Anais do XX Congresso Brasileiro de Automática
Belo Horizonte, MG, 20 a 24 de Setembro de 2014

# AN ALGORITHM FOR OPTIMAL ROUTING IN FINITE MULTISERVER QUEUEING NETWORKS

F. R. B. Cruz*, T. van Woensel†

*Departamento de Estatística,
Universidade Federal de Minas Gerais,
31270-901 - Belo Horizonte - MG, Brazil

†Department of Industrial Engineering & Innovation Sciences,
Eindhoven University of Technology,
P.O. Box 513, 5600 MB, Eindhoven, The Netherlands

Emails: fcruz@est.ufmg.br, t.v.woensel@tue.nl

**Abstract**— In this paper, we examine the optimal routing problem in acyclic finite general-service queueing networks. The optimization is done by means of a heuristics based on Powell algorithm coupled with a known approximate performance evaluation method. The proposed algorithm is then applied to determine the optimal routing probability vector that maximizes the throughput of the queueing network. We show preliminary numerical results to quantify the quality of the routing vector approximations obtained.

**Keywords**— Optimal Routing Problems; Queueing Networks; General Service.

## 1 Introduction and Motivation

There are several distinct *network design* optimization problems associated with finite queueing networks. Following Daskalaki and MacGregor Smith (2004) the optimal network design problem can be split up into three related optimization problems below.

1. The optimal topology problem (OTOP) deals with decisions of the design of the network itself, that is, the number of nodes (*e.g.* workstations, warehouses, delivery points, *etc.*) and arcs (*e.g.* corridors, conveyors, escalators, *etc.*) and the general configuration of these two components;

2. The optimal routing problem (OROP) deals with determining the routing probabilities in the network defined by the first problem;

3. Finally, the optimal resource allocation problem (ORAP) deals with the optimal allocation of the scarce resources in the network, *e.g.* the number of buffers (*i.e.*, the buffer allocation problem, BAP) and the number of servers (*i.e.*, the server allocation problem, CAP).

These three problems are challenging and difficult optimization problems. For an arbitrary topology, the OTOP is shown to be $\mathcal{NP}$-hard (Garey and Johnson, 1979). The same $\mathcal{NP}$-hardness is conjectured for the general ORAP (MacGregor Smith and Daskalaki, 1988).

Previous work focused mainly on the ORAP in open finite acyclic queueing network settings. Both BAP and CAP are probably one of the most well-known optimal resource allocation problems (Dallery and Gershwin, 1992). For instance, MacGregor Smith et al. (2010b) looked into the BAP, both in a single and in a multiserver setting, and MacGregor Smith et al. (2010a) proposed algorithms to solve the CAP. However, the routing probabilities are usually assumed to be known beforehand for BAP and CAP.

In this paper the focus is specifically in solving the OROP being the overall objective to maximize the system throughput by optimizing the routing probabilities through the queueing network. A similar research question has been tackled by Daskalaki and MacGregor Smith (2004) in which the joint effect of buffer allocation and routing on the throughput was evaluated. Earlier, Gosavi and MacGregor Smith (1997) focused on the routing optimization problem related to the overall objective of throughput maximization. The common ground of both papers is that they used queueing networks with *single* servers while in this paper we examine the OROP for *multiserver* queues.

The algorithm presented here is a heuristic based on the Powell method (Himmelblau, 1972). Notice that the Powell technique is not the unique but just a first approach for the multiserver OROP and a basis for further improvements in the area. The algorithm is specific for acyclic networks of $M/G/c/K$ queues, which in Kendall notation means a queueing system with **M**arkovian arrival rates, **G**eneral service times, **c** parallel servers, and a total capacity of **K** items (*including* those in service). Practical applications to $M/G/c/K$ queueing networks include manufacturing and service systems (MacGregor Smith, 2008) and transportation and material handling systems (Bedell and MacGregor Smith, 2012).

This paper is organized as follows. In Section 2 we describe in detail the mathematical model formulation for the routing problem. The optimization algorithm is presented in Section 3

when we elaborate further on both the optimization procedure and the performance evaluation tool. Section 4 gives preliminary computational results for some test networks. Finally, Section 5 closes the paper with some conclusions and final remarks.

## 2 Model Formulation

Mathematically the optimal routing problem can be formulated on a digraph $\mathcal{D} = (V, A)$ as follows, in which $V$ is the set of vertexes (finite queues) and $A$ is the set of arc (connections between the queues). Each vertex (queue) is characterized by Poisson arrivals, general service, and multiservers. The mathematical programming formulation is as follows.
(OROP):

$$\max \Theta(\boldsymbol{\alpha}), \tag{1}$$

subject to:

$$0 \leq \alpha_{i,j} \leq 1, \qquad \forall\, (i,j) \in A, \tag{2}$$

$$\sum_{\forall j \in \delta(i)} \alpha_{i,j} = 1, \qquad \forall\, i \in V, \tag{3}$$

in which $\Theta(\boldsymbol{\alpha})$ is the throughput, which is the objective that must be maximized, $\boldsymbol{\alpha}$ the optimal routing probability matrix, $\boldsymbol{\alpha} \equiv [\alpha_{i,j}]$, *i.e.* the matrix that maximizes the objective function of this predefined network, and $\delta(i)$ is the set of succeeding vertexes of vertex $i$, that is, $\delta(i) \equiv \{j|\ (i,j) \in A\}$.

