JOINT BUFFER AND SERVER ALLOCATION IN GENERAL FINITE QUEUEING NETWORKS

F. R. B. Cruz
Departamento de Estatística, Universidade Federal de Minas Gerais,
31.270-901 – Belo Horizonte – MG
fcruz@est.ufmg.br

T. van Woensel
School of Industrial Engineering and Innovation Sciences, Eindhoven University of Technology, P.O. Box 513, 5600 MB, Eindhoven, The Netherlands
t.v.woensel@tue.nl

ABSTRACT

The focus of this paper is on the optimization of finite queueing networks which may represent the manufacturing networks in a joined manufacturing and product engineering environment. The performances of the queueing networks are evaluated in terms of their throughput by means of an advanced queueing network analyzer, the generalized expansion method. The problem solved is the joint buffer and server optimal allocation. Given the difficulty of the objective function which is not known in closed form, a heuristic method based on the Powell algorithm is used. Preliminary numerical results are presented to attest for the quality of the approach. Some new insights are given for this challenging and important stochastic optimization problem.

KEYWORDS. Queueing networks; Optimization models; Manufacturing systems.

Main area: MP - Probabilistic Models

1. Introduction and Motivation

The optimization of manufacturing systems and complex production lines has been the focus of numerous studies. Queueing networks are commonly used to model such complex systems (Suri, 1985). This paper discusses an optimization approach from a queueing theory point of view. More specifically, the focus are queueing networks that have finite buffer spaces, as seen in Figure 1, which is characterized by a blocking effect that eventually degrades the performance, commonly measured via the throughput Θ of the network.

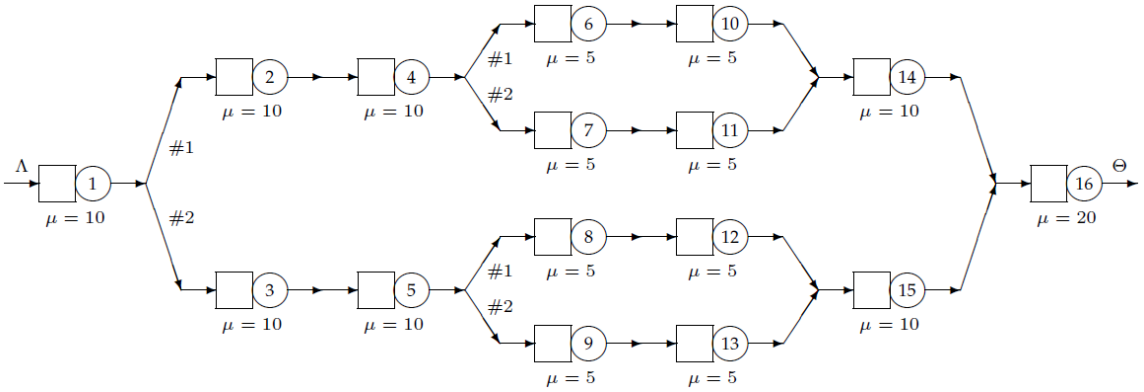

Figure 1: Network of finite general-service queues in an arbitrary series-split-merge topology

A practical application for the finite queueing models includes the manufacturing step. Since product engineering is inevitably connected to the manufacturing process, a better understanding of the manufacturing part in the product engineering phase could lead to a strong and sustainable competitive advantage (Simchi-Levi *et. al.*, 2009).

Finite Queueing Networks

Queueing networks are defined as open, closed, or mixed. In open queueing networks, entities enter the system from outside, receive some service at one or more nodes, and then leave the system. In closed queueing networks, entities never leave or enter the system: a fixed number of entities circulates within the network (Whitt, 1984). Mixed queueing networks are systems that are open with respect to some entities and are closed with respect to others (Balsamo *et al.*, 2001). Research in the area of queueing networks is very active, resulting in a vast amount of journal papers, books, and reports. For a general and a more complete discussion on queueing networks, the reader is referred to *e.g.* Walrand (1988). In the remaining of this paper, we will focus on open queueing networks.

