Closed loop repairable item systems

Lieneke Spanjers

Master's Thesis

University of Twente, The Netherlands

Faculty of Electrical Engineering, Mathematics,
and Computer Science

Department of Applied Mathematics

Supervising committee:

Prof. Dr. W.H.M. Zijm

Dr. J.C.W. van Ommeren

Dr. M.C. van der Heijden
(Faculty of Business, Public Administration
and Technology)

February 2004

Summary

Repairable inventory theory involves designing inventory systems for items which are repaired and returned to use rather than discarded. This thesis is on closed loop repairable item systems. The goal of these systems is to maintain a number of production facilities in optimal operational condition. Each production facility consists of a number of identical machines. These machine may fail incidentally.

In this thesis two types of (stochastic) models are considered; two-echelon models and two-indenture models. The two-echelon model consists of multiple bases and a depot. Each base consists of a production cell and a local repair facility to repair broken machines. The depot consists of a central repair facility. Machines that cannot be repaired locally are repaired here.
In the two-indenture model every machine consists of multiple components. These components are subject to failures. The model considers one base. This base consists of a production cell, a disassembly facility, a repair facility and an assembly facility.

In both of these models spare machines are available to replace broken machines immediately. If more machines are broken than spares are available, a backlog occurs. In that case the production cell performs worse. The aim is to develop an approximation method to obtain relevant performance measures for the system. The final goal is to determine what stock allocation will lead to the highest availability, that is, we optimize the probability that the maximum number of machines is operational.

To determine the steady state probabilities of the systems, we develop a slightly aggregated system model and propose a special near-product-form solution. For the two-echelon model the central repair facility is modeled as a typical server with state dependent service rates, of which the parameters follow from an application of Norton's theorem for Closed Queuing Networks. An adaptation to a general Multi-Class MDA-algorithm is proposed, by which numerical results are obtained.
For the two-indenture model the repair facility is also modeled as a typical server with state dependent service rates. However, for this model it is more complicated to obtain these rates because there are components of different types in the repair shop. An approximation based on Norton's theorem and the hypergeometric distribution is presented. Again, an adaptation to a general MDA-algorithm is proposed to obtain the required performance measures.
To determine what stock allocation will lead to the highest availability, a greedy algorithm is proposed. With this greedy algorithm a fairly good approximation can be obtained.

Samenvatting

Deze doctoraalscriptie gaat over gesloten systemen met repareerbare machines. Het doel van deze systemen is het onderhouden van een aantal productieafdelingen. Deze productieafdelingen zijn bij voorkeur zoveel mogelijk volledig operationeel. Elke productieafdeling bestaat uit een aantal identieke machines. Deze machines kunnen af en toe kapot gaan.

In deze scriptie worden twee typen (stochastische) modellen beschouwd: *twee-echelon* modellen en *twee-indenture* modellen. Het twee-echelon model bestaat uit meerdere lokale productieposten en een centrale post. Elke lokale post is opgebouwd uit een productieafdeling en een werkplaats om kapotte machines te repareren. De centrale post bestaat uit een centrale werkplaats. Hier worden machines gerepareerd die niet lokaal te repareren zijn.
In het twee-indenture model bestaat elke machine uit meerdere onderdelen. Als een machine kapot gaat komt dit doordat een van deze onderdelen kapot is. Het model beschouwt slechts één locatie. Op deze locatie bevindt zich de productieafdeling, een demontage afdeling, een werkplaats en een montage afdeling.

In beide modellen zijn reservemachines aanwezig om kapotte machines direct te vervangen. Als er meer machines kapot gaan dan er reservemachines zijn, treedt er een werkachterstand op. De productieafdeling zal dan minder kunnen produceren. Het doel is om een benaderingsmethode te vinden voor een aantal prestatiematen van het systeem. Het uiteindelijke doel is dan om te bepalen welke toewijzing van reservemachines aan de verschillende locaties tot de hoogste beschikbaarheid van machines in de productieafdeling zal leiden. Anders gezegd, we willen de kans dat een productieafdeling volledig operationeel is maximaliseren.

Om de evenwichtsverdeling van de systemen te bepalen, wordt een kleine aggregatiestap uitgevoerd. Het nieuwe model kan dan worden beschouwd als een productvorm netwerk. In het twee-echelon model wordt de centrale werkplaats gemodelleerd als een server met toestandsafhankelijke bedieningsduren. Om de parameters hiervoor te bepalen wordt de Stelling van Norton toegepast. De benaderingen worden uiteindelijk verkregen met een aangepaste versie van het standaard Multi-Class MDA-algoritme.
In het twee-indenture model wordt de werkplaats eveneens gemodelleerd als een server met toestandsafhankelijke bedieningsduren. Voor dit model is het iets ingewikkelder om deze bedieningsduren te bepalen. Dit komt doordat zich in de werkplaats onderdelen van verschillende typen bevinden. Er wordt een benadering gegeven op basis van de Stelling van Norton en de hypergeometrische verdeling. Wederom is er een aanpassing gedaan op een bestaand MDA-algoritme om de prestatiematen van het systeem te bepalen.
Om te bepalen welke voorraadallocatie tot de hoogste beschikbaarheid van machines in de productieafdeling leidt, wordt gebruik gemaakt van een greedy algoritme. Met dit algoritme kan de optimale oplossing redelijk goed benaderd worden.

Preface

The research described in this Master's thesis has been conducted at the University of Twente, as the final part of my study of Applied Mathematics. I have chosen to do this project at the university to experience working in an academic environment. The field of logistics has always appealed to me. Therefore I am grateful for the opportunity to work on the subject *Closed loop repairable item systems*. I have been working on the subject with pleasure.

I would like to thank Henk Zijm and Jan-Kees van Ommeren for their supervision. In spite of his busy schedule Henk Zijm offered to supervise me, which made it possible for me to work on this subject. Jan-Kees van Ommeren has always been there to answer my questions and to listen to my complaints about computers.

Moreover, I want to thank my parents for their loving support throughout the years. I appreciate their confidence in me and my decisions.

Last but not least I would like to thank Gustaaf. Whether things were well, not so well, or just not well in my perspective, he has always been supportive. Gustaaf, after five years I am looking forward to finally find out what you are like on weekdays!

Lieneke Spanjers,
Enschede, 6 February 2004

Contents

[1]The research presented in this section is the outcome of previous research by A. Daryanto, J.C.W. van Ommeren and W.H.M. Zijm.

Chapter 1

Introduction

In this chapter an introduction will be given on closed loop repairable item systems. An example of closed loop repairable item systems will be presented to clarify the concept. The economic need for mathematical analysis of it will be explained. We will briefly discuss what research on this subject has been done in the past and what research objective is defined for this thesis. It consists of three parts, which will be explained below. An outline of the entire report will be given in Section 1.4 and the chapter finishes with some remarks on the notation of variables that are used throughout this thesis.

1.1 Closed loop repairable item systems

Consider a company that runs several copy shops all over the Netherlands. Each of these copy shops features a number of identical copying machines. The company gets its revenues from customers using the copiers.
Since the copiers are all used continuously, a broken copier will lead to loss of profit, if not replaced immediately. Therefore, each copy shop holds a number of spare machines available for the replacement of broken down machines. If the copy shop is out of spare machines, a broken copier will not be replaced until a spare machine becomes available.
Furthermore, each copy shop has a repair facility to repair broken copying machines. The repair of some failures however, requires specific expertise. This kind of repairs is performed at the specialized central repair shop, located in the center of the Netherlands. Because of transportation times, the central repair facility also holds some spare machines. Whenever a copier is planned to be sent from a copy shop to the central repair shop, a spare copier is immediately shipped from the central facility to that copy shop to replenish its stock of spare copying machines.

Holding spare copiers and paying for repair capacity costs money. However, insufficient repair capacity or a shortage of spare machines will lead to decreased profits. Since machine failures are unpredictable, several failures can follow upon each other shortly and a shortage of spare machines will be the result. The capacity of the copy shop will be diminished, which will lead to less profit.
Furthermore, when several failures follow upon each other shortly, the repair capacity might not be sufficient. A broken machine then has to 'wait' before it is repaired. It is likely to take longer to replenish the spare machines stock. Hence a shortage of spare machines is more likely to occur.

Figure 1.1: Copy shops with repair facility and a central repair facility on different locations in the Netherlands

As a profit maximizing entity, the company will be interested to learn how to achieve the highest possible performance, given a fixed budget for spares and repair capacity. Or, given a required performance, what is the most economical repair strategy? Moreover, what is the most economical number of spares on all different locations?

To find the answers, the company could try its hand at closed loop repairable item systems: a mathematical model that can be used for the analysis. The copy company can be modeled as a *closed loop two-echelon repairable item system*. The *system* contains the copy shops and its repair shops, the central repair shop and all transport lines in between. It is a *repairable item* system because all copiers are repaired instead of discarded and it is a *closed loop* system because all items (copiers) in the system, remain in the system and no new items enter the system. It is a *two-echelon* system because there are two levels of locations. The central repair shop represents the first level and the local copy shops are the second level. In case there would be more levels of locations the term multi-echelon would apply.
Because the machines in the system described above, are considered as a whole, the system can also be called a single-indenture system, but to simplify the terminology we will not mention this classification any further is this thesis. Also *multi-indenture* systems exist. In that case every part (machine) is modeled as a layered product structure, consisting of multiple components. These components can again be constructed of smaller components, etcetera. In a two-indenture model there are two levels in the product structure of the parts. For the copying machines for instance, the machine itself is level one, and the components that take care of the paper supply and the printing process can be considered as level two.

For a company like the copy company, it is important to know how to invest its budget to attain maximal availability of copying machines, since this will maximize their profits.

A bad allocation of spare machines over the different locations, can lead to much lower profits.
For many other companies the analysis of closed loop repairable item systems can be of great value as well. For instance, every company that has a production line wants to have as much machines operational as possible. Next to this, closed loop repairable item systems can be applied on other items than machines. Think for example of aviation and transportation equipment. The mathematical models are applicable to all sorts of items which are repaired and returned to use rather than discarded because the items are less expensive to repair than to replace.
From now on, we will no longer discuss the copy company from the introduction. Every machine mentioned from this point on, will be considered to be part of production cells. These production cells are located at different bases. The central repair facility will be referred to as the depot.

In this thesis we are interested in closed loop repairable item systems. We are interested in multi-echelon systems as well as multi-indenture systems. The analysis will entirely be focused on the mathematical models. Of course, when a model will be used for a real-life situation, it must be checked whether all assumptions are satisfied. Moreover, all parameters that are used in the model must be obtained from empirical findings. This can be quite difficult. We will not go into that here.
As will become clear in the next chapters, the analysis will not be very simple. For large systems it will not even be possible to find exact solutions. The main effort will therefore be to find good approximations for several performance measures.

1.2 Previous research

Multi-echelon inventory theory has been the subject of research for over 35 years now. Most models focused on steady state distributions. In 1968 Sherbrooke [12] was the first to present the METRIC (Multi-Echelon Technique for Recoverable Item Control) model. This model was developed for the US Air Force at the Rand Corporation. In this model a machine at failure is replaced by a spare if one is available, otherwise it is backordered. A certain proportion of the failed machines is repaired at the base and the rest at a repair depot, thereby creating a two-echelon repairable-item system. Machines are returned from the depot using a one-for-one reordering policy. The aim is to determine the optimal level of spares to be maintained at each of the bases and at the depot. Muckstadt [11] managed to generalize the METRIC model to allow for a multi-indenture parts structure. His new model was called MODMETRIC.
Over the years, some of the constraints imposed on the original METRIC model have been relaxed and different versions have been developed. However, a shortfall of the METRIC model and its successors, is that it assumes that failures are Poisson from an infinite source and that the repair capacity is unlimited. Therefore, others have continued the research to gain results more useful for real life applications. They investigated the possibility to introduce models with limited repair capacity. Avsar and Zijm successfully did this for both a two-echelon model [4] and a two-indenture model [13]. Díaz and Fu [6] also developed a multi-echelon model with limited repair capacity.
Gross [7], Albright and Soni [1] and Albright [2] took another step in the analysis of the two-echelon model. Not only did they drop the assumption of unlimited repair capacity,

but also the assumption of Poisson failures from an infinite source. They completely focused their attention on closed queuing network models. This means that the intensity by which machines enter the repair shops depends on the number of machines operating in the production cell and is therefore not constant. In case of a backlog at a base, this intensity is smaller than in the optimal case where the maximum number of machines is operating in the production cell.
For a closed loop two-indenture model, a similar approach is developed by Gupta [9]. In collaboration with Albright, he even extended this model to a two-echelon two-indenture model in [8] and [3].
Recently Daryanto, Van Ommeren and Zijm examined how the approximation method for open two-indenture systems as described in [13], could be applied to (simple) closed loop two-indenture systems. Their research will be presented in Section 3.1 since it is used as a basis for research in this thesis.

1.3 Research objective

We are interested in closed loop repairable item systems. We assume repair capacity is limited and that the total failure rate at a certain time depends on the number of operational machines at that time. Our attention is focused on multi-echelon models as well as multi-indenture models. The focus will be on steady state solutions, particularly on how to distribute spares in the system, so as to achieve the best possible performance of the system.
As described above, the models by Gross, Albright and Gupta already discussed these systems and quite reasonable approximation methods were found. For now, an attempt to develop a different approach will be made. The aim is to develop a method that can be applied on more complicated systems as well. Daryanto, Van Ommeren and Zijm previously examined how to apply the approximation method for open two-indenture systems as described in [13], on (simple) closed systems. Similar research will be done here. The research by Daryanto, Van Ommeren and Zijm for two-indenture systems will be extended and the approximation method for open two-echelon systems as described in [4], will be adjusted to comply with closed two-echelon repairable item systems.

The aim of this thesis is to answer the following questions.

- Consider a closed loop two-echelon repairable item system consisting of several bases and a depot, with (simple) repair facilities and spare machines both at the depot and at the bases. Lateral supply between bases is not allowed and all failed items are assumed to be repairable. Find a reasonable approximation for the availability (full occupation of a production cell) and the expected number of operational machines at the production cells, based on the approach in [4]. With reasonable we denote a maximum error of 5%.

- Consider a closed loop two-indenture repairable item system with one base. Every machine consists of several components that may fail independently and need to be repaired in the (simple) repair facility before being assembled in the (simple) assembly facility. In the system there are spare machines as well as spare components. Find a reasonable approximation for the availability (full occupation of the production cell) and the expected number of operational machines at the production cell, based on the approach in [13]. With reasonable we denote a maximum error of 5%.

- Find an algorithm to optimize the availability for the models described above, by allocating spares to different locations while there is a budget constraint for stocking costs.

The combination of two-echelon models and two-indenture models, that is models that describe a system that consists of several bases and a depot in which the machines consist of several repairable components, is also very interesting. However, these models are beyond the scope of this thesis. Models with complex repair structures are also not taken care of in this thesis. More about such extensions is said in the concluding Chapter 5.

1.4 Structure of this thesis

In the next chapter a two-echelon system will be analyzed. The aim is to find a good approximation for several performance measures, such as availability and the expected number of operational machines. First a system that contains only one base will be discussed. This simple model serves to show what the suggested approximation method is like. This model is then extended to allow for multiple bases, where a similar approximation step is performed. For this more complicated case, this leads to an adapted Marginal Distribution Analysis algorithm. Results for the approximations are compared to simulation results.
In Chapter 3 a similar construction is used to analyze a two-indenture repairable item system. The model consists of one base and in the first section a simple model is discussed where machines consist of only one type of critical components. The analysis for this simple model was carried out as part of previous research by Daryanto, Van Ommeren and Zijm. In the subsequent section my attempts are shown to extend the analysis for a model consisting of two critical components. An even more complicated adaptation to a Marginal Distribution Analysis algorithm is the result. Again, the aim is to find an approximation for several performance measures. The results for these approximations are compared to simulation results so as to be able to judge the accuracy of the proposed algorithm.
Next, in Chapter 4 an algorithm is developed to optimize the utilization of spares in the systems. Given a certain budget to cover stocking costs, it is determined what distribution of spares in the system will lead to the highest availability. In combination with the approximation methods for the two-echelon system, results are obtained to validate the algorithm.
In the last chapter conclusions will be drawn and some remarks will be made on future research.

The Appendix contains all supporting materials. Something will be said on the simulation models that were used, extra results are given and additional theory is presented. Furthermore, auxiliary theorems are given and some representative program code is shown. The Appendix also contains a list of symbols used throughout this thesis.

1.5 Remarks on the notation of variables

In this thesis various symbols are used to represent variables. The complete list of symbols that is used, is presented in Appendix F. Please note that each chapter has its own list of symbols. Although some symbols occur in more than one chapter, the meaning of the symbol does not necessarily have to be the same. An attempt has been made to let the

notation correspond to the most common notation in the field. Unfortunately, the result of this is that in the presentation of the two-echelon model (Chapter 2) and the presentation of the two-indenture model (Chapter 3) several symbols had to be reused.
Another remark concerns the notation of random variables. Even though the most common notation in the field is to represent random variables by capital letters, in this thesis random variables are denoted by a dash under the symbol. This choice has been made because several parameters are denoted by capital letters and we do not want to confuse the reader.

Chapter 2

Closed loop two-echelon repairable item systems

In this chapter we will discuss closed loop two-echelon repairable item systems. The aim is to find the steady state probabilities of the model, so that performance measures such as availability and expected number of working machines can be obtained. As will be explained in Subsection 2.1.1, for large systems it will be very time consuming to find exact solutions. We therefore focus on developing good approximation procedures. The analysis in this chapter is based on the analysis in [4].
In the next section we consider a very simple two-echelon system, consisting of a single base and a depot. Both at the base and at the depot there is a repair shop. These repair shops are modeled as single servers. The model mainly serves to explain the essential elements of the suggested approximation procedure. Numerical results are presented to demonstrate the accuracy of the approximation. Next, in Section 2.2, we turn to more general repairable item network structures, containing multiple bases and transport lines from the depot to the bases. The repair shops are modeled as multi-servers. The approximation method leading to an adapted Multi-Class MDA algorithm is presented and some numerical results are discussed. In the last section, conclusions will be drawn and several remarks will be made.

2.1 Analysis of a simple two-echelon system with single server facilities

In this section a simplified repairable item system is discussed, to explain how a slight modification turns this system into a near-product form network that can be completely analyzed. In the next section we turn to more complex systems.

2.1.1 The single base model without transportation

Consider the system as depicted in Figure 2.1. The system consists of a single base and a depot. At the base a maximum of J_1 machines can be operational in the production cell. Operational machines fail at exponential rate λ_1 and are replaced by a machine from the base stock (if available). Both at the base and at the depot there is a repair shop. Failed machines are base-repairable with probability p_1 and consequently depot-repairable with probability $1 - p_1$. The repair shops are modeled as single servers with exponential service

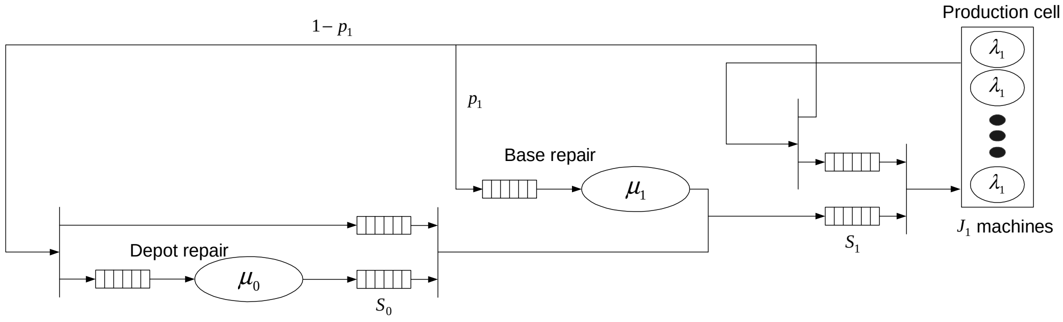


Figure 2.1: The single base two-echelon repairable item system

rate μ_0 for the depot and exponential service rate μ_1 for the base. In addition to the J_1 machines another group of S_1 machines is dedicated to the base to act as spares. When a machine fails, the failed machine goes to a repair shop while at the same time a spare machine from the base stock is placed in the production cell. If there are no spare machines at the base, a backlog occurs. As soon as there is a repaired machine available, it becomes operational. A number of S_0 machines is dedicated to the depot to act as spares. When a failed machine cannot be repaired at the base and hence is sent to the depot, a spare machine is shipped from the depot to the base to replenish the base stock, or - in case of a backlog - to become operational immediately. When no spares are available at the depot, a backorder is created. In that case, as soon as a machine is repaired at the depot repair shop, it is sent to the base. In this simple model, transport times from the base to the depot and vice versa are not taken into account.

In Figure 2.1, the matching of a request and a ready-for-use machine is modeled as a synchronization queue, both at the base and at the depot. At the base however, some reflection reveals that the synchronization queue can be seen as a normal queue where machines are waiting to be moved into the production cell. This is only possible when the production cell does not contain the maximum number of machines, that is, if a machine in the production cell has failed. This leads to the model in Figure 2.2. In this figure the variables n_1, n_2,

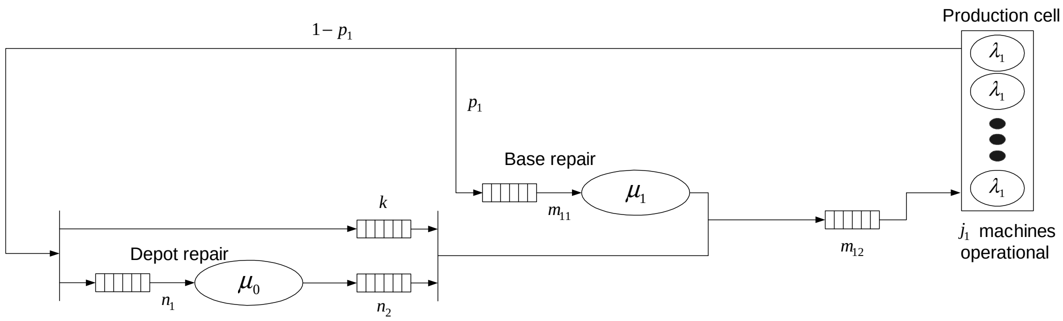


Figure 2.2: The modified single base two-echelon repairable item system

k, m_{11} and m_{12} indicate the lengths of the various queues in the system. The number of machines in (or awaiting) depot repair is denoted by the random variable $\underline{n}_1$, the number of spare machines at the depot is denoted by the random variable $\underline{n}_2$ and the backlog of machines at the depot is denoted by $\underline{k}$. At the base there are $\underline{m}_{11}$ machines waiting for repair or being repaired and $\underline{m}_{12}$ machines are acting as spares. In the production cell $\underline{j}_1$ machines are operational.

As a result of the operating inventory control policies, the following equations must hold for $\underline{n}_1 = n_1$, $\underline{n}_2 = n_2$, $\underline{k} = k$, $\underline{m}_{11} = m_{11}$, $\underline{m}_{12} = m_{12}$ and $\underline{j}_1 = j_1$:

$$n_1 + n_2 - k = S_0, \tag{2.1}$$

$$n_2 \cdot k = 0, \tag{2.2}$$

$$k + m_{11} + m_{12} + j_1 = S_1 + J_1, \tag{2.3}$$

$$m_{12} \cdot (J_1 - j_1) = 0. \tag{2.4}$$

Equations (2.2) and (2.4) follow from the fact that it is impossible to have a backlog and to have spare machines available at the same time. If spare machines are available, a request is satisfied immediately. In case of a backlog, a request is not satisfied until a repair completion. The repaired machine is merged with the longest waiting request.
From these relations it follows immediately that n_1 and m_{11} completely determine the state of the system, including the values of n_2, k, m_{12} and j_1. Therefore, the system can be modeled as a continuous time Markov chain with state description (n_1, m_{11}). The corresponding transition diagram is displayed in Figure 2.3.

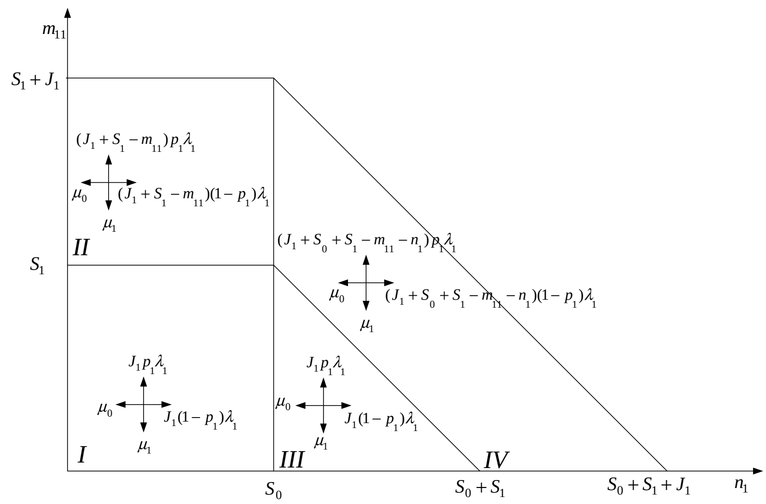


Figure 2.3: Transition diagram for state description (n_1, m_{11})

Let $P(n_1, m_{11}) = P(\underline{n}_1 = n_1, \underline{m}_{11} = m_{11})$ be the steady state probability of being in state (n_1, m_{11}). This steady state probability can be found by solving the global balance equations of the system. These can be deduced from the transition diagram. Note however that for large systems with many machines and e.g. multiple bases, the state space will also be rather large. As a consequence the number of global balance equations that need to be solved to obtain the steady state distribution is extremely large and the computational effort becomes prohibitive.
Unfortunately, it is also not possible to find an algebraic expression for the steady state probabilities. Therefore the system will be slightly adjusted in the next subsection, in order to arrive at a near-product form network. Product form networks have the pleasant

characteristic that an algebraic expression for the steady state probabilities does exist. Also several other techniques are known that can be applied to product form networks. The near-product form network that will be obtained in the next subsection can be analyzed as if it is a product form network. This leads to good approximations for the steady state probabilities, even for large systems.

2.1.2 Approximation

A first step towards an approximation for the steady state probabilities is to aggregate the state space. The most difficult parts of the transition diagram are regions I and II, that is, the parts with $n_1 \leq S_0$ or, equivalently, the parts with $k = 0$. The parts with $k > 0$ are equivalent to the states with $n_1 = k + S_0$. A natural aggregation of the system is a description through the states (k, m_{11}). The states (n_1, m_{11}) with $n_1 = 0, 1, \ldots, S_0$ are then aggregated into one state $(0, m_{11})$. Denote the steady state probabilities for the new model by $\tilde{P}$ then the following holds for any m_{11}:

$$\tilde{P}(\underline{k} = 0, \underline{m}_{11} = m_{11}) = \sum_{n_1=0}^{S_0} P(\underline{n}_1 = n_1, \underline{m}_{11} = m_{11}), \tag{2.5}$$

$$\tilde{P}(\underline{k} = k, \underline{m}_{11} = m_{11}) = P(\underline{n}_1 = S_0 + k, \underline{m}_{11} = m_{11}). \tag{2.6}$$

The transition diagram corresponding to the alternative state space description is displayed in Figure 2.4. The rates only differ from the transition diagram in Figure 2.3 for the case

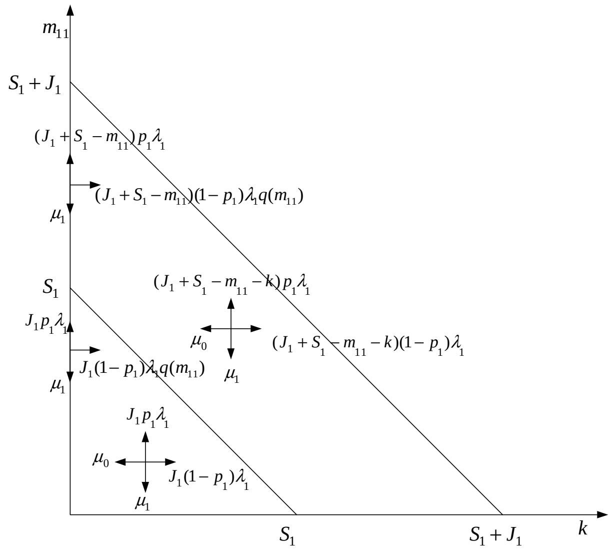


Figure 2.4: Transition diagram for state description (k, m_{11})

$k = 0$. Let $q(m_{11})$ be the steady state probability that an arriving request for a machine at the depot has to wait, given that it finds no other waiting requests in front of it ($k = 0$) and $\underline{m}_{11} = m_{11}$. Given the (aggregated) state $(0, m_{11})$, the state does not change in case of an arriving request with probability $1 - q(m_{11})$, because spares are available. With probability $q(m_{11})$ no spares are available and the state changes into $(1, m_{11})$. The transition rate from $(0, m_{11})$ to $(1, m_{11})$ equals $j_1\,(1 - p_1)\,\lambda_1\,q(m_{11})$, where $j_1 = J_1 - (m_{11} - S_1)^+$. To determine

$q(m_{11})$ one needs

$$q(m_{11}) = P(\underline{n}_1 = S_0 \mid \underline{n}_1 \leq S_0, \underline{m}_{11} = m_{11}). \tag{2.7}$$

However, to compute this, one needs to know the steady state distribution of the original system, which is exactly what we attempt to approximate. Therefore, we approximate the $q(m_{11})$'s by their weighted average, i.e. we focus on the conditional probability q defined by

$$q = \sum_{m_{11}} q(m_{11})\, P(\underline{m}_{11} = m_{11} \mid \underline{n}_1 \leq S_0) = P(\underline{n}_1 = S_0 \mid \underline{n}_1 \leq S_0) \tag{2.8}$$

and for every m_{11} we replace $q(m_{11})$ in the transition diagram by this q. In the next subsection will be explained how a reasonable approximation for this q can be obtained by means of an application of Norton's theorem.

Lemma 2.1.1
The steady state probabilities for the model with state description (k, m_{11}) and transition rates as denoted in Figure 2.4 with $q(m_{11})$ replaced by arbitrary q have a product form.

Proof. To find the steady state probabilities, consider both the original model in Figure 2.2 and the alternative model in Figure 2.5. In Figure 2.5 the depot repair shop with

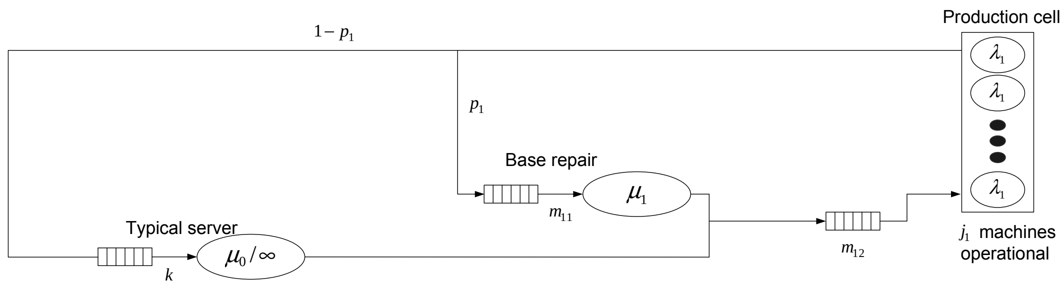


Figure 2.5: Typical-server Closed Queuing Network (TCQN)

synchronization queue is replaced by a typical server. For jobs that find the server idle the server has infinite service rate with probability $1-q$ (the case spares are available) and service rate μ_0 with probability q (the case no spares are available). Let $\underline{b}_1$ be the random variable equal to $\underline{m}_{12} + \underline{j}_1$, then by looking at the system with the typical server, and conditioning on the fact that the network contains exactly $J_1 + S_1$ jobs, it can be verified that the following expression for $\tilde{P}(\underline{k} = k, \underline{m}_{11} = m_{11}, \underline{b}_1 = b_1)$ satisfies the balance equations of the TCQN:

$$\tilde{P}(k, m_{11}, b_1) = \begin{cases} G'\, q \left(\frac{p_1}{\mu_1}\right)^{m_{11}} \left(\frac{1-p_1}{\mu_0}\right)^{k} \frac{\left(\frac{1}{\lambda_1}\right)^{b_1}}{J_1! J_1^{b_1 - J_1}}, & b_1 > J_1, k > 0, \\ G'\, q \left(\frac{p_1}{\mu_1}\right)^{m_{11}} \left(\frac{1-p_1}{\mu_0}\right)^{k} \frac{\left(\frac{1}{\lambda_1}\right)^{b_1}}{b_1!}, & b_1 \leq J_1, k > 0, \\ G' \left(\frac{p_1}{\mu_1}\right)^{m_{11}} \frac{\left(\frac{1}{\lambda_1}\right)^{b_1}}{J_1! J_1^{b_1 - J_1}}, & b_1 > J_1, k = 0, \\ G' \left(\frac{p_1}{\mu_1}\right)^{m_{11}} \frac{\left(\frac{1}{\lambda_1}\right)^{b_1}}{b_1!}, & b_1 \leq J_1, k = 0, \end{cases} \tag{2.9}$$

with $k + m_{11} + b_1 = J_1 + S_1$ and G' the normalization constant. □

Expressed in terms of the state variables (k, m_{11}), this result immediately leads to:

Lemma 2.1.2
The steady state distribution for the aggregate model is given by

$$\tilde{P}(k, m_{11}) = \begin{cases} \frac{Gq}{J_1! J_1^{S_1-k-m_{11}}} \left(\frac{p_1\lambda_1}{\mu_1}\right)^{m_{11}} \left(\frac{(1-p_1)\lambda_1}{\mu_0}\right)^k, & k + m_{11} \leq S_1, k > 0, \\ \frac{Gq}{(S_1+J_1-k-m_{11})!} \left(\frac{p_1\lambda_1}{\mu_1}\right)^{m_{11}} \left(\frac{(1-p_1)\lambda_1}{\mu_0}\right)^k, & k + m_{11} > S_1, k > 0, \\ \frac{G}{J_1! J_1^{S_1-m_{11}}} \left(\frac{p_1\lambda_1}{\mu_1}\right)^{m_{11}}, & m_{11} \leq S_1, k = 0, \\ \frac{G}{(S_1+J_1-m_{11})!} \left(\frac{p_1\lambda_1}{\mu_1}\right)^{m_{11}}, & m_{11} > S_1, k = 0, \end{cases} \quad (2.10)$$

with G the normalization constant.

The previous lemma gives an explicit expression for the steady state probabilities. Obtaining the steady state probabilities is now considerably easier, when compared to the approach of solving the balance equations, as described in Subsection 2.1.1. For large systems it may be difficult to calculate the normalization constant G. However, since we are dealing with a product form network, Marginal Distribution Analysis (see e.g. Buzacott and Shanthikumar [5]) can be used to calculate the appropriate performance measures directly.

The results presented so far hold true for any value of $q \in [0, 1]$. In the derivation of the lemmas above the interpretation of q as the conditional probability that a request at the depot has to wait given that it finds no other requests in front of it (see (2.8)), has not been used. Therefore any $q \in [0, 1]$ will do, but it is expected that a good approximation will be obtained by using a q that does correspond to this interpretation. In the next subsection Norton's theorem will be used to find a q with a meaningful interpretation that gives good results.

2.1.3 Applying Norton's theorem to approximate q

Although we have stated in the previous subsection that the product form does not depend on q, we still need to find a q that gives a good approximation for the performance measures in the system of Figure 2.2. In this subsection, the basic idea behind Norton's theorem (see Harrison and Patel [10] for an overview) is used to find an approximation for q that gives good results. This basic idea is that a product form network can be analyzed by replacing a subnetwork by state dependent servers where, for each number of customers i, the service rate is obtained by treating the subnetwork as a closed queuing network with appropriate rerouting mechanisms. This is called shortcircuiting. Norton's theorem states that the joint distribution of the number of customers at all nodes outside the subnetwork remains unchanged when replacing the subnetwork by a state dependent server in this way.