The throughput will thus be affected by the effective routing of jobs through the network, the variability of the service distribution, the number of servers, and the number of buffers. It should be clear that the above described model is a highly difficult stochastic optimization problem to handle due to the inherent non-linearity of the objective function combined with the absence of any closed-form expression for the throughput $\Theta(\boldsymbol{\alpha})$.

## 3 Proposed Algorithm

### *3.1 Optimization Procedure*

The algorithm to solve the OROP is presented in Figure 1. The initial routing probability vector is conveniently set to the inverse of the number of nodes after a split,

$$\alpha_{i,j}^{(\text{init})} = \frac{1}{n_i},\ \forall (i,j) \in A,$$

in which $n_i$ is the number of succeeding nodes to node $i$, that is, the cardinality of set $\delta(i)$ as defined earlier. The optimization step itself is an iteration in which new solutions are generated following Powell (1964) logic until convergence, that is, until the difference in $\Theta$, $\Delta\Theta \equiv (\Theta^{(k)} - \Theta^{(k-1)})$, is less than a predefined tolerance $\varepsilon$.

**algorithm**
*/* Step 1: Initialization */*
1.1 **read** $\mathcal{D}(V, A)$
*/* initialize the routing probabilities */*
1.2 $k \leftarrow 0$
1.3 $\alpha_{i,j}^{(k)} = \alpha_{i,j}^{(\text{init})},\ \forall (i,j) \in A$
*/* evaluate with GEM */*
1.4 $\Theta^{(\text{k})} \leftarrow \Theta(\boldsymbol{\alpha}^{(k)})$
*/* Step 2: Optimization & Evaluation */*
*/* generate new solution using Powell */*
2.1 $k \leftarrow k + 1$
2.2 $\alpha_{i,j}^{(k)} \leftarrow \text{Powell}\left(\alpha_{i,j}^{(k-1)}, \Theta^{(k-1)}\right),\ \forall (i,j) \in A$
*/* evaluate with GEM */*
2.3 $\Theta^{(k)} \leftarrow \Theta(\boldsymbol{\alpha}^{(k)})$
2.4 **if** $|\Theta^{(k)} - \Theta^{(k-1)}| > \varepsilon$ **then goto** 2.1
*/* Step 3: Print Results */*
3.1 **print** $\boldsymbol{\alpha}^{(k)}$ and $\Theta^{(k)}$
**end algorithm**

Figure 1: Optimization algorithm

Powell (1964) algorithm can be described as an unconstrained optimization procedure that does not require the calculation of first derivatives of the function. Numerical examples has shown that the method is capable of minimizing a function with up to twenty variables (Himmelblau, 1972). Powell algorithm locates the minimum of a non-linear function $f(\mathbf{x})$ by successive unidimensional searches from an initial starting point $\mathbf{x}^{(k)}$ along a set of conjugate directions. These conjugate directions are generated within the procedure itself. 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. We have seen a remarkable success with Powell algorithm coupled with a well-known approximate algorithm for performance evaluation of the finite queueing networks, namely the generalized expansion method (GEM). We will describe it in detail now.

### *3.2 Performance Evaluation*

Described in many papers (Kerbache and MacGregor Smith, 1987; Kerbache and MacGregor Smith, 1988), the GEM has been successfully used to evaluate the performance measures of finite queueing networks. The method is a robust and effective approximation technique that is basically a combination of repeated trials and node-by-node decomposition in which each queue is analyzed separately and then corrections are made in order to take into account the interrelation between

the queues in the network. The method is composed by three stages, *Network Reconfiguration*, *Parameter Estimation*, and *Feedback Elimination*.

The first stage involves a network reconfiguration. That is, an artificial vertex $h_j$ is added preceding each finite vertex $j$ in the network. The artificial vertex is added to register the blocked customers at node $j$ and is modeled as an $M/G/\infty$ queue, as shown in Figure 2, for two queues in tandem.

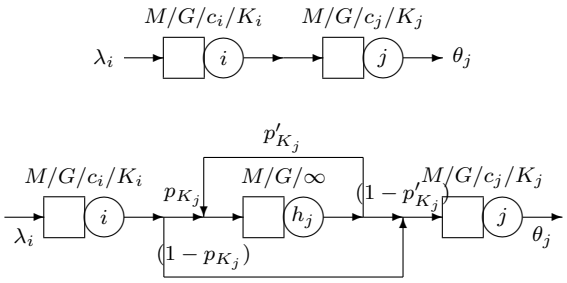



Figure 2: The generalized expansion method

When an entity, arriving at rate $\lambda_i$, leaves vertex $i$, vertex $j$ may be blocked with probability $p_{K_j}$, or unblocked, with probability $(1-p_{K_j})$. Under blocking, the entities are rerouted to vertex $h_j$ for a delay while node $j$ is still busy. Vertex $h_j$ helps to accumulate the time an entity has to wait before entering vertex $j$ and to compute the effective arrival rate to vertex $j$. In other words, the GEM ultimate goal is to provide an approximation scheme to *update* the service rates at the upstream vertex $i$ to take into account all blocking after service caused by the downstream vertex $j$:

$$\tilde{\mu}_i^{-1} = \mu_i^{-1} + p_{K_j}(\mu'_{h_j})^{-1}, \tag{4}$$

in which $\mu_i$ is the service rate at vertex $i$, $p_{K_j}$ is the blocking probability of a finite queue $j$ of size $K_j$, $\mu'_{h_j}$ is the corrected service rate at the artificial vertex $h_j$, and $\tilde{\mu}_i$ is the *updated* (that is, reduced) service rate at vertex $i$.

