University of Massachusetts Amherst

ScholarWorks@UMass Amherst

July 2015

Optimal Topological Arrangement of Queues in Closed Finite Queueing Networks

lening wang
University of Massachusetts Amherst

Recommended Citation

OPTIMAL TOPOLOGICAL ARRANGEMENTS OF QUEUES IN CLOSED FINITE QUEUEING NETWORKS

A Thesis Presented

by

LENING WANG

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

May 2015

Mechanical and Industrial Engineering

OPTIMAL TOPOLOGICAL ARRANGEMENTS OF
QUEUES IN CLOSED FINITE QUEUEING NETWORKS

A Thesis Presented

by

LENING WANG

Approved as to style and content by:

_____

J. MacGregor Smith, Chair

_____

Ian R. Grosse, Member

_____

Patrick A. Kelly, Member

_____

Jonathan P. Rothstein, Department Chair
Mechanical and Industrial Engineering

ABSTRACT

OPTIMAL TOPOLOGICAL ARRANGEMENTS OF QUEUES IN CLOSED FINITE QUEUEING NETWORKS

MAY 2015

LENING WANG

M.S.M.E., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor J. MacGregor Smith

Closed queueing networks are widely used in many different kinds of scientific and business applications. Since the demands of saving energy and reducing costs are becoming more and more significant with developing technologies, finding a systematic methodology for getting the best arrangement is very important. In this thesis, design rules are proposed for tandem and various other topologies, to help the designer find the best arrangements which maximize the throughput. Our topological arrangements problem (TAP) can be established as: the system has m-service stations in a network and each one may have different design parameters. To relax the queueing system, the original finite buffer queue is decomposed into a buffer and an infinite buffer server system. Mean Value Analysis (MVA) is used to measure the performance of each topology arrangement. Finally, mixed-integer sequential quadratic programming (MISQP) is used to solve the optimization problem and it is compared with enumeration and a simulation model of Arena (a discrete-event model).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION TO THE PROBLEM

Queueing network models are widely used to analyze manufacturing systems, scheduling systems, service systems and computer network systems. There are many applications for closed queueing networks such as a central server network in computer science which is shown in figure 1.1. A computer with one CPU (Central Processing Unit) and some I/O (Input/Output) devices is presented. New jobs enter the system by different I/O devices, and a single CPU resource is needed to process the jobs. When a job is finished, it returns to the CPU to ask for more resources. The number of jobs in the system is fixed, and it is also called as the 'degree of multiprogramming'. Not just in computer science, but in many other applications, optimizing the critical resources is necessary because the demands of saving energy and reducing costs are becoming more and more significant with developing technologies. This is one of the most important reasons why better performance and optimization tools are needed which play a vital role during design and operations. It would help companies to save cost and resources, and reach the goal in the most economical way. If the best arrangements could be found during the planning stage, it would not only maximize profits but also avoid the extra cost of modification in the future. So the key factor is to develop a systematic methodology to help one reach this goal, not limited in one or two applications, but could be widely used in many different kinds of network applications.
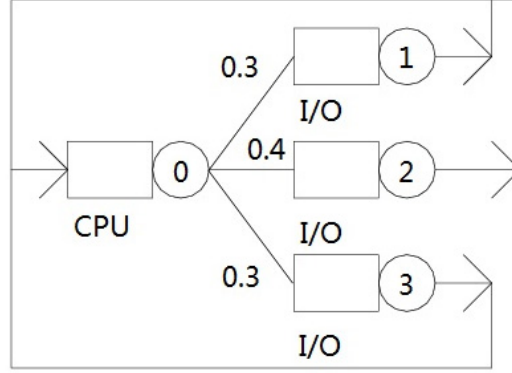
Figure 1.1. Example of open central server network

1.1 Outline of Thesis

In the first chapter, we will present the problem in detail and show the complexity of the problem. The motivation and mathematical models used in our research also will be presented. In the second chapter, the past research carried out before by many others will be shown. In the third chapter, the notations, assumptions, queue decomposition method and procedures to build the optimization problem will be illustrated. The performance algorithm which is the mean value analysis (MVA) will be discussed. Then the optimization algorithm and the enumeration algorithm will be shown. After that, experimental results for tandem and split systems, large topologies, and also state-dependent queues networks will be presented. Based on experimental data and analysis, the design rules will be developed. The final section is the summary and conclusion of the research.

1.2 Motivation

Figure 1.2 gives an indication of our problem. In this example, a split topology of a series of five work stations connected by a material handling system is presented, and we wish to topologically arrange the workstations so that the throughput is maximized in our manufacturing system. This problem is challenging because the designer not

only needs to arrange every workstation in each position, but also needs to measure the performance of the system and compute the throughput, work-in-process, and sojourn or cycle times of the topology where a finite circulating population of component parts and finite buffers at the workstations are included. K_i represent the size of finite capacities, μ_i are the service rates, θ_i represent the throughput and N is the finite population in the system.
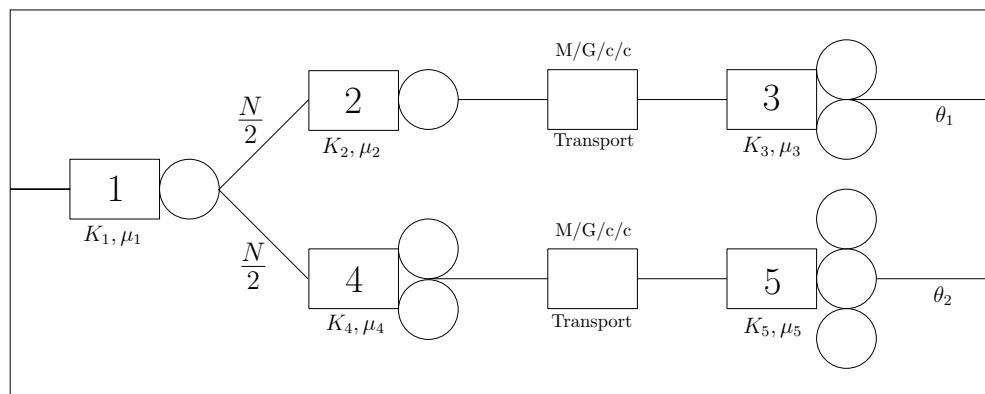
Figure 1.2. Five-node Split Topology

In this thesis, all queueing nodes are considered as $M/M/c/K$ queueing models. This acronym is based on Kendall's notation[18], for example, the system $A/B/C/D$ stands for:

- A: Arrival process (how the jobs or parts arrive to the system, e.g. Poisson Process)

- B: Server process (how the jobs or parts are processed, e.g. Exponential service)

- C: Number of servers (number of computers, servers, or machines)

- D: Queue capacity (including waiting room and number of severs)

Especially, for the $M/M/c/K$ systems, the spelling M means Markovian or memory-less. It also represents the arrival process of the model as the Poisson process. The

3

second M represents the service time distribution which is the Exponential distribution. 'c' is the number of servers at the station and 'k' is the capacity of the queue. The M/G/c/c queues represent the material handing or transportation system (or conveyor). The difference between the M/M/c/K model and the M/G/c/c queue is that the latter follows the general distribution and has the same number of capacities and service channels which are equal to the coefficient 'c'. [3][18].

1.3   Problem overview

This thesis mainly concentrates on closed queueing networks. The definition of closed queuing networks is: there are no external arrivals into the networks and no departures from it either. So one of the most vital properties is: the number of customers or tasks in the network is fixed, or in other words, is a constant. As we mentioned before, the closed queueing networks have many applications such as manufacturing systems, multi-programmed computer systems, telecommunication networks, etc. The closed queueing networks which we study are called a closed Jackson network. They should satisfy the following assumptions[9]:

- I: It has m service stations (nodes)

- II: The service rate at node i is $\mu_i(n)$ when the number of customers at node i is given by n. $\mu_i(0) > 0$

- III: There is a probabilistic choice when a customer completing service at node i and joins the node j. The probability is defined as r_{ij} , it is independent of the number of customers at any nodes and the history of customer routing.

- IV: The routing matrix $R = [r_{ij}]$ is a transition probability matrix of an irreducible DTMC (discrete-time Markov chain)

So the transition matrix of figure 1.1 could be written as below:

$$R = \begin{pmatrix} 0 & 0.3 & 0.4 & 0.3 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

The performance algorithm is also based on the transition matrix as it will be shown later. To find a systematic methodology which could help one get the best arrangement of closed queueing networks, we need to consider and build the topological arrangements problem, which consists of m-service stations, each server's rate μ having an exponential distribution and they may have different rate parameters. The most interesting question is what arrangement should we adopt to maximize the profit, such as throughput, and minimize the cycle time or sojourn time. The different arrangement pattern of nodes would make a great influence for the queueing system. Transport systems (the M/G/c/c nodes) such as conveyors, people, carts and lifts also will be considered. The material handling system (MHS) is an integrated transport network with the buffers of the machine centers. The closed queueing networks which are presented in this thesis have series, merge, and split elements in its topology. Multiple products can be captured with the network approach as they follow specific production plan routes within the facility. Analytical models of such networks should be developed in order to provide fast, efficient and accurate ways of computing the performance. Further, the models can be used for making optimal arrangements of the system.

The topological arrangements problem (TAP) is very similar to the quadratic assignment problem (QAP), which can be simply formulated as:

- There are a set of n facilities and a set of n locations. The distance between each pair of location is specified, and the flows transported between two facilities

Facilities	A	B	C
A	0	40	50
B	40	0	60
C	50	60	0

Table 1.1. Distance matrix of QAP

Facilities	A	B	C
A	0	5	6
B	5	0	7
C	6	7	0

Table 1.2. Flow costs per unit distance matrix of QAP

are specified. The arrangement problem is to assign all facilities to different locations to minimize the sum of flow costs, which is defined as the distance multiplied by the number for flow supplies of each pair of facilities. The data of distance between each location and the flow cost per unit distance are shown in tables 1.1 and 1.2. The matrices are symmetric with zeros along the diagonal. So the cost function can be represented by:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} f_{ij}d(g(i),g(j)) \tag{1.1}$$

The decision variables are defined as: $x_{ij} = 1$ if $g(i) = j$. Otherwise, $x_{ij} = 0$

Each assignment $f_{ij}d(g(i),g(j))$ means the cost of assigning facilities i to location $g(i)$ and j to $g(j)$. The constraints are shown in equations 1.2 and 1.3, which means every facility only can be fixed once, and every location can only be allocated to one facility.

$$\sum_{i=1}^{n} x_{ij} = 1 \quad j = 1, 2, ..., n \tag{1.2}$$

$$x_{ij} \in \{0, 1\}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad i = 1, 2, ..., n \tag{1.3}$$

$$x_{ij} \in \{0, 1\}$$

When formulating the TAP later in the thesis, many similarities will be found between the QAP and the TAP. But since the objective function is quite different, we cannot use the same methods to solve it. What we expect in the TAP is to find the
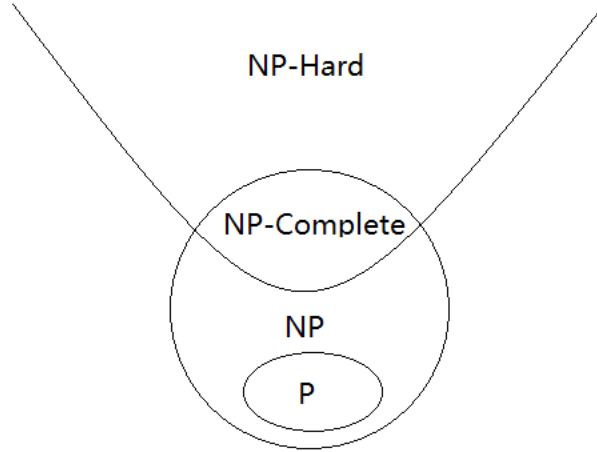
Figure 1.3. classify of computational complexity

best arrangements which can maximize the throughput or minimize the cycle time of system. However, the objective function in the TAP is a bit more complicated than the function of the QAP. Measuring the throughput is dependent upon the queueing model which is used to calculate the flow matrix performance measure and it is highly nonlinear. In most cases, the previous problem requires an exhaustive algorithm to find the best arrangement. While the approach has its limitations,

the performance tool will be helpful in finding and comparing the best arrangement. Similar to the QAP, the TAP is a NP-Hard (Non-deterministic Polynomial-time Hard) problem, which is the hardest class in computational complexity theory. It is widely suspected that there are no polynomial-time algorithms for the NP-hard problem and approximate methods are always going to be used to solve this type of problem. The Venn diagram breakdown of computational complexity classes is shown in figure 1.3. (Note: $P \neq NP$)[12][14]

CHAPTER 2

LITERATURE REVIEW

In this chapter, the past research carried out before by many others will be reviewed. In the first section, the general modeling algorithm for closed queueing network arrangements will be discussed. Next, the difficulty of the optimal arrangements problem and some methods to deal with the problem will be presented. In the last section, some design rules for generating queueing arrangements will be reviewed.

2.1 Closed Queueing Network Models

The paper by Gordan and Newell[10][11], which is central to modeling algorithms for closed queueing networks, provided the key result along with a First-Come First-serve (FCFS) priority queue discipline, and they introduced closed queueing network models with the exponentially distributed inter-arrival and service time distributions. Baskett, Chandy,Muntz and Palacios[7] expanded the research of Gordan and Newell, working on different job classes and mixed networks for both closed and open systems. Shanthikumar and Yao[15] presented new research for solving the optimal server allocation problem. Koenigsberg [4] is probably the first to discuss the closed queueing networks with material handling systems (MHS). A paper by Smith and Kerbache[16] discussed the MHS which is based on the finite closed queueing systems with the M/G/c/c queues. Bedell and Smith[24] provided a study of topological arrangements of finite closed queueing networks which are mainly the foundation of this thesis and proved that the TAP (topological arrangements problem) is a NP-hard problem.

2.2 Problem Difficulty

The optimal arrangement of queues is difficult to solve since there are an exponential number of possible combinations for a topology system. Also measuring the performance of the system is complicated because the blocking in the system is disruptive to the flow of parts in the system. Perros[13] presented the first book for finite closed queueing networks to emphasize the blocking in the network. To deal with the blocking in the system, a modified version of the mean value analysis (MVA) algorithm is used. Reiser and Lavenburg [22] developed a good solution of closed system which depends on the MVA. An approximation is introduced by Akyildiz, I.F [1] which is for the MVA of queueing network with blocking. Also Zhuang, Buzacot, Liu [21] discussed an approximation for the exponential service system with blocking based on a modified MVA approach. Smith[27] provided a methodology called queue decomposition for closed finite network models which is the foundation for the approach needed in this thesis.

2.3 Design Rules

The design rules are the systematic methodologies which can help one reach the best arrangements of queueing system during the design stage. For tandem systems, which are the simplest system for queueing networks, Tembe and Wolff [30] developed an algorithm about the optimal order of service in tandem systems. Genji, Hirotaka and J. George[8] presented two rules which are for the optimal arrangement of tandem queueing systems with blocking to maximize throughput. The first design rule, which is most important, is placing the two worst stations one in the first stage and the second in the last stage. For the $M/M/\infty$ queueing systems, station i is superior to station j if $\mu_i \geq \mu_j$. μ_i represent the service rate of station i $(i = 1, 2, 3...)$. Next, we can posit the second rule to assign the remaining (m-2) stations after ordering them according to the previous method. Suppose the stations are numbered such

10

$m \geq m - 1 \geq \cdots \geq 1$ and station 1 and 2 should be the first and the last stage, then assign station 3 to stage m-1, station 4 to 2, stage 5 to m-2, stage 6 to 3 and so on. Because a function plot of optimal stage vs. station number has the cross-section shape of the interior of a bowl, it is also called the 'bowl phenomenon'. The two rules can be simply stated as:

- I: Place the two worst stations one at the first stage and the second at the last stage.

