Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Estatística

A Selective Review of Optimization Models in Manufacturing

T. van Woensel & F. R. B. Cruz

Relatório Técnico RTP-02/2014

Relatório Técnico
Série Pesquisa

A Selective Review of Optimization Models in Manufacturing

T. van Woensel*
t.v.woensel@tue.nl

F. R. B. Cruz†
fcruz@est.ufmg.br

Abstract — **An overview of the modeling issues and the related performance evaluation and optimization approaches is provided in this selective review framed in a joined manufacturing and product engineering environment. Queuing networks represent the manufacturing networks. The performance of the manufacturing networks is evaluated using an advanced queueing network analyzer, the generalized expansion method (GEM). Secondly, different model approaches are described and optimized with regards to the key parameters in the network (*e.g.* buffers, servers and so on).**

Keywords — Queueing networks; performance evaluation; optimization models; manufacturing systems.

1 Introduction and Motivation

Optimization of large scale manufacturing systems and complex production lines has been and continues to be the focus of numerous studies for decades. Queueing networks are commonly used to model such complex systems (Suri, 1985). This review provides an overview of modeling, performance evaluation, and optimization approaches from a queueing theory point of view[1]. More specifically, finite buffer queueing networks are characterized by blocking that eventually degrades the performance, commonly measured via the throughput of the network.

Product engineering is inevitably connected to the manufacturing process. Explicitly recognizing the role of manufacturing in the product engineering phase could lead to a strong and sustainable competitive advantage. Following Simchi-Levi, et. al (2008), we adopt the same approach as in Figure 1. Clearly, the development chain (product engineering) and the supply chain (product manufacturing) are interacting. In this paper, we focus on the manufacturing step as it will be important to consider its role, its characteristics and the consequences in the product engineering phase.

Finite Queueing Networks

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

The assumption is that the capacity of the buffer space between two consecutive connected service stations is 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 service, a customer sees that the buffer in front of him is full and as a consequence cannot continue its way in the network. Blocking-before-service implied that a server can start processing a next customer only if there is a space available in the downstream buffer. If not, the customer 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 and Gershwin, 1992).

Performance evaluation tools include product form methods, numerical methods, approximate methods and simulation. Let's discuss each of these methods a bit more in detail. More in-depth information can be

*School of Industrial Engineering and Innovation Sciences, Eindhoven University of Technology, P.O. Box 513, 5600 MB, Eindhoven, The Netherlands

†Departamento de Estatística, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brazil. Corresponding author. Phone: +55 31 3409 5929. Fax: +55 31 3409 5924.

[1] Queueing theory is the mathematical study of waiting lines and enables the mathematical analysis of several related processes, including arrivals at the queue, waiting in the queue, and being served by the server. The theory enables the derivation and calculation of several performance measures which can be used to evaluate the performance of the queueing system under study.

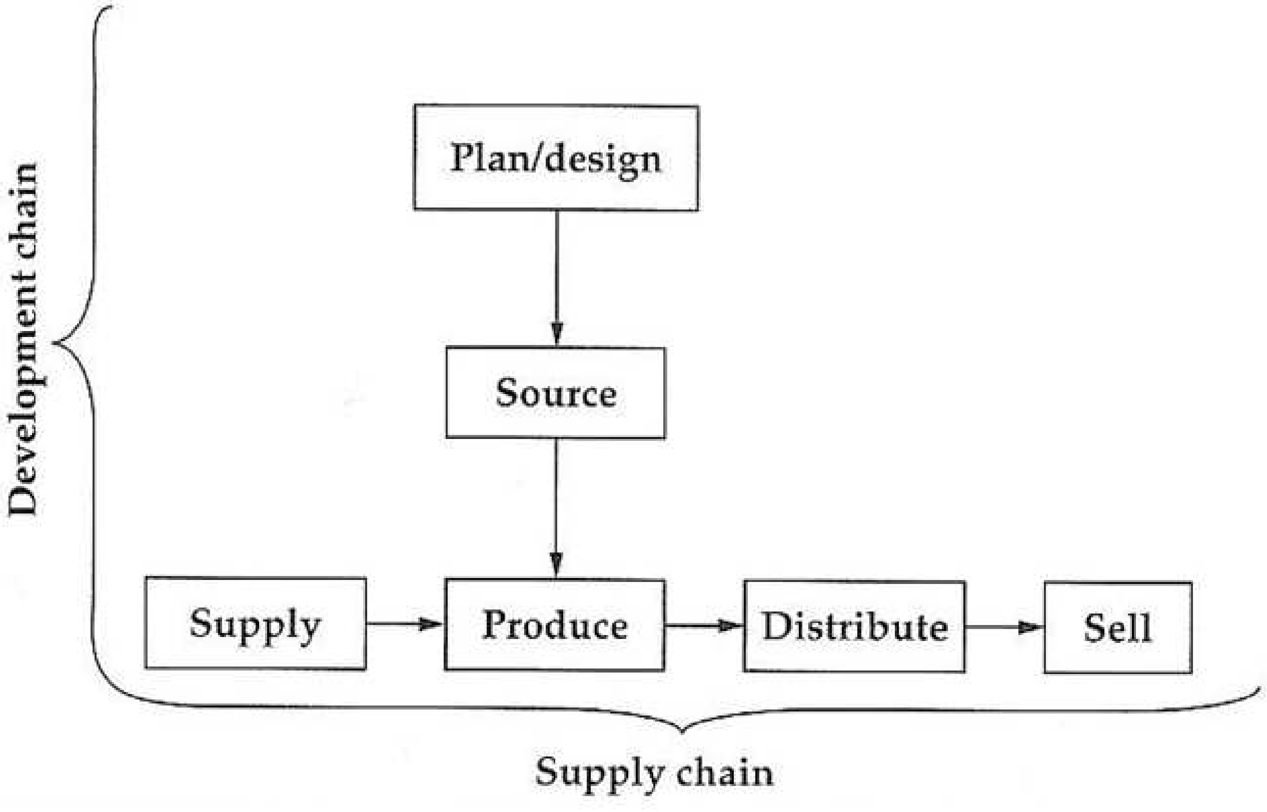

Figure 1: The development chain versus the supply chain

found in the references mentioned below.

