Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2012, Article ID 692593, 19 pages
doi:10.1155/2012/692593

*Research Article*

# Throughput Maximization of Queueing Networks with Simultaneous Minimization of Service Rates and Buffers

**F. R. B. Cruz,[1] G. Kendall,[2] L. While,[3] A. R. Duarte,[4] and N. L. C. Brito[5]**

[1] *Departamento de Estatística, Universidade Federal de Minas Gerais, 31270-901 Belo Horizonte, MG, Brazil*
[2] *School of Computer Science, University of Nottingham, Jubilee Campus, Wollaton Road, Nottingham NG8 1BB, UK*
[3] *School of Computer Science & Software Engineering, The University of Western Australia, 35 Stirling Highway, Crawley, WA 6009, Australia*
[4] *Departamento de Matemática, Universidade Federal de Ouro Preto, 35400-000 Ouro Preto, MG, Brazil*
[5] *Departamento de Ciências Exatas, Universidade Estadual de Montes Claros, 39401-089 Montes Claros, MG, Brazil*

Correspondence should be addressed to F. R. B. Cruz, fcruz@est.ufmg.br

Received 27 August 2011; Revised 11 November 2011; Accepted 12 November 2011

Academic Editor: Hung Nguyen-Xuan



The throughput of an acyclic, general-service time queueing network was optimized, and the total number of buffers and the overall service rate was reduced. To satisfy these conflicting objectives, a multiobjective genetic algorithm was developed and employed. Thus, our method produced a set of efficient solutions for more than one objective in the objective function. A comprehensive set of computational experiments was conducted to determine the efficacy and efficiency of the proposed approach. Interesting insights obtained from the analysis of a complex network may assist practitioners in planning general-service queueing networks.

## 1. Introduction

In this study, the maximization of throughput ($\Theta$) (the number of jobs, parts, clients, etc., served per unit of time) in an acyclic, general-service time queueing network (for an example, see Figure 1) was evaluated. To obtain the maximum $\Theta$, the minimum number of buffers ($\mathbf{K} = \{K_1, K_2, \ldots, K_n\}$) and service rates ($\boldsymbol{\mu} = \{\mu_1, \mu_2, \ldots, \mu_n\}$) that must be allocated to a queueing network in a given topology and external arrival rate ($\boldsymbol{\Lambda} = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_n\}$) was determined. Potential users of general-service time, finite-queueing network-based optimization models include computer scientists and industrial engineers. Indeed, these models may help