- II: Following the 'bowl phenomenon', which means better stations are closer to center and worse stations are further from the center.

Lehtonen[20] presented an ordering problem of tandem queues with exponential severs. Also, Betsy and Ronald[2] provided useful insights for tandem systems based on a light traffic approximation which minimizes expected customer delay. They considered in the situation of sufficiently light traffic along with a Poisson arrival process, total expected delay in the queue is minimized when all costumers first go to the slower server, and then to the faster one. Whitt[29] developed possible approximation methods which are called parametric-decomposition approximations for queueing networks whose procedure for each distribution is partially characterized by its first two moments or its mean and squared coefficient of variation. It would help us to determine the order of the queueing system that minimizes the expected equilibrium sojourn time per customer. He provided a useful heuristic design principle. After that Suresh and Whitt[25] gave a simulation experiment for arranging queues in series depending on the parametric-decomposition approximation which was mentioned before. The results show that the most vital factor is variability and if the service-time variability is the same or nearly the same at the all queues, then the order should not matter so much.

CHAPTER 3

QUEUE DECOMPOSITION MODELLING APPROACH

3.1 Assumptions

We assume that the system is similar to an un-paced, asynchronous flow line or flexible manufacturing system where jobs are often routed through the network over a material handling or transportation system. This thesis mainly focuses on the closed queueing network models with a finite population where the service rates are from an Exponential distribution and examine arbitrary topologies of series, merge, and split systems.

Before designing an algorithm for closed finite queueing networks, the appropriate type of blocking mechanism must be specified. Basically there are three kinds of blocking that are relevant in finite closed queueing networks: [23]

1.Type I (Production Blocking): The upstream node i gets blocked if the service on a customer is completed but it cannot move downstream due to the queue at the downstream node j being full. This is sometime referred to as Blocking After Service (BAS)

2.Type II (Communication Blocking): The upstream node is blocked when the downstream node becomes saturated and service must be suspended on the upstream customer regardless of whether service is completed or not. This is sometimes referred to as Blocking Before Service (BBS)

3.Type III (Repetitive Service Blocking): It is also appropriate in situations where the customer cannot enter the upstream node but loops back continually where it receives another independent service again from the upstream node.

Type 1 (Production Blocking) is adopted in this thesis. This blocking policy is very common during daily life applications, since it is the standard policy for queues in which people wait for service, such as determining restaurant seating, processing sales orders, and etc. Some of the most important definitions and notations occur in the following subsection.

3.2 Notation

In order to make a better understanding of the system, concept and calculations, the following presents most all of the notation needed for paper:

$A :=$ Number of queueing chains in the network.

$a_{ij} :=$ Generic routing probability from node i to j in the network.

$c :=$ Number of servers in the $M/M/c$ and $M/G/c/c$ queue.

$G :=$ General service time distribution.

$g :=$ Normalization constant.

$G(V, E) :=$ Graph topology of the closed queueing network with a finite set of nodes V and a finite set of edges E.

$k :=$ Index on a chain in the network.

$K :=$ Capacity of a queue (including service).

$\lambda_{lk} :=$ Poisson arrival (throughput) rate to node l in chain k.

$L :=$ Length of an $M/G/c/c$ queue (in meter).

$M :=$ Markovian arrivals (i.e. Poisson arrivals)

$m :=$ Number of stages in the network.

$\mu_i :=$ Exponential mean service rate at node i

$\mu_c :=$ The processing rate for an M/G/c/c queue with fixed length and width(i.e.$\mu_c = f(V_1, L, W)$ free-flow speed V_1,capacity $c \sim f\ (length \times width)$

$N :=$ Total population of customers in the network.

π_l := Stationary probability of the number of customers at queue l in a closed queueing network.

$\rho := \frac{\lambda}{\mu c}$:= Proportion of time each server is busy.

W := width of an $M/G/c/c$ queue.

$\theta(W_k)$:= Throughput of the closed queueing network as a function of the finite population W_k.

W_k := The number of products(customers) in a single chain (class).

w_{lk} := Average delay at node l in closed network.

x_{lh} := Decision variable in transition matrix for node l and chain k.

3.3 Queue Decomposition Methodology

The Erlang distribution is usually applied when one studies queueing networks since it is the distribution of the sum of k independent exponential distributions. In other words:

$$if \ X \sim Exponential(\lambda) \quad then \ \sum_{i=1}^{k} X_i \sim Erlang(k, \lambda)$$

The probability density function is shown in 3.1:

$$f(x; k, \lambda) = \frac{\lambda^k x^{k-1} e^{-\lambda x}}{(k-1)!} \tag{3.1}$$

Where k is the number of independent exponential distribution in system. The Erlang distribution has many applications. For instance, there are several procedures at airports when traveling, e.g. ticketing, luggage, boarding, seating, etc., and each procedure can follow an exponential distribution, so the total time a passenger would spend during the entire process is an Erlang distribution. The Erlang distribution was developed by A. K. Erlang to examine of servers or lines. In general, it is a very robust probability distribution and relates to the Gamma distribution.

In queueing networks research, there are many ways to decompose finite buffer queues with the Erlang service times. The primary advantage of queueing systems with Erlang service time is their robustness which is well known. The $M/G/K/K$ system is crucial to the decomposition and relaxation process. Smith has presented and examined when the decomposition of the $M/M/1/k$ queue and the $M/G/K/K+M/M/1$ can be made equivalent so that the throughput will be identical. Figure 3.2 illustrates the decomposition concept. The decomposition creates a "holding node" for the flow processes similar to a modeling concept for blocking processes in the Expansion method process. The original system can be simplified by decomposing procedure so this is a very constructive process.
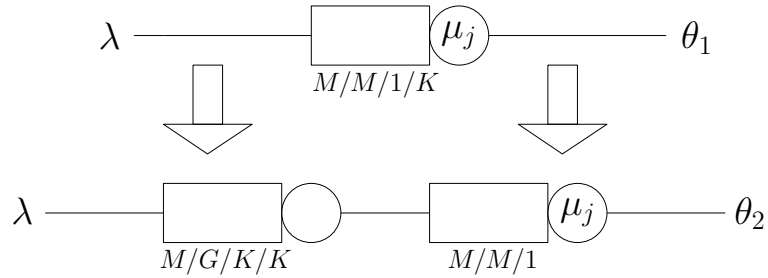
Figure 3.1. Open Network Decomposition Process
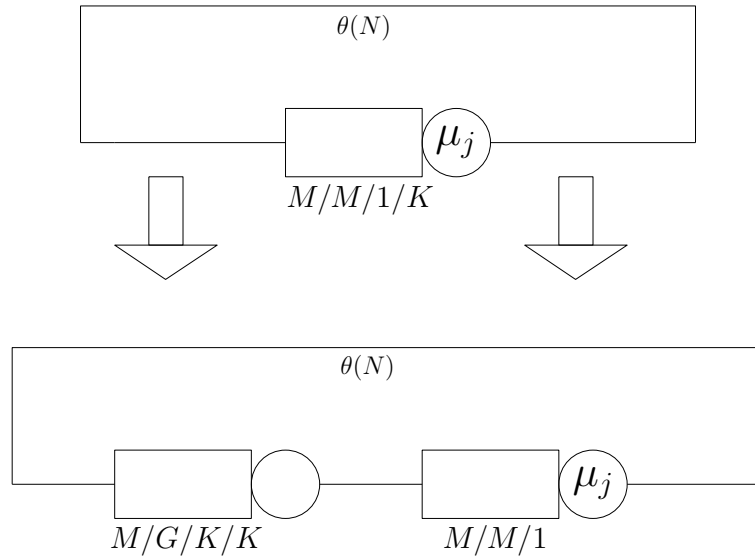
Figure 3.2. Closed Network Decomposition Process

The basic idea behind this methodology is to decompose the finite buffer queues into a buffer and the server. The buffer is modeled as a state-dependent M/G/K/K queue and the server is modeled as an infinite buffer mulit-server system. Let's consider that of an M/M/1/K system is set in the M/M/1 infinite space. Adding the M/G/K/K node is equal to adding a holding node for blocking as in the analytical modeling known as the Generalized Expansion method[19]. The queue decomposition is virtually a modified service rate method to capture the blocking in the system. One can show that adding the queues in this manner provides an upper bound on the system throughput.

Queue decomposition will then result in an approximation for the real network which has under certain circumstances a product form distribution. The key to the process is the state dependent M/G/K/K node which captures the blocking in the network.

With the queue decomposition approach, we have a relaxed approach to the closed queueing network performance modeling. While for each finite queue, we have to add two queues: One is of the M/G/K/K and another is the M/M/c queue, this still makes for a polynomial bounded performance algorithm running within a branch-and-bound environment. Adding the M/G/K/K node is tantamount to adding a holding node for blocking as in the analytical modeling approach or even in the digital simulation program Arena[17]. The queue decomposition approach is essentially a modified service rate method to capture the blocking in the system. For a more detailed explanation of the queue decomposition approach, the reader is referred to the recent paper by Smith[27] which illustrated as follow:

For an M/G/c/c queue, the steady state probabilities are normally generated by the following equations:

$$p_n = \frac{\lambda_0 \lambda_1 ... \lambda_{n-1}}{\mu_0 \mu_1 ... \mu_n} p_0 \tag{3.2}$$

λ_i is Poisson arrival rate to node i, and μ_i is Exponential service rate at node i. p_n represent the probability of the situation whose number of customers is equal to n at steady state. Based on 3.2, we have

$$\frac{1}{p_0} = 1 + \sum_{n=1}^{c} \frac{\lambda_0 \lambda_1 ... \lambda_{n-1}}{\mu_0 \mu_1 ... \mu_n} \tag{3.3}$$

In the context of our investigation, the arrival rates are not influenced by n, and thus, we define λ, such that $\lambda = \lambda_0 = \lambda_1 = ... = \lambda_c$ which yields:

$$p_n = \frac{\lambda^n}{\prod_{i=1}^{n} \mu_i} p_0, n = 1, ...c \tag{3.4}$$

and

$$\frac{1}{p_0} = 1 + \sum_{n=1}^{c} \left\{ \frac{\lambda^n}{\prod_{i=1}^{n} \mu_i} \right\} \tag{3.5}$$

In developing the $M/G/c/c$ model we assume that, μ_n, the service rate of each of the n occupied servers, is dependent upon the number of parts or products on the transport device by an exponential function. The explicit form of the exponential function is based on the speed density curves relevant to the transport device.

The exponential state dependent delay curve we utilize to fit the material handing speed or transport velocity is derived in the following way[26]. If we assume an exponential decay relationship of the following form:

$$V_n = V_1 exp \left[- \left(\frac{n-1}{\beta} \right)^\gamma \right] \tag{3.6}$$

where V_n is the velocity of the $n^t h$ customer, V_1 is free-flow speed of an occupant, and β and γ are parameters. β and γ are determined algebraically by solving for the

following equations where a and b are abscissas of the ordinates V_a, V_b fitted to the exponential curve, see[26]

$$\gamma = \frac{ln\left(\frac{ln(V_a/V_1)}{ln(V_b/a)}\right)}{ln(\frac{a-1}{b-1})} \tag{3.7}$$

$$\beta = \frac{a-1}{[ln(V_1/V_a)]^{\frac{1}{\gamma}}} = \frac{b-1}{[ln(V_1/V_b)]^{\frac{1}{\gamma}}} \tag{3.8}$$

With β, γ then the service rate which is used in the Mean Value Analysis algorithm is:

$$\mu_n = nexp[-(\frac{n-1}{\beta})^{\gamma}] \tag{3.9}$$

Then substituting μ_n into Equation 3.3 and 3.4 we obtain

$$P_n = \frac{\lambda^n}{\prod_{j=1}^{n} \frac{V_1}{L} exp[-(\frac{j-1}{\beta})^{\gamma}]} P_0, \qquad n = 1, ...c \tag{3.10}$$

and where

$$\frac{1}{P_0} = 1 + \sum_{n=1}^{c} \left\{ \frac{\lambda^n}{\prod_{j=1}^{n} j\frac{V_1}{L} exp[-(\frac{j-1}{\beta})^{\gamma}]} \right\} \tag{3.11}$$

3.4   Performance Modeling algorithm

One of the most practical and effective methods for the performance analysis of our problem is the mean value analysis (MVA) method. The MVA is a recursive technique for computing queue lengths, waiting times at each node, and throughput in a closed queueing network. It is based on the arrival theorem, which states that "upon arrival at a station, a job observes the system as if in steady state at an arbitrary instant for the system with one job removed." [28]The mean value analysis is based on the mean waiting time and the mean queue size of the system. Reiser and Lavenberg developed an efficient algorithm for the performance of closed networks. The MVA starts with zero customers, and then increases the number of customers by one at each iteration until reaching the defined number of total customers N in the

system. Also, Little's law which is the base of the MVA is necessary to be presented before we discuss the algorithm. Little's law is shown in equation 3.12:

$$L = \lambda W \tag{3.12}$$

Where L means the long-term average number of customers in a stable system. λ is long-term average effective arrival rate. W is the average time a customer spends in the system. Next, the MVA is shown below:

- 1.Little's equation for each queue to compute the mean queue length for $i = 1, 2, ..., A$

$$n_l(i) = \sum_{k=1}^{A} \lambda_{lk} \omega_{lk}(i) \tag{3.13}$$

where $n_l(i)$is the expected length of queue l.

- 2.Using Little's law computing the system throughput.

$$\lambda_{lk}(i) = \frac{i_k}{[\sum_{i=1}^{N} \omega_{lk}(i) a_{lk}(i)]} \tag{3.14}$$

- 3.Reiser and Lavenberg's property of product-form network:

$$\omega_{lk}(i) = \tau_{lk}[1 + n_l(i - e_k)] \tag{3.15}$$

where ω_{lk} is the expected delay of chain(k) products at queues$(l).e_k$ is a vector of all zeroes except in the $k^t h$ component which is set to 1 and τ_{lk} is the exponential service time at queue l.[17]

3.5 Optimization Modeling Problem

Since developing a design rule, we need to consider and make a best decision for the position of each queue location. To find the best arrangements, an optimization

model is needed which is formulated by determining the objective function, constraints and decision variables. The procedure of formulating decision variables for the optimization problem is very similar to what we did in section 1.3, the property of Jackson closed network. In that model, there is a probabilistic choice when a customer completing service at node i and joins node j. The probability is defined as r_{ij}. So we can also define the problem which is maximizing the throughput of queueing system and consider the exhaustive order or arrangement of nodes by the transition matrix [A]. The element a_{ij} means the probability of the flow departure from node i and arrival on node j is equal to a_{ij} . The transition matrix is a good tool to plan the arrangement, the elements of matrix are easily changed and determined by optimization method at each iteration to represent the different arrangements of system.

