Buffer Space Allocation in General Closed Finite Queuing Networks

J. MacGregor Smith
* email: jmsmith@ecs.umass.edu

Laoucine Kerbache †
email:kerbache@hec.fr

February 25, 2001

Abstract — **This paper studies the buffer space allocation problem in closed, finite queuing networks with the objective of optimizing a nonlinear cost function of system throughput, cycle time, and number of buffer spaces. The problem is highly complex, since no known closed form expression exists relating the objective function and the design variables and system parameters. A new closed form expression for the blocking probability in general service finite queueing networks is presented. The performance measures are evaluated numerically using a methodology based on the Expansion Method and Mean Value Analysis, and the solution is incorporated into a nonlinear optimization scheme to arrive at the sub-optimal buffer space vector. Discussions of applicability, convergence, and computational analysis of the procedure are presented, as well as comparisons against simulation results.**

Keywords — Closed Networks, Optimization, Buffer Allocation

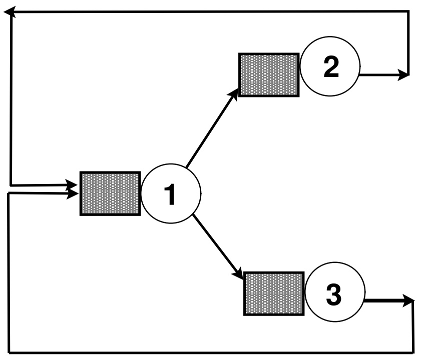

Figure 1: Split Topology in Closed Network

1 Introduction

In most manufacturing and service systems, the presence of finite buffers at the work stations of the system interrupts the flow of products through the system. Often when products seek buffer space from adjacent work centers, they will not be able to capture the buffer space because the buffer is full. In addition, when the service time distributions are generally distributed, then very difficult problems arise for the analysis of these systems. Any way or methodology in which the performance of these manufacturing and service systems can be modelled and optimized would be a significant contribution to the design and analysis of these systems.

1.1 Motivation

To illustrate the complexity of the above problems, let us examine Figure 1 which represents a small closed network where N products or customers circulate between the three queues. Nodes 2 and 3 can alternate in blocking node 1. In fact, node 1 can also block both nodes 2 and 3 if its buffer is full. Finally, because of the finite population N, there is a complex relationship between the throughput of this system and the other performance measures *i.e.*, cycle time, utilization, etc. How to allocate the buffers and at the same time optimize N is the subject of this paper.

1.2 Outline of Paper

In §2 of the paper we provide necessary background and in §3 we review the underlying mathematical models of the problem for M/G/1/K systems and our approximation approach. In §4, we review the performance algorithm and its coupling with the optimization procedure to allocate the buffers. In §5, we demonstrate our approach on series, merge and splitting systems and in §6 we summarize our results.

2 Background

Most of the literature in queueing networks is concerned with networks with product form solutions. Finite open and closed queueing networks generally do not have product form solutions because the blocking in the network creates strong dependencies. In general, the literature on finite closed queueing network models with blocking

*Department of Mechanical and Industrial Engineering, University of Massachusetts Amherst Massachusetts 01003
†HEC School of Management, Paris, France

can generally be broken down into *i). performance modelling tools;* and *ii) optimization tools.* The objective of our paper is ambitious since we aim at uniting both these aspects.

The Expansion method which is a performance modelling tool will be coupled with Mean Value Analysis(MVA) and a search optimization procedure to optimally allocate buffers in closed networks with general service. What makes this possible is that we have arrived at a closed form solution of the blocking probability in finite networks which is quite accurate and robust as we will demonstrate.

2.1 Literature Review

There are few papers dealing with the analysis of finite closed queueing networks with general service times [1, 3], notwithstanding the optimization of the buffers and the population of these systems. One of the main reasons for the paucity of these results is the deadlock phenomenon explained below.

2.2 Deadlock

One of the unique features of closed finite queueing networks which makes them different from open finite queueing networks is the deadlock phenomenon. Figure 2 illustrates the deadlock phenomenon which links the clockwise flow of the population of the closed network with the total number of buffers allocated within the network. This deadlock phenomenon is a problematic feature of the closed model but as it turns out an important aspect of the optimization of these networks, since it will often provide a lower bound on an optimal allocation of buffers in a network.

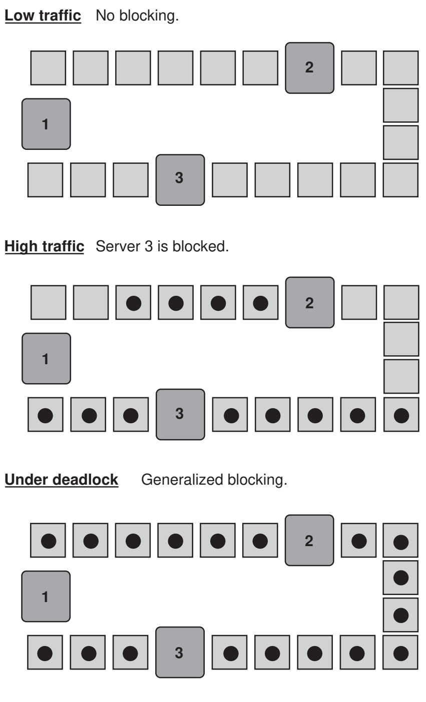

Figure 2: Deadlock Phenomenon

3 Mathematical Models

In this section of the paper we provide a set of mathematical models which will help guide us in our optimization process.

3.1 Notation

The following section presents some of the notation we need for the paper.

a^2 := squared coefficient of variation of the arrival process.

Λ := External Poisson arrival rate to the network.

μ_j := Exponential mean service rate at node j.

$\epsilon \in (0,1)$ threshold for the blocking probability.

$\rho = \frac{\lambda}{\mu}$:= the traffic intensity

K_j := Buffer capacity at node j including those in service

p_K := Blocking probability of finite queue of size K.

s^2 := squared coefficient of variation of the service process.

Θ := Mean throughput rate.

3.2 M/M/1/K Systems

The blocking probability for an $M/M/1/K$ system with $\rho < 1$ is well-known:

$$\frac{(1-\rho)\,\rho^K}{1-\rho^{K+1}} = p_K$$

If we relax the integrality of K, we can express K in terms of ρ and p_K and arrive at a closed-form expression for the buffer size which is the largest integer as follows:

$$K = \left\lceil \frac{\ln(\frac{p_K}{1-\rho+p_K\rho})}{\ln(\rho)} \right\rceil$$

This is a very useful formula for determining the buffer size K of an individual queue.