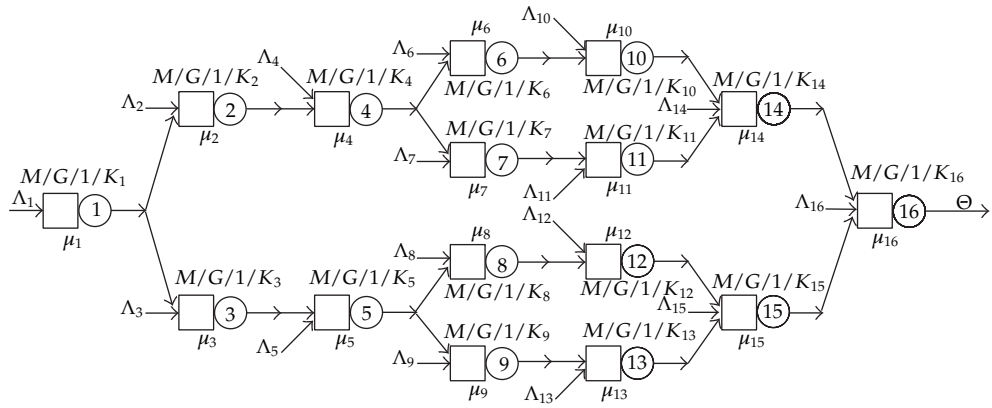



**Figure 1:** A complex network (adapted from Smith and Cruz [20]).

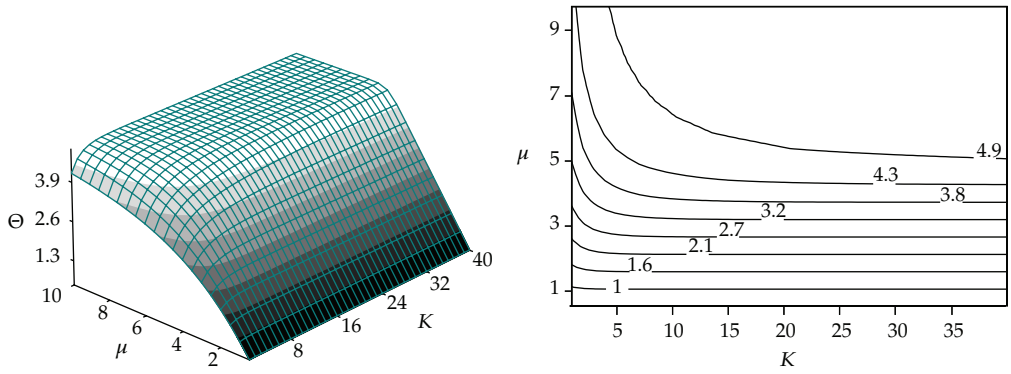
to understand and improve various real-life systems, including manufacturing [1–5], production [6–8], and health [9–11] systems, urban or pedestrian traffic [12, 13], computer and communication systems [14–17], web-based applications with tiered configurations [18], and quality-of-service (QoS) requirements measured in terms of response times, throughput, server availability, and security [19].

This study focused on single-server queueing networks with exponentially distributed interarrival times and generally distributed service times, configured in an arbitrary acyclic, series-parallel topology. An example of the type of network under consideration is shown in Figure 1. In particular, buffer allocation, server allocation, and throughput tradeoff were optimized in networks of $M/G/1/K$ queues. Thus, in Kendall [21] notation, we focused on Markovian arrivals, generally distributed service times, a single server, and the total capacity of $K$ items, *including* the item of service.

Indeed, there is a critical tradeoff between the overall number of buffers and service rates and the resulting throughput. Buffers and service capacities can be very expensive; thus, the total number of buffers and the overall service capacity should be reduced as much as possible. On the other hand, the highest possible network throughput is also desired. Unfortunately, throughput is directly affected by the number of buffers allocated, where an increase in buffers generally leads to a higher throughput. Likewise, the service capacity also directly affects the throughput.

Figure 2 shows the throughput, $\Theta$, for a single $M/G/1/K$ queue with $s^2 = 1.5$ (squared coefficient of variation of the service time) and $\Lambda = 5$ users per time unit (external arrival rate), as a function of several values for buffer size, $K$, and service rate, $\mu$ (see (2.1) and (2.2)), as well as the respective contour plot. Similar throughput behavior is also observed in a network of queues. The surface of the plot shown in Figure 2 is smooth, and convexity is apparent, which is similar to the results of simple queueing networks [22, 23]. However, the top of the surface plot near the maximum throughput is flat, which creates difficulties for traditional optimization methods. For instance, Smith and Cruz [20] used the Powell method and multiple starts to avoid premature convergence to a local optimum and to derive a successful optimization algorithm.

From a modeling point of view, throughput maximization can be defined by a mixed-integer mathematical programming formula, where the total buffer and server costs are minimized, and the throughput, subject to integer buffer allocation and nonnegative service




(a) Throughput ($\Theta$) *versus* service rate ($\mu$) and buffer size ($K$)

(b) Contour plot

**Figure 2:** Results of a single $M/G/1/K$ queue for $\Lambda = 5.0$.

rates, is maximized. By defining a queueing network as a digraph of $G(N, A)$, where $N$ is a finite set of nodes, and $A$ is a finite set of arcs, the mixed-integer mathematical programming formula was obtained [24]

$$\text{minimize } F(\mathbf{K}, \boldsymbol{\mu}) \tag{1.1}$$

subject to

$$\begin{aligned} K_i &\in \{1, 2, \ldots\}, \quad \forall i \in N, \\ \mu_i &\geq 0, \quad \forall i \in N, \end{aligned} \tag{1.2}$$