To use this idea, consider the TCQN from Figure 2.5. The base, consisting of the production cell and the base repair shop, is taken apart and replaced by a state dependent server. The new network with the state dependent server is displayed in Figure 2.6 (left graph). In order to find the service rates for this state dependent server, the original network is short circuited by setting the service rate at the typical server to infinity. This short circuited network is also depicted in Figure 2.6 (right graph). The service rate for the new state dependent server with i jobs present is equal to the throughput of the short circuited

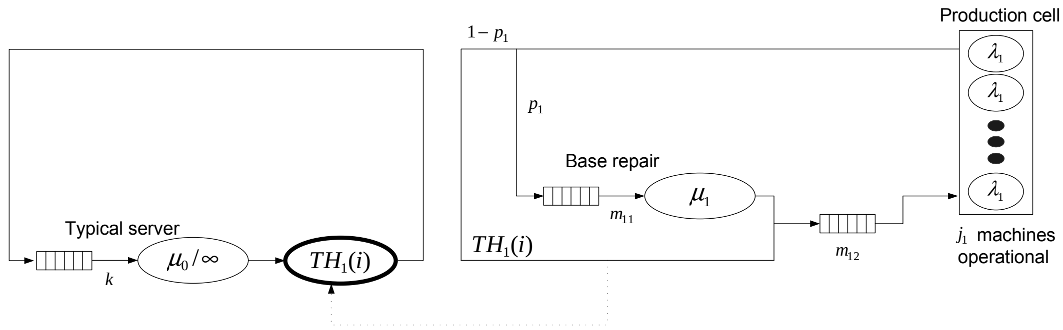


Figure 2.6: The new network with state dependent server (left graph) and the short circuited network (right graph)

network with i jobs present, denoted by $TH_1(i)$.

Now consider the original model as shown in Figure 2.2. Machines visit the repair shop with the same throughput rate $TH_1(i)$ as they visit the typical server in the TCQN-model above. See Figure 2.7.

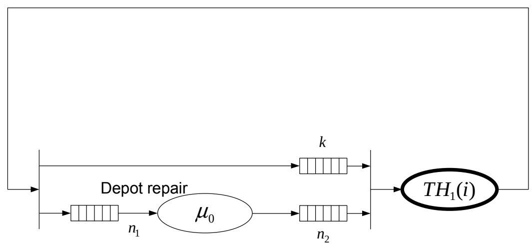


Figure 2.7: The original network with state dependent server

We want to obtain q, the conditional probability that a request corresponding with a machine failure finds no spare parts in stock at the depot, although there was no backlog so far. That is, according to (2.8)

$$q = P(\underline{n}_1 = S_0 \mid \underline{n}_1 \leq S_0).$$

To obtain this q, the evolution of $\underline{n}_1 = n_1$ is needed. This evolution can be described as a birth-death process. The transition diagram, as obtained from Figure 2.7 is shown in Figure 2.8.

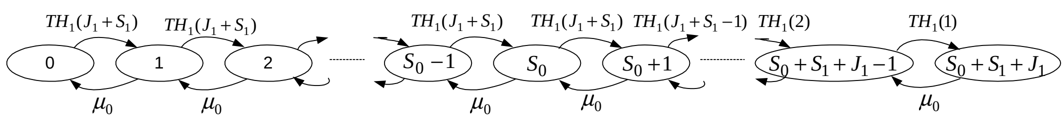


Figure 2.8: Transition diagram for n_1

Note that this is just an approximation due to the fact that Norton's theorem is only valid for product form networks. In case $S_0 = 0$, we would have a product form network and the results would be exact. From the diagram one can observe that

$$P(\underline{n}_1 = n_1)\, TH_1(J_1 + S_1 - (n_1 - S_0)^+) = P(\underline{n}_1 = n_1 + 1)\, \mu_0 \tag{2.11}$$

for $n_1 = 0, \ldots, S_0 + S_1 + J_1 - 1$. In principle one can derive an approximation of the distribution of $\underline{n}_1$ from this. However, by the definition of q, we only need to study the behavior for $\underline{n}_1 \leq S_0$. For these states, the service rate of the state dependent server is equal to $TH_1(J_1 + S_1)$. Let $\delta = TH_1(J_1 + S_1)/\mu_0$. From (2.11) we observe that $P(\underline{n}_1 = n_1) = \delta^{n_1} P(\underline{n}_1 = 0)$ for $n_1 = 0, \ldots, S_0$ so

$$\begin{aligned} q &= \frac{P(\underline{n}_1 = S_0)}{P(\underline{n}_1 \leq S_0)} = \frac{\delta^{S_0} P(\underline{n}_1 = 0)}{\sum_{n_1=0}^{S_0} P(\underline{n}_1 = n_1)} = \frac{\delta^{S_0} P(\underline{n}_1 = 0)}{\sum_{n_1=0}^{S_0} \delta^{n_1} P(\underline{n}_1 = 0)} = \frac{\delta^{S_0}}{\frac{1-\delta^{S_0+1}}{1-\delta}} \\ &= \delta^{S_0} \frac{1-\delta}{1-\delta^{S_0+1}}. \end{aligned} \tag{2.12}$$

It remains to find the throughput of the short circuited network in Figure 2.6 (right graph) with $J_1 + S_1$ jobs present. A simple observation reveals that

$$P(\underline{b}_1 = b_1)\, min(b_1, J_1)\, \lambda_1\, p_1 = P(\underline{b}_1 = b_1 - 1)\, \mu_1$$

for $b_1 = 1, \ldots, J_1 + S_1$ from which the steady state probabilities of $\underline{b}_1$ are immediately deduced. Moreover, the throughput satisfies

$$\begin{aligned} TH_1(J_1 + S_1) &= (1 - p_1) \sum_{b_1=1}^{J_1+S_1} P(\underline{b}_1 = b_1)\, min(b_1, J_1)\, \lambda_1 \\ &= \frac{1 - p_1}{p_1}\, \mu_1\, (1 - P(\underline{b}_1 = J_1 + S_1)). \end{aligned} \tag{2.13}$$

We can determine q with (2.12) and (2.13). This q can be used to approximate the steady state distribution using (2.10) or using Marginal Distribution Analysis. Results of this approximation are presented in the next subsection.

2.1.4 Results

In this subsection numerical results obtained by the approximation described above will be presented. To be able to judge the approximation, the results are compared to exact results. The exact results are obtained by solving the balance equations for the original model.

The performance measures we are interested in are the availability, i.e. the probability that the maximum number of machines is working in the production cell, denoted by A, and the expected number of machines operating in the production cell ($E\underline{j}_1$). These are defined as follows:

$$A = P(\underline{j}_1 = J_1) = P(\underline{b}_1 \geq J_1) = P(\underline{k} + \underline{m}_{11} \leq S_1), \tag{2.14}$$

$$\begin{aligned} E\underline{j}_1 &= E(J_1 - [\underline{k} + \underline{m}_{11} - S_1]^+) \\ &= \sum_{k, m_{11}} (J_1 - [k + m_{11} - S_1]^+)\, P(k, m_{11}). \end{aligned} \tag{2.15}$$

The performance measures are computed for several values of J_1, S_0, S_1, p_1, λ_1, μ_0 and μ_1. The results are given in Table 2.1 and in Tables B.1 and B.2 in Appendix B. The error

percentages are also given.
The numbers reveal that in these systems, the approximation gives an error of at most 1 %. In all other cases that we tested, we got similar results. The largest errors are attained in the cases with only a small number of spares ($S_0 > 0$) in the system. For the case $S_0 = 0$ the results are exact.

Table 2.1: Results for the single base model, $p_1 = 0.5, \lambda_1 = 1, \mu_0 = 2J_1, \mu_1 = J_1$

J	S_0	S_1	A_{exact}	A_{appr}	% error	$E\underline{j}_1 exact$	$E\underline{j}_1 appr$	% error
3	1	0	0.5651	0.5674	0.42	2.4225	2.4246	0.09
3	3	0	0.5889	0.5892	0.05	2.4572	2.4576	0.01
3	5	0	0.5901	0.5901	0.00	2.4589	2.4590	0.00
3	1	1	0.7945	0.7952	0.09	2.7283	2.7286	0.01
3	3	1	0.8110	0.8111	0.02	2.7506	2.7507	0.00
3	5	1	0.8120	0.8120	0.00	2.7518	2.7518	0.00
3	1	3	0.9506	0.9506	0.00	2.9349	2.9348	0.01
3	3	3	0.9554	0.9554	0.00	2.9412	2.9412	0.00
3	5	3	0.9557	0.9557	0.00	2.9416	2.9416	0.00
3	1	4	0.9755	0.9754	0.00	2.9677	2.9676	0.00
3	3	4	0.9779	0.9779	0.00	2.9709	2.9709	0.00
3	5	4	0.9781	0.9781	0.00	2.9711	2.9711	0.00
5	1	0	0.5369	0.5387	0.33	4.3147	4.3160	0.03
5	3	0	0.5625	0.5628	0.05	4.3581	4.3584	0.01
5	5	0	0.5639	0.5639	0.00	4.3604	4.3604	0.00
5	1	1	0.7759	0.7765	0.08	4.6703	4.6704	0.00
5	3	1	0.7940	0.7941	0.01	4.6978	4.6979	0.00
5	5	1	0.7950	0.7950	0.00	4.6994	4.6994	0.00
5	1	3	0.9453	0.9453	0.00	4.9198	4.9196	0.00
5	3	3	0.9506	0.9506	0.00	4.9276	4.9276	0.00
5	5	3	0.9510	0.0000	0.00	4.9281	4.9281	0.00
5	1	4	0.9727	0.9727	0.00	4.9601	4.9600	0.00
5	3	4	0.9755	0.9755	0.00	4.9641	4.9640	0.00
5	5	4	0.9757	0.9757	0.00	4.9643	4.9643	0.00
10	1	0	0.5091	0.5102	0.22	9.1830	9.1837	0.01
10	3	0	0.5363	0.5365	0.03	9.2375	9.2377	0.00
10	5	0	0.5379	0.5379	0.00	9.2406	9.2406	0.00
10	1	1	0.7565	0.7569	0.05	9.5979	9.5977	0.00
10	3	1	0.7762	0.7762	0.01	9.6321	9.6321	0.00
10	5	1	0.7774	0.7774	0.00	9.6341	9.6341	0.00
10	1	3	0.9395	0.9395	0.00	9.9006	9.9004	0.00
10	3	3	0.9455	0.9455	0.00	9.9104	9.9104	0.00
10	5	3	0.9458	0.9458	0.00	9.9110	9.9110	0.00
10	1	4	0.9698	0.9698	0.00	9.9504	9.9503	0.00
10	3	4	0.9728	0.9728	0.00	9.9554	9.9554	0.00
10	5	4	0.9730	0.9730	0.00	9.9557	9.9557	0.00

2.2 General two-echelon repairable item systems

In this section the simple system from Section 2.1 will be extended to a more realistic one. The system will contain multiple bases and transport lines. Furthermore, the single servers that are used in the repair shops are replaced by multi-servers. These adjustments will make the analysis of the system more complicated. Nevertheless, the basic idea of the aggregation step will be the same.

2.2.1 The multi-base model with transportation

The system in this section consists of multiple bases, where the number of bases is denoted by L. A graphical representation of the system is given in Figure 2.9 for the case $L = 2$. As in the simple system described before, at base $l = 1, \ldots, L$ at most J_l machines are

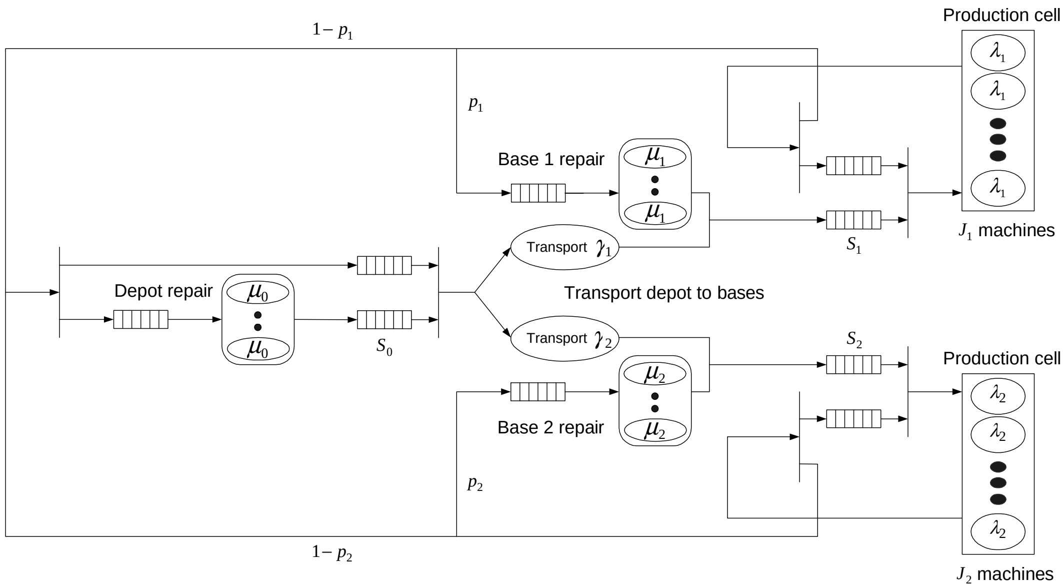


Figure 2.9: The multi-base two-echelon repairable item system for $L = 2$

operating in the production cell. The machines fail at exponential rate λ_l and are always replaced by a machine from the corresponding base stock (if available). Failed machines from base l are base-repairable with probability p_l and depot-repairable with probability $1 - p_l$. In contrast to the simple model described before, the repair shops are modeled as multi-servers. That is, at the repair shop of base $l = 1, \ldots, L$ R_l repairmen are working, each at exponential rate μ_l. At the depot repair shop R_0 repairmen are working at exponential rate μ_0. Consistent with the simple model S_l machines are dedicated to base l to act as spares and S_0 spare machines are dedicated to the depot. Broken machines at a certain base l that are base-repairable are sent to the base l repair shop. After repair they fill up the spares buffer at base l or, in case of a backlog at that base, become operational immediately. Broken machines from base l that are considered depot-repairable are sent to the depot repair shop. When depot spares are available, a spare is immediately sent to the stock of base l. In case there are no spares available a backlog occurs. Machines that have completed repair are sent to the base that has been waiting the longest. That is, an FCFS return policy is used. In this model the transportation from the depot to the bases is taken into account explicitly. The transport lines are modeled as ample servers with exponential service rate γ_l for the transport to base $l = 1, \ldots, L$. The transport from the bases to the depot is not taken into account.

As in the simple model, the synchronization queues at the bases can be replaced by normal queues as is depicted in Figure 2.10.

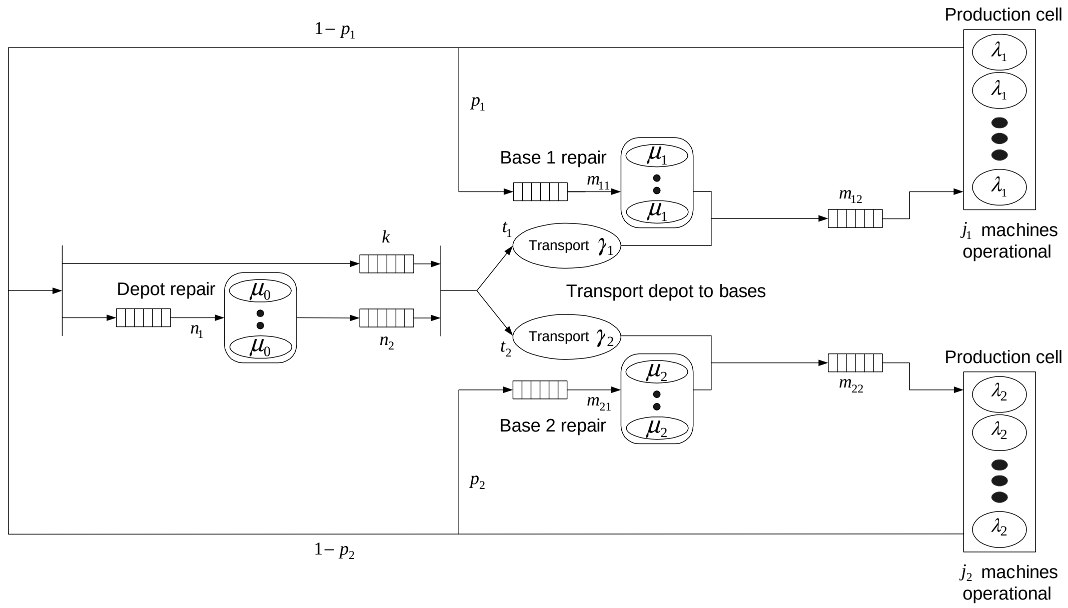


Figure 2.10: The modified multi-base two-echelon repairable item system for $L = 2$

The random vector $\underline{\mathbf{m}}_1 = (\underline{m}_{11}, \ldots, \underline{m}_{l1}, \ldots, \underline{m}_{L1})$ denotes the number of machines in base repair ($l = 1, \ldots, L$) and the random vector $\underline{\mathbf{m}}_2 = (\underline{m}_{12}, \ldots, \underline{m}_{l2}, \ldots, \underline{m}_{L2})$ denotes the number of spares at the bases ($l = 1, \ldots, L$). The random variable $\underline{n}_1$ stands for the number of machines in depot repair and $\underline{n}_2$ is the number of spare machines at the depot. The vector $\underline{\mathbf{k}}_0 = (\underline{k}_{01}, \ldots, \underline{k}_{0l}, \ldots, \underline{k}_{0L})$ denotes the backorders at the depot, originating from base l ($l = 1, \ldots, L$). The total number of backorders at the depot equals $\underline{k} = \sum_{l=1}^{L} \underline{k}_{0l}$. The machines in transit to the bases are given by the random vector $\underline{\mathbf{t}} = (\underline{t}_1, \ldots, \underline{t}_l, \ldots, \underline{t}_L)$ and the numbers of machines operating in the production cells are expressed in vector $\underline{\mathbf{j}} = (\underline{j}_1, \ldots, \underline{j}_l, \ldots, \underline{j}_L)$. For each base, the sum of the number of machines in base stock and the number of machines operating in the production cell is denoted in the random vector $\underline{\mathbf{b}} = (\underline{b}_1, \ldots, \underline{b}_l, \ldots, \underline{b}_L)$, where $\underline{b}_l = \underline{m}_{l2} + \underline{j}_l$.

As a result of the operating inventory control policies, the following equations must hold for $\underline{n}_1 = n_1$, $\underline{n}_2 = n_2$, $\underline{\mathbf{k}}_0 = \mathbf{k}_0$, $\underline{k} = k$, $\underline{\mathbf{t}} = \mathbf{t}$, $\underline{\mathbf{m}}_1 = \mathbf{m}_1$, $\underline{\mathbf{m}}_2 = \mathbf{m}_2$ and $\underline{\mathbf{j}} = \mathbf{j}$:

$$n_1 + n_2 - k = S_0, \tag{2.16}$$

$$n_2 \cdot k = 0, \tag{2.17}$$

and for $\quad l = 1, 2, \ldots, L:$

$$k_{0l} + t_l + m_{l1} + m_{l2} + j_l = S_l + J_l, \tag{2.18}$$

$$m_{l2} \cdot (J_l - j_l) = 0. \tag{2.19}$$

From these relations it follows immediately that $\mathbf{k}_0$, n_1, $\mathbf{t}$ and $\mathbf{m}_1$ completely determine the state of the system. Therefore, the system can be modeled as a continuous time Markov chain with state description $(\mathbf{k}_0, n_1, \mathbf{t}, \mathbf{m}_1)$.

Remark 2.2.1
In the vector that denotes the number of backorders originating from the bases, $\mathbf{k}_0 = (k_{01}, k_{02}, \ldots, k_{0L})$, it is not taken into account that the order of the backorders matters. Since an FCFS return policy is assumed, this order should be known. Nevertheless, in this model all states with similar numbers of backorders per base, are aggregated into one state. This aggregation step will not have a big influence on the results, but it will considerably simplify the analysis.

2.2.2 Approximation

In correspondence with the simple model as described in Section 2.1 a similar aggregation step is performed to tackle this extended model. Once more, all states with $0 \leq n_1 \leq S_0$ are aggregated into one state. Let the steady state probabilities for the original model be described by $P(\mathbf{k}_0, n_1, \mathbf{t}, \mathbf{m}_1) = P(\underline{\mathbf{k}}_0 = \mathbf{k}_0, \underline{n}_1 = n_1, \underline{\mathbf{t}} = \mathbf{t}, \underline{\mathbf{m}}_1 = \mathbf{m}_1)$ and for the new model be described by $\bar{P}(\mathbf{k}_0, k, \mathbf{t}, \mathbf{m}_1) = \bar{P}(\underline{\mathbf{k}}_0 = \mathbf{k}_0, \underline{k} = k, \underline{\mathbf{t}} = \mathbf{t}, \underline{\mathbf{m}}_1 = \mathbf{m}_1)$. The aggregation step is performed as follows

$$\bar{P}(\mathbf{0}, 0, \mathbf{t}, \mathbf{m}_1) = \sum_{n_1=0}^{S_0} P(\mathbf{0}, n_1, \mathbf{t}, \mathbf{m}_1), \tag{2.20}$$

$$\bar{P}(\mathbf{k}_0, k, \mathbf{t}, \mathbf{m}_1) = P(\mathbf{k}_0, S_0 + k, \mathbf{t}, \mathbf{m}_1). \tag{2.21}$$

The aggregated system can be described by $(\mathbf{k}_0, k, \mathbf{t}, \mathbf{m}_1)$. Furthermore, because $\underline{k} = \sum_{l=1}^{L} \underline{k}_{0l}$ the state space can also be described by $(\mathbf{k}_0, \mathbf{t}, \mathbf{m}_1)$.

Define q as before, that is q is the conditional probability that an arriving request at the depot cannot be fulfilled immediately, given that there are no other requests waiting. In a formula it says $q = P(\underline{n}_1 = S_0 \mid \underline{n}_1 \leq S_0)$. So, given there is no backlog at the depot, an arriving request has to wait with probability q. The waiting time depends on the number of spares already in the queue. The first spare that finishes repair will fulfill the just arrived request. With probability $1 - q$ spares are available and the arriving request does not have to wait. This aggregated network is depicted as a Typical-server Closed Queuing Network in Figure 2.11. The depot repair shop is modeled as a typical server. In case of no backlog $(k = 0)$ the service rate equals infinity with probability $1-q$ and equals $min(S_0, R_0)\mu_0$ with probability q. In all other cases $(k > 0)$ the service rate equals $min(k + S_0, R_0)\mu_0$.

To determine q Norton's theorem is used once more. As in Subsection 2.1.3 each base (the transport line, the base repair shop and the production cell) in the TCQN is replaced by a state dependent server. To determine the service rate of this state dependent server, each base-part of the network is short circuited and its throughput is calculated. This throughput operates as the service rate of the state dependent server. The new network with the state dependent servers and the short circuited networks are depicted in Figure 2.12.

Now consider the original model as shown in Figure 2.10. Machines visit the repair shop with the same throughput rates $TH_1(i)$ and $TH_2(i)$ as they visit the typical server in the

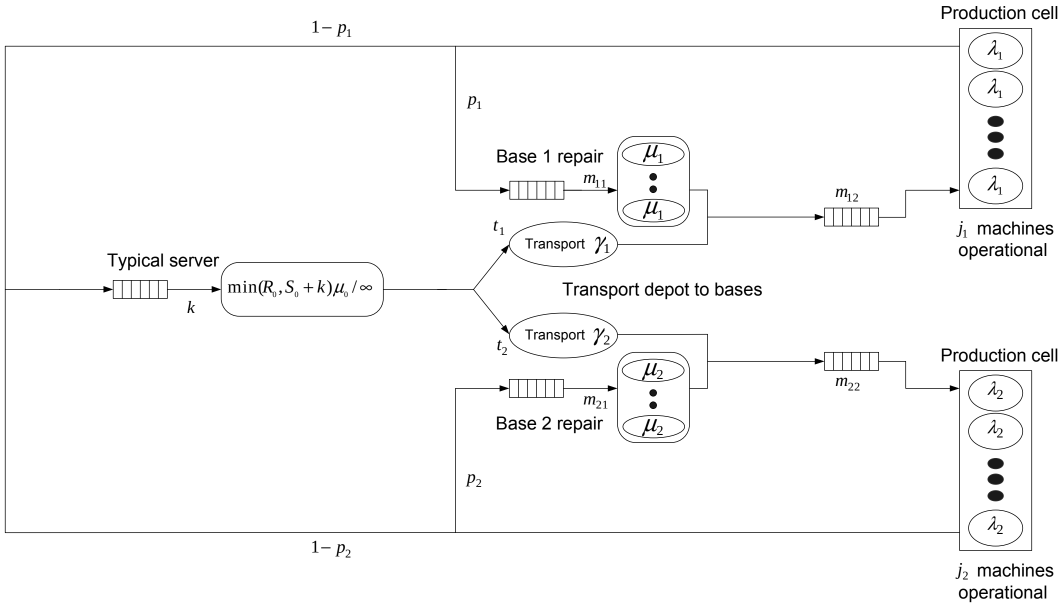


Figure 2.11: The Typical-server Closed Queuing Network

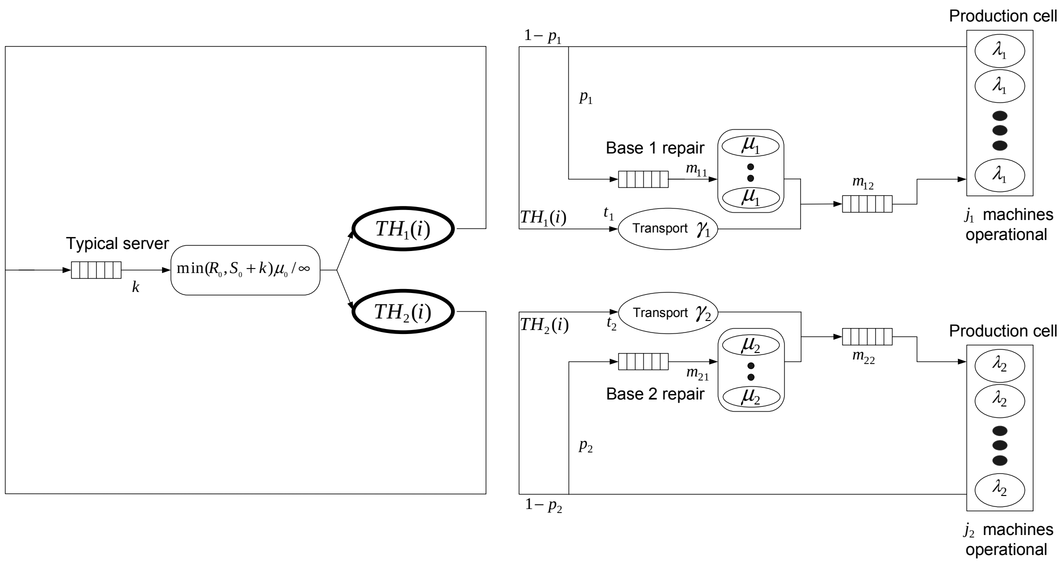


Figure 2.12: The new network with state dependent servers (left graph) and the short circuited networks (right graphs)

TCQN-model above. See Figure 2.13.

Once again the evolution of $\underline{n}_1$ can be described as a birth-death process. The (approximated) transition diagram for $n_1 = 0, \ldots, S_0$ is given in Figure 2.14. Let $TH_l(i)$ be the throughput of the subnetwork replacing base l ($l = 1, \ldots, L$) with i jobs present. As in the simple model only the behavior for $\underline{n}_1 \leq S_0$ needs to be studied to determine q. Take

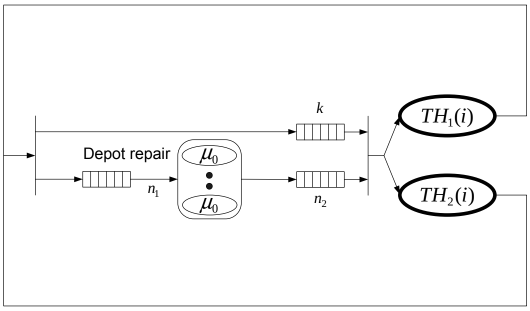


Figure 2.13: The original network with state dependent servers

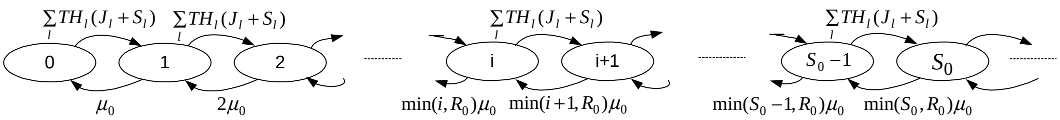


Figure 2.14: Transition diagram for n_1

$\delta = \sum_l TH_l(J_l + S_l)/\mu_0$, then

$$P(\underline{n}_1 = n_1) = \delta^{n_1} \frac{1}{\prod_{k=1}^{n_1} min(k, R_0)} P(\underline{n}_1 = 0) \quad \text{for} \quad n_1 = 0, \dots, S_0 \tag{2.22}$$

and

$$\begin{aligned} q &= \frac{P(\underline{n}_1 = S_0)}{P(\underline{n}_1 \leq S_0)} = \frac{P(\underline{n}_1 = S_0)}{\sum_{n_1=0}^{S_0} P(\underline{n}_1 = n_1)} = \frac{\delta^{S_0} \frac{1}{\prod_{k=1}^{S_0} min(k,R_0)} P(\underline{n}_1 = 0)}{\sum_{n_1=0}^{S_0} \delta^{n_1} \frac{1}{\prod_{k=1}^{n_1} min(k,R_0)} P(\underline{n}_1 = 0)} \\ &= \frac{\delta^{S_0} \frac{1}{\prod_{k=1}^{S_0} min(k,R_0)}}{\sum_{n_1=0}^{S_0} \delta^{n_1} \frac{1}{\prod_{k=1}^{n_1} min(k,R_0)}}. \end{aligned} \tag{2.23}$$

The throughputs can be obtained by applying a standard MDA algorithm (see Buzacott and Shanthikumar [5]) on the short circuited product form networks as shown in Figure 2.12.

The steady state marginal probabilities as well as the main performance measures for the aggregated system can be obtained by using an adapted Multi-Class Marginal Distribution Analysis algorithm (see Buzacott and Shanthikumar [5] for ordinary Multi-Class MDA). To see this, introduce tokens of class l with $l = 1, \dots, L$ that either represent machines present at base l (in the production cell, in the base repair shop, in the base stock or in transit to this base) or represent requests to the depot stock emerging from a failure of a machine at base l that cannot be repaired locally. Recall that machines that have to be repaired in the depot repair shop in fact lose their identity, i.e. after completion they are placed in the depot stock, from which they can in principle be shipped to any arbitrary base. However, the request arriving jointly with that broken machine at the depot, maintains its identity, meaning that it is matched with the first spare machine available, after which the combination is transported to the base the request originated from. Therefore, a token can be seen as connected to a machine as long as that machine is at the base (in any status) and connected with the corresponding request as soon as the machine is sent to the depot. This request matches with an available machine from stock (which generally is different from the one sent to the depot, unless $S_0 = 0$) and the combination returns to the base that generated the request. Hence, in this way, a multi-class network arises in a natural way.

The adapted algorithm is given below. An important aspect of an MDA algorithm is the computation of the expected sojourn time in the stations. Since the depot repair shop is modeled as a typical server, the standard sojourn time as described in [5] will not do for this station. As denoted before, in case of no backlog ($k = 0$) the service rate equals infinity with probability $1 - q$ and equals $min(S_0, R_0)\mu_0$ with probability q. In all other cases ($k > 0$) the service rate equals $min(k + S_0, R_0)\mu_0$. The expected sojourn time of an arriving request is the time it takes until all requests in front of it (k) are fulfilled and the request itself is fulfilled. That is, the time until $k + 1$ machines come out of repair. In case $k = 0$ with probability $1 - q$ the sojourn time equals 0 because a spare fulfills the request. These adaptations to the sojourn time reveal themselves in the algorithm in step 4.
Another adaptation to the ordinary algorithm is found in step 6. The transition rates from the states with 0 machines in depot repair to the states with 1 machine in depot repair now equal q times the throughput, instead of just the throughput.

Algorithm 2.2.2
The depot repair shop is defined as station 0 and all other stations are defined as station li, where l denotes the number of the base ($l = 1, \dots, L$) and i denotes the specific station associated with that base. The production cell is denoted by $i = b$, the base repair shop by $i = m$ and the transport line from the depot to the base by $i = t$.
Let $V_j^{(r)}$ be the visit ratio of station j for class r type machines. Let z denote the number of machines in the system and $\boldsymbol{z} = (z_1, \dots, z_r, \dots, z_L)$ the vector denoting the state that indicates the number of machines per class. The marginal probability that y machines are in station j, given vector $\boldsymbol{z}$ is denoted by $p_j(y \mid \boldsymbol{z})$. The expected sojourn time for type r machines arriving at station j given that $\boldsymbol{z}$ machines are wandering through the system is given by $EW_j^{(r)}(\boldsymbol{z})$ and $TH_j^{(r)}(\boldsymbol{z})$ denotes the throughput of type r machines given state $\boldsymbol{z}$. The algorithm is executed as follows:

1. *(Initialization) For $l = 1, \dots, L$ set $V_0^{(l)} = 1$, $V_{lb}^{(l)} = \frac{1}{1-p_l}$, $V_{lm}^{(l)} = \frac{p_l}{1-p_l}$ and $V_{lt}^{(l)} = 1$. For $l = 1, \dots, L$, $r = 1, \dots, L$, $r \neq l$, $i \in \{b, m, t\}$ set $V_{li}^{(r)} = 0$. Set $z = 0$ and $p_j(0 \mid \boldsymbol{0}) = 1$ for $j \in \bigcup_l \{lb, lm, lt\} \cup \{0\}$.*

2. *z:=z+1.*

3. *For all states $\boldsymbol{z} \in \{\boldsymbol{z} \mid \sum_{l=1}^{L} z^{(l)} = z$ and $z^{(l)} \leq J_l + S_l\}$ execute steps 4 through 6.*

4. *Compute the sojourn times for $l = 1, \dots, L$ for which $z^{(l)} > 0$ from:*

$$
\begin{aligned}
EW_0^{(l)}(\boldsymbol{z}) &= \sum_{k=1}^{z-1} \frac{k+1}{min(R_0, S_0 + k + 1)\mu_0} p_0(k \mid \boldsymbol{z} - \boldsymbol{e}_l) \\
&+ \frac{q}{min(R_0, S_0 + 1)\mu_0} p_0(0 \mid \boldsymbol{z} - \boldsymbol{e}_l),
\end{aligned}
$$

$$\begin{aligned}
EW_{lb}^{(l)}(\boldsymbol{z}) &= \sum_{b_l=J_l}^{z-1} \frac{b_l - J_l + 1}{J_l \lambda_l} p_{lb}(b_l \mid \boldsymbol{z} - \boldsymbol{e}_l) + \frac{1}{\lambda_l}, \\
EW_{lm}^{(l)}(\boldsymbol{z}) &= \sum_{m_{l1}=R_l}^{z-1} \frac{m_{l1} - R_l + 1}{R_l \mu_l} p_{lm}(m_{l1} \mid \boldsymbol{z} - \boldsymbol{e}_l) + \frac{1}{\mu_l}, \\
EW_{lt}^{(l)}(\boldsymbol{z}) &= \frac{1}{\gamma_l}.
\end{aligned}$$

5. *Compute* $TH_0^{(l)}(\boldsymbol{z})$ *for* $l = 1, \ldots, L$ *if* $z^{(l)} > 0$ *from:*

$$TH_0^{(l)}(\boldsymbol{z}) = \frac{z^{(l)}}{V_0^{(l)} EW_0^{(l)} + \sum_{i \in \{b,m,t\}} V_{li}^{(l)} EW_{li}^{(l)}}$$

and if $z^{(l)} = 0$ *then* $TH_0^{(l)}(\boldsymbol{z}) = 0$. *Compute* $TH_{li}^{(l)}(\boldsymbol{z})$ *for* $l = 1, \ldots, L$ *and* $i \in \{b, m, t\}$ *from:*

$$TH_{li}^{(l)}(\boldsymbol{z}) = V_{li}^{(l)} TH_0^{(l)}(\boldsymbol{z}).$$

6. *Compute the marginal probabilities for all stations from:*

$$\begin{aligned}
\mu_0 \, min(R_0, S_0 + 1) \, p_0(1 \mid \boldsymbol{z}) &= \sum_{l=1}^{L} TH_0^{(l)}(\boldsymbol{z}) \, q \, p_0(0 \mid \boldsymbol{z} - \boldsymbol{e}_l), \\
\mu_0 \, min(R_0, S_0 + k) \, p_0(k \mid \boldsymbol{z}) &= \sum_{l=1}^{L} TH_0^{(l)}(\boldsymbol{z}) \, p_0(k - 1 \mid \boldsymbol{z} - \boldsymbol{e}_l) \quad \textit{for } k = 2, \ldots, z,
\end{aligned}$$

and for $l = 1, \ldots, L$ *from:*

$$\begin{aligned}
\lambda_l \, min(J_l, b_l) \, p_{lb}(b_l \mid \boldsymbol{z}) &= TH_{lb}^{(l)}(\boldsymbol{z}) \, p_{lb}(b_l - 1 \mid \boldsymbol{z} - \boldsymbol{e}_l) \quad \textit{for } b_l = 1, \ldots, z, \\
\mu_l \, min(R_l, m_{l1}) \, p_{lm}(m_{l1} \mid \boldsymbol{z}) &= TH_{lm}^{(l)}(\boldsymbol{z}) \, p_{lm}(m_{l1} - 1 \mid \boldsymbol{z} - \boldsymbol{e}_l) \quad \textit{for } m_{l1} = 1, \ldots, z, \\
\gamma_l \, t_l \, p_{lt}(t_l \mid \boldsymbol{z}) &= TH_{lt}^{(l)}(\boldsymbol{z}) \, p_{lt}(t_l - 1 \mid \boldsymbol{z} - \boldsymbol{e}_l) \quad \textit{for } t_l = 1, \ldots, z.
\end{aligned}$$

Compute $p_j(0 \mid \boldsymbol{z})$ *for* $j \in \bigcup_l \{lb, lm, lt\} \cup \{0\}$ *from:*

$$p_j(0 \mid \boldsymbol{z}) = 1 - \sum_{y=1}^{z} p_j(y \mid \boldsymbol{z}).$$

7. *If* $z = \sum_{l=1}^{L} J_l + S_l$ *then stop; else go to step 2.*

With the adapted Multi-Class MDA algorithm presented above, the marginal probabilities of the system as well as the throughputs and the sojourn times can be approximated. From these, various performance measures can be computed. In the next subsection some results obtained by the algorithm will be compared with results from simulation.