Instead of analyzing the entire system at once, product form methods decompose the system into single, pairs or triplets of nodes (Perros, 1994). 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 (see *e.g.* Gross and Harris, 1998). Jackson (1957, 1963) first showed that the joint distribution of the entire network is made up of the product of the marginal distributions at each of the nodes under some strict conditions (*e.g.* exponential arrivals/service and no blocking). A decomposition technique yields exact results for queueing networks with product form solutions. For networks without a product form solution, it gives a good approximation (Balsamo et al., 2001).

Decomposition methods are also approximations because the subnetworks are only a part of the whole line and as such, do not have the same behavior (Dallery and Gershwin, 1992). Moreover, if obtaining an exact solution is too expensive in terms of (computer) time, *approximate methods* are used. The main challenge with approximate methods is to be as close as possible to the exact values. The accuracy of an approximate method can be tested with numerical solutions (for smaller networks) or by using simulation. The main idea of the decomposition methods is try to generalize the ideas of independence and product form solutions from the Jackson networks to more general systems. Reiser and Kobayashi (1974) and Kuehn (1979) were the first to develop this approach. After them, several researchers came up with a similar approach (*e.g.* Buzacott and Shanthikumar, 1993).

In theory, every Markovian model can be solved using numerical methods. The problem however with numerical solutions is that the state space of queueing networks grows exponentially with the number of nodes, the number of customers and the number of buffers. As a consequence, numerical methods consume extensive computer time to get to the solution. Numerical methods are sometimes applied to smaller networks (see *e.g.* Balsamo et al., 2001). A last way to obtain all relevant performance measures for a queueing network is making use of simulation (Law and Kelton, 2000).

The generalized expansion method (GEM) is used in this paper as the prime performance evaluation tool. Consequently, this paper provides a selected review based on the GEM and does not explicitly consider other methodologies to obtain the performance measures. Note that the models described fit any performance evaluation tool.

Structure of the paper

The paper is structured as follows. In Section 2, we present the performance evaluation of the queueing networks considered. In this paper, we use the GEM to obtain the relevant performance measures. In Section 3, we elaborate on the different optimization models that exist and discuss some of the optimization tools that are used to optimize these models. Section 4 gives for a complex network the results for some selected optimization models. The last section concludes the paper and gives some pointers for future research.

2 Network Performance Evaluation

In general, we evaluate the performance of the network via its throughput θ. This throughput (and all other measures) can be obtained via a queueing network representation. This queueing network representation[2] then needs to be 'solved' to obtain the performance of the given network.

The GEM transforms the queueing network into an equivalent Jackson network, which can be decomposed so that each node can be solved independently of each other (similar to a product form solution approach). The GEM is an effective and robust approximation technique to measure the performance of open finite queueing networks. The effectiveness of GEM as a performance evaluation tool has been presented in many papers, including Kerbache and Smith (1987, 1988, 2000), Jain and Smith (1994), Smith (2003), and Andriansyah et al. (2010). The GEM uses blocking after service (BAS), which is prevalent in most production and manufacturing, transportation, and other similar systems. Developed by Kerbache and Smith (1987), the GEM has become an appealing approximation technique for performance evaluation of queueing networks due to its accuracy and relative simplicity. Moreover, exact solutions to performance measurement are restricted only to very simple networks and simulation requires a considerable amount of time.

The GEM is basically a combination of two approximation methods, namely repeated trials and node-by-node decomposition. To evaluate the performance of a queueing network, the GEM first divides the network into single nodes with revised service and arrival parameters. Blocked customers are registered into an artificial 'holding node' and are repeatedly sent to this node until they are serviced. The addition of the holding node *expands* the network and transforms the network into an equivalent Jackson network, where each node can be solved independently. Generally, the GEM assumes a *type I blocking* that is commonly referred to as transfer blocking. This occurs when the service of a job is completed at a certain node but it cannot proceed to the next node because the queue is full.

In the remainder of this section, we will present a high-level overview of the method. For more detailed information and applications of the GEM, the reader is referred to *e.g.* Kerbache and Smith (1988). The GEM described below assumes that one wants to solve $M/G/c/K$ queueing networks. Note that the methodology is generic such that also $M/M/1/K$, $M/M/c/K$, $M/G/1/K$, $GI/G/1/K$, and $GI/G/c/K$ queueing networks can be analyzed. Only the relevant equations (*e.g.* the blocking probabilities) need to be adapted for these other cases.

There are three main steps in the GEM:

1. Stage I: network reconfiguration
2. Stage II: parameter estimation
3. Stage III: feedback elimination.

The notation for the GEM, presented in Table 1, will be used throughout the paper (Kerbache and Smith, 1988). The steps are described as follows.

Table 1: Basic network notation

Variable	Description
Λ	external Poisson arrival rate to the network;
λ_j	Poisson arrival rate to node j;
$\tilde{\lambda}_j$	effective arrival rate to node j;
μ_j	exponential mean service rate at finite node j;
$\tilde{\mu_j}$	effective service rate at finite node j due to blocking;
p_K	blocking probability of finite queue of size K;
p'_K	feedback blocking probability in the GEM;
h	the artificial holding node created in the GEM;
c	number of servers;
B_j	buffer capacity at node j *excluding* those in service;
K_j	buffer capacity at node j *including* those in service;
N	number of nodes in the network;
ρ	$\lambda/(\mu c)$ = traffic intensity;
θ	mean throughput rate;
s^2	squared coefficient of variation of the service time distribution.