3.3 Non-Markovian Systems

In principle, for the $M/G/1/K$ system, one could develop the blocking probability for fixed K and all the other probabilities, but this would be tedious and not as useful when it comes to modelling the design of queueing networks with varying buffer sizes, see Chapter 5 §5.1.8 in Gross and Harris [8].

3.4 Different approximations

There are numerous other approximations for the blocking probability possible for $M/G/1/K$ systems. One survey article by Springer and Makens [25] analyzes in some detail five approximation formula and concludes that the formula by Gelenbe is the most accurate and robust. In the following analysis, we shall examine this closed-form approximation formula, and, as in a previous paper [22], examine the nature of the blocking probability. There are other buffer allocation approximations [13, 14, 29, 26] that could be analyzed, and we will discuss in particular the approximations of Tijms and Kimura. The Tijms and Kimura's approximations are not closed-form but are very accurate.

3.5 Gelenbe's Formula

Gelenbe's formula is based on approximating the discrete queueing process as a continuous diffusion process. The blocking probability from Gelenbe's equation with squared arrival and service process coefficient of variation is given by the following equation [6] where a^2 and s^2 are respectively the squared coefficient of variations of the arrival and service processes:

$$\frac{\lambda\,(\mu-\lambda)\,e^{-2\,\frac{(\mu-\lambda)(k-1)}{\lambda\,a^2+\mu s^2}}}{\left(\mu^2-\lambda^2 e^{-2\,\frac{(\mu-\lambda)(k-1)}{\lambda\,a^2+\mu s^2}}\right)}=p_K$$

The closed-form expression derivable for K from Gelenbe's formula is the following:

$$K=1/2\,\frac{2\,\lambda-2\,\mu+\ln(\frac{pk\,\mu^2}{\lambda\,(-\lambda+\mu+pk\,\lambda)})\lambda\,a^2+\ln(\frac{pk\,\mu^2}{\lambda\,(-\lambda+\mu+pk\,\lambda)})\mu\,s^2}{(\lambda-\mu)}$$

In the full paper, we will compare our approximation with Gelenbe's.

3.6 Two-moment Approximation Schema

Tijm's two-moment approximation [4, 26, 28] relies on a weighted combination of an exact (if available) expression of the $M/D/1/K$ blocking probability as well as the blocking probability of the $M/M/1/K$ formula.

$$K_\epsilon^T(s^2)=s^2K_\epsilon(M)+(1-s^2)K_\epsilon(D)$$

Of course, if exact expressions are available for both formulas, then Tijm's approximation is exact for the two extreme cases.

Kimura, on the other hand, has also a two-moment approximation that turns out to be a little simpler and is the one we shall build upon since it utilizes Markovian approximations as its basis. His expression is:

$$\tilde{K}_\epsilon(s^2)=K_\epsilon(M)+NINT(1/2(s^2-1)\sqrt{\rho}K_\epsilon(M))$$

where $NINT$ is the nearest integer. An important observation about Kimura's expression is that his formula estimates the pure buffer without the space for the customers in service, while Tijm's formula is more general and includes those in service.

Rather than simply use Kimura's expression, we shall substitute the formula for the optimal buffer size of the $M/M/1/K/$ system and relax the integrality of K to yield the following closed-form expression for the optimal buffer size for $M/G/1/K$ systems.

Let's repeat for clarity the formula for the optimal buffer size for the M/M/1/K formula.

$$K=\frac{\ln(\frac{pk}{1-\rho+pk\,\rho})}{\ln(\rho)}$$

Here, we add Kimura's expression for the approximation of the optimal buffer size based on his two-moment approximation formula. It is important to note here that we subtract the space for the server in each of the two terms of Kimura's formula in order to estimate the true buffer space in the queue:

$$(\ln(\frac{pk}{1-\rho+pk\,\rho})\,(\ln(\rho))^{-1}-1)$$
$$+1/2\,\left(s^2-1\right)\sqrt{\rho}\left(\ln(\frac{pk}{1-\rho+pk\,\rho})\,(\ln(\rho))^{-1}-1\right)$$

Now we factor the terms of the above expression to give the following simplified expression for the optimal buffer size in M/G/1/K formulas:

$$\frac{\left(\ln(\frac{pk}{1-\rho+pk\,\rho})-\ln(\rho)\right)\,(2+\sqrt{\rho}s^2-\sqrt{\rho})}{2\ln(\rho)}$$

If $s^2=1$, then the formula yields the same expression as for the $c=1$ formula, when we subtract the space for the server.

As an added side benefit for developing the closed form expression for the optimal buffer, if we invert the last expression we can obtain the blocking probability for the M/G/1/K system as:

$$p_K=\frac{\rho^{\left(\frac{2+\sqrt{\rho}s^2-\sqrt{\rho}+2\,K}{2+\sqrt{\rho}s^2-\sqrt{\rho}}\right)}(-1+\rho)}{\left(\rho^{\left(2\,\frac{2+\sqrt{\rho}s^2-\sqrt{\rho}+K}{2+\sqrt{\rho}s^2-\sqrt{\rho}}\right)}-1\right)}$$

The closed form expression is fairly accurate and computational results are included in [24]. While it could be argued that a $GI/G/1/K$ formula is really needed, we have not developed one at this time based upon our approach.

3.7 Optimization Problem

The primal optimization problem with $M/M/1/K$ and $M/G/1/K$ systems that will be examined here is essentially the following:

$$Maximize\ f(\mathbf{x})=C_\theta\theta-C_W W-C_B\sum_{i=1}^{Q}x_i$$

s.t.

$$\sum_{i=1}^{Q} x_i \geq N+1 \tag{1}$$

$$x_i \in \{0,1,2,\ldots\} \forall i \tag{2}$$

in which

- *the x_i is the buffer space of the $i-th$ queue system.*
- C_θ := *the profit of the throughput*
- C_W := *the cost of the cycle time*
- C_B := *the cost of the buffers which is due to the floor space and equipment costs as well as the Work-in-Process (WIP) costs.*

In the particular formulation of the problem, the x_i $\forall i$ are the decision variables under optimization control. While these are essentially integer variables, they can be reasonably approximated by round off from the non-linear programming solver.

