Simulation and Analytical Models for Autonomous Vehicle Storage and Retrieval System

By

Banu Yetkin Ekren

Ph.D. Candidacy Proposal

Department of Industrial Engineering

University of Louisville, Louisville, KY, 40292 USA

Examining Committee:

____

Sunderesh S. Heragu, Thesis Advisor

____

Gerald Evans, Member

____

Gail W. DePuy, Member

____

John Usher, Member

____

Thomas Riedel, Member

April, 17th 2009

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

In this proposal, we present simulation and analytical models for automated, unit-load (UL) storage and retrieval systems based on autonomous vehicle (AV) technology. Autonomous vehicle storage and retrieval systems (AVS/RSs) represent a relatively new technology for automated, UL storage systems (Malmborg, 2002). AVs function as storage/retrieval (S/R) devices. A key distinction of AVS/RSs relative to traditional crane-based automated storage and retrieval systems (AS/RS) is the movement pattern of the S/R device. In AS/RSs, aisle-captive storage cranes capable of simultaneous movement in the horizontal and vertical dimensions store or retrieve unit loads. While the travel patterns in an AS/RS are generally more efficient within storage racks, an AVS/RS has a significant potential advantage in the adaptability of system throughput capacity to transactions demand by changing the number of vehicles operating in a fixed storage configuration. For example, decreasing the number of vehicles increases the transaction cycle times and utilization which are also key measures of system performance. The vehicles in an AVS/RS share a fixed number of lifts for vertical movement and follow rectilinear flow patterns for horizontal travel.

In this proposal, to be able to benefit from both modeling approaches, we develop simulation and analytical models for a particular AVS/RS. The former simulates the sequence of events that could occur over a period time via a computer program. There are some advantages and disadvantages with the simulation methodology. The most important advantage is the ability to model complex systems in great detail, so it provides more accurate results. For example, a verified and validated simulation model could provide results – performance measure estimates - that are very close to those seen in the actual system. However, this high accuracy comes at the

expense of high modeling and computational effort. Developing a detailed, more accurate simulation model for a large system is time consuming.

Analytical modeling is the second approach used for evaluating the performance of the system. This approach uses mathematical relationships between inputs and outputs. The most important advantage of analytical method is that it is not time consuming. It has an ability to evaluate the system's performance in a reasonable time. However, to develop an analytical model for a complicated system is not a simple task. Also, changing an assumption in an analytical model may render the model invalid. These are some of the disadvantages of analytical modeling approach. When properly designed, analytical models, however, are capable of providing reasonably accurate estimates of complex systems in a relatively short time.

The simulation model of the AVS/RS is completed in ARENA 12.0, a commercial simulation software. We use the simulation model of AVS/RS to validate the analytical model. After simulating the system we complete a simulation based experimental design to identify factors affecting the performance of AVS/RS. The factors considered in the design of experiment (DOE) are: dwell point policy, the vehicle-lift combination, scheduling rule, input/output (I/O) locations and interleaving rule. Three different responses, storage and retrieval transactions' average cycle time, average utilizations of vehicles and lifts, are considered. After determining the main and the interaction effects, a Tukey test analysis is completed on the responses to determine the best levels of the factors which provide the minimum average cycle time and utilizations that are statistically significant.

The analytical model of the system is modeled as a semi-open queuing network (SOQN) model. An SOQN consists of jobs, pallets and servers. Each job is paired with a pallet and the two visit the set of servers required for processing the job in the specified sequence. In an

AVS/RS, storage/retrieval (S/R) transactions are the ‘jobs’ and the AVs are the ‘pallets’. If an S/R transaction requires a vertical movement it uses a lift. The lifts and horizontal travel times to and from a storage space are modeled as servers. First, we describe all possible scenarios and their probabilities to derive the general service times. Second, we combine all the service times of the network. Third, we extend Marie’s (1980) approximation to a load-dependent general network. Fourth, we solve the SOQN using this extended approximate method and aggregation technique (Avi-Itzhak, 1973) to obtain the performance measures. Last, we compare the approximate and simulation results.

In this proposal, the case study is completed for a company that uses AVS/RS in France.

CHAPTER 1
INTRODUCTION

1.1 Autonomous Vehicle Storage and Retrieval System

AVS/RS represent a relatively new technology for automated UL storage systems (Malmborg, 2002). This technology has recently been implemented at over fifty facilities in Europe and is being introduced in other world markets. In an AVS/RS, AVs function as S/R devices that follow three-dimensional rectilinear movement patterns. Within the storage rack, the key distinction of AVS/R systems relative to traditional crane-based AS/RS is the movement pattern of the S/R device. In AS/RSs, aisle-captive storage cranes capable of simultaneous movement in the horizontal and vertical dimensions store or retrieve ULs. While the travel patterns in an AS/RS are generally more efficient within storage racks, an AVS/RS has a significant potential advantage in the adaptability of system throughput capacity to transactions demand by changing the number of vehicles operating in a fixed storage configuration. For example, decreasing the number of vehicles increases the transaction cycle times and utilization which are also key measures of system performance. The vehicles in an AVS/RS share a fixed number of lifts for vertical movement and follow rectilinear flow patterns for horizontal travel.

It is crucial to design an AVS/RS in such a way that it can efficiently handle the current and future demand requirements while avoiding bottlenecks and excess capacity. Due to the relative inflexibility of the physical layout and the equipment, it is important to design it right the first time. Figure 1.1 illustrates the key components of an AVS/RS. Figure 1.1a is a three-dimensional view of an AVS/RS, whereas Figure 1.1b is a plan view. The major system components of this technology are lift mechanisms that are mounted along the periphery of the storage rack to provide vertical movement. The dominant cost component of the AVS/RS technology is the vehicle. Lifts costs are approximately 25% of those for the vehicles. The storage rack costs per load position for AVS/RSs are comparable to those in an AS/RS. Lifts are typically not the bottleneck in an AVS/RS but they do have a significant influence on throughput capacity because vertical movement is generally slower than horizontal movement. They can also contribute substantially to transaction cycle times.

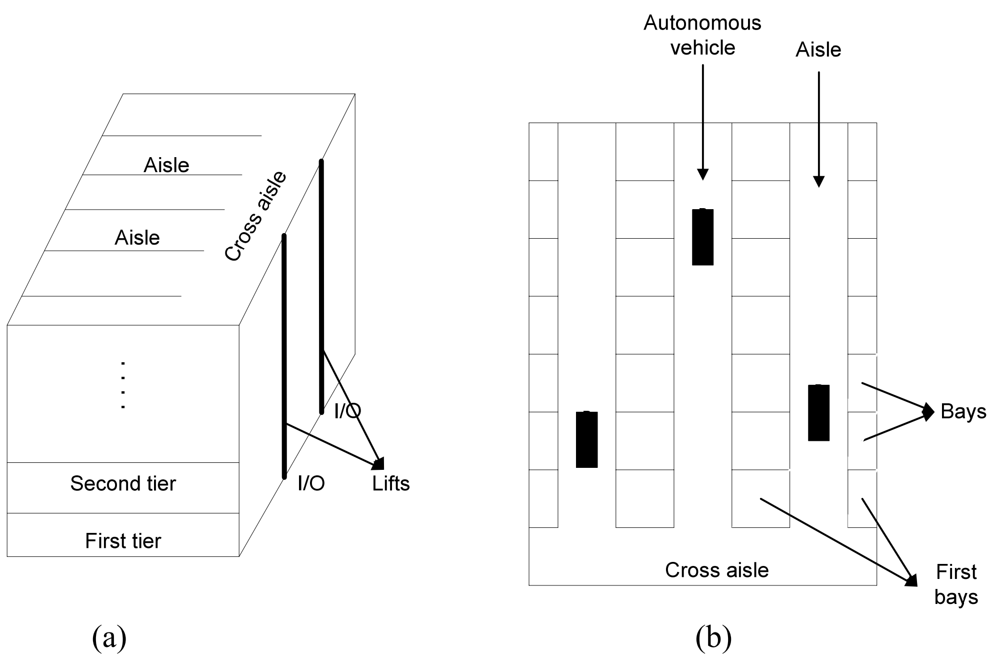

Figure 1.1: The key components of an AVS/RS

1.2 Simulation and Analytical Models

In general, there are two kinds of approaches to evaluate a system's performance. The first one is by simulating the system. Simulation models simulate the sequence events that could occur as the system operates via a computer program. The probabilistic nature of many events, such as machine failure and processing times, can be represented by sampling from a distribution representing the pattern of the occurrence of the event. Thus, to represent the typical behavior of the system, it is necessary to run the simulation model for a sufficiently long time, so that all events can occur a sufficiently large number of times.

There are some advantages and disadvantages in simulation methodology. The most important advantage is the ability to model complex systems in great detail, so it provides more accurate results. For example, a verified and validated simulation model could provide results - performance measure estimates – that are very close to those seen in the actual system. However, this high accuracy comes at the expense of high modeling and computational effort. Developing a detailed, more accurate simulation model for a large system is time consuming. A large amount of data must be collected and used. To obtain the performance measure estimates with high confidence levels, the simulation model must be run many times. Thus, simulation models are inefficient for quick what-if analyses. They are useful, however, in validating analytical models.

Analytical modeling is the second approach for evaluating the performance of systems. This approach uses mathematical relationships between inputs and outputs. These mathematical relationships are then used to derive a formula or to define an algorithm by which the performance measures of the system can be evaluated. It is not possible to perform an exact analysis of many systems including the AVS/RS, using an analytical method. So, approximate methods are used. Many of these methods are capable of providing reasonably accurate estimates of complex systems in a relatively short time.

We summarize the differences between simulation and analytical models in Table 1.1.

Table 1.1 Comparison between simulation and analytical models

	Simulation Model	**Analytical Model**
Model Complexity	Unlimited	Limited
Run Time	Long	Short
Data Requirements	Large	Small
Model Development	Predictable	Unpredictable
Flexibility	High	Low

We simply explain the above properties below:

Model complexity: Simulation models can be modeled at any desired level of complexity. On the other hand, analytical models are limited in the complexity of the system that they can incorporate.

Run time: The computational complexity for exact analytical models typically increases exponentially with an increase in the size of the system. Approximate analytical models may be able to handle larger systems, but because of the difficulty in developing and testing approximations, the range of manufacturing systems for which proven approximate models exist is small. On the other hand the computation time required to produce results from simulation models at a sufficiently high level of accuracy is very long.

Data requirements: Most analytical models have minimal data requirements, because they tend to be fairly simple descriptions of the system. Simulation models, because of their ability to model the real system, may require large amounts of data.

Model development: One of the advantages of simulation modeling is that the time required to develop a model can usually be assessed reasonably accurately. Analytical models are much more unpredictable in the time and effort required to develop an adequate model.

Flexibility: Simulation models are usually easy to adapt and so are suitable for what-if analyses. For example, little effort is needed to add something to or change something within the model. On the other hand, changing an assumption in an analytical model may render the model invalid and may require a new one to be developed.

1.3 Queuing Models

Queuing models are very useful in many practical applications such as, manufacturing, inventory and communication systems. They are extensively used for estimating performance measures of these systems. A simple queuing model has a single station with an infinite buffer in front of it. The jobs are all identical and arrive according to some type of arrival process. The work station processes the jobs one by one. This system can be exactly modeled by a single-server queuing model (see Figure 1.2).

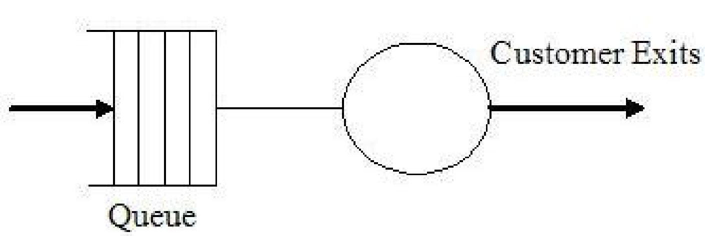

Figure 1.2: Basic queuing model

A queuing model can be characterized by:

- *The arrival process of customers*: Usually, we assume that the interarrival times are independent and have a common distribution. In many practical situations, customers arrive according to a Poisson stream (i.e., exponential interarrival times). Customers may arrive one by one, or in batches.
- *The behavior of customers*: Customers may be patient and willing to wait for a long time. Or customers may be impatient and leave after a while. For example, in a call center, customers may wait for an operator to become available, or they may hang up and never return.
- *The service times*: Service time is the time delay after the queue waiting time. We usually assume that the service times are independent and identically distributed, and that they are independent of the interarrival times. For example, the service times can be deterministic, or follow exponential or general distributions. Service times could also depend on the queue length. For example, the processing rates of the machines in a production system may depend on the number of jobs waiting to be processed.
- *The service discipline*: Customers can be served one by one or in batches. There are many possibilities for the order in which they enter service. For example, first-come, first-served (FCFS), i.e., in order of arrival; random order; last-come, first-served (e.g., in a computer stack); priorities (e.g., closest due-date first, shortest processing time first); processor sharing (in computers that equally divide their processing power over all jobs in the system).
- *The service capacity*: There may be a single server or a group of servers serving the customers.

- *The waiting room*: There can be limits with respect to the number of customers that can be accommodated in the system. For example, the available space in the machining department can be limited; or in a data communication network, only a finite number of calls can be buffered at a switch.

A single-server queuing model, where the arrival process is Poisson and the service times are exponentially distributed, is shown as *M*/*M*/1 queue. Kendall (1953) introduced this shorthand notation for queuing models. It is a three part code a/b/c, where the first letter, a, specifies the inter-arrival time distribution and the second one, b, the service time distribution. For example, *G* is used for a general distribution and *M* is used for an exponential distribution (*M* stands for the memoryless property at the exponential distribution). The third letter stands for the number of servers. If the inter-arrival and service times are generally distributed in a single-server queue, it is represented as a *GI*/*G*/1 queue, where *GI* stands for general and mutually independent arrivals. The notation can be extended with additional letters to provide additional information about the queuing model. For example, a system with exponential interarrival and service times, one server and having waiting room for only *N* customers is represented as *M*/*M*/1/*N*.

1.4 Queuing Network Models

A queuing network is a collection of service stations organized in the sequence customers go from one station to another in order to satisfy their service requirements. Queuing networks are usually classified in two types, open and closed queuing networks. However, sometimes it is referred *mixed queuing network* when there are a number of different classes of jobs in the system. In this case, the network may be open for some of the classes while it may be closed for some other classes of customers.

An *open queuing network* (OQN) is characterized by one or more sources of job arrivals from outside to one or more queues, receiving service at a finite number of servers in a specified sequence and departing the network from the last queue. The jobs move from one station to another in the network in an order given by the routing probabilities. Assembly lines and sequence of traffic intersections are examples of systems that can be modeled as OQNs.

In a *closed queuing network* (CQN), jobs neither enter nor depart from the network (see Figure 1.3). In other words, the number of jobs in the network is fixed and the sum of the jobs at all the individual queues always equals the total number of jobs in the system. A CQN seems unusual at first glance because it does not allow jobs to enter or leave the network. A typical system of this type would be one where there is a large (infinite) number of jobs waiting to enter the system at all times. When the job completes its service, it leaves the system and a new job enters the system immediately. Thus, the number of jobs inside the system always remains fixed. For example, there may be a fixed number of customers in the service area of a bank or in a time-shared computer system model, a fixed number of jobs individually may require a number of computing and completed IO services. Whenever the tasks required for a job are completed, another job enters the system for its required services. It should be noted that these examples ensure the CQN condition under the assumption that customers and jobs are always available whenever one inside leaves the system.

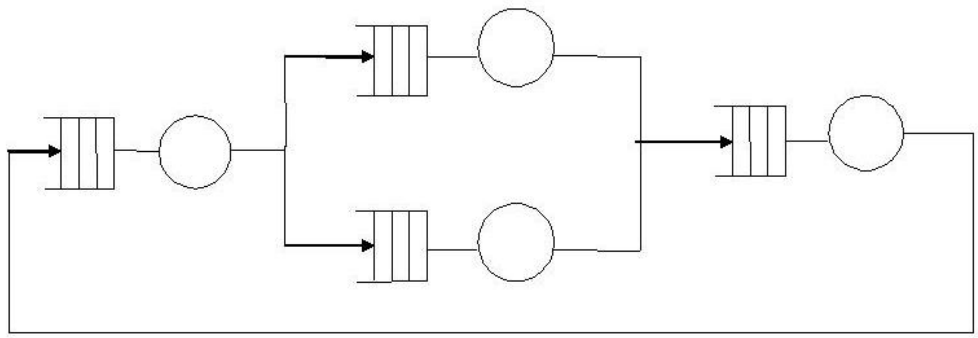

Figure 1.3: A CQN model

Mixed queuing networks are usually referred when there are multiple classes of customers/jobs in the system, operating as closed network for some of the jobs, and open for the others.

Due to the extensive applicability, queuing networks have attracted a large number of researchers. In this thesis, we will focus on a special type of queuing network called the SOQN. We will model an AVS/RS using a SOQN approach.

1.5 Semi-open Queuing Network

An SOQN consists of jobs, pallets and servers. Each job is paired with a pallet and the two visit the set of servers required for processing the job in the specified sequence. If this resource (pallet) is available, the customer/job enters the network immediately. Otherwise, it waits in an external queue until a resource becomes available (see Figure 1.4). When the customer exits the system, the resource associated with the customer returns to a 'pallet' pool and waits to be paired with the next arriving customer. Buitenhek et al. (2000) and Dallery

(1990) call this system as an OQN with population constraint. However, we will use the SOQN terminology as Buzacott and Shanthikumar (1993).

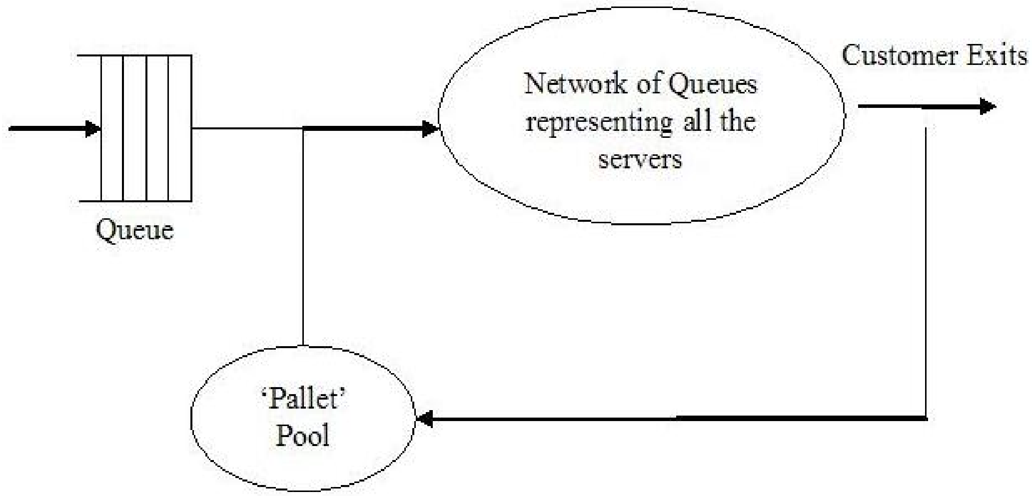

Figure 1.4: SOQN

If we have an unlimited number of pallets, then customers will never wait in the external queue. Thus, the system becomes an OQN. In contrast, if the number of pallets is small compared to the numbers of customers, then the customer typically waits in the external queue before it enters the system. So, the system becomes a CQN. SOQN is in between these two cases. We have a sufficient number of pallets in the system so that the system never explodes.

There are many systems that only can be modeled using SOQN. For example, because of the limited number of tellers in a bank, limited number of customers may be allowed to enter the bank. Or at a hospital, because of safety conditions a limited number of patients may be allowed to enter inside, the others may have to wait in a reception area. Multiprogramming computer system (Avi-Itzhak and Heyman, 1973) and communication networks with window flow control (Fdida et al., 1990) can also be modeled as SOQN.

The term product-form, introduced by Jackson (1963) and Gordon and Newell (1967), states that all stations have equilibrium state probabilities which can be expressed as product of factors describing the state of each individual station within the network. So, the individual stations behave as if they are separate queuing systems.

The early study of Jackson (1963) showed that for an OQN with Poisson arrivals, FCFS service disciplines, exponential service times, and probabilistic routing, the steady-state joint probability has a product-form solution. Because of the external queue, an SOQN does not have an exact product-form solution. Therefore, approximate techniques have been used. In the following chapter, we give a literature review of AVS/RS and SOQN models. In chapter three, we propose an approximation procedure for solving load-dependent CQNs having general service times with low variability. The proposed procedure is the extension of Marie's CQN approximation. In chapter four, we model the AVS/RS using an approximate analytical and the extended approximate procedure in chapter three, SOQN model and compare the analytical results with the simulation results. In chapter five, we develop a simulation model of a real application of the AVS/RS, and design a set of experiments to identify the significant factors that could affect the performance of the AVS/RS. After DOE, we implement Tukey test to determine the best levels of the factors which are statistically significant. In chapter six, we give brief information about an exact analysis procedure, Matrix Geometric Method (MGM) to solve the SOQN. In chapter seven, we discuss the studies we propose to undertake after completing the candidacy exam.

CHAPTER 2
LITERATURE REVIEW

In this chapter, we give a literature review of the AVS/RS and SOQN solution methods.

