Proceeding Series of the Brazilian Society of Applied and Computational Mathematics, Vol. 2, N. 1, 2014.
Trabalho apresentado no CMAC-Sul, Curitiba-PR, 2014.

Buffers and Servers Allocation in General Finite Queueing Networks

F. R. B. Cruz
Departamento de Estatística, Universidade Federal de Minas Gerais,
31270-901, Belo Horizonte, MG, Brazil
E-mail: fcruz@est.ufmg.br

T. van Woensel
Operations, Planning, Accounting and Control (OPAC) Group, School of Industrial Engineering,
Technische Universiteit Eindhoven, The Netherlands
t.v.woensel@tue.nl

Abstract: *This paper presents preliminaries results for the resource allocation problem framed in a joined manufacturing, product engineering, and service environment. Such networks are represented as queueing networks. The performance of the queueing 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, namely the buffer and server sizes.*
Keywords: *Queues, Networks, Performance evaluation, Optimization*

1 Introduction

The resource allocation problem has been the focus of numerous studies for decades. The focus here is on problems modeled as queueing networks [16]. This paper presents 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 *e.g.* the throughput of the network. Following Simchi-Levi et al. [11], we adopt the same approach as in Figure 1. Clearly, the development chain (product engineering) and the supply chain (product manufacturing) are interacting with the service network. In this paper, we focus on the service step as it will be important to consider its role, its characteristics and the consequences in the product engineering phase.

The paper is structured as follows. In Section 2, we present the problem formulation, the performance evaluation method considered for the queueing networks, which is generalized expansion (GEM) method, and we elaborate on the optimization tools that are used to optimize the models. Section 3 gives for a complex network the results for some selected optimization models. The last Section 4 concludes the paper and gives some pointers for future research in the area.

[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.

DOI: 10.5540/03.2014.002.01.0108

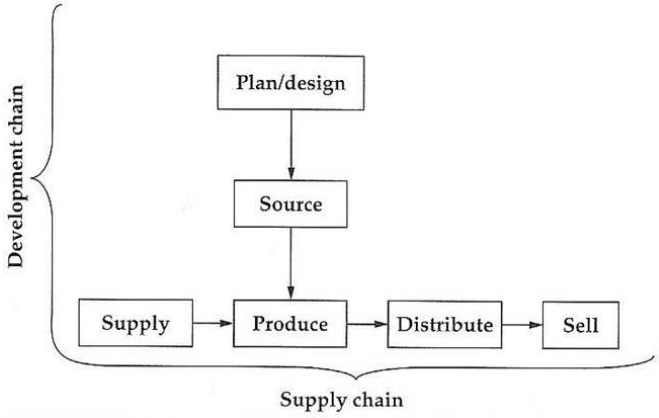

Figure 1: The development chain versus the supply chain

2 Problem Formulation and Resolution Method

2.1 Mathematical Programming Formulation

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:

$$Z = \min f(\mathbf{X}), \tag{1}$$

subject to:

$$\begin{aligned} \Theta(\mathbf{X}) &\geq \Theta^{\tau}, & (2)\\ \mathbf{X} &\geq 0, & (3) \end{aligned}$$

that minimizes the total allocation $f(\mathbf{X}) \equiv \sum_{i \in V} X_i$ (*i.e.* over all nodes $i \in V$), constrained to provide at least the minimumm required 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, that is, $B_i \in \mathcal{N}, \forall i \in V$. The objective function is then

$$Z_{\text{BAP}} = \min \sum_{i \in V} B_i. \tag{4}$$

  This is a model formulation discussed in details by Smith et al. [15].

- The server allocation problem (CAP) appears if we have $\mathbf{X} \equiv \mathbf{c}$. Again, an extra integrality constraint is needed, that is $c_i \in \mathcal{N}, \forall i \in V$. The objective function is then

$$Z_{\text{CAP}} = \min \sum_{i \in V} c_i. \tag{5}$$

  See the paper by Smith et al. [14] for more information.

DOI: 10.5540/03.2014.002.01.0108

- 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 $B_i \in \mathcal{N}, c_i \in \mathcal{N}, \forall i \in V$. 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:

$$Z_{\text{BCAP1}} = \min \omega \sum_{i \in V} c_i + (1 - \omega) \sum_{i \in V} B_i. \tag{6}$$

  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:

$$Z_{\text{BCAP2}} = \min \Big\{ f_1(\mathbf{c}), f_2(\mathbf{B}) \Big\}, \tag{7}$$

  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. A discussion on this multiobjective formulation can be found in the paper by van Woensel et al. [17], Cruz et al. [3, 4].