Stage I: Network Reconfiguration

For each finite node in the queueing network, an artificial node is created to register the blocked jobs. By introducing such artificial node, we also create new routing probabilities in the network. The result of network reconfiguration can be seen from Figure 2.

There are two possible states of the finite node, namely *saturated* and *unsaturated*. Arriving jobs will try to access the finite node j. With a probability of $(1-p_K)$, the job will find the the finite node unsaturated, where it will enter the queue and eventually get serviced. However, if the finite node is saturated (with a probability of

[2] In order to refer to the queueing models, we use Kendall's notation, in which $M/G/1/K$ means a queueing system with **M**arkovian arrival rates, **G**eneral service times, **1** server in the node and **K** capacity of the node (including the server).

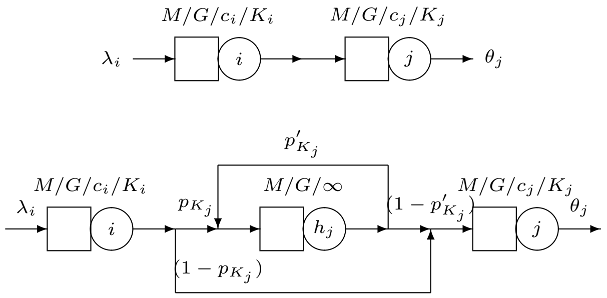

Figure 2: The generalized expansion method

p_K), then the job will be directed to the artificial holding node h where it will get a delay. The delay at the artificial node is modeled using a $M/G/\infty$ queue, representing delay time without queueing. Afterward, the blocked job will try to re-enter the finite queue with a success probability of $(1-p'_K)$. There is a probability of p'_K that the blocked job still finds the finite node saturated and thus it will be directed again to the artificial node h. This process repeats until the blocked job is able to enter the finite node.

Stage II: Parameter Estimation

At this stage, the values for parameters p_K, p'_K, and μ_h are determined.

- In order to determine p_K, exact analytical formulas should be used whenever possible (Kerbache and Smith, 2000). For cases where exact p_K formula is unavailable, approximations for p_K in $M/G/c/K$ setting provided by Smith (2003) can be used. These approximations are based on a closed-form expression derivable from the finite capacity exponential queue ($M/M/c/K$) using Kimura's (1996) two-moment approximation. The following p_K formula for $M/G/2/K$ is presented as an example

$$p_K = \frac{2\,\rho^{2\,\frac{\left(2+\sqrt{\frac{\rho}{e}}s^2-\sqrt{\frac{\rho}{e}}+B\right)}{2+\sqrt{\frac{\rho}{e}}s^2-\sqrt{\frac{\rho}{e}}}}\,(2\,\mu-\lambda)}{-2\,\rho^{2\,\frac{\left(2+\sqrt{\frac{\rho}{e}}s^2-\sqrt{\frac{\rho}{e}}+B\right)}{2+\sqrt{\frac{\rho}{e}}s^2-\sqrt{\frac{\rho}{e}}}}\,\lambda+2\,\mu+\lambda}.$$

- Since no exact method is available to calculate p'_K, an approximation from Labetoulle and Pujolle (1980), based on diffusion techniques, is used

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

  in which r_1 and r_2 are the roots of the polynomial

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

  in which $\lambda=\lambda_j-\lambda_h(1-p'_K)$, and λ_j and λ_h are the actual arrival rates to the finite and artificial holding notes, respectively. Furthermore, it can be argued that

$$\lambda_j = \tilde{\lambda_i}(1-p_K) = \tilde{\lambda_i}-\lambda_h.$$

- The delay distribution at the holding node h is actually nothing but the remaining service time of the finite node j. Based on the renewal theory, one can formulate the remaining service time distribution as the following rate μ_h

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

  in which σ_j^2 is the service time variance of the finite node. At this point, one should notice that if the service time of the finite node is exponentially distributed with rate μ_j, then the memoryless property of exponential distribution will hold such that

$$\mu_h = \mu_j.$$

Stage III: Feedback Elimination

As a result of the feedback loop at the holding node, a strong dependency on the arrival process is created. In order to eliminate such dependency, the service rate at the holding node must be adjusted as follows

$$\mu'_h = (1-p'_K)\mu_h.$$

As a consequence, the service rate at node i preceding the finite node j is affected as well. One can see that the mean service time at node i is μ_i^{-1} when the finite node is unsaturated, and $\mu_i^{-1}+{\mu'_h}^{-1}$ when the finite node is saturated. Thus, on average, the mean service time of node i preceding the finite node j is

$$\tilde{\mu}_i^{-1} = \mu_i^{-1}+p_K{\mu'_h}^{-1}.$$

The above equations apply to all finite nodes in the queueing network. To sum up, all performance measures of the network can be obtained by solving the following equations simultaneously

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

Note that Eq. (9), for p_K, only applies to an $M/G/2/K$ setting. Other expressions for p_K for $M/G/c/K$ queues, with $c = 3$ to $c = 10$, have been developed by Smith (2003) and can be used in the above set of equations.

3 Optimization Models

In this section, we review some of the optimization models found in the literature. Given a network structure $G(V, A)$ characterized by Poisson arrivals, $|V|$ nodes with non-negative buffers, multiple servers, a general service distribution and interconnected with arcs A, we can optimize on the number of buffers or the number of servers used in each vertex V_i, the characteristics of the service distribution (*e.g.* the service rates and the variability), on the routings used on the arcs A or any combination of these possible decision variables.