In order to couple the optimization problem with the Expansion method, Powell's algorithm is used to search for the optimal buffer vector(s) while the Expansion method computes the performance measures of throughput and cycle time.

4 Algorithms

The basic procedure for solving our optimization problem involves the performance modelling tool of the Expansion Method along with an equalization phase and the Mean Value Analysis (MVA) for computing the performance measures in closed queueing networks. Figure 3 illustrates the general concepts of the algorithmic procedures which were originally developed in [7].

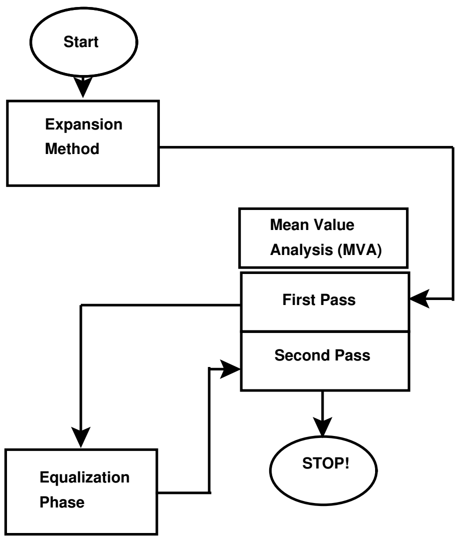

Figure 3: Algorithmic Decomposition Procedure

4.1 Expansion Method

The Expansion Method is a robust and effective approximation technique developed by Kerbache and Smith [12]. As described in previous papers, this method is characterized as a combination of Repeated Trials and Node-by-node Decomposition solution procedures. Methodologies for computing performance measures for a finite queueing network use primarily the following two kinds of blocking:

1. *Type I:* 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 sometimes referred to as Blocking After Service (BAS) [17].
2. *Type II:* 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) [17].

The Expansion Method uses *Type I* blocking, which is prevalent in most production and manufacturing, transportation and other similar systems.

Consider a single node with finite capacity K (including service). This node essentially oscillates between two states - the saturated phase and the unsaturated phase. In the unsaturated phase, node j has at most $K-1$ customers (in service or in the queue). On the other hand, when the node is saturated no more customers can join the queue. Refer to Figure 4 for a graphical representation of the two scenarios.

Expansion Method has the following three stages :

- *Stage I: Network Reconfiguration.*
- *Stage II: Parameter Estimation.*
- *Stage III: Feedback Elimination.*

The following additional notation defined by Kerbache and Smith [12, 11] shall be used in further discussion regarding this methodology :

h := The holding node established in the Expansion method.

$\tilde{\lambda}_j$:= Effective arrival rate to node j.

$\tilde{\mu}_j$:= Effective service rate at node j due to blocking.

p_h:=Feedback blocking probability in the Expansion method.

4.2 Stage I: Network Reconfiguration

Using the concept of two phases at node j, an artificial node h is added for each finite node in the network to register blocked customers. Figure 4 shows the additional delay, caused to customers trying to join the

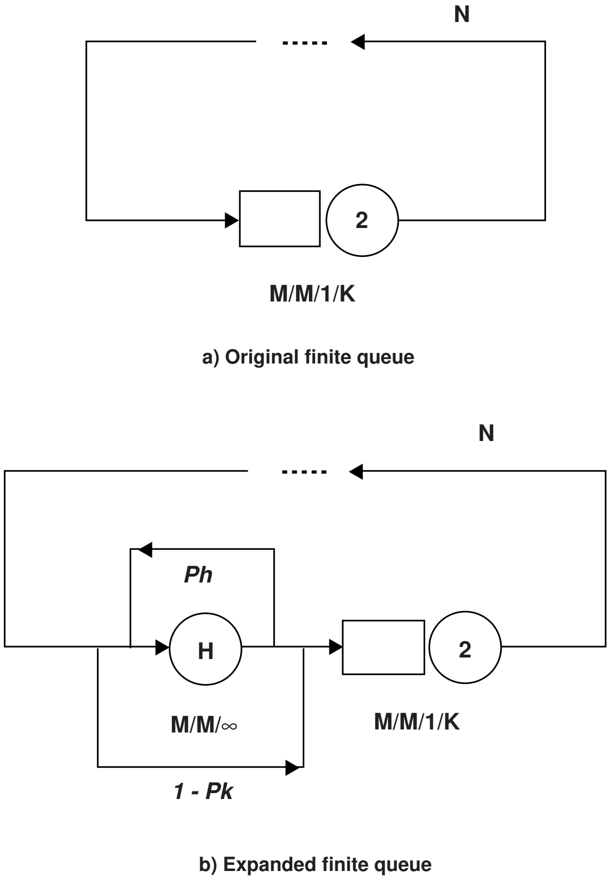

Figure 4: Type I Blocking in Finite Queues

queue at node j when it is full, with probability p_K. The customers successfully join queue j with a probability $(1-p_K)$. Introduction of an artificial node also dictates the addition of new arcs with p_K and $(1-p_K)$ as the routing probabilities.

The blocked customer proceeds to the finite queue with probability $(1-p_h)$ once again after incurring a delay at the artificial node. If the queue is still full, it is re-routed with probability p_h to the artificial node where it incurs another delay. This process continues till it finds a space in the finite queue. A feedback arc is used to model the repeated delays. The artificial node is modeled as an $M/M/\infty$ queue. The infinite number of servers is used simply to serve the blocked customer a delay time without queueing.

4.3 Stage II: Parameter Estimation

This stage essentially estimates the parameters p_K, p_h and μ_h utilizing known results for the $M/M/1/K$ model.

- p_K: Utilizing our analytical results for the $M/G/1/K$ model provides the following expression for p_K.

$$p_K = \frac{\rho^{\left(\frac{2+\sqrt{\rho}s^2-\sqrt{\rho}+2K}{2+\sqrt{\rho}s^2-\sqrt{\rho}}\right)}(-1+\rho)}{\left(\rho^{\left(2\,\frac{2+\sqrt{\rho}s^2-\sqrt{\rho}+K}{2+\sqrt{\rho}s^2-\sqrt{\rho}}\right)}-1\right)}$$

- p_h: Since there is no closed form solution for this quantity an approximation is used given by Labetoulle and Pujolle obtained using diffusion techniques [16].

$$p_h = \left[\frac{\mu_j+\mu_h}{\mu_h} - \frac{\lambda[(r_2^K - r_1^K)-(r_2^{K-1}-r_1^{K-1})]}{\mu_h[(r_2^{K+1}-r_1^{K+1})-(r_2^K-r_1^K)]}\right]^{-1}$$

where r_1 and r_2 are the roots to the polynomial:

$$\lambda - (\lambda+\mu_h+\mu_j)x + \mu_h x^2 = 0$$

while, $\lambda = \lambda_j - \lambda_h(1-p_h)$ and λ_j and λ_h are the actual arrival rates to the finite and artificial holding nodes respectively.