2.2 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 Generalized Expansion Method (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 [7, 8, 9], Jain and Smith [6], Smith [12], and Andriansyah et al. [2]. 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 [7], 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.

[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).

DOI: 10.5540/03.2014.002.01.0108

2.3 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 [1] and the references therein. We describe a methodology which has proven to be successful for the above described models, the Powell algorithm. 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 (see Powell [10] and Himmelblau [5]). 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.

3 Results

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 Figure 2. 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 [13]. 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 Figure 2 for the position of each node in the network.

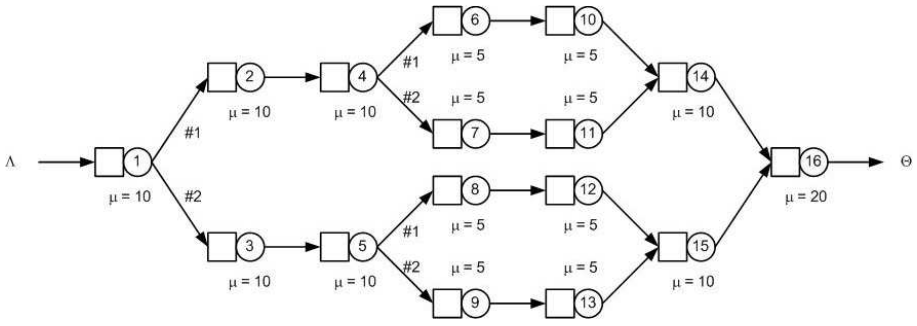

Figure 2: Combined topology

3.1 The Buffer Allocation (BAP)

We reproduce in Table 1 the results for the BAP, Equation (4), taken from Smith and Cruz [13], 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 1. 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.

Table 1: Results for the BAP (taken from Smith and Cruz [13])

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 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 5 6)	16	87	4.9877

3.2 The Server Allocation (CAP)

Let us now fix the number of buffers beforehand and then optimize on the number of servers used, the CAP, Equation (5). More specifically, we set all buffers equal to 1 and look at the resulting server allocation (Table 2). 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.

Table 2: 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

4 Conclusions and Future Directions

In this paper we have reviewed some of the most important models for resource allocation from a queueing point of view. We discussed the merits of the GEM, as a performance evaluation tool of the finite queueing networks, and the Powell method, as an heuristic optimization tool. The paper also presented results for a complex queueing network. We saw that the BAP and CAP give different results in terms of number of servers versus number of buffers used. 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 a certain amount of increase in throughput, while the first added buffer gives a smaller increase. Important to mention is that, in order to achieve the same increase in throughput by only using buffers, we need more extra buffer spaces, rather than only one server space.

We have not considered here but 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, the work-in-process (WIP)or other performance measures. Topics for future research on the area 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 re-work, or even the extension to $GI/G/c/c$ queueing networks, *i.e.* including generally distributed and independent arrivals.

Acknowledgments

The research of Prof. Cruz has been partially funded by the Brazilian agencies, CNPq, CAPES, and FAPEMIG.

References

[1] Aarts, E., Lenstra, J. K., 2003. Local Search in Combinatorial Optimization, 2nd Edition. Princeton University Press, Princeton, NJ.

[2] 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.

[3] Cruz, F. R. B., Duarte, A. R., Brito, N. L. C., 3. Multiobjective optimization of finite queueing networks. In: Congresso de Matemtica Aplicada e Computacional - CMAC-CO 2013 [CD-ROM]. Cuiabá, Brasil, pp. 1–4.

[4] Cruz, F. R. B., Kendall, G., While, L., Duarte, A. R., Brito, N. L. C., 2012. Throughput maximization of queueing networks with simultaneous minimization of service rates and buffers. Mathematical Problems in Engineering 2012 (Article ID 348262), 19 pages.

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

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

[7] Kerbache, L., Smith, J. M., 1987. The generalized expansion method for open finite queueing networks. European Journal of Operational Research 32, 448–461.

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

[9] 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.

[10] 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.

[11] Simchi-Levi, D., Kaminsky, P., Simchi-Levi, E., 2008. Designing and Managing the Supply Chain: Concepts, Strategies and Case Studies. McGraw-Hill.

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

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

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

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

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

[17] 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.