2.1 AVS/RS

AVS/RS is first studied by Malmborg (2002). In that study, he proposes analytical conceptualizing tools to model the expected performance of AVS/RS as a function of key system attributes - storage capacity, rack configuration and fleet size. The new conceptualizing tool exploits the analogies between AVSRS and automated guided vehicle systems (AGVS), as well as similarities with traditional AS/RS.

In most UL S/R systems, space-conserving random storage policies are used because of capital cost considerations (Heragu, 2008). Under a random storage policy, the location of a specific load in a storage system is a random variable over time. This policy ensures space occupancy rate maximization by allowing different loads to occupy the same addresses at different times (Malmborg, 1996).

AVS/RS technology can provide cost effective automation for UL S/R systems with varying transaction volumes because it enables the designer to vary the number of S/R devices (i.e., AVs), depending on the number of S/R transactions in a system. With traditional crane-based technology, aisle-captive S/R devices utilize highly efficient Chebychev movement patterns within aisles. However, the high cost of crane-based systems (one crane is required per aisle) raises the threshold for cost effective automation and crane-based AS/RSs can be justified only for high throughput operations in a stable demand environment.

In his later study, Malmborg (2003) proposes a state equation model for predicting the proportion of dual command cycles in AVS/RSs that use interleaving. The model extends AVS/RS concepting tools that require an estimate of this proportion to predict system utilization and throughout capacity. This proportion is used to estimate storage and retrieval cycle times, system utilization and throughput capacity for alternative system design profiles defined by the number of storage aisles, storage rack height and depth, the vehicle fleet size and the number of lifts used for vertical movement.

Kuo et al. (2007) also propose design conceptualization models for AVS/RSs. They formulate vehicle and lift travel times and the probability distribution for 12 service scenarios occurring under realistic operating assumptions to generate expected transaction service times. They formulate the transaction waiting time and vehicle utilization using random storage and point-of-service-completion dwell point rules. The models provide a practical method for predicting key aspects of system performance based on five design variables that drive the majority of system costs.

A cycle-time model for class-based storage policies proposed for AVS/RS is given in Kuo et al. (2008). The model is based on a queuing network approach to achieve sufficient accuracy and computational efficiency for system design conceptualization applications. The model is applied on a real problem.

Another computationally efficient cycle time model for conceptualizing AVS/RSs is presented in Fukunari and Malmborg (2008). They compare the performance of an AVS/RS with crane-based AS/RSs. This model is based on an iterative computational scheme exploiting random storage assumptions and queuing model approximations. The proposed procedure has an ability to scale up efficiently for large problems so, enabling more extensive search of a design solution space.

Fukunari and Malmborg (2009) also show another queuing network approach to estimate performance measures for AVS/RSs using opportunistic interleaving. The model exploits the distribution of cycle types and random storage assumptions to estimate the proportion of single and dual command cycles in a system. The proposed technique provides additional advantage of flexibility for modeling the interfaces between a storage system and the overall material flow system in a facility.

Recently, Zhang et al. (2009) have proposed approximation procedures for estimating transaction waiting times in AVS/RSs. They consider, non-Poisson arrivals and non-exponential service times in the AVS/RS. The proposed model uses a series of queuing approximations for estimating the transaction waiting times. The technique selects among three alternative queuing approximations based on squared coefficient of variation of transaction inter-arrival times.

2.2 Closed Queuing Networks

Although our aim in this thesis is to model the AVS/RS as an SOQN, it is necessary to briefly review CQN, because our SOQN approach relies on CQN solution tools.

The early study of Jackson (1963) showed that for an OQN with Poisson arrivals, FCFS service disciplines, exponential service times, and probabilistic routing, the steady-state joint probability has a product form solution. In addition, each station can be analyzed in isolation as an $M/M/1$. Later, Gordon and Newell (1967) proved that the product form solution also holds for

CQNs of Jackson type. Buzen (1973) presented methods for computing the equilibrium distribution of customers in CQNs with exponential servers. The results are extended for open, closed, and mixed networks with several job classes, non-exponentially distributed service times and different queuing disciplines (Baskett et al., 1975).

Product-form networks can be analyzed by efficient algorithms such as convolution algorithm (Buzen, 1973), the mean value analysis (MVA) algorithm, developed by Reiser and Lavenberg (1980), or Chandy et al. (1975) algorithm. MVA has gained popularity as an exact technique for providing solutions to product form CQNs. The basic concept of MVA is the application of an iterative procedure and repeated use of the famous Little's law (Little, 1961) to calculate mean residence time, system throughput and the mean number of jobs. The advantage of this method is that the performance measures can be computed without explicitly computing the normalization constant. In this thesis, because we use load-dependent MVA, we briefly describe the algorithm and the notation used in that algorithm:

L : the number of stations in the network

n : the current population in the network

M : the maximum population in the network

$\mu_i(n)$: the load-dependent service rates of station i

e_i : the visit ratio to station i, found through the routing matrix

$T_i(n)$: the average waiting time at station i when the network population is n

$W_i(n)$: the average number of jobs at station i when the network population is n

$\lambda(n)$: the throughput rate of the system when the network population is n

$\lambda_i(n)$: the throughput rate at station i when the network population is n

$p_i(l/n)$: the conditional probability of l jobs at station i given the network population is n

MVA

1. Initialization

   For i = 1, …., L, set $p_i(0|0) = 1$

2. Iteration

   (a) Compute mean waiting time

   For i = 1, ….., L, $T_i(n) = \sum_{l=0}^{n-1} \frac{l+1}{\mu_i(l+1)} p_i(l|\mathrm{n}-1);$

   (b) Compute throughput

   $\lambda(n) = n/\sum_{i=1}^{L} e_i T_i(n);$

   $\lambda_i(n) = e_i \times \lambda(n);$

   (c) Compute conditional probabilities

   For i = 1,…., L,

   For l = 1,...., n, $p_i(l|n) = \lambda_i(n) \times p_i(l-1|n-1)/\mu_i(l);$

   $$p_i(0|n) = 1 - \sum_{l=1}^{n} p_i(l|n)$$

The visit ratios stations, e_i , i = 1, …, L can be calculated by (2.1):

$$e_i = \sum_{j=1}^{L} e_j p_{ji} \tag{2.1}$$

The MVA method is basically based on two fundamental laws:

1. *Little's theorem*, which is introduced by (2.2) to express a relation between the mean number of jobs, the throughput, and the mean residence time at a station or in the overall system:

$$W_i = \lambda \times T_i \tag{2.2}$$

2. *Theorem of distribution at arrival time*, proven by Lavenberg and Reiser (1980) and Sevcik and Mitrani (1981), for all networks that have a product-form solution. The arrival theorem says that in a closed product-form queuing network, the probability mass function (pmf) of the number of jobs seen at the time of arrival to a station *i* when there are *n* jobs in the network is equal to the pmf of the number of jobs at this station with one less job in the network. At the instant a job arrives at a station, this job itself is not already in the queue of this station. Thus, there are only *n*-1 other jobs that could possibly interfere with the new arrival.

The MVA has some limits. For example it cannot be applied to non-product-form networks, e.g., CQNs with general service time stations. Approximate MVA (AMVA) methods are developed to attack this problem. For a large network and/or population size, Bard (1979) and Schweitzer (1979) constructed a fixed-point problem to directly calculate the performance measures for the actual population without touching the reduced population situations. Another AMVA is developed by Buzacott and Shanthikumar (1993). They propose an algorithm incorporating the first two moments of the service time at each station to calculate the probability that a job arriving at a station finds all servers are busy. This probability is then used in an MVA algorithm to calculate mean waiting times.

2.3 Semi-open Queuing Networks

Networks containing FCFS, PS (processor sharing), IS (infinite server), and LCFS-preemptive resume service disciplines have product form solution. In addition, three new types, SIRO (service-in–random order) (Spirn, 1979), LBPS (last batch processor sharing) where jobs arriving as batches are processed as LCFS (Noetzel, 1979) and WEIRDP (a *p* portion of the servers is allocated to the first job and the remaining portion is allocated to the remaining jobs in the queue) (Chandy and Martin, 1983), are also shown to have product-form solutions. In the FCFS service discipline service cases, the service times are assumed to have an exponential distribution. The other service disciplines allow general service time distributions.

Product-form networks can be analyzed by efficient algorithms such as convolution algorithm (Buzen, 1973), the MVA algorithm (Reiser and Lavenberg, 1980) or Chandy et al. (1975) algorithm. However, these solutions are known to be too restrictive for many applications. If a queuing network does not have product-form, it is usually necessary to solve it as a Markov chain (Stewart, 1978). This approach requires the number of states to be small.

A large number of approximation methods are available in the literature. Most of them consider non-exponential distributions. Examples are: diffusion approximation (Gelenbe, 1975; Kobayashi, 1974), EPF-technique of Shum and Buzen (1977), maximum entropy method (Kouvatsos, 1993), decomposition approach (Kuhn, 1979; Whitt, 1983; Gelenbe and Pujolle, 1987), method of Marie (1979, 1980) and MVA approximations (Reiser, 1979; Akyildiz, 1987). The extended MVA is also used for approximating priority networks (Bryant et al., 1984). There are also other approximations which can be found in Bolch et al. (1988).

SOQN does not have an exact product-form solution. Therefore, approximate methods are developed for the solution procedure. In general, the SOQN methods can be classified into three groups: aggregation based methods, CQN based methods, and MGM based methods.

2.3.1 Aggregation based methods

The earliest study on SOQN was by Avi-Itzhak and Heyman (1973). They modeled multiprogramming computer systems as SOQN where the number of jobs in a computer was limited by its capacity and there was an external queue to buffer the extra jobs. The first step in their method is to calculate the throughput of the network in isolation i.e., considered as a CQN, for all populations 1 to N where N is the capacity of the network. Then, the network is replaced by a single-server with load-dependent service rates obtained in the first step. Thus, the aggregate system reduces to an $M/M/1$ queue with load-dependent service rates. This aggregation technique may also be used in the case of general service time distribution. In that case, the closed model may be analyzed using, for instance, Marie's method (1979). Buzacott and Shantikumar (1980) applied a similar technique on modeling flexible manufacturing systems. They considered an aggregated Markov process (a birth-death process) whose states are given by the number of customers inside the network as well as the external queue, and rate out of a state to the next (adjacent) higher state is equal to the arrival rate whereas the rate out to an adjacent lower state is estimated by the throughput of a CQN with appropriate population constraint. They use AMVA algorithms to estimate the CQN throughput.

Heragu and Srinivasan (2008) provide solutions for various single-class, SOQN models assuming multiple servers and general interarrival and service time distributions. They demonstrate the effectiveness of their approach using some experimental results. The analysis of multiple servers is done by aggregating all the servers except the bottleneck as the second server and the bottleneck server as the first. The aggregate server is a load-dependent exponential server with a service rate equal to the load-dependent throughput rate of the closed network formed by the aggregated servers. State equations were truncated and solved for the aggregate two server SOQN. They extended the results to multiple general, tandem servers using well-known approximations and correction factors to the results obtained for a comparable exponential SOQN.

2.3.2 CQN based methods

Dallery (1990) formulated a different approach that is based on Marie's (1980) approximation. He assumes coxian type service stations and Poisson arrivals in the network and derives an equivalent CQN and analyzes this closed model using an approximate product-form solution. In the Dallery (1990) method, the equivalent view of the system is provided by exchanging the roles of jobs and the tokens (pallets). Tokens are considered as the customers of the network, and the jobs are considered as resources. There is a fixed number of customers in the network which is equal to the number of tokens. Dallery (1990) considers a synchronization station with external queue plus the original network. This network is solved using Marie's (1980) approximation. The key challenge in this process is to find the throughput rates for the synchronization station. By using this methodology he also aims to decrease the number of iterations. Buitenhek et al. (2000) assumed the same type of network as Dallery and adopted the CQN view as well. However, they use AMVA based solution to the equivalent CQN. They also completed a different analysis on the isolated synchronization station to provide the throughput rate of the resulting CQN to be consistent with the job arrival rate.

Both Dallery (1990) and Buitehenk (1998) have discussed the stability condition of an SOQN. In order for the external job queue not to grow infinitely, the network must have sufficient capacity to service jobs in time. It is obvious that this capacity depends on both the service rates of the stations in the network and the total number of pallets provided. When all the pallets are inside the network, the throughput rate of the system reaches its maximum which is denoted as $X(N)$. Intuitively, stability requires the job arrival rate λ to be less than $X(N)$. $X(N)$ can be approximately obtained by solving the network (without the external queue) as a CQN with a fixed population size. We assume the systems we study satisfy this stability condition.

2.3.3 Matrix geometric based methods

The MGM was developed by Marcel Neuts in the 1980s. It is a numerical approach to solve Markov processes having a special property called the matrix-geometric property. Buitenhek (1998) developed an alternative approach for tandem lines. He decomposed the network into two subnetworks, aggregated each as a load-dependent exponential server, then solved the system via the MGM. Jia and Heragu (2008) also apply a similar MGM to solve the single-class and multi-class SOQNs exactly for many examples.

Krishnamurthy et al. (2000) analyzed some aspects of SOQNs in the analysis of kanban systems. Although they do not model kanban systems as SOQNs, they have analyzed the fork-join station, which is another way of analyzing the synchronization station in SOQN models. However, kanban stays with an arriving customer for only one process and it cannot be used for SOQNs where a customer (paired with a kanban or pallet) visits multiple stations in tandem. A single stage kanban sytem is solved in Krishnamurthy and Suri (2006) and more complex networks in Granger et al., (2006).

2.4 Summary

In this chapter we presented a literature review on AVS/RS, CQN and SOQN problems. In this thesis, because our objective is to model AVS/RS problem using SOQN approach, the literature review is completed through this subject. In the following chapter we will demonstrate an extended approximate analytical model for calculating the performance of CQNs having general service times with low variability. Then, we will model the AVS/RS using this method and compare the analytical results with the simulation results.

CHAPTER 3

APPROXIMATE ANALYSIS OF LOAD-DEPENDENT GENERAL QUEUING NETWORKS WITH LOW SERVICE TIME VARIABILITY

We model AVS/RS with SOQN approach where stations have general service times. General distributions are typically described by their first two moments -the mean and the squared coefficient of variation (scv). Scv represents the ratio of variance to the mean square.

In this chapter, because of our network's structure (see Chapter 4) we present an approximate method for solution of load-dependent CQNs having general service time distributions with low variability. The proposed technique is an extension of Marie's (1980) method. In modeling of the SOQN, we use Marie's well-known approximation for general queuing network to calculate the load-dependent throughput rates. However, Marie (1979, 1980) has developed this technique for general networks having single server stations. Because our actual network has multi-servers at each station, in this chapter we extend Marie's method to load-dependent general queuing network. Then, we implement the new technique for our problem.

3.1 Approximate Analysis of Load-Dependent General Queuing Networks

Marie (1979, 1980) presents an efficient technique for the approximate analysis of general CQN. The basic idea is to approximate the performance of the general network by the performance of a product-form CQN. It provides satisfactory results for a wide range of queuing networks. The proposed technique is based on an iterative procedure and the concept of the conditional throughputs, $v_i\ (n)$, of a station. In the method, each station is analyzed under a Markovian process with load-dependent arrival rate, $\lambda_i(n)$. Marie suggests using an Erlang phase-type distribution for networks having low service time variability (Marie, 1980). Because in our network the squared scv values of service times smaller than 0.5, in this section we analyze a CQN for $\lambda(n)/E_k/r$ and FCFS station types (see Section 4.4.3).

3.1.1 Extension of Marie's approximation

Existing approximate solutions are applied to networks with load-independent stations. However, like in our system in many systems, one or more stations may have more than one parallel server. For example, in a manufacturing environment there may be multiple drilling machines at a station; or in a computer system there may be multi-processors at a station, or a station's service time may change according to the queue length (Sauer, 1983; Akyildiz, 1988). Therefore, the extended model introduced in this section can be used for any of the above type of load-dependent networks.

The assumptions of the CQN are:

- There is a single class of jobs and N stations
- A constant number of M jobs exist in the network
- Each station has generally distributed service times with load-dependent $1/\mu_i(n)$ for $\forall n$.
- The scv of service times are smaller than 0.5 at each station
- Each station has FCFS service discipline and infinite buffer capacity

- A job proceeds to station j with probability p_{ij} when it completes its service at station i, $\forall i, j$

In Marie's (1980) method, each station is considered in isolation with Poisson arrival rates. To determine the load-dependent arrival rates, $\lambda_i(n)$, he considers a product-form network corresponding to the given non-product form network. The product-form network is derived from the original network by simply substituting the FCFS stations with generally distributed load-dependent service times. Because the service times $1/\mu_i(n)$ for the stations of the network are determined in the isolated analysis of each station, the method is iterative where the network is initialized with the service rates of the original network. The load-dependent arrival rates $\lambda_i(n)$ of the stations are computed by short-circuiting station i in the network (Marie, 1980). If there are n jobs at station i, then there will be M-n jobs at the composite station. We compute the $\lambda_i(n)$ given by (3.1).

Load Dependent Arrival Rates

We assume that the sub-network has a product-form solution and has a local balance. Thus, the $\lambda_i(n)$ values can be calculated using MVA (Reiser, 1981) or convolution algorithms (Buzen, 1973). We calculate the $\lambda_i(n)$ values, for each station i = 1, 2…, L using Marie (1980):

$$\lambda_i(n-1) \times \pi_i(n-1) = \mu_i(n) \times \pi_i(n) \tag{3.1}$$

$\pi_i(n)$ is the probability of having n jobs at station i when there are M jobs in the entire network. The MVA method gives the $\pi_i(n)$ values automatically when the iteration terminates (see Section 2.2).

Load Dependent Service Rates

Marie (1980) suggests an Erlang-k type distribution for a network having service time scv less than 0.5. For each separated sub-network, the number of phases k_i is calculated by $1/c_i^2$ where, $c_i^2 = \frac{1}{k_i} + \varepsilon$. So, the new load-dependent $\hat{\mu}_i(n)$ for Erlang-k is calculated by:

$$\hat{\mu}_i(n) = k_i \times \mu_i(n) \tag{3.2}$$

In an Erlang-k phase type distribution, the job starting from the first phase enters the next phase with probability of one until it completes phases. The total time a job spends in a phase is exponentially distributed. Here, each phase has the same service rate as shown by (3.2).

Maritas and Xirokostas (1977) studied an $M/E_k/r$ model. They show a machine interference problem in which the run times follow the negative exponential distribution and the repair times follow the Erlang distribution. Their numerical procedure depends on the matrix solutions. Therefore, for large M values the method does not give accurate results. Besides, for large k and r, the technique represents difficulty in evaluating the model. However, in our proposed method we can handle the large k and r conditions by an iterative procedure.

3.1.2 Iterative algorithm for solving an $M/E_k/r$ model

We present the iterative algorithm for solving an $M/E_k/r$ model. A state of a station is denoted by

$$n = (n, j)$$

where,

n is the number of jobs, $n = 0, 1, 2, \ldots, M$

j is the number of phases, $j = 1, 2, \ldots, k_i$.

A transition from one state to another state takes place when a new job arrives or a job leaves a phase. The arrival rates are represented by $\lambda_i(n)$ and computed by (3.1). $\hat{\mu}_i(n)$ is the departure rate of a job from one phase to another and also it is the rate of leaving the station after the last phase. The state (0) denotes that there is no job at the ith station.

Figure 3.1 shows the state transition diagram of the Erlang-k phase-type. $p_i(n, j)$ is the probability of being at the ith station of the network, and there is n number of jobs, and the current phase is j. We use Chapman-Kolmogorov equations to calculate the state probabilities, $p_i(n, j)$. It should be noted from the figure that, $\lambda_i(M) = 0$. According to Figure 3.1, there are five cases to consider.