In the second stage, the parameter $p_K$ (index $j$ is omitted for simplicity), among others, must be estimated which is done essentially utilizing known results for queueing theory. Analytical results from the $M/M/c/K$ queue provide the following expression for the blocking probability $p_K$.

$$p_K = \frac{1}{c^{K-c}c!}\left(\frac{\lambda}{\mu}\right)^K p_0, \tag{5}$$

in which

$$p_0^{-1} = \sum_{n=0}^{c-1}\frac{1}{n!}\left(\frac{\lambda}{\mu}\right)^n + \frac{(\lambda/\mu)^c}{c!}\frac{1-[\lambda/(c\mu)]^{K-c+1}}{1-\lambda/(c\mu)}, \tag{6}$$

for $\lambda/(c\mu) \neq 1$.

However, the interest here is on $M/G/c/K$ queues, for which $p_K$ is not known so far in closed form. Then approximations must be used and Kimura's two moment approximation (Kimura, 1996) has proven to be very effective in similar cases (MacGregor Smith, 2003; MacGregor Smith, 2008). For example, let us fix $c = 2$ and the following closed form expression can be developed from Eq. (5), for the optimal buffer size $B_{\rm M} = K-2$ for Markovian $M/M/2/K$ queues, as a function of the blocking probability:

$$B_{\rm M} = \frac{\ln\left(\frac{1}{2}\frac{p_K(2\mu+\lambda)}{2\mu-\lambda+p_K\lambda}\right)}{\ln(\rho)} - 2. \tag{7}$$

The following Kimura's two moment approximation can be used to approximate the optimal buffer size $B_\epsilon(s^2)$ of a general service $M/G/2/K$ queue:

$$B_\epsilon(s^2) = B_{\rm M} + {\rm NINT}\left(\frac{s^2-1}{2}\sqrt{\rho}B_{\rm M}\right), \tag{8}$$

in which $s^2$ is the squared coefficient of variation of the service time distribution at the queue, $\rho \equiv \lambda/(c\mu)$ is the traffic intensity, $B_{\rm M}$ is the exact buffer size for a respective Markovian system, and ${\rm NINT}(x)$ is the nearest integer to $x$. Now, if we invert Eq. (8) to solve for $p_K$ we achieve:

$$p_K = \frac{2\rho^{2\left(\frac{\sqrt{\frac{\rho}{e}}s^2-\sqrt{\frac{\rho}{e}}+K}{2+\sqrt{\frac{\rho}{e}}s^2-\sqrt{\frac{\rho}{e}}}\right)}(2\mu-\lambda)}{-2\rho^{2\left(\frac{\sqrt{\frac{\rho}{e}}s^2-\sqrt{\frac{\rho}{e}}+K}{2+\sqrt{\frac{\rho}{e}}s^2-\sqrt{\frac{\rho}{e}}}\right)}\lambda+2\mu+\lambda}. \tag{9}$$

This is a process that can be extended for $c > 2$. In fact, expressions for $p_K$ of up to $c = 500$ are available (MacGregor Smith, 2003). Other expression for $c = 3, \ldots, 10$, are included in the software so that we have a complete set of blocking probabilities for $c \in \{1, \ldots, 10\}$.

The remaining details of the second and third stages will not be given here since they are easily found in the literature (Kerbache and MacGregor Smith, 1987; Kerbache and MacGregor Smith, 1988). As a final note, an important point about this process is that we do not physically modify the networks, only represent the expansion process through the software.

## 4 Numerical Results

The software is implemented in fortran and is available from the authors upon request for research purposes. In our implementation, we set $\varepsilon = 10^{-3}$, which was proved to be effective based on the experiments. We first discuss the shape of the objective function. Secondly, we will give

more insights for a number of split structures. We remind that the range of possible experiments is exponential itself and we have determined a select sample to present.

### *4.1 Shape of the Objective Function*

It is interesting to analyze the shape of the objective function for the OROP. The case discussed here is defined as follows. We have a three-node network with a split into two branches, as seen in Figure 3. The general parameters for the base case are $c_1 = 4$, $K_1 = 20$, and $c_i = 2$ and $K_i = 2$, for $i = 2, 3$. The number of servers $c_1$ and the total capacity $K_1$ of node 1 is larger than the others as to prevent it of becoming a bottleneck.

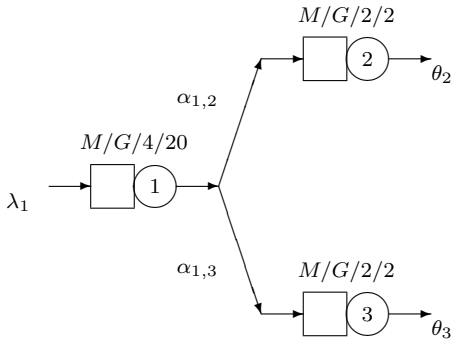
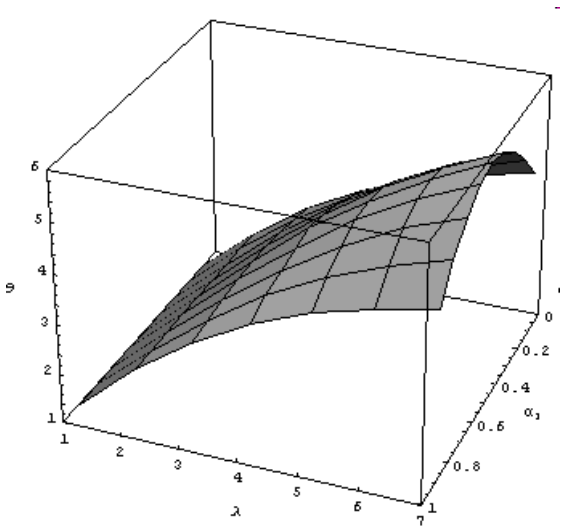