2.2.3 Results

In this subsection results obtained by the adapted Multi-Class MDA algorithm from the previous subsection will be presented. They will be compared to results obtained by simulation. For each base we are interested in the availability, that is the probability that the maximum number of machines is operating in the production cell. For base l this is denoted by A_l for $l = 1, \ldots, L$. Furthermore we are interested in the expected number of machines operating in the production cell, denoted by $E\underline{j}_l$ for base $l = 1, \ldots, L$. For $l = 1, \ldots, L$ the performance measures can be computed by

$$\begin{aligned} A_l &= P(\underline{j}_l = J_l) = P(\underline{b}_l \geq J_l) = P(\underline{k}_{0l} + \underline{m}_{l1} \leq S_l), \qquad (2.24) \\ E\underline{j}_l &= E(J_l - [\underline{k}_{0l} + \underline{m}_{l1} - S_l]^+) \\ &= \sum_{k_{0l}, m_{l1}} (J_l - [k_{0l} + m_{l1} - S_l]^+)\, P(k_{0l}, m_{l1}). \qquad (2.25) \end{aligned}$$

In Table 2.2 and Tables B.3 and B.4 in Appendix B, the parameter settings for some representative test problems are given. It is obvious that a large number of input parameters is required to specify a given problem. This makes it difficult to vary these parameters in a totally systematic manner. In Albright [2] it is shown that traffic intensities are good indicators of whether a system will work well (minimal backorders) and are better indicators than the stock levels. Therefore we selected most of the test problem parameter settings by selecting values of the traffic intensities, usually well less than 1, and then selecting parameters to achieve these traffic intensities. For the base l repair facility, the traffic intensity ρ_l is defined as

$$\rho_l = J_l \lambda_l p_l / R_l \mu_l, \qquad (2.26)$$

the maximum failure rate divided by the maximum repair rate. Similarly, the depot traffic intensity ρ_0 is defined as

$$\rho_0 = \sum_{l=1}^{L} J_l \lambda_l (1 - p_l) / R_0 \mu_0. \qquad (2.27)$$

The results are given in Table 2.3 and Table B.5 in Appendix B. To obtain the simulation results, a simulation model was built in EM-Plant. In Appendix A more information about these simulations is given. The simulation leads to 95 % confidence intervals. To compare the approximations with the simulation results, the deviation from the approximation to the midpoint of the confidence interval is calculated. These percentage deviations are given as well.

From the results it can be concluded that the approximations are extremely accurate. The maximum deviation is well less than 1 % and all approximating values lie within the confidence intervals. Furthermore, all types of problems exhibited similar levels of accuracy.

Table 2.2: Parameter settings for test problems multi-base model with transportation (1)

Problem	L	J_l	S_l S_0	λ_l	μ_l μ_0	R_l R_0	p_l	γ_l	ρ_l ρ_0
1	2	10	2	1	10	1	0.5	∞	0.5
		10	2	1	10	1	0.5	∞	0.5
			1		20	1			0.5
2	2	5	2	1	5	1	0.5	∞	0.5
		5	2	1	5	1	0.5	∞	0.5
			1		10	1			0.5
3	2	5	2	1	5	2	0.5	∞	0.25
		5	2	1	5	2	0.5	∞	0.25
			1		10	2			0.25
4	2	5	2	1	5	1	0.5	10	0.5
		5	2	1	5	1	0.5	10	0.5
			1		10	1			0.5
5	2	5	2	1	5	1	0.5	2	0.5
		5	2	1	5	1	0.5	2	0.5
			1		10	1			0.5
6	2	5	2	1	5	2	0.5	2	0.25
		5	2	1	5	2	0.5	2	0.25
			1		10	2			0.25
7	2	5	2	1	1	5	0.5	2	0.5
		5	2	1	1	5	0.5	2	0.5
			1		2	5			0.5
8	2	5	2	1	1	5	0.5	2	0.5
		5	2	1	1	5	0.5	2	0.5
			7		2	5			0.5
9	2	5	5	1	3	1	0.5	∞	0.83
		5	5	1	3	1	0.5	∞	0.83
			5		6	1			0.83
10	2	5	5	1	5	1	0.5	∞	0.5
		5	5	1	5	1	0.5	∞	0.5
			5		10	1			0.5

Table 2.3: Results for test problems from Table (2.2)

Problem	$A_l sim$	$A_l appr$	% dev	$E\underline{j}_l sim$	$E\underline{J}_l appr$	% dev
1	(0.8529,0.8563)	0.8542	0.05	(9.7533,9.7615)	9.7562	0.01
	(0.8505,0.8559)	0.8542	0.11	(9.7489,9.7611)	9.7562	0.01
2	(0.8638,0.8750)	0.8683	0.13	(4.7957,4.8161)	4.8043	0.03
	(0.8636,0.87210	0.8683	0.05	(4.7983,4.8116)	4.8043	0.01
3	(0.9695,0.9714)	0.9701	0.04	(4.9626,4.9655)	4.9633	0.02
	(0.9689,0.9709)	0.9701	0.02	(4.9617,4.9649)	4.9633	0.00
4	(0.8311,0.8403)	0.8353	0.04	(4.7461,4.7640)	4.7543	0.02
	(0.8274,0.8346)	0.8353	0.51	(4.7399,4.7549)	4.7543	0.15
5	(0.6548,0.6639)	0.6605	0.18	(4.4542,4.4737)	4.4672	0.07
	(0.6583,0.6652)	0.6605	0.18	(4.4610,4.4753)	4.4672	0.02
6	(0.7490,0.7539)	0.7514	0.00	(4.6463,4.6545)	4.6521	0.04
	(0.7458,0.7529)	0.7514	0.27	(4.6416,4.6538)	4.6521	0.09
7	(0.2938,0.3008)	0.2978	0.17	(3.6284,3.6497)	3.6445	0.15
	(0.2949,0.3001)	0.2978	0.09	(3.6352,3.6529)	3.6445	0.01
8	(0.3781,0.3883)	0.3800	0.83	(3.8866,3.9096)	3.8907	0.19
	(0.3779,0.3836)	0.3800	0.19	(3.8855,3.9000)	3.8907	0.05
9	(0.8165,0.8361)	0.8234	0.34	(4.6622,4.7032)	4.6770	0.12
	(0.8142,0.8304)	0.8234	0.13	(4.6582,4.6941)	4.6770	0.02
10	(0.9854,0.9894)	0.9875	0.01	(4.9785,4.9851)	4.9817	0.00
	(0.9874,0.9894)	0.9875	0.09	(4.9815,4.9851)	4.9817	0.03

2.3 Conclusions

In this chapter we have analyzed a closed loop two-echelon repairable item system with a fixed number of items circulating in the network. The exact analysis of a Markov chain model for this system is difficult to handle. Therefore, we aggregated a number of states and adjusted some rates to obtain a special near-product-form solution, in accordance with [4]. The new system can be observed as a Typical-server Closed Queuing Network (TCQN). An adapted Multi-Class Marginal Distribution Analysis algorithm is developed to compute the marginal probabilities. From these marginal probabilities several performance measures can be obtained, such as the availability and the expected number of machines operating in the production cells. Numerical results show that the approximations are extremely accurate, when compared to simulation results.

However, a disadvantage of the adapted Multi-Class Marginal Distribution Analysis algorithm is the computational slowness. Especially for large systems with multiple bases, many machines and large inventories, the algorithm is not very fast. Further research might lead to improvements. Further aggregation steps may speed up the system evaluation considerably, however inevitably at the cost of some accuracy.

The model considered is quite a realistic model. Nevertheless, it could be more realistic by including transport from the bases to the depot and to allow for more complicated networks in the repair facilities. In the model described in this chapter, each repair shop is modeled as a multi-server. An interesting extension to this, is to consider the repair facility to be a job shop and model it as a limited capacity open queuing network, as has been done in [4] for the case of an open multi-echelon repairable item system. Then, it is easy to include transport to the depot repair facility as just an additional ample server node in the job shop.

Chapter 3

Closed loop two-indenture repairable item systems

In this chapter closed loop two-indenture repairable item systems will be discussed. As in the previous chapter, it is not possible to find an exact algebraic expression for the steady state probabilities. An attempt will be made to find reasonable approximations. From the steady state probabilities several performance measures can be obtained.
The first section contains previous research by Daryanto, Van Ommeren and Zijm. It is an approximation method for a simple two-indenture system that contains machines that consist of only one type of components. In the second section, this research will be extended to a model for machines that consist of two types of components. Both sections provide numerical results to determine the accuracy of the approximation methods. In the last section of this chapter, conclusions will be drawn and several remarks will be made.

3.1 Analysis of a single type two-indenture system with single server facilities[1]

In this section a simplified two-indenture repairable item system is discussed. It contains machines with only one type of critical components. The purpose of presenting such a simple system is to explain how a slight modification turns this system into a near-product form network that can be completely analyzed. In the next section the system will be extended to a system with machines consisting of two types of critical components.

3.1.1 The single type model

Consider the system as depicted in Figure 3.1. The system represents a single base, containing a production cell, a repair shop and an assembly facility. In the production cell a maximum of J machines are ideally working properly. Each machine may fail at Poisson rate λ. Such a failure is always due to the failure of exactly one component. For this simple model it is assumed that all critical components are of the same type. Assume that in addition to the working machines there are another S_0 ready-for-use machines to act as spares. When a machine fails, it is, ideally, immediately replaced by such a ready-for-use machine. In case there are no spares available (because these have been used to replace

[1] The research presented in this section is the outcome of previous research by A. Daryanto, J.C.W. van Ommeren and W.H.M. Zijm.

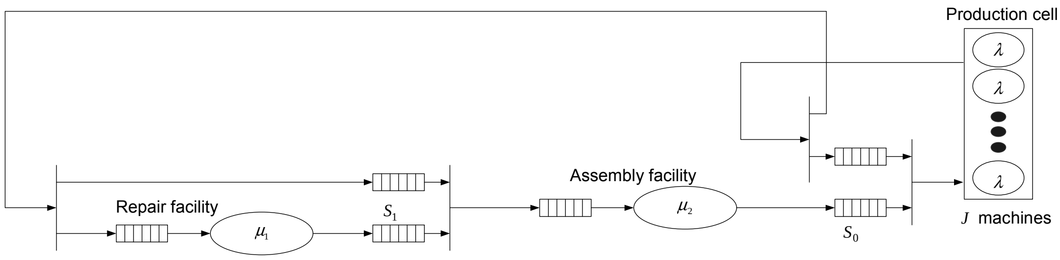


Figure 3.1: The two-indenture repairable item system with a single critical component

other machines that failed), a backorder occurs. The broken machine is disassembled and for this simple model we assume that this can be done in negligible time. The disassembled machine is matched with a spare component and assembled again in the assembly facility. In principle there are S_1 spare components, but in practice this number can be lower, because components can be in or awaiting repair. If there are no spare components available when needed, a backlog occurs. The failed component is sent to the repair shop. In the repair shop, components are repaired in FCFS order and after repair they fill up the components stock. In case of a backlog, a repaired component is immediately matched with a disassembled machine and sent to the assembly facility. The assembly facility and the repair shop are both modeled as single servers and operate at exponential service rate μ_2 and μ_1 respectively.

In Figure 3.1 the matching of a request and a spare machine is modeled as a synchronization queue. As in the two-echelon model described in Chapter 2, this synchronization queue can be replaced by a normal queue where machines are waiting to be moved into the production cell. This leads to the model in Figure 3.2.

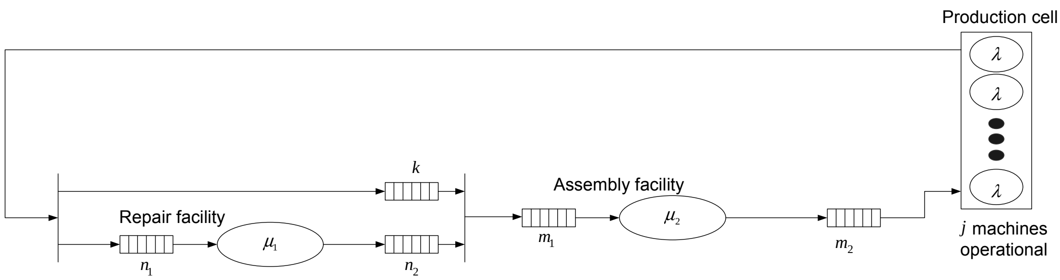


Figure 3.2: The modified two-indenture repairable item system with a single critical component

The number of machines operating in the production cell is denoted by the random variable $\underline{j}$ and the number of machines in base stock is denoted by the random variable $\underline{m_2}$. The sum of these is denoted by $\underline{b} = \underline{j} + \underline{m_2}$. The random variable $\underline{n_1}$ denotes the number of components in the repair facility and $\underline{n_2}$ denotes the number of spare components. A possible backlog of components is denoted by the random variable $\underline{k}$. The number of disassembled machines and matching components waiting to be or being assembled in the assembly facility is presented by the random variable $\underline{m_1}$.

As a result of the operating inventory control policies we immediately observe that, for $\underline{n}_1 = n_1$, $\underline{n}_2 = n_2$, $\underline{k} = k$, $\underline{m}_1 = m_1$, $\underline{m}_2 = m_2$ and $\underline{j} = j$:

$$n_1 + n_2 - k = S_1, \tag{3.1}$$
$$k \cdot n_2 = 0, \tag{3.2}$$
$$k + m_1 + m_2 + j = J + S_0, \tag{3.3}$$
$$(J - j) \cdot m_2 = 0, \tag{3.4}$$

where Equations 3.2 and 3.4 follow from the fact that it is impossible to have a backlog and to have spares available at the same place at the same time. From these relations it follows immediately that n_1 and m_1 completely determine the state of the system. The values of n_2 and k follow from (3.1) and (3.2) and the values of m_2 and j follow from (3.3) and (3.4). Because of this, the system can be modeled as a continuous time Markov chain with state description (n_1, m_1). The corresponding transition diagram is depicted in Figure 3.3.

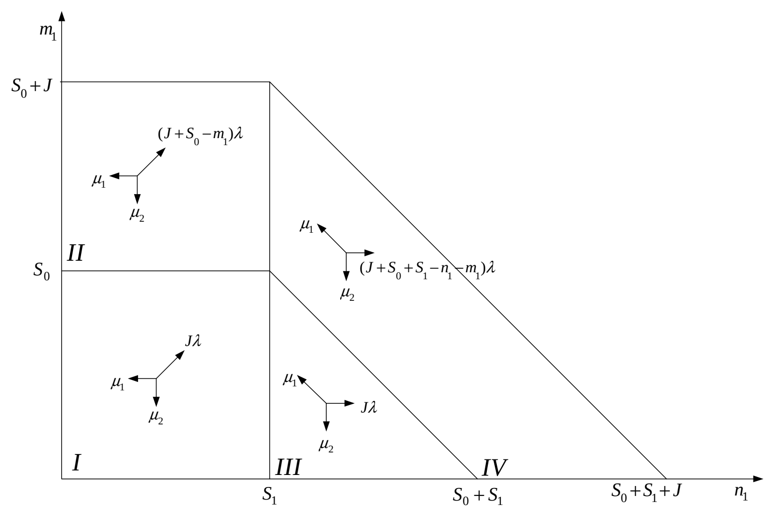


Figure 3.3: Transition diagram for state description (n_1, m_1)

Let $P(n_1, m_1) = P(\underline{n}_1 = n_1, \underline{m}_1 = m_1)$ be the steady state probability of being in state (n_1, m_1). Similar to Section 2.1 this steady state probability can be found by solving the global balance equations of the system. These can be deduced from the transition diagram. However, as stated before, for large systems with many machines and components the computational effort to find such an exact solution will become prohibitive.

In the next section an aggregation step will be applied on the state space in order to create a near-product form network, just like we did in Chapter 2. The near-product form network can be analyzed as if it is a product form network and algebraic expressions for the steady state probabilities can be obtained. Of course these expressions give approximations and the accuracy of these must be examined.

3.1.2 Approximation

As in the previous chapter, an aggregation step is performed on the state space to simplify the analysis. The aggregated system is described through the states (k, m_1). All states

(n_1, m_1) with $n_1 = 0, \ldots, S_1$ are aggregated into one state $(0, m_1)$. Denote the steady state probabilities for the new model by $\tilde{P}$ then the following holds for any m_1:

$$\tilde{P}(\underline{k} = 0, \underline{m}_1 = m_1) = \sum_{n_1=0}^{S_1} P(\underline{n}_1 = n_1, \underline{m}_1 = m_1), \tag{3.5}$$

$$\tilde{P}(\underline{k} = k, \underline{m}_1 = m_1) = P(\underline{n}_1 = S_1 + k, \underline{m}_1 = m_1). \tag{3.6}$$

The transition diagram corresponding to this state description is displayed in Figure 3.4. The transition rates are the same as the transition rates in Figure 3.3, except for states with

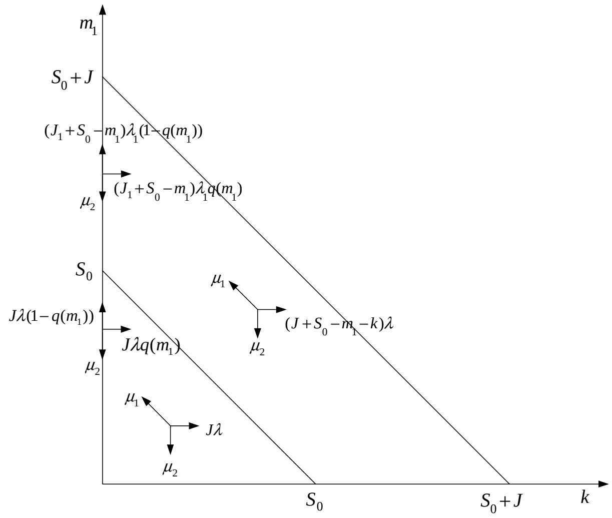


Figure 3.4: Transition diagram for state description (k, m_1)

$k = 0$. For these states the rates are adjusted. Let $q(m_1)$ be the steady state probability that an arriving request for a spare component at the repair facility has to wait, given that it finds no other waiting requests in front of it ($k = 0$) and $\underline{m}_1 = m_1$. Hence, given the (aggregated) state $(0, m_1)$, with probability $1-q(m_1)$ an arriving request for a component is satisfied immediately because spares are available and the state changes into $(0, m_1+1)$. The transition rate from $(0, m_1)$ to $(0, m_1 + 1)$ equals $j\,\lambda\,(1 - q(m_1))$, where $j = J - (m_1 - S_0)^+$. With probability $q(m_1)$ no spares are available and the state changes into $(1, m_1)$. The transition rate from $(0, m_1)$ to $(1, m_1)$ equals $j\ \lambda_1\ q(m_1)$, where $j = J - (m_1 - S_0)^+$. To determine $q(m_1)$ one needs

$$q(m_1) = P(\underline{n}_1 = S_1 \mid \underline{n}_1 \le S_1, \underline{m}_1 = m_1). \tag{3.7}$$

Since the problem is that we do not know the steady state distribution $P(n_1, m_1)$, we approximate the $q(m_1)$'s by their weighted average, similar to Chapter 2. We focus on the conditional probability q defined by

$$q = \sum_{m_1} q(m_1)\ P(\underline{m}_1 = m_1 \mid \underline{n}_1 \le S_1) = P(\underline{n}_1 = S_1 \mid \underline{n}_1 \le S_1) \tag{3.8}$$

and for every m_1 we replace $q(m_1)$ in the transition diagram by this q. In the next subsection will be explained how a reasonable approximation for this q can be obtained by means of an application of Norton's theorem, similar to Subsection 2.1.3.

Lemma 3.1.1
The steady state probabilities for the model with state description (k, m_1) and transition rates as denoted in Figure 3.4 with $q(m_1)$ replaced by arbitrary q have a product form.

Proof. To find the steady state probabilities, consider both the original model in Figure 3.2 and the alternative model in Figure 3.5. In Figure 3.5 the repair shop with synchronization

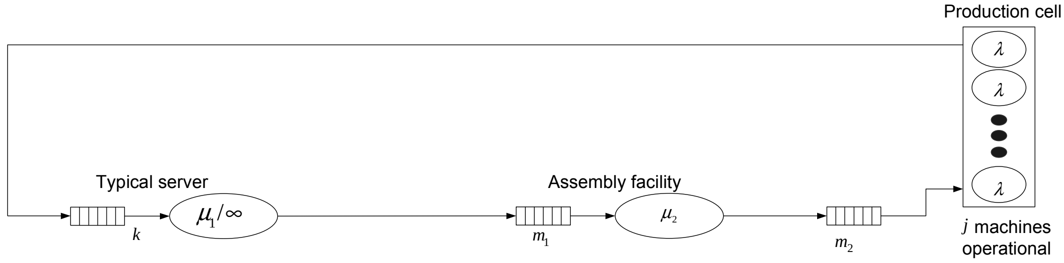


Figure 3.5: Typical-server Closed Queuing Network (TCQN)

queue is replaced by a typical server. For jobs that find the server idle the server has infinite service rate with probability $1-q$ (the case spare components are available) and service rate μ_1 with probability q (the case no spare components are available). Let $\underline{b}$ be the random variable equal to $\underline{m}_2 + \underline{j}$, then by looking at the system with the typical server, and conditioning on the fact that the network contains exactly $J + S_0$ jobs, it can be verified that the following expression for $\tilde{P}(\underline{k} = k, \underline{m}_1 = m_1, \underline{b} = b)$ satisfies the balance equations of the TCQN:

$$\tilde{P}(k, m_1, b) = \begin{cases} G'\, q \left(\frac{1}{\mu_1}\right)^k \left(\frac{1}{\mu_2}\right)^{m_1} \frac{\left(\frac{1}{\lambda}\right)^b}{J!J^{b-J}}, & b > J, k > 0, \\ G'\, q \left(\frac{1}{\mu_1}\right)^k \left(\frac{1}{\mu_2}\right)^{m_1} \frac{\left(\frac{1}{\lambda}\right)^b}{b!}, & b \leq J, k > 0, \\ G' \left(\frac{1}{\mu_2}\right)^{m_1} \frac{\left(\frac{1}{\lambda}\right)^b}{J!J^{b-J}}, & b > J, k = 0, \\ G' \left(\frac{1}{\mu_2}\right)^{m_1} \frac{\left(\frac{1}{\lambda}\right)^b}{b!}, & b \leq J, k = 0, \end{cases} \tag{3.9}$$

with $k + m_1 + b = J + S_0$ and G' the normalization constant. □

Expressed in terms of the state variables (k, m_1), this result immediately leads to:

Lemma 3.1.2
The steady state distribution for the aggregate model is given by

$$\tilde{P}(k, m_1) = \begin{cases} \frac{Gq}{J!J^{S_0-k-m_1}} \left(\frac{\lambda}{\mu_1}\right)^k \left(\frac{\lambda}{\mu_2}\right)^{m_1}, & k + m_1 \leq S_0, k > 0, \\ \frac{Gq}{(S_0+J-k-m_1)!} \left(\frac{\lambda}{\mu_1}\right)^k \left(\frac{\lambda}{\mu_2}\right)^{m_1}, & k + m_1 > S_0, k > 0, \\ \frac{G}{J!J^{S_0-m_1}} \left(\frac{\lambda}{\mu_2}\right)^{m_1}, & m_1 \leq S_0, k = 0, \\ \frac{G}{(S_0+J-m_1)!} \left(\frac{\lambda}{\mu_2}\right)^{m_1}, & m_1 > S_0, k = 0, \end{cases} \tag{3.10}$$

with G the normalization constant.

This lemma gives an explicit expression for the steady state probabilities. For large systems it may be difficult to calculate the normalization constant G. By applying MDA however, the required performance measures can be obtained directly.

As in the previous chapter, the results presented so far hold true for any value of $q \in [0, 1]$. Yet, it is expected that a good approximation will be obtained by using a q that corresponds to the interpretation as given in (3.8). In the next subsection Norton's theorem will be used to find a q with a meaningful interpretation that gives good results.

3.1.3 Applying Norton's theorem to approximate q

As in Chapter 2 a reasonable value for q can be determined by an application of Norton's theorem (see Harrison and Patel [10] for an overview). Let us repeat the basic idea of Norton's theorem. The theorem states that a product form network can be analyzed by replacing a subnetwork by state dependent servers where, for each number of customers i, the service rate is obtained by treating the subnetwork as a closed queuing network with appropriate rerouting mechanisms. This is called short circuiting. The joint distribution of the number of customers at all nodes outside the subnetwork remains unchanged when replacing the subnetwork by a state dependent server in this way.

To use this idea, consider the TCQN from Figure 3.5. The production cell and the assembly facility are taken apart and replaced by a state dependent server. The new network

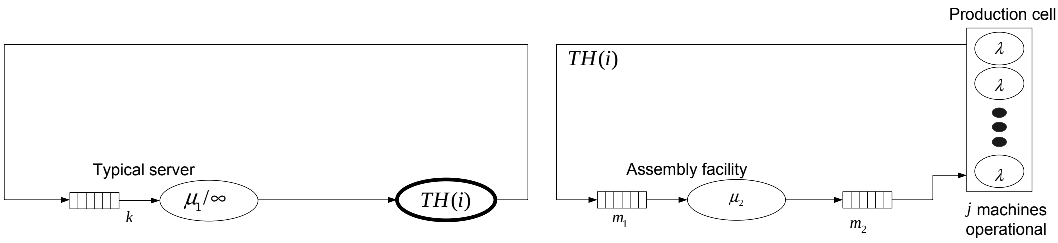


Figure 3.6: The new network with state dependent server (left graph) and the short circuited network (right graph)

with the state dependent server is displayed in Figure 3.6 (left graph). In order to find the service rates for this state dependent server, the original network is short circuited by setting the service rate at the typical server to infinity. This short circuited network is the right graph in Figure 3.6. The service rate for the new state dependent server with i jobs present is equal to the throughput of the short circuited network with i jobs present, denoted by $TH(i)$.

Now consider the original model as shown in Figure 3.2. Machines visit the repair shop with the same throughput rate $TH(i)$ as they visit the typical server in the TCQN-model above. See Figure 3.7.
We want to obtain q, the conditional probability that a request corresponding with a machine failure finds no spare parts in stock at the repair shop, although there was no backlog so far. That is, according to (3.8)

$$q = P(\underline{n}_1 = S_1 \mid \underline{n}_1 \leq S_1).$$

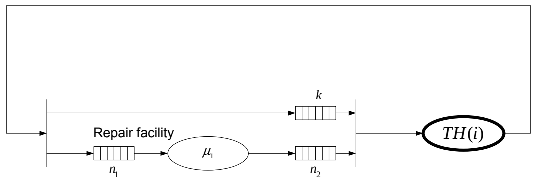


Figure 3.7: The original network with state dependent server

The evolution $\underline{n}_1 = n_1$, the number of components in the repair station, can be described as a birth-death process. The transition diagram, as obtained from Figure 3.7 is shown in Figure 3.8. Again, this is just an approximation due to the fact that Norton's theorem

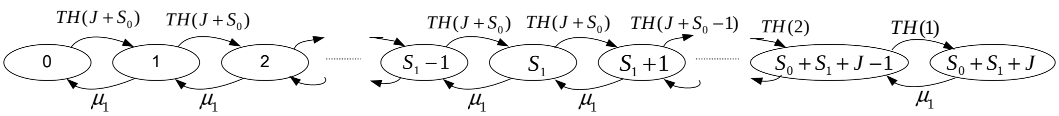


Figure 3.8: Transition diagram for n_1

is only valid for product form networks. In case $S_0 = 0$, we would have a product form network and the results would be exact.

From the diagram one can observe that

$$P(\underline{n}_1 = n_1)\, TH(J + S_0 - (n_1 - S_1)^+) = P(\underline{n}_1 = n_1 + 1)\, \mu_1 \tag{3.11}$$

for $n_1 = 0, \ldots, S_0 + S_1 + J - 1$. In principle one can derive an approximation of the distribution of $\underline{n}_1$ from this. However, by the definition of q, only the behavior for $\underline{n}_1 \leq S_0$ needs to be studied. For these states, the service rate of the state dependent server is equal to $TH(J + S_0)$. Let $\delta = TH(J + S_0)/\mu_1$. From (3.11) we observe that $P(\underline{n}_1 = n_1) = \delta^{n_1} P(\underline{n}_1 = 0)$ for $n_1 = 0, \ldots, S_1$ so

$$\begin{aligned} q &= \frac{P(\underline{n}_1 = S_1)}{P(\underline{n}_1 \leq S_1)} = \frac{\delta^{S_1} P(\underline{n}_1 = 0)}{\sum_{n_1=0}^{S_1} P(\underline{n}_1 = n_1)} = \frac{\delta^{S_1} P(\underline{n}_1 = 0)}{\sum_{n_1=0}^{S_1} \delta^{n_1} P(\underline{n}_1 = 0)} = \frac{\delta^{S_1}}{\frac{1-\delta^{S_1+1}}{1-\delta}} \\ &= \delta^{S_1} \frac{1-\delta}{1-\delta^{S_1+1}}. \end{aligned} \tag{3.12}$$

It remains to find the throughput of the short circuited network in Figure 3.6 (right graph) with $J + S_0$ jobs present. A simple observation reveals that $P(\underline{b} = b)\, min(b, J)\, \lambda = P(\underline{b} = b - 1)\, \mu_2$ for $b = 1, \ldots, J + S_0$ from which the steady state probabilities of $\underline{b}$ are immediately deduced. Moreover, the throughput satisfies

$$\begin{aligned} TH(J + S_0) &= \sum_{b=1}^{J+S_0} P(\underline{b} = b)\, min(b, J)\, \lambda \\ &= \mu_2\, (1 - P(\underline{b} = J + S_0)). \end{aligned} \tag{3.13}$$

We can determine q with (3.12) and (3.13). This q can be used to approximate the steady state distribution using (3.10) or using Marginal Distribution Analysis. Results of this approximation are presented in the next subsection.

3.1.4 Results

In this subsection two performance measures of the system are studied. As before, we consider the availability A, the probability that the maximum number of machines is working in the production cell. Furthermore, we consider the expected number of machines operating in the production cell ($E\underline{j}$). These are defined as follows:

$$A = P(\underline{j} = J) = P(\underline{b} \geq J) = P(\underline{k} + \underline{m}_1 \leq S_0), \tag{3.14}$$

$$E\underline{j} = E(J - [\underline{k} + \underline{m}_1 - S_0]^+) = \sum_{k,m_1} (J - [k + m_1 - S_0]^+)\, P(k, m_1), \tag{3.15}$$

To indicate the accuracy of the proposed approximations, the performance measures derived by the product form solution of the aggregated model are compared with exact results. These exact results are obtained by solving the balance equations.
The performance measures are computed for several values of J, S_0, S_1, λ, μ_1 and μ_2. The results are given in Table 3.1 and Table B.6 in Appendix B. The error percentages are also given. The numbers show that in these systems, the approximation gives an error of at most 5 %. Cases with only a small number of spares ($S_1 > 0$) in the system give the largest errors. For the case $S_1 = 0$ the results are exact.

3.2 Analysis of a two-type two-indenture system with single server facilities

In this section the single type model from Section 3.1 will be extended to a two-type model. A machine no longer consists of one critical component but consists of two critical components. This will make the analysis of the system much more difficult, as will be revealed below.

3.2.1 The two-type model

As in the simple system described before, the system in this section consists of a single base with a production cell where at most J machines can operate. These machines fail at exponential rate λ and are replaced by a machine from stock (if available). In contrast to the simple system, there is no longer just one type of critical components, but there are two types of critical components. Therefore, in the disassembling process, one also needs to determine which component has failed. With probability r_1 a component of type 1 has failed and with probability r_2 a component of type 2 has failed. It is assumed that only one component fails at a time and that a machine failure is always induced by a failure of a component of one of these types. Therefore $r_1 + r_2 = 1$. The disassembling process and the failure detection is performed at the disassembly and failure detection station, which is modeled as a single server and operates at exponential rate μ_0. From there the failed component and the remaining machine move separately to the repair shop. At the repair shop only one component can be repaired at a time, that is, the repair shop is also modeled as a single server. Components are repaired with exponential rate μ_1, independent of the component type. Repairs are performed in an FCFS order.
At the repair facility a number of spare components is kept in stock. There are S_1 spare components of type 1 and S_2 spare components of type 2. The remaining machine is matched with a spare component of the type the machine was missing and moves to the assembly

Table 3.1: Results for the single type two-indenture model, $\lambda = 1, \mu_1 = 2J, \mu_2 = J$

J	S_0	S_1	A_{exact}	A_{appr}	% error	$E\underline{j}_{exact}$	$E\underline{j}_{appr}$	% error
3	0	1	0.3248	0.3142	3.27	1.9173	1.8963	1.10
3	0	3	0.3452	0.3428	0.71	1.9596	1.9547	0.25
3	0	5	0.3461	0.3458	0.09	1.9615	1.9608	0.03
3	1	1	0.4899	0.4808	1.87	2.1854	2.1696	0.72
3	1	3	0.5127	0.5097	0.58	2.2257	2.2205	0.23
3	1	5	0.5142	0.5137	0.10	2.2284	2.2275	0.04
3	3	1	0.6620	0.6566	0.81	2.4623	2.4537	0.35
3	3	3	0.6777	0.6754	0.34	2.4880	2.4843	0.15
3	3	5	0.6791	0.6786	0.08	2.4903	2.4895	0.03
3	4	1	0.7121	0.7079	0.59	2.5423	2.5357	0.26
3	4	3	0.7245	0.7226	0.26	2.5623	2.5593	0.12
3	4	5	0.7257	0.7252	0.06	2.5643	2.5636	0.03
5	0	1	0.2625	0.2520	4.01	3.5108	3.4833	0.78
5	0	3	0.2836	0.2807	1.02	3.5719	3.5640	0.22
5	0	5	0.2848	0.2843	0.16	3.5754	3.5742	0.04
5	1	1	0.4176	0.4080	2.31	3.8312	3.8100	0.55
5	1	3	0.4415	0.4381	0.77	3.8869	3.8793	0.20
5	1	5	0.4433	0.4426	0.15	3.8911	3.8896	0.04
5	3	1	0.5953	0.5893	1.01	4.1923	4.1801	0.29
5	3	3	0.6126	0.6099	0.43	4.2282	4.2228	0.13
5	3	5	0.6142	0.6136	0.10	4.2316	4.2303	0.03
5	4	1	0.6502	0.6455	0.73	4.3026	4.2930	0.22
5	4	3	0.6642	0.6620	0.34	4.3311	4.3266	0.10
5	4	5	0.6656	0.6650	0.08	4.3339	4.3328	0.03
10	0	1	0.1933	0.1835	5.07	7.7578	7.7208	0.48
10	0	3	0.2131	0.2098	1.52	7.8470	7.8338	0.17
10	0	5	0.2145	0.2139	0.29	7.8536	7.8510	0.03
10	1	1	0.3277	0.3180	2.95	8.1461	8.1165	0.36
10	1	3	0.3512	0.3474	1.08	8.2258	8.2136	0.15
10	1	5	0.3532	0.3523	0.23	8.2326	8.2300	0.03
10	3	1	0.5017	0.4951	1.32	8.6351	8.6167	0.21
10	3	3	0.5202	0.5171	0.59	8.6886	8.6799	0.10
10	3	5	0.5220	0.5212	0.14	8.6939	8.6918	0.02
10	4	1	0.5602	0.5548	0.96	8.7969	8.7820	0.17
10	4	3	0.5756	0.5730	0.46	8.8403	8.8330	0.08
10	4	5	0.5772	0.5765	0.12	8.8448	8.8429	0.02

facility. In case there is no spare component of a certain type available when needed, a backlog occurs. The assembling process is modeled as a single server and operates with exponential service rate μ_2. Machines and components are assembled at the assembly facility on an FCFS basis. This means, both the incomplete machine and the separate component need to be present before joining the queue.
In the production cell ideally J machines are operational. When a failure occurs, the failed machine is replaced by a spare machine. In principle there are S_0 machines available as spares. However, in practice the expected base stock will be smaller because some machines will be in use because a broken down machine needed to be replaced. The system is presented in Figure 3.9.

