilde{N}_b]$. 

We claim that the following procedure provides an upper bound for $S(m)$. 

\textit{Step 0:} Set $S = S(m)$ from [5.27].

\textit{Step 1:} Given $N^*(m)$ and $S$, compute the stockout probability from [5.34]. If the probability is at most $p'$; stop, the upper bound for $S(m)$ is $S_*(m) = S$. Otherwise, go to Step 2.

\textit{Step 2:} Increment $S$ by one and go to Step 1.

Intuitively, we believe $S_*(m)$ is an upper bound, simply because we have ignored the repair subsystem. That is, when machines fail, they will immediately become operational if a part is in stock; otherwise, they will wait for a part in the ordering subsystem. If a machine is waiting in the ordering subsystem, it will become operational once a part is available. As a result, the rate at which machines enter the ordering subsystem is faster than the true rate when a repair subsystem exists. Therefore, the stocking level of the parts needed to yield the stockout probability of not more than $p'$ in the approximate model, is at least $S(m)$.

While the above procedure has been found to yield upper bounds in all the examples considered, the upper bounds are too loose to be of practical value. We found empirically that better upper bounds are produced if $\lambda$ in the above approximate model is replaced by

$$\lambda' = \frac{\lambda^*}{D}.$$  \hspace{1cm} [5.35]

We have used [5.35] in all the examples considered here, and as reported in Table 5, excellent results were produced.
5.4.5 Indirect Solution Method

Our indirect solution method is basically a summary of all the results and arguments made earlier. That is, given $D$, $\lambda$, $\tau$, $\mu$, and $n^*_u$, the idea is first to generate the boundary points for the two extreme stocking policies: $(m, N^\infty(m))$ and $(m, N^\mu(m))$, and obtain the lower bounds for $S(m)$. For each point $(m, N^\infty(m))$, we then find $\tau^*$ to compute the stockout probability, from which we obtain the upper bound for $S(m)$. Once the bounds $S_L(m)$ and $S_u(m)$ are available, any line search method can be used to converge to $S(m)$. By repeating the process for every $m \in \mathcal{M}$, we will generate the boundary points $\{(m, N^\infty(m), S(m))\}$, at least one of which will be the optimal solution.

There are numerous line search methods in existence, and they are fairly easy to implement. One possible method for our problem is to use simple linear interpolation, since we are seeking a point that is bounded from below and above. That is, given our initial lower bound $S_L(m)$, and upper bound $S_u(m)$, we compute the respective aggregate failure rates $\lambda_L$ and $\lambda_u$. Assuming $\lambda_L$ does not satisfy the constraint $\lambda^*$, we find a new stocking level to be

$$S_g(m) = \left[ S_L(m) + (S_u(m) - S_L(m)) \frac{\lambda_g^* - \lambda_L}{\lambda_u - \lambda_L} \right]. \quad \text{[5.36]}$$

$S$ is an integer, and if the interpolation yields any fractions, then we should round $S_g(m)$ down. As Figure 14 shows, if we believe that the aggregate failure rate is a monotonically nondecreasing function of $S$ that asymptotically approaches $\lambda(S = \infty)$, then $S(m)$ lies to the left of $S_g(m)$, or $S(m) \leq S_g(m)$. After computing $S_g(m)$, we can tighten the bounds on $S(m)$ by setting the upper bound at $S_g(m)$, and incrementing the lower bound by one, if the aggregate rate at $S_g(m)$ is greater than $\lambda^*$. However, it is computationally more efficient to retain the original lower bound, since it involves the numerical evaluation of an additional point that might be costly in terms of execution time. Finally, it is possible that $S_g(m)$ will be $S(m) - 1$ as a result of rounding down. For example,
[5.36] may yield 3.5. We know this point is greater than that required to produce \( \lambda^*_g \), which may be 3.3, for example. When we round down we find \( S_g(m) = 3 \). But, of course, we must have at least 3.3 so, \( S(m) \) is actually 4. When the value \( S_g(m) = 3 \) is tested, we find \( \lambda_f < \lambda^*_g \) and we know that the above possibility has occurred and thus, \( S(m) = 4 \) without further computation.

![Figure 14. The Linear Interpolation Method](image)

A more formal statement of the indirect solution method is now provided. Given \( D, \lambda, \tau, \mu \) and \( r^*_g \), generate the boundary points \( (m, N^0(m)) \) and \( (m, N^\infty(m)) \) based on the iterative procedure in Chapter 3. For \( m = M_b \) to \( N_d(M_b) \) do the following:

**Step 1:** Obtain the lower bound \( S(m) \) from [5.27].

**Step 2:** Based on \( (m, N^\infty(m), S(m)) \), compute the aggregate failure rate \( \lambda_f \). If \( \lambda_f \geq \lambda^*_g \), then set \( S(m) = S(m) \) and stop; the feasible boundary point \( (m, N^\infty(m), S(m)) \) has been reached. Otherwise, go to Step 3.