where the decision variables $K_i$ and $\mu_i$ indicate the total capacity of the service and the service rate for the $i$th $M/G/1/K$ queue, respectively. The objective functions, $F(\mathbf{K}, \boldsymbol{\mu}) \equiv (f_1(\mathbf{K}), f_2(\boldsymbol{\mu}), -f_3(\mathbf{K}, \boldsymbol{\mu})$, are the total buffer allocation, $f_1(\mathbf{K}) = \sum_{\forall i \in N} K_i$, the overall service allocation, $f_2(\boldsymbol{\mu}) = \sum_{\forall i \in N} \mu_i$, and the overall throughput, $f_3(\mathbf{K}, \boldsymbol{\mu}) = \Theta(\mathbf{K}, \boldsymbol{\mu})$.

Throughput is often modeled as a constraint that must be greater than a target minimum, rather than as an objective that must be maximized [20, 25]. However, to successfully solve the problem, throughput constraints must be relaxed. Thus, parameters such as the threshold throughput ($\Theta_\tau$) must be determined beforehand. However, establishing an appropriate threshold is not a trivial task. Moreover, it is possible that a small decrease in throughput can result in a significant reduction in buffer allocation (and costs). The tradeoff between throughput and the number of buffers is not apparent in a single-objective formula. Indeed, the weights ($\omega$) of a single-objective function have a significant effect on both the objectives and parameters, including errors ($\varepsilon$) on performance measure estimates and threshold throughput ($\Theta_\tau$). Thus, weight determination is difficult, and the results of single objective optimization techniques can be arbitrary.

In this study, an optimization approach was developed that determines the entire set of Pareto-optimal solutions. Thus, our method produces a set of efficient solutions for more than one objective in the objective function [26]. With the proposed approach, the decision maker is able to evaluate the effect of solution replacement. Moreover, the multiobjective approach

also allows the user to increase one objective (e.g., throughput) while simultaneously reducing another objective (e.g., buffer and service rate allocation). A multiobjective evolutionary algorithm (MOEA) was used in combination with a generalized expansion method (GEM), which is a well-known method for obtaining accurate approximations of queueing network performance [27–29]. MOEAs are particularly suitable for multiobjective problems and have been shown to perform well in similar multiobjective problems of networks (e.g., see Carrano et al. [30] and references therein).

In this paper, a MOEA, specifically developed to multiobjective optimization, is presented (see Section 2). Additionally, the GEM, a performance evaluation tool used to approximate throughput, is also presented. In Section 3, the results of a comprehensive set of computational experiments are presented to show the efficiency of the approach. Finally, the article is concluded in Section 4, where final remarks and suggestions for future research are discussed.

## 2. Proposed Algorithms

The exposition of proposed algorithms was conducted in two parts. First, the performance evaluation algorithm was presented, which allowed the overall performance of the system to be estimated in terms of overall throughput, $\Theta(\mathbf{K}, \boldsymbol{\mu})$, for a given configuration of the buffer and service allocation. Then the proposed optimization algorithm was developed, which was applied to obtain the optimal buffer and service allocation.

### 2.1. Performance Evaluation Algorithm

#### 2.1.1. Single Queues

To maximize the throughput, $\Theta(\mathbf{K}, \boldsymbol{\mu})$ must be estimated. In a single $M/G/1/K$ queue, the estimation of $\Theta(\mathbf{K}, \boldsymbol{\mu})$ can be achieved with a computationally efficient and accurate closed-form approximate expression of the blocking probability, $p_K$. The method proposed by Smith [31], which is based on a two-moment approximation from Kimura [32], was employed

$$p_K = \frac{\rho^{\left((2+\sqrt{\rho}s^2-\sqrt{\rho}+2(K-1))/(2+\sqrt{\rho}s^2-\sqrt{\rho})\right)}(\rho-1)}{\rho^{\left(2((2+\sqrt{\rho}s^2-\sqrt{\rho}+(K-1))/(2+\sqrt{\rho}s^2-\sqrt{\rho}))\right)}-1}, \tag{2.1}$$

where $\rho < 1$ is the system utilization, which is the ratio between the total arrival rate and the service rate, $\rho = \lambda/\mu$. $s^2$ is the squared coefficient of variation of the service time, $T_s$; thus, $s^2 = \mathrm{Var}(T_s)/\mathrm{E}(T_s)^2$. The results indicated that the approximation of $p_K$ was accurate for a wide range of values [20, 25, 33].