Figure 3: Basic split network B1

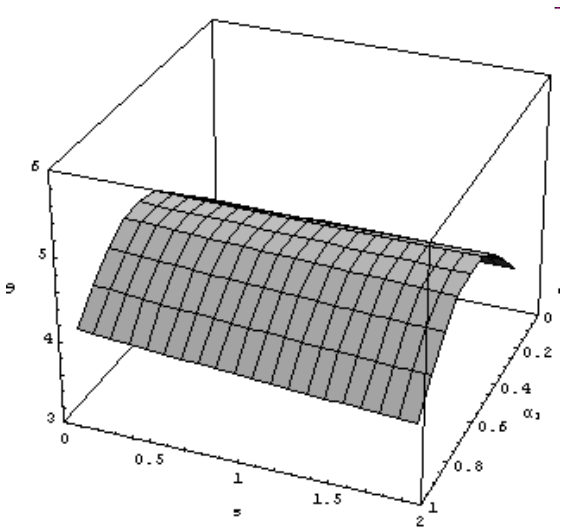
We are particularly interested in the relationship between the overall throughput $\Theta = \theta_1 + \theta_2$, the routing probability $\alpha_{1,2}$, the arrival rate $\lambda_1$, and the squared coefficient of variation of node 2, $s_2^2$. Consequently, we set $\mu_i = 2$, for all nodes, and $s_1^2 = s_3^2 = 1$. The sensitivity of these settings on the throughput is not analyzed. Next, we enumerate all possible combinations for $\lambda_1$, $\alpha_{1,2}$, and $s_2^2$, and then analytically obtain the corresponding throughput $\Theta$, which is shown in Figure 4 (always on the vertical axis), as a function of $\lambda_1$, $\alpha_{1,2}$, and $s_2^2$.

Figure 4-(a) clearly shows that the arrival rate is interacting with the routing probability. For low arrival rates, the network has low utilization. Consequently, different routing probabilities do not result in large changes in throughput $\Theta$. On the other hand, for large arrival rates, $\lambda_1 > 5$, one clearly sees an optimal point in regard to the routing probability. Figure 4-(b) looks into the joint effect of changing the squared coefficient of variation, $s_2^2$, together with the routing probability $\alpha_{1,2}$.

Again the inverted U-shape effect with a maximum around the 50% routing probability is visible. Next to this, it is clear that increasing the squared coefficient of variation from 0 to 2 reduces the overall throughput $\Theta$ but it has a smaller impact on throughput than the routing probability.




(a) effect of $\lambda_1$ versus $\alpha_{1,2}$ on throughput $\Theta$




(b) effect of $s_2^2$ versus $\alpha_{1,2}$ on throughput $\Theta$

Figure 4: The shape of the objective function

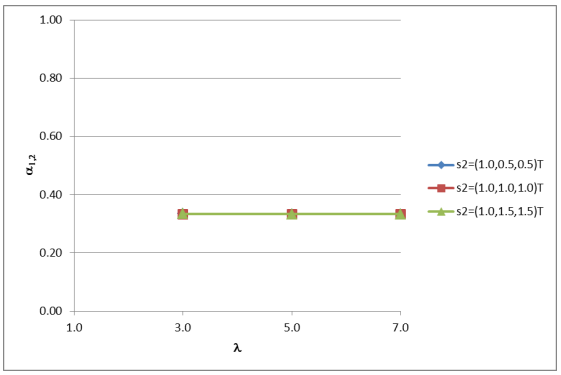
Concluding, based on this simple network structure, it is evident that correctly setting the routing matrix $\boldsymbol{\alpha}$ leads to significant gains in the throughput.

### *4.2 Basic Split Networks*

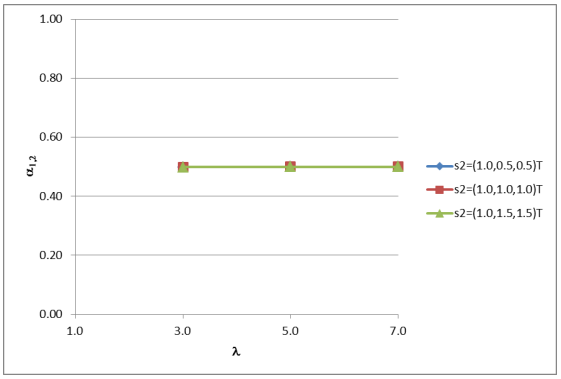
In this section, we analyze further some basic split networks. We are interested in assessing for the OROP the influence of the number of servers $c_i$, the total capacities $K_i$, the service rates $\mu_i$, and the squared coefficient of variation of the service times $s_i^2$, $\forall i \in V$. The nodes after the splits are the ones of interest here.