**Step 3:** Based on \( (m, N^\infty(m), 0) \), find \( \tau_{\infty} \), the minimum ordering rate that yields an aggregate failure rate of at least \( \lambda^*_g \), and compute the stockout probability \( p^* \) from [5.31].
Step 4: Based on $p^*$, use the approximate model and the upper bound procedure to obtain $S_*(m)$. If $S_*(m) = S_*(m) + 1$, stop, $S(m) = S_*(m)$. Otherwise, go to Step 5.

Step 5: Based on $(m,N^\infty(m),S_*(m))$, compute the aggregate failure rate $\lambda^*_s$.

Step 6: Based on $(S_*(m), \lambda^*_p)$ and $(S_*(m), \lambda^*_s)$, compute $S'_*(m)$ from [5.36].

Step 7: Based on $(m,N^\infty(m), S'_*(m))$, compute the aggregate failure rate $\lambda^*_s$. If $\lambda^*_s \leq \lambda^*_s$, then set $S(m) = S'_*(m) + 1$, and stop. If $S'(m) = S'_*(m) + 1$, then set $S(m) = S'_*(m)$ and stop. Otherwise, go to Step 8.

Step 8: Set $S'_*(m) = S'_*(m)$, and $\lambda^*_s = \lambda^*_s$. Go to Step 6.

5.5 Numerical Examples

In this section, we demonstrate the efficiency of the indirect solution method through the presentation of a detailed numerical example. Numerous additional examples, some of which are reported here, were tested and support the efficiency of the solution method. We note that performance evaluations were performed using uniformization in conjunction with the Gauss-Seidel method. The stopping criterion used for the numerical method was if the residual error (the total difference between the steady-state probabilities from two consecutive iterations) is less than 0.000001. We also note that to find $\tau^\infty$ in Step 3 of the algorithm, we used an iterative procedure similar to that of Chapter 3.

For our illustrative example, we choose the system parameters $\lambda = 1$, $\tau = 2$, $\mu = 3$, $D = 10$ and $\eta^*_s = 8.5$. Using the iterative procedure from Chapter 3, we obtain the boundary points $\{(m,N^\infty(m))\} = \{(3,17),(4,12)\}$ for the unlimited stocking case, and $\{(m,N^\infty(m))\} =$
{(3,22),(4,17),(5,17),(6,17),(7,16)} for the zero stocking case. Based on the two extreme policies, we know that the optimal solution \((M^*, N^*, S^*)\), lies in the region that is defined by the two sets of points \{((3, 17, \infty), (4, 12, \infty))\} and \{((3, 22, 0), (4, 17, 0))\}. Therefore, starting with \(M = 3\) and \(N^\infty(3) = 17\), we obtain the lower bound \(S(3) = 22 - 17 = 5\). The point \((3, 17, 5)\) yields an aggregate failure rate of \(\lambda_f = 8.478 < \lambda_f^* = 8.5\), and we need to find the upper bound \(S_u(3)\) for \(S(3)\).

The modified ordering rate for the point \((3, 17, 0)\) is \(\tau^\infty = 35.823\). Hence, the probability of stockout of parts is \(\rho^* = \frac{2}{35.823} = 0.056\). Based on \(\rho^*\) and the upper bound procedure, we determine the upper bound to be \(S_u(3) = 8\), so that, the point \((3, 17, 8)\) produces an aggregate failure rate \(\lambda_f = 8.518\). From [5.36], we get a new upper bound

\[
S_u(3) = \left[ S_l(3) + (S_u(3) - S_l(3)) \frac{8.5 - \lambda_f}{\lambda_f - \lambda_f^*} \right] \\
= \left[ 5 + \frac{0.022}{0.04} \right] = 6.
\]

Since \(S_u(3) = S_l(3) + 1\), then \(S(3)\) must be 6, and we have generated our first boundary point with a nonzero finite stock of parts; \((3, 17, 6)\). Incidentally, the aggregate failure rate for the solution point \((3, 17, 6)\) is 8.503.

To obtain the second boundary point, we start again with \(M = 4\), and \(N^\infty(4) = 12\). The lower bound for the new point is \(S_l(4) = 5\), and \(\lambda_f\) for \((4, 12, 5)\) is 8.309, which makes it necessary to obtain an upper bound. With \(\tau^\infty = 164.718\), we get \(\rho^* = 0.012\), which produces an upper bound \(S_u(4) = 9\). By interpolating, we obtain a new upper bound \(S_u(4) = S_u(4) = 8\). Going through another iteration, we determine the second boundary point, which is \((4, 12, 8)\), and the procedure is terminated. Based on our cost function then, we either choose \((3, 17, 6)\) or \((4, 12, 8)\) as our optimal solution.