In general, we can write the generic optimization model as follows:

$$Z = \min f(\mathbf{X}), \qquad (10)$$

subject to:

$$\begin{aligned}
\Theta(\mathbf{X}) &\geq \Theta^\tau, & (11)\\
\mathbf{X} &\geq 0, & (12)
\end{aligned}$$

that minimizes the total allocation $f(\mathbf{X}) = \sum_{i \in V} X_i$ (*i.e.* over all vertices $i \in V$), constrained to provide a minimum throughput of Θ^τ. A number of specific models can be specified based on the above generic model:

- When we set $\mathbf{X} \equiv \mathbf{B}$, the buffer allocation problem (BAP) appears. One extra constraint needs to be added to reflect the integrality condition, $B_i, \forall i \in V$. The objective function is then $Z_{\text{BAP}} = \min \sum_{i \in V} B_i$. This is a model formulation presented in Smith (2004); Smith et al. (2010b).

- The server allocation problem (CAP) appears if we have $\mathbf{X} \equiv \mathbf{c}$. Again, an extra integrality constraint is needed: $c_i, \forall i \in V$. The objective function is then $Z_{\text{CAP}} = \min \sum_{i \in V} c_i$. See Smith et al. (2010a) for more information.

- Combining the server and buffer allocation problems by setting $\mathbf{X} \equiv (\mathbf{B}, \mathbf{c})$ results in the joint buffer and server allocation problem (BCAP). In this case, the integrality constraints are $B_i, c_i, \forall i \in V$. Next to this integrality constraint, one more constraint is needed. It is necessary to ensure that there is at least one server per vertex, $c_i \geq 1, \forall i \in V$. Note that buffers can be equal to zero, hence having a zero-buffer system. Secondly, note that the objective function needs to be adapted slightly to take into account the two objectives (*i.e.* buffers and servers). We consider two options to rewrite the objective function depending on how to deal with the multi-objective issue.

  First, the objective function can be written as a weighted sum of the two objectives, $Z_{\text{BCAP1}} = \min \omega \sum_{i \in V} c_i + (1 - \omega) \sum_{i \in V} B_i$. We assign a cost of ω to servers and $(1 - \omega)$ to buffers. We can then modify the value of ω, such that $0 < \omega < 1$, to reflect the relative cost of servers versus buffers. As ω is decreased, the cost of servers will become relatively lower than that of buffers. That is, buffers are then more expensive than servers. Alternatively, when the value of ω is increased, the servers become more costly relative to the buffers and therefore the servers become more expensive than the buffers. In this way, we evaluate whether different pricing of servers and buffers results in a significantly different buffer and server allocation. It is worthwhile to mention that if $\omega = 0$, the above problem reduces to the pure buffer allocation problem and if $\omega = 1$, the pure server allocation problem is obtained.

  Secondly, the objective function can formulated in a multi-criteria way:

  $$Z_{\text{BCAP2}} = \min \Big[f_1(\mathbf{c}), f_2(\mathbf{B})\Big], \qquad (13)$$

  in which each objective is considered explicitly. Consequently, one obtains an approximation of the Pareto set of solutions for the two objectives. As such, this perspective is more general than the first objective function formulation. For more information, see van Woensel et al. (2010).

- A slightly different formulation, is the routing allocation problem (RAP). Here the routing probabilities α_i are determined such that the throughput is maximized. Of course, the sum of all routing probabilities α_i leaving each vertex $i \in V$ should sum up to one.

$$Z_{\text{RAP}} = \max \Theta, \tag{14}$$

subject to:

$$0 \leq \alpha_i \leq 1, \forall i \in V, \tag{15}$$

$$\sum_i \alpha_i = 1, \forall i \in V. \tag{16}$$

The throughput will thus be affected by the effective routings of jobs through the network, the variability of the service distribution, the number of servers and the number of buffers.

- A last variation considered is the profit maximization model. The models are thus expanded with financial indicators in order to maximize the profit generated. This profit will be a function of the quantity one can set in the market (i.e. throughput $\Theta(\mathbf{X})$) and the costs to realize this throughput, which could be the buffer and/or server allocation. The decision variable is thus the investment in buffers or servers ($\mathbf{X}$). Assume the cost of the buffers or servers is γ and the gain of a unit of throughput is equal to ϕ. Then we can formulate the objective function as follows:

$$Z_{\text{profit}} = \max \left[\phi\Theta(\mathbf{X}) - \gamma \sum_{\mathbf{i} \in \mathbf{V}} \mathbf{X_i} - \beta \lceil \mathbf{\Theta}^\tau - \mathbf{\Theta}(\mathbf{X}) \rceil \right], \tag{17}$$

in which $\lceil \Theta^\tau - \Theta \rceil$ is either positive or zero. Penalty costs of size β are charged when the system throughput does not meet the market demand Θ^τ. Penalty costs can include the cost of outsourcing production to another factory. Figure 3 displays the behavior of this optimization function for $\Theta^\tau = 8$, it shows the achieved throughput at the optimal buffer allocation for different cost settings. When the operational expense increases ($X = \gamma/\phi$), it is more attractive to underachieve market demand and the optimal throughput decreases. When the penalty costs increase ($Y = \beta/\phi$), it becomes less attractive to underachieve market demand and the optimal throughput increases.

It is worthwhile to state that the models described above are difficult non-linear integer programming problem. Considering the BCAP model, it can be shown that for a network with N nodes, the complexity involved is:

$$\left[\frac{K(K+1)}{2} \right]^N . \tag{18}$$

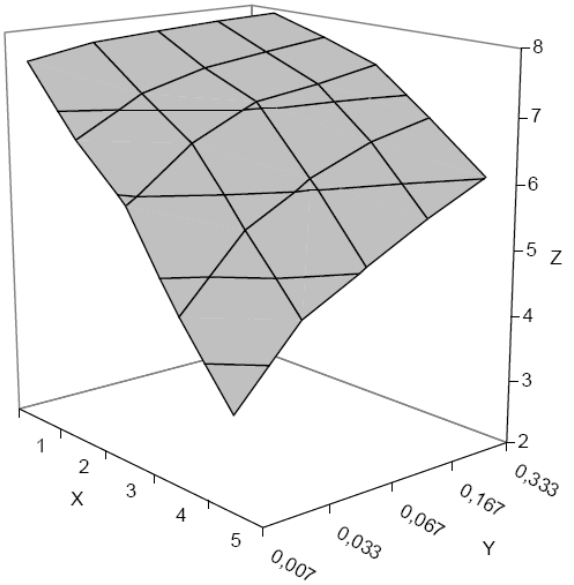

Figure 3: Achieved throughput at the optimal buffer allocation for $\Theta^\tau = 8$

Clearly, the solution space grows exponentially in the number of nodes, but not (exponentially) in the capacity of each node. The complexity of the BCAP model can thus be written as $O(K^N)$.