For a tandem system, the constraints can be shown as:

$$\sum_{j=1}^{m} a_{ij} = 1 \qquad i = 1, 2, ...m \qquad (3.16)$$

$$\sum_{i=1}^{m} a_{ij} = 1 \qquad j = 1, 2, ...m \qquad (3.17)$$

Where n is the number of nodes. The above equation represents the sum of each row and each column in the transition matrix is equal to 1. In a tandem system, there is only one choice of arriving node for each departure node. Since what we concerned with is the closed networks, there should be one and only one departure node and arrival node for each station. So both the sum of each row and each column are 1. But in the topology network, it is not quite true. The transition matrix of figure 1.1 shown below (also shown in chapter 1) is a good example.

$$R = \begin{pmatrix} 0 & 0.3 & 0.4 & 0.3 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Noticing that the constraint 3.16 is always true for all kinds of networks which means the total probability of departure from node i is one. Obviously we can find that the sum of each row is 1, which means the total departure probability is equal to 1, but the sum of each column does not equal 1, because there are more than one nodes having the same destination. So, using the transition matrix to formulate the optimization problem directly is impossible because the constraints are different between tandem and split/merge system. But actually it is possible to transform the topologies to the tandem system. For example, as in figure 3.3 showing, the topology split network can be considered as a tandem system "2 1 3" , and the transition matrix is shown below.
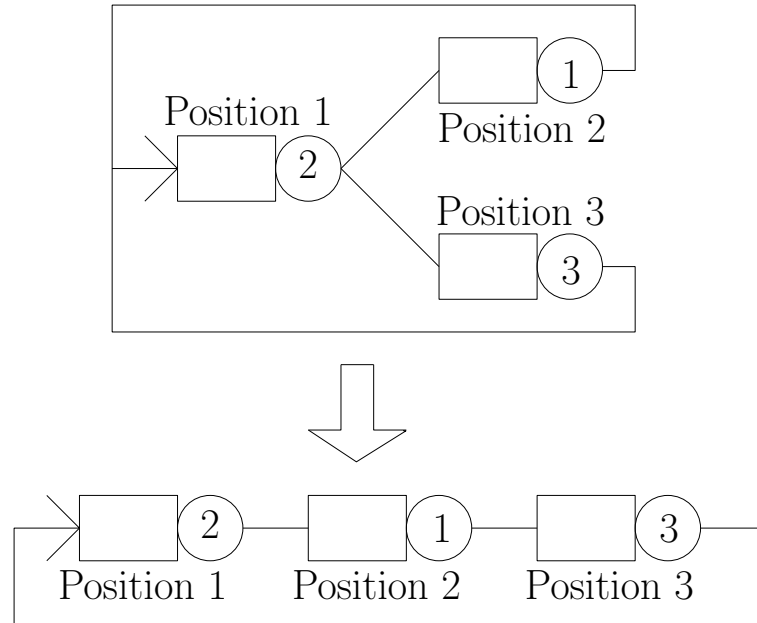
Figure 3.3. Simply Topological to Tandem

$$R = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Since the purpose of using the transition matrix in the optimization problem is to reflect the position of each node in the queueing network, what the optimization algorithm exactly wants to know from the matrix is the information of the arrangement and transfers it to performance algorithm, finally the performance algorithm calculates the throughput of the network. So the topology networks can be represented the same as tandem systems during the initial stage of the optimization. In other words, the transition matrix is the same for tandem arrangement "2 1 3" and split topology "2 1 3", and during the performance procedure, they are still treated as different systems.

So with this transformation process, all types of topologies can be represented by the transition matrix n and with decision variables a_{ij}. Generally, the optimization problem can be represented as:

$$Maximize: \quad Z = \sum_{l=1}^{m} \sum_{k=1}^{A} \theta_{lk}(W_k)$$

$Subject\ to:$

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad i = 1, 2, ...m$$
$$\sum_{i=1}^{m} x_{ij} = 1 \qquad j = 1, 2, ...m$$
$$x_{ij} = \{0, 1\} \quad \forall\ i, j$$

The objective function Z presents the total throughput of the closed queueing network as a function of the finite population W_k. x_{ij} is the decision variable in the transition matrix. The first constraint represents that one and only one activity

22

is assigned to one site, while the second constraint indicates that each activity is assigned for certain.

3.6 Optimization Algorithm

After formulating the optimization problem, a method to solve the problem should be chosen. A most appropriate optimization method needed to deal with the problem should work well for nonlinear programming problems and integer variables since the throughput function is nonlinear and the elements of transition matrix are binary variables. A Fortran subroutine which called the MISQP is employed, a modified sequential quadratic programming (SQP) method. The SQP method is one of the most popular algorithms to solve nonlinear optimization problems.[6]

For the SQP, generally, the optimization problem could be presented as blow:

$$Minimize: \quad f(x)$$

Subject to

$$\begin{cases} g_i(x) \geq 0 \\ h_i(x) = 0 \end{cases}$$

$$\mathbf{x} \in \mathbb{R}^n,$$

So the Lagrangian of the problem is:

$$L(x, \lambda, \sigma) = f(x) - \lambda^t g(x) - \sigma^t h(x)$$

To find the local minimize/maximize points, the first derivatives of objection function should equal to zero, so:

$$\nabla L(x, \lambda, \sigma) = 0,$$

Equivalent to:

$$\begin{cases} \nabla_x L(\lambda, \sigma) = \nabla f'(x) - g'(x)\lambda - h'(x)\sigma \\ \nabla_\lambda = -g'(x) \\ \nabla_\sigma = -h'(x) \end{cases}$$

g' and h' are the Jacobian matrix for $g_i(x)$ and $h_i(x)$. Usually Newton method is employed to solve these equations as shown below:

$$\begin{pmatrix} x_{k+1} \\ \lambda_{k+1} \\ \sigma_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ \lambda_k \\ \sigma_k \end{pmatrix} + \begin{pmatrix} dx_k \\ d\lambda_k \\ d\sigma_k \end{pmatrix}$$

The step size is defined as:

$$\nabla^2 L(x_k, \lambda_k, \sigma_k) \begin{pmatrix} dx_k \\ d\lambda_k \\ d\sigma_k \end{pmatrix} = \nabla L(x_k, \lambda_k, \sigma_k)$$

Newton's method is a method for finding approximate roots of a real-value function. For example, in order to get the root of $f(x) = 0$, the basic concept and iteration is shown as:

$$f(x_0) = (x_0 - x)f'(x_0)$$

x_0 is the start point, and $f'(x_0)$ is the slope of the objective function at x_0. x is the new root point. So each iteration can be shown as :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n))}$$

Actually it is based on Taylor series expansion objective function at $f(x) = 0$, which is:

$$f(x) = 0 = f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''}{2!}(x - x_0)^2 + ... + \frac{f^{(n)}}{n!}(x - x_0)^n$$

The MISQP is very similar to the SQP but adding the functions for solving integer problems based on trust region analysis[5]. The convergence is usually very fast. The flow chart of MISQP is shown in Figure 3.4

24
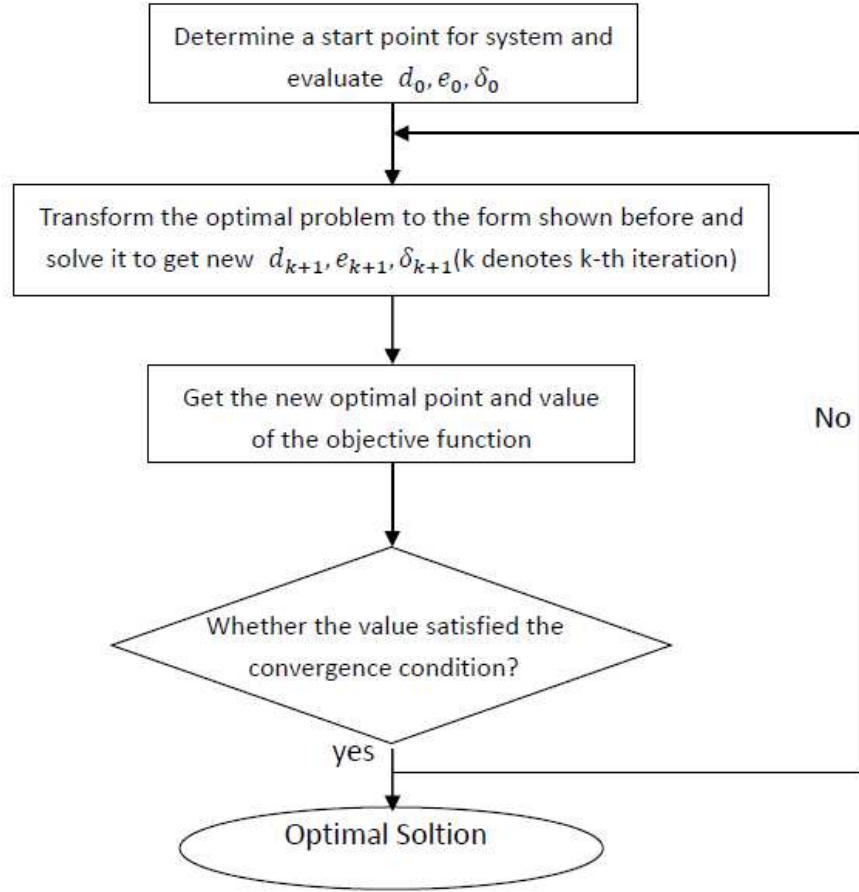
Figure 3.4. Flow chart of MISQP mehtod

3.7 Enumeration Algorithm

Enumeration algorithms, or also called as Exhaustive algorithms, are employed since to compare and verify the MISQP method, an exact comparison method is needed which could generate all combinations of system arrangements. We can generate and compare all of the probabilities in a relatively small system to help us evaluate the rules of arrangement. Also, as said before, we can verify whether the MISQP gives the optimal or sub-optimal solution of the problem.

The enumeration algorithm is built through the Lexicographical Order Method, which is a generalization of all array permutations. It is widely used in computer science. Usually it start at arrangement $'1234....n'$, where n is the number of nodes,

and the algorithm generate a new permutation based on previous one. Each iteration is shown as below:

- Step 1. Let $[P]$ is permutation: $[P] = p_1p_2p_3...p_n$

  From the right side of the permutation, find the first number p_j, which is smaller than its right number. In other words, find: $j = max(i \mid p_i < p_{i+1})$

- Step 2. Find the smallest number p_k from the right side of p_j which is bigger than p_j In other words, find: $k = max(i \mid p_i > p_j)$

- Step 3. Exchange p_k and p_j

- Step 4. Reverse the numbers which are after p_j

The flow chart of the enumeration algorithm is shown as figure 3.5:
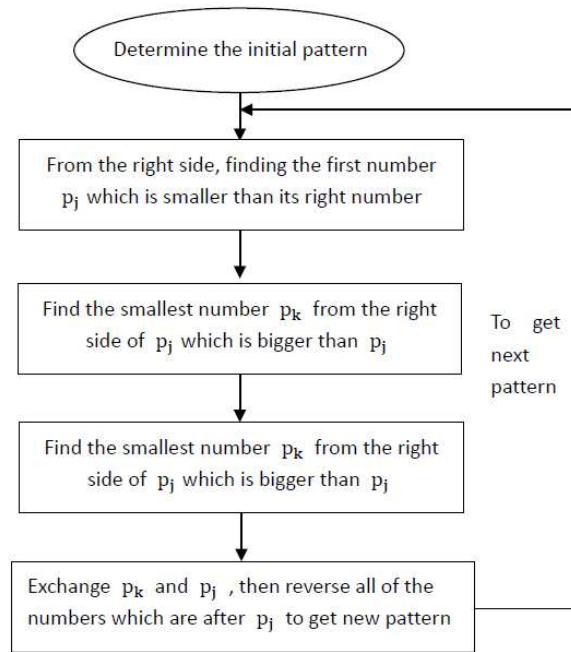
Figure 3.5. Flow chart of Flow chart of enumeration algorithm

3.8 Explanation of algorithm and program

Since the performance algorithm is based on the queue decomposition method which was explained in section 3.3 of the thesis, the program should transform the original network (before decomposition) to the transformed network (after decomposition). For instance, (shown in figures 3.6 and 3.7) station 1 is split into two nodes which are node 1 and node 2. The new node 1 is always next to new node 2 and node 2 is always after the node 1. The transition matrix is shown in table 3.1, from the figure we can see that 2 3 1 means the alternative of original 3 nodes network, and the numbers below it, 3 4 5 6 1 2, imply the pattern of transformed network. The elements $a_{34} = a_{45} = a_{56} = a_{61} = a_{12} = a_{23} = 1$ are the probabilities of departure and arrival which also imply the arrangement of the system.
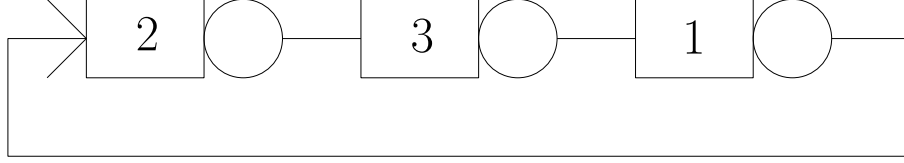
Figure 3.6. The Original Three Nodes Network
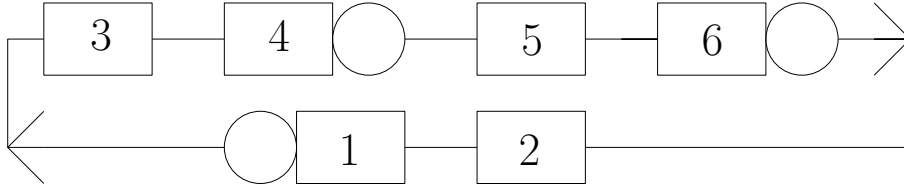
Figure 3.7. After Decomposition Transformation Network

$$[A] = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Table 3.1. Transition matrix for 2 1 3

The output from the enumeration algorithm program for a 3 node tandem system is shown in Table 3.2. To explain the result, the detail of pattern 123 should be discussed, which is the first one illustrated in the table. 1 2 3 presents the arrangement of system, 1 2 3 4 5 6 is the transformed network which has been decomposed. The matrix below it is the transition matrix which is similar with the table 3.1, and also discussed on chapter 3 of the thesis . So the program could provide all possible combinations, the decomposition networks and transition matrices which are required by the MVA analysis to calculate the throughput of systems.

1 2 3						1 3 2					
1	2	3	4	5	6	1	2	5	6	3	4
0	1	0	0	0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	0	1	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	1
1	0	0	0	0	0	0	0	1	0	0	0
2 1 3						2 3 1					
3	4	1	2	5	6	3	4	5	6	1	2
0	1	0	0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0	0	1
0	0	1	0	0	0	1	0	0	0	0	0
3 1 2						3 2 1					
5	6	1	2	3	4	5	6	3	4	1	2
0	1	0	0	0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	0	1	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	1
1	0	0	0	0	0	0	0	1	0	0	0

Table 3.2. The output of enumeration method

As discussed in section 3.4, the buffer speed is one of the coefficients of the through-put function in the performance algorithm. To study the impact of different arrange-ments, the buffer speed should be fixed at a reasonable value to not have bottlenecks which can yield a better approximation in the MVA. The comparison of the through-put in different queueing system by the Arena and the MVA is shown in table 3.3, and we found the buffer speed should be considered individually for each queueing system. For instance, in 3 and 4 tandem system, the best choice of buffer speed is 692.192 cm/s whose throughput is closest to the simulation value.Also the level of blocking makes a big influence on buffer speed. The most suitable buffer speed is lower than the tandem system since the split/merge system has more blocking than the tandem.