To see how the indirect method compares with the direct method, suppose we let the initial value \(S_0(m) = N^0(m) - N^\infty(m)\), as in [5.27]. Starting with \((m, N^\infty(m), S_u(m)) = (3, 17, 5)\), we iterate.
once to get to the boundary point (3,17,6). However, if \( M = 4 \) and \( N_w(4) = 12 \), three iterations will be required to generate the sequence \( \{S_0(4), S_1(4), S_2(4), S_3(4)\} = \{5, 6, 7, 8\} \) which produces a sequence of aggregate failure rates \( \{\lambda_{K_0}, \lambda_{K_1}, \lambda_{K_2}, \lambda_{K_3}\} = \{8.478, 8.481, 8.487, 8.513\} \), stopping at (4,12,8).

Table 5, which contains the boundary points for all stocking policies, shows some of the cases that were tested using both the direct and indirect solution methods. The set of examples considered here is the same set as Table 3 in Chapter 4, where \( \rho_1 = \frac{\lambda}{\tau} \) and \( \rho_2 = \frac{\lambda}{\mu} \). As before, the utilization ratios were varied from 10% to 100%, and in general, the indirect method performed better than the direct method, especially when the utilization ratios approached one. A possible explanation for the better performance is that as the stocking level increases, the aggregate failure rate function approaches \( \lambda(S = \infty) \). The rate of change of the function approaches zero as the function value gets closer to the limit. Therefore, for large values of \( S \), the derivative of the failure rate curve is essentially constant, which makes it ideal for linear interpolation.

The disadvantage of the indirect method is that it approaches the boundary point from above, and upper bounds require bigger state spaces, which is critical when dealing with problems of large sizes. However, the bounds \( S_L(m) \) and \( S_u(m) \) can be obtained no matter how large the problem is, and we can always find a reasonably good approximate solution from the bounds without the use of numerical methods. For instance, the midpoint of \( S_L(m) \) and \( S_u(m) \).