An additional assumption is that the capacities (the buffer spaces) between two consecutively connected service stations are finite. As a consequence, each node in the network might be affected by events at other nodes, leading to the phenomena of blocking and starvation.

In the literature, two general blocking mechanisms are presented, which are: *blocking-after-service* and *blocking-before-service*. *Blocking-after-service* occurs when after the service an entity sees that the buffer in front of it is full and as a consequence cannot continue its way throughout the network. *Blocking-before-service* implies that a server can start processing the next entity only if there is a space available in the downstream buffer. If not, the entity has to wait until a space becomes available. Most production lines operate under the blocking-after-service system. Moreover, in the literature it is the most commonly made assumption regarding the buffer behavior (Dallery & Gershwin, 1992).

Performance evaluation tools include *product form methods*, *numerical methods*, and *Monte Carlo simulation*. Concerning the *product form methods*, the queueing system is

decomposed into single, pairs, or triplets of nodes. Each decomposed node can then be treated as an independent service provider, for which all results and insights of the single node queueing models can be used (*e.g.*, see Gross *et al.*, 2009). Decomposition techniques yield exact results for queueing networks with product form solutions. For networks without a product form solution, they will give only an approximation. If obtaining an exact solution is too difficult, *numerical methods* may be a good option. The main challenge is to be as close as possible to the exact values. Numerical methods are sometimes restricted to small networks (see, *e.g.*, Balsamo *et al.*, 2001). Finally, another strategy to obtain all relevant performance measures for a queueing network is making use of *Monte Carlo simulation*, a computationally intensive method (Law & Kelton, 2000).

In this paper, the generalized expansion method (GEM) is used as the prime performance evaluation tool. The method was proposed originally by Kerbache & Smith (1987) and it is combination of a node-by-node decomposition and an iterative numerical approximation. Details will not be given here but may be found in the literature (see, *e.g.*, Kerbache & Smith, 1988).

Structure of the paper

This paper is structured as follows. In Section 2, we detail the mathematical programming formulation for the queueing network optimization problem. In Section 3, the optimization methodology is discussed. Preliminary results are presented in Section 4. Finally, Section 5 concludes the paper with final remarks and topics for future research in this area.

2. Mathematical Programming Formulation

The network structure is defined on a digraph $G(V,A)$, in which V is the set of queues, characterized by Poisson arrivals, multiple servers, generally distributed service times, and a total capacity K (*i.e., including* the items under service), that is, in Kendall notation, $M/G/c/K$ queues. Additionally, the queues are interconnected by a set of arcs A, with a given routing probability. Then, we seek for the optimal number of buffers and servers in each queue V_i. We can write the generic optimization model as follows:

$$Z = \min f(\mathbf{X}),$$

subject to:

$$\Theta(\mathbf{X}) \geq \Theta^{\tau},$$
$$\mathbf{X} \geq 0,$$

that minimizes the total allocation cost, $f(\mathbf{X}) \equiv \sum_{\forall i \in V} X_i$, constrained to provide a minimum throughput Θ^{τ}.

A number of specific models can be specified based on the above generic model. In this paper, we are interested in the combination of buffers B_i and servers c_i allocation, which is done by setting $\mathbf{X} \equiv (\mathbf{B},\mathbf{c})$. In this case, some integrality constraints must be included, $B_i \in \mathbb{N}$, $c_i \in \mathbb{N}$, $\forall i \in V$. Next to this integrality constraint, more constraints are needed. It is necessary to ensure that there is at least one server per vertex, $c_i \geq 1$, $\forall i \in V$. Note also that buffers, defined as $B_i = K_i - c_i$, $\forall i \in V$, can be equal to zero, hence leading to a zero-buffer system.