1) if $n > 1$ and $j = 1$

$$\lambda_i(n-1)p_i(n-1,1) + \hat{\mu}_i(n+1)p_i(n+1,k_i) = [\lambda_i(n) + \hat{\mu}_i(n)]p_i(n,1) \tag{3.3}$$

2) if $n > 1$ and $j = 2,3,\ldots, k_i$

$$\lambda_i(n-1)p_i(n-1,j) + \hat{\mu}_i(n)p_i(n,j-1) = [\lambda_i(n) + \hat{\mu}_i(n)]p_i(n,j) \tag{3.4}$$

3) if $n = 0$

$$\lambda_i(0)p_i(0) = \hat{\mu}_i(1)p_i(1,k_i) \tag{3.5}$$

4) if $n = 1$ and $j = 1$

$$\lambda_i(0)p_i(0) + \hat{\mu}_i(2)p_i(2,k_i) = [\lambda_i(1) + \hat{\mu}_i(1)]p_i(1,1) \tag{3.6}$$

5) if $n = 1$ and $j = 2,3,\ldots, k_i$

$$\hat{\mu}_i(1)p_i(1, j-1) = [\lambda_i(1) + \hat{\mu}_i(1)]p_i(1,j) \tag{3.7}$$

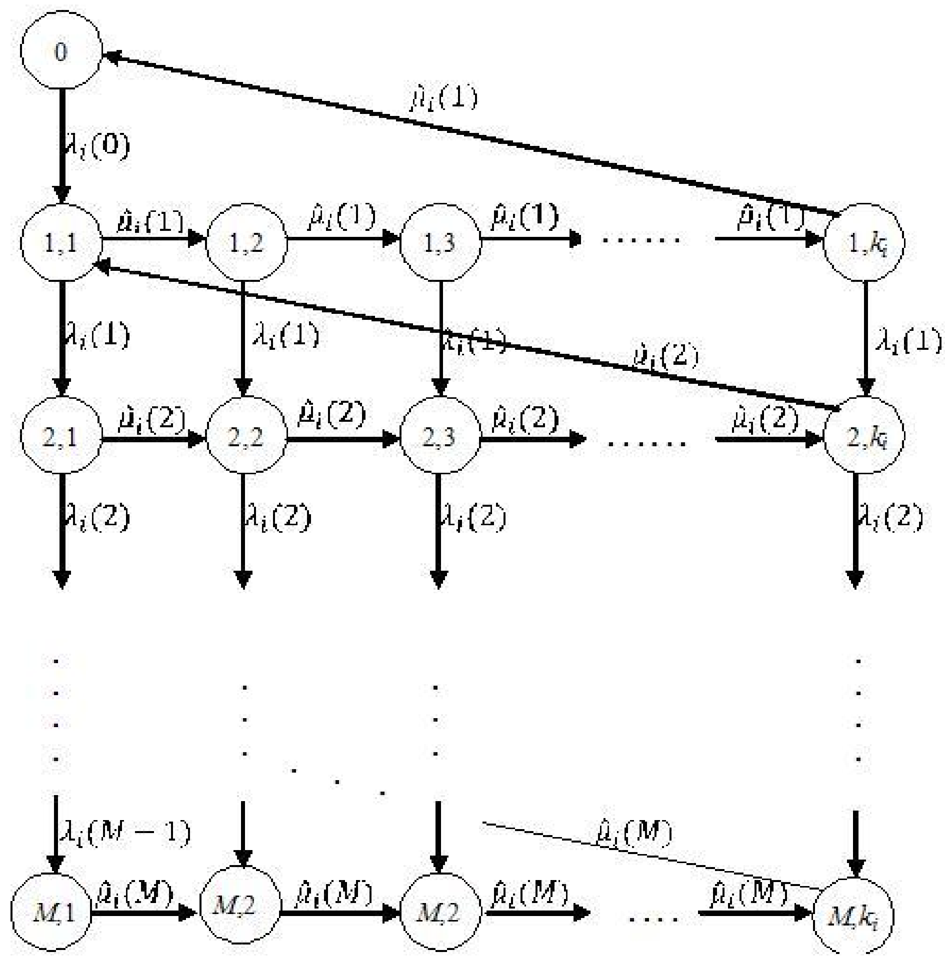

Figure 3.1: State transition diagram of Erlang-k phase type

For the above queue to be in equilibrium, Marie (1980) proves the results in (3.8)-(3.9) for each station $i = 1, 2, \ldots, L$.

$$\hat{\mu}_i(n)p_i(n, k_i) = \lambda_i(n-1)\sum_{j=1}^{k_i} p_i(n-1, j) \quad n = 1,2, \ldots, M \tag{3.8}$$

$$v_i(n)P_i(n) = \hat{\mu}_i(n)p_i(n, k_i) \tag{3.9}$$

where

$$P_i(n) = \sum_{j=1}^{k_i} p_i(n, j) \tag{3.10}$$

Thus, (3.8) becomes:

$$v_i(n) \times P_i(n) = \lambda_i(n-1) \times P_i(n-1) \tag{3.11}$$

And the equilibrium state probabilities should yield:

$$\sum_{j=1}^{M} P_i(n) = 1 \tag{3.12}$$

Improvement Check Test:

After each iteration, we calculate an improvement value, ε, as in (3.13). According to this test, the iterative procedure stops if the current ε value is greater than its previous value. And, the previously obtained performance measures become the solution.

$$\frac{|M - \sum_{i=1}^{L} \sum_{n=1}^{M} nP_i(n)|}{M} = \varepsilon \tag{3.13}$$

The aim of this test is to check whether or not the total number of jobs is equal to the pre-defined M. Smaller the ε value, the more accurate the result.

3.1.3 Algorithm

We can summarize the algorithm for calculating the load-dependent general queuing network as shown below (Ekren and Heragu, 2009):

1. Assign a large value to ε and calculate the load-dependent arrival rates, $\lambda_i(n)$, and service rates, $\hat{\mu}_i(n)$, by (3.1) and (3.2), respectively.
2. Assign an initial value to $p_i(0) = 0.5$.
3. Calculate all the state probabilities, $p_i(n, j)$s, using (3.5), (3.7), (3.6), (3.4), (3.3) in that order.
4. Normalize each $p_i(n, j)$ value by dividing each by the sum of all $p_i(n, j)$'s (see, (3.12)).
5. Compute the conditional throughput of each station, $v_i(n)$, using (3.9).
6. Calculate the new ε value by (3.13). If the current ε value is smaller than the previous one, then adjust the service rates by assigning conditional throughputs, $\mu_i(n) = v_i(n)$. Calculate $\hat{\mu}_i(n)$ by (3.2) and go to step 3; otherwise stop.

3.1.4 Performance measure calculations

After the iterative procedure stops we compute the performance measures by formulas (3.14)-(3.17).

Throughput of each station:

$$\lambda_i(M) = \sum_{n=1}^{M} P_i(n)\mu_i(n) \tag{3.14}$$

Mean number of jobs at each station:

$$\bar{n}_i(M) = \sum_{n=1}^{M} nP_i(n) \tag{3.15}$$

Mean residence time:

$$\bar{t}_i(M) = \frac{\bar{n}_i(M)}{\lambda_i(M)} \tag{3.16}$$

Throughput of the entire network:

$$\frac{1}{L}\sum_{i=1}^{L}\frac{1}{e_i}\sum_{n=1}^{M} P_i(n)v_i(n) \tag{3.17}$$

Here, e_i is the mean number of visits that a job makes to station i and is calculated by (1.1).

3.1.5 Implementation of the algorithm

We implement the proposed algorithm on eight different load-dependent CQNs examples. The algorithm's results are compared with those of simulation. For the simulation model ARENA 12.0, a commercial software, is used (Kelton et al., 2004). Ten independent replications are completed and the results are obtained within a 95% confidence interval. The results show that on average the analytical results deviate less than 5 percent from the simulation results. Our approximation method is thus effective in modeling load-dependent general queuing network with low service time variability. It should be noted that the deviations increases when the c_i^2 values are fairly small and the number of stations is large (Examples 8 and 4). In most cases, the algorithm converges in 3-4 iterations. The deviations are computed by (3.18).

$$\delta = \frac{|\text{Simulation} - \text{Approximation}|}{\text{Simulation}} \times 100 \tag{3.18}$$

In the first three examples, a tandem network, a network having probabilistic routings and a central server model with three stations are considered. m_i shows the number of servers at station i. In the fourth example, the number of stations and jobs in the network are increased to 6 and 20, respectively. When the network's number of stations is large, the deviations between the analytical and simulations results increase. In the fifth example, we implement the algorithm to the real AVS/RS network given in Figure 4.4. We implement this example for three different numbers of jobs - M_1 = 28, M_2 = 21 and M_3 = 13. It should be noted that the deviations between the analytical and simulations results are fairly small in this example. A large network with 4 stations and M = 40 jobs is given in Example 6. The last two examples are given for the networks with equal and small c_i^2 values. As a result, on average the algorithm could estimate the performance measures less than 5% deviation.

Example 1: Tandem network with M = 5 jobs

Station	$1/\mu_i$	m_i	c_i^2
1	0.4	2	0.2
2	0.6	3	0.25
3	0.4	2	0.25

	Approximation	**Simulation**	**δ (%)**
$\bar{n}_1$	1.4626	1.4552	0.5118
$\bar{n}_2$	2.0204	2.0546	1.6626
$\bar{n}_3$	1.4600	1.4900	2.0076
λ	3.1998	3.2981	2.9773

Example 2: Network with probabilistic routings, $M = 9$ jobs

Station	$1/\mu_i$	m_i	c_i^2	p_{i1}	p_{i2}	p_{i3}
1	1	9	0.333	0.0	1	0.0
2	0.2	1	0.1	0.3	0.0	0.7
3	0.5	2	0.2	0.5	0.5	0.0

	Approximation	**Simulation**	**δ (%)**
$\bar{n}_1$	3.1795	3.1431	1.1581
$\bar{n}_2$	3.3916	3.3753	0.4829
$\bar{n}_3$	2.4709	2.4817	0.4352
λ	3.0165	3.1456	4.1054

Example 3: Central server model, M = 8 jobs

Station	$\mathbf{1/\mu_i}$	$\boldsymbol{m_i}$	$\boldsymbol{c_i^2}$	$\boldsymbol{p_{i1}}$	$\boldsymbol{p_{i2}}$	$\boldsymbol{p_{i3}}$
1	2.5	5	0.25	0.0	0.7	0.3
2	2.2	4	0.05	1	0.0	0.0
3	3.8	2	0.2	0.8	0.2	0.0

	Approximation	**Simulation**	**δ (%)**
$\bar{n}_1$	3.4840	3.4989	0.4259
$\bar{n}_2$	2.3484	2.3456	0.1185
$\bar{n}_3$	2.0825	2.1553	3.3777
λ	1.3591	1.3672	0.5939

Example 4: Large tandem network with M = 20 jobs

Station	$\mathbf{1/\mu_i}$	$\boldsymbol{m_i}$	$\boldsymbol{c_i^2}$
1	1	3	0.15
2	2.45	4	0.35
3	3.5	5	0.15
4	0.6	1	0.17
5	2	4	0.07
6	5	7	0.06

	Approximation	Simulation	δ (%)
$\bar{n}_1$	1.1336	1.2636	10.2881
$\bar{n}_2$	3.1061	3.3769	8.0213
$\bar{n}_3$	4.5456	4.9465	8.1047
$\bar{n}_4$	1.3943	1.2870	8.3356
$\bar{n}_5$	2.3129	2.5063	7.7158
$\bar{n}_6$	6.1307	6.6195	7.3844
λ	1.1280	1.2387	8.9368

Example 5: A network representing the real warehouse of an AVS/RS given in Chapter 4, with $M_1 = 28$, $M_2 = 21$ and $M_3 = 13$ jobs.

Station	$1/\mu_i$	m_i	c_i^2	p_{i1}	p_{i2}	p_{i3}
1	0.511	21	0.115	1	0.0	0.0
2	0.413	7	0.233	0.317	0.0	0.683
3	0.756	21	0.065	0.5	0.5	0.0

	M=28 jobs			*M*=21 jobs			*M*=13 jobs		
	Approx.	Simul.	δ (%)	Approx.	Simul.	δ (%)	Approx.	Simul.	δ (%)
$\bar{n}_1$	5.7507	5.6962	0.9566	5.2585	5.2474	0.2111	3.4445	3.4441	0.0116
$\bar{n}_2$	13.5956	13.5651	0.2248	7.5690	7.7074	1.7958	4.2534	4.2666	0.3093
$\bar{n}_3$	8.8077	8.7382	0.7953	8.0984	8.0451	0.6627	5.2909	5.2892	0.0321
λ	11.1226	11.1520	0.2636	10.2373	10.2610	0.2307	6.7469	6.7383	0.1276

Example 6: Large network with 4 stations, M = 40 jobs

Station	$\mathbf{1/\mu_i}$	$\boldsymbol{m_i}$	$\boldsymbol{c_i^2}$	$\boldsymbol{p_{i1}}$	$\boldsymbol{p_{i2}}$	$\boldsymbol{p_{i3}}$	$\boldsymbol{p_{i4}}$
1	0.5	9	0.4	0.0	0.0	0.8	0.2
2	0.2	2	0.05	0.7	0.0	0.3	0.0
3	0.5	10	0.4	0.0	0.2	0.0	0.8
4	1	20	0.2	0.5	0.5	0.0	0.0

	Approximation	**Simulation**	**δ (%)**
$\bar{n}_1$	7.4685	7.5798	1.4685
$\bar{n}_2$	10.3616	10.8161	4.2012
$\bar{n}_3$	7.2022	7.3199	1.6079
$\bar{n}_4$	14.0241	14.2832	1.8127
λ	13.9045	14.0794	1.2427

Example 7: Tandem network with M = 8 jobs and $c_1^2 = c_2^2 = c_3^2 = c_4^2$

Station	$\mathbf{1/\mu_i}$	$\boldsymbol{m_i}$	$\boldsymbol{c_i^2}$
1	0.4	2	0.2
2	1.4	5	0.2
3	0.8	3	0.2
4	0.3	1	0.2

	Approximation	Simulation	δ (%)
$\bar{n}_1$	1.1120	1.0283	8.1396
$\bar{n}_2$	3.4531	3.5629	3.0817
$\bar{n}_3$	2.0665	2.1633	4.4746
$\bar{n}_4$	1.2704	1.2454	2.0074
λ	2.3559	2.4977	5.6772

Example 8: Network with small c_i^2 values and $M = 7$ jobs

Station	$1/\mu_i$	m_i	c_i^2	p_{i1}	p_{i2}	p_{i3}
1	1.2	3	0.02	0.0	0.4	0.6
2	0.5	1	0.01	0.3	0.0	0.7
3	1	3	0.03	0.8	0.2	0.0

	Approximation	Simulation	δ (%)
$\bar{n}_1$	3.2331	3.3630	3.8656
$\bar{n}_2$	1.0681	1.0882	1.8471
$\bar{n}_3$	2.4955	2.5487	2.0873
λ	2.1419	2.2813	6.1105

Consequently, on average the proposed technique can solve the problem with less than 5 percent deviation from the exact (simulation) solutions. Besides, as seen in Example 5, the algorithm could solve the real AVS/RS model as CQN with less than 1% deviation. It should be noted that the deviations increase when the network's service time has very small scv values and/or number of stations is high. However, the proposed technique still can solve the problem in reasonable time and deviation. Therefore, in the following chapter we implement the proposed extended algorithm on our real AVS/RS. By this algorithm, we aggregate and obtain the load-dependent throughput rates of stations other than the synchronization station, given in Figure 4.4.

3.2 Summary

In this chapter, we propose an approximation method for load-dependent general queuing networks having low service time variability. The proposed technique is the extension of Marie's approximation method (Marie, 1980). We solve the problem using a six step algorithm, iteratively. To be able to see the performance of the algorithm, we apply the technique on eight different examples. We believe the algorithm is capable of solving such networks effectively. On average, the results show less than 5 percent deviation from the exact (simulation) solutions. The deviations increase when the network's service time has very small scv values and/or number of stations is high. However, the proposed technique still can solve the problem in reasonable time and deviation. In the following chapter, we use this extended approximate method to calculate the load-dependent throughput rates of the SOQN of an AVS/RS.

CHAPTER 4
APPROXIMATE ANALYSIS OF AVS/RS

In this chapter, we apply an approximate SOQN and the load-dependent CQN methodology developed in Chapter 3, for an existing AVS/RS. The AVS/RS has a non-product-form due to both the general service time and SOQN properties. In the flow of the modeling part first, we describe all possible scenarios and their probabilities of AVS/RS to derive the general service times. Second, we combine all the service times of the network. Third, we model the AVS/RS as SOQN using the extended approximate method shown in the previous chapter. We compare the performance measures with the simulation results.

4.1 SOQN Approach for Modeling AVS/RS

An SOQN consists of jobs, pallets and servers. Each job is paired with a pallet and the two visit the set of servers required for processing the job in the specified sequence. The AVS/RS queuing system can be modeled as an SOQN. In the contest of an AVS/RS, S/R transactions are jobs and the AVs are pallets. The lifts and horizontal travel times to/from a storage space are the servers and the transactions are customers. Each transaction needs a vehicle before it can enter the network. Thus, the total number of transactions in service or waiting for service in the network cannot be larger than the number of vehicles. Figure 4.1 shows the SOQN model of an AVS/RS. If a vehicle is available, then an arriving transaction enters the network of servers immediately, along with this vehicle. Otherwise, it waits in the transaction queue until a vehicle becomes available. When a transaction exits the system, the vehicle returns to a 'vehicle' pool and waits to be paired with the next arriving transaction.

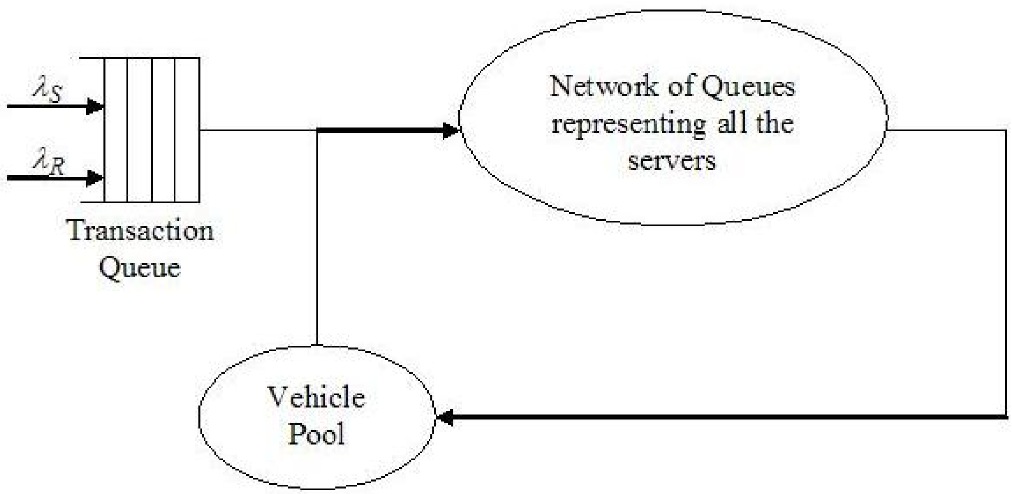

Figure 4.1: SOQN of an AVS/RS

In an AVS/RS, there are two types of transactions arriving into the system -storage and retrieval transactions. Storage transactions refer to the storage of a UL from the I/O point to an available location in the racks. Retrieval transactions refer to the retrieval of a UL from its current location. All storage transactions are assumed to arrive at the I/O point and all retrieval transactions end at the I/O point.

The assumptions and formulations of a typical configuration of an AVS/RS – the one seen in the warehouse in France - are given below:

1. The dwell point of a vehicle is the place where the last storage or retrieval transaction is completed.
2. The dwell point of the lift is where the last vertical movement is completed.
3. The system uses pure random storage policy.

4. The actual distance between two aisles, the width of a bay and the height of one tier in an actual AVS/RS installation are used to derive the expected travel distances.
5. The number of aisles, columns (bays) and tiers of the actual installation are used to derive the probabilities.
6. The storage area is divided into as many zones as there are lifts.
7. Each zone has one lift and three vehicles. Due to lift and vehicle speeds, the company has determined it is best to pair a lift with three vehicles. This allows them to balance the utilization of the two resources so that neither is a bottleneck.
8. Lifts are located on the middle of each individual zone's z-axis (see Figure 4.2).
9. Each zone has I/O locations located near the lift.
10. The arrival rates for storage and retrieval transactions are independent Poisson processes and equal.
11. The transactions are served by the vehicles on a first-come, first-served (FCFS) rule. The vehicles requiring lifts for vertical movement are also served by FCFS order. Although the company sometimes uses a closest-request-first dispatching strategy, because our analytical model can only handle FCFS rule, we make this assumption.
12. The vehicle transfer time into the lift is assumed to be zero.

The notation used in the AVS/RS model is given below:

A : number of aisles

C : number of bays (columns) per aisle

T : number of tiers

D : the distance between two aisles

W : the width of one storage bay

H : the height of one tier

V : the number of vehicles

N_L : the number of lifts

Y_x : the distance from the firts bay to the

T_T : load or unload transfer time between cross aisle

the lift and the I/O point.

v_L : the velocity of the lift

v_v : the velocity of the vehicle

λ_S : the arrival rate of storage transactions per hour

$T_{L/U}$: the time to load/unload to or from the storage rack

λ_R : the arrival rate of retrieval transactions per hour

The rack configuration of the AVS/RS under study is shown in Figure 4.2.

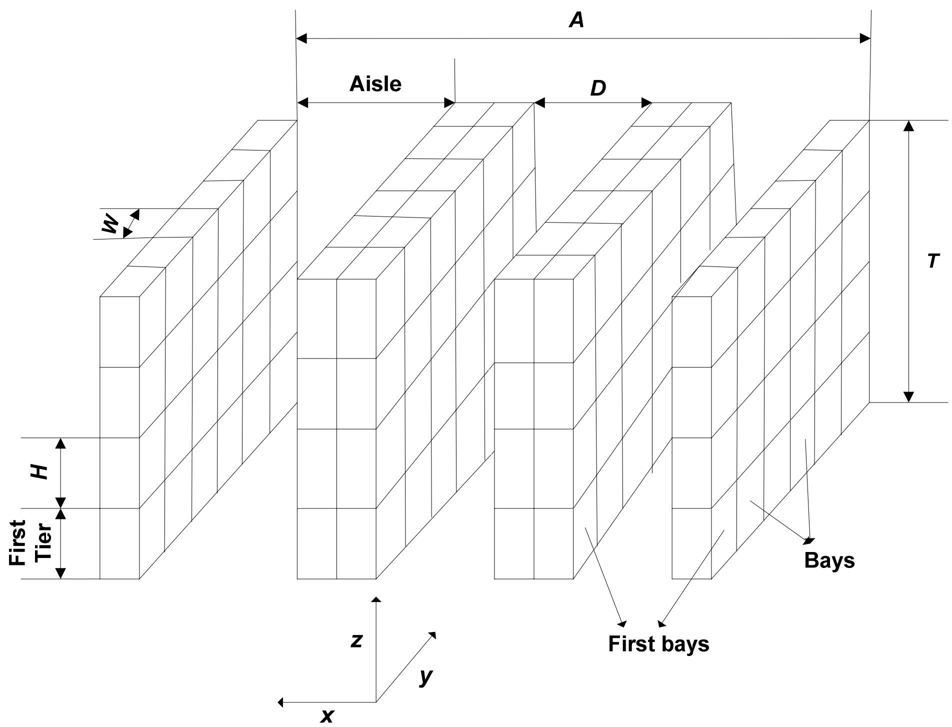

Figure 4.2: The rack configuration of the AVS/RS

Because of the agreement with the company, we do not give the specific values of the parameters defined above, except the values summarized below:

A	= 42	N_L	= 7	C	= 27
T	= 7	λ_S	= 225 ULs	λ_R	= 225 ULs

At most, three ULs can be accommodated in each bay. So, the total storage capacity of the AVS/RS system is 27 × 3 × 2 × 42 × 7 = 47,628 UL positions. Each aisle has S/R locations on both sides.