#### Split with Two Branches

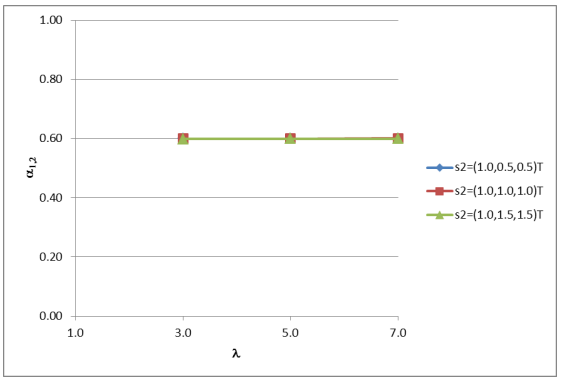
Firstly, we will analyze the two-branch network from Figure 3. The total capacity of node 1 is larger ($K_1 = 20$) than for nodes 2 and 3 (both




(a) $\boldsymbol{\mu} = (2, 1, 2)^{\mathrm{T}}$




(b) $\boldsymbol{\mu} = (2, 2, 2)^{\mathrm{T}}$




(c) $\boldsymbol{\mu} = (2, 3, 2)^{\mathrm{T}}$

Figure 5: Results for two-branch split networks

are equal to 2), as to prevent node 1 to become a bottleneck. The arrival rate $\lambda_1$ is set equal to the values $\{3, 5, 7\}$. Figure 5 gives the results for balanced and unbalanced service rates $\boldsymbol{\mu}$ and different squared coefficients of variation $\mathbf{s}^2$. In these cases the service rate of node 2 is made either relatively lower ($\mu_2 = 1$ versus $\mu_3 = 2$), either equal ($\mu_2 = 2$ versus $\mu_3 = 2$), or higher than the service rate of node 3 ($\mu_2 = 3$ versus $\mu_3 = 2$).

Figure 5-(b) shows that the routing probability is roughly equal to 0.50 when the nodes after the split are identical (that is, same number of servers, capacities, service rates, and squared coefficient of variation). Moreover, these results appear to be insensitive to changes in the squared coefficient of variation of both nodes after the split. As we are now focusing on the small scale networks, this conclusion does not mean that the squared coefficient of variation has little effect *in general* though. Of course, the throughput $\Theta$ is reduced due to the increase in the service variability (results not shown).

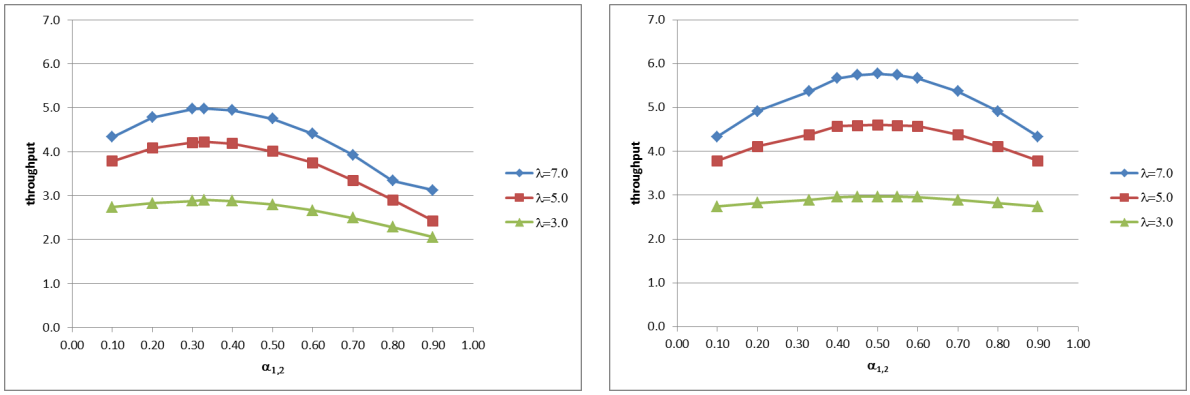
Secondly, changing the service rate of node 2 (and keeping all the other parameters equal), it shows clearly that the fast nodes receive preference over the slow nodes. In Figure 5-(a), for example, when node 2 is slower than node 3, a lower routing probability is set to node 2 (0.3334) than to node 3 (0.6666). This is a confirmation of what we have observed when evaluating the objective function earlier in the previous section.

For the two-branch split networks, we evaluated a number of routing vectors around the optimal routing obtained. Figure 6-(a) shows that the algorithm seems to have reached the 50%-50% optimal allocation for the routing probabilities into nodes 2 and 3. Of course, one might argue that the optimization is rather easy due to the symmetric setting of the parameters. Therefore, we did the same analysis for the same parameter settings but with a network with unbalance in the service rates. As seen in Figure 6-(b), the 33%-67% optimal allocation seems to be reached by the algorithm. Concluding, we have observed that the optimization algorithm tries to balance out the flow taking into account the differences (in service rates and squared coefficient of variation) among the two nodes after the split, which is intuitively logical as this strategy leads to the highest throughput.