5. The Spares Provisioning Problem with Parts Inventory
Table 5. Performance of the Algorithm with Parts Inventory

<table>
<thead>
<tr>
<th>$D$</th>
<th>$n^*_g$</th>
<th>$\rho_1$</th>
<th>$\rho_2$</th>
<th>Lower and Upper Bounds ($S(M), S_u(M)$)</th>
<th>Boundary Point ($M, N^\infty(M), S(M)$)</th>
<th>No. of Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>4</td>
<td>0.1</td>
<td>0.1</td>
<td>(0.0) (0.0) (1.5,0) (1.7,0)</td>
<td>0.0 (2.5,1) (3.7,0) (5.9,2)</td>
<td>0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2</td>
<td>0.2</td>
<td>(1.1) (2.0) (1.7,1) (2.5,2)</td>
<td>0.5 (3.7,0) (5.9,3) (6.9,0)</td>
<td>0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5</td>
<td>0.5</td>
<td>(1.1) (2.0) (1.7,1) (2.5,2)</td>
<td>0.5 (3.7,0) (5.9,3) (6.9,0)</td>
<td>0 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.0</td>
<td>1.0</td>
<td>(1.3) (2.3) (1.5,3) (3.7,0)</td>
<td>1.0 (3.6) (5.9,3) (6.9,0)</td>
<td>0 2</td>
</tr>
<tr>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>(2.3) (2.3) (1.7,3) (2.5,4)</td>
<td>0.5 (2.3) (3.7,3) (5.9,5)</td>
<td>1 2</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>(2.3) (3.4) (1.7,3) (2.5,4)</td>
<td>0.5 (3.5) (3.7,3) (5.9,5)</td>
<td>1 2</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>(5.10) (5.10) (5.9,9) (6.9,5)</td>
<td>0.1 (5.5) (5.5) (5.9,9)</td>
<td>0 1</td>
</tr>
<tr>
<td>20</td>
<td>17</td>
<td>0.1</td>
<td>0.1</td>
<td>(2.4) (2.4) (2.2,13) (3.19,4)</td>
<td>0.2 (2.2) (2.2,13) (3.19,4)</td>
<td>0 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.2</td>
<td>0.2</td>
<td>(2.2) (2.2) (2.2,13) (3.19,4)</td>
<td>0.5 (2.2) (2.2,13) (3.19,4)</td>
<td>0 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5</td>
<td>0.5</td>
<td>(2.2) (2.2) (2.2,13) (3.19,4)</td>
<td>1.0 (2.2) (2.2,13) (3.19,4)</td>
<td>0 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.0</td>
<td>1.0</td>
<td>(2.2) (2.2) (2.2,13) (3.19,4)</td>
<td>0.5 (2.2) (2.2,13) (3.19,4)</td>
<td>0 3</td>
</tr>
<tr>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>(4.7) (4.7) (2.2,16) (3.19,7)</td>
<td>0.5 (4.4) (4.4) (9.30,4)</td>
<td>0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.5</td>
<td>0.5</td>
<td>(4.7) (4.7) (2.2,16) (3.19,7)</td>
<td>0.5 (4.4) (4.4) (9.30,4)</td>
<td>0 0</td>
</tr>
<tr>
<td>5. The Spares Provisioning Problem with Parts Inventory 126</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D$</td>
<td>$\rho_1$</td>
<td>$\rho_2$</td>
<td>Lower and Upper Bounds $(S_1(M), \sigma_1(M))$</td>
<td>Boundary Point $(\mathcal{N}^\infty(M), S(M))$</td>
<td>No. of Iterations</td>
<td>Indirect Method</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>---------------------------------</td>
<td>--------------------------------------------</td>
<td>-----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>1.0</td>
<td>(3,5)</td>
<td>(18,38,4)</td>
<td>0</td>
<td>1</td>
<td>(4,4)</td>
<td>(19,36,4)</td>
</tr>
<tr>
<td></td>
<td>(3,5)</td>
<td>(22,35,4)</td>
<td>0</td>
<td>1</td>
<td>(4,5)</td>
<td>(21,35,5)</td>
</tr>
<tr>
<td>0.5</td>
<td>(10,16)</td>
<td>(2,21,13)</td>
<td>2</td>
<td>3</td>
<td>(9,14)</td>
<td>(3,19,14)</td>
</tr>
<tr>
<td>0.2</td>
<td>(9,13)</td>
<td>(4,23,10)</td>
<td>0</td>
<td>1</td>
<td>(9,15)</td>
<td>(5,21,12)</td>
</tr>
<tr>
<td></td>
<td>(9,14)</td>
<td>(10,27,11)</td>
<td>2</td>
<td>2</td>
<td>(9,14)</td>
<td>(10,27,11)</td>
</tr>
<tr>
<td></td>
<td>(9,19)</td>
<td>(11,26,16)</td>
<td>3</td>
<td>6</td>
<td>(9,19)</td>
<td>(11,26,16)</td>
</tr>
<tr>
<td>1.0</td>
<td>(9,12)</td>
<td>(19,36,10)</td>
<td>0</td>
<td>1</td>
<td>(9,13)</td>
<td>(19,36,10)</td>
</tr>
<tr>
<td>0.1</td>
<td>(19,28)</td>
<td>(2,21,23)</td>
<td>3</td>
<td>3</td>
<td>(18,26)</td>
<td>(3,19,26)</td>
</tr>
<tr>
<td>0.2</td>
<td>(18,24)</td>
<td>(4,23,10)</td>
<td>0</td>
<td>1</td>
<td>(18,27)</td>
<td>(5,21,22)</td>
</tr>
<tr>
<td></td>
<td>(18,24)</td>
<td>(10,27,21)</td>
<td>2</td>
<td>2</td>
<td>(18,26)</td>
<td>(11,26,28)</td>
</tr>
<tr>
<td>0.5</td>
<td>(18,25)</td>
<td>(19,36,20)</td>
<td>2</td>
<td>2</td>
<td>(18,25)</td>
<td>(20,35,22)</td>
</tr>
<tr>
<td>1.0</td>
<td>(17,25)</td>
<td>(22,35,20)</td>
<td>2</td>
<td>2</td>
<td>(17,25)</td>
<td>(22,35,20)</td>
</tr>
</tbody>
</table>

5. The Spares Provisioning Problem with Parts Inventory
6. Summary and Conclusions

In this research, we have analyzed the spares provisioning problem, where machines become inoperable due to the failure of a critical built-in part. Upon failure, a machine must obtain a replacement part from inventory before the machine can be repaired. If a part is in stock, the machine picks up the part and is immediately transferred to the repair facility; otherwise, the machine must wait in the ordering subsystem for the arrival of the needed part. The ordering subsystem brings a new aspect to the spares provisioning problem and dramatically increases its difficulty.