	Queueing System	Throughput from MVA(buffer speed=692.192)	Throughput from MVA(buffer speed=92.192)	Throughput from MVA(buffer speed=24.192)	Throughput from Arena
Lighter Blocking	Three Tandem	0.998931289	0.995385826	0.986213088	0.99842
	Four Tandem	0.997770607	0.987361670	0.972364366	0.99754
Heavier blocking	Five merge and split	3.53955054	3.48239431	3.44109821	3.1716
	Seven Split	4.57928467	4.50348463	4.47420835	4.137

Table 3.3. Data for different buffer speeds

CHAPTER 4

DESIGN OF EXPERIMENTS

In this chapter, both tandem and split/merge topologies will be examined. The MISQP method and enumeration algorithm are applied to the results. It is known that the enumeration algorithm should present the best arrangement but it will cost more time than the MISQP algorithm. Also, sensitivity analysis is implemented to examine the results. Simulation analysis is set up for several networks to verify the MVA. In order to discover design rules to predict the best arrangements, the parameters of nodes are set in this way: each node has a service and buffer capacity which is not to dissimilar in order not to create bottlenecks.

4.1 Experiment for tandem system

Tandem networks are the simplest and easiest systems in closed networks. They are also the bases and references for topological networks. Three-nodes, four-nodes, five-nodes, six-nodes and seven-nodes tandem networks are shown in each subsection. Also the simulation results are shown for some systems by Arena. Some rules are expected to be found which can help one to predict the best arrangements, like the "bowl phenomenon", which was discussed earlier.

4.1.1 Three-Node Tandem System

The three nodes tandem system and the network after decomposition is shown in figures 4.1, 4.2. Coefficients of each node are shown in table 4.2. First, let μ_1=1, μ_2=2, μ_3=3 . After that to implement sensitivity analysis, putting $\mu_1 = 2, \mu_2 = 3, \mu_3 = 4$ to

check the influence of service rate changing . The simulation is employed by Arena to verify the MISQP arrangement. The result of two methods are shown in tables 4.2, 4.3. (The total population N=11)
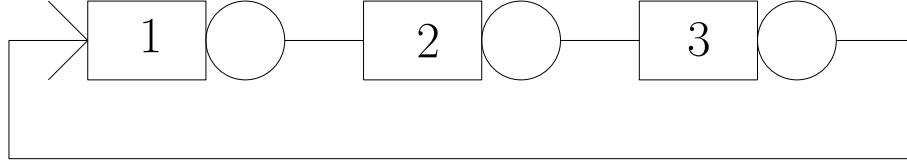
Figure 4.1. The Original Three Node Network
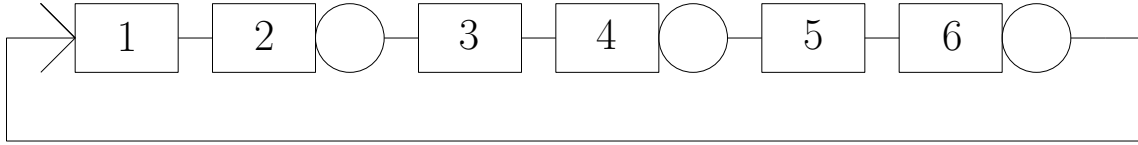
Figure 4.2. Three-Node Tandem Network After Decomposed

Node	Length	Width of buffer	Buffer speed	Capacity(K)
1	2.1336	0.3048	692.192	7
2	2.1336	0.3048	692.192	7
3	2.1336	0.3048	692.192	7

Table 4.1. Three-Node Tandem Network

The %Dev in the tables means the percentage deviation of throughput between the MISQP and the simulation. The results show the difference is really small. Two methods (the enumeration and MISQP) give the same arrangement in both experiments. From the first experiment we found the best station should be put in the center of the system, however in the sensitivity analysis, the central node is the worst one. This may be caused by the limitation of capacities since increasing the service rate in sensitivity analysis would create more blocking in the network than in the

	Enumeration Methods	MISQP	Arena	%Dev (The MISQP and Arena)
Best Arrangement	2 3 1	2 3 1	2 3 1	
Throughput	0.998931289	0.998931289	0.99842	0.5%
Cycle Time	11.0117826	11.0117826	11.0176	0.5%

Table 4.2. Three-Nodes Tandem Network, μ_1=1, μ_2=2, μ_3=3

	Enumeration Methods	MISQP	Arena	%Dev (The MISQP and Arena)
Best Arrangement	2 1 3	2 1 3	2 1 3	
Throughput	1.98359865	1.98359865	1.973	0.54%
Cycle Time	5.545478	5.545478	5.57526	0.53%

Table 4.3. Three-Nodes Tandem Network, $\mu_1 = 2, \mu_2 = 3, \mu_3 = 4$

original one. To prove the hypothesis, an experiment with infinite capacity for each node was done by Arena and shown as in the last table 4.4.

	From Arena	
Arrangement	2 1 3	2 3 1
Throughput	1.9861	1.9871
Cycle Time	5.53849	5.53570

Table 4.4. Arean for Three-Node Tandem Network With Infinite Capacities

So in the infinite capacities situation, the best arrangement is to set the best node at the center. The rule for best arrangements may be the same as the "bowl phenomenon" which is also putting the two worst stations at the first and last position, however still more experiments are needed to prove it.

4.1.2 Four-node tandem system

The network is shown in figure 4.3. Same as three node network, the algorithm decomposed the original networks to the $M/G/k/k$ and the $M/M/1$. The coefficient for each node is shown in table 4.5. Similar to the procedures were done before, first,

let $\mu_1 = 1, \mu_2 = 2, \mu_3 = 3, \mu_4 = 4$. After that, to test the influence of capacities changing, $k_1 = 6, \ k_2 = 7, \ k_3 = 9, \ k_4 = 10$ is set in sensitivity analysis, the capacities of each node are changed but the total number of capacities is same as the original one. The result of the two experiments are shown in tables 4.6, 4.7.
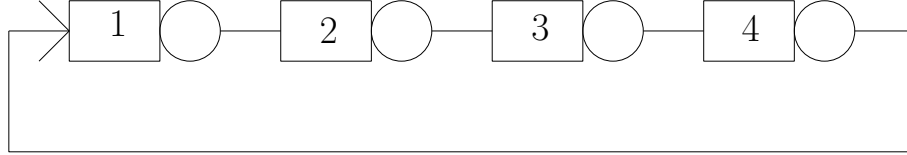
Figure 4.3. The Original Four Node Network

Node	Length	Width of buffer	Buffer speed	Capacity(K)
1	2.4384	0.3048	692.192	8
2	2.4384	0.3048	692.192	8
3	2.4384	0.3048	692.192	8
4	2.4384	0.3048	692.192	8

Table 4.5. Four-Node Tandem Network

So one can see that the best arrangements for both experiments are satisfied by the rules which made before. The worse nodes are set on both sides of the center position and better nodes are closer to the center. Also, in sensitivity analysis, the difference in throughput is really minor, so the influence of capacity is not significant when the size is already large enough to avoid most of the blocking. In both experiments, the maximum throughput is bigger than the number which is given by the MISQP. But the deviation between the MISQP and the enumeration algorithm is around 0.000001%, so it is close and acceptable. Actually one can see that the influence of different arrangements is very small in the tandem system.

The throughput of the system is close to 1, which is equal to the lowest service rate of nodes in the system. In addition, from the experiments in section 4.1.1, this

	Enumeration Methods	MISQP	Arena	%Dev (The MISQP and Arena)
Best Arrangement	1 3 4 2	1 3 4 2	1 3 4 2	
Throughput	0.997770607	0.997770607	0.9954	0.24%
Cycle Time	17.03799	17.03799	17.07856	0.24%

Table 4.6. Four-Node Tandem Network, $\mu_1 = 1, \mu_2 = 2, \mu_3 = 3, \mu_4 = 4$

	Enumeration Methods	MISQP	Arena	%Dev (The MISQP and Arena)
Best Arrangement	1 3 4 2	1 3 4 2	1 3 4 2	
Throughput	1.19057668	1.19057656	1.18846	0.25%
Cycle Time	21.04925	21.04983	21.099167	0.23%

Table 4.7. Four-Node Tandem Network, $k_1 = 6, \ k_2 = 7, \ k_3 = 9, \ k_4 = 10$

phenomenon is also be found, and the throughput is close to 2 in sensitivity analysis since the lowest service rate is also 2 in that case.

4.1.3 Five-node tandem system

The network is shown in figure 4.4. First, let $\mu_1 = 1, \mu_2 = 2, \mu_3 = 3, \mu_4 = 4, \mu_5 = 5$. As before the phenomenon that the throughput is almost the same as the lowest service rate in the system was found in both original and sensitivity experiments of the 3 node and 4 node tandem systems. So to prove this hypothesis, $\mu_1 = 1.2, \mu_2 = 2, \mu_3 = 3, \mu_4 = 4, \mu_5 = 5$ is set in the sensitivity analysis, to measure the relationship between service rate and throughput. The result is shown in tables 4.9, 4.10.
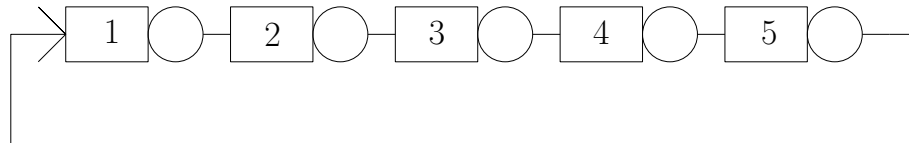
Figure 4.4. The Original Five Node Network

Node	Length	Width of buffer	Buffer speed	Capacity(K)
1	1.8288	0.3048	692.192	6
2	2.1336	0.3048	692.192	7
3	2.4384	0.3048	692.192	8
4	2.7432	0.3048	692.192	9
5	3.0480	0.3048	692.192	10

Table 4.8. Five-Node Tandem Network

	Enumeration Methods	MISQP	Arena	%Dev (The MISQP and Arena)
Best Arrangement	1 3 5 4 2	1 2 3 4 5	1 3 5 4 2	
Throughput	0.9976601	0.9976326	0.9943	0.34%
Cycle Time	21.04925	21.04983	21.120386	0.33%

Table 4.9. Five-Node Tandem Network, $\mu_1 = 1, \mu_2 = 2, \mu_3 = 3, \mu_4 = 4, \mu_5 = 5$

	Enumeration Methods	MISQP	Arena
Best Arrangement	1 3 5 4 2	1 3 4 2 5	1 3 5 4 2
Throughput	1.19057668	1.19057656	1.18846
Cycle Time	21.04925	21.04983	21.099167

Table 4.10. Five-Nodes Tandem Network, $\mu_1 = 1.2, \mu_2 = 2, \mu_3 = 3, \mu_4 = 4, \mu_5 = 5$

From the tables, the results also followed the design rules. The improvement of throughput is significant in the sensitivity experiment. So the slowest station is the bottleneck of the system and makes a big influence on throughput. This time the MISQP gives a different arrangement between the enumeration algorithm. But the difference between best and sub-optimal arrangement is really small, and the MISQP cannot yield the best arrangement since it may has already converged to a sub-optimal point.

4.1.4 Six-node tandem system

The data and results are shown in table 4.14. Again, the results show that the order in the tandem system is not able to make a huge influence on throughput of the system, and a lot of different arrangements have the same throughput. From the experiments before we find the throughput of system is usually determined by the worst node, and expect this rule is also established in six node system.
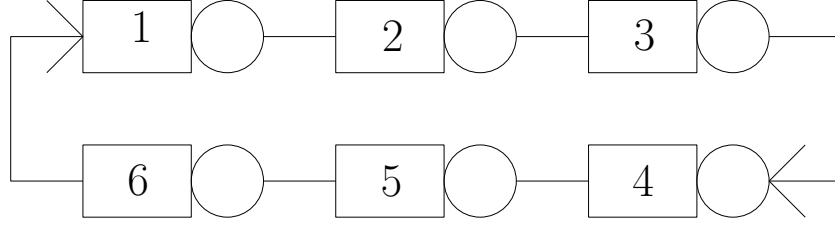
Figure 4.5. The Original Six-Node Network

Node	Length	Width of buffer	Buffer Speed	Capacity
1	2.1336	0.3048	692.192	7
2	2.1336	0.3048	692.192	7
3	2.1336	0.3048	692.192	7
4	2.1336	0.3048	692.192	7
5	2.1336	0.3048	692.192	7
6	2.1336	0.3048	692.192	7

$(\mu_1 = 1 \; \mu_2 = 2 \; \mu_3 = 3 \; \mu_4 = 4 \; \mu_5 = 5 \; \mu_6 = 6)$

Table 4.11. Six-Node Tandem Network

	Arrangement	Throughput from MVA	Throughput From Arena	%Dev (The MISQP and Arena)
The best arrangement	2 4 5 3 1 6	0.997656584	0.9937	0.4%
Follows the rule	1 3 5 6 4 2	0.997656465	0.9934	0.43%

Table 4.12. Throughput of Six-Node Tandem Network

The results did not satisfy the rule ("bowl phenomenon") since the capacity is not large enough to avoid most of the blocking in the system, like what was discussed

earlier in the three node tandem system. However, the throughput is close to the lowest service rate in the system, and the difference between the best and the arrangement which follows the "bowl phenomenon" is really small. So it still could give an acceptable result.

4.1.5 Seven-node tandem system
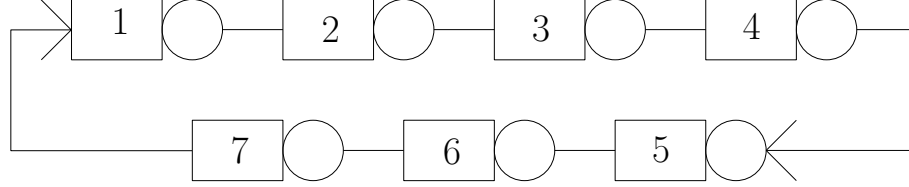
Figure 4.6. The Original Seven-node Network

Node	Length	Width of buffer	Buffer Speed	Capacity
1	2.1336	0.3048	692.192	7
2	2.1336	0.3048	692.192	7
3	2.1336	0.3048	692.192	7
4	2.1336	0.3048	692.192	7
5	2.1336	0.3048	692.192	7
6	2.1336	0.3048	692.192	7
7	2.1336	0.3048	692.192	7

$(\mu_1 = 1 \ \mu_2 = 2 \ \mu_3 = 3 \ \mu_4 = 4 \ \mu_5 = 5 \ \mu_6 = 6, \mu_7 = 7)$

Table 4.13. Seven-Node Tandem Network

	Arrangement	Throughput from MVA	Throughput From Arena	%Dev (The MISQP and Arena)
The best arrangement	5 7 1 2 3 4 6	0.997582341	0.9923	0.53%
Follows the rule	1 3 5 7 6 4 2	0.997581453	0.9921	0.55%

Table 4.14. Throughput of Seven-Node Tandem Network

The seven node experiments proved what was discussed above. Since the difference is small, the arrangement which follows the design rule is acceptable even though it

37

is not the best. So considering this rule is enough for tandem system, and more attention should be paid to the split/merge topology system in which the different arrangements may have bigger improvement, rather than tandem system.

4.2   Experiment for Split/Merge Topology networks

Different from the tandem systems, the split/merge topology systems are thought of having much more variability between the different arrangements. Also, the expected applications of split/merge topology closed queueing networks can be more varied than tandem. In the topology experiments, some rules are expected to help one get the best arrangements, also it is useful to find a better starting point in the MISQP.