In fact, λ_j the arrival rate to the finite node is given by:

$$\lambda_j = \tilde{\lambda}_i(1-p_K) = \tilde{\lambda}_i - \lambda_h$$

Let's examine the following argument to determine the service time at the artificial node. If an arriving customer is blocked, the queue is full and thus a customer is being serviced, so the arriving customer to the holding node has to remain in service at the artificial holding node for the remaining service time interval of the customer in service. The delay distribution of a blocked customer at the holding node has the same distribution as the remaining service time of the customer being serviced at the node doing the blocking. Using renewal theory Kleinrock [15], showed that the remaining service time distribution has the following rate μ_h:

$$\mu_h = \frac{2\mu_j}{1+\sigma^2\mu_j^2}$$

where, σ^2 is the service time variance.

4.4 Stage III: Feedback Elimination

Due to the feedback loop around the holding node, there are strong dependencies in the arrival processes. We remove the feedback to get rid of these dependencies.

The probabilities of being in any of the two phases (saturated or unsaturated) are p_K and $(1-p_K)$. The mean service time at a node i preceding the finite node is μ_i^{-1} when in the unsaturated phase and $(\mu_i^{-1}+\mu_h^{-1})$ in the saturated phase. Thus, on an average, the mean service time at the node i preceding a finite node is given by

$$\tilde{\mu_i^{-1}} = \mu_i^{-1} + p_K\mu_h^{-1}$$

Similar equations can be established with respect to each of the finite nodes. Ultimately, we have simultaneous non-linear equations in variables p_K, p_h, μ_h^{-1} along with auxiliary variables such as μ_j and $\tilde{\lambda}_i$. Solving these equations simultaneously we can compute all the performance measures of the network.

$$
\begin{aligned}
\lambda &= \lambda_j - \lambda_h(1-p_h) && (3)\\
\lambda_j &= \tilde{\lambda_i}(1-p_K) && (4)\\
\lambda_j &= \tilde{\lambda_i} - \lambda_h && (5)\\
p_h &= \left[\frac{\mu_j+\mu_h}{\mu_h} - \frac{\lambda[(r_2^K - r_1^K)-(r_2^{K-1}-r_1^{K-1})]}{\mu_h[(r_2^{K+1}-r_1^{K+1})-(r_2^K-r_1^K)]}\right]^{-1} && (6)\\
z &= (\lambda+2\mu_h)^2 - 4\lambda\mu_h && (7)\\
r_1 &= \frac{[(\lambda+2\mu_h)-z^{\frac{1}{2}}]}{2\mu_h} && (8)\\
r_2 &= \frac{[(\lambda+2\mu_h)+z^{\frac{1}{2}}]}{2\mu_h} && (9)\\
p_K &= \frac{\rho^{\left(\frac{2+\sqrt{\rho}s^2-\sqrt{\rho}+2K}{2+\sqrt{\rho}s^2-\sqrt{\rho}}\right)}(-1+\rho)}{\left(\rho^{\left(2\,\frac{2+\sqrt{\rho}s^2-\sqrt{\rho}+K}{2+\sqrt{\rho}s^2-\sqrt{\rho}}\right)}-1\right)} && (10)
\end{aligned}
$$

Equations 3 to 6 are related to the arrivals and feedback in the *holding* node. The equations 7 to 9 are used for solving equation 6 with z used as a dummy parameter for simplicity of the solution. Lastly, equation 10 gives the approximation to the blocking probability for the $M/G/1/K$ queue. Hence, we essentially have five equations to solve, *viz.* 3 to 6 and 10.

To recapitulate, we first expand the network; followed by approximation of the routing probabilities, due to blocking, and the service delay in the holding node and finally the feedback arc at the holding node is eliminated. Once these three stages are complete, we have an expanded network which can then be used to compute the performance measures for the original network. As a decomposition technique this approach allows successive addition of a holding node for every finite node, estimation of the parameters and subsequent elimination of the holding node.

Figure 5 and 6 illustrate the process of expanding the network topologies for the split and merge topologies in the Expansion method. An important point about this process is that we do not physically modify the networks, only represent the expansion process through the software.

4.5 Mean Value Analysis

We shall not go into great detail describing the MVA algorithm other than to say that it is due to the Expansion method's alteration of the mean service rate that allows the MVA to calculate the performance measures for the general network. For more details regarding the MVA algorithm, please see [20]. Thus, when the holding nodes' service rate is affected by the squared coefficient of variation in the service time, that change is thus reflected in the MVA. Thus, no real change is made to the MVA itself other than what the Expansion method passes to the MVA.