A maximum of $D$ machines can be deployed simultaneously in order to meet demand. Given a service-level constraint; that is, the expected number of machines operational must be at least some specified level $n^*$, our aim was to determine the number of machines ($N$) that the system must own, the number of repair stations ($M$), and the stocking level of the parts ($S$) to maintain in inventory, that satisfies the availability constraint and minimizes the total expected system cost. The cost function in our case, could include the cost of the machines, the repair stations and the inventory cost of the parts.
We have analyzed the above described problem under three policies. Starting with the assumption that there are unlimited parts in stock, we have reduced the spares provisioning problem to the classical machine repair problem. Based on Little's result, we then developed an algorithm that generates all the boundary points of the feasible solution space defined by the availability constraint. From the set of boundary points, we can easily select the optimal solution.

By extending the results from the unlimited stock case, and with the use of potentials from Markov chain theory, we then analyzed the spares provisioning problem with the other extreme policy of zero parts in stock. The approach was again to generate the boundary points which include the optimal solution. The Little's result-based algorithm for the zero and unlimited stocking cases proved to be extremely efficient in terms of computer storage and CPU time, and can handle problems of almost any size. The efficiency of the algorithm was due mainly to the characteristics of the problem. Specifically, the queueing network models that describe the zero and unlimited stocking policies are closed cyclic network models that possess product-form solutions.

Finally, we considered the case in which the stock of the parts is greater than zero and finite. This finite stocking policy creates special problems, because the queueing network model which describes the system is a non-product-form network. As a result, we cannot obtain a closed-form solution under general conditions, and we are forced to solve for the system numerically. By generating the boundary points for the zero and unlimited stocking cases, however, we developed an algorithm that bounds and converges to the optimal solution quickly in order to minimize the use of numerical methods. Even if getting an exact solution is prohibitive due to the size of the problem, we can always find a "reasonably" good approximate solution.

In summary, our main contributions to the area of spares provisioning problems are four fold. First, Little's result appears to provide extremely good initial guesses when searching for the optimal solution. The result is very easy to implement, applies under general conditions, and can reduce our search efforts considerably. It is quite surprising that this result is hardly used in optimizing spares provisioning problems. Second, the iterative schemes developed here seem to converge to
the optimal solutions fairly quickly, and can be applied to all problems that exhibit the same monotonicity properties. Third, the computation of the aggregate failure rate for the zero and unlimited stocking cases require almost negligible computer storage and CPU time, and can be performed for problems of almost any size. Finally, for the computationally intractable model of finite stocking, we can find bounds on the optimal solution that can be used to get an approximate solution for problems of almost any size.

We considered the spares provisioning problem under the following assumptions. A machine has one critical part that fails when it is operational, and the time to failure is exponentially distributed. When a machine is repaired, it is restored to its original condition, and repair times are exponentially distributed. The stocking policy used for the parts is the lot-for-lot or \((S - 1, S)\) inventory policy, and the times between the placement of orders and the arrival of parts is also exponentially distributed.

Relaxing any of the assumptions stated above leads to interesting and challenging research areas. For instance, it is worth investigating the behavior of the spares provisioning system when the times are not exponentially distributed, and to see if the algorithms developed here will still apply under general distributions. It seems that even if performance measures are sensitive to distributions, the algorithms might be useful in obtaining good initial solutions. It is also worth considering the system under batch ordering policies for the parts, instead of the lot-for-lot ordering policy. While the lot-for-lot policy is useful for parts that are expensive, this policy is a restricted case of the more general batch ordering policy.

A major extension to the spares provisioning problem would be to consider the case of multiple parts with different failure, ordering, and repair rates. The model, which has great practical applications, can be generalized further if there are different types of machines with a hierarchical structure of parts as illustrated in Figure 1. These types of models create special problems at the repair facility, if a repair station can handle repairs of different types, and priority disciplines for repairs may need to be implemented to increase the availability of the machines. For different types of
machine failures the following are some of the references that are helpful, if this area is to be pursued: [29]¹, [35], [49], [50], [51], [52], [53] and [54].

Finally, throughout this research we have implicitly assumed that all the cost coefficients of the total cost function are known constants. In addition, we assumed that the design parameters λ, τ and μ are fixed and cannot be controlled by the system manager. It seems interesting and worthwhile to investigate the spares provisioning problem if none of the just mentioned parameters were actually fixed. It is possible, for instance, for the system manager to be faced with a set of mutually exclusive design alternatives, where each alternative corresponds to a certain failure, repair, and ordering rate. Using the computationally efficient algorithms developed here, it is fairly easy to conduct as many design iterations as desired. This would facilitate the decision making process and make it easier to select the best design alternative. It is considerably more challenging, however, when the cost coefficients are random. In addition to operations research tools, sophisticated statistical methods, such as response surface methodology, may have to be used in order to reach the right decision.