4.2.1   Four nodes split and merge topology system

Similar to the tandem system, decomposing the original networks is necessary. The original system is shown in figure 4.7. Let $\mu_1 = 1, \mu_2 = 2, \mu_3 = 3, \mu_4 = 4$. Especially, in sensitivity analysis, the capacity of each node is changed to $k_1 = k_2 = k_3 = k_4 = 4$ in order to check the influence of capacity size. The results are shown in tables 4.16, 4.17. (N=5)
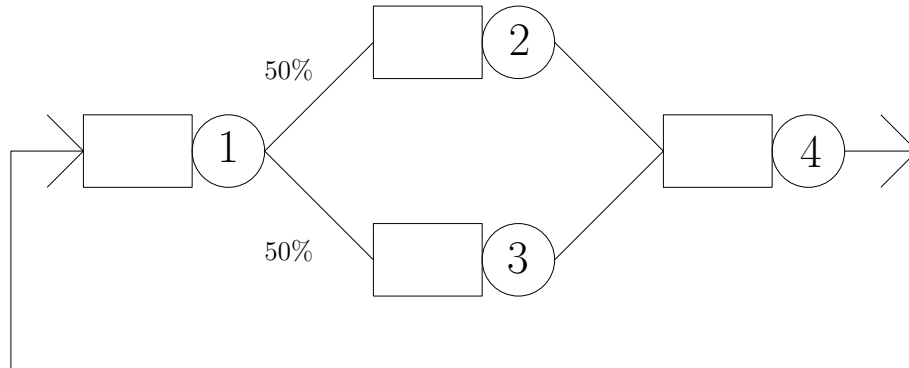
Figure 4.7. Four node split and merge topology system; Transformation

Node	Length	Width of buffer	Buffer speed	Capacity(K)
1	0.6096	0.3048	24.192	2
2	0.9144	0.3048	24.192	3
3	1.2192	0.3048	24.192	4
4	1.524	0.3048	24.192	5

Table 4.15. Four node split and merge topology system

	Enumeration Methods	MISQP	Arena	%Dev (The MISQP and Arena)
Best Arrangement	3 1 2 4	3 2 1 4	3 1 2 4	
Throughput	1.92876124	1.9287492	1.89967	1.53%
Cycle Time	2.5923	2.5923	2.6303	1.46%

Worst Throughput	0.99951649
Best Throughput	1.92876124

Table 4.16. Four node split and merge topology system, $\mu_1 = 1, \mu_2 = 2, \mu_3 = 3, \mu_4 = 4$

	Enumeration Methods	MISQP	Arena	%Dev (The MISQP and Arena)
Best Arrangement	4 2 1 3	4 2 1 3	4 2 1 3	
Throughput	1.91586149	1.91586149	1.88679	1.54%
Cycle Time	2.60979	2.60979	2.65000	1.53%

Worst Throughput	0.98088795
Best Throughput	1.91586149

Table 4.17. Four node split and merge topology system, $k_1 = k_2 = k_3 = k_4 = 4$

From the results, the following hypothesis can be made: in a split/merge topology system, to maximize throughput, the best nodes should be placed at the merge positions. So in these experiments, the two best nodes need to be set on position 1 and

4. More experiments and evidence should be provided to prove it. This hypothesis also could help one to choose the starting point of the MISQP method.

In sensitivity analysis, even though the capacities were larger, the throughput was decreased instead of improving it. In fact, the total of population and the total of the capacities should be changed at the same time, The throughput is a concave function of population when the capacities are fixed. So to maximize the throughput, the number of tasks in system need to match the total capacity. Actually the proper number came from the following formula:

$$N = \left(\frac{\sum_{i=1}^{n} k_i + n}{2} \right) - 1 \tag{4.1}$$

N is the proper population, n is the number of nodes. k_i is the capacity of node i. For $k_1 = k_2 = k_3 = k_4 = 4$, $N = 9$.

4.2.2 Five node split topology system(balance)
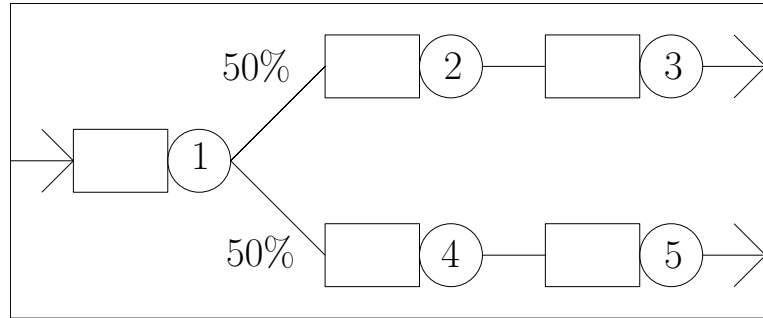
Figure 4.8. Five node split topology system

From the system figure 4.8, obviously, position 1 in the system is the most important since it is the split node. To maximize the throughput, the best node which has the largest capacity and the highest service rate should be placed on it. Because the two split chains are parallel, discussing the arrangement on each chain is meaningless and more experiments are needed to determine the arrangement rules.

Node	Length	Width of buffer	Buffer speed	Capacity(K)
1	1.524	0.3048	692.192	5
2	1.524	0.3048	692.192	5
3	1.524	0.3048	692.192	5
4	1.524	0.3048	692.192	5
5	1.524	0.3048	692.192	5

(Let $\mu_1 = 1$ $\mu_2 = 2$ $\mu_3 = 3$ $\mu_4 = 4$ $\mu_5 = 5$ N=14)

	Enumeration Methods	MISQP	Arena	%Dev (The MISQP and Arena)
Best Arrangement	5 1 2 3 4	5 1 2 3 4	5 1 2 3 4	
Throughput	1.99567437	1.99567437	1.9631	1.6%
Cycle Time	7.0152	7.0152	7.1315	1.7%

Table 4.18. Five node split topology system

4.2.3   Five node split topology system(imbalance)
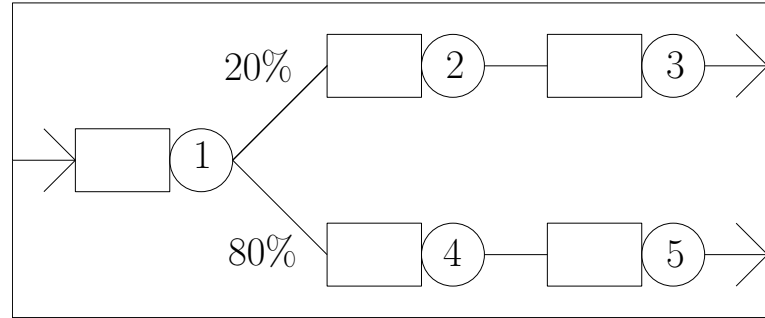
Figure 4.9. Five node split topology system(imbalance)

From the results in table 4.19, since this system is an imbalanced system, the difference between each arrangement became greater, even though the split node (position 1) is the same. Different arrangements on the chains could make a big influence. The network on a split chain can be substantially equivalent to a tandem system, so the worst node is the bottleneck in the network. From these two experiments (balance and imbalance), two design rules are developed to maximize the throughput for these systems:

Node	Length	Width of buffer	Buffer speed	Capacity(K)
1	1.524	0.3048	692.192	5
2	1.524	0.3048	692.192	5
3	1.524	0.3048	692.192	5
4	1.524	0.3048	692.192	5
5	1.524	0.3048	692.192	5

($\mu_1 = 1$ $\mu_2 = 2$ $\mu_3 = 3$ $\mu_4 = 4$ $\mu_5 = 5$ N=13.)

	Enumeration Methods	MISQP	Arena	%Dev (The MISQP and Arena)
Best Arrangement	5 1 2 3 4	5 1 2 3 4	5 1 2 3 4	
Throughput	3.4534049	3.4534049	3.2245	7.0%
Cycle Time	3.7644	3.7973	4.0316	7.1%

Arrangement	Throughput
5 1 4 2 3	2.39957666
5 2 3 4 1	1.1971513
5 1 2 3 4	3.4534049

Table 4.19. Five node split topology system(imbalance)

- I: Place the best node on the split position (in this case, it is position 1)

- II: Place the worse nodes on the same chain whose arrival probability is lowest.

To test our rules, more experiments are demanded. Also these rules could help one to choose the starting point of the MISQP method.

4.2.4 Seven node split topology system(balance)

The network is shown in figure 4.10

From the design rules made above, to maximize the throughput, node 7, which is the best node in system, should be chosen for position 1. Because the two split chains are parallel, the second design rule is unnecessary in this case.

42
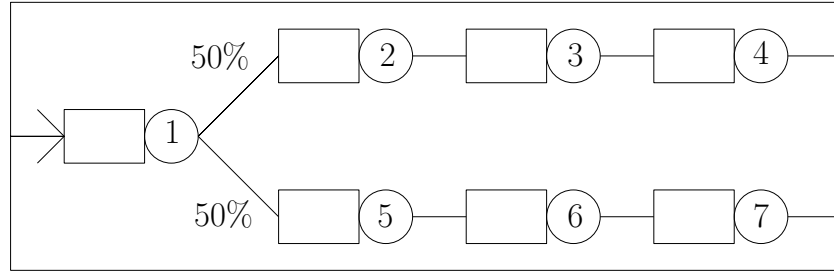
Figure 4.10. Seven node split topology system(balance)

Node	Length	Width of buffer	Buffer Speed	Capacity
1	1.524	0.3048	24.192	5
2	1.524	0.3048	24.192	5
3	1.524	0.3048	24.192	5
4	1.524	0.3048	24.192	5
5	1.8288	0.3048	24.192	6
6	2.1336	0.3048	24.192	7
7	2.4384	0.3048	24.192	8

($\mu_1 = 1$ $\mu_2 = 2$ $\mu_3 = 3$ $\mu_4 = 4$ $\mu_5 = 5, \mu_6 = 6, \mu_7 = 7$ N=20)

	Enumeration Methods	MISQP	Arena	%Dev (The MISQP and Arena)
Best Arrangement	7 1 2 3 4 5 6	7 1 2 3 4 5 6	7 1 2 3 4 5 6	
Throughput	4. 47420835	4. 47420835	4.1478	7.8%
Cycle Time	4.69357	4. 69357	5.0629	7.8%

Table 4.20. Seven node split topology system(balance)

The result is satisfied with the first rule, which requires putting the best node at position 1. However, since the system is a balanced one, to prove the second rule, some imbalance experiments are indispensable.

4.2.5 Seven node split topology system(imbalance)

The coefficients for each node are not changed. From the first design rule, node 7 is required to be placed at position 1. By the second rule, determining node 1,2 and 3 should be sited on the chain whose arrival rate is 20%. So the prediction of the
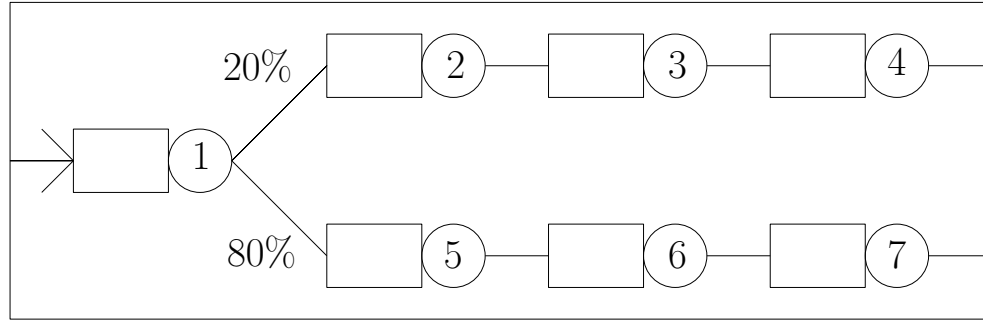
Figure 4.11. Seven node split topology system(imbalance)

Node	Length	Width of buffer	Buffer Speed	Capacity
1	1.524	0.3048	24.192	5
2	1.524	0.3048	24.192	5
3	1.524	0.3048	24.192	5
4	1.524	0.3048	24.192	5
5	1.8288	0.3048	24.192	6
6	2.1336	0.3048	24.192	7
7	2.4384	0.3048	24.192	8

($\mu_1 = 1$ $\mu_2 = 2$ $\mu_3 = 3$ $\mu_4 = 4$ $\mu_5 = 5, \mu_6 = 6, \mu_7 = 7$ N=20)

	Enumeration Methods	MISQP	Arena	%Dev (The MISQP and Arena)
Best Arrangement	7 1 2 3 4 5	7 1 2 3 4 5	7 1 2 3 4 5	
Throughput	1.9745624	1.9745624	1.9374	1.9%
Cycle Time	10.6329	10.6329	10.8415	2.0%

Table 4.21. Seven node split topology system(imbalance)

best arrangement is: 7 1 2 3 4 5 6. To compare with the experiment result, it totally matches the prediction.

4.2.6 Six node split system($20\% \times 5$)

By the first rule, obviously the best station should be set on position 1. Next, actually no matter how we arrange the nodes expect position 1, the throughput is same. Since all split chains have the same arrival probability. To consider and prove the second design rule, an imbalance system is made.
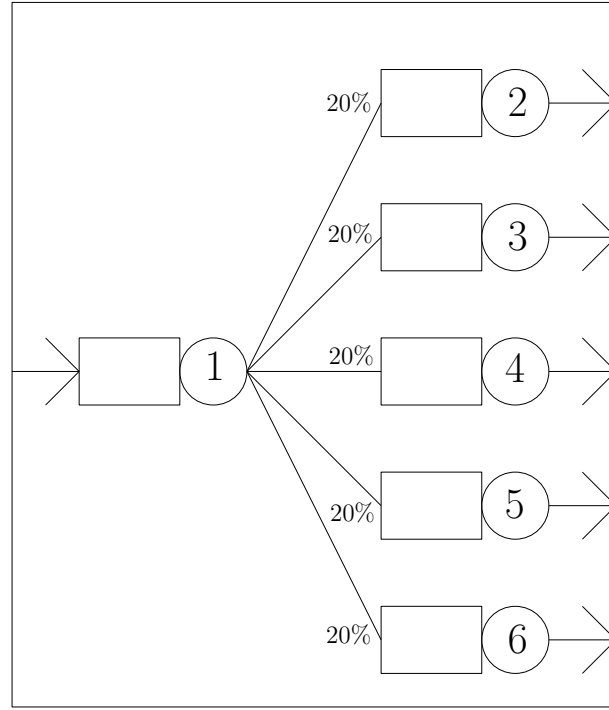
Figure 4.12. Six node split system$(20\% \times 5)$

Node	Length	Width of buffer	Buffer Speed	Capacity
1	0.6096	0.3048	24.192	2
2	0.9144	0.3048	24.192	3
3	1.2192	0.3048	24.192	4
4	1.524	0.3048	24.192	5
5	1.8288	0.3048	24.192	6
6	2.1336	0.3048	24.192	7

$(\mu_1 = 1 \; \mu_2 = 2 \; \mu_3 = 3 \; \mu_4 = 4 \; \mu_5 = 5, \mu_6 = 6, \; \text{N=16})$

	Enumeration Methods	MISQP	Arena	%Dev (The MISQP and Arena)
Best Arrangement	6 1 2 3 4 5	6 1 2 3 4 5	6 1 2 3 4 5	
Throughput	4.89632797	4.89632797	4.6865	4.47%
Cycle Time	3.26775	3.26775	3.41404	4.47%

Table 4.22. Six node split system$(20\% \times 5)$

4.2.7 Six node split system(imbalance)

The system is shown in figure 4.13
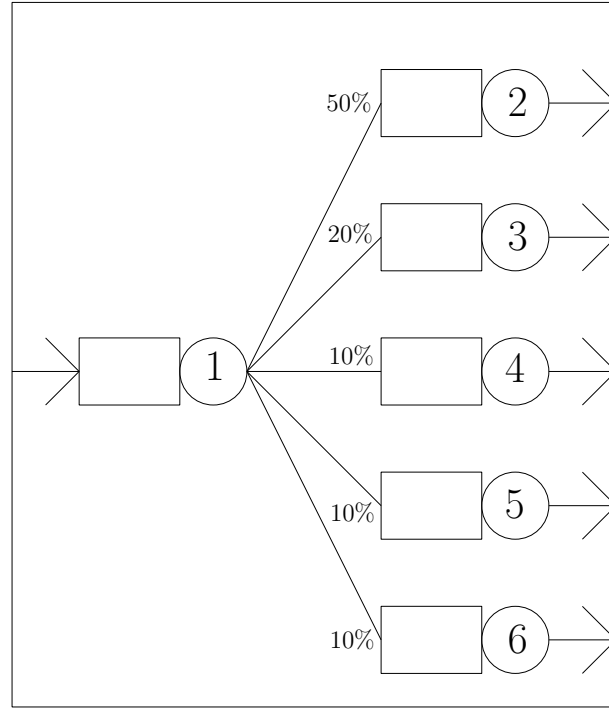
Figure 4.13. Six nodes split system(imbalance)

Node	Length	Width of buffer	Buffer Speed	Capacity
1	0.6096	0.3048	24.192	2
2	0.9144	0.3048	24.192	3
3	1.2192	0.3048	24.192	4
4	1.524	0.3048	24.192	5
5	1.8288	0.3048	24.192	6
6	2.1336	0.3048	24.192	7

$(\mu_1 = 1 \ \mu_2 = 2 \ \mu_3 = 3 \ \mu_4 = 4 \ \mu_5 = 5, \mu_6 = 6, \ N=16)$

	Enumeration Methods	MISQP	Arena
Best Arrangement	6 5 4 3 2 1	6 5 4 1 2 3	6 5 4 1 2 3
Throughput	5.953156	5.953156	5.4676
Cycle Time	2.68765	2.68765	2.7434

Table 4.23. Six nodes split system(imbalance)

The same as before, from the first rule, the node on position 1 is a split station and the best node should be allocated to it. From the second design rule, better nodes are sited at the position whose arrival rate is higher than others, which means one

needs to set node 5 at position 2, and node 4 at position 3. The throughput does not change no matter how we arrange the others since the probability is same at position 4, 5 and 6. The result shown in table 4.23 proves the prediction.