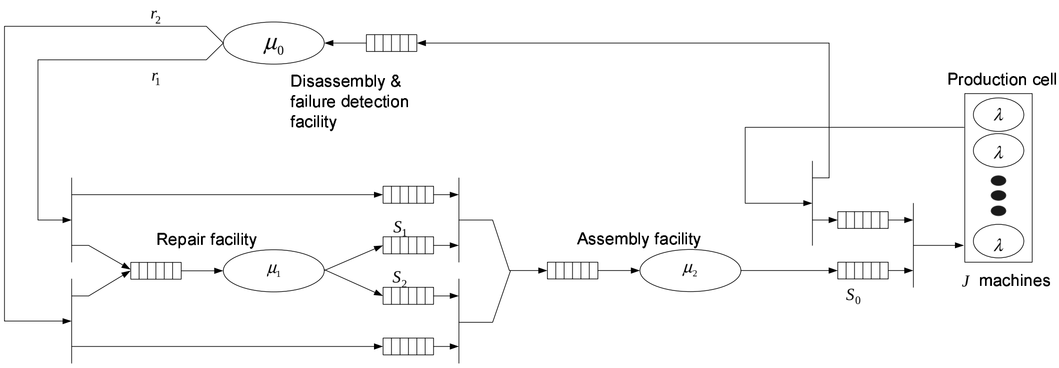


Figure 3.9: The two-indenture repairable item system with two types of components

For the sake of convenience, the disassembly and failure detection station will from now on be left out of consideration. That is, it is assumed that $\mu_0 = \infty$. This will make the presentation of the analysis simpler. However, including the station will not make the analysis itself more difficult. Therefore, in Section 3.3 will be explained how this station can be included in the model and how this effects the analysis and results.
As in the simple model, the synchronization queue at the base can be replaced by a normal queue. This, and the previous assumption lead to the modified model in Figure 3.10.

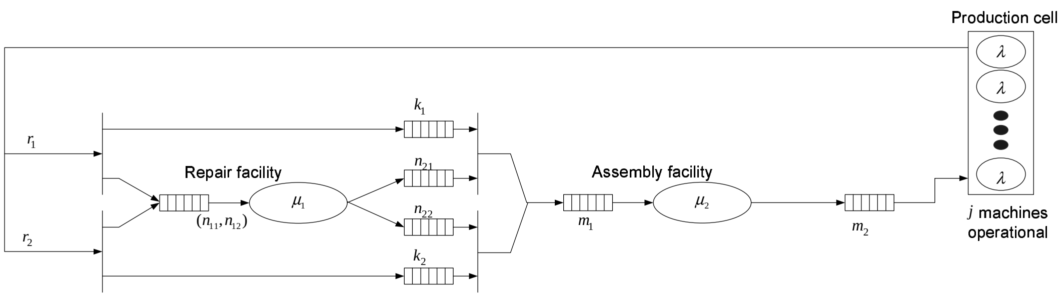


Figure 3.10: The modified two-indenture repairable item system with two types of components

The random variable $\underline{j}$ denotes the number of machines operating in the production cell and the random variable $\underline{m}_2$ denotes the number of machines in stock. The sum of these is denoted by the random variable $\underline{b}$, that is $\underline{b} = \underline{j} + \underline{m}_2$. The number of components in or awaiting repair is presented by the random vector $\underline{\mathbf{n}}_1 = (\underline{n}_{11}, \underline{n}_{12})$ and the random vector $\underline{\mathbf{n}}_2 = (\underline{n}_{21}, \underline{n}_{22})$ denotes the number of spare components at the repair facility. The backlog of components at the repair facility is given in the random vector $\underline{\mathbf{k}} = (\underline{k}_1, \underline{k}_2)$ and the number of incomplete machines and matching components that is actually waiting to be or being assembled at the assembly facility is denoted by the random variable $\underline{m}_1$.

As a result of the operating inventory control policies, the following equations must hold for $\underline{\mathbf{n}}_1 = \mathbf{n}_1$, $\underline{\mathbf{n}}_2 = \mathbf{n}_2$, $\underline{\mathbf{k}} = \mathbf{k}$, $\underline{m}_1 = m_1$, $\underline{m}_2 = m_2$ and $\underline{j} = j$:

$$\begin{aligned}
n_{11} + n_{21} - k_1 &= S_1, && (3.16)\\
n_{12} + n_{22} - k_2 &= S_2, && (3.17)\\
k_1 \cdot n_{21} &= 0, && (3.18)\\
k_2 \cdot n_{22} &= 0, && (3.19)\\
m_1 + m_2 + k_1 + k_2 + j &= J + S_0, && (3.20)\\
(J - j) \cdot m_2 &= 0. && (3.21)
\end{aligned}$$

From these relations it follows immediately that $\mathbf{n}_1$ and m_1 completely determine the state of the system. Therefore, the system can be modeled as a continuous time Markov chain with state description $(\mathbf{n}_1, m_1)$.

Remark 3.2.1
The attentive reader will have noticed that in the previous description of the state space it is tacitly assumed that only the number of components (per type) in the repair facility matters. In reality also the order in which the components are present in the queue is important. The assumption has been made to drastically simplify the model. However, it cannot be justified so easily. It might have a big influence on the results. In Subsection 3.2.6 more comments will be made on this assumption.

3.2.2 Approximation

As in all previous models, an aggregation step is carried out to simplify the analysis. The state description $(\mathbf{n}_1, m_1)$ can be replaced by the alternative state description $(\mathbf{k}, m_1)$. This aggregate model is fully expressed in terms of the former one by defining the following sets for every $\mathbf{k}$:

$$\begin{aligned}
\mathcal{D}_0(\mathbf{k}) &= \{l \mid k_l = 0\}, && (3.22)\\
\mathcal{T}(\mathbf{k}) &= \{\mathbf{n}_1 \mid n_{1d} \le S_d \ \ for \ \ d \in \mathcal{D}_0(\mathbf{k}),\ n_{1d} = S_d + k_d \ \ for \ \ d \notin \mathcal{D}_0(\mathbf{k})\}. && (3.23)
\end{aligned}$$

For any $\mathbf{k}$ and m_1, the steady state probabilities of the partially aggregated model satisfy the following relation which is directly implied by the aggregation:

$$\bar{P}(\mathbf{k}, m_1) = \sum_{\mathbf{n}_1 \in \mathcal{T}(\mathbf{k})} P(\mathbf{n}_1, m_1). \tag{3.24}$$

In Figure 3.11 the aggregate model is depicted as Typical-server Closed Queuing Network. The repair shop and its inventories are replaced by a typical server with state dependent service rates.
As in Subsection 2.2.2 this TCQN can be solved as a (near) product form network by using a Marginal Distribution Analysis algorithm. Again, an adaptation to the original algorithm

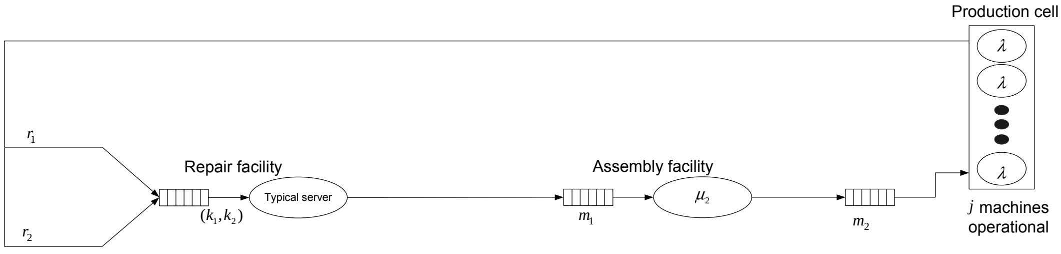


Figure 3.11: Typical-server Closed Queuing Network (TCQN) to represent the aggregated two-indenture model

(as described in [5]) must be made.
Let the production cell and the spare machines stock together be the first station, denoted as base station. This station has J servers, that all operate at exponential service rate λ. Let the repair shop be the second station and let the assembly facility be the third station. Both stations are single server stations. The second station has a state dependent service rate and the third station operates at exponential service rate μ_2. Tokens are circulating between these stations. An (adapted) MDA algorithm is applied on these tokens and these three stations.

Let a token represent the basic part of a machine, that is, a machine excluding two critical components, one of each type. In the base station (production cell plus spare machines stock), a token is always connected to the two components (one of each type). The sojourn time of the token in the base station is the time until the token becomes operational (waiting time in the spare machines stock) plus the time until one of the critical components fails (operational time in the production cell). When a failure occurs, the machine is sent to the disassembly and failure detection station (which has infinite service rate here). The failed component is separated (and sent to repair) and the remaining token and component are sent to the repair shop to be matched with a component of the type that had just failed. The sojourn time for the token in this station is the time until a matching component is present. In case a spare is available, this time equals zero. In case no spare component is available, this time is the time until a component of the required type finishes repair. Tokens arriving at the repair shop are served in an FCFS order. That is, in case of a backlog, all waiting requests are served before the new request. For example, if k_1 tokens in the queue are waiting for a type 1 component, a new token requiring a type 1 component will have a sojourn time in this station that equals the time until $k_1 + 1$ components of type 1 are repaired. Two things are important to note:

- The sojourn time in the repair shop depends on the component type. For instance, it is possible that there are spare components of type 1 available and there is a backlog of k_2 type 2 components. In that case the expected sojourn time for tokens requiring a type 1 component equals zero and the expected sojourn time for tokens requiring a type 2 component is the time it takes until $k_2 + 1$ type 2 components come out of repair.

- The sojourn time for a token requiring a certain component in the repair shop depends on the total number of components in the repair shop and not just on the number of components of the type you need. Repairs take place in an FCFS order. The consequence is that it is possible that a component of type 2 is being repaired, even though there are type 2 spares available and there is a backlog of type 1 components. By intuition, it is obvious that this is not very efficient. An alternative would be to include priority scheduling in the model. This will not be done here.

The token and accompanying component are matched with a repaired component of the type that was missing and are sent to the assembly facility. Here, the sojourn time equals the waiting time plus the service time required for the assembling process itself.

The adapted algorithm is given below. Because of the complicated nature of the sojourn times in the repair station, the variables $EW_l^a(\mathbf{k})$ are introduced for $l \in \{1,2\}$ to represent the expected sojourn time for an arriving token requiring a class l component, given there is a backlog of $\mathbf{k}$. Furthermore, because of the aggregation step described before, the variables $q_1(0,k_2)$ and $q_2(k_1,0)$ are introduced. The variable $q_1(0,k_2)$ denotes the probability that there are no spare components of type 1 available, given that there is no backlog of type 1 components and there is a backlog of k_2 type 2 components. Likewise, the variable $q_2(k_1,0)$ denotes the probability that there are no spare components of type 2 available, given that there is no backlog of type 2 components and there is a backlog of k_1 type 1 components. In the next subsections several procedures will be given to compute the $EW_l^a(\mathbf{k})$ for $l \in \{1,2\}$ and to compute the $q_1(0,k_2)$ and $q_2(k_1,0)$.
Since the visit ratios in the MDA algorithm are quite simple, these are no longer explicitly mentioned in the algorithm.

Algorithm 3.2.2
The base station (production cell plus spare machines stock) is denoted as station 0*, the repair shop as station* 1 *and the assembly facility as station* 2*. Let the expected sojourn time for tokens requiring type* l $(l \in \{1,2\})$ *components in station 1 given there are already* $\boldsymbol{k}$ *tokens present be denoted by* $EW_l^a(\boldsymbol{k})$*. Let* z *be the number of tokens in the system. The expected sojourn time for tokens arriving at station* $j \in \{0,2\}$ *given that* z *machines are in the system is given by* $EW_j(z)$ *and* $TH_j(z)$ *denotes the throughput of station* $j \in \{0,2\}$ *given* z*.*
For station 1 *the expected sojourn time for arriving tokens requiring a type* l *component is denoted by* $EW_{1l}(z)$ *and the throughput for tokens requiring a type* l *component is denoted by* $TH_{1l}(z)$*. The marginal probability that* y *tokens are in station* 0*, given the total number of tokens in the system* (z) *is denoted by* $p_0(y\,|\,z)$*. Similarly* $p_2(y\,|\,z)$ *denotes the probability that* y *tokens are in station* 2*, given that the total number of tokens in the system is* z*. In station* 1 *the probability of having* k_1 *requests for type* 1 *and* k_2 *requests for type* 2 *in the station while the total number of tokens in the system is* z *is denoted by* $p_1(k_1,k_2\,|\,z)$*. The* $q_1(0,k_2)$ *and the* $q_2(k_1,0)$ *are variables that are used to make a good approximation for the aggregation step described before. The algorithm is executed as follows:*

1. *(Initialization)*
   - *Determine the* $EW_l^a(\boldsymbol{k})$ *for all* $\boldsymbol{k}$ *for* $l \in \{1,2\}$ *by using the theory from the next subsections.*
   - *Determine the* $q_1(0,k_2)$ *for* $k_2 = 0,\ldots,J+S_0$ *and the* $q_2(k_1,0)$ *for* $k_1 = 0,\ldots,J+S_0$ *by using the theory from the next subsections.*
   - *Set* $p_0(0\,|\,0) = 1$*,* $p_1(\boldsymbol{0}\,|\,0) = 1$ *and* $p_2(0\,|\,0) = 1$*.*

- *Set* $z = 0$.

2. *z:=z+1.*

3. *Compute the expected sojourn times for all stations from:*

$$\begin{aligned} EW_0(z) &= \sum_{b=J}^{z-1} \frac{b-J+1}{J\lambda}\, p_0(b \mid z-1) + \frac{1}{\lambda}, \\ EW_{11}(z) &= \sum_{k_1=0}^{z-1} \sum_{k_2=0}^{z-1-k_1} EW_1^a(k_1,k_2)\, p_1(k_1,k_2 \mid z-1), \\ EW_{12}(z) &= \sum_{k_1=0}^{z-1} \sum_{k_2=0}^{z-1-k_1} EW_2^a(k_1,k_2)\, p_1(k_1,k_2 \mid z-1), \\ EW_2(z) &= \sum_{m_1=1}^{z-1} \frac{m_1}{\mu_2}\, p_2(m_1 \mid z-1) + \frac{1}{\mu_2}. \end{aligned}$$

4. *Compute* $TH_0(z)$ *from:*

$$TH_0(z) = \frac{z}{EW_0(z) + r_1 EW_{11}(z) + r_2 EW_{12}(z) + EW_2(z)}$$

and compute the throughputs for the other stations from:

$$\begin{aligned} TH_{11}(z) &= r_1 TH_0(z), \\ TH_{12}(z) &= r_2 TH_0(z), \\ TH_2(z) &= TH_0(z). \end{aligned}$$

5. *Compute the marginal probabilities* $p_0(b \mid z)$ *for* $b = 1, \ldots, z$ *from:*

$$\lambda\, min(J,b)\, p_0(b \mid z) = TH_0(z)\, p_0(b-1 \mid z-1)$$

and compute $p_0(0 \mid z)$ *from:*

$$p_0(0 \mid z) = 1 - \sum_{b=1}^{z} p_0(b \mid z).$$

Compute the marginal probabilities for station 1 *from:*

$$\begin{aligned} \mu_1\, p_1(1,0 \mid z) &= TH_{11}(z)\, q_1(0,0)\, p_1(0,0 \mid z-1), \\ \mu_1\, p_1(0,1 \mid z) &= TH_{12}(z)\, q_2(0,0)\, p_1(0,0 \mid z-1), \end{aligned}$$

for $z > 1$ *from:*

$$\begin{aligned} \mu_1\, p_1(1,1 \mid z) &= TH_{11}(z)\, q_1(0,1)\, p_1(0,1 \mid z-1) \\ &+ TH_{12}(z)\, q_2(1,0)\, p_1(1,0 \mid z-1), \\ \mu_1\, p_1(k_1,0 \mid z) &= TH_{11}(z)\, p_1(k_1-1,0 \mid z-1), \\ &\qquad \textit{for } k_1 = 2, \ldots, z, \\ \mu_1\, p_1(0,k_2 \mid z) &= TH_{12}(z)\, p_1(0,k_2-1 \mid z-1), \\ &\qquad \textit{for } k_2 = 2, \ldots, z, \end{aligned}$$

for $z > 2$ from:

$$\begin{aligned}
\mu_1\, p_1(k_1, 1 \mid z) &= TH_{11}(z)\, p_1(k_1 - 1, 1 \mid z - 1) \\
&+ TH_{12}(z)\, q_2(k_1, 0)\, p_1(k_1, 0 \mid z - 1), \\
&\qquad\qquad \textit{for } k_1 = 2, \ldots, z - 1, \\
\mu_1\, p_1(1, k_2 \mid z) &= TH_{11}(z)\, q_1(0, k_2)\, p_1(0, k_2 \mid z - 1) \\
&+ TH_{12}(z)\, p_1(1, k_2 - 1 \mid z - 1), \\
&\qquad\qquad \textit{for } k_2 = 2, \ldots, z - 1,
\end{aligned}$$

and for $z > 3$ from:

$$\begin{aligned}
\mu_1\, p_1(k_1, k_2 \mid z) &= TH_{11}(z)\, p_1(k_1 - 1, k_2 \mid z - 1) \\
&+ TH_{12}(z)\, p_1(k_1, k_2 - 1 \mid z - 1), \\
&\qquad\qquad \textit{for } k_1 = 2, \ldots, z - 2 \textit{ and } k_2 = 2, \ldots, z - k_1.
\end{aligned}$$

Compute the marginal probability $p_1(0, 0 \mid z)$ from:

$$p_1(0, 0 \mid z) = 1 - \sum_{\substack{k_1 = 0 \\ (k_1, k_2) \neq (0,0)}}^{z} \sum_{k_2=0}^{z-k_1} p_1(k_1, k_2 \mid z).$$

Compute the marginal probabilities $p_2(m_1 \mid z)$ for $m_1 = 1, \ldots, z$ from:

$$\mu_2\, p_2(m_1 \mid z) = TH_2(z)\, p_2(m_1 - 1 \mid z - 1)$$

and compute $p_2(0 \mid z)$ from:

$$p_2(0 \mid z) = 1 - \sum_{m_1=1}^{z} p_2(m_1 \mid z).$$

6. *If $z = J + S_0$ then stop; else go to step 2.*

3.2.3 Determining the values for $EW_l^a(\mathbf{k})$, $q_1(0, k_2)$ and $q_2(k_1, 0)$

In the previous subsection an adapted Marginal Distribution Analysis algorithm has been presented to obtain the marginal probabilities of the two-indenture system with two critical components. In this algorithm one uses $EW_l^a(\mathbf{k})$ for $l \in \{1, 2\}$, $q_1(0, k_2)$ and $q_2(k_1, 0)$. The $EW_l^a(\mathbf{k})$ with $l \in \{1, 2\}$ represent the expected sojourn time for an arriving token requiring a class l component. The variable $q_1(0, k_2)$ denotes the probability that there are no spare components of type 1 available, given that there is no backlog of type 1 components and there is a backlog of k_2 type 2 components. The variable $q_2(k_1, 0)$ is defined similarly.
In this subsection a procedure will be given to obtain reasonable values for these variables. In the next subsection this procedure will be simplified and in Subsection 3.2.5 we deal with a procedure that incorporates another aggregation step.
Here, it is assumed that $S_1 > 0$ and $S_2 > 0$. For the formulas for cases with $S_1 = 0$ and/or $S_2 = 0$ we refer to Appendix C.

3.2.3.1 Determining the marginal distribution for $\mathbf{n}_1$

In the upcoming paragraphs one needs to know the marginal steady state distribution for $\mathbf{n}_1 = (n_{11}, n_{12})$ to be able to find appropriate values for the expected sojourn times ($EW_l^a(\mathbf{k})$) and the $q_1(0, k_2)$ and $q_1(k_1, 0)$. Therefore, a method to obtain the marginal distribution for (n_{11}, n_{12}) will be presented in this paragraph.

The state space described by (n_{11}, n_{12}) is depicted in Figure 3.12. In this figure the transition rates are given. These transition rates are obtained as follows.

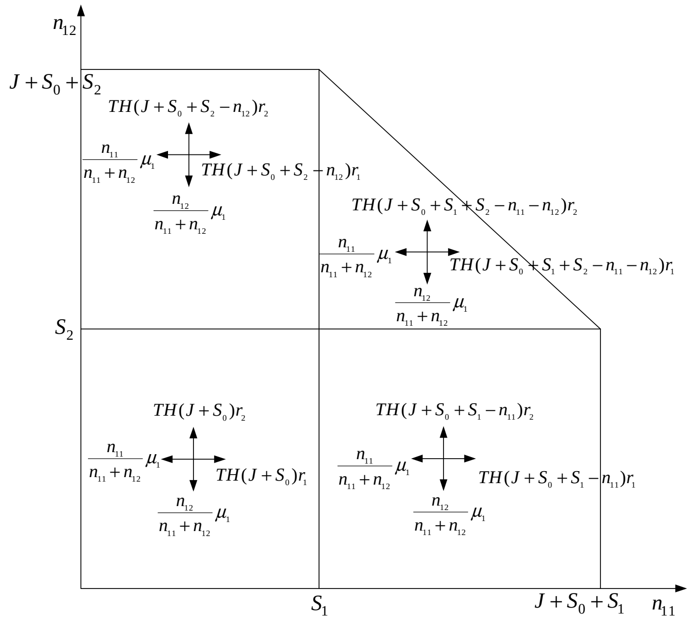


Figure 3.12: The state space (n_{11}, n_{12})

- The *down rates*, that is, the rates from (n_{11}, n_{12}) to $(n_{11} - 1, n_{12})$ and $(n_{11}, n_{12} - 1)$ are obtained by assuming that when n_{11} components of type 1 and n_{12} components of type 2 are in the repair shop, every possible order of the components is just as likely to occur. Therefore the probability that the first component to finish repair is of type 1 is $\frac{n_{11}}{n_{11}+n_{12}}$ and the probability that it is of type 2 is $\frac{n_{12}}{n_{11}+n_{12}}$. Multiplying these expressions with the service rate μ_1 leads to the transition rates as given in Figure 3.12.
  Note that the assumption that every possible order of the components takes place just as likely will lead to approximating transition rates. Some orders of the components might not even be possible in reality. However, including all possible orders explicitly in the model, will make it extremely complex and therefore unusable. See also Remark 3.2.1.

- The *up rates*, that is, the rates from (n_{11}, n_{12}) to $(n_{11} + 1, n_{12})$ and $(n_{11}, n_{12} + 1)$ are obtained by an application of Norton's theorem, as has been done before in Subsection 3.1.3. Norton's theorem states that a product form network can be analyzed by replacing a subnetwork by state dependent servers. For each number of customers i, the service rate is obtained by treating the subnetwork as a closed queuing network with appropriate rerouting mechanisms. The joint distribution of the number of customers at all nodes outside the subnetwork remains unchanged when replacing the subnetwork by a state dependent server in this way.

Consider the TCQN from Figure 3.11. The production cell and the assembly facility are taken apart and are replaced by a state dependent server. The new network

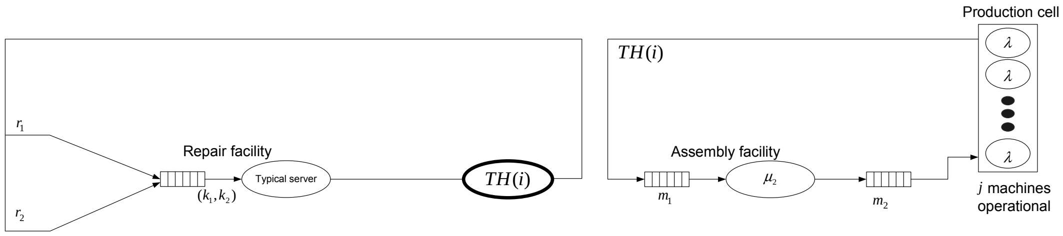


Figure 3.13: The new network with state dependent server and the short circuited network

with the state dependent server is displayed in Figure 3.13 (left graph). In order to find the service rates for this state dependent server, the original network is short circuited by setting the service rate at the repair facility to infinity. This short circuited network is also depicted in Figure 3.13 (right graph). Note that this short circuited network is the same as in the single type model, see Figure 3.6. The service rate for the new state dependent server with i jobs present is equal to the throughput of the short circuited network with i jobs present, denoted by $TH(i)$.

Consider the original model as shown in Figure 3.10. Machines visit the repair shop with the same throughput rate $TH(i)$ as they visit the typical server in the TCQN-model above. This is depicted in Figure 3.14.

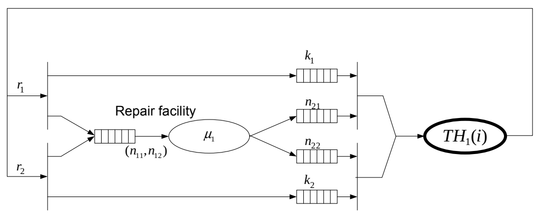


Figure 3.14: The original network with state dependent server

To find the transition rates for the state space (n_{11}, n_{12}) the value of i must be determined for each pair (n_{11}, n_{12}) by

$$i = J + S_0 - (n_{11} - S_1)^+ - (n_{12} - S_2)^+$$

and the $TH(i)$ must be computed and multiplied with either r_1 or r_2, as denoted in Figure 3.12.
The throughput $TH(i)$ of the short circuited network in Figure 3.13 (right graph) with i jobs present, where $i = 0, 1, \ldots, J + S_0$, can be obtained as follows. For

$i = 0, 1, \ldots, J + S_0$

$$TH(i) = \sum_{b=0}^{i} P(\underline{b} = b)\, min(b, J)\, \lambda. \tag{3.25}$$

A simple observation reveals that $P(\underline{b} = b)\, min(b, J)\, \lambda = P(\underline{b} = b)\, \mu_2$ for $b = 1, \ldots, i$ from which the steady state probabilities of $\underline{b}$, given i, are immediately deduced. Note that since Norton's theorem is only valid for product form networks and we are dealing with a near-product form network here, the obtained transition rates are approximations.

From the transition rates as given in Figure 3.12 the balance equations can be obtained. Solving these balance equations will lead to the marginal distribution for $\mathbf{n}_1 = (n_{11}, n_{12})$. This steady state distribution is denoted by $P_1(n_{11}, n_{12})$. In the next paragraphs this distribution will be used to obtain appropriate values for the expected sojourn times $(EW_l^a(\mathbf{k}))$ and the $q_1(0, k_2)$ and $q_2(k_1, 0)$.

3.2.3.2 The case $k_1 > 0$ and $k_2 > 0$

The case $k_1 > 0$ and $k_2 > 0$ means that there is a backlog of k_1 components of type 1 and of k_2 components of type 2. Since there are also S_1 spare components of type 1 in the repair shop, the total number of type 1 components in the repair shop is $k_1 + S_1$. Likewise, there are in total $k_2 + S_2$ components of type 2 in the repair shop. Nothing is known about the order in which these components are in the repair shop.

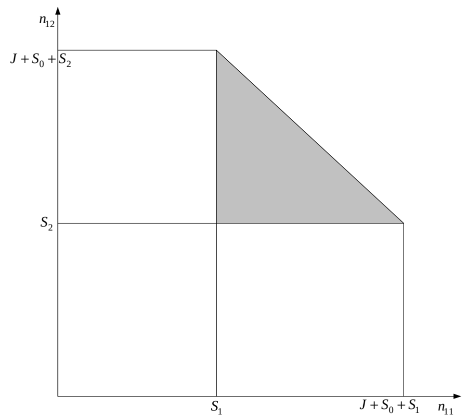


Figure 3.15: The case $k_1 > 0$ and $k_2 > 0$

When a token requiring a type 1 component arrives at the repair shop, it has to wait until $k_1 + 1$ type 1 components have come out of repair. The first k_1 type 1 components are matched with the k_1 tokens requiring a type 1 component that were already waiting and the $(k_1 + 1)$th type 1 component is matched with this token. The expected sojourn time in the repair shop for a token requiring a type 1 component, given that there is a backlog for k_1 components of type 1 and k_2 components of type 2, therefore equals the time until $k_1 + 1$ type 1 components are repaired, given that there are $k_1 + S_1$ type 1 components and $k_2 + S_2$ type 2 components in the repair shop.

Of course, at the same time at which the token arrives, also a broken type 1 component arrives. However, since an FCFS repair schedule has been assumed, this component will not interfere with the existing queue. Even though the order of the existing queue is not known, it is known that this component will end the queue. The same argument holds for all other broken components arriving at the repair shop while the token is waiting for a matching component. All these components will join the queue at the end and do not interfere with the existing queue.

Define for $k_1 > 0$, $k_2 > 0$, $S_1 > 0$ and $S_2 > 0$:

$$H_1(k_1 + S_1, k_2 + S_2, y) = \sum_{x=k_1+1}^{min(y,k_1+S_1)} \frac{\binom{k_1+S_1}{x}\binom{k_2+S_2}{y-x}}{\binom{k_1+k_2+S_1+S_2}{y}}, \tag{3.26}$$

as the probability that when y repairs have been performed at least $k_1 + 1$ components of type 1 have been repaired, given that there were $k_1 + S_1$ type 1 components and $k_2 + S_2$ type 2 components waiting to be or being repaired.
To obtain this equation the hypergeometric distribution is applied to obtain the probability that x type 1 components are repaired, given a total repair number of y. The $k_1 + S_1$ type 1 components are defined as 'successes' in the hypergeometric distribution while the total population is $k_1 + k_2 + S_1 + S_2$. The probabilities are summed over all possible values of $x \geq k_1+1$ to obtain the probability that at least k_1+1 type 1 components have been repaired.

The difference

$$H_1(k_1 + S_1, k_2 + S_1, y) - H_1(k_1 + S_1, k_2 + S_2, y - 1), \tag{3.27}$$

represents the probability that the yth repair is the $(k_1 + 1)$th repair of a type 1 component. Summing and weighing these probabilities over all possible y values leads to

$$EW_1^a(k_1, k_2) = \sum_{y=k_1+1}^{k_1+1+k_2+S_2} (H_1(k_1+S_1, k_2+S_2, y) - H_1(k_1+S_1, k_2+S_2, y-1))\frac{y}{\mu_1} \tag{3.28}$$

as the expected sojourn time for a token arriving at the repair shop and requiring a type 1 component.

Similarly, an expression for the expected sojourn time for a token arriving at the repair shop and requiring a type 2 component can be deduced. Define for $k_1 > 0$, $k_2 > 0$, $S_1 > 0$ and $S_2 > 0$:

$$H_2(k_1 + S_1, k_2 + S_2, y) = \sum_{x=k_2+1}^{min(y,k_2+S_2)} \frac{\binom{k_1+S_1}{y-x}\binom{k_2+S_2}{x}}{\binom{k_1+k_2+S_1+S_2}{y}}, \tag{3.29}$$

as the probability that when y repairs have been performed at least $k_2 + 1$ components of type 2 have been repaired, given that there were $k_1 + S_1$ type 1 components and $k_2 + S_2$ type 2 components waiting to be or being repaired.

For the expected sojourn time for a token arriving at the repair shop and requiring a type 2 component the following holds:

$$EW_2^a(k_1, k_2) = \sum_{y=k_2+1}^{k_1+S_1+k_2+1} (H_2(k_1+S_1, k_2+S_2, y) - H_2(k_1+S_1, k_2+S_2, y-1))\frac{y}{\mu_1}. \tag{3.30}$$

3.2.3.3 The case $k_1 = 0$ and $k_2 = 0$

The case $k_1 = 0$ and $k_2 = 0$ means that there is no backlog of any components and there might be spare components available. For the original system $\underline{n}_{11} \le S_1$ and $\underline{n}_{12} \le S_2$ hold, as is depicted in Figure 3.16.

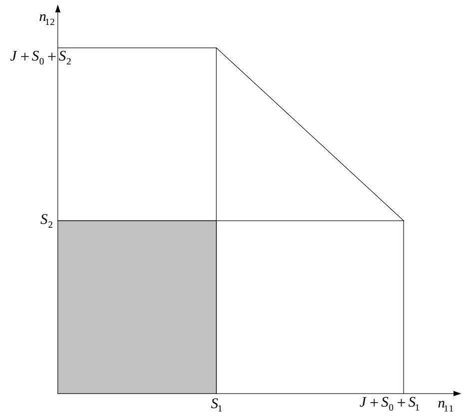


Figure 3.16: The case $k_1 = 0$ and $k_2 = 0$

Since this part of the state space is aggregated into one single state ($\underline{k}_1 = 0, \underline{k}_2 = 0$), also the expected sojourn times for each state in this part of the state space must be aggregated into one expected sojourn time for the state ($\underline{k}_1 = 0, \underline{k}_2 = 0$). Note that the expected sojourn time for a token requiring a type 1 component is zero when type 1 spare components are available, that is if $\underline{n}_{11} < S_1$. Therefore only the sojourn times for the cases with $\underline{n}_{11} = S_1$ need to be taken into consideration. Like wise for tokens requiring a type 2 component only the expected sojourn times for the cases with $\underline{n}_{12} = S_2$ need to be considered. With the marginal probabilities $P_1(n_{11}, n_{12})$ that have been computed in Paragraph 3.2.3.1 the weighted sum of the expected sojourn times per state can be taken to obtain the expected sojourn time for the aggregated state.

First of all, the steady state probabilities $P_1(n_{11}, n_{12})$ must be scaled to the sub state space $\mathcal{A} = \{(n_{11}, n_{12}) \mid n_{11} \le S_1 \text{ and } n_{12} \le S_2\}$. That is

$$\tilde{P}_1(n_{11}, n_{12}) = \tilde{G}\, P_1(n_{11}, n_{12}) \quad for \quad (n_{11}, n_{12}) \in \mathcal{A}, \tag{3.31}$$

where $\tilde{G}$ is chosen such that

$$\sum_{(n_{11}, n_{12}) \in \mathcal{A}} \tilde{P}_1(n_{11}, n_{12}) = 1. \tag{3.32}$$

The expected sojourn times can be deduced as follows.

Define for $S_1 > 0$, $S_2 > 0$ and $n_{12} = 0, 1, \ldots, S_2$:

$$H_3(S_1, n_{12}, y) = \sum_{x=1}^{min(y, S_1)} \frac{\binom{S_1}{x}\binom{n_{12}}{y-x}}{\binom{S_1+n_{12}}{y}}, \tag{3.33}$$

as the probability that when y repairs have been performed at least 1 component of type 1 has been repaired, given that there were S_1 spare components of type 1 awaiting to be or

being repaired (so no spare components were available to be assembled) and there were n_{12} type 2 components awaiting to be or being repaired.
Similar to the previously described case, for the expected sojourn time for a token arriving at the repair shop and requiring a type 1 component the following holds:

$$EW_1^a(0,0) = \sum_{n_{12}=0}^{S_2} \Big(\sum_{y=1}^{n_{12}+1} (H_3(S_1, n_{12}, y) - H_3(S_1, n_{12}, y-1)) \frac{y}{\mu_1} \Big) \tilde{P}_1(S_1, n_{12}). \tag{3.34}$$

As explained above, the weighted sum is taken over all (n_{11}, n_{12}) with $n_{11} = S_1$ and $n_{12} \le S_2$. There is no need to sum over the states with $n_{11} < S_1$ since the sojourn times for those states are zero, because spares are available. In that case, an arriving token is directly matched with a spare component and does not have to wait.