3.1 Optimization Methodologies

While the GEM computes the performance measures for the queueing network, many of the above discussed models need to be optimized on the decision variables defined in $\mathbf{X}$. Note that there, of course, exist many optimization methods. An exhaustive discussion is left out of this paper, but the interested reader is referred to Aarts and Lenstra (2003) and the references therein. We describe two methodologies which have proven to be successful for the above described models, the Powell algorithm and a Genetic Algorithm approach. 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. Numerical examples have shown that the method is capable of minimizing a function with up to twenty variables (Powell, 1964; Himmelblau, 1972)). Powell's method locates the minimum of $f(\mathbf{x})$ of a non-linear function 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. 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 sub-directions of the search.

Genetic algorithms (GA's) are optimization algo-

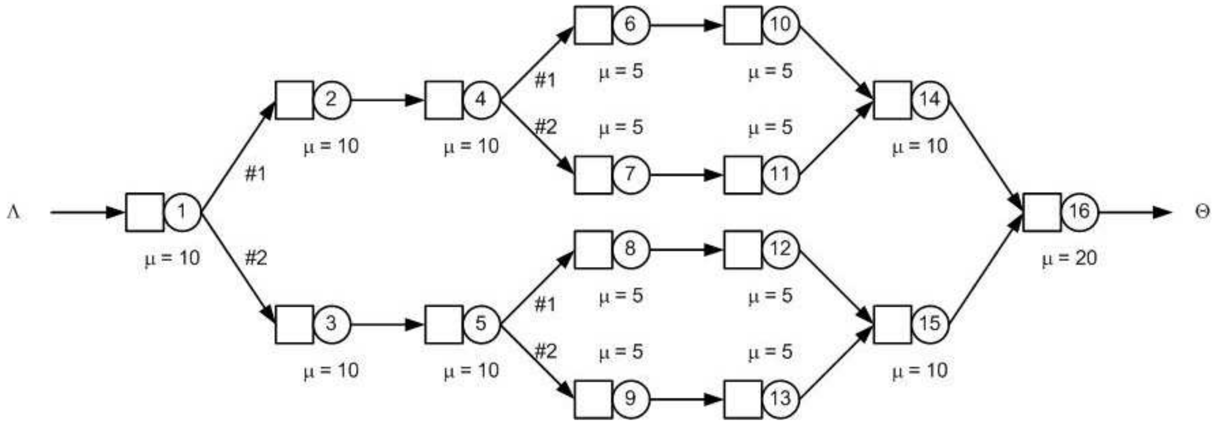

Figure 4: Combined topology

rithms to perform an approximate global search relaying on the information obtained from the evaluation of several points in the search space and obtaining a population of these points that converges to the optimum through the application of the genetic operators *mutation*, *crossover*, *selection*, and *elitism*. Each of these operators may be implemented in several different ways, each one of them characterizing a specific instance of GA. Additionally, convergence of GA's is guaranteed by assigning fitness to each population member and preserving diversity at the same front. For instance, recent successful applications of GA's were reported by Lin (2008) and Calvete et al. (2008), for single-objective applications, and by Carrano et al. (2006), for multiple-objective applications. A wealth of references is given by these authors. For a direct application of the GA's to manufacturing problems, see Andriansyah et al. (2010).

4 Results and Insights

In this section, we will focus on one example network and describe the results for some of the different optimization models discussed above.

We consider a combination of the three basic topologies (series, split and merge), as shown in Fig. 4. This network consist of 16 nodes with the processing rate of servers in each node given in the figure. The network is adopted from Smith and Cruz (2005). We use exactly the same values for Λ, μ, s^2, and routing probabilities for the splitting node (#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 Fig. 4 for the position of each node in the network.

The Buffer Allocation Problem (BAP)

We reproduce in Table 2 the results from Smith and Cruz (2005) for this network structure with $\Lambda = 5$ and the routing probabilities equal to 0.5 (Table 29 in their paper). The results are in Table 2. Note that they considered an $M/G/1/K$ setting and therefore the number of servers in all nodes is set to 1 while optimizing on the buffer allocation. Based on the table, we see that the first node (most congested) is receiving more buffers to cope with the relatively high arrival rate.

The Server Allocation Problem (CAP)

Let us now fix the number of buffers beforehand and then optimize on the number of servers used. More specifically, we set all buffers equal to 1 and look at the resulting server allocation (Table 3). Interestingly, we observe the same behavior as for the buffer allocation. The first node is receiving more resources than the remaining nodes. On the other hand, the number of servers added is relatively low compared to the buffers added (5 versus 8). This is because a server is also acting as a buffer, but a server adds more value, measured in throughput.

The joint buffer-server allocation (BCAP)

Before going to the results for the example network, we analyze the difference between buffers and servers. We saw that the BAP and CAP give different results in terms of number of servers versus number of buffers used.

Let us assume that we have a zero-buffer node with one server (*i.e.*, $K = 1$, $B = 0$, and $c = 1$), submitted to an external arrival rate $\Lambda = 1.0$, service rate $\mu = 2.0$ and a squared coefficient of variation of the service time distribution $s^2 = \{0.5, 1.0, 1.5\}$. Figure 5 gives the percentage increase of adding either a server (adding one to four servers compared to the base case) or a buffer (adding one to seven buffers compared to the base case) to the zero buffer base situation.