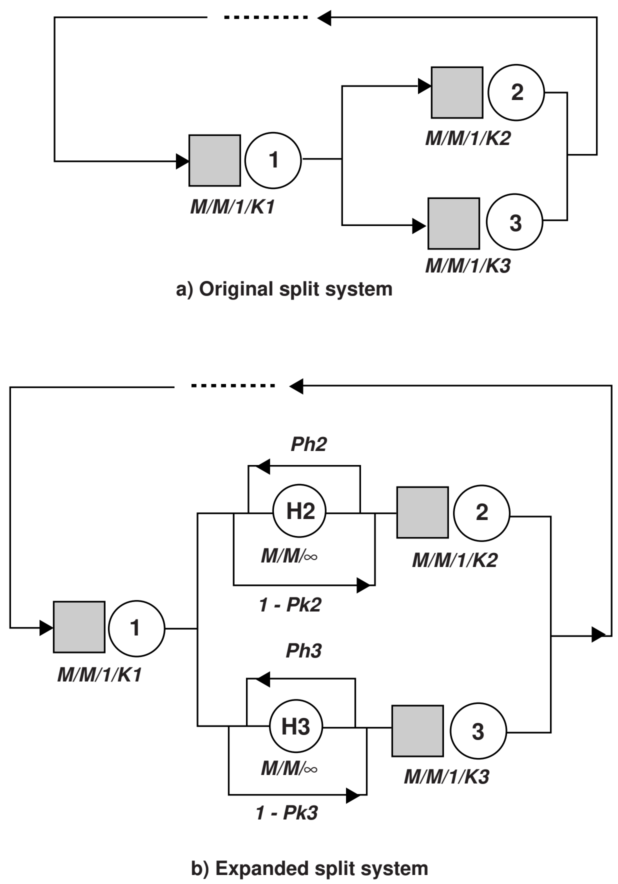

Figure 5: Split Topologies

4.6 Powell's Algorithm

The heart of the optimization procedure is Powell' unconstrained derivative free optimization method. Powell's method, as presented in (Himmelblau, 1972)[10], locates the minimum of $f(\mathbf{x})$ of a non-linear function by successive unidimensional searches from an initial starting point $\mathbf{x_0}^{(k)}$ along a set of conjugate directions. These conjugate directions are generated within the procedure itself. Powell's method is based on the idea that if a minimum of a non-linear function $f(\mathbf{x})$ is found along p conjugate directions in a stage of the search, and an appropriate step is made in each direction, the overall step from the beginning to the p^{th} step is conjugate to all of the p subdirections of the search. Figure 7 illustrates the steps of the optimization algorithm to simultaneously find the optimal buffers and N^*.

We have had remarkable success in the past in coupling Powell's algorithm with the Expansion method. Let us examine how well it does with the addition of the new general blocking probability formula.

5 Experimental Results

We will focus on networks where the $s^2 = \{\frac{1}{2}, 1, \frac{3}{2}\}$, although the algorithm is not restricted to such a constraint. In the full paper, we will have additional computational results. We will determine the optimal buffer

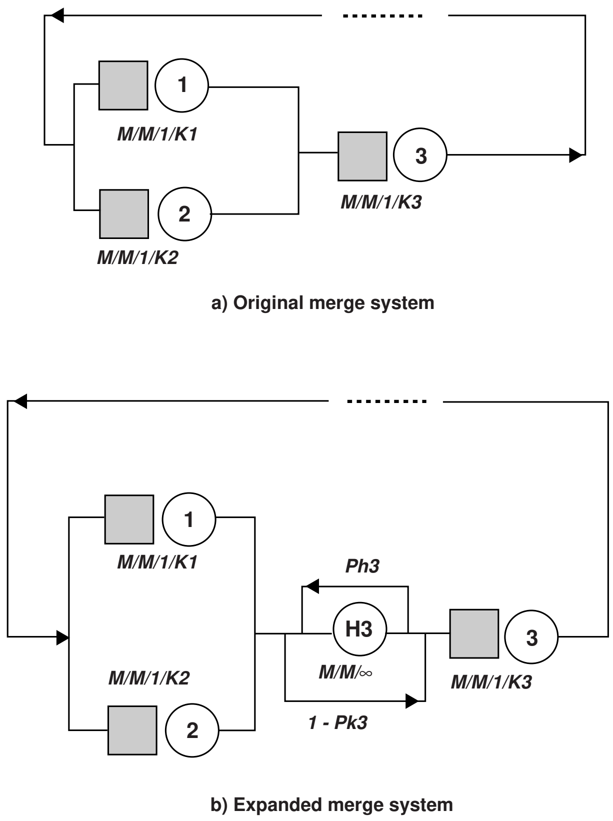

Figure 6: Merge Topologies

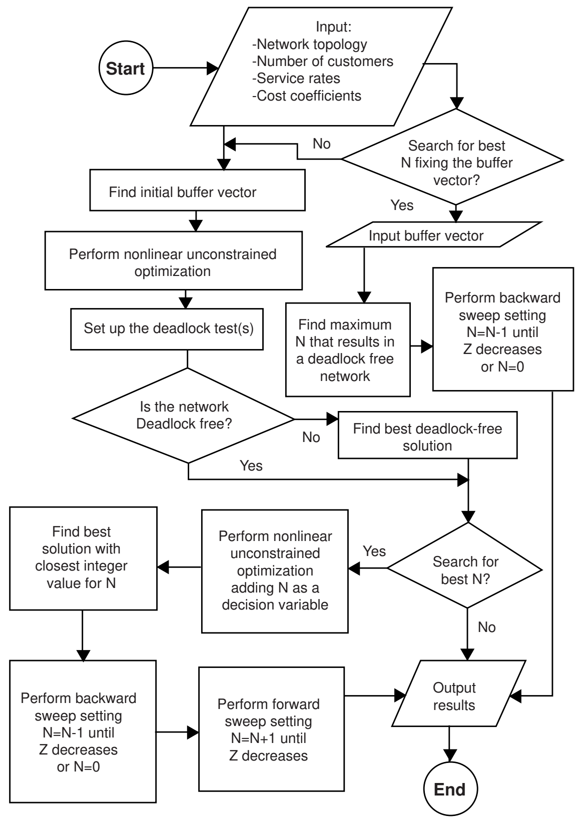

Figure 7: General Algorithm Flowchart

N	θ	W	Z	x
1	0.78	1.28	4.63	(1,1,1,1,1,1,1,1,1,1)
4	2.26	1.77	51.49	(1,1,1,1,1,1,1,1,1,1)
7	3.19	2.19	73.80	(1,2,2,2,2,1,2,1,2,1)
11	3.95	2.79	89.24	(2,2,3,2,2,2,2,2,2,2)
15	4.46	3.36	84.50	(3,3,4,3,2,2,3,3,3,2)
17	4.64	3.66	94.81	(3,3,4,3,3,3,3,3,3,3)
21	4.89	4.29	92.29	(4,3,4,4,3,4,3,4,4,3)

Table 1: 10-node Mixed network, $s^2 = 1.0$

allocation and also at the same time the optimal population N^* for the particular configuration of queues. For most of the following experiments in estimating Z, we have set $C_\theta := 30$ the profit of the throughput; $C_W := 10$ the cost of the cycle time; $C_B := 1$ the cost of the buffers. For all the simulation runs, 50,000 time units were used, with the first 2,000 time units were discarded and 10 replications revealed a small standard deviation in the performance measures.