For tokens that require a type 2 component similar formulas apply. For $S_1 > 0$, $S_2 > 0$ and $n_{11} = 0, 1, \dots, S_1$ define:

$$H_4(n_{11}, S_2, y) = \sum_{x=1}^{min(y,S_2)} \frac{\binom{n_{11}}{y-x}\binom{S_2}{x}}{\binom{n_{11}+S_2}{y}}. \tag{3.35}$$

The expected sojourn time for a token arriving at the repair shop and requiring a type 2 component is

$$EW_2^a(0,0) = \sum_{n_{11}=0}^{S_1} \Big(\sum_{y=1}^{n_{11}+1} (H_4(n_{11}, S_2, y) - H_4(n_{11}, S_2, y-1)) \frac{y}{\mu_1} \Big) \tilde{P}_1(n_{11}, S_2). \tag{3.36}$$

In the adapted MDA algorithm as described in Section 3.2.2 also the variables $q_1(0, k_2)$ and $q_2(k_1, 0)$ with $k_1 = 0, 1, \dots, J + S_0$ and $k_2 = 0, 1, \dots, J + S_0$ are used. The variable $q_1(0, k_2)$ denotes the probability that there are no spare components of type 1 available, given that there is no backlog of type 1 components and there is a backlog of k_2 type 2 components. The definition of $q_2(k_1, 0)$ is similar.
The variables $q_1(0,0)$ and $q_2(0,0)$ can be deduced as follows:

$$q_1(0,0) = \frac{P_1(\underline{n}_{11} = S_1 \mid \underline{n}_{12} \le S_2)}{P_1(\underline{n}_{11} \le S_1 \mid \underline{n}_{12} \le S_2)} = \sum_{n_{12}=0}^{S_2} \tilde{P}_1(S_1, n_{12}), \tag{3.37}$$

$$q_2(0,0) = \frac{P_1(\underline{n}_{12} = S_2 \mid \underline{n}_{11} \le S_1)}{P_1(\underline{n}_{12} \le S_2 \mid \underline{n}_{11} \le S_1)} = \sum_{n_{11}=0}^{S_1} \tilde{P}_1(n_{11}, S_2). \tag{3.38}$$

The variables $q_1(0, k_2)$ and $q_2(k_1, 0)$ with $k_1 = 1, \dots, J + S_0$ and $k_2 = 1, \dots, J + S_0$ will be determined in the next paragraphs.

3.2.3.4 The case $k_1 > 0$ and $k_2 = 0$

The case $k_1 > 0$ and $k_2 = 0$ means that there is a backlog of k_1 type 1 components and there is no backlog of type 2 components. There might be spare type 2 components available. For the original system $n_{11} > S_1$ and $n_{12} \le S_2$ hold, as is depicted in Figure 3.17.
For each $n_{11} = k_1 + S_1$ the states (n_{11}, n_{12}) with $n_{12} \le S_2$ are aggregated into one state $(\underline{k}_1 = k_1, \underline{k}_2 = 0)$. The expected sojourn times for the aggregated states are therefore the

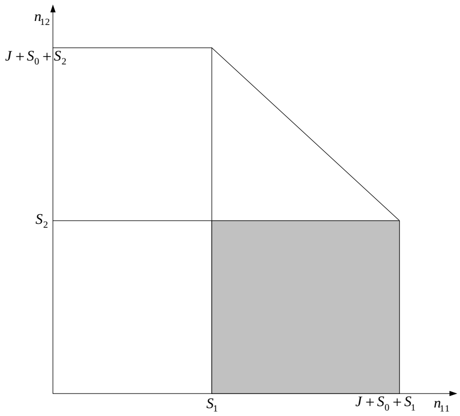


Figure 3.17: The case $k_1 > 0$ and $k_2 = 0$

weighted mean of the separate sojourn times. This weighted mean is taken by considering the marginal probabilities $P_1(n_{11}, n_{12})$ that have been computed in Paragraph 3.2.3.1. As in the previous paragraph these probabilities must be scaled for the current case. That is, for $n_{11} = S_1+1, \ldots, J+S_0+S_1$ and $n_{12} = 0, \ldots, S_2$:

$$\bar{P}_1(n_{11}, n_{12}) = \bar{G}\, P_1(n_{11}, n_{12}), \tag{3.39}$$

where $\bar{G}$ is chosen such that for $n_{11} = S_1+1, \ldots, J+S_0+S_1$:

$$\sum_{n_{12}=0}^{S_2} \bar{P}_1(n_{11}, n_{12}) = 1. \tag{3.40}$$

To compute the expected sojourn times in the repair shop for tokens requiring a type 1 component, note that in this case at the arrival of a token requiring a type 1 component there are already k_1 tokens waiting for a type 1 component. This means that this arriving token has to wait until k_1+1 type 1 tokens come out of repair. At the same time there are also n_{12} type 2 components in the repair shop, where n_{12} can be anything between 0 and S_2. Define for $S_1 > 0$, $S_2 > 0$ and $n_{12} = 0, 1, \ldots, S_2$:

$$H_5(k_1+S_1, n_{12}, y) = \sum_{x=k_1+1}^{min(y,k_1+S_1)} \frac{\binom{k_1+S_1}{x}\binom{n_{12}}{y-x}}{\binom{k_1+S_1+n_{12}}{y}} \tag{3.41}$$

as the probability that when y repairs have been performed at least k_1+1 components of type 1 have been repaired, given that there were k_1+S_1 components of type 1 awaiting to be or being repaired and there were n_{12} type 2 components awaiting to be or being repaired. For the expected sojourn time for a token arriving at the repair shop and requiring a type 1 component the following holds:

$$EW_1^a(k_1, 0) = \sum_{n_{12}=0}^{S_2} \Big(\sum_{y=k_1+1}^{k_1+1+n_{12}} (H_5(k_1+S_1, n_{12}, y) - H_5(k_1+S_1, n_{12}, y-1)) \frac{y}{\mu_1} \Big) \bar{P}_1(k_1+S_1, n_{12}). \tag{3.42}$$

For tokens that require a type 2 component similar formulas can be deduced. In case $n_{12} < S_2$ the expected sojourn time will be zero because spare type 2 components will be available. In case $n_{12} = S_2$ the expected sojourn time is the time it takes until one type 2 component comes out of repair, given that there are S_2 type 2 components and $k_1 + S_1$ type 1 components in the repair shop. For $S_1 > 0$ and $S_2 > 0$ define:

$$H_6(k_1 + S_1, S_2, y) = \sum_{x=1}^{min(y,S_2)} \frac{\binom{k_1+S_1}{y-x}\binom{S_2}{x}}{\binom{k_1+S_1+S_2}{y}}. \tag{3.43}$$

Then, the expected sojourn time for a token arriving at the repair shop and requiring a type 2 component is

$$EW_2^a(k_1, 0) = \sum_{y=1}^{k_1+S_1+1} (H_6(k_1+S_1, S_2, y) - H_6(k_1+S_1, S_2, y-1))\frac{y}{\mu_1})\bar{P}_1(k_1+S_1, S_2). \tag{3.44}$$

As the end of this paragraph let us define $q_2(k_1, 0)$ with $k_1 = 1, \ldots, J + S_0$. The variable $q_2(k_1, 0)$ denotes the probability that there are no spare components of type 2 available, given that there is no backlog of type 2 components and there is a backlog of k_1 type 1 components. For $k_1 = 1, \ldots, J + S_0$, let

$$q_2(k_1, 0) = \frac{P_1(\underline{n}_{12} = S_2 \mid \underline{n}_{11} = k_1 + S_1)}{P_1(\underline{n}_{12} \leq S_2 \mid \underline{n}_{11} = k_1 + S_1)} = \bar{P}_1(k_1 + S_1, S_2). \tag{3.45}$$

3.2.3.5 The case $k_1 = 0$ and $k_2 > 0$

This case exhibits great similarity with the case described in the previous paragraph. Therefore we will only briefly discuss this case. The case $k_1 = 0$ and $k_2 > 0$ means that there is no backlog of type 1 components and there is a backlog of k_2 type 2 components. There might be spare type 1 components available.

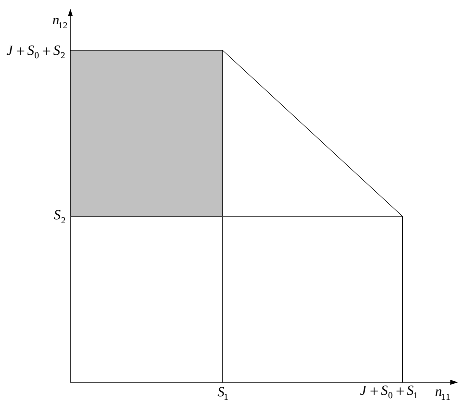


Figure 3.18: The case $k_1 = 0$ and $k_2 > 0$

For each $n_{12} = k_2 + S_2$ the states (n_{11}, n_{12}) with $n_{11} \leq S_1$ are aggregated into one state $(\underline{k}_1 = 0, \underline{k}_2 = k_2)$. To impose this aggregation step on the sojourn times of the tokens in the repair shop, the marginal probabilities that have been computed in Paragraph 3.2.3.1

are needed again. As in the previous paragraphs these probabilities must be scaled for the current case. That is, for $n_{11} = 0, \ldots, S_1$ and $n_{12} = S_2 + 1, \ldots, J + S_0 + S_2$:

$$\hat{P}_1(n_{11}, n_{12}) = \hat{G}\, P_1(n_{11}, n_{12}), \tag{3.46}$$

where $\hat{G}$ is chosen such that for $n_{12} = S_2 + 1, \ldots, J + S_0 + S_2$:

$$\sum_{n_{11}=0}^{S_1} \hat{P}_1(n_{11}, n_{12}) = 1. \tag{3.47}$$

To compute the expected sojourn times for tokens in the repair shop the following is used. For $S_1 > 0$ and $S_2 > 0$ define:

$$H_7(S_1, k_2 + S_2, y) = \sum_{x=1}^{min(y,S_1)} \frac{\binom{S_1}{x}\binom{k_2+S_2}{y-x}}{\binom{k_2+S_1+S_2}{y}}. \tag{3.48}$$

Then, the expected sojourn time for a token arriving at the repair shop and requiring a type 1 component, while there is a backlog of k_2 type 2 components and no token is waiting for a type 1 component, is

$$EW_1^a(0, k_2) = \sum_{y=1}^{k_2+S_2+1} (H_7(S_1, k_2{+}S_2, y) - H_7(S_1, k_2{+}S_2, y{-}1))\frac{y}{\mu_1})\hat{P}_1(S_1, k_2{+}S_2). \tag{3.49}$$

For tokens arriving at the repair shop that require a type 2 component, define for $S_1 > 0$, $S_2 > 0$ and $n_{11} = 0, 1, \ldots, S_1$:

$$H_8(n_{11}, k_2 + S_2, y) = \sum_{x=k_2+1}^{min(y,k_2+S_2)} \frac{\binom{n_{11}}{y-x}\binom{k_2+S_2}{x}}{\binom{n_{11}+k_2+S_2}{y}}, \tag{3.50}$$

then the expected sojourn time for a token arriving at the repair shop and requiring a type 2 component is

$$EW_2^a(0, k_2) = \sum_{n_{11}=0}^{S_1} (\sum_{y=k_2+1}^{n_{11}+k_2+1} (H_8(n_{11}, k_2{+}S_2, y) - H_8(n_{11}, k_2{+}S_2, y{-}1))\frac{y}{\mu_1})\hat{P}_1(n_{11}, k_2{+}S_2). \tag{3.51}$$

For $k_2 = 1, \ldots, J + S_0$, let

$$q_1(0, k_2) = \frac{P_1(\underline{n}_{11} = S_1 \mid \underline{n}_{12} = k_2 + S_2)}{P_1(\underline{n}_{11} \leq S_1 \mid \underline{n}_{12} = k_2 + S_2)} = \hat{P}_1(S_1, k_2 + S_2). \tag{3.52}$$

3.2.4 Alternative determination of the $EW_l^a(\mathbf{k})$

In this subsection the expressions for the $EW_l^a(\mathbf{k})$ for $l \in \{1, 2\}$ $\forall$ $\mathbf{k}$ from the previous subsection will be simplified. To obtain the simplified expressions the following proposition is used:

Proposition 3.2.3
Define for $S_1 > 0,\ S_2 > 0,\ k_1 = 0,\dots,J+S_0,\ Z = 0,\dots,J+S_0+S_2$ *and* $y = k_1+1,\dots,k_1+1+Z$*:*

$$H(k_1+S_1,Z,y) = \sum_{x=k_1+1}^{min(y,k_1+S_1)} \frac{\binom{k_1+S_1}{x}\binom{Z}{y-x}}{\binom{k_1+S_1+Z}{y}},$$

then the following holds:

$$\sum_{y=k_1+1}^{k_1+1+Z} (H(k_1+S_1,Z,y) - H(k_1+S_1,Z,y-1))y = \frac{k_1+1}{k_1+S_1+1}(k_1+S_1+Z+1).$$

Proof. Define

$$\binom{s}{s'} = 0 \text{ if } s' > s.$$

First we prove that for $k_1 \in \mathbb{N}$, $S_1 \in \mathbb{N}$ and $Z \in \mathbb{N}$:

$$\sum_{y=k_1+1}^{k_1+Z} \sum_{x=0}^{k_1} \frac{\binom{k_1+S_1}{x}\binom{Z}{y-x}}{\binom{k_1+S_1+Z}{y}} = \frac{Z(k_1+1)}{k_1+S_1+1}$$

Proof by induction:

(Basis) The result is true for $k_1 = 0$, $S_1 \in \mathbb{N}$ and $Z \in \mathbb{N}$ since
for $Z = 0$:

$$\sum_{y=1}^{0}\sum_{x=0}^{0} \frac{\binom{S_1}{x}\binom{0}{y-x}}{\binom{S_1}{y}} = 0 = \frac{0}{S_1+1},$$

and for $Z \in \mathbb{N}^+$:

$$\begin{aligned}
\sum_{y=1}^{Z}\sum_{x=0}^{0} \frac{\binom{S_1}{x}\binom{Z}{y-x}}{\binom{S_1+Z}{y}} &= \sum_{y=1}^{Z} \frac{\binom{S_1}{0}\binom{Z}{y}}{\binom{S_1+Z}{y}} \\
&= \frac{Z!}{(S_1+Z)!} \sum_{y=1}^{Z} \frac{(S_1+Z-y)!}{(Z-y)!} \\
&= \frac{Z!}{(S_1+Z)!} \frac{(S_1+Z)!}{(Z-1)!(S_1+1)} \quad \text{(by Theorem D.1)} \\
&= \frac{Z}{S_1+1}.
\end{aligned}$$

(Induction hypothesis) Suppose the result is true when $k_1 = a$, $S_1 \in \mathbb{N}$ and $Z \in \mathbb{N}$, that is,

$$\sum_{y=a+1}^{a+Z} \sum_{x=0}^{a} \frac{\binom{a+S_1}{x}\binom{Z}{y-x}}{\binom{a+S_1+Z}{y}} = \frac{Z(a+1)}{a+S_1+1}.$$

Then

$$\begin{aligned}
\sum_{y=a+2}^{a+1+Z}\sum_{x=0}^{a+1}\frac{\binom{a+1+S_1}{x}\binom{Z}{y-x}}{\binom{a+1+S_1+Z}{y}} &= \sum_{y=a+2}^{a+1+Z}\sum_{x=0}^{a}\frac{\binom{a+1+S_1}{x}\binom{Z}{y-x}}{\binom{a+1+S_1+Z}{y}}+\sum_{y=a+2}^{a+1+Z}\frac{\binom{a+1+S_1}{a+1}\binom{Z}{y-a-1}}{\binom{a+1+S_1+Z}{y}}\\
&= \sum_{y=a+1}^{a+Z}\sum_{x=0}^{a}\frac{\binom{a+1+S_1}{x}\binom{Z}{y-x}}{\binom{a+1+S_1+Z}{y}}+\sum_{x=0}^{a}\frac{\binom{a+1+S_1}{x}\binom{Z}{a+1+Z-x}}{\binom{a+1+S_1+Z}{a+1+Z}}\\
&- \sum_{x=0}^{a}\frac{\binom{a+1+S_1}{x}\binom{Z}{a+1-x}}{\binom{a+1+S_1+Z}{a+1}}+\sum_{y=a+2}^{a+1+Z}\frac{\binom{y}{a+1}\binom{a+1+S_1+Z-y}{S_1}}{\binom{a+1+S_1+Z}{a+1+S_1}}\\
\text{(induction hypothesis)} &= \frac{Z(a+1)}{a+S_1+2}+0-\left(1-\frac{\binom{a+1+S_1}{a+1}\binom{Z}{0}}{\binom{a+1+S_1+Z}{a+1}}\right)\\
&+ \sum_{y=a+1}^{a+1+Z}\frac{\binom{y}{a+1}\binom{a+1+S_1+Z-y}{S_1}}{\binom{a+1+S_1+Z}{a+1+S_1}}-\frac{\binom{a+1}{a+1}\binom{S_1+Z}{S_1}}{\binom{a+1+S_1+Z}{a+1+S_1}}\\
&= \frac{Z(a+1)}{a+S_1+2}-1+\frac{(a+1+S_1)!}{(a+1+S_1+Z)!}\,\frac{(S_1+Z)!}{S_1!}\\
\text{(by Theorem D.2)} &+ \frac{\binom{a+S_1+Z+2}{a+S_1+2}}{\binom{a+1+S_1+Z}{a+1+S_1}}-\frac{(S_1+Z)!}{(a+1+S_1+Z)!}\,\frac{(a+1+S_1)!}{S_1!}\\
&= \frac{Z(a+1)}{a+S_1+2}-1+\frac{a+S_1+Z+2}{a+S_1+2}\\
&= \frac{Z(a+1)-(a+S_1+2)+(a+S_1+Z+2)}{a+S_1+2}\\
&= \frac{Z(a+2)}{a+S_1+2}.
\end{aligned}$$

So the result is true when $k_1 = a+1$, $S_1 \in \mathbb{N}$ and $Z \in \mathbb{N}$, and by the principle of induction, it is true for all $k_1 \in \mathbb{N}$, $S_1 \in \mathbb{N}$ and $Z \in \mathbb{N}$.

Now, note that $H(k_1+S_1, Z, k_1+1+Z) = 1$ and $H(k_1+S_1, Z, k_1) = 0$. Therefore:

$$\begin{aligned}
&\sum_{y=k_1+1}^{k_1+1+Z}(H(k_1+S_1,Z,y)-H(k_1+S_1,Z,y-1))y\\
&= k_1+1+Z-\sum_{y=k_1+1}^{k_1+Z}H(k_1+S_1,Z,y)\\
&= k_1+1+Z-\sum_{y=k_1+1}^{k_1+Z}\sum_{x=k_1+1}^{min(y,k_1+S_1)}\frac{\binom{k_1+S_1}{x}\binom{Z}{y-x}}{\binom{k_1+S_1+Z}{y}}\\
&= k_1+1+Z-\sum_{y=k_1+1}^{k_1+Z}\left(1-\sum_{x=0}^{k_1}\frac{\binom{k_1+S_1}{x}\binom{Z}{y-x}}{\binom{k_1+S_1+Z}{y}}\right)
\end{aligned}$$

$$
\begin{aligned}
&= k_1 + 1 + \sum_{y=k_1+1}^{k_1+Z} \sum_{x=0}^{k_1} \frac{\binom{k_1+S_1}{x}\binom{Z}{y-x}}{\binom{k_1+S_1+Z}{y}} \\
&= k_1 + 1 + \frac{Z(k_1+1)}{k_1+S_1+1} \quad \text{(as proved above)} \\
&= \frac{k_1+1}{k_1+S_1+1}(k_1+S_1+Z+1).
\end{aligned}
$$

□

This proposition leads almost directly to the simplified expressions for the $EW_l^a(\mathbf{k})$ for $l \in \{1,2\}$ and all $\mathbf{k}$.

Corollary 3.2.4
For $S_1 \geq 0$ and $S_2 \geq 0$ the following expressions for the $EW_l^a(\boldsymbol{k})$ for $l \in \{1,2\}$ for all $\boldsymbol{k}$ are the same as the expressions in Subsection 3.2.3.

$$
\begin{aligned}
EW_1^a(k_1,k_2) &= \frac{k_1+1}{k_1+S_1+1}(k_1+k_2+S_1+S_2+1)\frac{1}{\mu_1}, \\
&\quad for \quad k_1 = 1,\ldots,J+S_0 \; k_2 = 1,\ldots,J+S_0-k_1, \qquad (3.53) \\
EW_2^a(k_1,k_2) &= \frac{k_2+1}{k_2+S_2+1}(k_1+k_2+S_1+S_2+1)\frac{1}{\mu_1}, \\
&\quad for \quad k_1 = 1,\ldots,J+S_0 \; k_2 = 1,\ldots,J+S_0-k_1, \qquad (3.54) \\
EW_1^a(0,0) &= \sum_{n_{12}=0}^{S_2} \frac{1}{S_1+1}(S_1+n_{12}+1)\frac{1}{\mu_1}\tilde{P}_1(S_1,n_{12}), \qquad (3.55) \\
EW_2^a(0,0) &= \sum_{n_{11}=0}^{S_1} \frac{1}{S_2+1}(n_{11}+S_2+1)\frac{1}{\mu_1}\tilde{P}_1(n_{11},S_2), \qquad (3.56) \\
EW_1^a(k_1,0) &= \sum_{n_{12}=0}^{S_2} \frac{k_1+1}{k_1+S_1+1}(k_1+S_1+n_{12}+1)\frac{1}{\mu_1}\bar{P}_1(k_1+S_1,n_{12}), \\
&\quad for \quad k_1 = 1,\ldots,J+S_0, \qquad (3.57) \\
EW_2^a(k_1,0) &= \frac{1}{S_2+1}(k_1+S_1+S_2+1)\frac{1}{\mu_1}\bar{P}_1(k_1+S_1,S_2), \\
&\quad for \quad k_1 = 1,\ldots,J+S_0, \qquad (3.58) \\
EW_1^a(0,k_2) &= \frac{1}{S_1+1}(k_2+S_1+S_2+1)\frac{1}{\mu_1}\hat{P}_1(S_1,k_2+S_2), \\
&\quad for \quad k_2 = 1,\ldots,J+S_0, \qquad (3.59) \\
EW_2^a(0,k_2) &= \sum_{n_{11}=0}^{S_1} \frac{k_2+1}{k_2+S_2+1}(n_{11}+k_2+S_2+1)\frac{1}{\mu_1}\hat{P}_1(n_{11},k_2+S_2), \\
&\quad for \quad k_2 = 1,\ldots,J+S_0. \qquad (3.60)
\end{aligned}
$$

Proof. We start this proof by looking at the tokens that arrive at the repair shop and require a type 1 component. Equation 3.53 follows immediately when Proposition 3.2.3 is applied on Equation 3.28 and taking $Z = k_2 + S_2$. Equation 3.55 is obtained by applying Proposition 3.2.3 on Equation 3.34 and taking $k_1 = 0$ and $Z = n_{12}$. Applying Proposition 3.2.3 on Equation 3.42 and taking $Z = n_{12}$ leads to (3.57) and similarly Equation 3.59 is obtained by applying Proposition 3.2.3 on Equation 3.49 and taking $k_1 = 0$ and $Z = k_2+S_2$.

To prove Corollary 3.2.4 for tokens requiring a type 2 component, Proposition 3.2.3 can be rewritten into:

Define for $S_1 > 0$, $S_2 > 0$, $k_2 = 0, \ldots, J + S_0$, $\bar{Z} = 0, \ldots, J + S_0 + S_1$ and $y = k_2 + 1, \ldots, k_2 + 1 + \bar{Z}$:

$$H'(\bar{Z}, k_2 + S_2, y) = \sum_{x=k_2+1}^{min(y,k_2+S_2)} \frac{\binom{\bar{Z}}{y-x}\binom{k_2+S_2}{x}}{\binom{\bar{Z}+k_2+S_2}{y}},$$

then the following holds:

$$\sum_{y=k_2+1}^{\bar{Z}+k_2+1} (H'(\bar{Z}, k_2 + S_2, y) - H'(\bar{Z}, k_2 + S_2, y - 1))y = \frac{k_2 + 1}{k_2 + S_2 + 1}(\bar{Z} + k_2 + S_2 + 1).$$

With this rewritten proposition it is easy to obtain Equation 3.54 from Equation 3.30 by taking $\bar{Z} = k_1 + S_1$, to obtain (3.56) from (3.36) by taking $k_2 = 0$ and $\bar{Z} = n_{11}$ and to obtain (3.58) from (3.44) by taking $k_2 = 0$ and $\bar{Z} = k_1 + S_1$. Equation 3.60 is obtained by applying this rewritten proposition on Equation 3.51 and taking $\bar{Z} = n_{11}$. $\square$

The expressions from Corollary 3.2.4 are a lot easier to compute than the expressions from Subsection 3.2.3 and the results are the same! Therefore, to obtain results for the two-type two-indenture system the expressions from Corollary 3.2.4 will be used. These results will be presented in Subsection 3.2.6. In the next subsection an even faster method to obtain the expected sojourn times for the tokens in the repair shop will be described, however at the cost of some accuracy.

3.2.5 Further approximation to the values for $EW_l^a(\mathbf{k})$, $q_1(0, k_2)$ and $q_1(k_1, 0)$

The approximations in the previous subsections all use the marginal distribution for $\mathbf{n}_1$ as has been obtained in Paragraph 3.2.3.1. In the derivation of this distribution all transitions between the (n_{11}, n_{12}) are taken into account. For large systems this will be a time consuming operation. Furthermore, in case the model will be extended to three or more component types, this approach will make the computations even more complex. Therefore, in this subsection the state space (n_{11}, n_{12}) is divided into smaller sub state spaces and these sub state spaces are solved individually. Because the sub state spaces are smaller, it will be easier and therefore less time consuming to find a distribution for the (n_{11}, n_{12}) that satisfies the balance equations for such a sub state space. By a smart choice of the partitioning of the state space, the number of calculations will decrease even more. Of course, since the transitions between the sub state spaces are not taken into account, this approach will lead to less accurate approximations. However, as will be shown in Subsection 3.2.6, the deviations are not very large, so this is a reasonable approach.

The state space (n_{11}, n_{12}) is partitioned into the same sub state spaces as as has been done in Subsection 3.2.3 to obtain the values for the expected sojourn times in the repair shop $EW_l^a(\mathbf{k})$. The partition is depicted in Figure 3.19. Since we are only interested in the marginal distribution of (n_{11}, n_{12}) for $n_{11} \leq S_1$ and/or $n_{12} \leq S_2$, there is no need to consider the case with $n_{11} > 0$ and $n_{12} > 0$. The sub state spaces are denoted by $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$. For each of these sub state spaces will be described how the marginal distribution for $\mathbf{n}_1$ can be approximated.

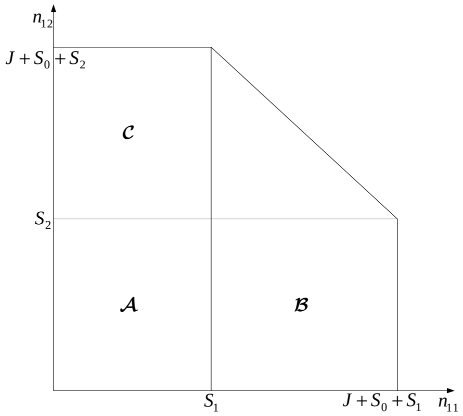


Figure 3.19: The partitioning of the state space.

3.2.5.1 Sub state space $\mathcal{A}$

The sub state space $\mathcal{A} = \{(n_{11}, n_{12}) \mid n_{11} \leq S_1, n_{12} \leq S_2\}$ as depicted in Figure 3.19 is isolated from the rest of the state space. The transition rates for this sub state space are given in Figure 3.20. From these transition rates the balance equations for this sub

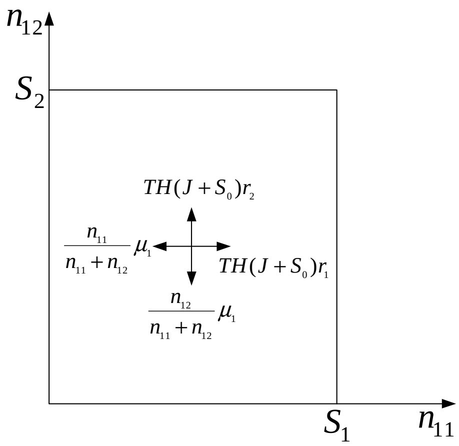


Figure 3.20: The transition rates for sub state space $\mathcal{A}$

state space can be obtained. Solving these equations leads to the marginal steady state probabilities $\tilde{P}_2(n_{11}, n_{12})$ for $n_{11} \leq S_1$ and $n_{12} \leq S_2$. In Paragraph 3.2.3.3 and Subsection 3.2.4 the expected sojourn times $EW_l^a(0,0)$ are obtained with the use of the $\tilde{P}_1(n_{11}, n_{12})$. To use the marginal probabilities as obtained in this paragraph, these $\tilde{P}_1(n_{11}, n_{12})$ must be replaced by the $\tilde{P}_2(n_{11}, n_{12})$.

3.2.5.2 Sub state space $\mathcal{B}$

The sub state space $\mathcal{B}$ is defined as $\mathcal{B} = \{(n_{11}, n_{12}) \mid S_1 < n_{11} \leq J + S_0 + S_1, n_{12} \leq S_2\}$. This sub state space can be partitioned even further. Let $\mathcal{B}_{n_{11}} = \{(n_{11}, n_{12}) \mid n_{12} \leq S_2\}$ for $n_{11} = S_1 + 1, \ldots, J + S_0 + S_1$. The transition rates for $\mathcal{B}_{n_{11}}$ are given in Figure 3.21.

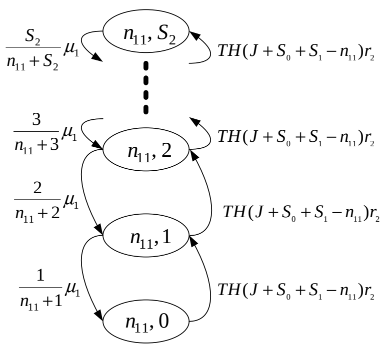


Figure 3.21: The transition rates for sub state space $\mathcal{B}_{n_{11}}$

Since the transition diagram for the $\mathcal{B}_{n_{11}}$ is a one-dimensional Markov chain, the balance equations can be solved very easily. The resulting marginal probabilities are denoted by $\bar{P}_2(n_{11}, n_{12})$.
To implement these probabilities in the approximation algorithm for the two-type two-indenture repairable item system, the $\bar{P}_1(n_{11}, n_{12})$ in the previous subsections must be replaced by these $\bar{P}_2(n_{11}, n_{12})$.

3.2.5.3 Sub state space $\mathcal{C}$

The last sub state space that needs to be considered is $\mathcal{C} = \{(n_{11}, n_{12}) \mid n_{11} \leq S_1, S_2 < n_{12} \leq J + S_0 + S_2\}$. Similar to sub state space $\mathcal{B}$ this sub state space can be partitioned further. For $n_{12} = S_2 + 1, \ldots, J + S_0 + S_2$, let $\mathcal{C}_{n_{12}} = \{(n_{11}, n_{12}) \mid n_{11} \leq S_1\}$. The transition diagram for the $\mathcal{C}_{n_{12}}$ can be depicted as a one-dimensional Markov chain, as has been done in Figure 3.22. The balance equations that follow from the transition diagram can easily be

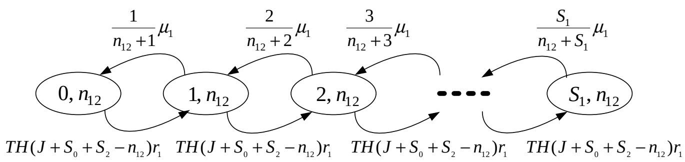


Figure 3.22: The transition rates for sub state space $\mathcal{C}_{n_{12}}$

solved. This will lead to the marginal distribution denoted by $\hat{P}_2(n_{11}, n_{12})$. As before, the $\hat{P}_1(n_{11}, n_{12})$ in the previous subsections must be replaced by these $\hat{P}_2(n_{11}, n_{12})$ to implement this extra approximation step in the algorithm.

3.2.6 Results

In this subsection numerical results will be presented for several performance measures. These performance measures can be obtained from the adapted MDA-algorithm as has been described in Subsection 3.2.2. We are interested in the following:

$$\begin{aligned}
A &= P(\underline{j} = J) = P(\underline{b} \geq J) = P(\underline{k}_1 + \underline{k}_2 + \underline{m}_1 \leq S_0), &(3.61)\\
E\underline{j} &= E(J - [\underline{k}_1 + \underline{k}_2 + \underline{m}_1 - S_0]^+)\\
&= \sum_{\mathbf{k}, m_1} (J - [k_1 + k_2 + m_1 - S_0]^+)\, P(\mathbf{k}, m_1), &(3.62)
\end{aligned}$$

where A denotes the availability, that is, the probability that the maximum number of machines is operating in the production cell and $E\underline{j}$ is the expected number of machines operating in the production cell.

In Table 3.2 the parameter settings for several test problems are given and in Tables 3.3 and 3.4 the results for these test problems are presented. 'Appr 1' in Tables 3.3 and 3.4 denotes the approximation obtained by the adapted MDA algorithm from Subsection 3.2.2 in combination with the expected sojourn times and q-values as obtained in Subsections 3.2.3 and 3.2.4. The approximations for the expected sojourn times in the repair shop and the q-values as obtained in Subsection 3.2.5 are used for 'appr 2' in Tables 3.3 and 3.4. Both of these approximations are compared with simulation results. To obtain the simulation results, a simulation model was built in EM-Plant. In Appendix A more information about these simulations is given. The simulation leads to 95 % confidence intervals. To compare the approximations with the simulation results, the deviation from the approximation to the midpoint of the confidence interval is calculated. The simulation results and the percentage deviations are given in the tables as well.

The first thing to notice, when looking at the results, is that the results for approximation 1 and approximation 2 do not differ very much. In fact, one would expect that the results for approximation 1 would be better than those of approximation 2, but for most test problems this is not the case. The latter can not be clarified, since it is remarkable that an approximation that uses a partitioning of the state space instead of the whole state space gives better results.

To discuss the accuracy of the results, the test problems are divided into three groups of problems. For test problems 1 to 4, the repair rate is greater than the assembly rate ($\mu_1 > \mu_2$), for problems 5 to 24, these rates are equal ($\mu_1 = \mu_2$) and for test problems 25 to 40 the repair rate is smaller than the assembly rate ($\mu_1 < \mu_2$).

For the first group of test problems (1 to 4) a failed machine will consider the assembly facility to be the bottleneck station in the 'repair and assembly process'. Since the assembly rate is smaller than the repair rate, the largest queues will form at the assembly facility and the expected sojourn time in the assembly facility will be larger than the expected sojourn time in the repair shop. The performance measures are therefore mainly affected by the assembly rate and a small change in the repair rate will not have a big influence. The adapted MDA algorithm is exact for the assembly station and approximates the repair station. However, since the repair station does not have a big influence on the results, this approximation at the repair shop also does not have a big influence. As can be noticed in Table 3.3 the results are indeed quite good. All results have an error of at most one percent.