4.2.8 Five Node Split merge(balance)
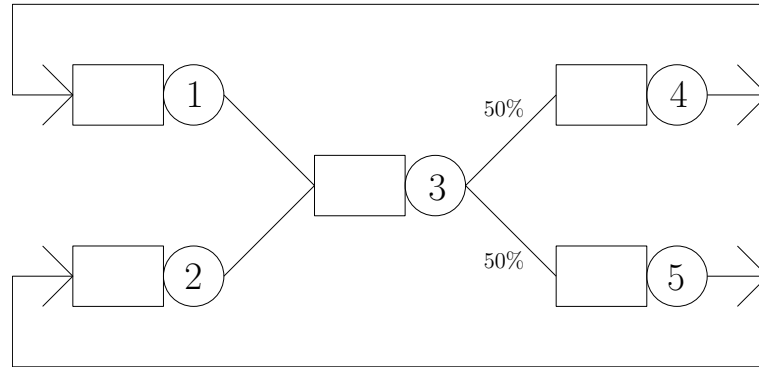
Figure 4.14. Five node merge and split system

Node	Length	Width of buffer	Buffer Speed	Capacity
1	0.6096	0.3048	24.192	2
2	0.9144	0.3048	24.192	3
3	1.2192	0.3048	24.192	4
4	1.524	0.3048	24.192	5
5	1.8288	0.3048	24.192	6

$(\mu_1 = 1 \; \mu_2 = 2 \; \mu_3 = 3 \; \mu_4 = 4 \; \mu_5 = 5, \; N=13)$

	Enumeration Methods	MISQP	Arena	%Dev (The MISQP and Arena)
Best Arrangement	1 3 5 2 4	2 1 5 3 4	1 3 5 2 4	
Throughput	1.99823976	1.99823976	1.91832	4.17%
Cycle Time	6.50573	6.50573	6.77676	4.16%

Table 4.24. Five node merge and split system

The network is shown in figure 4.14. Notice that the split stage is position 3. By the first design rule, the most important procedure to maximize the throughput

is setting the best node at position 3. Since other nodes are not as important as it, considering how to assign them is not necessary. Also, the final result from the experiments proves our hypothesis.

4.2.9 Five Node Split merge(imbalance)
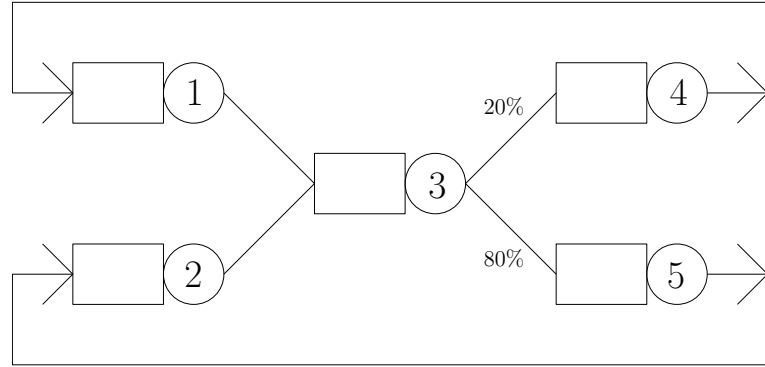
Figure 4.15. Five node merge and split system(imbalance)

Node	Length	Width of buffer	Buffer Speed	Capacity
1	0.6096	0.3048	24.192	2
2	0.9144	0.3048	24.192	3
3	1.2192	0.3048	24.192	4
4	1.524	0.3048	24.192	5
5	1.8288	0.3048	24.192	6

$(\mu_1 = 1 \ \mu_2 = 2 \ \mu_3 = 3 \ \mu_4 = 4 \ \mu_5 = 5, \ N=13)$

	Enumeration Methods	MISQP	Arena	%Dev (The MISQP and Arena)
Best Arrangement	1 3 5 2 4	1 4 5 2 3	1 3 5 2 4	
Throughput	3.44110107	3.4410944	3.1716	7.83%
Cycle Time	3.7786	3.7786	4.099	7.81%

Table 4.25. Five node merge and split system(imbalance)

Similar to the previous network, according to the first rule, node 5 should be placed at position 3. Additionally, since the imbalance situation, by the second rule, determining node 1,2 should be sited on the chain whose arrival rate is 0.2,

48

so the prediction of best arrangement is: 1 3 5 2 4. The result totally matches the prediction.

4.2.10 Three node split topology system
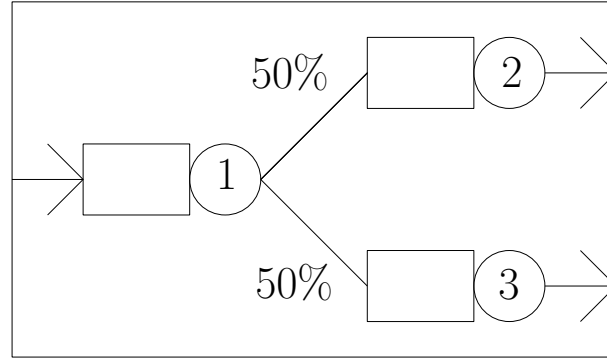
Figure 4.16. Three node split system(imbalance)

All experiments we did before are based on this situation: every node is exactly better(or worse) than others. However, when nodes are not easily distinguished, it is really difficult to say which arrangement would be the best before doing the experiments. For instance, the system shown in figure 4.16, first, let $\mu_1 = 1.667, \mu_2 = 1.25, \mu_3 = 1$. After that to implement sensitivity analysis, let $\mu_1 = \mu_2 = \mu_3 = 1$. The result of two methods is shown in tables 4.27 and 4.28. (N=10)

Node	Length	Width of buffer	Buffer speed	Capacity(K)
1	1.8288	0.3048	24.192	6
2	1.524	0.3048	24.192	5
3	2.4384	0.3048	24.192	8

Table 4.26. Three node split system(imbalance)

When the nodes are not distinguishable, it is really complicated to present the best arrangement by the design rules. Either considering service rate or capacity is not enough. Generally speaking, higher service rate means better performance,

	Enumeration Methods	MISQP
Best Arrangement	1 3 2	1 2 3
Throughput	1.56216812	1.56215942
Cycle Time	6.401639	6.333976

Arrangement	Throughput
1 2 3	1.56215942
1 3 2	1.56216812
2 1 3	1.21860158
2 3 1	1.21863246
3 1 2	0.99073714
3 2 1	0.99073839

Table 4.27. Three node split and merge topology system, $\mu_1 = 1.667$ $\mu_2 = 1.25$ $\mu_3 = 1$

	Enumeration Methods	MISQP
Best Arrangement	3 1 2	3 2 1
Throughput	0.98975337	0.98975140
Cycle Time	10.10356	10.10356

Arrangement	Throughput
1 2 3	0.98439425
1 3 2	0.98441380
2 1 3	0.97952193
2 3 1	0.97953677
3 1 2	0.98975337
3 2 1	0.98975140

Table 4.28. Three node split and merge topology system, $\mu_1 = \mu_2 = \mu_3 = 1$

however, when the capacity is not large enough, there would be more blocking to offset the service rate advantage. On the other hand, bigger capacity would not improve the throughput when the service rate is too low. It may be true that less blocking would happen, but it is not to say that less blocking would definitely improve the performance of network. So there is a trade-off relationship between capacity and

service rate when nodes compare with each node, and instead of simply judge the node by these two coefficients, A method may be needed to determine which node is better. After evaluating the nodes, the design rule can work more reliably.

4.3 Experiments for M/G/c/c system

In many applications of closed queueing network, especially in the manufacturing engineering field, usually a transport system or conveyor is included in the system, which can be treated as M/G/c/c nodes in the queueing network. The M/G/c/c nodes are fixed in the system, so one should consider the arrangement without it. It acts more like a big buffer in the system, and one may expect the property of the M/G/c/c networks is similar to the previous model. Also the rules made for tandem and topologies are examined in the experiments.

4.3.1 Six node M/G/c/c tandem system

The system is shown in figure 4.17.
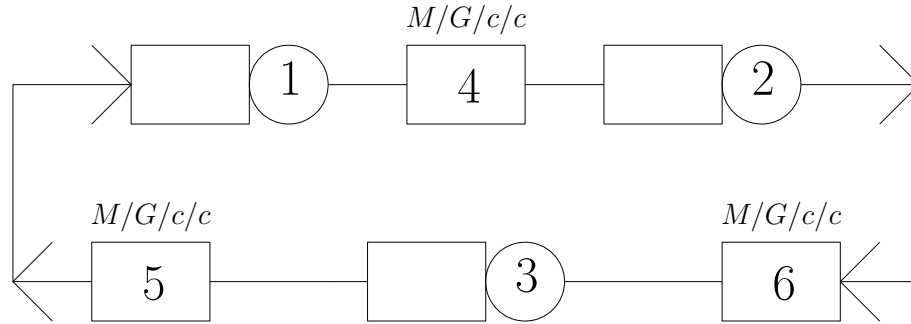
Figure 4.17. Six node M/G/c/c tandem system

From this tandem network experiment, the six node M/G/c/c system is very similar to the four node tandem experiment we did before. The best arrangement is satisfied by the "bowl phenomenon". The difference between each arrangement is very small. However, to test our rules made in the split/merge topology system, more network experiments are needed.

Node	Length	Width of buffer	Buffer Speed	Capacity
1	2.1336	0.3048	692.192	7
2	2.1336	0.3048	692.192	7
3	2.1336	0.3048	692.192	7
4	12.192	0.3048	692.192	40
5	12.192	0.3048	692.192	40
6	12.192	0.3048	692.192	40

$(\mu_1 = 1 \; \mu_2 = 2 \; \mu_3 = 3 \; N = 13)$

	Enumeration Methods	MISQP
Best Arrangement	1 4 3 2	1 2 3 4
Throughput	0.98486645	0.984866331
Cycle Time	16.4125	16.4125

Table 4.29. Six node M/G/c/c tandem system

4.3.2   Five node split M/G/c/c system
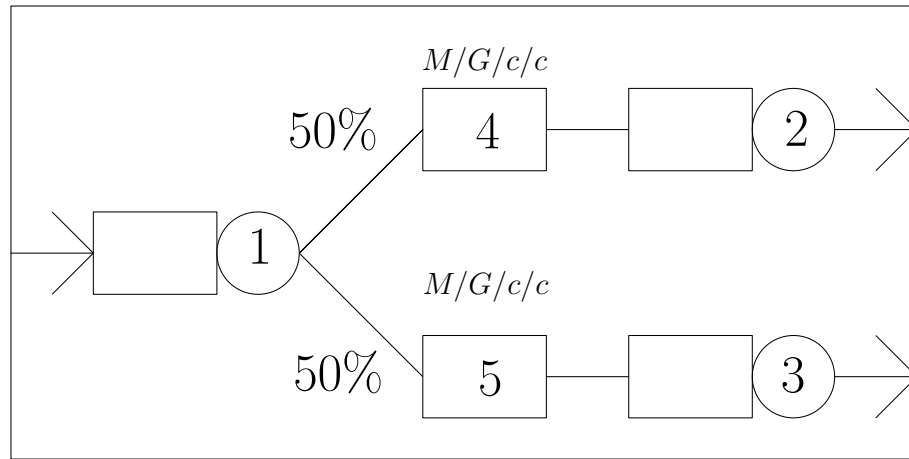
Figure 4.18. Five Node split M/G/c/c system

As we said before, since the M/G/c/c stations could be considered as a big buffer, so the design rules are still can be empolyed to predict the best arrangement for topology networks. In this case, it can be treated as a three node split system, the best node should be set on position 1 by first rule. The result matches our prediction. To test the second rule, an imbalance system is needed.

Node	Length	Width of buffer	Buffer Speed	Capacity
1	2.1336	0.3048	24.192	7
2	2.1336	0.3048	24.192	7
3	2.1336	0.3048	24.192	7
4	12.192	0.3048	24.192	40
5	12.192	0.3048	24.192	40

$(\mu_1 = 1 \ \mu_2 = 2 \ \mu_3 = 3 \ \ N=13)$

	Enumeration Methods	MISQP
Best Arrangement	3 1 2	3 2 1
Throughput	1.96556616	1.9655689
Cycle Time	6.61387	6.61387

Table 4.30. Five Node split M/G/c/c system

4.3.3 Five node split M/G/c/c system(imbalance)
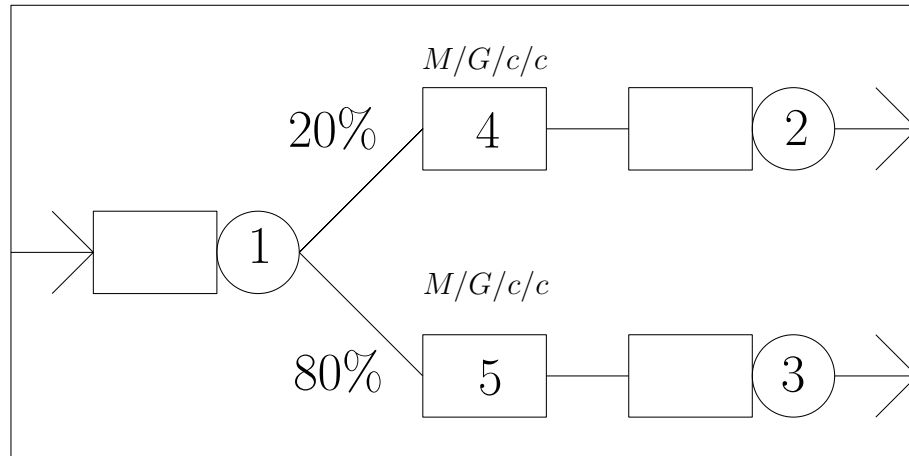
Figure 4.19. Five Node split M/G/c/c system(imbalance)

The system is shown in figure 4.19. This network is similar to the previous system, however, this time since it is imbalanced, both rules are necessary. So after setting node 3 on position 1, node 1 should be placed at position 2 because the arrival probability is lower than the other chain.