Figure 8 and Table 1 illustrates a general mixed topology network which can be modelled with our methods. This is the type of layout planning problem our methodology is especially suitable for. The results show the range of buffer allocations as a function of the population in the network. This is a very complex network modelling problem. In the example in Figure 8, $N^* = 17$ was found and as we shall demonstrate, it is the effectiveness of this algorithm which should yield a near optimal solution to this rather complex topology with unequal service rates.

If one inspects Table 1, node #3 turns out to be the bottleneck since as N increases node #3 is always added a buffer as the total buffer increases. This is another important feature of our tool, that it will quickly point out the bottleneck.

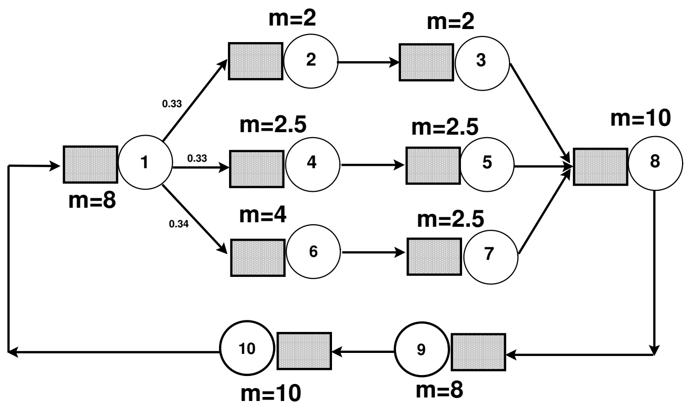

Figure 8: Mixed Network Topology, m:= service rate

In Table 2, we present an example of a 3-node tandem network with identical exponential service times. From the tabular results, it is apparent that the throughput, cycle time and objective function curves of the analytical and simulation models roughly parallel each other. The simulation model results for these configurations in Table 2 are expected to be near the optimal results for this system with the given parameters. It appears that the optimal number of customers in the system is somewhere around $N^* = 6$. For $N^* = 6$ the analytical model has a $Z = 51.7474, \theta = 2.6992, W = 2.2229$ with an optimal buffer allocation of $x^* = (3, 3, 3)$. In fact, for $x^* = (4, 4, 4)$, with $N = 7$, the simulation model finds an optimal objective function value of $Z = 60.50$. Thus, the analytical model was close but did not agree with the N^* for the simulation model. Notice also that for the $N = 12, 16$ the analytical model found a deadlock solution for the buffer allocation. If one examines the details of the outputs for both the simulation and analytical models (not included here for the sake of brevity), one sees that for around 12 customers in the system, the utilization of the stations $\rho_i \rightarrow 1.0$ so the buffer allocation is most difficult to compute at this stage.

In Table 3, we model the tandem system with uniform service rates and $s^2 = 0.50$. We used a Gamma distribution with parameters $\alpha = 2.00; \beta = 0.125$ to model the service times. Similar results with regards to the throughput, cycle time, and objective function curves occur. We see that the optimal solution for the number of customers is around $N^* = 7$, and, in fact, for the analytical mode, we achieve with an expected buffer $x^* = (3, 3, 3), Z = 59.1959, \theta = 3.0140, W = 2.3225$ while with the simulation mode we also have $N^* = 7, x^* = (3, 3, 3), Z = 71.30, \theta = 3.3700, W = 2.0800$.

Finally, in Table 4, we model the tandem system with uniform service rates and $s^2 = 1.50$. Here, we used a Gamma distribution with parameters $\alpha = 0.667; \beta = 0.375$ to model the service times. Here, we see that the optimal solution for the number of customers is around $N^* = 5$, and, in fact, for the analytical mode, we achieve with an expected buffer $x^* = (2, 2, 2), Z = 43.0344, \theta = 2.3.453, W = 2.1319$ while with the simulation mode we also have $N^* = 5, x^* = (2, 2, 2), Z = 52.40, \theta = 2.5900, W = 1.9300$.

In Tables 5, 6, and 7 we have a 5-node tandem topology with uniform service rates. The results are similar to those of the 3-node tandem systems.

In Table 8, we have 5-node tandem network with variable service times, $s^2 = 1$. The buffer allocation again changes with N and $N^* = 6$ customers in the analytical model with $Z = 36.82, \theta = 2.32, W = 2.58$ with an optimal buffer allocation of $x^* = (1, 1, 1, 3, 1)$ and the simulation model also has $N^* = 6$ with $Z = 44.40, \theta = 2.51, W = 2.39$. Notice that the bottleneck node #4 attracts the most buffers as N varies.

Table 9 illustrates an arbitrary eight-node tandem network with $s^2 = 1$. Here we varied the cost parameters with $C_\theta = 100, C_W = 8.0, C_B = 1.0$. Only a sample of the population parameter variations are presented. It is surprising that the network is optimized for such a low population. In Table 9 we see that around 6 customers appears to be optimal. Unfortunately, in the analytical model $N^* = 7, Z = 41.63, \theta = 1.05, W = 6.67, x^* = (1, 1, 1, 1, 2, 1, 2, 1)$ but in the simulation model, $N^* = 6, Z = 43.56, \theta = 1.01, W = 5.93, x^* = (1, 1, 1, 1, 2, 1, 2, 1)$. However, the general comparison of the performance measures comparing the analytical and simulation model for this 8-node network are extremely accurate. In fact, one could argue that the larger the network, the better our approximation should be. More experiments will follow.

6 Summary and Conclusions

In this paper we have presented an approximation approach concerned with finite closed queueing networks with general service times. Optimization of these finite systems has been the main concern. An analytical performance modelling tool was coupled with a search methodology optimization program to search for the optimal buffer patterns and also the optimal population N for these closed network topologies of series, merge, and split topologies. Preliminary results indicate that the methodology is an effective procedure for allocating buffers and determining N^*, especially for larger networks.

References

[1] Akyildiz, I.F., 1988. “General Closed Queueing Networks with Blocking,” in P.J. Coutois and G. Latouche (eds.) *Performance 87*, North-Holland, Amsterdam, 282-303.

[2] Baker, K.R., S.G. Powell. and D.F. Pyke, 1990. “Buffered and Unbuffered Assembly Systmes with Variable Processing Times,” *J. Mfg. Oper. Mgmt.* **3**, 200-223.

[3] Bouhchouch, A., Y. Frein, and Y. Dallery, 1996. “Performance Evaluation of closed tandem queueing networks with finite buffers,” *Performance Evaluation* **26**, 115-132.