4.2 Scenarios For Movement Kinematics

To be able to derive the expected service time of a transaction, all probable scenarios are considered. In this section, we define all probable scenarios of transactions and their probabilities. Because of the service completion dwell point policy, an idle vehicle can be at the I/O point or at a storage rack. Also, because the expected travel distance depends on the position of the vehicle and the requested storage or retrieval location, the scenarios are described according to the idle vehicle's position and the requested location. Consequently, the storage transaction can have six, and the retrieval transactions can have seven travel scenarios.

4.2.1 Six travel scenarios for storage transactions

We define six travel scenarios and their probabilities for the storage transactions. *Sf* indicates that the requested storage location is on the first tier. *Sn* means the requested storage location is not on the first tier. *Vo* and *Vf* denote that the seized vehicle is at the I/O point, and at the first tier but not at the I/O point, respectively. *Vn* indicates that the seized vehicle is not on the first tier.

Scenario 1 (VnSn) - The seized vehicle is not on the first tier and the requested storage position is not on the first tier. Under this condition, the expected travel distance is calculated by considering five movements:

(i) The vehicle travels from its current position to the lift,

(ii) Lift travels from its current tier to the vehicle's tier,

(iii) The vehicle travels in the lift to the I/O tier,

(iv) Load is charged on the vehicle and, lift and vehicle travel to the destination tier,

(v) The vehicle travels to the destination storage location and discharges the load.

Scenario 2 (VnSf) - The seized vehicle is not on the first tier and the requested storage position is on the first tier. Under this condition:

(i) The vehicle travels from its current position to the lift,

(ii) Lift travels from the current tier to the vehicle's tier,

(iii) Vehicle in the lift travels to the I/O,

(iv) Load is charged on the vehicle,

(v) The vehicle travels to the destination storage location and discharges the load.

Scenario 3 (VfSn) - The seized vehicle is on the first tier and the requested storage position is not on the first tier. Under this condition:

(i) The vehicle travels from its current position to the lift,

(ii) Lift travels to the first tier,

(iii) Load is charged on the vehicle and, lift and vehicle travels to the destination tier,

(iv) The vehicle travels to the destination storage location and discharges the load.

Scenario 4 (VoSn) - The seized vehicle is at the I/O point and the requested storage position is not on the first tier. The expected travel distance is:

(i) Lift travels from its current tier to the first tier,

(ii) Load is charged on the vehicle and, the lift and vehicle travel to the destination tier,

(iii) The vehicle travels to the destination storage location and discharges the load.

Scenario 5 (VfSf) - The seized vehicle is on the first tier and the requested storage position is on the first tier. The expected travel distance is:

(i) The vehicle travels from its current position to the lift,

(ii) Lift travel from its current tier to the first tier,

(iii) Load is charged on the vehicle,

(iv) The vehicle travels to the destination storage location and discharges the load.

Scenario 6 (VoSf) - The seized vehicle is at the I/O point and the requested storage position is on the first tier.

(i) Lift travels from its current tier to the first tier,

(ii) Load is charged on the vehicle,

(iii) The vehicle travels to the destination storage location and discharges the load.

4.2.2 Storage scenarios' probability calculations

The occurence probabilities for the above six scenarios are calculated as below:

Scenario 1:

Pr (*VnSn*) = *Pr* (the transaction type is storage) × *Pr* (the seized vehicle is not on the first tier) × *Pr* (requested storage position is not on the first tier)

Pr (the transaction type is storage) = $\frac{\lambda_S}{\lambda_S+\lambda_R}$

Pr (the seized vehicle is not on the first tier) = $\frac{T-1}{T} \times$ *Pr* (the vehicle is on racks)

Pr (the seized vehicle is not on the first tier) = $\frac{T-1}{T} \times \frac{\lambda_S}{\lambda_S+\lambda_R}$

Pr (requested storage position is not on the first tier) = $\frac{T-1}{T}$

$$Pr\ (VnSn) = \left(\frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{T-1}{T} \times \frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{T-1}{T}\right)$$

Scenario 2:

Pr (*VnSf*) = *Pr* (the transaction type is storage) × *Pr* (the seized vehicle is not on the first tier) × *Pr* (requested storage position is on the first tier)

$$Pr\ (VnSf) = \left(\frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{T-1}{T} \times \frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{1}{T}\right)$$

Scenario 3:

Pr (*VfSn*) = *Pr* (the transaction type is storage) × *Pr* (the seized vehicle is on the first tier) × *Pr* (requested storage position is not on the first tier)

$$Pr\ (VfSn) = \left(\frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{1}{T} \times \frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{T-1}{T}\right)$$

Scenario 4:

Pr (*VoSn*) = *Pr* (the transaction type is storage) × *Pr* (the seized vehicle is at the I/O) × *Pr* (requested storage position is not on the first tier)

Pr (the seized vehicle is at the I/O) = $\frac{\lambda_R}{\lambda_S+\lambda_R}$

$$Pr\ (VoSn) = \left(\frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{\lambda_R}{\lambda_S+\lambda_R}\right) \times \left(\frac{T-1}{T}\right)$$

Scenario 5:

Pr (*VfSf*) = *Pr* (the transaction type is storage) × *Pr* (the seized vehicle is on the first tier) × *Pr* (requested storage position is on the first tier)

$$Pr\ (VfSf) = \left(\frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{1}{T} \times \frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{1}{T}\right)$$

Scenario 6:

Pr (*VoSf*) = *Pr* (the transaction type is storage) × *Pr* (the seized vehicle is at the I/O) × *Pr* (requested storage position is on the first tier)

$$Pr\ (VoSf) = \left(\frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{\lambda_R}{\lambda_S+\lambda_R}\right) \times \left(\frac{1}{T}\right)$$

The probability calculations are summarized in Table 4.1:

Table 4.1: Probabilities of six storage scenarios

$Pr\ (S_1)$	=	$\left(\frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{T-1}{T} \times \frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{T-1}{T}\right)$
$Pr\ (S_2)$	=	$\left(\frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{T-1}{T} \times \frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{1}{T}\right)$
$Pr\ (S_3)$	=	$\left(\frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{1}{T} \times \frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{T-1}{T}\right)$
$Pr\ (S_4)$	=	$\left(\frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{\lambda_R}{\lambda_S+\lambda_R}\right) \times \left(\frac{T-1}{T}\right)$
$Pr\ (S_5)$	=	$\left(\frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{1}{T} \times \frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{1}{T}\right)$
$Pr\ (S_6)$	=	$\left(\frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{\lambda_R}{\lambda_S+\lambda_R}\right) \times \left(\frac{1}{T}\right)$

4.2.3 Seven travel scenarios for retrieval transactions

The same calculations are also completed for the retrieval transactions. Because the travel sequences are different from the storage transactions, we describe one more scenario in the retrieval scenarios. This is the first scenario listed below. *Rf* and *Rn* denote that the retrieval location is (is not) on the first tier. *Vo*, *Vf*, *Vn* denote the same idea for storage transaction scenarios. The calculations are summarized below:

Scenario 1 (Vn_1Rn_2): The seized vehicle and the requested retrieval location are not on the same tier and the retrieval location is not on the first tier. Under this condition, the expected travel distance is calculated by considering seven movements:

(i) The vehicle travels from its current position to the lift,

(ii) Lift travels from its current tier to the vehicle's tier,

(iii) The vehicle travels in the lift to the destination tier,

(iv) The vehicle travels to the retrieval location and the load is charged on the vehicle,

(v) The vehicle travels from its current position to the lift,

(vi) Lift travels from its current tier to the vehicle's tier,

(vii) The vehicle travels in the lift to the first tier (I/O) and discharges the load.

Scenario 2 (VoRn): The seized vehicle is at the I/O point and the retrieval location is not on the first tier. Under this condition, the expected travel distance is calculated by:

(i) Lift travels from its current tier to the I/O,

(ii) The vehicle travels in the lift to the destination tier,

(iii) The vehicle travels to the retrieval location and the load is charged on the vehicle,

(iv) The vehicle travels from its current position to the lift,

(v) Lift travels from its current tier to the vehicle's tier,

(vi) The vehicle travels in the lift to the first tier and discharges the load.

Scenario 3 (VnRn): The seized vehicle and the retrieval location are on the same tier other than the first. The expected travel distance is calculated by:

(i) The vehicle travels to the retrieval location and the load is charged on the vehicle,

(ii) The vehicle travels from its current position to the lift,

(iii) Lift travels from its current tier to the vehicle's tier,

(iv) The vehicle travels in the lift to the first tier and discharges the load.

Scenario 4 (VnRf): The seized vehicle is not on the first tier and the retrieval location is on the first tier. The expected travel distance is calculated by:

(i) The vehicle travels from its current position to the lift,

(ii) Lift travels from its current tier to the vehicle's tier,

(iii) The vehicle travels in the lift to the first tier,

(iv) The vehicle travels to the retrieval location and the load is charged on the vehicle,

(v) The vehicle travels from its current position to the lift,

(vi) Lift travels from its current tier to the first tier and the load is discharged.

Scenario 5 (VfRf): The seized vehicle and the retrieval location are on the first tier. Under this condition, the expected travel distance is calculated by:

(i) The vehicle travels from its current position to the retrieval location and the load is charged on the vehicle,

(ii) The vehicle travels from its current position to the lift,

(iii) Lift travels from its current tier to the first tier and the load is discharged.

Scenario 6 (VoRf): The seized vehicle is at the I/O point and the retrieval location is on the first tier. The expected travel distance is:

(i) Lift travels from its current tier to the first tier,

(ii) The vehicle travels from I/O to the retrieval location and the load is charged on the vehicle,

(iii) The vehicle travels from its current position to the lift,

(iv) The lift travels from its current tier to the first tier and the load is discharged.

Scenario 7 (VfRn): The seized vehicle is on the first tier and retrieval location is not on the first tier. The expected travel distance is:

(i) The vehicle travels from its current position to the lift,

(ii) Lift travels from its current tier to the first tier,

(iii) The vehicle travels in the lift to the destination tier,

(iv) The vehicle travels to the retrieval position and the load is charged on the vehicle,

(v) The vehicle travels from its current position to the lift,

(vi) Lift travels from its current tier to the vehicle's tier,

(vii) The vehicle travels in the lift to the first tier and discharges the load.

4.2.4 Retrieval scenarios' probability calculations

The occurence probabilities for the above seven scenarios are calculated as:

Scenario 1:

Pr (Vn_1Rn_2) = *Pr* (the transaction type is retrieval) × *Pr* (the seized vehicle is not on the first and same tier with the retrieval position) × *Pr* (the requested retrieval position is not on the first tier)

Pr (the transaction type is retrieval) = $\frac{\lambda_R}{\lambda_S+\lambda_R}$

Pr (the seized vehicle is not on the first and same tier with the retrieval position) = $\frac{T-2}{T}$ × *Pr* (the vehicle is on racks)

Pr (the seized vehicle is not on the first and same tier with the retrieval position) = $\frac{T-2}{T} \times \frac{\lambda_S}{\lambda_S+\lambda_R}$

Pr (the requested retrieval position is not on the first tier) = $\frac{T-1}{T}$

$$Pr\ (Vn_1Rn_2) = \left(\frac{\lambda_R}{\lambda_S+\lambda_R}\right) \times \left(\frac{T-2}{T} \times \frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{T-1}{T}\right)$$

Scenario 2:

Pr (*VoRn*) = *Pr* (the transaction type is retrieval) × *Pr* (the seized vehicle is at the I/O point) × *Pr* (the requested retrieval position is not on the first tier)

$$Pr\ (VoRn) = \left(\frac{\lambda_R}{\lambda_S+\lambda_R}\right) \times \left(\frac{\lambda_R}{\lambda_S+\lambda_R}\right) \times \left(\frac{T-1}{T}\right)$$

Scenario 3:

Pr (*VnRn*) = *Pr* (the transaction type is retrieval) × *Pr* (the seized vehicle and the retrieval location are on the same tier other than the first) × *Pr* (the requested retrieval position is not on the first tier)

$$Pr\ (VnRn) = \left(\frac{\lambda_R}{\lambda_S+\lambda_R}\right) \times \left(\frac{1}{T} \times \frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{T-1}{T}\right)$$

Scenario 4:

Pr (*VnRf*) = *Pr* (the transaction type is retrieval) × *Pr* (the seized vehicle is not on the first) × *Pr* (the requested retrieval position is on the first tier)

$$Pr\ (VnRf) = \left(\frac{\lambda_R}{\lambda_S+\lambda_R}\right) \times \left(\frac{T-1}{T} \times \frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{1}{T}\right)$$

Scenario 5:

Pr (*VfRf*) = *Pr* (the transaction type is retrieval) × *Pr* (the seized vehicle is on the first) × *Pr* (the requested retrieval position is on the first tier)

$$Pr\ (VfRf) = \left(\frac{\lambda_R}{\lambda_S+\lambda_R}\right) \times \left(\frac{1}{T} \times \frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{1}{T}\right)$$

Scenario 6:

Pr (*VoRf*) = *Pr* (the transaction type is retrieval) × *Pr* (the seized vehicle is at the I/O point) × *Pr* (the requested retrieval position is on the first tier)

$$Pr\ (VoRf) = \left(\frac{\lambda_R}{\lambda_S+\lambda_R}\right) \times \left(\frac{\lambda_R}{\lambda_S+\lambda_R}\right) \times \left(\frac{1}{T}\right)$$

Scenario 7:

Pr (*VfRn*) = *Pr* (the transaction type is retrieval) × *Pr* (the seized vehicle is on the first tier) × *Pr* (the requested retrieval position is not on the first tier)

$$Pr\ (VfRn) = \left(\frac{\lambda_R}{\lambda_S+\lambda_R}\right) \times \left(\frac{1}{T} \times \frac{\lambda_S}{\lambda_S+\lambda_R}\right) \times \left(\frac{T-1}{T}\right)$$

The probability calculations are summarized in Table 4.2:

Table 4.2: Probabilities of seven retrieval scenarios

$Pr\ (R_1) =$	$\left(\frac{\lambda_R}{\lambda_S + \lambda_R}\right) \times \left(\frac{T-2}{T} \times \frac{\lambda_S}{\lambda_S + \lambda_R}\right) \times \left(\frac{T-1}{T}\right)$
$Pr\ (R_2) =$	$\left(\frac{\lambda_R}{\lambda_S + \lambda_R}\right) \times \left(\frac{\lambda_R}{\lambda_S + \lambda_R}\right) \times \left(\frac{T-1}{T}\right)$
$Pr\ (R_3) =$	$\left(\frac{\lambda_R}{\lambda_S + \lambda_R}\right) \times \left(\frac{1}{T} \times \frac{\lambda_S}{\lambda_S + \lambda_R}\right) \times \left(\frac{T-1}{T}\right)$
$Pr\ (R_4) =$	$\left(\frac{\lambda_R}{\lambda_S + \lambda_R}\right) \times \left(\frac{T-1}{T} \times \frac{\lambda_S}{\lambda_S + \lambda_R}\right) \times \left(\frac{1}{T}\right)$
$Pr\ (R_5) =$	$\left(\frac{\lambda_R}{\lambda_S + \lambda_R}\right) \times \left(\frac{1}{T} \times \frac{\lambda_S}{\lambda_S + \lambda_R}\right) \times \left(\frac{1}{T}\right)$
$Pr\ (R_6) =$	$\left(\frac{\lambda_R}{\lambda_S + \lambda_R}\right) \times \left(\frac{\lambda_R}{\lambda_S + \lambda_R}\right) \times \left(\frac{1}{T}\right)$
$Pr\ (R_7) =$	$\left(\frac{\lambda_R}{\lambda_S + \lambda_R}\right) \times \left(\frac{1}{T} \times \frac{\lambda_S}{\lambda_S + \lambda_R}\right) \times \left(\frac{T-1}{T}\right)$

The probabilities in Tables 4.1 and 4.2 are calculated on the assumption that some of the factors are independent and that the vehicles are uniformly distributed across the tiers. These are reasonable because of the random storage policy observed in many automated warehouses, including the one in France. Also, the probabilities in Tables 4.1 and 4.2 sum to one. To be able to treat the system as a single class network, in Section 4.3 we show the combination procedure for these scenarios.

4.3 Customer Combination

Each of the storage and retrieval transaction scenarios has different routes and service times. Therefore, each scenario can be seen as a new transaction (customer) type in the queuing system. Thus, we have a multi-product SOQN which has thirteen types of transactions (customers).

The multi-class queuing network can be treated as a single-class network using Expressions (4.1)-(4.6) (see Whitt, 1983). Notations that are used for this technique are given below:

L	: number of stations	m_j	: number of servers at station j
k	: number of classes	p_{ij}^r	: routing probability of class r
λ_0^r	: external arrival rate of class r	c_j^2	: scv of the service-time distribution of station j
λ_i^r	: arrival rate of class r to station i	τ_j^r	: the mean service time of class r at the jth station of its route

c_{rj}^2 : the scv of the service-time distribution of class r at the jth station of its route

First, we obtain the external arrival rates to each station, as shown in (4.1):

$$\lambda_{0j} = \sum_{r=1}^{k}(b \times \lambda_0^r) \quad \begin{cases} b = 1, & \text{if the station } j \text{ is the first station on route of class } r \\ 0, & \text{otherwise} \end{cases} \quad , \forall j = 1,2,\dots,L \tag{4.1}$$

The external arrival rate at station j, λ_{0j}, is the sum of all arrival rates of classes whose first station is j. The flow rate from i to j is calculated by (4.2), the flow from i out of the network is calculated by (4.3).

$$\lambda_{ij} = \sum_{r=1}^{k}\left(\lambda_i^r \times p_{ij}^r\right) \quad \forall\, i,j = 1,2,..,L \tag{4.2}$$

$$\lambda_{i0} = \sum_{r=1}^{k} (\lambda_i^r \times p_{i0}^r) \;\; \forall\, i = 1,2,..,L \tag{4.3}$$

From Equations (4.2) and (4.3) the overall routing matrix is obtained by (4.4).

$$P_{ij} = \frac{\lambda_{ij}}{\lambda_{i0} + \sum_{l=1}^{n} \lambda_{il}} \quad \forall\, i,j = 1,2,..,n \tag{4.4}$$

The combined mean and the scv of service time of each station are obtained by (4.5)-(4.6):

$$\tau_j = \frac{\sum_{r=1}^{k} (\tau_j^r \times \lambda_j^r)}{\lambda_j}, \qquad \forall j = 1,2,\dots,n \tag{4.5}$$

$$\tau_j (c_j^2 + 1) = \frac{\sum_{r=1}^{k} ((\tau_j^r)^2 \times \lambda_j^r \times (c_{rj}^2 + 1))}{\lambda_j}, \qquad \forall j = 1,2,\dots,n \tag{4.6}$$

4.4 Service Time Calculations

The general view of SOQN of the AVS/RS is illustrated in Figure 4.3. According to that figure there are two types of physical queues in the system, one is the vehicle queue and the other is the lift queue. Transactions arriving to the system first enter the vehicle queue, then the lift queue. In the storage transactions, the vehicles travel to the I/O point to pick up the load. In the retrieval transactions, the vehicles travel to the load's location to retrieve it. For both transactions, the vehicles use a lift if it must travel to a tier other than its current tier. In Figure 4.3, the server VT_1 corresponds to the vehicle's horizontal travel from its current position to the lift's location. *LT* corresponds to the lift's travel. Lifts are released after completion of vertical travel. There are as many lifts in the entire system as there are parallel servers in *LT*.

VT_2 corresponds to the vehicle's horizontal travel to complete a storage or retrieval. In other words, it is the travel from the vehicle's current position to the storage/retrieval position after it reaches the destination tier, including load/unload activities. The difference between VT_1 and VT_2 is that, VT_2 considers the vehicle's travel from the lift location to the destination place plus the unload/load activities for the storage/retrieval process. VT_1 is the vehicle's travel to the lift not including load/unload. In the real AVS/RS system, the vehicle is seized and not released until it completes the entire storage or retrieval. VT_1 and VT_2 are servers used to capture vehicle travel times and thus are not real servers. Therefore, dummy vehicle queues are shown before VT_1 and VT_2 in Figure 4.3. It is assumed that there are as many vehicles in the entire system as there are parallel servers in VT_1 and VT_2.