It is clear that in this case, the first added buffer or first added server gives the largest contribution to the throughput value, which is limited by the arrival rate λ. Note that the addition of the first extra server, gives an increase in throughput of about 13.4% to 18.6% depending upon the coefficient of variation s^2, while the first added buffer only gives a 8.2% to 9.2% increase. Important to mention is that, in order to achieve the same in-

Table 2: Results for the BAP (Smith and Cruz, 2005)

s^2	$\mathbf{c}$	$\mathbf{B}$	$\sum_i c_i$	$\sum_i B_i$	$\theta(\mathbf{c},\mathbf{B})$
0.5	(1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1)	(8 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5)	16	69	4.9899
1.0	(1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1)	(10 5 5 5 5 4 4 4 4 4 4 4 4 5 5 5)	16	77	4.9879
1.5	(1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1)	(11 5 5 5 5 5 5 5 5 5 5 5 5 5 5 6)	16	87	4.9877

Table 3: Results for the CAP

s^2	$\mathbf{c}$	$\mathbf{B}$	$\sum_i c_i$	$\sum_i B_i$	$\theta(\mathbf{c},\mathbf{B})$	Z_α
0.5	(5 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2)	(1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1)	34	16	4.9997	35.29
1.0	(5 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2)	(1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1)	36	16	4.9996	35.33
1.5	(5 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2)	(1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1)	34	16	4.9996	35.37

crease in throughput by only using buffers, we need five to six extra buffer spaces, rather than only one server space.

The results for the joint buffer-server allocation are presented in Table 4. In Table 2, the c/B price ratio gives an indication of the relative costs of servers compared to buffers. A price ratio of 8:1, for example, means that servers are 8 times more expensive than buffers. The results from Table 4, show a higher throughput than for the pure Buffer Allocation Problem, Table 2, for every setting. As expected, we found that the optimal server allocation in the BCAP is different from the server settings in the pure BAP. This however depends strongly upon the price ratio of buffers versus servers. We found that $M/G/1/K$ is not an optimal configuration for this particular queueing network structure, except when buffers are becoming relatively too expensive. For these cases, we found that single-servers are optimal indeed (see rows where c/B ratio is 8:1).

We observe that (near) zero-buffer configurations are identified where appropriate, *i.e.*, where the servers are relatively cheaper compared to buffers. Varying the coefficient of variation does result in some changes in the optimal server and buffer allocation, which shows the importance of models dealing with general service times. The results show that the number of buffers seem to be larger with higher variability, which could be expected, since the increase in the squared coefficient of variation means a high variability. The extra buffers are there to handle this increased variability.

5 Practical Issues

In a number of industrial improvement projects carried out, we observed that the critical issue to be able to use the above 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 and 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.

Kock (2008) propose an EPT approach in four steps (see Figure 6). 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.

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.

6 Conclusions and Future Research Suggestions

This review provided an overview of the different modeling issues, the performance evaluation and optimization approaches of the manufacturing systems assuming a queueing theory approach. We discussed the merits of the GEM as a performance evaluation tool of the finite queueing networks. This methodology has proved in the literature to be a valuable approach. Secondly, different optimization models are discussed, the buffer allocation problem, the server allocation problem, the joint buffer and server allocation problem and some other models. The different optimization models are shown to be hard non-linear integer programming problems which are able to 'solved' with a Powell