[4] De Kok, A.G. and H. Tijms, 1985. “A Two-moment approximation for a Buffer Design Problem Requiring a Small Rejection Probability,” *Performance Evaluation***5**, 77-84.

[5] Garey,M. and D. Johnson, 1979. Computers and Intractability Freeman: San Francisco.

[6] Gelenbe, E., 1975. “On approximate Computer System Models,” *JACM* **22** (2), 261-269.

[7] Gonzalez, Edgar, 1997. “Optimal Resource Allocation in Closed Finite Queueing Networks with Blocking after Service.” Ph.D. Dissertation. University of Massachusetts, Amherst, MA. 01003.

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

[9] Harris, J.H. and S.G. and Powell, 1999. “An algorithm for optimal buffer placement in reliable serial lines.” *IIE Transactions,* **31**, 287-302.

N	θ			W			Z			**x**	
	$\mathcal{A}$	S	%	$\mathcal{A}$	S	%	$\mathcal{A}$	S	%	$\mathcal{A}$	S
1	1.25	1.33	-6.46	0.80	0.75	6.91	26.40	29.50	-10.52	(1,1,1)	(1,1,1)
4	2.51	2.66	-5.87	1.60	1.50	6.23	50.19	55.86	-10.15	(3,3,3)	(3,3,3)
8	2.92	3.15	-7.30	2.74	2.54	7.87	51.05	57.10	-10.60	(3,3,3)	(4,4,4)
12	3.17	3.39	-6.49	3.78	3.54	6.78	44.43	45.30	-1.92	(4,4,5)	(7,7,7)
16	3.33	3.45	-3.48	4.80	4.64	3.45	34.98	33.10	5.68	(6,5,6)	(8,8,8)

Table 2: 3-node Tandem Network with $\mu_i = 4, s^2 = 1.0, \forall i$

N	θ			W			Z			**x**	
	$\mathcal{A}$	S	%	$\mathcal{A}$	S	%	$\mathcal{A}$	S	%	$\mathcal{A}$	S
1	1.30	1.33	-2.26	0.77	0.75	2.60	28.20	29.50	-4.41	(1,1,1)	(1,1,1)
4	2.58	3.00	-14.00	1.55	1.33	16.54	55.85	70.67	-20.97	(2,2,2)	(2,2,2)
8	3.12	3.37	-8.01	2.57	2.37	8.44	58.89	68.40	-13.90	(3,3,3)	(3,3,3)
12	3.38	3.61	-6.37	3.56	3.32	7.23	52.70	57.10	-7.70	(4,5,4)	(6,6,6)
16	3.52	3.63	-3.03	4.54	4.41	2.94	43.25	43.80	-1.26	(5,6,6)	(7,7,7)

Table 3: 3-node Tandem Network with $s^2 = 0.50, \mu_i = 4\ \forall i$

[10] Himmelblau, D.M., 1972. *Applied Nonlinear Programming.* Mc-Graw-Hill.

[11] Kerbache, L. and J. MacGregor Smith, 1988. "Asymptotic Behaviour of the Expansion Method for Open Finite Queueing Networks." *Computers and Operations Research,* **15** (2), 157-169.

[12] Kerbache, L. and J. MacGregor Smith, 1987. "The Generalized Expansion Method for Open Finite Queueing Networks." *The European Journal of Operations Research* **32**, 448–461.

[13] Kimura, T., 1996. "A Transform-Free Approximation for the Finite Capacity M/G/s Queue," *Operations Research.* **44**(6), 984-988.

[14] Kimura, T., 1996. "Optimal Buffer Design of an M/G/s Queue with Finite Capacity," *Commun. Statist.-Stochastic Models,* **12**(1), 165-180.

[15] Leonard Kleinrock. *Queueing Systems, Volume I: Theory.* John Wiley and Sons, first edition, 1975.

[16] Labetoulle, J. and Pujolle, G. "Isolation Method in a network of queues," *IEEE Trans.on Software Eng., Vol. SE-6,* 4, 373-380, 1980.

[17] Onvural, Raif, 1990. "Survey of Closed Queueing Networks with Blocking," *ACM Computing Surveys* **22** (2), 83-121.

[18] M.J.D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Comput. J.*, (7):155–162, 1964.

[19] Powell, S.G., 1992, "Buffer Allocation in Unbalanced Serial Lines," Working Paper #289, Amos Tuck School of Business Administration, Dartmouth College, Hanover, N.H. 03755.

[20] Reiser, M. and S. Lavenberg, 1980. "Mean Value Analysis of Closed Multi-Chain Queueing Networks," *JACM* **27**(2), 313-322.

[21] J. MacGregor Smith and Nikhil Chikhale "Buffer Allocation for a class of Nonlinear Stochastic Knapsack Problems. *Annals of Operations Research* **58**, 323-360, 1995.

[22] Smith, J. MacGregor, 1999. "Convexity, Buffer Allocation and Finite Buffer Blocking Probabilities," *Second Aegean International Conference on Analysis and Modeling of Manufacturing Systems,* Editions ZITI, Thessaloniki, pp 17-26.

[23] Smith, J. MacGregor, S. B. Gershwin, and C.T. Papadopoulos, 2000. "Performance Evaluation and Optimization of Production Lines." *Annals of Operations Research,* **93**.

[24] Smith, J. MacGregor, and F.R. Cruz, 2000. "The Buffer Allocation Problem for General Finite Buffer Queueing Networks," *in review.*

[25] Springer, M. and P. Makens, 1991. "Queueing Models for performance analysis and Selection of Single Station Models," *EJOR* **58**, 123-145.

[26] Tijms, Henk, 1986. *Stochastic Modeling and Analysis.* New York:Wiley.

[27] Tijms, Henk, 1992. "Heuristics for Finite-Buffer Queues," *Probability in the Engineering and Informational Sciences* **6**, 277-285.

[28] Tijms, Henk, 1994. *Stochastic Models: An Algorithmic Approach.* New York:Wiley.

[29] Yao, D. and Buzacott, J.A., 1985. "Queueing Models for a Flexible Machining Station Part 1: The Diffusion Approximation." *EJOR***19**, 233-240.