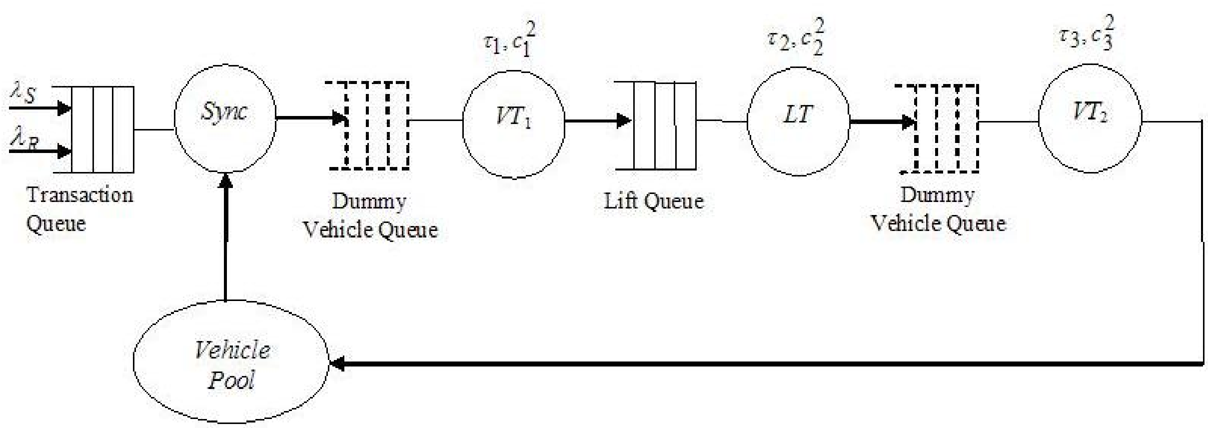

Figure 4.3: SOQN of the AVS/RS

We also consider a synchronization station after the external queue where transactions and vehicles synchronize. After a vehicle completes a storage or a retrieval, it becomes free and returns to the vehicle pool to be used by a waiting transaction.

Each scenario has its own route and service times at each station. For instance, because the vehicles are at the I/O point in the S_4 and S_6 storage scenarios, the vehicles will not visit VT_1. In addition, in the S_3 and S_5 retrieval scenarios, the transactions will go to VT_2 directly after seizing the vehicle.

In the following sections, the derivation of the service times of VT_1, LT, and VT_2 of each scenario are shown. Because the service times have general distributions, we derive both means and the scv of all the service times.

4.4.1 Storage transaction service time calculations

For the service time calculations we consider all probable scenarios. For example, VT_1 is the vehicle travel time from its current position to the lift. Because the lift's location is fixed, the vehicle's travel from its current position to the lift depends only on the vehicle's current position. The vehicle can be at any of the 27 bays and 6 aisles. Therefore, we consider 162 (27×6) scenarios for this travel time. The travel distance is calculated along two axes, x and y. The travel on the x-axis shows the travel between the aisles, whereas travel on the y-axis shows the travel between bays (see Fig. 4.2). As a result, the total travel time is calculated by dividing the total distance to vehicle's velocity. In the service time calculations we also consider the vehicle's and lift's acceleration and deceleration times, and the vehicle's turning delays when changing its current coordinate. All these are based on company supplied data. Consequently, the expected arrival rates and service times of six storage scenarios are obtained as shown in Table 4.3.

Table 4.3: Expected service times of six storage scenarios

Scenario	Arrival Rates	Mean Service Time and scv (min)
Scenario 1(VnSn):		
VT_1 LT VT_2	$\left[\left(\frac{\lambda_S}{\lambda_S+\lambda_R}\right)\times\left(\frac{T-1}{T}\right)\times\left(\frac{\lambda_S}{\lambda_S+\lambda_R}\times\frac{T-1}{T}\right)\right]\times 450\ \text{units/min} = \frac{18}{98}\times 450\ \text{units/min}$	$(0.511, 0.115)$ $(0.631, 0.016)$ $(0.744, 0.054)$
Scenario 2 (VnSf):		
VT_1 LT VT_2	$\frac{3}{98}\times 450\ \text{units/min}$	$(0.511, 0.115)$ $(0.493, 0.014)$ $(0.744, 0.054)$
Scenario 3 (VfSn):		
VT_1 LT VT_2	$\frac{3}{98}\times 450\ \text{units/min}$	$(0.511, 0.115)$ $(0.492, 0.022)$ $(0.744, 0.054)$
Scenario 4 (VoSn):		
VT_1 LT VT_2	$\frac{3}{14}\times 450\ \text{units/min}$	Null $(0.492, 0.022)$ $(0.744, 0.054)$
Scenario 5 (VfSf):		
VT_1 LT VT_2	$\frac{1}{196}\times 450\ \text{units/min}$	$(0.511, 0.115)$ $(0.492, 0.022)$ $(0.744, 0.054)$
Scenario 6 (VoSf):		
VT_1 LT VT_2	$\frac{1}{28}\times 450\ \text{units/min}$	Null $(0.492, 0.022)$ $(0.744, 0.054)$

From Table 4.3, it can be seen that all the servers are visited in each scenario except the scenarios, 4 and 6. In scenarios 4 and 6, transactions do not visit VT_1 because the vehicles are at the I/O point and they do not need to realize VT_1 (see Section 4.2.1).

4.4.2 Retrieval transaction service time calculations

The same calculations are also completed for retrieval transaction scenarios (see table 4.4).

Table 4.4: Expected service times of seven retrieval scenarios

Scenario	**Arrival Rates**	**Mean Service Time and scv (min)**
Scenario 1 (Vn1Rn2):		
VT_1 LT VT_2 VT_1 LT	$\frac{15}{98} \times 450$ units/min	$(0.511, 0.115)$ $(0.196, 0.132)$ $(0.744, 0.054)$ $(0.511, 0.115)$ $(0.493, 0.014)$
Scenario 2 (V0Rn):		
VT_1 LT VT_2 VT_1 LT	$\frac{3}{14} \times 450$ units/min	Null $(0.229, 0.162)$ $(0.744, 0.054)$ $(0.511, 0.115)$ $(0.493, 0.014)$
Scenario 3 (VnRn):		

VT_1 LT VT_2 VT_1 LT	$\frac{3}{98} \times 450$ units/min	Null Null (1.088, 0.105) (0.511, 0.115) (0.493, 0.014)
Scenario 4 (VnRf):		
VT_1 LT VT_2 VT_1 LT	$\frac{3}{98} \times 450$ units/min	(0.511, 0.115) (0.231, 0.12) (0.744, 0.054) (0.511, 0.115) (0.492, 0.022)
Scenario 5 (VfRf):		
VT_1 LT VT_2 VT_1 LT	$\frac{1}{196} \times 450$ units/min	Null Null (1.088, 0.105) (0.511, 0.115) (0.492, 0.022)
Scenario 6 (VoRf):		
VT_1 LT VT_2 VT_1 LT	$\frac{1}{28} \times 450$ units/min	Null (0.092, 0.641) (0.744, 0.054) (0.511, 0.115) (0.492, 0.022)
Scenario 7 (VfRn):		
VT_1 LT VT_2 VT_1 LT	$\frac{3}{98} \times 450$ units/min	(0.511, 0.115) (0.229, 0.162) (0.744, 0.054) (0.511, 0.115) (0.493, 0.014)

Unlike a storage transaction queuing system, transactions may visit a station twice in the retrieval transaction. For example, VT_1 and LT are visited twice in scenarios 1, 4, and 7.

4.4.3 Storage and retrieval transaction service time combinations

In this section, we aggregate storage and retrieval transactions' service times, obtained in the previous section, using the Section 4.3 formulations (4.2)-(4.6).

The combined figure of the network is illustrated in Figure 4.4. According to this, there are two ways to leave the network, from LT or VT_2. This is because the storage transactions leave the system after VT_2 and the retrieval transactions leave the system after LT. Consequently, the combined P, τ, and c_s^2 values are obtained as:

$P_1 = 0.464$, $P_2 = 0.5$, $P_3 = 0.036$, $P_4 = 0.659$, $P_5 = 0.341$, $P_6 = 0.5$, $P_7 = 0.5$

$\tau_1 =$ 0.511 min. $c_1^2 =$ 0.115 (VT_1)
$\tau_2 =$ 0.413 min. $c_2^2 =$ 0.233 (LT)
$\tau_3 =$ 0.756 min. $c_3^2 =$ 0.065 (VT_2)

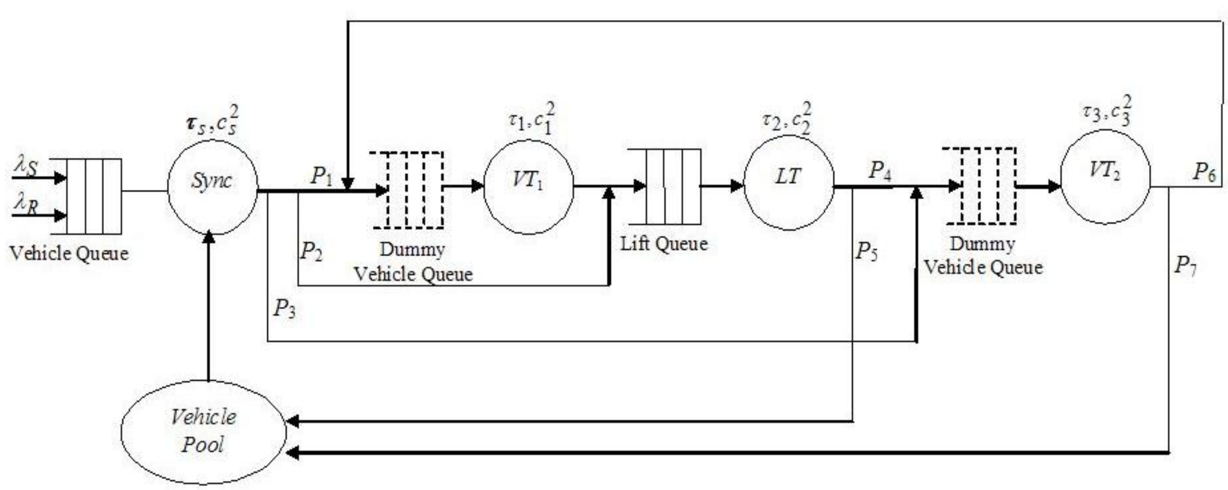

Figure 4.4: SOQN of combined storage and retrieval transactions

After verifying the VT_1, LT and VT_2 time values using simulation, we proceed to analyze further.

4.5 Approximate Solution and Performance Measure Calculations of AVS/RS

In this section, we demonstrate an approximation solution procedure for the SOQN of the AVS/RS. This technique is an extension of an aggregation technique originally proposed by Avi-Itzhak and Heyman (1973) for single-class exponential networks and follows four steps:

1. Transform the SOQN into a CQN in which transactions are seen as customers. Identify the synchronization station as the first station where transactions may have to wait for a vehicle.
2. Consider all the stations except the synchronization station as a CQN, and obtain the load-dependent throughput rates.
3. Replace the synchronization station by a load-dependent exponential server with the values obtained in step 2.
4. Find the mean number of transactions in the external queue by considering the synchronization station in isolation and by solving the birth and death process ($M/M/1$) that describes its behavior.

We use Marie's extended approximation shown in the previous chapter to calculate the load-dependent throughput rates in Step 2.

4.5.1 SOQN solution of AVS/RS

The four step approximate SOQN algorithm in Section 4.5 is implemented for two different numbers of vehicles (M_1 = 21, M_2 = 28) and four different arrival rates, $\lambda_1 = 425, 450, 475, 500$ and $\lambda_2 = 475, 500, 525, 550$ for the AVS/RS. After obtaining the load-dependent throughput rates using the approximate technique explained in the previous section, the synchronization station is solved as *M*/*M*/1 (birth and death process). The state space solution is shown in Figure 4.5. According to this figure (i, j) represents the state of the synchronization station where *i* is the number of transactions waiting in the external queue and *j* is the number of transactions in the synchronization station. It should be noted that *j* cannot be greater than *M* which represents the number of vehicles in the system. After the number of transactions in the synchronization station reaches to *M*, the arriving transactions starts to wait in the external queue. In the figure, λ values are the external arrival rates and the service rates are load-dependent service rates which are obtained by the previous section's technique. Four different performance measures for the AVS/RS system are observed from the analytical solution. These are, the external queue length (*Leq*), average number of transactions (vehicles) in the network including waiting for service (*Lv*), average number of vehicles in the vehicle pool (*Lp*) and average waiting time in the external queue (*Weq*).

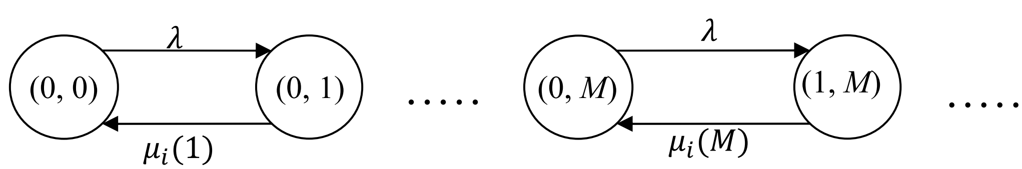

Figure 4.5: State space for *M*/*M*/1 queue

The results are compared with the simulation results. The simulation model is run for two years, with three months warm-up period. Ten independent replications are completed and the results are obtained within 95% confidence interval. The results are illustrated in Tables 4.5 and 4.6.

Table 4.5: Analytical and simulation results ($M = 21$)

	$\lambda = 425$		$\lambda = 450$		$\lambda = 475$		$\lambda = 500$	
	Analy.	Simul.	Analy.	Simul.	Analy.	Simul.	Analy.	Simul.
Leq	0.4872	0.8616	0.7990	1.2613	1.6363	1.9102	3.4592	3.1153
Ln	15.1473	15.1094	16.1380	16.1356	17.1520	17.1682	18.1890	18.2153
Lp	5.8532	5.8905	4.8619	4.8644	3.8480	3.8317	2.8110	2.7847
Weq	0.4814	0.8516	0.7457	1.1773	1.4468	1.6893	2.9057	2.6173

Table 4.6: Analytical and simulation results ($M = 28$)

	$\lambda = 475$		$\lambda = 500$		$\lambda = 525$		$\lambda = 550$	
	Analy.	Simul.	Analy.	Simul.	Analy.	Simul.	Analy.	Simul.
Leq	0.22819	0.5481	0.58670	0.7879	1.3067	1.1523	2.4984	1.7379
Ln	18.1610	18.4861	19.6070	19.7740	21.2550	21.1015	23.0560	22.4570
Lp	9.8390	9.5138	8.3930	8.2261	6.7450	6.8985	4.9440	5.5430
Weq	0.2017	0.4847	0.4928	0.6619	1.0453	0.9219	1.9078	1.3274

Several observations from Tables 4.5 and 4.6 are summarized below.

- Our extended algorithm works reasonably well. It estimates the *Ln* and *Lp* values within 5% of the simulation values for multiple values of arrival rates and number of vehicles.

- The algorithm works better – provides better estimates of the transaction waiting time in the external queue (*Weq*) – under heavy traffic conditions than low traffic conditions. A similar result was also shown in Avi-Itzhak and Heyman (1973). The reason may be due to the treatment of the synchronization station as an exponential single server and solving the model as *M*/*M*/1.
- The proposed solution procedure provides reasonably good solutions for the SOQN having load-dependent general service time with low variability.

4.6 Summary

In this chapter, we apply an approximate methodology developed in Chapter 3 for an AVS/RS. First, the mean service times and their scv values are derived using probabilities of pre-defined scenarios of an AVS/RS. Second, the AVS/RS system is modeled as a SOQN by considering a synchronization station in the network. Third, to obtain load-dependent throughput rates of the CQN, Marie's extended approximation is used. Fourth, the load-dependent throughput rates are assigned as exponential service times to the synchronization station. Last, the remaining system is solved as M/M/1, birth and death process. We implement the algorithm for two different numbers of vehicles. Four different performance measures are obtained from the model and compared with the simulation results. The performance measures are shown in Tables 4.5and 4.6. The results show that the key performance measure estimates such as the number of vehicles inside the network or at the vehicle pool are within 5% of the simulation estimates under alternate conditions.

CHAPTER 5
SIMULATION MODELING OF AVS/RS

In this chapter, we develop simulation model of AVS/RS. In this thesis, the simulation model of the system is used for two objectives. The first one is that the developed model is used for validating the approximate analytical model and also, will be used for the MGM results of the AVS/RS. The second objective is that by the simulating the system, we perform a DOE for the AVS/RS to identify factors affecting its performance.

In DOE, the factors considered include: dwell point policy, the vehicle-lift combination, scheduling rule, I/O locations and interleaving rule. Besides, three different responses, storage and retrieval transactions' average cycle time, average utilizations of vehicles and lifts, are considered. However, because the ANOVA assumptions are not met for the average cycle time response, an inverse transformation method is applied on this response. The statistical results are analyzed in MINITAB at a 95% confidence level. After determining the main and the interaction effects, a Tukey test analysis is completed on the responses to determine the best levels of the factors. In the following sections, we give the definition of DOE, simulation modeling procedure and the implementation of DOE on the AVS/RS.

5.1 Design of Experiments

DOE is a statistical technique that investigates which and/or how factors affect the response (output), significantly. It makes purposeful changes to the independent (input) variables and evaluates the effects on the dependent (output) variables. By DOE, the performance, reliability and robustness of a system can be improved. Besides, it provides evaluation of various design alternatives to increase the productivity of a system (Montgomery, 1996).

In this section, our objective is not only to investigate how the performance measures of AVS/RS are affected by the pre-defined factors but also to ascertain how they can be improved by adjusting these factors. The performance measures considered are the average cycle time per transaction and the lift and vehicle average utilizations. Cycle time is the time between when a request originates until it is completed. Along with resource utilizations, it is a comprehensive performance measure and employed extensively in many industries. AVS/RS cycle times are determined by the storage rack configuration, vehicle movement kinematics, storage policy and vehicle dwell point policy. In most AS/RSs, the S/R device resides at the point-of-service-completion (POSC). Because the vehicles utilize lifts for vertical movement, it may be advantageous to position the vehicles close the corresponding lift location. In our study, we test the POSC dwell point policy as well as the point-of-lift-location (POLL) dwell point policy.

The factors that are likely to have an effect on the performance measures are: dwell point policy, the vehicle-lift combination, scheduling rule, I/O locations and interleaving rule. Details of the design factors and their levels are explained in Section 5.2. First, we simulate our model using these factors at different levels. Then the responses from the simulation runs are analyzed using DOE to determine the main and interaction effects of the factors.

5.2 Simulation Modeling of the System

The AVS/RS uses lifts to store or retrieve unit loads from the bays located on tiers other than the first, and vehicles for horizontal movement within a tier. In the AVS/RS, we consider the storage area had 42 aisles, 7 tiers. Racks on either side of an aisle consist of 27 bays, and each bay could hold three unit loads (see also Section 4.1). Thus, the warehouse capacity has $27 \times 3 \times 2 \times 42 \times 7$ = 47,628 UL positions. The entire system has 21 vehicles and 7 lifts. The storage area (consisting of 42 aisles) is divided into 7 zones. Each zone has 1 lift, and 3 designated vehicles are assigned to each lift. The lifts are located at the middle of each individual zone, e.g. the first lift is located at the end of the 3rd aisle.

In Figures 5.1 and 5.2 simulation flow charts of storage and retrieval transactions are given, respectively. According to these figures, storage and retrieval transactions arrive into the system at the same rate. Unit loads are randomly assigned to an available (empty) pallet position. The retrieval transactions are assumed to arrive into the system after 20% of the storage capacity is filled. Each zone has its own queue. When a vehicle becomes available in a zone the first transaction waiting in that zone's queue seizes the free vehicle. If more than one vehicle is available, the transaction seizes the free vehicle, randomly. For a storage transaction on a tier other than the first, the AV uses the lift in the corresponding zone. The other assumptions that are used in our simulation model are summarized below:

- 10% of the transactions generated are for the first tier, and the remaining are for the other tiers.
- The input station is at the 39th aisle, and the output station is at the 3rd aisle.
- Arrivals to the system follow a poisson process and the mean arrival rate is the $\lambda_S = \lambda_R = 182$ transactions per hour.
- The storage transactions originate at the input station, and the retrieval transactions end at the output station.
- The time for picking up and depositing the unit loads are assumed 14 sec., each.

- The transfer times of a vehicle to and from a lift is assumed to be negligible.
- The vehicles and lifts have an extra delay because of acceleration and deceleration times which are assumed to be 5 and 3 seconds, respectively.

The simulation model is assumed to be a non-terminating system, allowing us to conduct a steady state analysis. The warm-up is determined as three months. The length of each simulation run is two years, 280,800 min. ($(365 \times 2 \times 24 \times 60)$ + warm up period = 1,180,800 min.). The model is run for 10 independent replications. So, a total of 480 (48*10) response values are used for ANOVA in MINITAB. Simulation runs are conducted based on the five different factors shown in Table 5.1. And the experiments are conducted based on two levels for each factor except the vehicle-lift combination factor which has three levels. As a result, the simulation model is run for a full factorial design having 48 experiments. In the simulation model, the *common random numbers* (CRN) variance reduction technique is used. CRN is a popular and useful variance reduction technique when we compare two or more alternative configurations. It requires synchronization of the random number streams, and uses the same random numbers to simulate all configurations. In CRN, a specific random number used for a specific purpose in one configuration is used for exactly the same purpose in all other configurations. Thus, variance reduction is ensured.