¹ There is a modeling error in [29] which led to misleading results.
Appendix A: The Steady-State Probability

Distribution of the Spares Provisioning System

The spares provisioning problem is a special case of the more general problem with parts inventory; that is, we have an unlimited supply of consumable parts in inventory and in effect ignore the ordering subsystem completely. We will, therefore, present the steady-state probability distribution of the model given in Chapter 4, which implements the other extreme inventory policy of setting the stocking level of the parts at zero. Given the more general probability distribution, we can then easily obtain the distribution of the model presented in Chapter 3.

To identify uniquely the number of machines at each subsystem, we denote the state of the system by $\eta = (n,s,m)$, where $n$ represents the number of operational machines, $s$ is the number of machines in the ordering subsystem waiting for parts, and $m$ is the number of machines either being repaired or waiting for repair. Let $\lambda(\eta)$ denote the rate at which the system leaves the state $\eta$ due to a machine failure. Then,
\[ \lambda(\eta) = \begin{cases} n \lambda, & \text{for } n = 0, 1, 2, \ldots, D \\ D \lambda, & \text{for } n = D + 1, D + 2, \ldots, N \end{cases} \tag{A.1} \]

When a machine failure occurs, the number of operational machines decreases by one, and the number of orders outstanding of the parts increases by one according to the \((S - 1, S)\) inventory policy. Let \(\tau(\eta)\) denote the rate at which the system leaves the state \(\eta\) due to the arrival of a part to the ordering subsystem. Then

\[ \tau(\eta) = s \tau, \text{ for } s = 0, 1, \ldots, N. \tag{A.2} \]

Let \(\mu(\eta)\) denote the rate at which the system leaves the state \(\eta\) due to the completion of a repair. Then

\[ \mu(\eta) = \begin{cases} m \mu, & \text{for } m = 0, 1, \ldots, M - 1 \\ M \mu, & \text{for } m = M, M + 1, \ldots, N \end{cases} \tag{A.3} \]

The probability of being in state \(\eta\) at time \(t\), which is denoted by \(p(\eta, t)\), is governed by the forward Chapman-Kolmogorov equation which describes the queueing system. This can be written as follows:

\[
\frac{d}{dt} p(\eta, t) = - \left[ \lambda(\eta) + \tau(\eta) + \mu(\eta) \right] p(\eta, t) \\
+ \delta_1 \mu(n - 1, s, m + 1) p(n - 1, s, m + 1, t) \\
+ \delta_2 \lambda(n + 1, s - 1, m) p(n + 1, s - 1, m, t) \\
+ \delta_3 \tau(n, s + 1, m - 1) p(n, s + 1, m - 1, t) \\
\text{s.t. } n + s + m = N, \tag{A.4} \]

where \(\delta_i (i = 1, 2, 3)\) is the usual Kronecker delta; that is it takes on a value of zero if any of the components of the state vector is less than zero, otherwise, it is equal to one. Note, from now on, we will ignore the time variable \(t\), since as \(t\) tends to infinity, the steady-state solution is not dependent on time and the derivatives of \(p(\eta, t)\) with respect to \(t\) are equal to zero.

Appendix A: The Steady-State Probability Distribution of the Spares Provisioning System

133
The system of equations in [A.4] has a unique probability solution and can be obtained by induction, which is

\[ p(n,s,m) = \frac{D!D^{(N-D)}}{\beta(n)! \beta(m)} \left( \frac{\lambda}{s} \right)^m \left( \frac{\lambda}{\mu} \right)^m \mu^N \alpha \beta(N,0,0), \]  

[A.5]

where

\[ \beta(n) = \begin{cases} 
  n! & \text{if } n \leq D \\
  D!D^{(n-D)} & \text{if } n > D 
\end{cases} \]

and

\[ \beta(m) = \begin{cases} 
  m! & \text{if } m \leq M \\
  M!M^{(m-M)} & \text{if } m > M 
\end{cases} \]

The solution [A.5] is true only for the case \( N \geq D \).

When there is an infinite number of parts in stock, the probability that a machine has to wait in the ordering subsystem is zero, and a typical machine in the system is either operational or demanding repair. That is, \( p(n,s,m) = 0 \) for every \( s > 0 \). Therefore, we can obtain the probability distribution for the spares provisioning problem with no ordering subsystem by setting \( s = 0 \) in [A.5]. Since the ordering subsystem essentially vanishes, the state of the system can be represented by the 2-tuple \( (n,m) \).