Secondly, note that the objective function needs to be adapted slightly to take into account the two objectives (*i.e.*, the buffers and servers allocation). We consider that the objective function can be written as a weighted sum of these two objectives, giving the so called joint buffer and server allocation problem (BCAP):

$$Z_{\text{BCAP}} = \min\left[(1-\omega) \sum_{\forall i \in V} B_i + \omega \sum_{\forall i \in V} c_i \right].$$

We assign a cost of $(1-\omega)$ to buffers and ω to servers. We can then modify the value

of ω, such that $0 \leq \omega \leq 1$, to reflect the relative cost of buffers versus servers. As ω is increased, the cost of buffers will become relatively lower than that of the servers. That is, servers are then more expensive than buffers. Alternatively, when the value of ω is decreased, the buffers become more costly relative to the servers and therefore the buffers become more expensive than the servers. In this way, we evaluate whether different pricing of buffers and servers results in a significantly different buffer and server allocation. It is worthwhile to mention that if ω=0, the above problem reduces to the pure buffer allocation problem (BAP) and if ω =1, the pure server allocation problem (CAP) is obtained.

3. Optimization Methodology

While the GEM computes the performance measures for the queueing network, $\Theta(\mathbf{B},\mathbf{c})$, the mathematical programming formulation described earlier need to be optimized on the decision variables $\mathbf{X} \equiv (\mathbf{B},\mathbf{c})$. Of course, there exist many optimization methods that could be applied to the BCAP. An exhaustive discussion is left out of this paper, but the interested reader is referred to Aarts & Lenstra (2003) and the references therein. We describe one of the methodologies that have proven to be successful to similar models, namely the Powell (1964) algorithm, mainly because of the difficulty of obtaining $\Theta(\mathbf{B},\mathbf{c})$ in closed form. Of course, small problems can always be enumerated.

The Powell algorithm can be described as an unconstrained optimization procedure that does not require the calculation of first derivatives of the function, which is very convenient for the problem on hand because of its *relaxed* objective function presented below which is not available in closed form:

$$Z_{\text{BCAP}}\varepsilon = \min\left[(1-\omega)\sum_{\forall i \in V} B_i + \omega \sum_{\forall i \in V} c_i - \alpha\left(\Theta(\mathbf{B},\mathbf{c}) - \Theta^{\tau}\right) \right].$$

Numerical examples have shown that the Powell algorithm is capable of minimizing a function with up to twenty variables (Powell, 1964; Himmelblau, 1972). The Powell algorithm locates the minimum of a non-linear function $f(\mathbf{X})$ by successive unidimensional searches from an initial starting point $\mathbf{X}_{(0)}$ along a set of conjugate directions. These conjugate directions are generated within the procedure itself. The Powell algorithm 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 sub-directions of the search.

4. Results and Insights

In this section, we will focus on one example network. We consider a combination of the three basic topologies (series, split, and merge), as shown in Figure 1 (Smith & Cruz, 2005). This network consists of 16 nodes (finite queues) with the server processing rates μ, as shown in Figure 1. Concerning its efficiency, the Powell algorithm seems to be dependent on the arrival rate at the network and also on the squared coefficient of the variation of the service times, as seen in Figure 2. Large arrival rates and large squared coefficient of variation imply large solution spaces that must be sought. However, the processing time does not increase dramatically, which is encouraging.

The sub-optimal allocations for buffers and servers are presented in Table 1 along with the optimal values for the relaxed objective function $Z_{\text{BCAP}}\varepsilon$ (for α=1,000). We used the values for the external arrival rate Λ and for the squared coefficient of variation of the service time s^2 as given in the table and the 50%-50% routing probabilities for the splitting node (*i.e.*, nodes #1 and #2). Note that the routing probability #1 refers to the up tier of the node, while #2 refers to the low tier. Refer to Figure 1 for the position of each node in the network.

In Table 1, the c/B price ratio gives the relative cost of servers compared to buffers. A price ratio of 1 means that servers are as much expensive as the buffers, that is,

$(1\text{-}\omega):\omega \equiv 0.5:0.5$. An extensive study about the influence of such a ratio was presented elsewhere (Authors, year) and it will not be repeated here. We only mention that it has been found that $M/G/1/K$ queues (*i.e.*, single-server queues) are not an optimal configuration for this particular queueing network, except when the buffers are very expensive compared to the servers (about 8 times or more). This makes sense since the servers play a double role, that is, servers are service providers and also spaces for staying. These results justify research efforts to extend the single-server based models into the multiple-server based models, as proposed here.