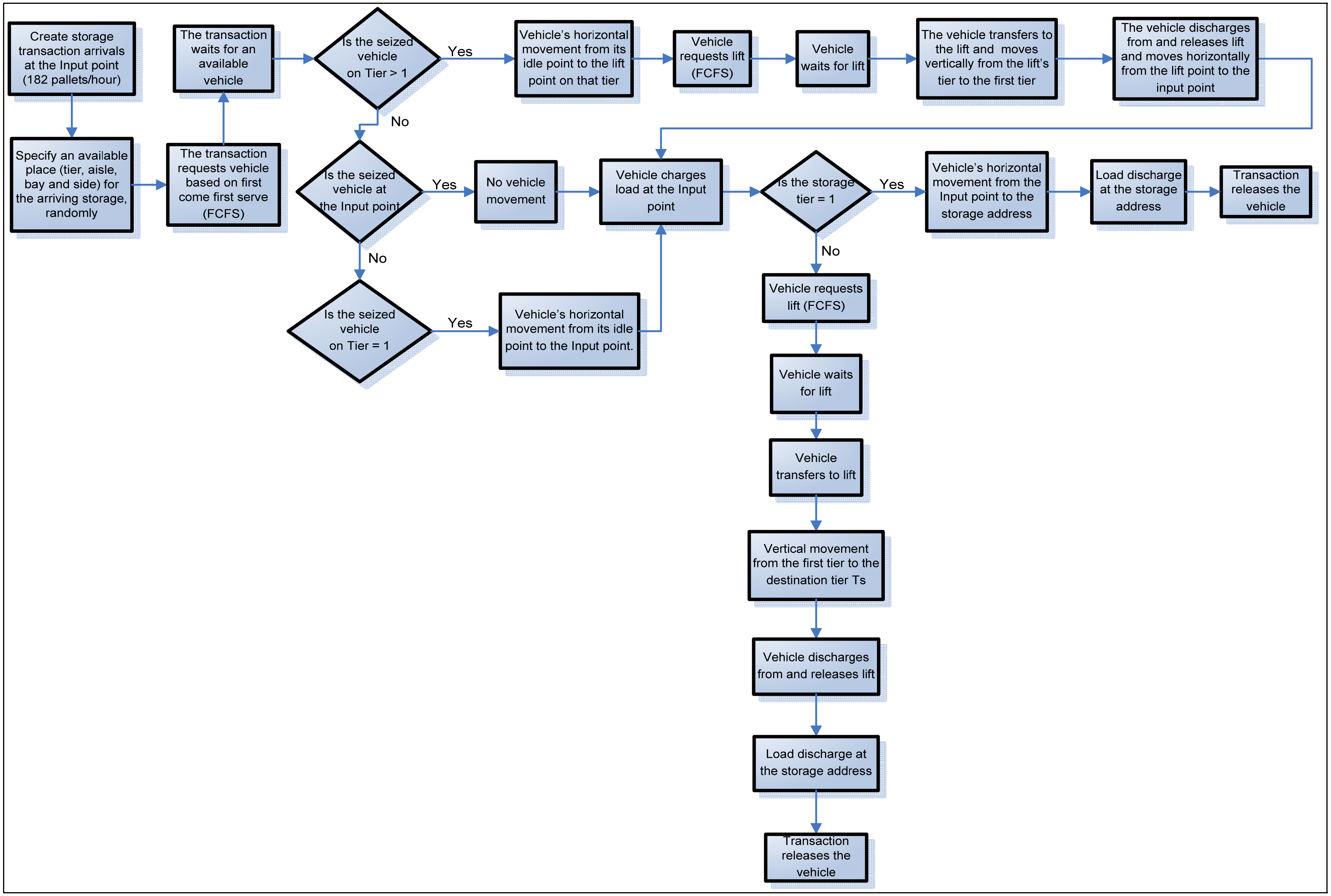

Figure 5.1: Storage transaction's simulation flow chart

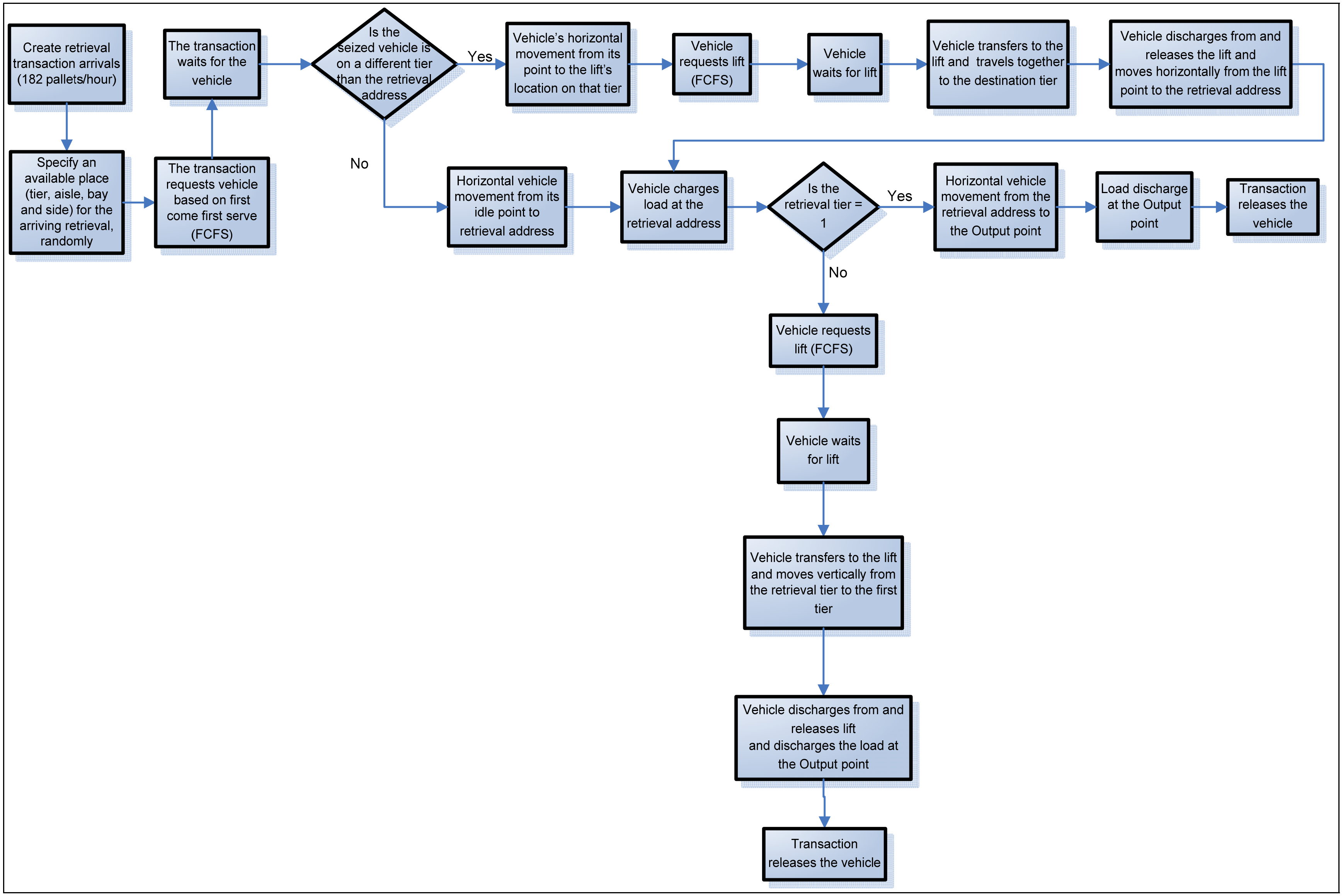

Figure 5.2: Retrieval transaction's simulation flow chart

Table 5.1: Levels of Factors

	Factors	Codes		Levels
1	Dwell Point Policy	-	(Low level)	POSC
		+	(High Level)	POLL
2	Vehicle-lift Combination	-	(Low level)	3*7
		*	(Medium level)	4*6
		+	(High Level)	4*7
3	Scheduling Rule	-	(Low level)	FCFS
		+	(High Level)	SDT
4	I/O Locations	-	(Low level)	Initial
		+	(High Level)	Middle
5	Interleaving Rule	-	(Low level)	No rule
		+	(High Level)	Opportunistic int.

5.2.1 Design factors

The five design factors considered in our experiments are detailed in the following subsections.

5.2.1.1 Dwell point policy

The position where an idle vehicle waits after completing a transaction is dictated by the dwell-point policy. A dwell-point is chosen to minimize the expected time to travel to the next request. In the current system, because retrieval transactions end at the output station, the dwell point policy is considered only for the storage transactions. The point-of-service-completion (POSC) dwell point policy is used in many warehouses. Under this type of policy, a vehicle waits at the destination of the last transaction, until it is seized by the next S/R transaction. Thus, the I/O location becomes the vehicle dwell point for cycles concluding with retrieval transactions and the load storage address becomes the dwell point for cycles concluding with storage transactions. In addition to the POSC policy another dwell point policy, POLL, is also used. In the latter policy, a vehicle waits in front of its designated lift until it is seized by the next S/R transaction. The POLL rule may be effective when the vehicle utilizations are low in the system because, we know that 90% of the transactions generated are on a tier other than the first.

5.2.1.2 The vehicle-lift combination

In the current system, the storage area is divided into 7 zones. Each zone has 1 lift and 3 vehicles. The transaction cycle time includes horizontal travel time to the lift, the time a vehicle waits for the lift, and the vertical movement time of the vehicle on the lift. This would be followed by vehicle travel time to its destination. The number of lifts and vehicles is thus likely to have an impact on cycle times and utilization. As a result, the vehicle-lift combination factor is considered to be a three-level factor in the design. It should be noted that the vehicle-lift combination factor is the combination of “number of vehicles per lift” and “number of lifts” whose multiplication gives the total number of vehicles in the entire system.

We do not consider the number lifts and number of vehicles as separate factors because some combinations of number of lifts and vehicles might not be necessary to experiment (e.g. 6 lifts and 3 vehicles per lift, blocks the system when we consider all combinations of the design). Consequently, three levels whose low (‘‘-”) level is the current 7 lifts with 3 vehicles per lift (zone), the medium (“*”) level is 6 lifts with 4 vehicles per lift, and the high level (‘‘+”) is 7 lifts with 4 vehicles per lift, are considered.

5.2.1.3 Scheduling rule

Storage requests in distribution or production environments are usually not time-critical. The exact point in time at which loads are stored is not of much importance for the performance of the system. Therefore, storage requests are usually stored according to the FCFS principle. However, because retrieval requests are time-sensitive, sequencing becomes important. By sequencing the storage and retrievals efficiently, the overall cycle time can be improved. Besides, studies show that the scheduling rules have an impact on the AS/RSs (Randhawa and Shroff, 1995; Bozer and White, 1984; Roodbergen and Iris, 2009). We consider FCFS and shortest destination time (SDT) rules in our study.

5.2.1.4 I/O location

An output location is where retrieved loads are dropped off, and an input location is where incoming loads are picked up for storage. I/O locations can be co-located or they can be separated. Unit loads arrive at the I/O location of an AVS/RS from other parts of the warehouse via automated guided vehicles, conveyors, or forklift trucks. The unit loads are stored in the AVS/RS by AVs and after a period of time they are retrieved again, for example, to be shipped to a customer. Because, loads are picked up and dropped off at I/O location by the AVs, the I/O location impacts the cycle times. We consider two levels for I/O location. In low (‘‘-”) level, the input location is near the first lift’s location, and the output point is near the last lift’s location. For the high (‘‘+”) level, the I/O location is near the middle aisle.

5.2.1.5 Interleaving rule

A unit-load AVS/RS can operate in two ways, namely in a single command (SC) cycle or in a dual command (DC) cycle (Bozer and White, 1984). In a SC cycle, the vehicle performs either a single storage or a single retrieval transaction. The storage cycle time then is equal to the sum of the times to pick-up a load at the input station, to travel to the storage location, and to place the load in the rack. The retrieval cycle time can be defined as the sum of the times to travel to the storage location, to pick-up a load from this location, and to return to the output station. If an AVS/RS performs both a storage and a retrieval transaction in a single cycle, we call this a DC cycle. In this case, the cycle time is defined as the sum of the times to pick-up the load, to travel to the storage location and store the load, the empty travel time (interleaving time) from the storage location to the retrieval location and the time to pick the UL and transport it to the output station. Clearly, the total time to perform all storage and retrieval transactions reduces if dual commands are performed.

When, interleaving is performed on an opportunistic basis, then storage or retrieval transaction is combined with retrieval or storage transaction at the start of the S/R cycle when one or more of both types are pending in the active queue. However, as more DC cycles are executed, queue sizes are reduced and SC cycles become more likely (Malmborg, 2003). In the current simulation model, because the arrival rates are equal for storage and retrieval transactions, we expect more DC cycles, and hence decreased average cycle times in our model.

5.2.2 DOE results

Because the AVS/RS under study is quite complex it makes it difficult for a manager to identify the parameters that could affect the average cycle time and utilization in the system, and the interaction effects of these factors. Hence, a carefully designed factorial experiment is undertaken to determine the relative importance of the factors and their interaction. Because we have five different factors at different levels, this leads to a total of 48 experiments as shown in Table 5.2. The ‘‘-”, ‘‘+” and ‘‘*” signs indicate the level assigned to these factors and can be obtained from Table 5.1. The output performance measures, namely the average cycle times for storage and retrieval transactions and the average utilizations of vehicles and lifts are given in Table 5.2.

To run the simulation model for each experiment, first, four main models are developed based on different dwell point policies and I/O locations, because they have basic structural differences. Then, the SDT scheduling rule and the vehicle-lift combination are integrated into the model via simple logical changes.

To analyze the effects of the factors obtained from the above experiments properly, they should be categorized into their main effects and the interactions among them. The following subsections explain the main and the interaction effects affecting the output.

Table 5.2: Design matrix and responses for all factor combinations

Exp.	Dwell Point Policy	Vehicle-lift comb.	Scheduling Rule	I/O Place	Interleaving Rule	Vehicle Avg. Utiliz.	Lift Avg. Utiliz.	Avg. Cycle Time (min.)
1	−	−	−	−	−	0.96	0.77	17.12
2	+	−	−	−	−	0.95	0.76	17.19
3	−	*	−	−	−	0.87	0.82	6.43
4	+	*	−	−	−	0.86	0.80	6.36
5	−	+	−	−	−	0.76	0.76	4.92
6	+	+	−	−	−	0.74	0.73	4.76
7	−	−	+	−	−	0.95	0.77	12.94
8	+	−	+	−	−	0.95	0.76	12.98
9	−	*	+	−	−	0.86	0.82	5.80
10	+	*	+	−	−	0.85	0.80	5.73
11	−	+	+	−	−	0.76	0.76	4.71
12	+	+	+	−	−	0.74	0.73	4.55
13	−	−	−	+	−	0.78	0.71	4.67
14	+	−	−	+	−	0.75	0.67	4.55
15	−	*	−	+	−	0.73	0.78	3.91
16	+	*	−	+	−	0.70	0.73	3.74
17	−	+	−	+	−	0.62	0.71	3.42
18	+	+	−	+	−	0.59	0.64	3.17
19	−	−	+	+	−	0.77	0.71	4.39
20	+	−	+	+	−	0.75	0.67	4.26
21	−	*	+	+	−	0.72	0.78	3.78
22	+	*	+	+	−	0.70	0.73	3.61
23	−	+	+	+	−	0.62	0.71	3.37
24	+	+	+	+	−	0.58	0.64	3.12
25	−	−	−	−	+	0.94	0.73	11.23

26	+	−	−	−	+	0.93	0.72	11.10
27	−	*	−	−	+	0.85	0.79	5.74
28	+	*	−	−	+	0.84	0.77	5.63
29	−	+	−	−	+	0.75	0.74	4.69
30	+	+	−	−	+	0.73	0.70	4.51
31	−	−	+	−	+	0.94	0.73	11.30
32	+	−	+	−	+	0.94	0.72	11.21
33	−	*	+	−	+	0.85	0.80	5.67
34	+	*	+	−	+	0.84	0.77	5.57
35	−	+	+	−	+	0.75	0.74	4.64
36	+	+	+	−	+	0.73	0.70	4.47
37	−	−	−	+	+	0.76	0.69	4.01
38	+	−	−	+	+	0.73	0.65	3.83
39	−	*	−	+	+	0.70	0.76	3.60
40	+	*	−	+	+	0.67	0.71	3.40
41	−	+	−	+	+	0.60	0.68	3.24
42	+	+	−	+	+	0.56	0.62	2.98
43	−	−	+	+	+	0.76	0.70	4.01
44	+	−	+	+	+	0.73	0.65	3.84
45	−	*	+	+	+	0.70	0.76	3.59
46	+	*	+	+	+	0.67	0.71	3.39
47	−	+	+	+	+	0.60	0.68	3.23
48	+	+	+	+	+	0.56	0.62	2.97

The factorial ANOVA is completed in MINITAB statistical software at 95% confidence level. To be able to interpret the ANOVA results, the model adequacy should be met. The model adequacy is that, residuals should be normally distributed, residuals should have a mean of zero and residuals should have constant variance. If one or more of these assumptions are not met, one of a suitable transformation such as, inverse, log, ln, square root, inverse, inverse square root, etc. should be applied on the response to achieve model adequacy. In the current model, because the average cycle time response ANOVA residuals are not normally distributed the inverse transformation is applied on this response. Table 5.3 presents the ANOVA results for the inverse transformation. Tables 5.4 and 5.5 illustrate ANOVA results for the average utilization of vehicles and lifts, respectively. Because the residual plots of these ANOVA results are normally distributed for these responses, we interpret our results based on tables 5.3, 5.4 and 5.5.

Table 5.3: ANOVA results for inverse average cycle time

Source	DF	Seq SS	Adj SS	Adj MS	*F*	*P*
Dwell Point	1	0.011252	0.011252	0.011252	31423.57	0.000
Vehicle-lift combin.	2	0.907064	0.907064	0.453532	1266551.30	0.000
Scheduling	1	0.005094	0.005094	0.005094	14226.39	0.000
I/O Loc.	1	1.786474	1.786474	1.786474	4988977.20	0.000
Interleaving	1	0.039197	0.039197	0.039197	109462.83	0.000
Dwell Point*Vehicle-lift combin.	2	0.002854	0.002854	0.001427	3985.37	0.000
Dwell Point*Scheduling	1	0.000002	0.000002	0.000002	6.18	0.013
Dwell Point*I/O Loc.	1	0.004586	0.004586	0.004586	12808.33	0.000
Dwell Point*Interleaving	1	0.000260	0.000260	0.000260	725.72	0.000
Vehicle-lift combin.*Scheduling	2	0.000335	0.000335	0.000167	467.13	0.000
Vehicle-lift combin.*I/O Loc.	2	0.092529	0.092529	0.046265	129200.51	0.000
Vehicle-lift combin.*Interleaving	2	0.004349	0.004349	0.002174	6072.31	0.000
Scheduling*I/O Loc.	1	0.000323	0.000323	0.000323	901.79	0.000
Scheduling*Interleaving	1	0.004057	0.004057	0.004057	11331.03	0.000
I/O Loc.*Interleaving	1	0.002404	0.002404	0.002404	6713.42	0.000
Dwell Point*Vehicle-lift combin. *Schdling	2	0.000000	0.000000	0.000000	0.44	0.645
Dwell Point*Vehicle-lift combin. *I/OLoc.	2	0.000386	0.000386	0.000193	539.41	0.000
Dwell Point*Vehicle-lift combin. *Intrlving	2	0.000001	0.000001	0.000001	1.50	0.225
Dwell Point*Schdlng*I/OLoc.	1	0.000001	0.000001	0.000001	3.35	0.068
Dwell Point*Schdlng*Intrlving	1	0.000005	0.000005	0.000005	14.40	0.000
Dwell Point*I/OLoc.*Intrlving	1	0.000078	0.000078	0.000078	218.49	0.000
Vehicle-lift combin.*Schdling *I/OLoc.	2	0.000034	0.000034	0.000017	47.12	0.000
Vehicle-lift combin.*Schdlng *Intrlving	2	0.000693	0.000693	0.000346	967.43	0.000
Vehicle-lift combin.*I/OLoc. *Intrlving	2	0.000005	0.000005	0.000002	6.95	0.001
Scheduling*I/O Loc.*Intrlvng	1	0.000160	0.000160	0.000160	447.79	0.000
Dwell Point*Vehicle-lift combin. *Schdling*I/OLoc.	2	0.000001	0.000001	0.000000	1.15	0.319
Dwell Point*Vehicle-lift combin. *Schdling*Interleaving	2	0.000000	0.000000	0.000000	0.02	0.979
Dwell Point*Vehicle-lift combin. *I/O Loc.*Interleaving	2	0.000001	0.000001	0.000001	1.91	0.149

Dwell Point*Scheduling*I/OLoc. *Interleaving	1	0.000001	0.000001	0.000001	1.54	0.216
Vehicle-lift combin.*Scheduling *I/OLoc.*Interleaving	2	0.000011	0.000011	0.000005	14.69	0.000
Dwell Point*Vehicle-lift combin. *Schdling*I/OLoc.*Interleaving	2	0.000000	0.000000	0.000000	0.67	0.514