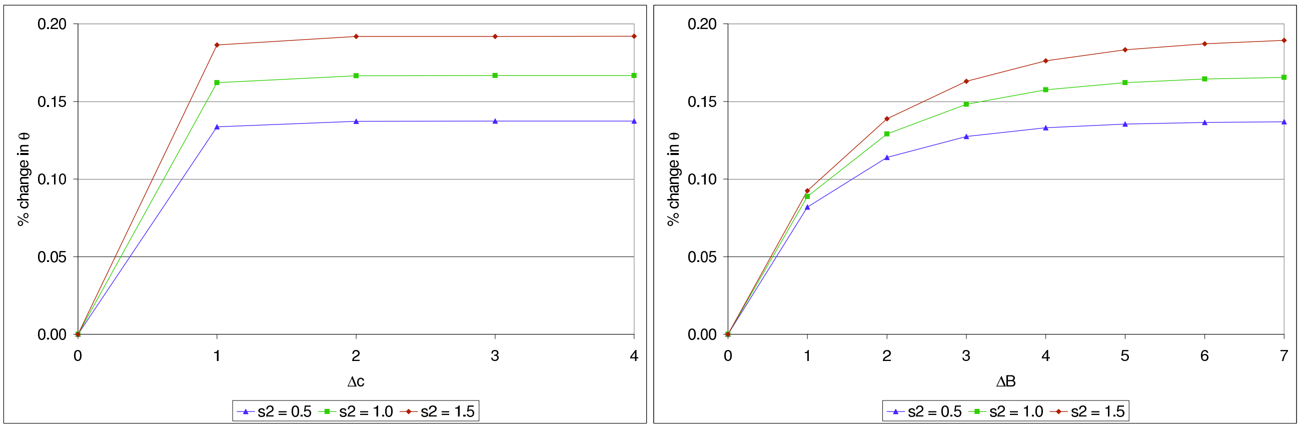

Figure 5: Throughput increase versus added number of buffers and servers

Table 4: Results for the BCAP

Λ	s^2	c/B	$\mathbf{c}$	$\mathbf{K}$	$\sum_i c_i$	$\sum_i K_i$	$\sum_i B_i$	$\theta(\mathbf{c}, \mathbf{B})$	Z_α
5.0	0.5	1:8	(3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3)	(3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3)	48	48	0	4.9996	5.76
		1:4	(3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3)	(3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3)	48	48	0	4.9996	10.0
		1:2	(3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3)	(3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3)	48	48	0	4.9996	16.4
		1:1	(5 3 3 3 3 2 2 2 2 2 2 2 2 3 3 5)	(5 3 3 3 3 2 2 2 2 2 2 2 2 3 3 5)	44	44	0	4.9998	22.2
		2:1	(2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2)	(5 3 3 3 3 2 2 2 2 2 2 2 2 3 3 5)	32	44	12	4.9989	26.5
		4:1	(3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3)	(3 5 5 5 5 3 3 3 3 3 3 3 3 5 5 3)	20	60	40	4.9974	26.6
		8:1	(1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1)	(11 6 6 6 6 4 4 4 4 4 4 4 4 6 6 11)	16	90	74	4.9994	23.0
	1.0	1:8	(3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3)	(3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3)	48	48	0	4.9994	5.94
		1:4	(5 3 3 3 3 2 2 2 2 2 2 2 2 3 3 5)	(5 3 3 3 3 2 2 2 2 2 2 2 2 3 3 5)	44	44	0	4.9997	9.09
		1:2	(5 3 3 3 3 2 2 2 2 2 2 2 2 3 3 5)	(5 3 3 3 3 2 2 2 2 2 2 2 2 3 3 5)	44	44	0	4.9997	15.0
		1:1	(5 3 3 3 3 2 2 2 2 2 2 2 2 3 3 5)	(5 3 3 3 3 2 2 2 2 2 2 2 2 3 3 5)	44	44	0	4.9997	22.3
		2:1	(3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3)	(3 3 3 3 3 2 2 2 2 2 2 2 2 3 3 3)	34	40	6	4.9984	26.2
		4:1	(2 2 2 2 2 1 1 1 1 1 1 1 1 2 2 3)	(6 3 3 3 3 4 4 4 4 4 4 4 4 3 3 4)	25	60	35	4.9989	28.1
		8:1	(1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1)	(13 6 6 6 6 4 4 4 4 4 4 4 4 6 6 13)	16	94	78	4.9987	24.1
	1.5	1:8	(5 3 3 3 3 2 2 2 2 2 2 2 2 3 3 5)	(5 3 3 3 3 2 2 2 2 2 2 2 2 3 3 5)	44	44	0	4.9996	5.24
		1:4	(5 3 3 3 3 2 2 2 2 2 2 2 2 3 3 5)	(5 3 3 3 3 2 2 2 2 2 2 2 2 3 3 5)	44	44	0	4.9996	9.15
		1:2	(5 3 3 3 3 2 2 2 2 2 2 2 2 3 3 5)	(5 3 3 3 3 2 2 2 2 2 2 2 2 3 3 5)	44	44	0	4.9996	15.0
		1:1	(5 3 3 3 3 2 2 2 2 2 2 2 2 3 3 5)	(5 3 3 3 3 2 2 2 2 2 2 2 2 3 3 5)	44	44	0	4.9996	22.4
		2:1	(3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3)	(3 3 3 3 3 2 2 2 2 2 2 2 2 3 3 3)	34	40	6	4.9979	26.8
		4:1	(2 2 2 2 2 1 1 1 1 1 1 1 1 2 2 3)	(6 3 3 3 3 4 4 4 4 4 4 4 4 3 3 4)	25	60	35	4.9983	28.7
		8:1	(1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1)	(15 7 7 7 7 4 4 4 4 4 4 4 4 7 7 15)	16	104	88	4.9986	25.4

heuristic. The paper ended with an overview of some results for the different models considered on a complex queueing network.

Future Research Suggestions

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 cycle time, WIP or other performance measures.

The advantage of the EPT approach is that various types of disturbances on the shop-floor are aggregated into EPT distributions, this enables 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.

Topics for future research on the queueing part, include the analysis and optimization of networks with cycles, *e.g.*, to model many important industrial systems that have loops, such as systems with captive pallets and fixtures or reverse streams of products due to rework, or even the extension to $GI/G/c/c$ queueing networks, *i.e.* including generally distributed and independent arrivals.

Acknowledgments

The research of prof. Frederico Cruz has been partially funded by CNPq (*Conselho Nacional de Desenvolvimento Científico e Tecnológico*) of the Ministry for Science and Technology of Brazil, grants 201046/1994-6, 301809/1996-8, 307702/2004-9, 472066/2004-8, 472877/2006-2, and 304944/2007-6, by CAPES (*Coordenação de Aperfeiçoamento de Pessoal de Nível Superior*), grant BEX-0522/07-4, by FAPEMIG (*Fundação de Amparo à Pesquisa do Estado de Minas Gerais*), grants CEX-289/98,