The results for the second group (5 to 24) are less accurate. The repair rate and the

Table 3.2: Parameter settings for test problems two-type two-indenture repairable item system

Problem	J	S_0	S_1	S_2	λ	μ_1	μ_2	r_1	r_2
1	5	2	1	1	1	10	5	0.5	0.5
2	5	2	2	0	1	10	5	0.5	0.5
3	5	4	2	2	1	10	5	0.5	0.5
4	7	4	2	2	1	15	10	0.5	0.5
5	3	0	5	5	1	3	3	0.5	0.5
6	4	1	1	1	1	5	5	0.5	0.5
7	5	1	1	5	1	6	6	0.2	0.8
8	7	2	0	4	1	15	15	0.25	0.75
9	7	2	1	5	1	15	15	0.25	0.75
10	8	0	3	3	1	15	15	0.3	0.7
11	10	5	5	5	1	12	12	0.5	0.5
12	10	1	3	0	1	12	12	0.5	0.5
13	10	1	2	6	1	15	15	0.25	0.75
14	10	3	3	3	1	15	15	0.25	0.75
15	10	1	6	2	1	15	15	0.25	0.75
16	10	1	6	0	1	12	12	0.5	0.5
17	10	1	1	0	1	12	12	0.5	0.5
18	10	1	1	1	1	12	12	0.5	0.5
19	10	2	2	2	1	12	12	0.5	0.5
20	10	1	2	2	1	24	24	0.5	0.5
21	20	0	1	1	1	24	24	0.5	0.5
22	20	0	5	0	1	24	24	0.5	0.5
23	20	0	5	0	1	24	24	0.7	0.3
24	20	0	5	0	1	40	40	0.5	0.5
25	5	2	2	2	1	10	15	0.5	0.5
26	10	0	1	5	1	15	20	0.5	0.5
27	10	5	5	5	1	12	20	0.5	0.5
28	10	1	1	1	1	12	20	0.5	0.5
29	10	1	6	0	1	12	20	0.5	0.5
30	10	1	10	0	1	12	20	0.5	0.5
31	10	1	6	0	1	12	30	0.5	0.5
32	10	1	1	0	1	12	30	0.5	0.5
33	10	2	2	2	1	15	20	0.5	0.5
34	10	3	6	0	1	15	20	0.5	0.5
35	10	3	2	2	1	15	20	0.25	0.75
36	20	0	1	1	1	24	40	0.5	0.5
37	20	1	6	0	1	24	40	0.5	0.5
38	20	1	3	0	1	24	40	0.5	0.5
39	20	1	1	0	1	24	40	0.5	0.5
40	20	6	6	0	1	30	40	0.5	0.5

Table 3.3: Results for test problems from Table 3.2 (1)

Problem		simulation	appr 1	% dev	appr 2	% dev
1	A	(0.5280,0.5356)	0.5267	0.97	0.5267	0.96
	$E\underline{j}$	(4.0565,4.0737)	4.0557	0.23	4.0559	0.23
2	A	(0.5040,0.5124)	0.5064	0.34	0.5065	0.33
	$E\underline{j}$	(4.0063,4.0268)	4.0150	0.04	4.0151	0.04
3	A	(0.6625,0.6696)	0.6628	0.49	0.6628	0.49
	$E\underline{j}$	(4.3272,4.3445)	4.3284	0.17	4.3284	0.17
4	A	(0.8859,0.8890)	0.8848	0.30	0.8848	0.30
	$E\underline{j}$	(6.7897,6.7971)	6.7874	0.09	6.7874	0.09
5	A	(0.3432,0.3459)	0.3414	0.92	0.3412	0.97
	$E\underline{j}$	(1.9551,1.9608)	1.9517	0.32	1.9516	0.33
6	A	(0.5037,0.5079)	0.4992	1.30	0.4991	1.31
	$E\underline{j}$	(3.1414,3.1530)	3.1392	0.25	3.1399	0.23
7	A	(0.5274,0.5331)	0.5227	1.43	0.5224	1.49
	$E\underline{j}$	(4.1297,4.1421)	4.1217	0.34	4.1213	0.35
8	A	(0.8861,0.8887)	0.8867	0.08	0.8867	0.08
	$E\underline{j}$	(6.8264,6.8313)	6.8277	0.02	6.8277	0.02
9	A	(0.9088,0.9112)	0.9085	0.17	0.9085	0.17
	$E\underline{j}$	(6.8616,6.8665)	6.8614	0.04	6.8614	0.04
10	A	(0.5102,0.5139)	0.5081	0.76	0.5081	0.77
	$E\underline{j}$	(7.1879,7.1990)	7.1861	0.10	7.1862	0.10
11	A	(0.7949,0.8018)	0.7993	0.13	0.7993	0.13
	$E\underline{j}$	(9.5192,9.5411)	9.5304	0.00	9.5304	0.00
12	A	(0.3191,0.3244)	0.3323	3.27	0.3315	3.04
	$E\underline{j}$	(8.2689,8.2964)	8.3450	0.75	8.3439	0.74
13	A	(0.6273,0.6314)	0.6255	0.61	0.6255	0.62
	$E\underline{j}$	(9.2683,9.2797)	9.2630	0.12	9.2630	0.12
14	A	(0.8287,0.8328)	0.8282	0.32	0.8282	0.31
	$E\underline{j}$	(9.6578,9.6691)	9.6587	0.05	9.6589	0.05
15	A	(0.5886,0.5939)	0.5824	1.50	0.5823	1.51
	$E\underline{j}$	(9.1630,9.1778)	9.1557	0.16	9.1559	0.16
16	A	(0.3229,0.3276)	0.3377	3.83	0.3365	3.45
	$E\underline{j}$	(8.2980,8.3217)	8.3856	0.91	8.3833	0.88
17	A	(0.2971,0.3015)	0.2996	0.10	0.2996	0.09
	$E\underline{j}$	(8.1329,8.1576)	8.1609	0.19	8.1617	0.20
18	A	(0.3658,0.3713)	0.3641	1.21	0.3641	1.21
	$E\underline{j}$	(8.3686,8.3947)	8.3884	0.08	8.3894	0.09
19	A	(0.5399,0.5459)	0.5437	0.15	0.5437	0.14
	$E\underline{j}$	(8.8739,8.8936)	8.8968	0.15	8.8973	0.15
20	A	(0.8316,0.8341)	0.8299	0.34	0.8300	0.34
	$E\underline{j}$	(9.7440,9.7489)	9.7422	0.04	9.7423	0.04

Table 3.4: Results for test problems from Table 3.2 (2)

Problem		simulation	appr 1	% dev	appr 2	% dev
21	A	(0.1632,0.1676)	0.1572	4.96	0.1571	5.00
	$E\underline{j}$	(17.1146,17.1640)	17.1539	0.09	17.1550	0.09
22	A	(0.1238,0.1257)	0.1278	2.46	0.1271	1.88
	$E\underline{j}$	(17.0438,17.0675)	17.1750	0.70	17.1726	0.69
23	A	(0.1547,0.1567)	0.1541	1.03	0.1533	1.56
	$E\underline{j}$	(17.2981,17.3225)	17.3519	0.24	17.3481	0.22
24	A	(0.3670,0.3702)	0.3671	0.40	0.3668	0.46
	$E\underline{j}$	(18.7644,18.7759)	18.7821	0.06	18.7822	0.06
25	A	(0.9553,0.9565)	0.9564	0.05	0.9564	0.05
	$E\underline{j}$	(4.9399,4.9417)	4.9419	0.02	4.9419	0.02
26	A	(0.4578,0.4617)	0.4528	1.51	0.4520	1.68
	$E\underline{j}$	(9.0341,9.0466)	9.0466	0.07	9.0460	0.06
27	A	(0.4727,0.4784)	0.5062	6.44	0.5042	6.03
	$E\underline{j}$	(8.8862,8.9048)	9.0159	1.35	9.0127	1.32
28	A	(0.5307,0.5368)	0.5460	2.29	0.5459	2.27
	$E\underline{j}$	(8.9601,8.9799)	9.0277	0.64	9.0288	0.66
29	A	(0.4764,0.4816)	0.5180	8.15	0.5145	7.42
	$E\underline{j}$	(8.9210,8.9375)	9.0850	1.74	9.0791	1.68
30	A	(0.4762,0.4821)	0.5188	8.29	0.5150	7.50
	$E\underline{j}$	(8.9221,8.9408)	9.0914	1.79	9.0856	1.73
31	A	(0.5406,0.5451)	0.6004	10.60	0.5957	9.74
	$E\underline{j}$	(9.1005,9.1126)	9.3035	2.16	9.2961	2.08
32	A	(0.4932,0.5003)	0.5140	3.46	0.5138	3.42
	$E\underline{j}$	(8.8700,8.8953)	8.9420	0.67	8.9428	0.68
33	A	(0.8388,0.8431)	0.8447	0.45	0.8448	0.46
	$E\underline{j}$	(9.7111,9.7207)	9.7270	0.11	9.7273	0.12
34	A	(0.8511,0.8549)	0.8761	2.70	0.8760	0.88
	$E\underline{j}$	(9.7357,9.7442)	9.7887	0.50	9.7887	0.50
35	A	(0.8935,0.8982)	0.9031	0.81	0.9032	0.82
	$E\underline{j}$	(9.8013,9.8130)	9.8270	0.20	9.8273	0.21
36	A	(0.2886,0.2925)	0.2912	0.22	0.2910	0.15
	$E\underline{j}$	(18.0508,18.0779)	18.1609	0.53	18.1625	0.54
37	A	(0.4232,0.4298)	0.4846	13.63	0.4820	13.03
	$E\underline{j}$	(18.5749,18.6053)	18.8921	1.62	18.8865	1.59
38	A	(0.4193,0.4241)	0.4664	10.61	0.4652	10.31
	$E\underline{j}$	(18.4911,18.5179)	18.7433	1.29	18.7408	1.28
39	A	(0.3795,0.3841)	0.3962	3.77	0.3961	3.76
	$E\underline{j}$	(18.2019,18.2312)	18.3109	0.52	18.3118	0.52
40	A	(0.9687,0.9716)	0.9800	1.02	0.9801	1.03
	$E\underline{j}$	(19.9386,19.9461)	19.9634	0.11	19.9634	0.11

assembly rate are the same. For an ordinary system both stations would be equal and therefore be the bottleneck of the repair and assembly process to the same extent. However, since this is not an ordinary system and the repair shop contains spare components, this argument can not simply be posed. In case there are spare components available of both types, the sojourn time in the repair shop will be smaller and the assembly facility will be the bottleneck again (see for instance test problems 5, 10, 11, 13, 14 and 19). The results are therefore quite good. In contrast to this, are test problems 12, 16, 21 and 22, that do not have enough spare components available of both types. These results are strikingly less good.
That it is not just this 'bottleneck argument' that matters, follows from test problem 24. This problem has the same parameters as test problem 22, but the repair and assembly rate are much bigger. Because of this increased repair intensity the expected sojourn time in the repair shop is drastically decreased and as a result the error percentage as well.

For the last group (25 to 40), with $\mu_1 < \mu_2$, the repair shop is definitely the bottleneck for the 'repair and assembly process'. The impact of the way the repair shop is modeled on the performance measures for the production cell, is therefore enormous. This last group of test problems really incorporates the test whether the suggested approximation is reasonable or not. As it is revealed in Table 3.4 the errors run up to almost 14 %. Taking $\mu_2 = \infty$ will even lead to higher error percentages (since the assembly facility will not have any influence on the results at all). Nevertheless, test problems 25, 33, 34, 35 and 40 indicate that the errors are smaller for systems with high availabilities. Since in practice one will always aim for systems with high availabilities, this approximation method is not so bad after all. But of course, more research needs to be done to improve the results for all possible input parameters.

As a conclusion to these results it can be stated that the approximation algorithm as it has been proposed in Section 3.2.2 does not give proper results for all possible input parameters. Further research needs to be done. A good way to start this research is to exclude the assembly facility (by setting $\mu_2 = \infty$) and to find a proper approximation for this simple case. From there on, the model can be extended for multiple stations.
But why did this approximation algorithm not work in the first place? A possible explanation is the assumption of 'random order' as was first pointed out in Remark 3.2.1. In the determination of the expected sojourn times and q-values in Subsections 3.2.3, 3.2.4 and 3.2.5 it was assumed that the ratios between the possible states could be obtained by the transition rates from Paragraph 3.2.3.1. However, in these transition rates it was assumed that every order of the components in the repair shop was possible. In reality, not every order is possible due to the FCFS assumption, and some orders will occur more often than others. Especially for systems with many spares of a particular type of components, and only few of the other type, the assumption of random order is doomed to lead to bad results. To be able to drop the assumption of random order, something clever needs to be thought up, which appears to be too complex for the moment.
On the other hand it can be expected that the errors will decrease as the number of stations will increase. In that case more stations will act as bottlenecks and deviations will subside. So reintroducing the disassembly and failure detection station (by setting $\mu_0 < \infty$) will have its advantages. How such a station can be (re)introduced and other possible extensions to the model will be discussed in the next subsection.

3.3 Conclusions

In this chapter several models were developed to cope with the two-indenture repairable item system. In the first model, all machines consisted of just one type of critical components and in the second model the machines consisted of two types of critical components. In both models the repair facility and the assembly facility are modeled as single servers.
The results for the single type model are good. The errors are at most 5 %. The results for the two-type model are less good, with errors up to 15 %. This is larger than the target of 5 % we set in Section 1.3. These errors are probably caused by the fact that the model assumes that machines arriving at the repair shop are serviced in FCFS order, while the approximation algorithm does not take the order of the machines at the repair shop into account. Further research needs to be done to point out whether this imbalance is the cause or not, and how it can be avoided. For systems with high availabilities the errors are reasonable.

An extension to the model that can easily be made, is the inclusion of extra stations in the system. The previously discussed failure detection and disassembly station can be included (by setting $\mu_0 < \infty$), but also transport lines can be included in the system. Since the two-type model is approximated by use of an adapted MDA algorithm, such an extra station can easily be added by adding a station to the MDA algorithm. Only a few simple adjustments must be made to the algorithm. The expected sojourn time and the marginal probabilities must be obtained for this station as well and to obtain the throughput of the system, also this new station must be taken into account. For the single type model this extra station cannot be included so easily. An extra variable must be taken up in the equations for the steady state distribution.
Other possible extensions to the model are the increase of the number of components and to allow for more complex structures in the repair and assembly facilities. The single servers can for instance be replaced by multi-servers, or can be considered to be a job shop as has been done in [13] for the case of an open two-indenture repairable item system.

Chapter 4

Optimization

In the preceding chapters, approximations for several performance measures of closed loop repairable item systems have been obtained. In Chapter 2 a quite accurate approximation method was developed for two-echelon models. In Chapter 3 a less successful approximation method was developed for two-indenture models. The approximation methods from the previous chapters can be used to find an optimal allocation of spares in the system, in order to achieve the best possible performance. How this can be done, will be explained in this chapter. Since the approximation method that was obtained for the two-indenture model has proved to be inaccurate, we will only focus on the two-echelon model here.
In the first section the problem will be formulated. In Section 4.2 an exact and straightforward solution method will be given. The subsequent section discusses a faster method that approximates the optimization problem. Section 4.4 provides results for both methods and in the last section conclusions will be presented.

4.1 The problem of optimizing stock levels

The aim is to maximize the overall performance of the system, taking a budget constraint for stocking costs into account. For the overall performance of the two-echelon repairable item system, the *total availability* A_{tot} is taken, which is defined as follows:

$$A_{tot} = \frac{\sum_{l=1}^{L} J_l \lambda_l A_l}{\sum_{l=1}^{L} J_l \lambda_l}.$$

It can be viewed as the weighted average of the availability per base. This total availability depends on the values of several parameters. The aim is to maximize its value by finding optimal values for $S_0, S_1, S_2, \ldots, S_L$. Therefore the total availability is considered as a function of $S_0, S_1, S_2, \ldots, S_L$, that is $A_{tot}(S_0, S_1, S_2, \ldots, S_L)$.

The constraints for the maximization problem can be found in the stocking costs. Define c_l as the investment associated with keeping one spare of item l, $l = 0, 1, \ldots, L$, at the corresponding stock point. The allowed total investment is C. Then the (non-linear) optimization problem considered in this chapter can be written as:

$$\max \; A_{tot}(S_0, S_1, S_2, \ldots, S_L),$$

subject to

$$\begin{aligned}\sum_{l=0}^{L} c_l S_l &\leq C,\\ S_l &\geq 0,\ for\ l = 0, 1, \ldots, L.\end{aligned}$$

In the next sections solution methods will be obtained to find these optimal values for $S_0, S_1, S_2, \ldots, S_L$.

4.2 Exact solution to the optimization problem

The most straightforward solution method to find the optimal stock levels, is to simply check for all allocations possible which allocation scheme achieves the highest total availability. This method leads to an exact solution. The method is formally described in the following algorithm:

Algorithm 4.2.1
Exact solution to the optimization problem.

1. *Determine the set* $\mathcal{S} = \{S = (S_0, S_1, S_2, \ldots, S_L) \mid \sum_{l=0}^{L} c_l S_l \leq C\}$.
2. *For every* $S \in \mathcal{S}$ *determine* $A_{tot}(S)$.
3. $S_{opt} = \arg\max_{S \in \mathcal{S}} A_{tot}(S)$.

Now, S_{opt} gives the optimal allocation of spares among the different locations. The total availability that is achieved equals $A_{tot}(S_{opt})$.

Of course, computing the values for the total availability for all possible allocations of spares, will be an extremely time-consuming exercise. Therefore the following remark is made.

Remark 4.2.2
Allocating an extra spare to a certain location, will always lead to an increase of the total availability[1]. The result is that any allocation of spares, that allows for allocating another spare to a certain location, without exceeding the total budget C, cannot be optimal. It will be a waste of effort to compute the values of the total availability for these allocations. To incorporate this observation in the algorithm, step 1 should be replaced by:

1. Determine the set $\mathcal{S}' = \{S = (S_0, S_1, S_2, \ldots, S_L) \mid \sum_{l=0}^{L} c_l S_l \leq C,\ \sum_{l=0}^{L} c_l S_l + c_i > C\ for\ i = 0, 1, \ldots, L\}$,

and $\mathcal{S}$ should in the following steps be replaced by $\mathcal{S}'$.

Nevertheless, even after taking this remark into account, this exact solution method is rather time-consuming. In the next section an approximation method will be developed, which will drastically decrease the computational effort required.

[1] Stated without proof. This statement might not be true for all possible parameters. Extensive testing with reasonable parameters did however not indicate this.

4.3 Approximative solution to the optimization problem

In [13] an approximation procedure is given to find the optimal allocation of stocks for an open loop two-indenture model. This method can also be applied on closed loop two-echelon models, as will be revealed here.
The method is a so-called greedy algorithm. At the start of the algorithm no spares are allocated yet. Next, one repeatedly allocates a spare to the location that leads to the maximum increase in total availability per unit of money invested. The algorithm stops when the budget does not allow for any more assignments of spares. The method builds on the assumption that $A_{tot}(S_0, S_1, S_2, \ldots, S_L)$ tends to behave as a multi-dimensional concave function in all its arguments. The algorithm can be presented as follows.

Algorithm 4.3.1
Approximative solution to the optimization problem (greedy approach)

1. *(Initialisation) Set* $\hat{S}_l = 0$, *for* $l = 0, 1, 2, \ldots, L$, *and also set* $\hat{C} = 0$.

2. *Define* Δ_l *for* $l = 0, 1, 2, \ldots, L$, *by*

$$\Delta_l = \begin{cases} \frac{A_{tot}(\hat{S}_0, \hat{S}_1, \ldots, \hat{S}_l + 1, \ldots, \hat{S}_L) - A_{tot}(\hat{S}_0, \hat{S}_1, \ldots, \hat{S}_l, \ldots, \hat{S}_L)}{c_l} & if\ \hat{C} + c_l \leq C, \\ 0 & otherwise. \end{cases}$$

   If $\max_l \Delta_l = 0$ *then go to step 4.*

3. *Let*

$$\begin{aligned} \hat{l} &= \arg\max_l \Delta_l, \\ \hat{C} &= \hat{C} + c_{\hat{l}}, \\ \hat{S}_{\hat{l}} &= \hat{S}_{\hat{l}} + 1, \end{aligned}$$

   and go to step 2.

4. *The resulting stock allocation is presented by*

$$\hat{S}_{opt} = (\hat{S}_0, \hat{S}_1, \hat{S}_2, \ldots, \hat{S}_L).$$

Now $\hat{S}_{opt}$ represents the stock allocation. The accompanying value for the total availability is $A_{tot}(\hat{S}_{opt})$.

Example 4.3.2
An implementation of this greedy algorithm is given for a simple case. Assume $L = 2$ and let $J_1 = 5$, $J_2 = 7$, $p_1 = p_2 = 0.5$, $\lambda_1 = \lambda_2 = 1$, $\mu_0 = \mu_1 = \mu_2 = 5$, $R_0 = R_2 = 2$, $R_1 = 1$ and $\gamma_1 = \gamma_2 = 10$. The total allowed budget for stocking costs is $C = 20$. Stocking at the depot costs $c_0 = 1$ per item and stocking at base 1 and base 2 costs $c_1 = 2$ and $c_2 = 2$ per item respectively. The steps of the greedy algorithm presented above are displayed in Table 4.1. At each step we determine for which l the maximum value of Δ_l is obtained. This is indicated by the numbers in bold type.
The resulting allocation $(\hat{S}_0, \hat{S}_1, \hat{S}_2) = (4, 4, 4)$ means that at the depot and at each of the bases 4 items should be kept in stock. The total availability for that stock allocation is 0.9716.

Table 4.1: Greedy approach for Example 4.3.2

$(\hat{S}_0, \hat{S}_1, \hat{S}_2)$	A_{tot}	$\hat{C}$	Δ_0	Δ_1	Δ_2
(0,0,0)	0.2357	0	**0.0872**	0.0586	0.0809
(1,0,0)	0.3229	1	0.0417	0.0565	**0.0884**
(1,0,1)	0.4997	3	0.0516	0.0554	**0.0571**
(1,0,2)	0.6138	5	0.0444	**0.0553**	0.0308
(1,1,2)	0.7244	7	**0.0448**	0.0328	0.0312
(2,1,2)	0.7692	8	0.0250	**0.0304**	0.0261
(2,2,2)	0.8300	10	0.0228	0.0160	**0.0265**
(2,2,3)	0.8829	12	**0.0171**	0.0163	0.0124
(3,2,3)	0.9000	13	0.0100	**0.0151**	0.0102
(3,3,3)	0.9301	15	0.0087	0.0077	**0.0104**
(3,3,4)	0.9509	17	0.0061	**0.0078**	0.0045
(3,4,4)	0.9666	19	**0.0051**	0	0
(4,4,4)	0.9716	20	0	0	0

This approximative solution method considerably decreases the amount of computation time required. For instance, for Example 4.3.2, the total availability needs to be evaluated 35 times for the greedy approach. In contrast to that, the exact algorithm from Section 4.2, needs 506 computations of the total availability. When the observations from Remark 4.2.2 are applied one still needs to compute the values of the total availability for 66 different stock allocations. For cases that concern more spares, even more improvement on computation time will be achieved.

On the other hand, the algorithm leads to an approximation instead of exact results. The decrease of computation time, will also lead to a decrease in accuracy. In the next section results will be presented for the approximation algorithm as well as for the exact algorithm. This will give insight in the extent of the decrease in accuracy.

4.4 Results

In this section results will be obtained for several test problems. The results obtained by the exact algorithm from Section 4.2 will be compared to the results obtained by the approximative algorithm from Section 4.3.

In Table 4.2 several test problems are presented. The parameters are given as well as the results for both algorithms. All test problems concern a two-base model. Of course, the algorithms can also be applied on models that contain more than two bases. This will however not be done here.

The results show that the greedy algorithm from Section 4.3 gives a quite accurate approximation. In most cases the algorithm gives the exact solution. In other cases the algorithm gives a solution that leads to a total availability value that is quite near the optimum. Of course, the number of test problems presented here is quite small. Nevertheless, for other test problems similar levels of accuracy were obtained. The approximation method has proved successful.

Table 4.2: Test problems optimization

	L	J_l	λ_l	μ_l μ_0	R_l R_0	p_l	γ_l	c_l c_0	C	A_{tot} ex	S_l ex S_0 ex	A_{tot} appr	S_l appr S_0 appr
1	2	5	1	5	1	0.5	10	1	10	0.7513	4	0.7513	4
		5	1	5	1	0.5	10	1			4		4
				5	1			1			2		2
2	2	5	1	5	1	0.5	10	1	20	0.8668	7	0.8662	7
		5	1	5	1	0.5	10	1			7		6
				5	1			1			6		7
3	2	5	1	5	1	0.5	10	2	20	0.8328	3	0.8328	3
		5	1	5	1	0.5	10	1			5		5
				5	1			1			9		9
4	2	5	1	5	1	0.5	10	2	20	0.7977	3	0.7977	3
		5	1	5	1	0.5	10	2			3		3
				5	1			1			8		8
5	2	5	1	5	1	0.5	1	2	20	0.6487	4	0.6438	5
		5	1	5	1	0.5	1	2			4		4
				5	1			1			4		2
6	2	5	1	5	1	0.5	10	2	20	0.9716	4	0.9716	4
		7	1	5	2	0.5	10	2			4		4
				5	2			1			4		4
7	2	5	1	5	1	0.5	10	1	20	0.9987	10	0.9987	10
		7	1	5	2	0.5	10	1			10		10
				5	2			2			0		0
8	2	10	1	5	2	0.5	10	2	20	0.9144	4	0.9144	4
		10	1	5	2	0.5	10	2			4		4
				5	3			1			4		4
9	2	10	2	5	4	0.5	10	2	20	0.6327	5	0.6234	4
		10	1	5	2	0.5	10	2			2		3
				5	3			1			6		6
10	2	3	1	3	1	0.2	1	2	20	0.6976	3	0.6976	3
		7	1	3	2	0.8	1	2			6		6
				3	2			1			2		2

4.5 Conclusions

In this chapter a method has been developed to obtain the optimal stock allocation for a closed loop two-echelon model. Given a certain budget for stocking costs, one wants to determine how to allocate spare machines in order to obtain maximal total availability. This total availability is the weighted mean of the availabilities per base. To obtain these availabilities per base, the approximation algorithm from Chapter 2 is used.
Since it takes quite some computation time to obtain an exact solution for the optimal stock allocation, an approximative algorithm has been developed. As intended, this approximative algorithm speeds up the computations considerably. It does not always obtain the optimal stock allocation, but the resulting total availability does not differ much from the optimum. The stock allocation obtained can therefore be considered as quite good.

The algorithms developed in this chapter, all consider the closed loop two-echelon model from Chapter 2, because we have a good approximation algorithm for the performance measures, as obtained in that chapter. It is expected that once a good approximation method for the performance measures of the closed loop two-indenture system is developed, the algorithm can also be applied on the two-indenture system. It might even be practicable already, since the approximation method works quite well for high availabilities. When optimizing with a reasonable budget one will always end up at a solution that leads to a high availability.
Because the two-indenture system contains just one base, only the availability of this base needs to be maximized. However, before implementing the approximation algorithm one should check whether the two-indenture model satisfies the assumption that the availability function is multi-dimensional concave in all its arguments. When this assumption is not met, the greedy algorithm will not lead to proper results.

Chapter 5

Conclusions and recommendations

In this chapter conclusions will be drawn and recommendations will be given. In the first section, an outline is given on the conclusions presented in the previous chapters. We will reflect on the research objective, as it was proposed in Section 1.3. In addition some overall conclusions are drawn. In the second section interesting recommendations for future research will be presented.

5.1 Conclusions

In Section 1.3 the research objective for this thesis was defined. It consisted of three parts. The first aim was to consider closed loop two-echelon repairable item systems and to find a reasonable approximation for the availability and the expected number of operational machines at the production cells. The maximum error had to be smaller than 5 %. In this thesis a closed loop two-echelon model has been presented and an approximation algorithm has been developed. The results for the performance measures availability and the expected number of machines operational have an error percentage of less than 1 %. The approximation is therefore considered to be very accurate.
Unfortunately the algorithm is not very fast. This leads to rather long computation times for large systems.

The second part of the research objective was on closed loop two-indenture repairable item systems. Again the aim was to obtain approximations for both the availability and the expected number of operational machines at the production cell, with an error percentage smaller than 5 %. For the two-indenture model an approximation algorithm similar to that of the previous model has been developed. For the single type model, the approximations have a maximum error of 5 %. For the two-type model, however, the errors are much larger than 5 %. The most likely cause for these large errors, is the assumption that every possible order of spare components in the repair shop, occurs with the same probability. In reality this will not be true, especially not for systems with many spares of a particular type of components, and only few of the other type. Because of the assumption of FCFS in the repair shop, this order of components is important. For systems with high availabilities the errors are reasonable.

Both of these models can be used to perform a 'what-if'-analysis. One can easily obtain results for models with different parameter inputs. This way one can quickly compare different scenarios. One no longer needs to learn from practice, or to perform time-consuming and costly simulation studies. It is interesting to find the optimal balance between repair capacity and the amount of spares.

In addition to this possibility of performing a 'what-if'-analysis, we wanted to find an algorithm to determine how spares should be allocated to the different locations, such that optimal availability is obtained. This goal was set in the last part of the research objective. An approximation algorithm has been developed for the two-echelon model. The results are quite good. We expect this algorithm to be applicable to the two-indenture model as well.

A final remark can be made on the models used. The models are quite realistic, but are focused on steady state solutions. In reality however, due to limited working hours, this steady state will never be reached.
Also the systems observed are quite simple. The transport from the bases to the depot, for instance, is left out of consideration. For extensions to the models we refer to the next section.

In summary, the research objective is reached, except for the two-indenture model. This model turned out to be too complicated. However, we did make some progress.
The main contribution of this thesis to science, is the development of the approximation algorithm for the two-echelon model. In contrast to the model by Albright [2], this algorithm can probably, in slightly adapted form, be applied on more complicated systems as well.

5.2 Recommendations

As mentioned in the previous section, the approximation algorithm for the two-echelon model is not very fast. Further aggregation steps may speed up the system evaluation considerably. More research needs to be done.

Further extensions to the two-echelon model can make it more realistic. Transport from the bases to the depot can be included and it would be good to allow for more complicated networks in the repair facilities. In the two-echelon model described in this thesis, each repair shop is modeled as a multi-server station. An interesting extension to this, is to consider the repair facility to be a job shop and model it as a limited capacity open queuing network, as has been done in [4] for the case of an open multi-echelon repairable item system. Then, it is easy to include transport to the depot repair facility as just an additional node in the job shop.

The approximation algorithm as developed for the two-indenture model should be improved. The order in which components are in the repair shop, should be taken into account explicitly. Or, at least, better assumptions on this order should be made. The assumption that all orders are just as likely proved not applicable.

The components in the repair shop of the two-indenture model are repaired in FCFS-order. Because of this, a situation can occur in which a type 1 component is being repaired, while spare type 1 components are available and there is a backlog of type 2 components.

Obviously, this is not efficient. It would be good to consider models that include priority-scheduling.

Furthermore, also for the two-indenture model, numerous extensions can be made. For instance, one can consider more than two components, incorporate transport lines, or take more complicated repair shops into account. Also the repair times can be made type-dependent. In the current model, it is quite easy to include extra stations. But of course, first a good approximation algorithm for the original model must be obtained.

Once an accurate approximation method for the two-indenture model is attained, one can attempt to analyze and approximate a combination of the two-echelon model and the two-indenture model. This leads to a two-echelon two-indenture model. In this model there are several bases and a depot and all machines consist of several components. A failed machine is replaced by a spare and is disassembled, after which the failed component can be repaired locally and centrally. The broken machine and a spare component are assembled. There are spare components and spare machines throughout the system.
The computations will be more complex, but once a good approximation for the two-indenture model is obtained, it should be possible to integrate the models.

The optimization algorithm in this thesis performs quite well. However, it only considers the stocking costs. It would be interesting to find an algorithm that also includes repair costs. Furthermore, one can set a certain required availability level for each base, and then determine the minimum number of spares necessary at all locations. In addition other performance measures are worth considering, such as the number of backorders.

Bibliography

[1] Albright, S.C. & Soni, A., Markovian multi-echelon repairable inventory system, *Naval Research Logistics*, 35 (1988), 1, p. 49-61.

[2] Albright, S.C., An approximation to the stationary distribution of a multiechelon repairable-item inventory system with finite sources and repair channels, *Naval Research Logistics*, 36 (1989), 2, p. 179-195.

[3] Albright, S.C. & Gupta, A., Steady-State Approximation of a Multiechelon Multi-Indentured Repairable-Item Inventory System with a Single Repair Facility, *Naval Research Logistics*, 40 (1993), 4, p. 479-494.

[4] Avsar, Z.M. & Zijm, W.H.M., Capacitated two-echelon inventory models for repairable item systems, in: S.B. Gershwin et al. (eds.): *Analysis and Modeling of Manufacturing Systems*, p. 1-36, Kluwer Academic Publishers, Boston (2002).

[5] Buzacott, J.A. & Shanthikumar, J.G., *Stochastic models of manufacturing systems*, Prentice-Hall, Englewood Cliffs, NJ (1993).

[6] Díaz, A. & Fu, M.C., Models for multi-echelon repairable item inventory systems with limited repair capacity, *European Journal of Operational Research*, 97 (1997), 3, p. 480-492.

[7] Gross, D., Kioussis, L.C. & Miller, D.R., A network decomposition approach for approximate steady state behavior of Markovian multi-echelon repairable item inventory systems, *Management Science*, 33 (1987), p. 1453-1468.

[8] Gupta, A. & Albright, S.C., Steady-state approximations for a multi-echelon multi-indentured repairable-item inventory system, *European Journal of Operational Research*, 62 (1992), 3, p. 340-353.

[9] Gupta, A., S.C., Approximate solution of a single-base multi-indentured repairable-item inventory system, *The journal of the Operational Research Society*, 44 (1993), 7, p. 701-710.

[10] Harrison, P.G. & Patel, N.M., *Performance modelling of communication networks and computer architectures*, Addison Wesley, (1993).

[11] Muckstadt, J.A., Model for a multi-item multi-echelon multi-indenture inventory system, *Management Science*, 20 (1973), 4, p. 472-481.

[12] Sherbrooke, C.C., METRIC: a multi-echelon technique for recoverable item control, *Operations Research*, 16 (1968), p. 122-141.

[13] Zijm, W.H.M. & Avsar, Z.M., Capacitated two-indenture models for repairable item systems, *International Journal of Production Economics*, 81-82 (2003), p. 573-588.

Appendix A

Simulations

In this appendix the simulations performed on the two-echelon model and the two-indenture model will be discussed. The simulations have been performed in the software package eM-Plant.

A.1 eM-Plant

eM-Plant is software for integrated, graphic and object oriented modeling, simulation and animation. Most complex systems, consisting of material, personnel, information and organization may be displayed true to nature and in great detail. For this thesis eM-Plant has been used to build a simulation model for the two-echelon model as well as for the two-indenture model.

For the two-echelon model first a model was built for the case of two bases. Later on a more flexible model was built that consisted of one up to ten bases. In these models all input parameters can be specified separately, to be able to run simulations for whatever test problem we want to examine.

For the two-indenture model another simulation model was built. The machines in this model consist of two types of components. In the graphical representation of the model each type of component is given its own color. As for the previous model, simulations can be performed for whatever input parameters we prefer.

It would go too far to present all programming that is performed to build these simulation models.

A.2 Statistics

The aim of the simulations was to obtain simulation values for the two performance measures availability and the expected number of machines operational. These values had to be obtained for each base separately. The expected number of machines operational can easily be obtained since this is simply the relative occupation of the production cell, which is calculated by eM-Plant, times the capacity of the production cell. To obtain the availability, the time is kept during which the maximum number of machines in the production cell is operating. Dividing this time by the total run time gives the availability. Each run starts with a warm-up period, to bring the system to steady state. The computations described before only start after this warm-up period has ended.

After all required runs have been carried out, the performance measures can be obtained. Each run leads to one value for each performance measure for each base. The average of these values is taken and a 95 % confidence interval is obtained. It is made sure that enough runs were performed in order to obtain confidence intervals for which the percentage deviation of the limits from the mean is less than 1 %.