Table 5.4: ANOVA results for average utilization of vehicles

Source	DF	Seq SS	Adj SS	Adj MS	*F*	*P*
Dwell Point	1	0.05610	0.05610	0.05610	537158.25	0.000
Vehicle-lift combin.	2	2.58137	2.58137	1.29069	12357499.90	0.000
Scheduling	1	0.00007	0.00007	0.00007	636.81	0.000
I/O Loc.	1	3.34938	3.34938	3.34938	32068125.76	0.000
Interleaving	1	0.03452	0.03452	0.03452	330540.25	0.000
Dwell Point* Vehicle-lift combin.	2	0.00410	0.00410	0.00205	19617.84	0.000
Dwell Point*Scheduling	1	0.00000	0.00000	0.00000	9.49	0.002
Dwell Point*I/O Loc.	1	0.01006	0.01006	0.01006	96325.71	0.000
Dwell Point*Interleaving	1	0.00018	0.00018	0.00018	1676.69	0.000
Vehicle-lift combin.*Scheduling	2	0.00000	0.00000	0.00000	1.21	0.300
Vehicle-lift combin.*I/O Loc.	2	0.03675	0.03675	0.01838	175937.41	0.000
Vehicle-lift combin.*Interleaving	2	0.00004	0.00004	0.00002	208.25	0.000
Scheduling*I/O Loc.	1	0.00001	0.00001	0.00001	74.54	0.000
Scheduling*Interleaving	1	0.00017	0.00017	0.00017	1608.54	0.000
I/O Loc.*Interleaving	1	0.00196	0.00196	0.00196	18726.70	0.000
Dwell Point*Vehicle-lift combin. *Schdling	2	0.00000	0.00000	0.00000	0.10	0.905
Dwell Point*Vehicle-lift combin. *I/OLoc.	2	0.00004	0.00004	0.00002	190.68	0.000
Dwell Point*Vehicle-lift combin. *Intrlving	2	0.00001	0.00001	0.00000	24.22	0.000
Dwell Point*Schdlng*I/OLoc.	1	0.00000	0.00000	0.00000	2.28	0.132
Dwell Point*Schdlng*Intrlving	1	0.00000	0.00000	0.00000	12.74	0.000
Dwell Point*I/OLoc.*Intrlving	1	0.00001	0.00001	0.00001	121.85	0.000
Vehicle-lift combin.*Schdlng *I/OLoc.	2	0.00000	0.00000	0.00000	0.19	0.826
Vehicle-lift combin.*Schdlng *Intrlving	2	0.00004	0.00004	0.00002	194.21	0.000
Vehicle-lift combin.*I/OLoc. *Intrlving	2	0.00026	0.00026	0.00013	1221.17	0.000
Scheduling*I/O Loc.*Intrlvng	1	0.00000	0.00000	0.00000	16.64	0.000
Dwell Point*Vehicle-lift combin. *Schdling*I/OLoc.	2	0.00000	0.00000	0.00000	0.40	0.669
Dwell Point*Vehicle-lift combin. *Schdling*Interleaving	2	0.00000	0.00000	0.00000	0.05	0.949
Dwell Point*Vehicle-lift combin. *I/O Loc.*Interleaving	2	0.00000	0.00000	0.00000	5.87	0.003
Dwell Point*Scheduling*I/OLoc. *Interleaving	1	0.00000	0.00000	0.00000	0.02	0.889
Vehicle-lift combin.*Scheduling *I/OLoc.*Interleaving	2	0.00000	0.00000	0.00000	9.01	0.000
Dwell Point*Vehicle-lift combin. *Schdling*I/OLoc.*Interleaving	2	0.00000	0.00000	0.00000	0.00	1.000

Table 5.5: ANOVA results for average utilization of lifts

Source	DF	Seq SS	Adj SS	Adj MS	*F*	*P*
Dwell Point	1	0.167218	0.167218	0.167218	2835675.35	0.000
Vehicle-lift combin.	2	0.474139	0.474139	0.237069	4020218.55	0.000
Scheduling	1	0.000787	0.000787	0.000787	13348.51	0.000
I/O Loc.	1	0.456369	0.456369	0.456369	7739089.63	0.000
Interleaving	1	0.088872	0.088872	0.088872	1507087.15	0.000
Dwell Point*Vehicle-lift combin.	2	0.014269	0.014269	0.007135	120990.91	0.000
Dwell Point*Scheduling	1	0.000008	0.000008	0.000008	127.99	0.000
Dwell Point*I/O Loc.	1	0.023841	0.023841	0.023841	404297.30	0.000
Dwell Point*Interleaving	1	0.000632	0.000632	0.000632	10715.89	0.000
Vehicle-lift combin.*Scheduling	2	0.000318	0.000318	0.000159	2694.34	0.000
Vehicle-lift combin.*I/O Loc.	2	0.004911	0.004911	0.002456	41644.34	0.000
Vehicle-lift combin.*Interleaving	2	0.000406	0.000406	0.000203	3439.27	0.000
Scheduling*I/O Loc.	1	0.000117	0.000117	0.000117	1977.16	0.000
Scheduling*Interleaving	1	0.000003	0.000003	0.000003	50.25	0.000
I/O Loc.*Interleaving	1	0.003132	0.003132	0.003132	53103.94	0.000
Dwell Point*Vehicle-lift combin. *Schdling	2	0.000000	0.000000	0.000000	0.25	0.777
Dwell Point*Vehicle-lift combin. *I/OLoc.	2	0.000315	0.000315	0.000158	2671.39	0.000
Dwell Point*Vehicle-lift combin. *Intrlving	2	0.000037	0.000037	0.000019	315.23	0.000
Dwell Point*Schdlng*I/OLoc.	1	0.000001	0.000001	0.000001	12.23	0.001
Dwell Point*Schdlng*Intrlving	1	0.000004	0.000004	0.000004	60.67	0.000
Dwell Point*I/OLoc.*Intrlving	1	0.000002	0.000002	0.000002	41.38	0.000
Vehicle-lift combin.*Schdlng *I/OLoc.	2	0.000052	0.000052	0.000026	439.18	0.000
Vehicle-lift combin.*Schdlng *Intrlving	2	0.000002	0.000002	0.000001	19.31	0.000
Vehicle-lift combin.*I/OLoc. *Intrlving	2	0.002139	0.002139	0.001069	18132.69	0.000
Scheduling*I/O Loc.*Intrlvng	1	0.000001	0.000001	0.000001	10.03	0.002
Dwell Point*Vehicle-lift combin. *Schdling*I/OLoc.	2	0.000000	0.000000	0.000000	3.95	0.020
Dwell Point*Vehicle-lift combin. *Schdling*Interleaving	2	0.000000	0.000000	0.000000	0.33	0.720
Dwell Point*Vehicle-lift combin. *I/O Loc.*Interleaving	2	0.000002	0.000002	0.000001	18.05	0.000
Dwell Point*Scheduling*I/OLoc. *Interleaving	1	0.000000	0.000000	0.000000	0.18	0.672
Vehicle-lift combin.*Scheduling *I/OLoc.*Interleaving	2	0.000001	0.000001	0.000001	11.16	0.000
Dwell Point* Vehicle-lift combin. *Schdling*I/OLoc.*Interleaving	2	0.000000	0.000000	0.000000	0.13	0.876

5.2.2.1 Main effects of the factors

The main effect of a factor is the average change in the output due to the factor shifting from its ‘‘-” level to its ‘‘+” or ‘‘*” level, while holding all other factors constant. A factorial ANOVA shows whether there are significant main effects of the independent variables and whether there are significant interaction effects between independent variables in a set of data or not (Montgomery, 1996).

In Tables 5.3, 5.4 and 5.5 last columns (labeled *P*) indicate whether or not the factor affects the performance measure. The *P* values smaller than 0.05 mean that these factors are important for that performance measure. As a result, according to the Tables 5.3, 5.4 and 5.5 all main effects are significant on the performance measure (all *P* values are smaller than 0.05). However, it should be noted that in Table 5.3 all factors are important for the inverse transformation, not the average cycle time. In Table 5.3, the most important factor affecting the inverse of the average cycle time is the I/O locations, because it has the biggest *F* value. Then, the vehicle-lift combination, interleaving rule, dwell point policy and scheduling rule follow. From tables 5.4 and 5.5, we see that the order of the factors affecting the utilizations are the same which as those affecting the inverse of the average cycle time.

The level of effects of factors can be traced from Table 5.2, as well. For instance, when I/O locations change to its high level (from 13rd experiment to 24th experiment) the decrease in the performance measures can be seen when pair-wise comparison is done between the 1st experiment and 13rd experiment, 2nd and 14th, etc. And the same pair-wise comparison can be done between 25th experiment and 37th experiment, 26th and 38th etc.

The dwell point policy factor effect can also be seen in Table 5.2 by comparing the 1st experiment with the 2nd and the 3rd experiment with the 4th, etc. There is not much difference in the average cycle times in the first 12 experiments, but this difference increases after that. This is most likely due to the interaction effect of the dwell point policy factor with the other factors.

5.2.2.2 Two-way interactions

When two factors interact, the effect on the response variable of one explanatory variable depends on the specific value or level of the other explanatory variable. Therefore, the presence of a main effect might not necessarily be useful and indicative when an interaction effect exists. As seen in Tables 5.3, 5.4 and 5.5 all two-way interaction effects are significant ($P < 0.05$) at 95% confidence level except the vehicle-lift combination*scheduling rule interaction effect for the utilization of vehicles performance measure ($P > 0.05$).

For instance, the effect of the dwell point policy and I/O locations two-way interactions can be traced in Table 5.2. In the first 12 experiments the change on dwell point policy factor does not cause a significant change on the performance measures. However, when the I/O locations factor's level changes to its high level after the 12th experiment, a relatively significant change occurs. Therefore, the existence of dwell point policy and I/O locations factors interaction effect can be expected. The same result can also be observed between experiments 25-36 and 37-48.

5.2.2.3 Three-way interactions

From Table 5.3, we see that all three-way interactions are significant on the inverse response except these dwell point policy*the vehicle-lift combination*scheduling rule, dwell point policy*the vehicle-lift combination*interleaving rule, dwell point policy*scheduling*I/O locations three-way interaction effects.

In Table 5.4, all three-way interaction effects are significant except the dwell point policy*the vehicle-lift combination*scheduling rule, dwell point policy*scheduling*I/O locations, the vehicle-lift combination*scheduling rule*I/O locations.

From Table 5.5, it is seen that all three-way interaction effects are significant except that dwell point policy* the vehicle-lift combination*scheduling rule.

5.2.2.4 Four-way interactions

For the inverse of the average cycle time, except the vehicle-lift combination*scheduling rule*I/O locations*interleaving rule four-way interaction, none of the other four-way interaction effects are significant.

For the average utilization of vehicles performance measure, none of the four-way interaction effects are significant, except the dwell point policy*the vehicle-lift combination*I/O locations*interleaving rule and the vehicle-lift combination*scheduling rule*I/O locations*interleaving rule four-way interactions.

For the average utilization of lifts performance measure, none of the four-way interaction effects are significant, except the dwell point policy*the vehicle-lift combination*scheduling rule*I/O locations, dwell point policy*the vehicle-lift combination*I/O locations*interleaving rule and the vehicle-lift combination*scheduling rule* I/O locations*interleaving rule four-way interactions.

5.2.2.5 Five-way interactions

As seen from Tables 5.3, 5.4 and 5.5 none of five-way interaction effects are significant on the responses because the *P* values are greater than 0.05.

5.2.3 Tukey Test

To be able to find out the best levels of the factors which provide the minimum average cycle time and utilizations that are statistically important, Tukey test is employed. Tukey test is a statistical test generally used in conjunction with an ANOVA to find which the mean values that are significantly different from one another. It compares the means of every treatment to the means of every other treatment, and identifies where the difference between two means is greater than the standard error.

To complete the Tukey test and determine the best experiments that give the significantly minimum average cycle times, first the interaction effect graphs that have maximum terms in them are investigated. Figure 5.3 illustrates the interaction plot of four-way effects for inversed average cycle time. Because, the four-way interaction effect occurs only among the vehicle-lift combination*scheduling rule*I/O locations*interleaving rule factors, interaction plot of these factors are investigated. This means that the response changes significantly at particular levels of these factors. The dwell point policy has a main effect on the response, so we should investigate this factor individually. Figure 6 shows the main effect of the dwell point policy factor.

Because we are interested in the inverse average cycle time, we must find the maximum response level to minimize the average cycle time. From Figure 5.3, we observe that the maximum response is obtained at high levels of each factor. The second maximum response is obtained at the low level of the scheduling rule factor and the high level of the remaining factors. And the third maximum response is obtained at the low level of the interleaving rule factor and the high level of the remaining factors. Besides, in Figure 5.4 it can be seen that the maximum response is obtained at the high level of the dwell point policy.

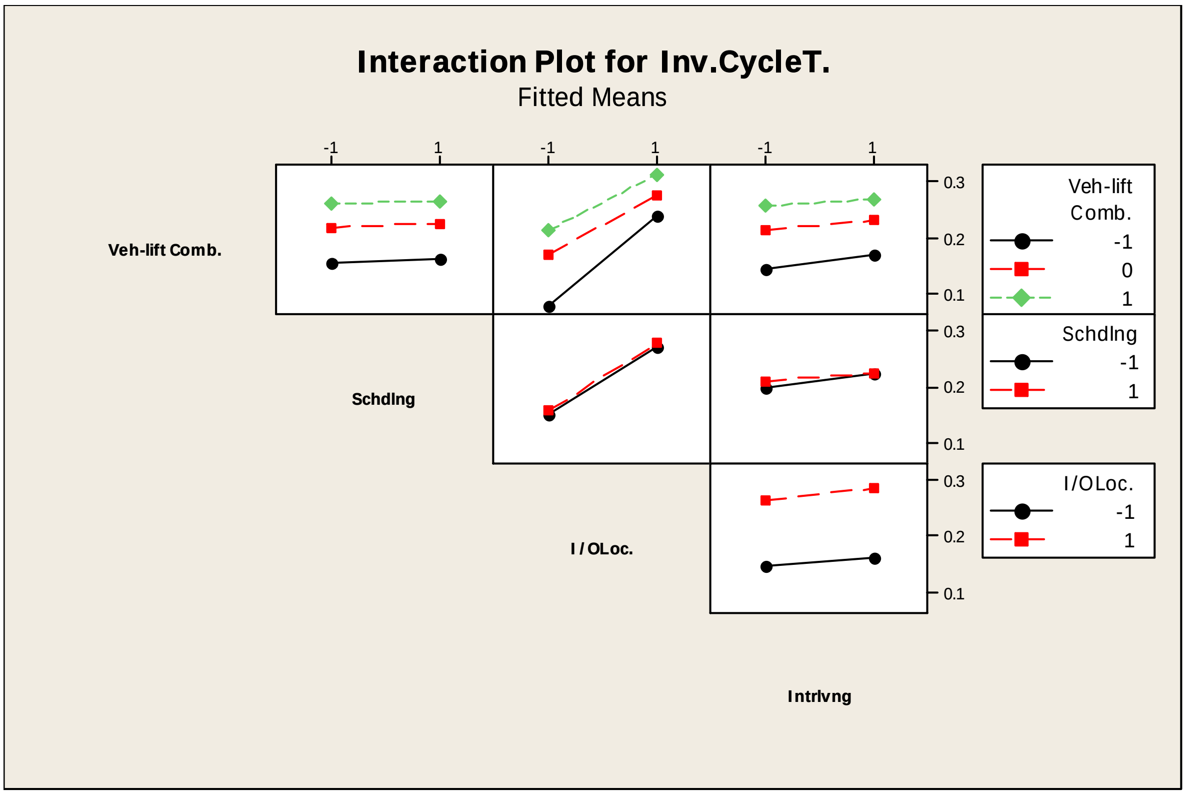

Figure 5.3: Interaction plot graph for inverse average cycle time

Consequently, we sort the responses in descending order to do pair-wise comparison by Tukey test to decide which experiment(s) is/are significantly important. The sorted order becomes 48th, 42nd and 24th for the first three best experiments. We compare the first experiment with the following experiments in Tukey test. Table 5.6 represents a part of the Tukey test results which shows the pair wise comparison of 42nd experiment with the 48th experiment. It is found out that the 48th experiment is not significantly different from the 42nd experiment ($P > 0.05$).

Therefore, we compare the 48th experiment with the 24th experiment and find that there is significant difference between these two experiments. So, there is no need to compare the other experiments with the 48th experiment. As a result, the experiments 48 and 42 give the statistically significant results for the performance measure, average cycle time.

The same analysis is also completed for the average utilizations of vehicles and lifts performance measures by the Tukey test. This time we sort the mean utilization values in ascending order, because now we investigate the factor levels that give the minimum utilization values. Consequently, for the average utilization of vehicles performance measure, it is found out that the best results for the minimum utilization are obtained by the 42nd and 48th experiments which are not significantly different from each other. However, for the average utilizations of lifts performance measure, it is found out that the minimum utilization is obtained by only the 48th experiment.

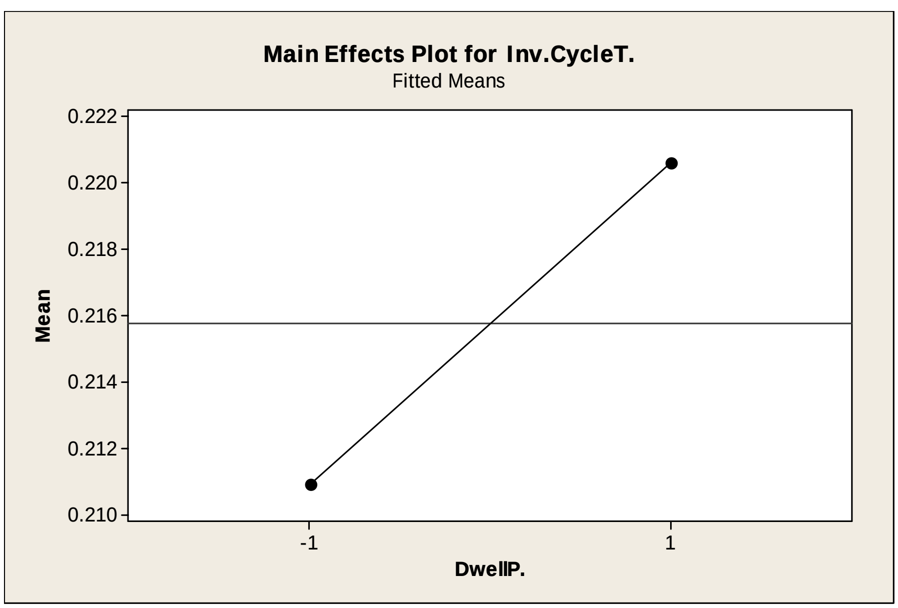

Figure 5.4: Main effect of dwell point policy for inversed average cycle time

Table 5.6: Tukey test comparison result

vehicle-lift combination = 1
Schdlng = -1
I/OLoc. = 1
Intrlvng = 1 subtracted from:

LiftN-Veh.N.	Schdlng	I/OLoc.	Intrlvng	Difference of Means	SE of Difference	T-Value	Adjusted P Value
1	1	-1	-1	-0.1062	0.000189	-561.4	0.0001
1	1	-1	1	-0.1026	0.000189	-542.2	0.0001
1	1	1	-1	-0.0137	0.000189	-72.3	0.0001
1	1	1	1	0.0006	0.000189	3.1	0.2410

5.3 Summary

In this chapter, simulation based experimental design is completed for an AVS/RS. The factors that could affect the response measures are defined as dwell point policy, the vehicle-lift combination, scheduling rule, I/O locations and interleaving rule. Three different responses –the average cycle time of the storage and retrieval transactions, average utilizations of vehicles and average utilizations of lifts- are investigated. Because the ANOVA assumptions are not met, the inverse transformation is applied on the average cycle time. Consequently, the ANOVA result shows that all main and two main effects are significant at 95% confidence level except the vehicle-lift combination*scheduling rule interaction effect for the average utilization of vehicles performance measure.

Tukey test analysis is completed at 95% confidence level to find the best experiment that gives the minimum average cycle time. The 48th experiment is found out to be the best combination for all three performance measures. Besides, the 42nd experiment also gives statistically significant result for average cycle times and average utilization of vehicles.

In conclusion, if the manager of the company wants to minimize the average cycle time and utilization of vehicles, he or she can use either the 48th or the 42nd experimental combinations for the AVS/RS. However, if he or she wants to minimize the average utilization of lifts then 48th experimental combination should be used.

CHAPTER 6
MATRIX GEOMETRIC METHOD

After completing the candidacy exam, we will model the AVS/RS using the MGM approach and compare its performance with those in Chapter 4 and simulation. We introduce MGM in this chapter.

As mentioned previously, the MGM was developed by Marcel Neuts in the 1980s. It is a numerical approach to solve Markov processes having a special repetitive property called matrix-geometric property. For systems with large or possibly infinite number of states, exact solutions can only be obtained if one can utilize structural properties of equations. For example, the solution for the infinite set of equations derived from a $M/M/1$ queue is easily determined because these equations have a repetitive structure. This repetition allows us to determine a recursive solution for the stationary state probabilities since it implies that if one knows the stationary probability for any state i, then stationary probability for state i+1 can be determined. The stationary state probability for the repeating portion of this process thus has a *geometric* form.

Neuts (1980) developed a methodology that allows us to exploit repetitive structure described above. If the states of Markov process can be grouped into vectors which possess a certain repetitive structure then a recursive procedure can be used to determine the stationary state probabilities of the i+1'st vector in terms of the probabilities for the i'th vector. The form of the solution for the stationary state probabilities is leading to Neuts' *matrix geometric* form.