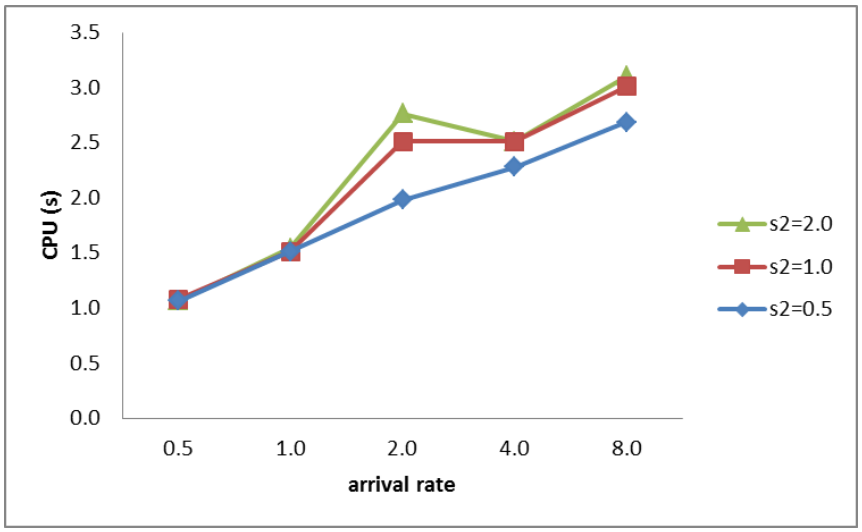

Figure 2: Running times

Table 1 shows suboptimal solutions for several external arrivals Λ (*i.e.*, from 0.5 to 8.0 entities per time unit). We observe the presence of the bowl effect (*i.e*, larger spaces are allocated at the borders than in the middle), as well-known long ago for lines in series (Rao, 1976). Additionally, zero-buffer configurations are identified almost everywhere which is expected since servers are comparably cheaper than buffers. Varying the coefficient of variation of the service time does result in some changes in the optimal buffer and server allocation, which shows the importance of models that deals with general service times (that is, with $s^2 \neq 1.0$). The results show that the number of servers seems to be large with high variability, as it could be expected, since the increase in the squared coefficient of variation of the service times means an increase in the variability. Additional servers are allocated to help handling the extra variability. Also noticeable is that usually under high traffic the preferred configurations are bufferless. Of course, this is because of the low c/B ratio considered.

Practical Issues

In a number of industrial improvement projects carried out, we observed that the critical issue to be able to use similar queue based models is related to data availability. More specifically, processing rates, arrival rates, uncertainty in the service process *etc.*, needs to be extracted from the available databases. An interesting approach to obtaining the relevant data is the effective process time (EPT) point of view.

According to Hopp & Spearman (1996) the random variable of primary interest in factory physics is the effective process time (EPT) of a job at a workstation. The label effective is used because the authors refer to the total time seen by a job at a station. From a logistical point of view, it does not matter whether the job is actually being processed or is being held up because the workstation is being repaired, undergoing a setup, reworking the part due to a quality problem, or waiting for an operator to return from a break. For this

reason, it is possible to combine these effects into one aggregate measure of variability.