### Split with Three Branches

It would be interesting to see to what extent the optimization algorithm balances the flow over three nodes after the split and to what extent this is affected by the characteristics of the different nodes after the split. Then we have include in our analysis the three-branch network seen in Figure 7. The first total capacity is $K_1 = 20$, which is larger than the other nodes (that is, $K_i = 2, i = 2, 3, 4$). As mentioned earlier, this settings prevents the first queue to becoming a bottleneck and to blur the analysis. The other parameters used were $c_1 = 4$, and $c_i = 2$, $\forall i = 2, 3, 4$. The external arrival rates at node 1 were $\lambda_1 \in \{5, 7\}$. The results are presented in Figure 9.

For the complete symmetric case, that is, Figure 9-(a) and -(b), $\mathbf{s}^2 = (1.0, 1.0, 1.0, 1.0)^{\mathrm{T}}$, it is shown that again the routing probabilities are symmetric, *i.e.* $\alpha_{i,j} = 0.3334, \forall\ (i, j) \in A$. For the unbalanced cases in the squared coefficient of variation, that is, $\mathbf{s}^2 = (1.0, 0.0, 1.0, 1.0)^{\mathrm{T}}$, $\mathbf{s}^2 = (1.0, 0.5, 1.0, 1.0)^{\mathrm{T}}$, $\mathbf{s}^2 = (1.0, 1.5, 1.0, 1.0)^{\mathrm{T}}$, and $\mathbf{s}^2 = (1.0, 2.0, 1.0, 1.0)^{\mathrm{T}}$, it can be observed that the routing probability into the two identical nodes ($\alpha_{1,3}$ and $\alpha_{1,4}$) are close to each other.




(a) set $\boldsymbol{\mu} = (2, 2, 2)^{\mathrm{T}}$ and $\mathbf{s}^2 = (1.0, 1.0, 1.0)^{\mathrm{T}}$

(b) set $\boldsymbol{\mu} = (2, 1, 2)^{\mathrm{T}}$ and $\mathbf{s}^2 = (1.0, 1.0, 1.0)^{\mathrm{T}}$

Figure 6: Perturbations around the optimal solution $\alpha^*_{1,2}$ for the two-branch split networks

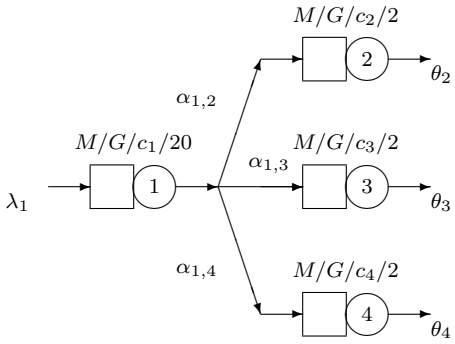



Figure 7: Basic split network B2

For the remaining asymmetrical cases, Figure 9-(c) and -(d), again the same conclusion holds. The faster (high number of servers) or more reliable (low squared coefficient of variation) are the nodes, more favored they are, resulting in high routing probabilities into these nodes.

### *4.3 Complex Networks*

The simple networks discussed so far are interesting as they make it possible to show the behavior and logic of the optimization model in the presence of one split. In this section, we will evaluate a more complex topology in regard to their routing probabilities. The network considered is an extension of the two- and three-branch split networks, as depicted in Figure 8.

Figure 10 gives an overview of a selected set of experiments for structure C1. The base setting is again a balanced case, that is, $\mathbf{K} = (20, 2, 2, 2, 2, 2, 2)^{\mathrm{T}}$, $\boldsymbol{\mu} = (2, 2, 2, 2, 2, 2, 2)^{\mathrm{T}}$, $\mathbf{s}^2 = (1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0)^{\mathrm{T}}$, and $\mathbf{c} = (5, 2, 2, 2, 2, 2, 2)^{\mathrm{T}}$. Additional set of experiments involves unbalancing the number of servers $c_i$ and the service rates $\mu_i$. With these experiments, we evaluate whether and how the methodology takes the characteristics of the complete sub-network after the split into account in determining the optimal routing vector.

We set up the experiments in such a way

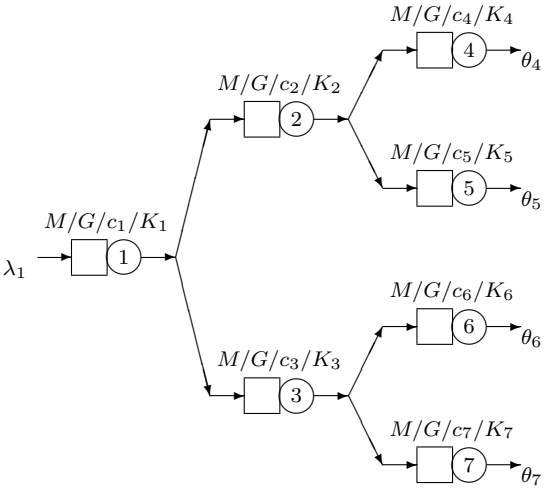
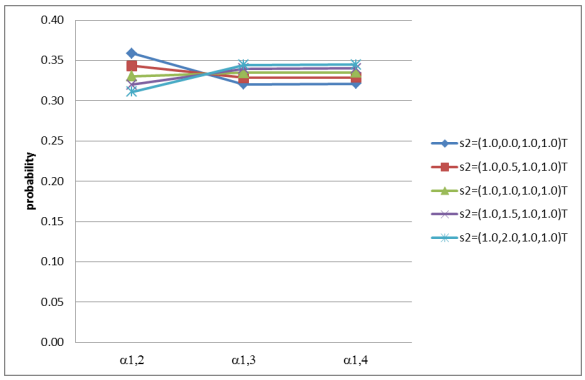