Appendix B

Additional results

Table B.1: Results for the single base model, $p_1 = 0.5, \lambda_1 = 1, \mu_0 = J_1, \mu_1 = J_1$

J	S_0	S_1	A_{exact}	A_{appr}	% error	$E\underline{j}_1 exact$	$E\underline{j}_1 appr$	% error
3	1	0	0.5056	0.5100	0.86	2.3178	2.3225	0.20
3	3	0	0.5749	0.5771	0.38	2.4338	2.4368	0.12
3	5	0	0.5874	0.5880	0.11	2.4544	2.4553	0.04
3	1	1	0.7322	0.7340	0.25	2.6331	2.6345	0.05
3	3	1	0.7948	0.7961	0.16	2.7264	2.7279	0.05
3	5	1	0.8082	0.8087	0.06	2.7463	2.7469	0.02
3	1	3	0.9171	0.9172	0.01	2.8875	2.8873	0.01
3	3	3	0.9465	0.9466	0.01	2.9287	2.9287	0.00
3	5	3	0.9535	0.9536	0.01	2.9385	2.9385	0.00
3	1	4	0.9538	0.9538	0.00	2.9376	2.9374	0.01
3	3	4	0.9722	0.9722	0.00	2.9630	2.9629	0.00
3	5	4	0.9766	0.9766	0.00	2.9691	2.9691	0.00
5	1	0	0.4690	0.4722	0.69	4.1654	4.1688	0.08
5	3	0	0.5452	0.5470	0.33	4.3224	4.3250	0.06
5	5	0	0.5602	0.5607	0.10	4.3529	4.3538	0.02
5	1	1	0.7045	0.7059	0.21	4.5407	4.5416	0.02
5	3	1	0.7748	0.7758	0.13	4.6643	4.6654	0.02
5	5	1	0.7905	0.7909	0.05	4.6915	4.6920	0.01
5	1	3	0.9068	0.9069	0.01	4.8573	4.8570	0.01
5	3	3	0.9403	0.9404	0.01	4.9111	4.9110	0.00
5	5	3	0.9484	0.9484	0.00	4.9240	4.9240	0.00
5	1	4	0.9480	0.9480	0.00	4.9207	4.9205	0.00
5	3	4	0.9689	0.9689	0.00	4.9537	4.9536	0.00
5	5	4	0.9740	0.9740	0.00	4.9617	4.9617	0.00
10	1	0	0.4318	0.4339	0.47	8.9658	8.9676	0.02
10	3	0	0.5150	0.5162	0.24	9.1819	9.1836	0.02
10	5	0	0.5329	0.5333	0.08	9.2279	9.2286	0.01
10	1	1	0.6746	0.6756	0.14	9.4175	9.4177	0.00
10	3	1	0.7535	0.7542	0.09	9.5842	9.5848	0.01
10	5	1	0.7718	0.7721	0.03	9.6225	9.6228	0.00
10	1	3	0.8953	0.8953	0.01	9.8165	9.8161	0.00
10	3	3	0.9335	0.9335	0.00	9.8880	9.8879	0.00
10	5	3	0.9428	0.9428	0.00	9.9054	9.9053	0.00
10	1	4	0.9414	0.9414	0.00	9.8980	9.8978	0.00
10	3	4	0.9652	0.9652	0.00	9.9415	9.9414	0.00

Table B.2: Results for the single base model, $p_1 = 0.25, \lambda_1 = 1, \mu_0 = 2J_1, \mu_1 = J_1$

J	S_0	S_1	A_{exact}	A_{appr}	% error	$E\underline{j}_1 exact$	$E\underline{j}_1 appr$	% error
3	1	0	0.5348	0.5383	0.66	2.3402	2.3436	0.15
3	3	0	0.6743	0.6783	0.59	2.5726	2.5777	0.20
3	5	0	0.7282	0.7310	0.38	2.6619	2.6658	0.15
3	1	1	0.7201	0.7208	0.09	2.5951	2.5956	0.02
3	3	1	0.8384	0.8394	0.12	2.7746	2.7757	0.04
3	5	1	0.8906	0.8914	0.10	2.8537	2.8548	0.04
3	1	3	0.8705	0.8705	0.00	2.8110	2.8109	0.00
3	3	3	0.9311	0.9311	0.00	2.8999	2.8999	0.00
3	5	3	0.9613	0.9613	0.00	2.9442	2.9443	0.00
3	1	4	0.9075	0.9075	0.00	2.8649	2.8649	0.00
3	3	4	0.9505	0.9505	0.00	2.9278	2.9278	0.00
3	5	4	0.9726	0.9726	0.00	2.9602	2.9602	0.00
5	1	0	0.4900	0.4923	0.45	4.1493	4.1514	0.05
5	3	0	0.6429	0.6455	0.40	4.4641	4.4675	0.08
5	5	0	0.7066	0.7085	0.26	4.5946	4.5975	0.06
5	1	1	0.6814	0.6818	0.06	4.4558	4.4560	0.00
5	3	1	0.8147	0.8154	0.08	4.6983	4.6990	0.01
5	5	1	0.8761	0.8767	0.06	4.8098	4.8105	0.01
5	1	3	0.8477	0.8477	0.00	4.7371	4.7370	0.00
5	3	3	0.9182	0.9182	0.00	4.8597	4.8596	0.00
5	5	3	0.9540	0.9540	0.00	4.9219	4.9219	0.00
5	1	4	0.8904	0.8904	0.00	4.8106	4.8106	0.00
5	3	4	0.9409	0.9409	0.00	4.8980	4.8980	0.00
5	5	4	0.9672	0.9672	0.00	4.9436	4.9436	0.00
10	1	0	0.4390	0.4401	0.25	8.8481	8.8489	0.01
10	3	0	0.6051	0.6064	0.22	9.2890	9.2906	0.02
10	5	0	0.6807	0.6817	0.14	9.4891	9.4906	0.02
10	1	1	0.6338	0.6340	0.03	9.2282	9.2282	0.00
10	3	1	0.7843	0.7846	0.04	9.5703	9.5706	0.00
10	5	1	0.8574	0.8576	0.03	9.7364	9.7367	0.00
10	1	3	0.8177	0.8177	0.00	9.6112	9.6112	0.00
10	3	3	0.9007	0.9007	0.00	9.7898	9.7898	0.00
10	5	3	0.9440	0.9440	0.00	9.8828	9.8828	0.00
10	1	4	0.8675	0.8675	0.00	9.7170	9.7170	0.00
10	3	4	0.9277	0.9277	0.00	9.8460	9.8460	0.00
10	5	4	0.9597	0.9597	0.00	9.9146	9.9146	0.00

Table B.3: Parameter settings for test problems multi-base model with transportation (2)

Problem	L	J_l	S_l S_0	λ_l	μ_l μ_0	R_l R_0	p_l	γ_l	ρ_l ρ_0
11	2	5	5	1	5	1	0.5	10	0.5
		5	5	1	5	1	0.5	10	0.5
			5		10	1			0.5
12	2	5	2	1	2	1	0.2	∞	0.5
		5	2	1	2	1	0.2	∞	0.5
			3		10	1			0.8
13	2	5	2	1	2	2	0.2	∞	0.25
		5	2	1	2	2	0.2	∞	0.25
			3		10	2			0.4
14	2	5	2	1	2	2	0.2	5	0.25
		5	2	1	2	2	0.2	5	0.25
			3		10	2			0.4
15	2	5	1	1	5	1	0.5	∞	0.5
		5	1	1	5	1	0.5	∞	0.5
			2		5	1			1
16	2	5	3	1	2	3	0.5	5	0.42
		5	3	1	2	3	0.5	5	0.42
			2		3	3			0.56
17	2	5	2	1	5	1	0.5	10	0.5
		5	2	1	5	1	0.5	10	0.5
			4		2	10			0.25
18	2	5	2	1	5	1	0.5	10	0.5
		5	2	1	5	1	0.5	10	0.5
			8		5	3			0.33
19	2	5	2	1	5	1	0.5	10	0.5
		5	2	1	5	1	0.5	10	0.5
			8		1	8			0.63
20	2	7	3	1	5	1	0.25	∞	0.35
		7	3	1	5	1	0.25	∞	0.35
			3		10	1			1.05

Table B.4: Parameter settings for test problems multi-base model with transportation (3)

Problem	L	J_l	S_l S_0	λ_l	μ_l μ_0	R_l R_0	p_l	γ_l	ρ_l ρ_0
21	2	5 10	1 3 3	1 1	5 10 10	1 1 1	0.5 0.5	∞ ∞	0.5 0.5 0.75
22	2	2 8	1 3 3	1 1	2 8 5	1 1 1	0.5 0.7	∞ ∞	0.5 0.7 0.68
23	2	5 7	2 2 1	1 1	5 5 10	1 1 1	0.5 0.5	∞ ∞	0.5 0.7 0.6
24	3	5 5 5	2 2 2 8	1 1 1	5 5 5 5	1 1 1 3	0.5 0.5 0.5	10 10 10	0.5 0.5 0.5 0.5
25	3	5 5 5	1 1 1 1	1 1 1	2 2 2 4	3 3 3 8	0.5 0.5 0.5	10 10 10	0.42 0.42 0.42 0.23
26	3	2 5 7	1 1 1 3	2 1 1	3 2 5 4	1 3 3 8	0.5 0.5 0.5	5 10 10	0.67 0.42 0.23 0.25
27	3	7 7 7	5 5 5 5	1 2 3	3 3 3 3	3 3 7 7	0.5 0.2 0.8	10 10 10	0.39 0.31 0.8 0.9
28	3	7 7 7	0 5 10 5	1 1 1	5 5 5 5	2 2 2 2	0.5 0.5 0.5	10 10 10	0.35 0.35 0.35 1.05
29	3	3 3 3	2 2 2 2	1 1 1	5 5 5 5	1 2 3 2	0.5 0.5 0.5	5 5 5	0.3 0.15 0.1 0.45
30	4	5 5 5 5	2 2 2 2 2	1 1 1 1	5 5 5 5 5	2 2 2 2 4	0.5 0.5 0.5 0.5	10 10 10 10	0.25 0.25 0.25 0.25 0.5

Table B.5: Results for test problems from Tables (B.3) and (B.4)

Problem	$A_l sim$	$A_l appr$	% dev	$E\underline{j}_l sim$	$E\underline{J}_l appr$	% dev
11	(0.9824,0.9853)	0.9840	0.02	(4.9735,4.9786)	4.9765	0.01
	(0.9826,0.9854)	0.9840	0.00	(4.9737,4.9792)	4.9765	0.00
12	(0.8148,0.8294)	0.8192	0.35	(4.7034,4.7322)	4.7129	0.10
	(0.8151,0.8266)	0.8192	0.20	(4.7062,4.7304)	4.7129	0.11
13	(0.9731,0.9756)	0.9731	0.12	(4.9666,4.9705)	4.9669	0.03
	(0.9720,0.9742)	0.9731	0.01	(4.9650,4.9690)	4.9669	0.00
14	(0.8536,0.8585)	0.8563	0.03	(4.8066,4.8149)	4.8118	0.02
	(0.8559,0.8607)	0.8563	0.23	(4.8108,4.8181)	4.8118	0.05
15	(0.5481,0.5550)	0.5526	0.19	(4.1956,4.2123)	4.2061	0.05
	(0.5462,0.5522)	0.5526	0.61	(4.1962,4.2112)	4.2061	0.06
16	(0.8470,0.8496)	0.8493	0.12	(4.7762,4.7823)	4.7804	0.02
	(0.8487,0.8510)	0.8493	0.06	(4.7788,4.7833)	4.7804	0.01
17	(0.8561,0.8626)	0.8594	0.01	(4.7876,4.7981)	4.7931	0.01
	(0.8567,0.8614)	0.8594	0.04	(4.7882,4.7982)	4.7931	0.00
18	(0.8684,0.8727)	0.8714	0.09	(4.8062,4.8150)	4.8113	0.01
	(0.8704,0.8752)	0.8714	0.16	(4.8103,4.8195)	4.8113	0.08
19	(0.8551,0.8594)	0.8555	0.20	(4.7859,4.7923)	4.7851	0.08
	(0.8526,0.8606)	0.8555	0.13	(4.7798,4.7945)	4.7851	0.04
20	(0.6532,0.6811)	0.6608	0.95	(6.2607,6.3417)	6.2806	0.33
	(0.6480,0.6813)	0.6608	0.58	(6.2491,6.3370)	6.2806	0.20
21	(0.7250,0.7325)	0.7305	0.25	(4.5786,4.5933)	4.5884	0.05
	(0.8776,0.8871)	0.8813	0.12	(9.7689,9.7944)	9.7783	0.03
22	(0.7985,0.8068)	0.8019	0.09	(1.7560,1.7676)	1.7607	0.06
	(0.7977,0.8020)	0.7994	0.05	(7.6077,7.6190)	7.6096	0.05
23	(0.8511,0.8587)	0.8561	0.14	(4.7765,4.7885)	4.7849	0.05
	(0.6898,0.6971)	0.6933	0.02	(6.4198,6.4366)	6.4237	0.07
24	(0.8703,0.8741)	0.8711	0.03	(4.8091,4.8166)	4.8109	0.00
	(0.8709,0.8756)	0.8711	0.08	(4.8098,4.8198)	4.8109	0.01
	(0.8676,0.8718)	0.8711	0.25	(4.8051,4.8128)	4.8109	0.08
25	(0.5066,0.5131)	0.5109	0.21	(4.2460,4.2587)	4.2558	0.06
	(0.5068,0.5144)	0.5109	0.06	(4.2466,4.2613)	4.2558	0.04
	(0.5041,0.5130)	0.5109	0.46	(4.2436,4.2618)	4.2558	0.07
26	(0.6456,0.6538)	0.6525	0.43	(1.5538,1.5658)	1.5638	0.26
	(0.5754,0.5821)	0.5790	0.04	(4.3828,4.3952)	4.3883	0.02
	(0.7056,0.7089)	0.7070	0.04	(6.5989,6.6031)	6.6016	0.01
27	(0.9577,0.9617)	0.9599	0.02	(6.9352,6.9433)	6.9400	0.01
	(0.7820,0.7903)	0.7859	0.03	(6.5683,6.5911)	6.5778	0.03
	(0.4492,0.4542)	0.4510	0.15	(5.8151,5.8303)	5.8196	0.05
28	(0.1745,0.1807)	0.1766	0.59	(5.0709,5.1143)	5.0859	0.13
	(0.8530,0.8624)	0.8575	0.03	(6.7166,6.7389)	6.7280	0.00
	(0.9845,0.9862)	0.9848	0.05	(6.9731,6.9760)	6.9738	0.01
29	(0.9430,0.9450	0.9443	0.04	(2.9308,2.9337)	2.9326	0.01
	(0.9649,0.9671	0.9670	0.10	(2.9595,2.9624)	2.9622	0.04
	(0.9674,0.9686)	0.9686	0.07	(2.9627,2.9644)	2.9644	0.03
30	(0.9251,0.9281)	0.9268	0.02	(4.9046,4.9094)	4.9074	0.01
	(0.9247,0.9274)	0.9268	0.08	(4.9039,4.9082)	4.9074	0.03
	(0.9249,0.9276)	0.9268	0.06	(4.9049,4.9087)	4.9074	0.01
	(0.9250,0.9273)	0.9268	0.07	(4.9047,4.9083)	4.9074	0.02

Table B.6: Results for the single type two-indenture model, $\lambda = 1, \mu_1 = J, \mu_2 = 2J$

J	S_0	S_1	A_{exact}	A_{appr}	% error	$E\underline{j}_{exact}$	$E\underline{j}_{appr}$	% error
3	0	1	0.3750	0.3609	3.74	2.0182	1.9917	1.31
3	0	3	0.4886	0.4702	3.77	2.2509	2.2144	1.62
3	0	5	0.5356	0.5200	2.93	2.3473	2.3159	1.34
3	1	1	0.5288	0.5205	1.57	2.2529	2.2395	0.60
3	1	3	0.6410	0.6273	2.14	2.4504	2.4271	0.95
3	1	5	0.6970	0.6833	1.97	2.5492	2.5256	0.92
3	3	1	0.6822	0.6802	0.30	2.4950	2.4921	0.12
3	3	3	0.7557	0.7514	0.57	2.6148	2.6083	0.25
3	3	5	0.7996	0.7947	0.61	2.6866	2.6789	0.28
3	4	1	0.7271	0.7262	0.13	2.5664	2.5651	0.05
3	4	3	0.7856	0.7834	0.28	2.6606	2.6574	0.12
3	4	5	0.8227	0.8201	0.32	2.7204	2.7164	0.15
5	0	1	0.3073	0.2948	4.06	3.6315	3.6037	0.77
5	0	3	0.4219	0.4045	4.12	3.9557	3.9120	1.11
5	0	5	0.4761	0.4601	3.36	4.1102	4.0684	1.02
5	1	1	0.4559	0.4481	1.73	3.9167	3.9021	0.37
5	1	3	0.5743	0.5610	2.32	4.1888	4.1615	0.65
5	1	5	0.6377	0.6240	2.15	4.3355	4.3063	0.67
5	3	1	0.6172	0.6151	0.33	4.2369	4.2335	0.08
5	3	3	0.6996	0.6953	0.61	4.4069	4.3992	0.17
5	3	5	0.7507	0.7457	0.66	4.5128	4.5035	0.21
5	4	1	0.6670	0.6660	0.15	4.3364	4.3349	0.04
5	4	3	0.7339	0.7317	0.30	4.4721	4.4683	0.09
5	4	5	0.7775	0.7748	0.35	4.5609	4.5560	0.11
10	0	1	0.2302	0.2198	4.53	7.9035	7.8765	0.34
10	0	3	0.3381	0.3228	4.55	8.3663	8.3180	0.58
10	0	5	0.3968	0.3819	3.76	8.6226	8.5716	0.59
10	1	1	0.3631	0.3561	1.93	8.2572	8.2423	0.18
10	1	3	0.4824	0.4703	2.51	8.6497	8.6196	0.35
10	1	5	0.5525	0.5397	2.31	8.8831	8.8489	0.38
10	3	1	0.5246	0.5227	0.36	8.6995	8.6960	0.04
10	3	3	0.6152	0.6111	0.66	8.9586	8.9500	0.10
10	3	5	0.6746	0.6699	0.70	9.1294	9.1186	0.12
10	4	1	0.5785	0.5776	0.16	8.8476	8.8459	0.02
10	4	3	0.6543	0.6521	0.33	9.0595	9.0551	0.05
10	4	5	0.7060	0.7034	0.37	9.2047	9.1991	0.06

Appendix C

Additional theory

In this appendix the theory from Subsection 3.2.3 is extended for the cases $S_1 = 0$ and/or $S_2 = 0$. It is about the determination of the values for $EW_l^a(\mathbf{k})$, $q_1(0, k_2)$ and $q_2(k_1, 0)$. Since the derivation of the formulas is quite straight forward, we will only present the resulting formulas.

C.1 Determining the values for $EW_l^a(\mathbf{k})$, $q_1(0, k_2)$ and $q_2(k_1, 0)$ for $S_1 = 0$ and $S_2 > 0$

For $k_1 > 0$ and $k_2 > 0$:

$$EW_1^a(k_1, k_2) = \frac{k_1 + k_2 + S_2 + 1}{\mu_1}, \tag{C.1}$$

$$EW_2^a(k_1, k_2) = \sum_{y=k_2+1}^{k_1+k_2+1} (H_2(k_1, k_2 + S_2, y) - H_2(k_1, k_2 + S_2, y - 1)) \frac{y}{\mu_1}, \tag{C.2}$$

where,

$$H_2(k_1, k_2 + S_2, y) = \sum_{x=k_2+1}^{min(y,k_2+S_2)} \frac{\binom{k_1}{y-x}\binom{k_2+S_2}{x}}{\binom{k_1+k_2+S_2}{y}}.$$

For $k_1 = 0$ and $k_2 = 0$:

$$EW_1^a(0, 0) = \sum_{n_{12}=0}^{S_2} \frac{n_{12} + 1}{\mu_1} \tilde{P}_1(0, n_{12}), \tag{C.3}$$

$$EW_2^a(0, 0) = \frac{1}{\mu_1} \tilde{P}_1(0, S_2), \tag{C.4}$$

$$q_1(0, 0) = 1, \tag{C.5}$$

$$q_2(0, 0) = \tilde{P}_1(0, S_2). \tag{C.6}$$

For $k_1 > 0$ and $k_2 = 0$:

$$
\begin{aligned}
EW_1^a(k_1,0) &= \sum_{n_{12}=0}^{S_2} \frac{k_1+n_{12}+1}{\mu_1}\bar{P}_1(k_1,n_{12}), && \text{(C.7)}\\
EW_2^a(k_1,0) &= \sum_{y=1}^{k_1+1}(H_6(k_1,S_2,y)-H_6(k_1,S_2,y-1))\frac{y}{\mu_1}\bar{P}_1(k_1,S_2), && \text{(C.8)}\\
\text{where,}&\\
H_6(k_1,S_2,y) &= \sum_{x=1}^{min(y,S_2)} \frac{\binom{k_1}{y-x}\binom{S_2}{x}}{\binom{k_1+S_2}{y}},\\
q_2(k_1,0) &= \bar{P}_1(k_1,S_2). && \text{(C.9)}
\end{aligned}
$$

And for $k_1 = 0$ and $k_2 > 0$:

$$
\begin{aligned}
EW_1^a(0,k_2) &= \frac{k_2+S_2+1}{\mu_1}, && \text{(C.10)}\\
EW_2^a(0,k_2) &= \frac{k_2+1}{\mu_1}, && \text{(C.11)}\\
q_1(0,k_2) &= 1. && \text{(C.12)}
\end{aligned}
$$

C.2 Determining the values for $EW_l^a(\mathbf{k})$, $q_1(0,k_2)$ and $q_2(k_1,0)$ for $S_1 > 0$ and $S_2 = 0$

For $k_1 > 0$ and $k_2 > 0$:

$$
\begin{aligned}
EW_1^a(k_1,k_2) &= \sum_{y=k_1+1}^{k_1+k_2+1}(H_1(k_1+S_1,k_2,y)-H_1(k_1+S_1,k_2,y-1))\frac{y}{\mu_1}, && \text{(C.13)}\\
\text{where,}&\\
H_1(k_1+S_1,k_2,y) &= \sum_{x=k_1+1}^{min(y,k_1+S_1)} \frac{\binom{k_1+S_1}{x}\binom{k_2}{y-x}}{\binom{k_1+k_2+S_1}{y}},\\
EW_2^a(k_1,k_2) &= \frac{k_1+k_2+S_1+1}{\mu_1}. && \text{(C.14)}
\end{aligned}
$$

For $k_1 = 0$ and $k_2 = 0$:

$$
\begin{aligned}
EW_1^a(0,0) &= \frac{1}{\mu_1}\tilde{P}_1(S_1,0), && \text{(C.15)}\\
EW_2^a(0,0) &= \sum_{n_{11}=0}^{S_1}\frac{n_{11}+1}{\mu_1}\tilde{P}_1(n_{11},0), && \text{(C.16)}\\
q_1(0,0) &= \tilde{P}_1(S_1,0), && \text{(C.17)}\\
q_2(0,0) &= 1. && \text{(C.18)}
\end{aligned}
$$

For $k_1 > 0$ and $k_2 = 0$:

$$EW_1^a(k_1,0) = \frac{k_1+1}{\mu_1}, \tag{C.19}$$

$$EW_2^a(k_1,0) = \frac{k_1+S_1+1}{\mu_1}, \tag{C.20}$$

$$q_2(k_1,0) = 1. \tag{C.21}$$

And for $k_1 = 0$ and $k_2 > 0$:

$$EW_1^a(0,k_2) = \sum_{y=1}^{k_2+1} (H_7(S_1,k_2,y) - H_7(S_1,k_2,y-1))\frac{y}{\mu_1}\hat{P}_1(S_1,k_2), \tag{C.22}$$

where,

$$H_7(S_1,k_2,y) = \sum_{x=1}^{min(y,S_1)} \frac{\binom{S_1}{x}\binom{k_2}{y-x}}{\binom{k_2+S_1}{y}},$$

$$EW_2^a(0,k_2) = \sum_{n_{11}=0}^{S_1} \frac{n_{11}+k_2+1}{\mu_1}\hat{P}_1(n_{11},k_2), \tag{C.23}$$

$$q_1(0,k_2) = \hat{P}_1(S_1,k_2). \tag{C.24}$$

C.3 Determining the values for $EW_l^a(\mathbf{k})$, $q_1(0,k_2)$ and $q_2(k_1,0)$ for $S_1 = 0$ and $S_2 = 0$

For $k_1 \geq 0$ and $k_2 \geq 0$:

$$EW_1^a(k_1,k_2) = \frac{k_1+k_2+1}{\mu_1}, \tag{C.25}$$

$$EW_2^a(k_1,k_2) = \frac{k_1+k_2+1}{\mu_1}, \tag{C.26}$$

$$q_1(0,k_2) = 1, \tag{C.27}$$

$$q_2(k_1,0) = 1. \tag{C.28}$$

Appendix D

Auxiliary theorems

This appendix contains two auxiliary theorems, that have been used in the proof of Proposition 3.2.3.

Theorem D.1
For $Z \in \mathbb{N}^+$ and $S_1 \in \mathbb{N}$ the following holds:

$$\sum_{y=1}^{Z} \frac{(S_1 + Z - y)!}{(Z - y)!} = \frac{(S_1 + Z)!}{(Z-1)!(S_1 + 1)}. \tag{D.1}$$

Proof. By induction:
(Basis) The result is true for $Z = 1$, $S_1 \in \mathbb{N}$ since

$$\sum_{y=1}^{1} \frac{(S_1 + 1 - y)!}{(1 - y)!} = S_1! = \frac{(S_1 + 1)!}{(1-1)!(S_1 + 1)}.$$

(Induction hypothesis) Suppose the result is true when $Z = a \in \mathbb{N}^+$, $S_1 \in \mathbb{N}$, that is,

$$\sum_{y=1}^{a} \frac{(S_1 + a - y)!}{(a - y)!} = \frac{(S_1 + a)!}{(a-1)!(S_1 + 1)}.$$

Then

$$\begin{aligned}
\sum_{y=1}^{a+1} \frac{(S_1 + a + 1 - y)!}{(a + 1 - y)!} &= \sum_{y'=0}^{a} \frac{(S_1 + a - y')!}{(a - y')!} \\
&= \sum_{y'=1}^{a} \frac{(S_1 + a - y')!}{(a - y')!} + \frac{(S_1 + a)!}{a!} \\
&= \frac{(S_1 + a)!}{(a-1)!(S_1 + 1)} + \frac{(S_1 + a)!}{a!} \quad \text{(induction hypothesis)} \\
&= \frac{(S_1 + a)!a + (S_1 + a)!(S_1 + 1)}{a!(S_1 + 1)} \\
&= \frac{(S_1 + a + 1)!}{a!(S_1 + 1)}.
\end{aligned}$$

So the result is true when $Z = a + 1$, $S_1 \in \mathbb{N}$, and by the principle of induction, it is true for all $Z \in \mathbb{N}^+$, $S_1 \in \mathbb{N}$. □

Theorem D.2
For $k_1 \in \mathbb{N}$, $Z \in \mathbb{N}$ *and* $S_1 \in \mathbb{N}$ *the following holds:*

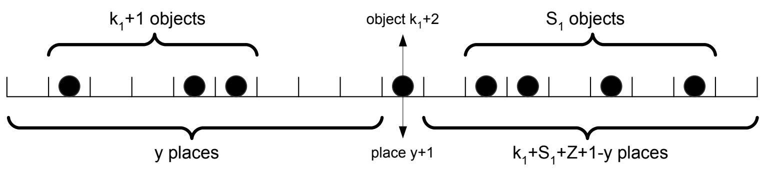
$$\sum_{y=k_1+1}^{k_1+Z+1} \binom{y}{k_1+1} \binom{k_1+S_1+Z+1-y}{S_1} = \binom{k_1+S_1+Z+2}{k_1+S_1+2}. \tag{D.2}$$

Proof. Consider k_1+S_1+2 identical objects and k_1+S_1+Z+2 ordered places. In how many ways can these k_1+S_1+2 objects be assigned to these k_1+S_1+Z+2 places?

(i) Directly:

$$\binom{k_1+S_1+Z+2}{k_1+S_1+2}.$$

(ii) Condition on the (k_1+2)th object. This object can be assigned to place $y+1 = k_1+2, \ldots, k_1+Z+2$.



The first k_1+1 objects are allocated to the first y places. The number of ways to do this is

$$\binom{y}{k_1+1}.$$

The last S_1 objects are allocated to the last $k_1+S_1+Z+1-y$ places. The number of ways to do this is

$$\binom{k_1+S_1+Z+1-y}{S_1}.$$

The number of ways in which the k_1+S_1+2 objects can be assigned to k_1+S_1+Z+2 places now equals

$$\sum_{y=k_1+1}^{k_1+Z+1} \binom{y}{k_1+1} \binom{k_1+S_1+Z+1-y}{S_1}.$$

Combining (i) and (ii) completes the proof. □

Appendix E

Program code

All algorithms developed and used for the research presented in this thesis, have been programmed in Maple. Representative program code is given in this appendix.

E.1 The simple two-echelon system with single server facilities (Section 2.1)

The Maple code that is presented in this section has been used to obtain the results from Subsection 2.1.4.

E.1.1 Exact solution by solving the balance equations

```
> Simple2echelonExact:=proc(J,p,S0,S1,lambda,mu0,mu1)
> local ZZ,n,m,G,A,SSP,f,x,Q,nm,JEx:
> # The 2-dimensional state space is represented by a 1-dimensional Markov chain
> ZZ:=(S0+1)*(J+S1+1)+(S1+J+1)*(S1+J)/2;
> Q:= matrix(ZZ,ZZ);
> f:= (N,M) -> 1+N+M*(J+S0+S1+1)-M*(M-1)/2;
> # Q is the generator matrix. At first it consists of 0's
> for n from 1 to ZZ do
>    for m from 1 to ZZ do
>       Q[n,m]:= 0;
>    od:
> od:
> # The transition rates are entered per region
> #--region I&II, north
> for n from 0 to S0 do
>    for m from 0 to J+S1-1 do
>       Q[f(n,m),f(n,m+1)]:=(J-max(m-S1,0))*p*lambda;
>    od:
> od:
> #--region I&II, west
> for n from 1 to S0 do
>    for m from 0 to J+S1 do
>       Q[f(n,m),f(n-1,m)]:=mu0;
>    od:
> od:
> #--region I&II, south
> for n from 0 to S0 do
>    for m from 1 to J+S1 do
>       Q[f(n,m),f(n,m-1)]:=mu1;
>    od:
> od:
> #--region I&II, east
> for n from 0 to S0-1 do
```

```
>    for m from 0 to J+S1 do
>       Q[f(n,m),f(n+1,m)]:=(J-max(m-S1,0))*(1-p)*lambda;
>    od:
> od:
> #--region III&IV, north
> for n from S0+1 to J+S0+S1-1 do
>    for m from 0 to J+S0+S1-n-1 do
>       Q[f(n,m),f(n,m+1)]:=(J-max(m+n-S0-S1,0))*p*lambda;
>    od:
> od:
> #--region III&IV, west
> for n from S0+1 to J+S0+S1 do
>    for m from 0 to J+S0+S1-n do
>       Q[f(n,m),f(n-1,m)]:=mu0;
>    od:
> od:
> #--region III&IV, south
> for n from S0+1 to J+S0+S1-1 do
>    for m from 1 to J+S0+S1-n do
>       Q[f(n,m),f(n,m-1)]:=mu1;
>    od:
> od:
> #--region III&IV, east
> for n from S0 to J+S0+S1-1 do
>    for m from 0 to J+S0+S1-n-1 do
>       Q[f(n,m),f(n+1,m)]:=(J-max(m+n-S0-S1,0))*(1-p)*lambda;
>    od:
> od:
> #--diagonal
> for x from 1 to ZZ do
>    Q[x,x]:= -sum(Q[x,y],y=1..ZZ);
> od:
>
> # Now solve
> # We have to find pi such that pi*Q=0
> # That is: (pi*Q)^T=0^T=0
> # Or: Q^T*pi^T=0
> # Or: find pi^T by obtaining the kernel of matrix Q^T
>
> #--find kernel
> SSP:=(kernel(transpose(Q)))[1]:
> #--normalize
> G:=sum( SSP[i],i=1..ZZ): SSP:=scalarmul(SSP,1/G):
>
> # Determine performance measures
> # availability
> A:=0:
> for n from 0 to S0 do
>    for m from 0 to S1 do
>       A:=A+SSP[f(n,m)];
>    od:
> od:
> for n from S0+1 to S0+S1 do
>    for m from 0 to S0+S1-n do
>       A:=A+SSP[f(n,m)];
>    od:
> od:
> # expected value j
> JEx:=A*J:
> for n from 0 to S0 do
>    for m from S1+1 to J+S1 do
>       JEx:=JEx+(J+S1-m)*SSP[f(n,m)];
>    od:
> od:
> for nm from S0+S1+1 to S0+S1+J do
>    for n from S0+1 to nm do
>       JEx:=JEx+(J+S0+S1-nm)*SSP[f(n,nm-n)];
>    od:
```

```
> od:
> print(evalf(A),evalf(JEx));
> end:
```

E.1.2 Approximation by use of Lemma 2.1.2

```
> # Determine throughput subnetwork
> Throughput:=proc(J,p,S0,S1,lambda,mu0,mu1)
> local i, P, G,Tb,Td:
> P:=array(1..J+S1+1,[]);
> for i from 0 to S1 do
>    P[i+1]:=(p*lambda/mu1)^i*J^i;
> od:
> for i from S1+1 to S1+J do
>    P[i+1]:=(p*lambda/mu1)^i*J^S1*J!/((J+S1-i)!);
> od:
> G:=sum('P[k+1]', 'k'=0..J+S1);
> for i from 0 to J+S1 do
>    P[i+1]:=P[i+1]/G;
> od:
> evalm(P);
> Tb:=(1-P[0+1])*mu1;
> Td:=(1-p)/p*Tb;
> end:
> # Determine q by Norton's theorem
> Q:=proc(J,p,S0,S1,lambda,mu0,mu1)
> local delta,q:
> delta:=Throughput(J,p,S0,S1,lambda,mu0,mu1)/mu0;
> q:=delta^S0*(1-delta)/(1-delta^(S0+1));
> end:
> # Main
> Simple2echelonAppr:=proc(J,p,S0,S1,lambda,mu0,mu1)
> local alpha, beta, P, q, k,m,G,A,km,JAp;
> alpha:=p*lambda/mu1;
> beta:=(1-p)*lambda/mu0;
> P[0,0]:=1/(J!*J^S1);
> q:=Q(J,p,S0,S1,lambda,mu0,mu1);
> for m from 1 to S1 do
>    P[m,0]:=P[m-1,0]*alpha*J;
> od:
> for m from S1+1 to S1+J do
>    P[m,0]:=P[m-1,0]*alpha*(S1+J-m+1);
> od:
> for k from 1 to S1 do
>    P[0,k]:=q/(J!*J^(S1-k))*beta^k;
> od:
> for k from S1+1 to J+S1 do
>    P[0,k]:=q/((S1+J-k)!)*beta^k;
> od:
> for k from 1 to J+S1 do
>    for m from 1 to S1-k do
>       P[m,k]:=P[m-1,k]*alpha*J;
>    od:
>    for m from max(1,S1-k+1) to S1+J-k do
>       P[m,k]:=P[m-1,k]*alpha*(S1+J-k-m+1);
>    od:
> od:
> G:=0:
> for k from 0 to J+S1 do
>    for m from 0 to J+S1-k do
>       G:=G+P[m,k];
>    od:
> od:
> for k from 0 to J+S1 do
>    for m from 0 to J+S1-k do
>       P[m,k]:=P[m,k]/G;
>    od:
> od:
```

```
> # Determine performance measures
> # availability
> A:=0:
> for k from 0 to S1 do
>    for m from 0 to S1-k do
>       A:=A+P[m,k];
>    od:
> od:
> # expected value j
> JAp:=A*J:
> for km from S1+1 to S1+J do
>    for k from 0 to km do
>       JAp:=JAp+(J+S1-km)*P[km-k,k];
>    od:
> od:
> print(evalf(A),evalf(JAp));
> end:
```

E.2 General two-echelon repairable item systems (Section 2.2)

The section concerns the Maple code that was used to obtain the approximation results from Subsection 2.2.3.