Table 1: Results for the BCAP

c/B	s^2	Λ	c	K	Σc	ΣK	ΣB	Θ	$Z_{BCAP\,\varepsilon}$	CPU(s)
1.0	0.5	0.5	(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)	(2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2)	16	18	2	0.4990	10.0	1.1
		1.0	(2,2,2,2,2,1,1,1,1,1,1,1,1,2,2,2)	(2,2,2,2,2,1,1,1,1,1,1,1,1,2,2,2)	24	24	0	0.9989	13.1	1.5
		2.0	(2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2)	(2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2)	32	32	0	1.9994	16.6	2.0
		4.0	(3,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3)	(3,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3)	34	34	0	3.9983	18.7	2.3
		8.0	(5,3,3,3,3,2,2,2,2,2,2,2,3,3,5)	(5,3,3,3,3,2,2,2,2,2,2,2,3,3,5)	44	44	0	7.9976	24.4	2.7
	1.0	0.5	(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)	(2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2)	16	18	2	0.4988	10.2	1.1
		1.0	(2,2,2,2,2,1,1,1,1,1,1,1,1,2,2,2)	(2,2,2,2,2,1,1,1,1,1,1,1,1,2,2,2)	24	24	0	0.9988	13.2	1.5
		2.0	(2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2)	(2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2)	32	32	0	1.9993	16.7	2.5
		4.0	(3,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3)	(3,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3)	34	34	0	3.9978	19.2	2.5
		8.0	(5,3,3,3,3,2,2,2,2,2,2,2,3,3,5)	(5,3,3,3,3,2,2,2,2,2,2,2,3,3,5)	44	44	0	7.9970	25.0	3.0
	2.0	0.5	(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)	(2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2)	16	18	2	0.4984	10.6	1.1
		1.0	(2,2,2,2,2,1,1,1,1,1,1,1,1,2,2,2)	(2,2,2,2,2,1,1,1,1,1,1,1,1,2,2,2)	24	24	0	0.9984	13.6	1.5
		2.0	(2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2)	(2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2)	32	32	0	1.9989	17.1	2.8
		4.0	(5,2,2,2,2,2,2,2,2,2,2,2,2,2,2,5)	(5,2,2,2,2,2,2,2,2,2,2,2,2,2,2,5)	38	38	0	3.9969	22.1	2.5
		8.0	(6,3,3,3,3,3,3,3,3,3,3,3,3,3,3,6)	(6,3,3,3,3,3,3,3,3,3,3,3,3,3,3,6)	54	54	0	7.9992	27.8	3.1

Kock *et al.* (2008) propose an EPT approach in four steps (see Figure 3). The first step is to measure realizations from the manufacturing system. An EPT-realization represents the time a job consumed capacity from the respective workstation. EPT realizations can be obtained from event data, such as arrivals and departures of jobs on workstations. The second step is to describe the EPT realizations by statistical distributions. The third step is to build an aggregate model (either simulation or analytical) from the obtained distributions. The fourth step is to validate the aggregate model by comparing the throughput and lead-time as estimated by the model to the throughput and lead-time observed in the actual system.

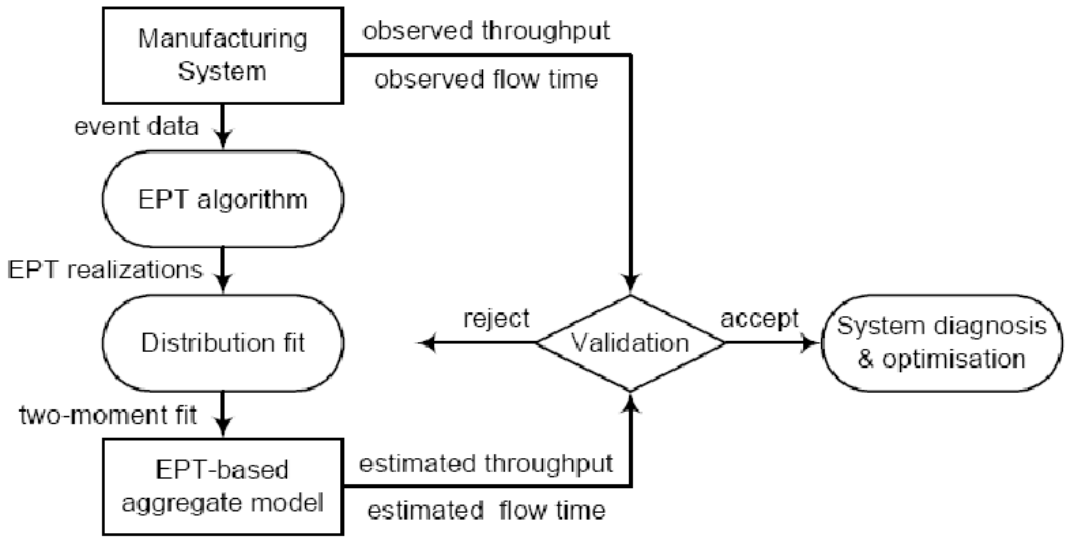

Figure 3: The effective processing time approach

Of course, if the project on-hand is a pure design issue in a green field study, it is not trivial to find the right data. In this case, specifications from machine builders or from similar situations could be used.