Figure 8: Network structure C1

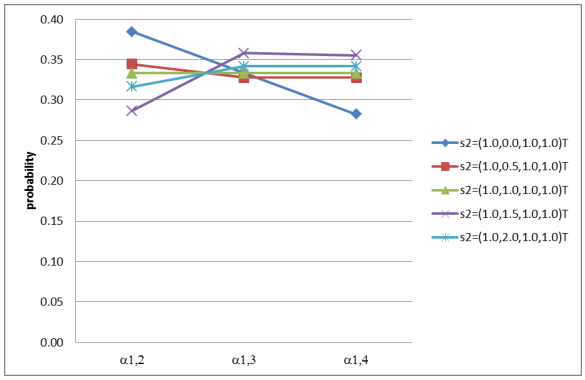
that either there are slow nodes (Figure 10-(a) and -(b), experiment $\mathbf{c} = (5, 2, 2, 5, 1, 5, 1)^{\mathrm{T}}$, and Figure 10-(c) and -(d), experiment $\boldsymbol{\mu} = (2, 2, 2, 1, 5, 1, 5)^{\mathrm{T}}$), or slow subsystems consisting of three connected nodes (Figures 10-(a)–(b), experiment $\mathbf{c} = (5, 2, 2, 1, 1, 5, 5)^{\mathrm{T}}$, and Figure 10-(c) and -(d), experiment $\boldsymbol{\mu} = (2, 2, 2, 1, 1, 5, 5)^{\mathrm{T}}$).

From the results, we observe that in general the slow subsystem of the network tends to receive less flow due to a low routing probability into the relevant part. When after the first split in node 1 there is the choice to go to either the fast or slow subsystem, the faster subsystem is preferred. This is very clear in experiments, when the routing probability always favors the fastest downstream subsystem.

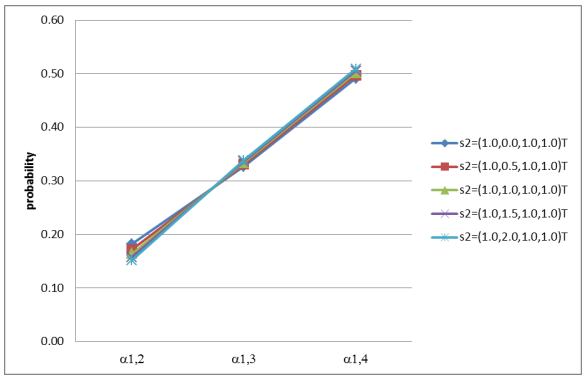
However, if the last nodes are different (experiments $\mathbf{c} = (5, 2, 2, 5, 1, 5, 1)^{\mathrm{T}}$ and $\boldsymbol{\mu} = (2, 2, 2, 1, 5, 1, 5)^{\mathrm{T}}$), the conclusions are different. In all these experiments, the first split is just exactly half. The imbalance in the last nodes (*i.e.*, node 4 and 5, 6, and 7 are different), is completely absorbed in the routing probability at the immediately preceding nodes (*i.e.*, nodes 2 and 3). Interestingly, this effect did not propagate upstream and did not affect the routing at the first split.