Finally, when considering only the extreme inventory policies for the parts, the system can be modeled as a cyclic queue. In other words, the machines must flow in a cyclic fashion and cannot bypass any subsystem in the system; a situation which arises when there is a finite nonzero stocking of the parts. For a better understanding of cyclic queueing networks, the reader should refer to Koenigsburg [57].
Appendix B: Monotonicity Properties

Let $\tilde{n}_o$ be a random variable which represents the equilibrium number of operational ready or "up" machines. Clearly, $\tilde{n}_o$ is a function of the number of machines and repair stations in the system, $N$ and $M$, and what needs to be shown is that as we increase $N$ or $M$, $\tilde{n}_o$ would increase stochastically at steady state. That is, $P(\tilde{n}_o(N,M) > n)$ is a nondecreasing function of $N$ and $M$. More formally stated; suppose $(N,M) \leq (N',M')$ if and only if $N \leq N'$ and $M \leq M'$, then $\tilde{n}_o(N,M) \leq \tilde{n}_o(N',M')$, where $\tilde{n} \leq \tilde{n}'$ if and only if $Pr(\tilde{n} \geq N) \leq Pr(\tilde{n}' \geq N)$ for all $N$.

To prove that $\tilde{n}_o$ is monotonically nondecreasing in $N$, we model the system with $N + 1$ machines as a preemptive priority network, with $N$ high priority machines and one low priority machine. The distribution of the total number of machines at each subsystem will be identical to that of the nonpriority system with $N + 1$ machines. This is because the two priority classes have identical failure and repair rates. The distribution of the high priority machines at each subsystem will be identical to that of the nonpriority system with $N$ machines. The one low priority machine will spend part of its time at the production floor; thus,
\[ \tilde{n}_q(N + 1, M)_{st} = \tilde{n}_q(N, M) + \tilde{n}_q \]  

[B.1]

where \( \tilde{n}_q \) is the number of low priority machines operational at steady state. Since \( \tilde{n}_q \geq 0 \), this implies \( \tilde{n}_q(N + 1, M)_{st} \geq \tilde{n}_q(N, M) \). This simple logic has become a "standard" proof in queueing network theory to show that traffic is heavier between nodes as the number of customers in the system increases. See, for example, Walrand [88], p. 219.

To verify the second case, \((N, M) \leq (N, M + 1)\), another concept of ordering between discrete random variables should be introduced. Suppose \( X_1 \) and \( X_2 \) are integer-valued random variables, and \( Pr(X_2 = N)/Pr(X_1 = N) \) is a nondecreasing function of \( N \). Then, \( X_1 \) and \( X_2 \) are ordered in the sense of monotone likelihood ratios, and this is denoted by \( X_1 \preceq X_2 \). It is known that \( X_1 \preceq X_2 \) implies \( X_1_{st} \leq X_2 \). See Lehmann [61] and Ross [74] for further discussion.

The following useful Lemma from Gross et al. [45] should be presented before proceeding with the proof.

**Lemma:** Suppose \( X_1, X_2 \) and \( Y \) are independent integer-valued random variables, and for some \( c \), \( Z_1_{st} = Y(X_1 + Y = c) \) and \( Z_2_{st} = Y(X_2 + Y = c) \). If \( X_1 \preceq X_2 \), then \( Z_1 \preceq Z_2 \).

By applying the definition of \( X_1 \preceq X_2 \), the Lemma was proved as follows:

\[
\frac{Pr(Z_1 = n)}{Pr(Z_2 = n)} = \frac{Pr(Y = n | X_1 + Y = c)}{Pr(Y = n | X_2 + Y = c)}
= \frac{Pr(Y = n, X_1 + Y = c) / Pr(X_1 + Y = c)}{Pr(Y = n, X_2 + Y = c) / Pr(X_2 + Y = c)}
= \frac{Pr(Y = n, X_1 = c - n) / Pr(X_1 + Y = c)}{Pr(Y = n, X_2 = c - n) / Pr(X_2 + Y = c)}
\]
\[
\begin{align*}
&= \frac{Pr(Y = n)Pr(X_1 = c - n)|Pr(X_1 + Y = c)}{Pr(Y = n)Pr(X_2 = c - n)|Pr(X_1 + Y = c)} \\
&\geq \frac{Pr(Y = n + 1)Pr(X_1 = c - (n + 1))|Pr(X_1 + Y = c)}{Pr(Y = n + 1)Pr(X_2 = c - (n + 1))|Pr(X_2 + Y = c)} \\
&= \frac{Pr(Y = n + 1, X_1 + Y = c)|Pr(X_1 + Y = c)}{Pr(Y = n + 1, X_2 + Y = c)|Pr(X_2 + Y = c)} \\
&= \frac{Pr(Z_1 = n + 1)}{Pr(Z_2 = n + 1)}.
\end{align*}
\]