5. Conclusions and Future Research Directions

This paper presented preliminary results for the joint buffer and server allocation problem. We used the general expansion method as the performance evaluation tool for the finite queueing networks. This methodology has proved in the literature to be a valuable approach. The joint buffer and server allocation problem was then 'solved' by means of a

Powell based heuristic. The paper ended with a summary of some results for different settings considered for a complex queueing network.

Concerning the practical use of the methodology we discussed briefly the advantages of the effective process time (EPT) approach. Thus various types of disturbances on the shop-floor can be aggregated into EPT distributions which enable effective modeling. However, it is important to note that disturbances which are aggregated into the EPT distribution cannot be analyzed afterwards. Hence, shop-floor realities or disturbances which are modeled explicitly and excluded from aggregation in the EPT are defined beforehand.

In this paper, we considered the throughput as the main performance measure. Instead of the throughput, it would be interesting to evaluate the behavior of the models based on the cycle time, the work-in-process (WIP), or some other performance measures. Topics for future research would also include the analysis and optimization of networks with cycles, for instance, to model systems with feed-back loops caused by re-work, or even the extension to more general queueing networks, such as networks of $GI/G/c/K$ queues (*i.e.*, that include generally distributed and independent arrivals).

Acknowledgments

The research of Frederico Cruz has been partially funded by CNPq of the Ministry for Science and Technology of Brazil and by FAPEMIG.

References

Aarts, E.H.L. & Lenstra, J.K. *Local Search in Combinatorial Optimization*. Princeton University Press, Princeton, NJ, 2nd ed., 2003.

Author. (year). Title. *Journal*, number, pages. (to be updated)

Balsamo, S.; de Nitto Personé, V. & Onvural, R. *Analysis of Queueing Networks with Blocking*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001.

Dallery, Y. & Gershwin, S.B. (1992). Manufacturing Flow Line Systems: A Review of Models and Analytical Results, *Queueing Systems*, 12, 3-94.

Gross, D.; Shortle, J. F.; Thompson, J. M. & Harris, C. M. *Fundamentals of Queueing Theory*. Wiley-Interscience, New York, NY, 4th edn., 2009.

Himmelblau, D.M. *Applied Nonlinear Programming*, McGraw-Hill Book Company, New York, NY, 1972.

Hopp, W.J. & Spearman, M.L. *Factory Physics, Foundations for Manufacturing Management*. Mc-Graw Hill, 1996.

Kerbache, L., Smith, J.M. (1987). The generalized expansion method for open finite queueing networks. *European Journal of Operational Research*, 32, 448-461.

Kerbache, L., Smith, J.M. (1988). Asymptotic behavior of the expansion method for open finite queueing networks. *Computers & Operations Research*, 15(2), 157-169.

Kock, A.A.A.; Etman, L.F.P.; Rooda, J.E. (2008). Effective process times for multi-server flow lines with finite buffers. *IIE Transactions*, 40(3), 177-186.

Law, A.M. & Kelton, W.D. *Simulation Modeling and Analysis*. McGraw-Hill Higher Education, New York, NY, 3rd ed., 2000.

Powell, M.J.D. (1964). An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer Journal*, 7, 155-162.

Rao, N. (1976). A generalization of the bowl phenomenon in series production systems. *International Journal of Production Research*, 14(4), 437-443.

Simchi-Levi, D.; Kaminsky, P. & Simchi-Levi, E. *Designing and Managing the Supply Chain Concepts Strategies and Case Studies*. McGrawHill/Irwin, 2009.

Smith, J. M., Cruz, F. R. B. (2005). The buffer allocation problem for general finite buffer queueing networks. *IIE Transactions*, 37(4), 343-365.

Suri, R. (1985). An overview of evaluative models for flexible manufacturing systems. *Annals of Operations Research*, 3, 13-21.

Walrand, J. *An Introduction to Queueing Networks*. Prentice-Hall, Englewoord Cliffs, 1988.

Whitt, W. (1984). Open and closed models for networks of queues. *AT&T Bell Laboratories Technical Journal*, 63(9), 1911-1979.