E.2.1 Approximation by use of Algorithm 2.2.2

Only the program code for the case $L = 2$ is presented here.

```
> Digits:=30:
> # Determine throughput subnetwork by applying MDA
> # transport is server 1 and at the same time server 0
> # baserepair is server 2
> # operationalmachines is server 3
> ThroughputMDA:=proc(J1,p1,S1,lambda1,mu1,R1,gamma1)
> local N, mu,c,V,P,n,j,EW,TH,k:
> # step 1
> N:=J1+S1:
> c[1]:=J1+S1: c[2]:=R1: c[3]:=J1:
> mu[1]:=gamma1: mu[2]:=mu1: mu[3]:=lambda1:
> V[1]:=1: V[2]:=p1/(1-p1): V[3]:=1/(1-p1):
> P[1,0,0]:=1: P[2,0,0]:=1: P[3,0,0]:=1:
> # step 2
> for n from 1 to N do
>    n;
> # step 3
>    for j from 1 to 3 do
>       if c[j]<=n-1 then
>         EW[j,n]:=sum((k-c[j]+1)/(c[j]*mu[j])*P[j,k,n-1],k=c[j]..n-1)+(1/mu[j]);
>       else
>         EW[j,n]:=(1/mu[j]);
>       fi:
>    od:
> # step 4
>    j:='j':
>    TH[1,n]:=n/(sum(V[j]*EW[j,n],j=1..3));
>    for j from 2 to 3 do
>       TH[j,n]:=V[j]*TH[1,n]:
>    od:
> # step 5
>    for k from 1 to n do
>       for j from 1 to 3 do
>          P[j,k,n]:=TH[j,n]/(mu[j]*min(c[j],k))*P[j,k-1,n-1];
```

```
>       od:
>    od:
>    k:='k':
>    for j from 1 to 3 do
>       P[j,0,n]:=1-sum(P[j,k,n],k=1..n);
>    od:
> od:
> TH[1,N];
> end:
> # Determine q by applying Norton's theorem
> Q:=proc(J,p,S,lambda,mu,R,gamma,M)
> local i,delta,q,numerator,denominator:
> delta:=0:
> for i from 1 to M do
>    delta:=delta+ThroughputMDA(J[i],p[i],S[i],lambda[i],mu[i],R[i],gamma[i]);
> od:
> delta:=delta/mu[M+1];
> numerator:=delta^(S[M+1])/product(min(k,R[M+1]),k=1..S[M+1]);
> denominator:=sum(delta^n/product(min(k,R[M+1]),k=1..n),n=0..S[M+1]);
> q:=numerator/denominator;
> end:
> # Main
> Multibase2echelonAppr:=proc(J,p,S,lambda,Mu,R,gamma,M)
> local q, i, l, V, N, NN,nn,r,k,j,n,EW,e,P,c,mu,TH:
> global A,EJ;
> q:=Q(J,p,S,lambda,Mu,R,gamma,M):
> # Define Visitratios
> # class l belongs to base l
> # station 0 is the depot
> # station l (l=1..M) is operational machines base l
> # station l (l=M+1..2M) is repair base l-M
> # station l (l=2M+1..3M) is transport depot -> base l-2M
> c[0]:=R[M+1];
> mu[0]:=Mu[M+1];
> for i from 1 to M do
>    c[i]:=J[i];
>    mu[i]:=lambda[i]
> od:
> for i from M+1 to 2*M do
>    c[i]:=R[i-M];
>    mu[i]:=Mu[i-M];
> od:
> for i from 2*M+1 to 3*M do
>    c[i]:=J[i-2*M]+S[i-2*M];
>    mu[i]:=gamma[i-2*M];
> od:
> for i from 0 to 3*M do
>    for l from 1 to M do
>       V[i,l]:=0:
>    od:
> od:
> NN:=0:
> for l from 1 to M do
>    V[0,l]:=1:
>    V[l,l]:=1/(1-p[l]):
>    V[M+l,l]:=p[l]/(1-p[l]):
>    V[2*M+l,l]:=1:
>    N[l]:=J[l]+S[l]:
>    NN:=NN+N[l]:
> od:
>
> for j from 0 to 3*M do
>    P[j,0,0,0]:=1:
> od:
>
> for n from 1 to NN do
>
>    for nn[1] from max(0,n-NN+N[1]) to min(n,N[1]) do
```

```
>       nn[M]:=n-sum('nn[i]','i'=1..M-1):
>       k:='k':
>       # r=1
>       if V[0,1]>0 and nn[1]>0 then
>         EW[0,1,nn[1],nn[2]]:=sum((k+1)/(min(c[0],S[M+1]+k+1)*mu[0])*P[0,k,nn[1]-1,nn[2]],k=1..n-1)+
          q/(min(c[0],S[M+1]+1)*mu[0])*P[0,0,nn[1]-1,nn[2]];
>       fi:
>       # r=2
>       if V[0,2]>0 and nn[2]>0 then
>         EW[0,2,nn[1],nn[2]]:=sum((k+1)/(min(c[0],S[M+1]+k+1)*mu[0])*P[0,k,nn[1],nn[2]-1],k=1..n-1)+
          q/(min(c[0],S[M+1]+1)*mu[0])*P[0,0,nn[1],nn[2]-1];
>       fi:
>
>       for j from 1 by M to 3*M do
>          # r=1
>          if V[j,1]>0 and nn[1]>0 then
>            if c[j]<=n-1 then
>              EW[j,1,nn[1],nn[2]]:=sum((k+1-c[j])/(c[j]*mu[j])*P[j,k,nn[1]-1,nn[2]],k=c[j]..n-1)+1/mu[j];
>            else
>              EW[j,1,nn[1],nn[2]]:=1/mu[j];
>            fi:
>          fi:
>       od:
>
>       for j from 2 by M to 3*M do
>          # r=2
>          if V[j,2]>0 and nn[2]>0 then
>            if c[j]<=n-1 then
>              EW[j,2,nn[1],nn[2]]:=sum((k+1-c[j])/(c[j]*mu[j])*P[j,k,nn[1],nn[2]-1],k=c[j]..n-1)+1/mu[j];
>            else
>              EW[j,2,nn[1],nn[2]]:=1/mu[j];
>            fi:
>          fi:
>       od:
>
>       j:='j':
>       for r from 1 to M do
>          if nn[r]>0 then
>            TH[0,r,nn[1],nn[2]]:=nn[r]/(V[0,r]*EW[0,r,nn[1],nn[2]]+V[r,r]*EW[r,r,nn[1],nn[2]]+
              V[r+M,r]*EW[r+M,r,nn[1],nn[2]]+V[r+2*M,r]*EW[r+2*M,r,nn[1],nn[2]]):
>          else
>            TH[0,r,nn[1],nn[2]]:=0:
>          fi:
>       od:
>
>       for j from 1 to M do
>          TH[j,j,nn[1],nn[2]]:=V[j,j]*TH[0,j,nn[1],nn[2]]:
>       od:
>       for j from M+1 to 2*M do
>          TH[j,j-M,nn[1],nn[2]]:=V[j,j-M]*TH[0,j-M,nn[1],nn[2]]:
>       od:
>       for j from 2*M+1 to 3*M do
>          TH[j,j-2*M,nn[1],nn[2]]:=V[j,j-2*M]*TH[0,j-2*M,nn[1],nn[2]]:
>       od:
>
>       P[0,1,nn[1],nn[2]]:=q/(mu[0]*min(c[0],S[M+1]+1))*(TH[0,1,nn[1],nn[2]]*P[0,0,nn[1]-1,nn[2]]+
          TH[0,2,nn[1],nn[2]]*P[0,0,nn[1],nn[2]-1]);
>
>       for k from 2 to n do
>          P[0,k,nn[1],nn[2]]:=(TH[0,1,nn[1],nn[2]]*P[0,k-1,nn[1]-1,nn[2]]+
            TH[0,2,nn[1],nn[2]]*P[0,k-1,nn[1],nn[2]-1])/(mu[0]*min(c[0],k+S[M+1]));
>       od:
>
>       for j from 1 by M to 3*M do
>          for k from 1 to n do
>             P[j,k,nn[1],nn[2]]:=(TH[j,1,nn[1],nn[2]]*P[j,k-1,nn[1]-1,nn[2]])/(mu[j]*min(c[j],k));
>          od:
>       od:
```

```
>       for j from 2 by M to 3*M do
>          for k from 1 to n do
>             P[j,k,nn[1],nn[2]]:=(TH[j,2,nn[1],nn[2]]*P[j,k-1,nn[1],nn[2]-1])/(mu[j]*min(c[j],k));
>          od:
>       od:
>
>       k:='k':
>       for j from 0 to 3*M do
>          P[j,0,nn[1],nn[2]]:=1-sum(P[j,k,nn[1],nn[2]],k=1..n);
>          if P[j,0,nn[1],nn[2]]<0 then
>            error(j,0,nn[1],nn[2],`rounding error, increase number of Digits`);
>          fi:
>       od:
>    od:
> od:
>
> # Determine performance measures
> for i from 1 to M do
>    # availability
>    A[i]:=0:
>    for k from J[i] to N[i] do
>       A[i]:=A[i]+P[i,k,N[1],N[2]]:
>    od:
>    # expected value j
>    EJ[i]:=A[i]*J[i]:
>    for k from 1 to J[i]-1 do
>       EJ[i]:=EJ[i]+k*P[i,k,N[1],N[2]]:
>    od:
> od:
> # Show results, remove when optimizing
> for i from 1 to M do
> print(A[i],EJ[i]);
> od:
>
> end:
```

E.3 The two-type two-indenture system with single server facilities (Section 3.2)

The approximation results from Subsection 3.2.6 have been obtained with the Maple code presented here.

E.3.1 Approximation by use of Algorithm 3.2.2

```
> Digits:=30:
> # Determine throughput subnetwork
> Throughput:=proc(J,S0,lambda,mu2,tot)
> local i, P, G,Tb:
> P:=array(1..tot+1,[]);
> P[0+1]:=1:
> for i from 1 to tot do
>    P[i+1]:=P[i]*min(tot-i+1,J)*lambda/mu2;
> od:
> G:=sum('P[k+1]', 'k'=0..tot);
> Tb:=(1-P[0+1]/G)*mu2;
> return evalf(Tb);
> end:
> EW00:=proc(J,S0,S1,S2,lambda,mu2,mu1,r1,r2)
> global f,EW11,EW12,Q01,Q02,SSP;
> local  ZZ, Q, s1,s2,Lambda,G,n1,n2,H1,H2,TH,SSP00;
> f:= (nn1,nn2) -> 1+nn1+nn2*(S1+1+J+S0)-(max(0,nn2-S2-1)*(max(0,nn2-S2-1)+1)/2);
> ZZ:= (S1+S0+J+1)*(S2+S0+J+1)-(J+S0+1)*(J+S0)/2;
> Q:= matrix(ZZ,ZZ);
```

```
> Lambda:= Throughput(J,S0,lambda,mu2,J+S0);
> for s1 from 1 to ZZ do
>    for s2 from 1 to ZZ do
>       Q[s1,s2]:= 0;
>    od:
> od:
> for s1 from 1 to J+S0+S1 do
>    Q[f(s1,0),f(s1-1,0)]:=mu1;
> od:
> for s2 from 1 to J+S0+S2 do
>    Q[f(0,s2),f(0,s2-1)]:=mu1;
> od:
> for s1 from 1 to S1+J+S0 do
>    for s2 from 1 to S2 do
>       Q[f(s1,s2),f(s1-1,s2)]:=s1/(s1+s2)*mu1;
>       Q[f(s1,s2),f(s1,s2-1)]:=s2/(s1+s2)*mu1;
>    od:
> od:
> for s2 from S2+1 to S2+J+S0 do
>    for s1 from 1 to S1+J+S0+S2-s2 do
>       Q[f(s1,s2),f(s1-1,s2)]:=s1/(s1+s2)*mu1;
>       Q[f(s1,s2),f(s1,s2-1)]:=s2/(s1+s2)*mu1;
>    od:
> od:
>
> # k1=k2=0
> for s1 from 0 to S1 do
>    for s2 from 0 to S2 do
>       Q[f(s1,s2),f(s1+1,s2)]:=Lambda*r1;
>       Q[f(s1,s2),f(s1,s2+1)]:=Lambda*r2;
>    od:
> od:
> # k1>0, k2=0
> for s1 from S1+1 to J+S0+S1-1 do
>    for s2 from 0 to S2 do
>       TH:=Throughput(J,S0,lambda,mu2,J+S0+S1-s1);
>       Q[f(s1,s2),f(s1+1,s2)]:=TH*r1;
>       Q[f(s1,s2),f(s1,s2+1)]:=TH*r2;
>    od:
> od:
> # k1=0,k2>0
> for s2 from S2+1 to J+S0+S2-1 do
>    for s1 from 0 to S1 do
>       TH:=Throughput(J,S0,lambda,mu2,J+S0+S2-s2);
>       Q[f(s1,s2),f(s1+1,s2)]:=TH*r1;
>       Q[f(s1,s2),f(s1,s2+1)]:=TH*r2;
>    od:
> od:
> # k1>0,k2>0
> for s1 from S1+1 to J+S0+S1-2 do
>    for s2 from S2+1 to J+S0+S2+S1-s1-1 do
>       TH:=Throughput(J,S0,lambda,mu2,J+S0+S2+S1-s2-s1);
>       Q[f(s1,s2),f(s1+1,s2)]:=TH*r1;
>       Q[f(s1,s2),f(s1,s2+1)]:=TH*r2;
>    od:
> od:
> #--diagonal
> s2:= 's2':
> for s1 from 1 to ZZ do
>    Q[s1,s1]:= -sum(Q[s1,s2],s2=1..ZZ);
> od:
> #--find kernel
> SSP:= (kernel(transpose(Q)))[1]:
> #--normalize
> G:=0:
> for s1 from 0 to S1 do
>    for s2 from 0 to S2 do
>       G:= G+SSP[f(s1,s2)]:
```

```
>    od:
> od:
> SSP00:= scalarmul(SSP,1/G):
> s1:='s1':
> s2:='s2':
> # Deterine Q11
> Q01:=evalf(sum(SSP00[f(S1,s2)],s2=0..S2));
> # Determine Q12
> Q02:=evalf(sum(SSP00[f(s1,S2)],s1=0..S1));
>
> # Determine EW11[0,0]
> EW11[0,0]:=0:
> for n2 from 0 to S2 do
>    EW11[0,0]:=EW11[0,0]+evalf(1/(S1+1)*(S1+n2+1)/mu1)*SSP00[f(S1,n2)];
> od:
>
> # Determine EW12[0,0]
> EW12[0,0]:=0:
> for n1 from 0 to S1 do
>    EW12[0,0]:=EW12[0,0]+evalf(1/(S2+1)*(S2+n1+1)/mu1)*SSP00[f(n1,S2)];
> od:
> end:
> EWk0:=proc(J,S0,S1,S2,lambda,mu2,mu1,r1,r2)
> global EW11,EW12,Q:
> local H1,H2,n1,n2,G,k,Throughput,P:
> for k from 1 to S0+J do
>    for n2 from 0 to S2 do
>       P[k,n2]:=SSP[f(S1+k,n2)];
>    od:
>    n2:='n2':
>    G:=sum(P[k,n2],n2=0..S2):
>    EW11[k,0]:=0:
>    for n2 from 0 to S2 do
>       EW11[k,0]:=EW11[k,0]+evalf((k+1)/(k+S1+1)*(k+S1+n2+1)/mu1)*P[k,n2]/G;
>    od:
>    Q[k,0]:=P[k,S2]/G;
>    EW12[k,0]:=evalf(1/(S2+1)*(S2+k+S1+1)/mu1)*Q[k,0];
> od:
> end:
> EW0k:=proc(J,S0,S1,S2,lambda,mu2,mu1,r1,r2)
> global EW11,EW12,Q:
> local H1,H2,n1,n2,G,k,Throughput,P:
> for k from 1 to S0+J do
>    for n1 from 0 to S1 do
>       P[n1,k]:=SSP[f(n1,S2+k)]:
>    od:
>    n1:='n1':
>    G:=sum(P[n1,k],n1=0..S1):
>    EW12[0,k]:=0:
>    for n1 from 0 to S1 do
>       EW12[0,k]:=EW12[0,k]+evalf((k+1)/(k+S2+1)*(k+S2+n1+1)/mu1)*P[n1,k]/G;
>    od:
>    Q[0,k]:=P[S1,k]/G;
>    EW11[0,k]:=evalf(1/(S1+1)*(S1+S2+k+1)/mu1)*Q[0,k];
> od:
> end:
> EWkk:=proc(J,S0,S1,S2,lambda,mu2,mu1,r1,r2,n)
> global EW11,EW12:
> local H1,H2,k1,k2:
> for k1 from 1 to n-2 do
>    for k2 from 1 to n-1-k1 do
>       EW11[k1,k2]:=(k1+1)/(k1+S1+1)*(k1+k2+S1+S2+1)/mu1;
>    od:
> od:
> for k1 from 1 to n-2 do
>    for k2 from 1 to n-1-k1 do
>       EW12[k1,k2]:=(k2+1)/(k2+S2+1)*(k1+k2+S1+S2+1)/mu1;
>    od:
```

```
> od:
> end:
> # Main
> IndentureAppr:=proc(J,S0,S1,S2,lambda,mu2,mu1,r1,r2)
> local V,p,n,N,EW,k1,k2,k,TH,A,EJ,s,AExact,GG,s1,s2,SSPxx:
> N:=J+S0:
> EW00(J,S0,S1,S2,lambda,mu2,mu1,r1,r2):
> EW0k(J,S0,S1,S2,lambda,mu2,mu1,r1,r2):
> EWk0(J,S0,S1,S2,lambda,mu2,mu1,r1,r2):
> EWkk(J,S0,S1,S2,lambda,mu2,mu1,r1,r2,N):
>
> # MDA algorithm
>
> V[0]:=1: V[11]:=r1: V[12]:=r2: V[2]:=1:
> p[0,0,0]:=1:
> p[1,0,0,0]:=1:
> p[2,0,0]:=1:
>
> for n from 1 to N do
>    k1:='k1':
>    k2:='k2':
>    k:='k':
>    if J<=n-1 then
>      EW[0,n]:=sum((k-J+1)/(J*lambda)*p[0,k,n-1],k=J..n-1)+1/lambda;
>    else
>      EW[0,n]:=1/lambda;
>    fi:
>    EW[11,n]:=sum(sum(EW11[k1,k2]*p[1,k1,k2,n-1],k1=0..n-1-k2),k2=0..n-1);
>    EW[12,n]:=sum(sum(EW12[k1,k2]*p[1,k1,k2,n-1],k1=0..n-1-k2),k2=0..n-1);
>    EW[2,n]:=sum(k/mu2*p[2,k,n-1],k=1..n-1)+1/mu2;
>
>    TH[0,n]:=n/(V[0]*EW[0,n]+V[11]*EW[11,n]+V[12]*EW[12,n]+V[2]*EW[2,n]);
>    TH[11,n]:=V[11]*TH[0,n];
>    TH[12,n]:=V[12]*TH[0,n];
>    TH[2,n]:=V[2]*TH[0,n];
>
>    for k from 1 to n do
>       p[0,k,n]:=p[0,k-1,n-1]*TH[0,n]/(lambda*min(J,k));
>    od:
>    k:='k':
>    p[0,0,n]:=1-sum(p[0,k,n],k=1..n);
>    p[1,1,0,n]:=p[1,0,0,n-1]*TH[11,n]*Q01/mu1:
>    p[1,0,1,n]:=p[1,0,0,n-1]*TH[12,n]*Q02/mu1:
>    if n>1 then
>      p[1,1,1,n]:=(p[1,0,1,n-1]*TH[11,n]*Q[0,1]+p[1,1,0,n-1]*TH[12,n]*Q[1,0])/mu1:
>    fi:
>    for k from 2 to n do
>       p[1,k,0,n]:=p[1,k-1,0,n-1]*TH[11,n]/mu1:
>       p[1,0,k,n]:=p[1,0,k-1,n-1]*TH[12,n]/mu1:
>    od:
>    for k from 2 to n-1 do
>       p[1,k,1,n]:=(p[1,k-1,1,n-1]*TH[11,n]+p[1,k,0,n-1]*TH[12,n]*Q[k,0])/mu1;
>       p[1,1,k,n]:=(p[1,0,k,n-1]*TH[11,n]*Q[0,k]+p[1,1,k-1,n-1]*TH[12,n])/mu1;
>    od:
>    for k1 from 2 to n-2 do
>       for k2 from 2 to n-k1 do
>          p[1,k1,k2,n]:=(p[1,k1-1,k2,n-1]*TH[11,n]+p[1,k1,k2-1,n-1]*TH[12,n])/mu1:
>       od:
>    od:
>    k1:='k1':
>    k2:='k2':
>    p[1,0,0,n]:=0:
>    p[1,0,0,n]:=1-sum(sum(p[1,k1,k2,n],k1=0..n-k2),k2=0..n):
>
>    for k from 1 to n do
>       p[2,k,n]:=p[2,k-1,n-1]*TH[2,n]/(mu2);
>    od:
>    k:='k':
```

```
>    p[2,0,n]:=1-sum(p[2,k,n],k=1..n);
> od:
>
> # Determine performance measures
> # availablity
> A:=0:
> for k from J to N do
>    A:=A+p[0,k,N]:
> od:
> # expected value j
> EJ:=A*J:
> for k from 1 to J-1 do
>    EJ:=EJ+k*p[0,k,N]:
> od:
> print(evalf(A),evalf(EJ));
> end:
```

E.4 Optimization (Chapter 4)

In this section the Maple code is presented that was used to obtain results for the optimization algorithms as have been presented in Section 4.4.

E.4.1 Exact solution by Algorithm 4.2.1

```
> Digits:=30;
> # Determine total availability
> TotalA:=proc(J,p,S,lambda,Mu,R,gamma,M)
> local TotalAvailability;
> Multibase2echelonAppr(J,p,S,lambda,Mu,R,gamma,M);
> TotalAvailability:=sum(A[i]*lambda[i]*J[i],i=1..M)/sum(lambda[i]*J[i],i=1..M);
> end:
> # Define arg max
> argmax:=proc()
> local r,i,s;
> r:=args[1];
> s:=1;
> for i from 2 to nargs do
>    if args[i] > r then
>      r:= args[i];
>      s:=i;
>    fi:
> od:
> s;
> end:
> # Main
> Optimize_exact:=proc(J,p,c,lambda,Mu,R,gamma,M,C)
> local x1,x2,x3,i,TA_opt,TA,SS_opt,SS;
> # step 1
> SS:=[];
> for x3 from 0 to trunc(C/c[3]) do
>    for x2 from 0 to trunc((C-c[3]*x3)/c[2])do
>       x1:=trunc((C-c[3]*x3-c[2]*x2)/c[1]);
>       SS:=[op(SS),[x1,x2,x3]];
>    od:
> od:
> # step 2
> TA:=[];
> for i from 1 to nops(SS) do
>    TA:=[op(TA),TotalA(J,p,SS[i],lambda,Mu,R,gamma,M)];
> od:
> # step 3
> TA_opt:=argmax(op(TA));
> SS_opt:=SS[TA_opt];
> # show results
```

```
> print(SS_opt,TA[TA_opt]);
> end:
```

E.4.2 Approximation by Algorithm 4.3.1

```
> Optimize_greedy:=proc(J,p,c,lambda,Mu,R,gamma,M,C)
> local CC,SS,SS1,TA,continue,Delta,l,ll,s;
> # step 1
> CC:=0:
> SS:=Vector[row](1..M+1,0):
> TA[old]:=TotalA(J,p,SS,lambda,Mu,R,gamma,M);
> continue:=true;
> # step 2
> while continue=true do
>      Delta:=[]:
>      for l from 1 to M+1 do
>         if CC+c[l]<=C then
>           SS1:=evalm(SS):
>           SS1[l]:=SS1[l]+1:
>           TA[l]:=TotalA(J,p,SS1,lambda,Mu,R,gamma,M);
>           Delta:=[op(Delta),(TA[l]-TA[old])/c[l]];
>         else
>           Delta:=[op(Delta),0]:
>         fi:
>      od:
>      if max(op(Delta))=0 then
>        continue:=false;
>      else
>        ll:=argmax(op(Delta)):
>        SS[ll]:=SS[ll]+1:
>        CC:=CC+c[ll]:
>        TA[old]:=TA[ll]:
>      fi:
> od:
> # show results
> print(SS,TA[old]);
> end:
```

Appendix F

List of symbols

All symbols used throughout this thesis are presented here. As already pointed out in Section 1.5 there is a separate list of symbols for each chapter. Although some symbols occur in more than one chapter, the meaning of the symbol does not necessarily have to be the same. Random variables are denoted by a dash under the symbol.

Symbols used in Chapter 2

Parameters

L	Number of bases
J_l	Maximum number of machines operating at the production cell of base l, with $l = 1, \dots, L$
λ_l	Failure rate of a machine operating at base l, with $l = 1, \dots, L$
p_l	Probability that a broken machine from base l can be repaired locally ($l = 1, \dots, L$)
μ_0	Repair rate at central repair facility
μ_l	Repair rate at repair facility of base l, with $l = 1, \dots, L$
R_0	Number of repairmen at central repair facility
R_l	Number of repairmen at repair facility of base l, with $l = 1, \dots, L$
S_0	Number of spare machines dedicated to central repair facility
S_l	Number of spare machines dedicated to base l, with $l = 1, \dots, L$
γ_l	Transport rate from the central repair facility to base l, with $l = 1, \dots, L$

Random variables

$\underline{n}_1$	Number of machines in depot repair
$\underline{n}_2$	Number of spare machines at the depot
$\underline{k}_{0l}$	Number of backorders at the depot, originating from base l, with $l = 1, \dots, L$
$\underline{\mathbf{k}}_0 = (\underline{k}_{01}, \dots, \underline{k}_{0L})$	Vector that denotes the number of backorders at the depot
$\underline{k}$	Total number of backorders at the depot, equals $\sum_{l=1}^{L} \underline{k}_{0l}$

Random variables (continued)

$\underline{m}_{l1}$	Number of machines in repair at base l, with $l = 1, \ldots, L$
$\underline{\mathbf{m}}_1 = (\underline{m}_{11}, \ldots, \underline{m}_{L1})$	Vector that denotes the number of machines in base repair
$\underline{m}_{l2}$	Number of spare machines at base l, with $l = 1, \ldots, L$
$\underline{\mathbf{m}}_2 = (\underline{m}_{12}, \ldots, \underline{m}_{L2})$	Vector that denotes the number of spare machines at the bases
$\underline{t}_l$	Number of machines in transit from the central repair facility to base l, with $l = 1, \ldots, L$
$\underline{\mathbf{t}} = (\underline{t}_1, \ldots, \underline{t}_L)$	Vector that denotes the number of machines in transit to the bases
$\underline{j}_l$	Number of machines operational at base l, with $l = 1, \ldots, L$
$\underline{\mathbf{j}} = (\underline{j}_1, \ldots, \underline{j}_L)$	Vector that denotes the number of machines operational at the bases
$\underline{b}_l$	$\underline{m}_{l2} + \underline{j}_l$ with $l = 1, \ldots, L$
$\underline{\mathbf{b}} = (\underline{b}_1, \ldots, \underline{b}_L)$	Vector that denotes the number of machines either operational or in stock at the bases

Other symbols

$P(n_1, m_{11})$	$P(\underline{n}_1 = n_1, \underline{m}_{11} = m_{11})$
$\tilde{P}(k, m_{11})$	$\tilde{P}(\underline{k} = k, \underline{m}_{11} = m_{11})$
$P(\mathbf{k}_0, n_1, \mathbf{t}, \mathbf{m}_1)$	$P(\underline{\mathbf{k}}_0 = \mathbf{k}_0, \underline{n}_1 = n_1, \underline{\mathbf{t}} = \mathbf{t}, \underline{\mathbf{m}}_1 = \mathbf{m}_1)$
$\bar{P}(\mathbf{k}_0, k, \mathbf{t}, \mathbf{m}_1)$	$\bar{P}(\underline{\mathbf{k}}_0 = \mathbf{k}_0, \underline{k} = k, \underline{\mathbf{t}} = \mathbf{t}, \underline{\mathbf{m}}_1 = \mathbf{m}_1)$
$q(m_{11})$	$q(m_{11}) = P(\underline{n}_1 = S_0 \mid \underline{n}_1 \leq S_0, \underline{m}_{11} = m_{11})$
q	$P(\underline{n}_1 = S_0 \mid \underline{n}_1 \leq S_0)$
G and G'	Normalization constants
$TH_l(i)$	Throughput of the short circuited network of base l, with $l = 1, \ldots, L$, when i machines are in the network
δ	$\sum_l TH_l(J_l + S_l)/\mu_0$
A, A_l	Availability of single base model and base l with $l = 1, \ldots, L$ respectively
$E\underline{j}_l$	Expected number of machines operational at base l, with $l = 1, \ldots, L$
A_{exact}	Exact availability of single base model
$A_l sim$	Availability of base l with $l = 1, \ldots, L$ as obtained by simulation
A_{appr}, $A_l appr$	Approximated availability of single base model and base l with $l = 1, \ldots, L$ respectively
$E\underline{j}_1 exact$	Expected number of machines operational in single base model when computed exactly
$E\underline{j}_l sim$	Expected number of machines operational at base l with $l = 1, \ldots, L$ as obtained by simulation
$E\underline{j}_l appr$	Approximated expected number of machines operational at base l, with $l = 1, \ldots, L$
ρ_0	$\sum_{l=1}^{L} J_l \lambda_l (1 - p_l)/R_0 \mu_0$
ρ_l	$J_l \lambda_l p_l / R_l \mu_l$

Algorithm 2.2.2

All symbols that are used in Algorithm 2.2.2 are either described above, or described in the algorithm itself. This will not be repeated here.

Symbols used in Chapter 3

Parameters

J	Maximum number of machines operating at the production cell
λ	Failure rate of an operating machine
μ_0	Disassembly rate at disassembly facility
μ_1	Repair rate at repair facility
μ_2	Assembly rate at assembly facility
S_0	Number of spare machines
S_1	Number of spare components of type 1
S_2	Number of spare components of type 2
r_1	Two-type model: probability that a machine failure is due to a failure of a type 1 component
r_2	Two-type model: probability that a machine failure is due to a failure of a type 2 component

Random variables

$\underline{n}_1$	Single type model: number of components in repair
$\underline{n}_{11}$, $\underline{n}_{12}$	Two-type model: number of type 1, respectively type 2, components in repair
$\underline{\mathbf{n}}_1 = (\underline{n}_{11}, \underline{n}_{12})$	Two-type model: vector that denotes the number of components in repair
$\underline{n}_2$	Single type model: number of spare components
$\underline{n}_{21}$, $\underline{n}_{22}$	Two-type model: number of spare components of type 1, respectively type 2
$\underline{\mathbf{n}}_2 = (\underline{n}_{21}, \underline{n}_{22})$	Two-type model: vector that denotes the number of spare components
$\underline{k}$	Single type model: number of component backorders
$\underline{k}_1$, $\underline{k}_2$	Two-type model: number of backorders of components of type 1, respectively type 2
$\underline{\mathbf{k}} = (\underline{k}_1, \underline{k}_2)$	Two-type model: vector that denotes the number of component backorders
$\underline{m}_1$	Number of machines in the assembly facility
$\underline{m}_2$	Number of spare machines
$\underline{j}$	Number of machines operational at the production cell
$\underline{b}$	$\underline{j} + \underline{m}_2$

Other symbols

$P(n_1, m_1)$	$P(\underline{n}_1 = n_1, \underline{m}_1 = m_1)$
$\tilde{P}(k, m_1)$	$\tilde{P}(\underline{k} = k, \underline{m}_1 = m_1)$
$P(\mathbf{n}_1, m_1)$	$P(\underline{\mathbf{n}}_1 = \mathbf{n}_1, \underline{m}_1 = m_1)$

Other symbols (continued)

$\bar{P}(\mathbf{k}, m_1)$	$\bar{P}(\underline{\mathbf{k}} = \mathbf{k}, \underline{m}_1 = m_1)$
$q(m_1)$	$P(\underline{n}_1 = S_1 \mid \underline{n}_1 \leq S_1, \underline{m}_1 = m_1)$
q	$P(\underline{n}_1 = S_1 \mid \underline{n}_1 \leq S_1)$
G, G', $\tilde{G}$, $\bar{G}$, $\hat{G}$	Normalization constants
$TH(i)$	Throughput of the short circuited network when i machines are in the network
δ	$TH(J + S_0)/\mu_1$
$\mathcal{D}_0(\mathbf{k})$	$\{l \mid k_l = 0\}$
$\mathcal{T}(\mathbf{k})$	$\{\mathbf{n}_1 \mid n_{1d} \leq S_d \;\; for \;\; d \in \mathcal{D}_0(\mathbf{k}),\; n_{1d} = S_d + k_d \;\; for \;\; d \notin \mathcal{D}_0(\mathbf{k})\}$
$EW_l^a(\mathbf{k})$	The expected sojourn time for an arriving token requiring a class l component ($l \in \{1, 2\}$), given there is a backlog of $\mathbf{k}$
$q_1(0, k_2)$	Probability that upon arrival there are no spare components of type 1 available, given that there is no backlog of type 1 components and there is a backlog of k_2 type 2 components
$q_2(k_1, 0)$	Probability that upon arrival there are no spare components of type 2 available, given that there is no backlog of type 2 components and there is a backlog of k_1 type 1 components
H, H', $H_{i=1,\ldots,8}$	Certain hyper geometric probability functions
$P_1(n_{11}, n_{12})$	Certain steady distribution of $\underline{n}_{11}$ and $\underline{n}_{12}$
$\tilde{P}_1$, $\bar{P}_1$, $\hat{P}_1$	Defined similarly
$\tilde{P}_2$, $\bar{P}_2$, $\hat{P}_2$	Defined similarly
$\mathcal{A}$, $\mathcal{B}$, $\mathcal{B}_{n_{11}}$, $\mathcal{C}$, $\mathcal{C}_{n_{12}}$	Certain subsets of the state space defined by (n_{11}, n_{12})
A	Availability
$E\underline{j}$	Expected number of machines operational
A_{exact}	Exact availability of single type model
A_{sim}	Availability of two-type model as obtained by simulation
A_{appr}	Approximated availability
$E\underline{j}_{exact}$	Expected number of machines operational in single type model when computed exactly
$E\underline{j}_{sim}$	Expected number of machines operational as obtained by simulation
$E\underline{j}_{appr}$	Approximated expected number of machines operational

Algorithm 3.2.2

All symbols that are used in Algorithm 3.2.2 are either described above, or described in the algorithm itself. This will not be repeated here.

Symbols used in Chapter 4

Parameters

L	Number of bases
J_l	Maximum number of machines operating at the production cell of base l, with $l = 1, \ldots, L$
λ_l	Failure rate of a machine operating at base l, with $l = 1, \ldots, L$
p_l	Probability that a broken machine from base l can be repaired locally ($l = 1, \ldots, L$)
μ_0	Repair rate at central repair facility
μ_l	Repair rate at repair facility of base l, with $l = 1, \ldots, L$
R_0	Number of repairmen at central repair facility
R_l	Number of repairmen at repair facility of base l, with $l = 1, \ldots, L$
γ_l	Transport rate from the central repair facility to base l, with $l = 1, \ldots, L$
C	Allowed total investment
c_0	Stocking costs for one spare machine at the depot
c_l	Stocking costs for one spare machine at base l, with $l = 1, \ldots, L$

Other symbols

A_l	Availability at base l, with $l = 1, \ldots, L$
A_{tot}	Total availability
S_0	Number of spares to dedicate to the depot
S_l	Number of spares to dedicate to base l, with $l = 1, \ldots, L$
$\mathcal{S}$	$\{S = (S_0, S_1, S_2, \ldots, S_L) \mid \sum_{l=0}^{L} c_l S_l \leq C\}$
$\mathcal{S}'$	$\{S = (S_0, S_1, S_2, \ldots, S_L) \mid \sum_{l=0}^{L} c_l S_l \leq C,\ \sum_{l=0}^{L} c_l S_l + c_i > C\ for\ i = 0, 1, \ldots, L\}$
S_{opt}	Optimal allocation of spares
$\hat{S}_{opt}$	Approximate optimal allocation of spares
A_{tot} exact, A_{tot} appr	Total availability as obtained by the exact optimal stock allocation and the approximated optimal stock allocation
S_0 exact, S_0 appr	Number of spares that is allocated to the depot in the exact solution, respectively the approximate solution
S_l exact, S_l appr	Number of spares that is allocated to base l with $l = 1, \ldots, L$ in the exact solution, respectively the approximate solution

Algorithm 4.3.1 and Example 4.3.2

All symbols that are used in Algorithm 4.3.1 and in Example 4.3.2 are either described above, or described in the algorithm or example.