Consider the form of the joint probability distribution of the number of units at each stage given by [A.5]. We can think of \( p(n,s,m) \) as the conditional distribution of three independent random variables given that their sum equals the total number of machines in the system. If we denote these independent random variables by \( Z_1(D), Z_2(S) \) and \( Z_3(M) \) corresponding to the operational, ordering and repair subsystems respectively, then

\[
Pr(Z_3(M) = m) = \begin{cases} 
\frac{1}{m!} \left( \frac{\lambda}{\mu} \right)^m \pi_0, & m \leq M \\
\frac{1}{M! M^{m-M}} \left( \frac{\lambda}{\mu} \right)^m \pi_0, & m > M
\end{cases}
\]  \[B.2\]

where \( \pi_0 \) is the appropriate normalization constant. It is easily seen that if \( m \geq M + 1 \) (the case where \( m \leq M + 1 \) being trivial),

\[
\frac{Pr(Z_3(M) = m)}{Pr(Z_3(M + 1) = m)} = \left( \frac{M + 1}{M} \right)^{m-M},
\]

is nondecreasing in \( m \) and, therefore,

\[
Z_3(M + 1) \leq Z_3(M). \]  \[B.3\]

From Eq. [A.5], it follows that

\[
\tilde{\eta}_d(N,M) + \tilde{\eta}_d(N,M) st = Z_1(D) + Z_2(S) | Z_1(D) + Z_2(S) + Z_3(M) = N, \]  \[B.4\]

Appendix B: Monotonicity Properties
where \( \tilde{n}_a \) is the number of machines waiting for parts in the ordering subsystem. From the Lemma and \([B.3]\) and \([B.4]\), it is seen that
\[
\tilde{n}_a(N,M) + \tilde{n}_o(N,M) \geq \tilde{n}_a(N,M + 1) + \tilde{n}_o(N,M + 1). \tag{B.5}
\]
Consider now
\[
Pr(Z_1(D) \geq n | Z_1(D) + Z_2(S) + Z_3(M) = N) = \sum_i Pr(Z_1(D) \geq n | Z_1(D) + Z_2(S) = i | Z_1(D) + Z_2(S) + Z_3(M) = N) = \sum_i Pr(Z_1(D) \geq n | Z_1(S) + Z_2(S) = i | Z_1(D) + Z_2(S) + Z_3(M) = N). \tag{B.6}
\]

Define \( \phi(i) = Pr(Z_1(D) \geq n | Z_1(D) + Z_2(S) = i) \), which is a nondecreasing function of \( i \). That is,
\[
Z_1(D) | Z_1(D) + Z_2(S) = i \, \text{s.t.} \, Z_1(D) + Z_2(S) = i + 1.
\]
Thus, using \([B.4]\), \([B.6]\) becomes
\[
Pr(\tilde{n}_a(N,M) \geq n) = E[\phi(\tilde{n}_a(N,M) + \tilde{n}_o(N,M))]. \tag{B.7}
\]
From Lehmann [61], it is known that \( X_1 \, \text{s.t.} \, X_2 \) implies \( E[\phi(X_1)] \leq E[\phi(X_2)] \); thus, \([B.5]\) and \([B.7]\) give
\[
Pr(\tilde{n}_a(N,M + 1) \geq n) \geq Pr(\tilde{n}_a(N,M) \geq n), \tag{B.8}
\]
which is equivalent to
\[
\tilde{n}_a(N,M + 1) \, \text{s.t.} \, \tilde{n}_a(N,M). \tag{B.9}
\]
For a proof that applies to general Jacksonian networks, the reader should refer to Gross et al. [45].

Using a sample path argument, Adan and Van der Wal [1] present a more recent proof on monotonicity which applies to non-product-form networks with general service times.

Appendix B: Monotonicity Properties 138
Monotonicity properties, however, were only shown in terms of the number of customers (machines in our case) in a closed queueing network.
References


References


References 144


Vita

Nadim Abboud was born on August 12, 1962, in Jeb-Jannine, Lebanon. He came to the United States in 1976, soon after the Lebanese civil war, and graduated from Blacksburg High School in 1979. During the same year, he joined Virginia Tech and graduated in the Spring of 1983 with a B.S. degree in Industrial Engineering and Operations Research. He then went to Syracuse University to obtain a Master's degree in Operations Research, where he stayed until December of 1984. In the Spring of 1985, the author came back to Virginia Tech and joined the Ph.D. program, which he successfully completed on August 2, 1990. It was during the Ph.D. years at Virginia Tech that the author met and fell in love with Jihane Hanania. Marriage, it seems, is eminent.

The author's immediate plan is to join the Operations Research staff at United Airlines. His long term goals are to become a millionaire, and buy a mansion and a yacht.