There is a wide range of applications of MGM in computer performance modeling. However, this method can also be applied to manufacturing and/or warehouse models leading to Markov models with a repetitive structure that fits within the matrix geometric framework.

To describe one version of the matrix geometric form consider a Markov process with states (i, j) where $i \geq 0$, and j is a vector. Assume that the number of possible values for i is unbounded, and that transitions can cause i to increase by at most 1 unit. If there exists a value $i^* \geq k$ so that for i', $i^* - k \leq i' \leq i^* + 1$, transition rates between states (i^*, j) and (i', j') are identical to transition rates between $(i^* + m, j)$ and $(i' + m, j')$ for all $m \geq 0$, then the process is matrix geometric. Open queuing systems consisting one queue with and infinite capacity satisfy these conditions. Many models can made to be approximately matrix geometric by truncating certain portions of the state space and assuming that subsequent state transitions repeat so as to satisfy the form. Additionally, classical queuing systems can be solved using MGM when service and arrival process are given by phase distribution.

6.1 Markov Process

We consider a simple example in a vector state process in Section 6.1.1 of a Markov process that has a geometric solution. This will be utilized to derive a general formulation vector state processes which leads to a matrix geometric form.

6.1.1 Example

Let us suppose there are two stations at a markov process whose service rates are μ_1 and μ_2. Assume that the states are denoted as (i, j) , $i \geq 0, j = 0, 1, 2$ where i is the number of customers in the queue (not including any receiving service) and j is the current stage of service of the customer in service. We set i to be equal to 0 if there is no customer in the system. The state transition diagram for this system is shown in Figure 6.1. According to this, interarrival time to the empty system is exponential with rate λ' .

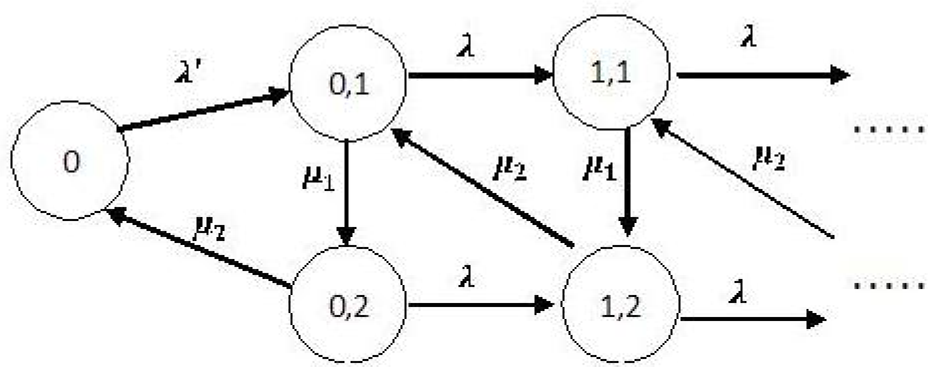

Figure 6.1: The state transition diagram of the example

We order states lexicographically, i.e. (0,0), (0,1), (0,2), (1,1), (1,2), …, and let $\pi_{(i,s)}$ be the stationary probability of state (i, s). We shall say that states at level i are those states defined by (i, 0) and (i, 1). The generator matrix is given by (6.1).

$$Q = \begin{bmatrix} -\lambda' & \lambda' & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . & . & . \\ 0 & -a_1 & \mu_1 & \lambda & 0 & 0 & 0 & 0 & 0 & . & . & . \\ \mu_2 & 0 & -a_2 & 0 & \lambda & 0 & 0 & 0 & 0 & . & . & . \\ 0 & 0 & 0 & -a_1 & \mu_1 & \lambda & 0 & 0 & 0 & . & . & . \\ 0 & \mu_2 & 0 & 0 & -a_2 & 0 & \lambda & 0 & 0 & . & . & . \\ 0 & 0 & 0 & 0 & 0 & -a_1 & \mu_1 & \lambda & 0 & . & . & . \\ 0 & 0 & 0 & \mu_2 & 0 & 0 & -a_2 & 0 & \lambda & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . & . \end{bmatrix} \qquad 6.1$$

where we define $a_i = \lambda + \mu_i,\ s = 1, 2.$

As seen, the matrix has repetitive structure. Let $\pi \equiv (\pi_{(i,1)}, \pi_{(i,2)})$ for $i \geq 1$, $\pi_0 \equiv (\pi_{(0,0)}, \pi_{(0,1)}, \pi_{(0,2)})$ and $\pi \equiv (\pi_0, \pi_1, \pi_2, \ldots)$. We define the following matrices:

$$A_0 = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix},\ A_1 = \begin{bmatrix} -a_1 & \mu_1 \\ 0 & -a_2 \end{bmatrix},\ A_2 = \begin{bmatrix} 0 & 0 \\ \mu_2 & 0 \end{bmatrix} \qquad 6.2$$

$$B_{1,0} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \mu_2 & 0 \end{bmatrix},\ B_{0,1} = \begin{bmatrix} 0 & 0 \\ \lambda & 0 \\ 0 & \lambda \end{bmatrix},\ B_{0,0} = \begin{bmatrix} -\lambda' & \lambda' & 0 \\ 0 & -a_1 & \mu_1 \\ \mu_2 & 0 & -a_2 \end{bmatrix} \qquad 6.3$$

With these definitions we can group the generator matrix into blocks as follows:

$$Q = \begin{array}{c} \\ (0,0) \\ (0,1) \\ (0,2) \\ (1,1) \\ (1,2) \\ (2,1) \\ (2,2) \\ \\ \\ \\ \end{array} \left[\begin{array}{ccc|cc|cc|cc|ccc} (0,0) & (0,1) & (0,2) & (1,1) & (1,2) & (2,1) & (2,2) & & & & & \\ -\lambda' & \lambda' & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . & . & . \\ 0 & -a_1 & \mu_1 & \lambda & 0 & 0 & 0 & 0 & 0 & . & . & . \\ \mu_2 & 0 & -a_2 & 0 & \lambda & 0 & 0 & 0 & 0 & . & . & . \\ \hline 0 & 0 & 0 & -a_1 & \mu_1 & \lambda & 0 & 0 & 0 & . & . & . \\ 0 & \mu_2 & 0 & 0 & -a_2 & 0 & \lambda & 0 & 0 & . & . & . \\ \hline 0 & 0 & 0 & 0 & 0 & -a_1 & \mu_1 & \lambda & 0 & . & . & . \\ 0 & 0 & 0 & \mu_2 & 0 & 0 & -a_2 & 0 & \lambda & . & . & . \\ \hline . & . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . & . \end{array}\right] \qquad 6.4$$

So, (6.4) becomes:

$$Q = \begin{bmatrix} B_{0,0} & B_{0,1} & 0 & 0 & 0 & . & . \\ B_{1,0} & A_1 & A_0 & 0 & 0 & . & . \\ 0 & A_2 & A_1 & A_0 & 0 & . & . \\ 0 & 0 & A_2 & A_1 & A_0 & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \end{bmatrix} \qquad 6.5$$

where a 0 entry in (6.5) is a matrix of all zeros of the appropriate dimension.

We call the repeating portion of the process to be the set of linear equations starting at the second blocked column and the boundary portion to be the equations associated with the 0^{th} and 1^{st} blocked column. We solve for the stationary probabilities, exactly. First we write down the equation for the repeating portion of the process which given in block matrix form is

$$\pi_{j-1}A_0 + \pi_j A_1 + \pi_{j+1}A_2 = 0, \qquad j \geq 2 \qquad 6.6$$

The value of π_j is a function only of the transition rates between states with *j*-1 queued customers and states with *j* queued customers. Since these transition rates do not depend upon the value of *j*, this suggests that there is some constant matrix *R* such that:

$$\pi_j = \pi_{j-1}R, \qquad j \geq 2 \qquad 6.7$$

and that the values of $\pi_j = \pi_1 R^{j-1}$, $j \geq 2$ have a matrix geometric form, i.e.

$$\pi_j = \pi_1 R^{j-1}, \quad j \geq 2 \qquad 6.8$$

Substituting this guess into (6.6) shows that

$$\pi_1 R^{j-2} A_0 + \pi_1 R^{j-1} A_1 + \pi_1 R^j A_2 = 0, \qquad j \geq 2 \tag{6.9}$$

which on simplifying yields

$$A_0 + RA_1 + R^2 A_2 = 0 \tag{6.10}$$

This is a quadratic in the matrix *R* which is typically solved numerically. After obtaining the *R* matrix the steady state probabilities can be calculated in a straight forward manner (Nelson, 1995).

CHAPTER 7
CONCLUSIONS AND FUTURE WORK

The study on SOQNs has been going on for more than two decades. In this thesis, we focus on developing a SOQN approach for a particular material handling system AVS/RS. We assume that the AVs are pallets and transactions are customers in the system. Because of the limited number of vehicles in the system the AVS/RS can be effectively modeled as an SOQN. Moreover, we consider general service time stations in the system having small scv.

In the study, first, we model a particular AVS/RS, using an approximate analytical technique given in Section 4.5. Because the obtained network has load-dependent stations and the existing techniques are for load-independent general network, first we extend Marie's (1980) approximation to load-dependent general networks (see Chapter 3). Second, we start the modeling, by describing all possible scenarios and their probabilities to derive the general service times of the system. Third, we combine all the service times to treat the system as a single-class network. Fourth, we model the AVS/RS as SOQN using Avi-Itzhak and Heyman (1973)'s approximate method and the extended approximation developed in Chapter 3, to obtain the performance measures. Four different performance measures for the AVS/RS system are observed both from the analytical and simulation models. These are, the external queue length (*Leq*), average number of transactions (vehicles) in the network including waiting for service (*Lv*), average number of vehicles in the vehicle pool (*Lp*) and average waiting time in the external queue (*Weq*). The results show that the algorithm works better under heavy traffic conditions than low traffic conditions. We can estimate the key performance measures - the *Ln* and *Lp* values - within 5% the simulation estimates under any condition.

In Chapter 5, we model the AVS/RS using simulation based DOE. In this approach our aim is to identify pre-defined factors affecting performance of AVS/RS, significantly. We consider five factors which are dwell point policy, the vehicle-lift combination, scheduling rule, I/O locations and interleaving rule; and, three different responses which are storage and retrieval transactions' average cycle time, average utilizations of vehicles and lifts. However, because the ANOVA assumptions are not met for the average cycle time response, an inverse transformation method is applied on this response. The statistical results are analyzed in MINITAB. And they show that all factors have significant effects on the responses at a 95% confidence level. In addition, all two-way interaction effects are also significant except the vehicle-lift combination and scheduling rule interaction effects on the average utilization of vehicles. After determining the main and the interaction effects, a Tukey test analysis is completed on the responses to determine the best levels of the factors. The 48th experiment is found out to be the best combination for all three performance measures. Besides, the 42nd experiment also gives statistically significant result for average cycle times and average utilization of vehicles.

In this thesis, the third modeling approach on AVS/RS will be the MGM. After completing the candidacy exam, we will model the AVS/RS using the MGM approach and compare its performance with those in Chapter 4 and simulation. The MGM is introduced in Chapter 6.

REFERENCES

Akyildiz, I.F. & Sieber, A. (1988). Approximate analysis of load dependent general queueing networks. *IEEE Transactions on Software Engineering* 14(11), 1537–1545.

Akyildiz, I.F. (1987). Mean value analysis of closed queueing networks with Erlang service time distributions. *Computing Journal* 39(3), 219-232.

Avi-Itzhak, B., & Heyman, D.P. (1973). Approximate queueing models for multiprogramming computer systems. *Operations Research* 21(1), 1212-1230.

Bard, Y. (1979). Some Extensions to Multiclass Queueing Network Analysis. *Proceedings of the Third International Symposium on Modelling and Performance Evaluation of Computer Systems: Performance of Computer Systems* 51-62.

Baskett, F., Chandy, K.M., Muntz R.R., & Palacios-Gomez F. (1975). Open, closed and mixed networks of queues with different classes of customers. *Journal of the ACM* 22(2), 248-260.

Bolch, G., Greiner, S., de Meer H., & Kishor, S.T. (1988). *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*, John Wiley & Sons: New York.

Bozer, Y.A., & White J.A. (1984). Travel time models for automated storage and retrieval systems. *IIE Transactions* 16(4), 329–338.

Bryant, R.M., Krzesinski A.E., Lakshmi, M.S., & Chandy K.M. (1984). The MVA Priority Approximation. *ACM Transactions on Computer Systems* 2(4), 335-359.

Buitenhek, R., van-Houtum, G-J, & Zijm, H. (2000). AMVA-based solution procedures for open queuing networks with population constraints. *Annals of Operations Research* 93, 15-40.

Buitenhek, R. (1998). Performance Evaulation of Dual Resource Manufacturing Systems. *PhD thesis*, Beta Research Institue, University of Twente, Netherlands.

Buzacott, J.A., & Shanthikumar, J.G. (1980). Models for understanding flexible manufacturing systems. *IIE Transactions* 12(4), 339-349.

Buzacott, J.A., & Shanthikumar, J.G. (1993). Stochastic Models of Manufacturing Systems. *Prentice-Hall*: New Jersey.

Buzen, J.P. (1973). Computational algorithms for closed queueing networks with exponential servers. *Communications of the ACM* 16(9) 527-531.

Chandy, K.M., Herzog, U., & Woo, L. (1975). Parametric analysis of queueing network models. *IBM Journal of Research and Development* 19(1), 36-42.

Chandy, K.M., & Martin, J. (1983). A characterization of product form queueing networks. *Journal of the ACM* 30(2), 286-299.

Dallery, Y. (1990). Approximate analysis of general queuing networks with restricted capacity. *Performance Evaulation* 11(3), 209-222.

Ekren, B.Y., & Heragu, S.S. (2009). *Approximate analysis of load dependent general queuing networks with low service time variability*. Working paper, Department of Industrial Engineering, University of Louisville, Louisville, KY, 40292, USA.

Fdida, S., Perros, H.G., & Wilk, A. (1990). Semaphore queues: modeling multilayered window flow controlmechanisms. *IEEE Transactions on Communications* 38(3), 309-317.

Fukunari, M., & Malmborg, C.J. (2009). A network queuing approach for evaluation of performance measures in autonomous vehicle storage and retrieval systems. *European Journal of Operational Research* 193(1), 152-167.

Fukunari, M., Malmborg, C.J. (2008). An efficient cycle time model for autonomous vehicle storage and retrieval systems. *International Journal of Production Research* 46(12) 3167–3184.

Gelenbe, E., & Pujolle, G. (1987). Introduction to Queueing Networks. *John Wiley & Sons Inc*: New York.

Gelenbe, E. (1975). On approximate computer system models. *Journal of the ACM* 22(2), 261-269.

Gordon, W.J., & Newell, G.F. (1967). Closed queueing networks with exponential servers. *Operations Research* 15(2), 254-265.

Granger, J., Krishnamurthy, A., & Robinson, S.M. (2006). Rapid improvement of stochastic networks using two-moment approximations. *Mathematical and computer Modeling* 43(9-10), 1038-1060.

Heragu, S.S. (2008). *Facilities Design*. *CRC Press*, 3rd ed., Clermont, FL.

Heragu, S.S., & Srinivasan M. (2009). *Analysis of Manufacturing Systems via Single-class, Semi-open Queuing Networks*. Working paper 2009, University of Louisville & University of Minnesota, USA.

Jackson, J.R. (1963). Jobshop-like queueing systems. *Management Science* 10(1), 131-142.

Jia, J., & Heragu, S.S. Analysis of Semi-Open Queuing Networks via Analytical Matrix Geometric Method. *Operations Research* (in press).

Kendall, D.G. (1953). Stochastic processes occuring in theory of queues and their analysis by the method of the imbedded Markov chain. *Annals of Mathematical Statistics* 24, 338-354.

Little, J. (1981). A proof for the queueing formula $L = \lambda W$. *Operations Research* 9(3), 383-387.

Sevcik, K.C., & Mitrani, I. (1981). The Distribution of Queuing Network States at Input and Output Instants. *Journal of the ACM* 28(2), 358-371.

Kobayashi, H. (1974). Application of the diffusion approximation to queueing networks Part II: Nonequilibrium distributions and computer modeling. *Journal of the ACM* 21(3), 459-469.

Kobayashi, H. (1974). Application of the diffusion approximation to queueing networks Part I: Equilibrium queue distributions. *Journal of the ACM* 21(2), 316-328.

Kouvatsos, D.D. (1993). Maximum entropy methods for general queueing models. *Performance Evaluation of Computer and Communication Systems* 729, 1611-3349.

Krishnamurthy, A. Suri, R. & Vernon, M.K. (2000). A New Approach for Analyzing Queueing Models of Material Control Strategies in Manufacturing Systems, *Proc. 4th Int. Workshop on Queueing Networks with Finite Capacity (QNETs2000)*, West Yorkshire, U.K.

Krishnamurthy, A., & Suri, R. Performance Analysis of Single Stage Kanban Controlled Production Systems Using Parametric Decomposition. (2006). *Queuing systems* 54(2), 141-162.

Kühn, P.J. (1979). Approximate analysis of general queueing networks by decomposition. *IEEE Transactions on Communications* 27(1), 113-126.

Kuo, P.H., Krishnamurthy, A., & Malmborg, C.J. (2007). Design models for unit load storage and retrieval systems using autonomous vehicle technology and resource conserving storage and dwell point policies. *Applied Mathematical Modelling* 31(10), 2332-2346.

Kuo, P-H., Krishnamurthy, A., & Malmborg, C.J. (2008). Performance modelling of autonomous vehicle storage and retrieval systems using class-based storage policies. *International Journal of Computer Applications in Technology* 31(3/4), 238 – 248.

Kelton, W.D., Sadowski, R.P., & Sturrock, D.T. (2004). Simulation with Arena, *McGrawHill*: New York.

Lavenberg, S.S., & Reiser, M. (1980). Stationary State Probabilities at Arrival Instants for Closed Queueing Networks With Multiple Types of Customers, *Journal of Applied Probability* 17(4), 1048-1061.

Malmborg, C.J. (2002). Conceptualizing tools for autonomous vehicle storage and retrieval systems. *International Journal of Production Research* 40(8), 1807–1822.

Malmborg, C.J. (2003). Interleaving rule dynamics in autonomous vehicle storage and retrieval systems. *International Journal of Production Research* 41(5), 1057–1069.

Malmborg, C.J. (1996). Storage assignment policy tradeoffs. *International Journal of Production Research* 34(2), 363–378.

Marie, R. (1979). An approximate analytical method for general queueing networks. *IEEE Transactions on Software Engineering* SE-5(5), 530-538.

Marie, R. (1980). Calculating equilibrium probabilities for $\lambda(n)/C_k/1/N$ queues. *ACM Sigmetrics Performance Evaluation Review* 9(2), 117-125.

Maritsas, D.G., & Xirokostas, D.A. (1977). The $M/E_k/r$ machine interference model. *European Journal of Operational Research* 1(2), 112-123.

Montgomery, D.C. (1996). *Design and Analysis of Experiments*. New York: John Wiley & Sons.

Nelson, R. (1995). Probability, *Stochastic Processes, And Queueing Theory: The Mathematics of Computer Performance Modeling*. New York: Springer-Verlag.

Neuts, M. (1980). Matrix-Geometric Solutions in Stochastic Modeling. *The Johns Hopkins University Press*.

Noetzel AS. A generalized queueing discipline for product form network solutions. *Journal of the ACM* 1979;26(4); 779-793.

Randhawa, S.U., & Shroff, R. (1995). Simulation-based design evaluation of unit load automated storage/retrieval systems. *Computers & Industrial Engineering* 28(1), 71–79.

Reiser, M., & Lavenberg, S.S. (1980). Mean value analysis of closed queueing networks. *Journal of the ACM* 27(2), 313-323.

Reiser, M. (1979). A queuing network analysis of computer communication networks with flow control. *IEEE Transactions on Communications* COM-27(8), 1199-1209.

Reiser, M. (1981). Mean value analysis and convolution method for queueing dependent servers in closed queueing networks. *Performance Evaluation* 1(1), 7-18.

Roodbergen, K.J., & Iris, FA V. (2009). A survey of literature on automated storage and retrieval systems. *European Journal of Operational Research* 194(2), 343-362.

Sauer, C.H. (1983). Computational algorithms for state dependent queueing networks. *ACM Transactions on Computer Systems* 1(1), 67-92.

Schweitzer, P. (1979). Approximate Analysis of Multi-class Closed Networks of Queues. *Proceedings of the International Conference on Stochastic Control and Optimization*. Amsterdam.

Shum, A., & Buzen, J. (1977). The EPF Technique: A method for obtaining approximate solutions to closed queueing networks with general service times. *In proceeding of 3rd Symp. on Measuring, Modeling and Evaluating Computer Systems* 201-220.

Spirn, J.P. (1979). Queuing networks with random selection for service. *IEEE Transactions on Software Engineering* SE-5(3), 287-289.

Stewart, W.J. (1978). A comparison of numerical techniques in Markov modeling. *Communications of the ACM* 21(2), 144-152.

Whitt, W. (1983). Performance of the Queueing Network Analyzer. *Bell System Technical Journal* 62(9), 2779-2815.

Zhang, L., Krishnamurthy, A., Malmborg, C.J., & Heragu, S.S. (2009). Variance-based approximations of transaction waiting times in autonomous vehicle storage and retrieval systems. *European Journal of Industrial Engineering* 3(2), 146-168.