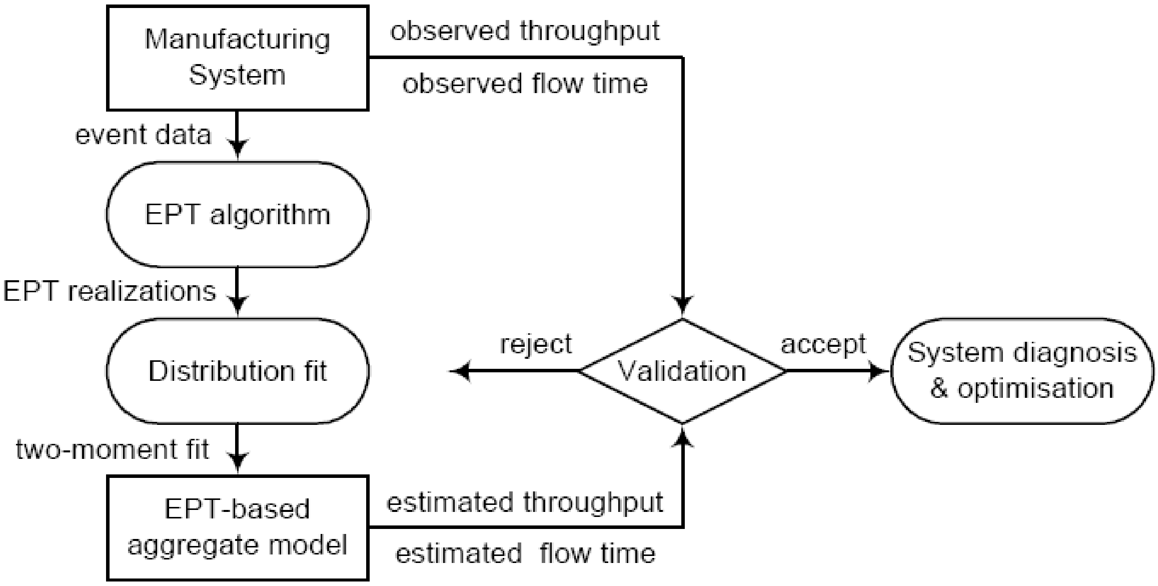

Figure 6: The effective processing time approach

CEX-855/98, and TEC-875/07, and by PRPq-UFMG, grant 4081-UFMG/RTR/FUNDO/PRPq/99.

References

Aarts, E.H.L. and J.K.Lenstra (2003). Local Search in Combinatorial Optimization (2nd revised edition). Princeton NJ, USA: Princeton University Press, 536 pp.

Andriansyah, R., van Woensel, T., Cruz, F. R. B., Duczmal, L., 2010. Performance optimization of open zero-buffer multi-server queueing networks. Computers & Operations Research 37 (8), 1472–1487.

Balsamo S. de Nitto Personé V. and R. Onvural, 2001. Analysis of Queueing Networks with Blocking. Kluwer Academic Publishers.

Buzacott J. and J.G. Shanthikumar, 1993. Stochastic Models of Manufacturing Systems, Prentice-Hall.

Calvete, H. I., Gale, C., Mateo, P. M., 2008. A new approach for solving linear bilevel problems using genetic algorithms. European Journal of Operational Research 188 (1), 14–28.

Carrano, E. G., Soares, L. A. E., Takahashi, R. H. C., Saldanha, R. R., Neto, O. M., April 2006. Electric distribution network multiobjective design using a problem-specific genetic algorithm. IEEE Transactions on Power Delivery 21 (2), 995–1005.

Cruz, F. R. B., Duarte, A. R., Van Woensel, T., 2008. Buffer allocation in general single-server queueing network. Computers & Operations Research 35 (11): 3581–3598.

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

Gross D. and C.M. Harris, 1998. Queueing Theory, John Wiley and sons.

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

Hopp WJ and ML Spearman (1996) *Factory Physics, Foundations for Manufacturing Management*. Mc-Graw Hill.

Jackson J.R., 1957. Networks of Waiting Lines, Operations research, 5 (4), 518–521.

Jackson J.R., 1963. Jobshop-Like Queueing Systems, Management science, 10 (1), 131–142.

Jain, S., Smith, J. M., 1994. Open finite queueing networks with $M/M/C/K$ parallel servers. Computers & Operations Research 21 (3), 297–317.

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.

Kerbache, L., Smith, J. M., 2000. Multi-objective routing within large scale facilities using open finite queueing networks. European Journal of Operational Research 121 (1), 105–123.

Kimura, T., 1996. A transform-free approximation for the finite capacity $M/G/s$ queue. Operations Research 44 (6), 984–988.

Kock A.A.A.,Etman, L.F.P., Rooda J.E., 2008. Effective process times for multi-server flowlines with finite buffers, IIE Transactions, Volume 40, Issue 3, 177–186.

Lin, F.-T., 2008. Solving the knapsack problem with imprecise weight coefficients using genetic algorithms. European Journal of Operational Research 185 (1), 133–145.

Reiser M. and H. Kobayashi, 1974. Accuracy of the Diffusion Approximation for some Queueing Systems, IBM Journal of Research and development, 18, 110–124.

Kuehn P.J., 1979. Approximate Analysis of General Queueing Networks by Decomposition, IEEE Trans. Comm, 27, 113–126.

Labetoulle, J., Pujolle, G., 1980. Isolation method in a network of queues. IEEE Transactions on Software Engineering SE-6 (4), 373–381.

Law A.M. and W.D. Kelton, 2000. Simulation Modeling and Analysis, McGraw Hill.

Perros H.G., 1994. Queueing Networks with Blocking, Oxford University Press.

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.

Simchi-Levi D., P. Kaminsky and E. Simchi-Levi, Designing and Managing the Supply Chain: Concepts, Strategies and Case Studies, McGrawHill.

Smith, J. M., 2003. $M/G/c/k$ blocking probability models and system performance. Performance Evaluation 52 (4), 237–267.

Smith, J. M., 2004. Optimal design and performance modelling of $M/G/1/K$ queueing systems. Mathematical and Computer Modelling 39 (9-10), 1049–1081.

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

Smith, J. M., Cruz, F. R. B., van Woensel, T., 2010a. Optimal server allocation in general, finite, multi-server queueing networks. Applied Stochastic Models in Business & Industry 26 (6), 705–736.

Smith, J. M., Cruz, F. R. B., van Woensel, T., 2010b. Topological network design of general, finite, multi-server queueing networks. European Journal of Operational Research 201 (2), 427–441.

Spinellis, D., Papadopoulos, C. T., Smith, J. M., 2000. Large production line optimization using simulated annealing. International Journal of Production Research 38 (3), 509–541.

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

Tempelmeier, H., 2003. Practical considerations in the optimization of flow production systems. International Journal of Production Research 41 (1), 149–170.

van Woensel, T., Andriansyah, R., Cruz, F. R. B., Smith, J. M., Kerbache, L., 2010. Buffer and server allocation in general multi-server queueing networks. International Transactions in Operational Research 17 (2), 257–286.

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

Whitt W., 1984. Open and closed models for networks of queues, AT&T Bell Laboraties Technical Journal, 63:9, 1911–1979.