Node	Length	Width of buffer	Buffer Speed	Capacity
1	2.1336	0.3048	24.192	7
2	2.1336	0.3048	24.192	7
3	2.1336	0.3048	24.192	7
4	12.192	0.3048	24.192	40
5	12.192	0.3048	24.192	40

$(\mu_1 = 1 \ \mu_2 = 2 \ \mu_3 = 3 \ N=13)$

	Enumeration Methods	MISQP
Best Arrangement	3 1 2	3 1 2
Throughput	2.39182425	2.39182425
Cycle Time	5.4352	5.4352

Table 4.31. Five Node split M/G/c/c system(imbalance)
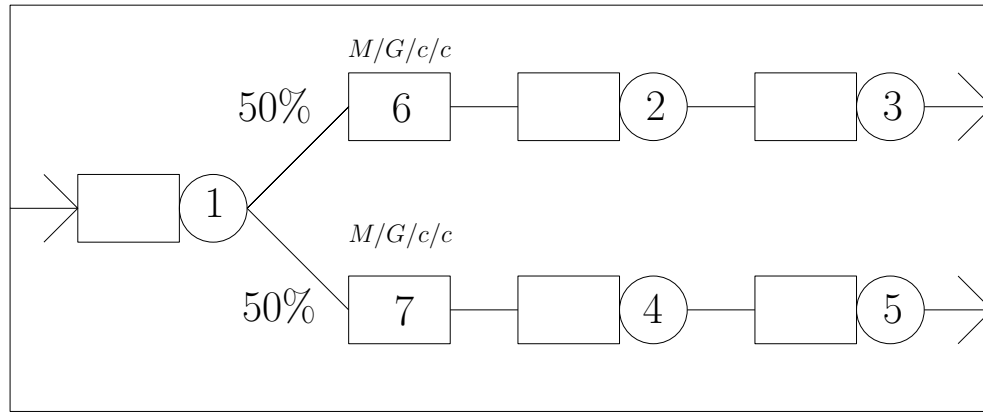
Figure 4.20. Seven Node split M/G/c/c system

4.3.4   Seven Node split M/G/c/c system

Since the M/G/c/c nodes can be considered as a buffer, this system is very similar to the five nodes split system which done before. To maximize the throughput, again, follows the design rule. Because the two split chains are parallel so one does not need to consider the arrangement of nodes on the chain. Also, it seems the rules suggested before also work well in the M/G/c/c networks.

Node	Length	Width of buffer	Buffer Speed	Capacity
1	2.1336	0.3048	24.192	7
2	2.1336	0.3048	24.192	7
3	2.1336	0.3048	24.192	7
4	2.1336	0.3048	24.192	7
5	2.1336	0.3048	24.192	7
6	12.192	0.3048	24.192	40
7	12.192	0.3048	24.192	40

$(\mu_1 = 1\ \mu_2 = 2\ \mu_3 = 3\ \mu_4 = 4\ \mu_5 = 5,\ \text{N}=20)$

	Enumeration Methods	MISQP
Best Arrangement	5 1 2 3 4	5 4 3 2 1
Throughput	1.92014885	1.92014885
Cycle Time	10.415859	10.415859

Table 4.32. Seven Node split M/G/c/c system
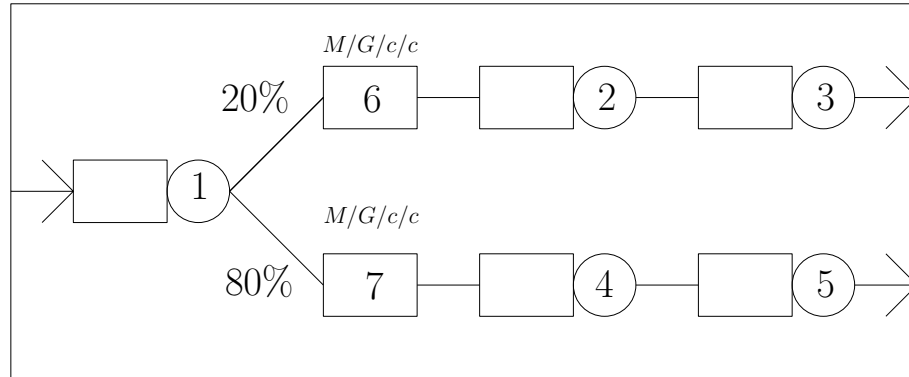
Figure 4.21. Seven Node split M/G/c/c system(imbalance)

4.3.5 Seven Node split M/G/c/c system(imbalance)

Also the system can be considered as a five node split, the best node should be set at position 1. This time the split probabilities are different so the second rule is needed to help one determine the best arrangement. Based on the previous experiences before, one could easily put the better node at position 4 and 5.

Node	Length	Width of buffer	Buffer Speed	Capacity
1	2.1336	0.3048	24.192	7
2	2.1336	0.3048	24.192	7
3	2.1336	0.3048	24.192	7
4	2.1336	0.3048	24.192	7
5	2.1336	0.3048	24.192	7
6	12.192	0.3048	24.192	40
7	12.192	0.3048	24.192	40

$(\mu_1 = 1 \; \mu_2 = 2 \; \mu_3 = 3 \; \mu_4 = 4 \; \mu_5 = 5, \; N=20)$

	Enumeration Methods	MISQP
Best Arrangement	5 1 2 3 4	5 1 2 3 4
Throughput	3.57688928	3.57688928
Cycle Time	5.59145	5.59145

Table 4.33. Seven Node split M/G/c/c system(imbalance)

4.3.6   Six Node split and merge M/G/c/c system

The network is shown in figure 4.22. This system can be treated as the 4 node split and merge system. So split position 1 and position 4 need to have the better nodes according to the first rule.
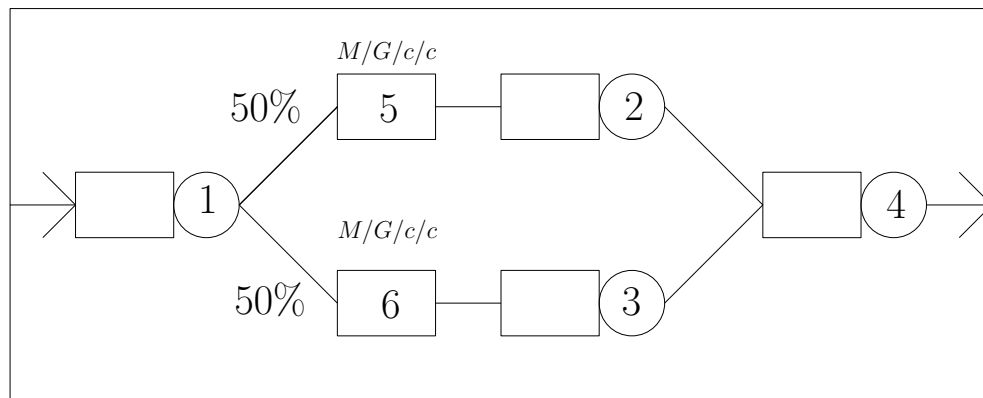
Figure 4.22. Six Node split and merge M/G/c/c system

Node	Length	Width of buffer	Buffer Speed	Capacity
1	2.1336	0.3048	24.192	7
2	2.1336	0.3048	24.192	7
3	2.1336	0.3048	24.192	7
4	2.1336	0.3048	24.192	7
5	12.192	0.3048	24.192	40
6	12.192	0.3048	24.192	40

$(\mu_1 = 1 \ \mu_2 = 2 \ \mu_3 = 3 \ \mu_4 = 4$, N=16)

	Enumeration Methods	MISQP
Best Arrangement	3 1 2 4	3 2 1 4
Throughput	1.96010077	1.96009755
Cycle Time	8.1628485	8.1628485

Table 4.34. Six Node split and merge M/G/c/c system

4.3.7   Six Node split and merge M/G/c/c system(imbalance)
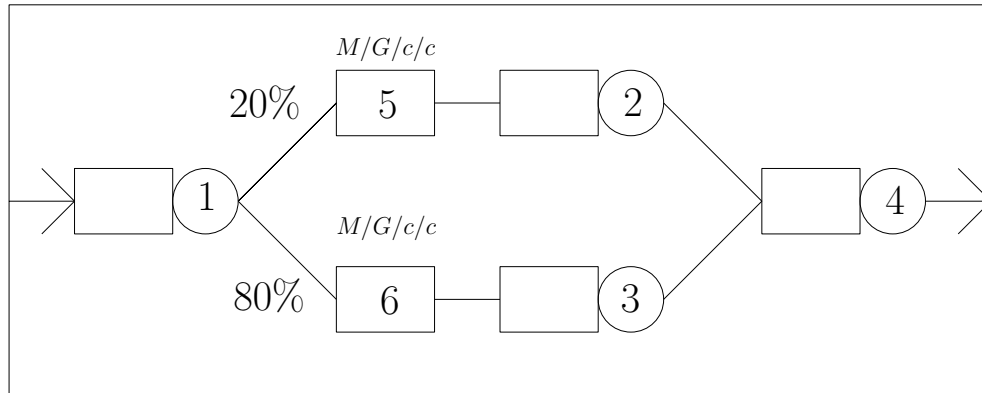
Figure 4.23. Six Node split and merge M/G/c/c system(imbalance)

The network is shown in figure 4.23. Based on the previous experiment result, since the imbalance situation one needs to put the worst node at position 2, which has a lower probability than the other.

Node	Length	Width of buffer	Buffer Speed	Capacity
1	2.1336	0.3048	24.192	7
2	2.1336	0.3048	24.192	7
3	2.1336	0.3048	24.192	7
4	2.1336	0.3048	24.192	7
5	12.192	0.3048	24.192	40
6	12.192	0.3048	24.192	40

$(\mu_1 = 1 \ \mu_2 = 2 \ \mu_3 = 3 \ \mu_4 = 4$, N=16$)$

	Enumeration Methods	MISQP
Best Arrangement	3 1 2 4	4 1 2 3
Throughput	2.39742875	2.39742851
Cycle Time	6.6738167	6.6738167

Table 4.35. Six Node split and merge M/G/c/c system(imbalance)

4.4 Simulation by Arena

To verify our performance method for closed queueing networks, some simulations by Arena are done to compare with the MVA and it also is expected to present an upper bound on the throughput.

4.4.1 Six Node Tandem System

The system is shown in figure 4.24
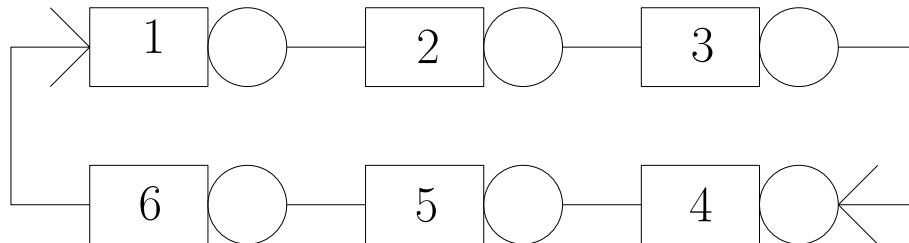
Figure 4.24. Six Node Tandem System for Arena

Node	Length	Width of buffer	Buffer Speed	Capacity
1	2.1336	0.3048	692.192	7
2	2.1336	0.3048	692.192	7
3	2.1336	0.3048	692.192	7
4	2.1336	0.3048	692.192	7
5	2.1336	0.3048	692.192	7
6	2.1336	0.3048	692.192	7

$(\mu_1 = 1\ \mu_2 = 2\ \mu_3 = 3\ \mu_4 = 4\ \mu_5 = 5\ \mu_6 = 6, N = 16)$

Arrangement	Throughput From the MVA	Throughput From Arena	%Dev
2 4 5 3 1 6	0.997656584	0.9937	0.4%
1 3 5 6 4 2	0.997656465	0.9934	0.43%

Arrangement	Cycle Time From the MVA	Cycle Time From Arena	%Dev
2 4 5 3 1 6	16.03678	16.1014	0.4%
1 3 5 6 4 2	16.03678	16.1063	0.43%

Table 4.36. Six Node Tandem System from Arena

4.4.2 Seven Node Tandem System

Figure 4.25. Seven Node Tandem System for Arena

From the results of the six nodes and the seven nodes tandem system, the difference of throughput between the MVA and Arena is around 0.5%. So it is acceptable and the MVA gives an upper bound.

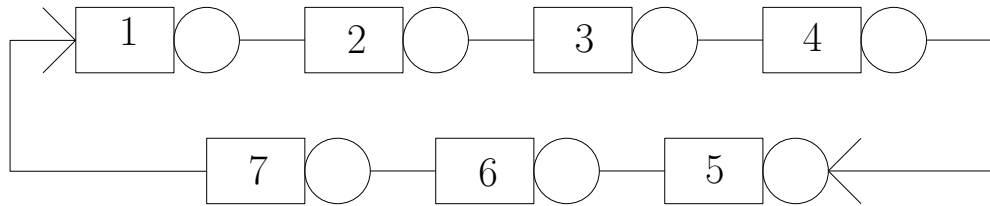| Node	Length	Width of buffer	Buffer Speed	Capacity
1	2.1336	0.3048	692.192	7
2	2.1336	0.3048	692.192	7
3	2.1336	0.3048	692.192	7
4	2.1336	0.3048	692.192	7
5	2.1336	0.3048	692.192	7
6	2.1336	0.3048	692.192	7
7	2.1336	0.3048	692.192	7

$(\mu_1 = 1 \; \mu_2 = 2 \; \mu_3 = 3 \; \mu_4 = 4 \; \mu_5 = 5 \; \mu_6 = 6, \mu_7 = 7, N = 20)$

Arrangement	Throughput From the MVA	Throughput From Arena	%Dev
5 7 1 2 3 4 6	0.997582341	0.9923	0.53%
1 3 5 7 6 4 2	0.997581453	0.9921	0.55%

Arrangement	Cycle Time From the MVA	Cycle Time From Arena	%Dev
5 7 1 2 3 4 6	20.04845	20.1552	0.53%
1 3 5 7 6 4 2	20.04845	20.1592	0.55%

Table 4.37. Seven Node Tandem System from Arena

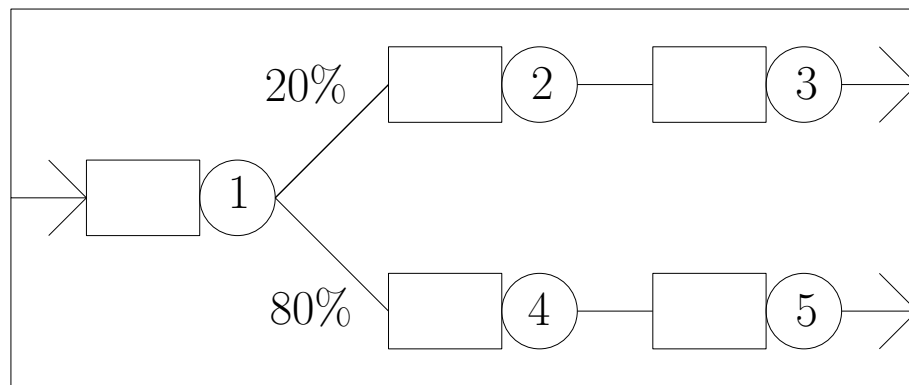### 4.4.3   Five node split topology network(imbalance)