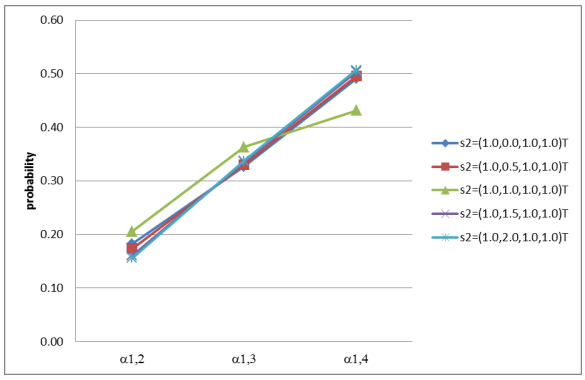

(a) set $\boldsymbol{\mu} = (2, 2, 2, 2)^{\mathrm{T}}$ and $\lambda = 5$




(b) set $\boldsymbol{\mu} = (2, 2, 2, 2)^{\mathrm{T}}$ and $\lambda = 7$

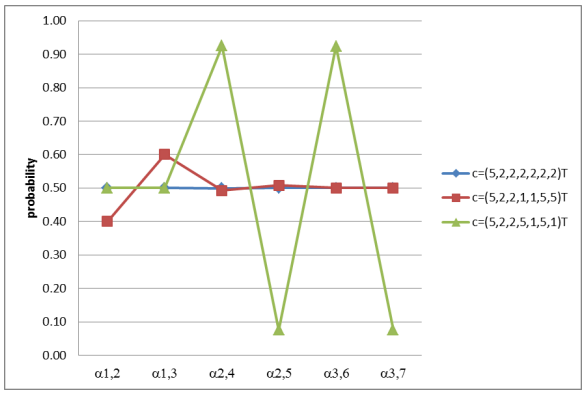



(c) set $\boldsymbol{\mu} = (2, 1, 2, 3)^{\mathrm{T}}$ and $\lambda = 5$

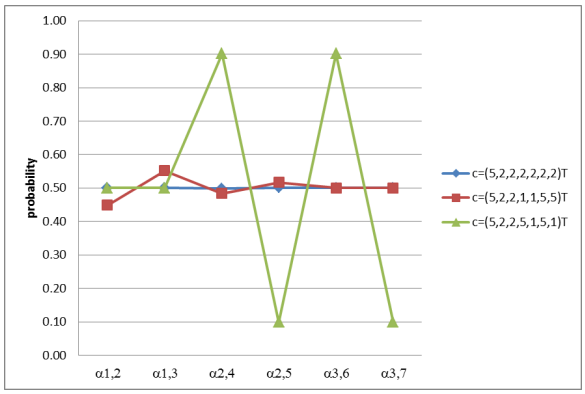



(d) set $\boldsymbol{\mu} = (2, 1, 2, 3)^{\mathrm{T}}$ and $\lambda = 7$

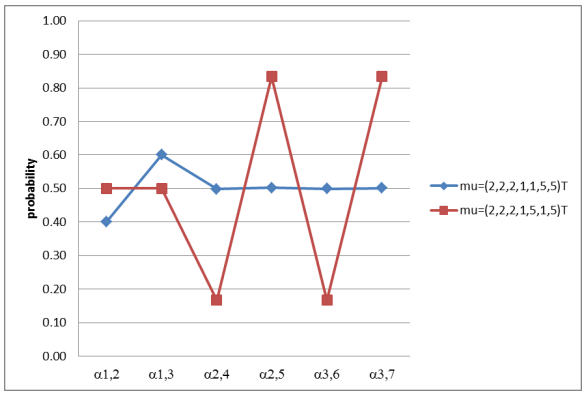
Figure 9: Results for three-branch split networks



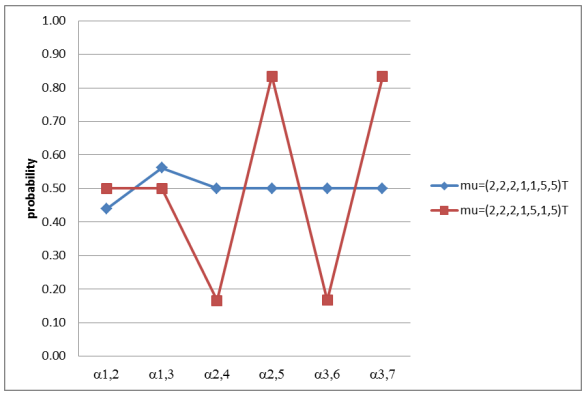

(a) set $\boldsymbol{\mu} = (2, 2, 2, 2, 2, 2, 2)^{\mathrm{T}}$, $s_i^2 = 1.0, \forall i$, and $\lambda = 5.0$




(b) set $\boldsymbol{\mu} = (2, 2, 2, 2, 2, 2, 2)^{\mathrm{T}}$, $s_i^2 = 1.0, \forall i$, and $\lambda = 7.0$




(c) set $\mathbf{c} = (5, 2, 2, 2, 2, 2, 2)^{\mathrm{T}}$, $s_i^2 = 1.0, \forall i$, and $\lambda = 5.0$




(d) set $\mathbf{c} = (5, 2, 2, 2, 2, 2, 2)^{\mathrm{T}}$, $s_i^2 = 1.0, \forall i$, and $\lambda = 7.0$

Figure 10: Results for network structure C1

## 5 Conclusions and Final Remarks

In this paper, we examine the optimal routing problem in finite multiserver queueing networks with generally distributed service times in a given open acyclic topology. We determine suboptimal routing probability vectors to maximize the throughput of the queueing networks, via a combination of the Powell optimization tool and the generalized expansion method. We present numerical results showing the merits of the approach.

We have considered here only the throughput as the main performance measure. It would also be interesting to evaluate the behavior of the routing algorithm to minimize the *cycle time*, or the *work-in-process* (WIP), or any another performance measure of interest. Topics for future research on the area also include the routing in queueing networks with cycles, *e.g.*, to model many important queueing systems that have reverse streams of items due to re-work, or even the extension to $GI/G/c/K$ queueing networks.

## Acknowledgments

The research of prof. Frederico Cruz has been partially funded by the Brazilian agencies, CNPq (*Conselho Nacional de Desenvolvimento Científico e Tecnológico*) of the Ministry for Science and Technology of Brazil, and FAPEMIG (*Fundação de Amparo à Pesquisa do Estado de Minas Gerais*).

## References

Bedell, P. and MacGregor Smith, J. (2012). Topological arrangements of $M/G/C/K$, $M/G/C/C$ queues in transportation and material handling systems, *Computers & Operations Research* **39**(11): 2800–2819.

Cruz, F. R. B., Duarte, A. R. and 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**(1-2): 3–94.

Daskalaki, S. and MacGregor Smith, J. (2004). Combining routing and buffer allocation problems in series-parallel queueing networks, *Annals of Operations Research* **125**(1-4): 47–68.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York.

Gosavi, H. D. and MacGregor Smith, J. (1997). An algorithm for sub-optimal routeing in series-parallel queueing networks, *International Journal of Production Research* **35**(5): 1413–1430.

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

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

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

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

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

MacGregor Smith, J. (2008). $M/G/c/K$ performance models in manufacturing and service systems, *Asia-Pacific Journal of Operational Research* **25**(4): 531–561.

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

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

MacGregor Smith, J. and Daskalaki, S. (1988). Buffer space allocation in automated assembly lines, *Operations Research* **36**(2): 343–358.

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.