N	θ			W			Z			**x**	
	$\mathcal{A}$	S	%	$\mathcal{A}$	S	%	$\mathcal{A}$	S	%	$\mathcal{A}$	S
1	1.18	1.33	-6.46	0.85	0.75	13.33	23.93	29.50	-18.88	(1,1,1)	(1,1,1)
4	2.22	2.47	-10.12	1.80	1.62	11.11	42.73	51.90	-17.67	(2,2,2)	(2,2,2)
8	2.66	2.79	-4.66	3.00	2.86	4.90	40.89	46.10	-11.30	(3,3,3)	(3,3,3)
12	2.91	3.19	-8.77	4.13	3.76	9.84	32.86	37.10	-11.43	(5,4,4)	(7,7,7)
16	3.06	3.26	-6.13	5.23	4.90	6.73	22.49	24.80	-9.31	(6,5,6)	(8,8,8)

Table 4: 3-node Tandem Network with $s^2 = 1.50, \mu_i = 4\ \forall i$

N	θ			W			Z			**x**	
	$\mathcal{A}$	S	%	$\mathcal{A}$	S	%	$\mathcal{A}$	S	%	$\mathcal{A}$	S
1	0.78	0.799	-2.38	1.29	1.25	3.20	9.28	10.47	-11.37	(1,1,1,1,1)	(1,1,1,1,1)
4	1.78	1.99	-10.55	2.25	2.01	11.94	34.78	44.55	-21.93	(1,1,1,1,1)	(1,1,1,1,1)
8	2.41	2.59	-6.95	3.31	3.08	7.47	41.32	49.85	-17.11	(2,2,2,2,2)	(2,2,2,2,2)
12	2.72	2.75	-1.09	4.42	4.36	1.38	37.88	39.65	-4.46	(2,3,3,3,2)	(2,3,3,3,2)
16	2.93	2.84	3.17	5.45	5.63	-3.20	31.16	26.10	19.39	(3,3,4,4,3)	(3,3,4,4,3)

Table 5: 5-node Tandem Network with $\mu_i = 4, s^2 = 1.0 \forall i$

N	θ			W			Z			**x**	
	$\mathcal{A}$	S	%	$\mathcal{A}$	S	%	$\mathcal{A}$	S	%	$\mathcal{A}$	S
1	0.79	0.80	-1.25	1.26	1.25	0.80	10.04	10.50	-4.38	(1,1,1,1,1)	(1,1,1,1,1)
4	1.90	2.28	-16.67	2.11	1.75	20.50	40.36	57.30	-29.56	(1,1,1,1,1)	(1,1,1,1,1)
8	2.58	3.00	-14.00	3.10	2.66	16.54	49.22	68.40	-28.04	(2,2,2,2,2)	(2,2,2,2,2)
12	2.91	3.17	-8.20	4.12	3.79	8.71	47.60	60.05	-20.73	(2,3,3,3,2)	(2,3,3,3,2)
16	3.13	3.24	-3.40	5.11	4.93	3.65	41.55	47.10	-11.78	(3,4,4,3,3)	(3,4,4,3,3)

Table 6: 5-node Tandem Network with $\mu_i = 4\ \forall i, s^2 = 0.50$

N	θ			W			Z			**x**	
	$\mathcal{A}$	S	%	$\mathcal{A}$	S	%	$\mathcal{A}$	S	%	$\mathcal{A}$	S
1	0.75	0.80	-6.25	1.33	1.25	8.00	8.04	10.50	-23.43	(1,1,1,1,1)	(1,1,1,1,1)
4	1.76	1.84	-4.35	2.27	2.18	4.18	29.08	32.60	-10.80	(2,2,2,2,2)	(2,2,2,2,2)
8	2.20	2.34	-5.98	3.63	3.42	6.14	30.84	37.70	-18.20	(2,2,2,2,2)	(2,2,2,2,2)
12	2.51	2.58	-2.71	4.77	4.64	2.80	25.24	28.90	-12.66	(3,3,3,3,3)	(3,3,3,3,3)
16	2.67	2.57	3.89	6.00	6.22	-3.54	16.34	10.75	52.00	(3,3,3,4,4)	(3,3,3,4,4)

Table 7: 5-node Tandem Network with $\mu_i = 4\ \forall i, s^2 = 1.50$

N	θ			W			Z			**x**	
	$\mathcal{A}$	S	%	$\mathcal{A}$	S	%	$\mathcal{A}$	S	%	$\mathcal{A}$	S
1	0.90	0.93	-3.23	1.12	1.08	3.70	10.71	12.16	-12.01	(1,1,1,1,1)	(1,1,1,1,1)
4	2.01	2.20	-8.82	1.99	1.82	9.34	34.24	41.80	-18.09	(1,1,1,2,1)	(1,1,1,2,1)
8	2.55	2.68	-4.85	3.14	2.98	5.37	35.99	41.60	-13.49	(1,1,1,4,2)	(1,1,1,4,2)
12	2.79	2.81	-0.68	4.30	4.28	0.46	27.73	28.50	-2.71	(1,1,1,6,4)	(1,1,1,6,4)
16	2.91	2.83	2.83	5.50	5.65	-2.65	15.20	10.40	46.15	(1,1,1,2,8,5)	(1,1,1,2,8,5)

Table 8: 5-node Tandem Network with unequal $\mu_i = (5, 8, 4, 3, 6), s^2 = 1.0$

N	θ			W			Z			**x**	
	$\mathcal{A}$	S	%	$\mathcal{A}$	S	%	$\mathcal{A}$	S	%	$\mathcal{A}$	S
5	0.93	0.95	-2.11	5.39	5.24	2.86	39.64	43.48	-8.83	(1,1,1,1,2,1,2,1)	(1,1,1,1,2,1,2,1)
6	1.00	1.01	-0.99	6.02	5.93	1.52	41.43	43.56	-4.89	(1,1,1,1,2,1,2,1)	(1,1,1,1,2,1,2,1)
7	1.05	1.05	0.00	6.67	6.68	-0.15	41.61	41.56	0.12	(1,1,1,1,2,1,2,1)	(1,1,1,1,2,1,2,1)
8	1.09	1.07	1.87	7.33	7.46	-1.74	40.54	37.32	8.63	(1,1,1,1,2,1,2,1)	(1,1,1,1,2,1,2,1)

Table 9: 8-node Tandem Network with $\mu_i = (2, 8, 5, 2.5, 2, 4, 1.25, 5), s^2 = 1.0$