Figure 4.26. Five node split topology network(imbalance) for Arena

In the worst arrangement case, the difference is around 5%. But it is more significant in the best arrangement, which is around 7%

Node	Length	Width of buffer	Buffer speed	Capacity(K)
1	1.524	0.3048	692.192	5
2	1.524	0.3048	692.192	5
3	1.524	0.3048	692.192	5
4	1.524	0.3048	692.192	5
5	1.524	0.3048	692.192	5

$(\mu_1 = 1 \ \mu_2 = 2 \ \mu_3 = 3 \ \mu_4 = 4 \ \mu_5 = 5 \ N=13.)$

Arrangement	Throughput From the MVA	Throughput From Arena	%Dev
5 1 2 3 4	3.4534049	3.2245	7.0%
1 2 3 4 5	0.9963425	0.9911	0.53%

Arrangement	Cycle Time From the MVA	Cycle Time From Arena	%Dev
5 2 1 4 3	3.7973	4.0316	7.1%
1 2 3 4 5	13.0477	13.1167	0.53%

Table 4.38. Five node split topology network(imbalance) from Arena

4.4.4 Seven node split topology network(imbalance)

The network is shown in figure 4.27
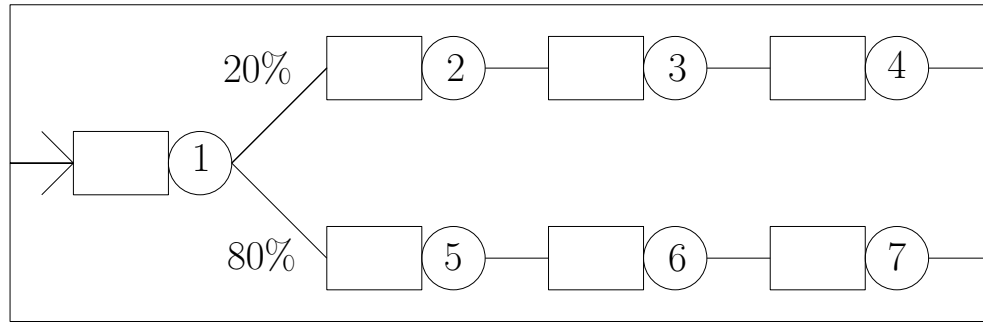

Figure 4.27. Seven node split topology network(imbalance) for Arena

Similar to five node spilt case, the difference in the worst arrangement case is less than 1%, but it is around 7.5% for best arrangement. It may mainly because there is more blocking in the simulation during high speed situation than MVA. The capacity of some nodes may be a little lower than expected.

Node	Length	Width of buffer	Buffer Speed	Capacity
1	1.524	0.3048	24.192	5
2	1.524	0.3048	24.192	5
3	1.524	0.3048	24.192	5
4	1.524	0.3048	24.192	5
5	1.8288	0.3048	24.192	6
6	2.1336	0.3048	24.192	7
7	2.4384	0.3048	24.192	8

$(\mu_1 = 1 \ \mu_2 = 2 \ \mu_3 = 3 \ \mu_4 = 4 \ \mu_5 = 5, \mu_6 = 6, \mu_7 = 7 \ N=20)$

Arrangement	Throughput From the MVA	Throughput From Arena	%Dev
7 1 2 3 4 5	4.47420835	4.1478	7.8%
1 2 3 4 5 6 7	0.99994498	0.9991	0.08%

Arrangement	Cycle Time From the MVA	Cycle Time From Arena	%Dev
7 1 2 3 4 5	4.69357	5.0629	7.8%
1 2 3 4 5 6 7	21.0011	21.0189	0.08%

Table 4.39. Seven node split topology network(imbalance) from Arena

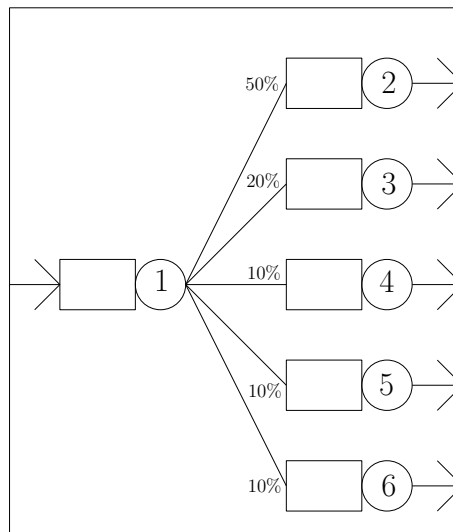## 4.4.5 Six node split topology network(50%+20%+10%+10%+10%)

Figure 4.28. Six node split topology network(imbalance) for Arena

Node	Length	Width of buffer	Buffer Speed	Capacity
1	2.1336	0.3048	24.192	7
2	2.1336	0.3048	24.192	7
3	2.1336	0.3048	24.192	7
4	2.1336	0.3048	24.192	7
5	2.1336	0.3048	24.192	7

Arrangement	Throughput From MVA	Throughput From Arena	%Dev
1 2 3 4 5 6	0.965350926	0.92318	4.37%
6 5 4 3 2 1	5.691473007	5.4676	3.93%

Arrangement	Cycle Time From MVA	Cycle Time From Arena	%Dev
1 2 3 4 5 6	15.5384	16.248	4.37%
6 5 4 3 2 1	2.6355	2.7434	3.93%

Table 4.40. Six node split topology network(imbalance) from Arena

The network is shown in figure 4.28.

This time the difference for the two cases are similar, around 4%. There are two reasons to explain it: first, the capacity of each of the nodes are larger, which would have less blocking, and second, the network in each split chain is much simpler than the previous, which is also results in less blocking.

4.5   Summary and Conclusions

Two optimization methods are employed to help one get the best topological arrangements for closed finite queueing networks: The first one is an exact enumeration algorithm to generate all possible patterns. It works well and the optimal solution can be found, however for larger networks, the time will grow exponentially. (For an m-node system, the number of solutions is $m!$). The second program is the MISQP. It requires fewer iterations so it is both efficient and effective. But the answer from the MISQP will not always guarantee the best optimal solution. It provides an acceptable

sub-optimal solution in shorter computing time. The number of iterations for each method is shown in table 4.41. It can be seen that the advantage of the MISQP is more significant when the system is larger. On the other hand, for small queueing networks, the enumeration method is good enough and the MISQP is not necessary.

System	Number of iterations	
	Enumeration Algorithm	The MISQP
3 node tandem	3!=6	20
4 node tandem	4!=24	46
5 node tandem	5!=120	112
6 node tandem	6!=720	234

Table 4.41. Number of iterations for each system

From the experimental results, it can be concluded that the best arrangement usually follows the "bowl phenomenon", even though it does not give the best arrangement all the time, the difference between a sub-optimal arrangement and best is very small. Also, tandem systems are less sensitive or more robust than the split/merge topologies systems. The MISQP works better on split/merge topologies than the tandem since the objective function has a more significant change between each alternative considered.

Based on the experimental results, we developed two rules to help one find the best arrangement in split/merge topology system.

- I: Place the best nodes at the merge position.

- II: Place the worse nodes along the chain whose arrival probability is lower.

Also many experiments were carried out to prove them and we found that they work well in many of split/merge topologies systems.

Networks which include the M/G/c/c nodes were also studied. They are treated as a big buffer and we found the design rules for different topologies work well for the M/G/c/c system.

To verify the MVA for closed networks, Arena has been used to simulate the throughput and cycle time of our networks. The difference is often acceptable and our analysis has verified it.

For future research, we can concentrate on systems whose nodes are hard to be compared by each others. For example, given in the three node split experiment, there is a trade-off relationship between capacity and service rate, and instead of simply judging the node by these two coefficients, a method is needed to determine which node is better. After evaluating the nodes, we can use the design rules reliably. Finally, more complicated experiments can be considered where service times other than exponential occur for a queue's service time distribution, and multiple servers exist at the nodes.

APPENDIX

COMPUTER PROGRAM FOR ENUMERATION ALGORITHM

```
INTEGER a

Print *,'Please type the number of nodes'

READ *, a

call pattern(a)

END

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

SUBROUTINE pattern(a)


integer a

INTEGER matrix(2*a,4)

INTEGER input(1,2*a)

integer n m j k num1 t

integer pattern(1,a)

integer patternnew(1,a)

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

c Let the initial matrix is [0]

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

DO i=1,2*a

DO j=1,4

matrix(i,j)=0

End DO
```

End DO

!Let the initial pattern is [1,2,3...a]

Do n=1,a

pattern(1,n)=n

END DO

f=1

!Write the first pattern on the file.

OPEN(13,FILE='matrix.txt')

WRITE(13,100),((pattern(i,j),j=1,a),i=1,1)

!get the first transform pattern

DO n=1,a

input(1,2*n-1)=2*pattern(1,n)-1

input(1,2*n)=2*pattern(1,n)

END DO

c WRITE(13,100) ,((input(i,j),j=1,2*a),i=1,1)

****

!get the first transform matrix

****

DO n=1,2*a

matrix(n,2)=1

matrix(n,4)=1

matrix(n,1)=input(1,n)

if (n==2*a) then

matrix(n,3)=input(1,1)

ELSE

matrix(n,3)=input(1,n+1)

END IF

end do

    c WRITE(13,100) ,((matrix(i,j),j=1,4),i=1,2*a)

matrix=0

    c print *,pattern

Do n=1,a

m=n

f=f*m

end do

c General the pattern

Do n=1,f-1

DO m=1,a-1

if (pattern(1,a-m)¡pattern(1,a+1-m)) then

j=a-m

exit

end if

END DO

if (j+1==a) then

k=j+1

    ELSE

DO m=j+1,a-1

if(pattern(1,m)¿pattern(1,j)) then

num1=pattern(1,m)

k=m

end if

```
if(pattern(1,m+1)<num1.and.pattern(1,m+1)>pattern(1,j)) then

k=m+1

end if

End Do

END IF

patternnew=pattern

patternnew(1,j)=pattern(1,k)

patternnew(1,k)=pattern(1,j)

if(j==a-1) then

continue

else

DO m=j+1,a-1

t=m-j

if (m>a+1-t) then

exit

end if

num1=patternnew(1,m)

patternnew(1,m)=patternnew(1,a+1-t)

patternnew(1,a+1-t)=num1

end do

end if

write(13,100),((patternnew(i,j),j=1,a),i=1,1)


   DO m=1,a

input(1,2*m-1)=2*patternnew(1,m)-1

input(1,2*m)=2*patternnew(1,m)

END DO
```

```fortran
c WRITE(13,100) ,((input(i,j),j=1,2*a),i=1,1)
DO m=1,2*a
matrix(m,2)=1
matrix(m,4)=1
matrix(m,1)=input(1,m)
if (m==2*a) then
matrix(m,3)=input(1,1)
ELSE
matrix(m,3)=input(1,m+1)
END IF
end do
c WRITE(13,100) ,((matrix(i,j),j=1,4),i=1,2*a)
pattern=patternnew
matrix=0

    end do
100 Format(7I3)
end
```

BIBLIOGRAPHY

[1] Akyildiz, I.F. Mean value analysis for blocking queueing networks. *Software Engineering, IEEE Transactions* (1988).

[2] Betsy S. Greenberg, Ronald W. Wolff. Optimal order of servers for tandem queues in light traffic. *Journal of Applied Probability* (1988).

[3] Bolch, G., S.Greinner H.de Meer, and Trivedi, K.S. *Queueing Network and Markov chains.* Wiley: New York.

[4] E., Koenigsberg. Cyclic queues. *Oper Res 9* (1958).

[5] Exler, O., and K.Schittkowski. A trust region sqp algorithm for mixed-integer nolinear programming. *Optimization letters* (2007), 269–280.

[6] Exler, O. T.Lehmann, and K.Schittkowski. Misqp:a fortran subroutine of a trust region sqp algorithm for mixed-integer nonlinear programming-user guide. *Optimization letters* (2007), 269–280.

[7] Forest Baskett, K. Mani Chandy, Richard R. Muntz Fernando G. Palacios. Open closed and mixed networks of queues with different classes of customers. *Journal of the ACM (JACM)* (1975).

[8] Genji Yamazaki, Hirotaka Sakasegawa, J. George Shanthikumar. On optimal arrangement of stations in a tandem queueing system with blocking. *Management Science* (1992).

[9] G.Kulkarni, Vidyadhar. *Modeling and Analysis of Stochastic Systems.* CRC, 2009.

[10] Gordon, William J., and Newell, Gordon F. Closed queuing systems with exponential servers. *Operations Research* (1967).

[11] Gordon, William J., and Newell, Gordon F. Cyclic queuing systems with restricted length queues. *Operations Research* (1967).

[12] Gross, D., and Harris, C. *Fundamentals of Queueing Theory.* Wiley, 1985.

[13] H., Perros. *Queueing Networks with Blocking.* Oxford, 1994.

[14] Horowitz, E., and S.Sahni. *Fundamentals of Computer Algorithms.* Computer Science Press, Inc, 1979.

[15] J. George Shanthikumar, David D. Yao. Optimal server allocation in a system of multi-server stations. *Management Science* (1987).

[16] J. MacGregor Smith, Laoucine Kerbache. State dependent models of material handing systems in closed queueing networks. *International Journal of Production Research* (2012).

[17] Kelton, D., R.Sadowski, and Zupick, Nancy B. *Simulation with Arena. Fifth edition.* McGraw-Hill, 2010.

[18] Kendall, D. G. Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded markov chain. *The Annals of Mathematical Statistics 24* (1953).

[19] Kerbache, L., and Smith, J.MacGregor. Asymptotic behavior of the expansion method for open finite queueing network. *Computers and Operations Research* (1988).

[20] Lehtonen, Tapani. On the ordering of tandem queues with exponential servers. *Journal of Applied Probability* (1986).

[21] Li Zhuang, John A. Buzacott, Xiao Gao Liu. Approximate mean value performance analysis of cyclic queueing networks with production blocking. *Oper Res* (1994).

[22] M. Reiser, S. S. Lavenberg. Mean-value analysis of closed multichain queuing networks. *Journal of the ACM (JACM)* (1980).

[23] Onvural, Raif. Survey of closed queueing networks with blocking. *ACM Computing surveys* (1990).

[24] Prince Bedell, J. MacGregor Smith. Topological arrangements of m/g/c/k, m/g/c/c queues in transportation and material handling systems. *Computers and Operations Research* (2012).

[25] S. Suresh, W. Whitt. Arranging queues in series: A simulation experiment. *Management Science* (1990).

[26] Smith, J.MacGregor. Robustness of state dependent queues and material handling systems. *International Journal of Production research* (2010).

[27] Smith, J.MacGregor. Queue decomposition and finite closed queueing network models. *Computers and Operations Research* (2015).

[28] Van Dijk, N. M. On the arrival theorem for communication networks. *Computer Networks and ISDN Systems 25* (1993).

[29] Whitt, Ward. The best order for queues in series. *Management Science* (1985).

[30] Wolff, Shantanu V. Tembe Ronald W. The optimal order of service in tandem queues. *Operations Research* (1974).