In order to obtain the throughput in a finite $M/G/1/K$ single queue, we need to adjust the arrival rate. In fact a fraction $p_K$ of the arrivals cannot join the system because they have come when there is no waiting space left. Thus the actual rate of arrivals to join the system must be adjusted accordingly. Since Poisson arrivals see time averages (the PASTA property), it follows that the effective arrival rate seen by the servers is $\lambda_{\mathrm{eff}} = \lambda(1 - p_K)$ [34]. Thus, the throughput may be given by

$$\theta = \lambda_{\mathrm{eff}} = \lambda(1 - p_K). \tag{2.2}$$

### 2.1.2. Network of Queues

For a network of queues, the estimation of throughput is considerably more complicated. The generalized expansion method (GEM) is an algorithm that has been successfully used to estimate the performance of arbitrarily configured, finite queueing, and acyclic networks [29]. The GEM is a combination of node-by-node decomposition and repeated trials, where each queue is analyzed separately, and corrections are made to account for interrelated effects between network queues. The GEM uses type I blocking, where the upstream node becomes blocked if the service for an individual customer is complete and the queue at the downstream node is full. This is often referred to as "blocking after service," which is prevalent in most production, manufacturing, and transportation systems.

With the help of Figure 3, we now describe the GEM. Firstly, we remark that the exponential distribution is a good approximation for the interdeparture times of entities leaving an $M/G/1/K$ node. In fact, it is possible to show the quasireversibility of a broader class of finite queues, which are the state-dependent $M/G/C/C$ queues [35]. When those entities that are lost are included, the output stream is Poisson. This assumption is supported by several empirical results [7, 8, 13, 20, 25, 36]. The following three stages are involved in the GEM: *network reconfiguration*, *parameter estimation*, and *feedback elimination*.

#### *Network Reconfiguration*

This stage involves reconfiguring the network. An auxiliary vertex $h_j$ is created, which is modeled as an $M/G/\infty$ queue with service rate $\mu_h$ and precedes each finite queue $j$. When an entity leaves $i$, 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 busy. After this delay, the entity may be blocked again with probability $p'_{K_j}$, for a second delay period. Vertex $h_j$ accumulates the time an entity has to wait before entering vertex $j$ and the effective arrival rate to vertex $j$.

#### *Parameter Estimation*

In this stage, the parameters $p_K$, $p'_K$, and $\mu_h$ are estimated (for clarity, we will omit the subscript for node $j$).

(1) $p_K$ is obtained by means of a two-moment approximation recently developed by Smith [31]

$$p_K = \text{ equation (2.1)}. \tag{2.3}$$

(2) $p'_K$ is obtained with the following approximation from diffusion techniques given by Labetoulle and Pujolle [37]:

$$p'_K = \left( \frac{\mu_j + \mu_h}{\mu_h} - \frac{\lambda\left(\left(r_2^K - r_1^K\right) - \left(r_2^{K-1} - r_1^{K-1}\right)\right)}{\mu_h\left(\left(r_2^{K+1} - r_1^{K+1}\right) - \left(r_2^K - r_1^K\right)\right)} \right)^{-1}, \tag{2.4}$$

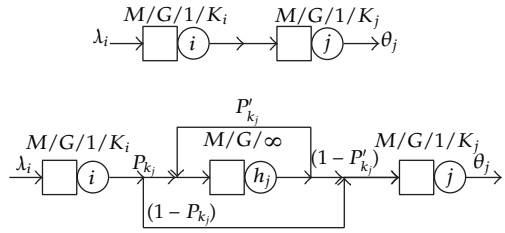



**Figure 3:** Generalized expansion method.

where $r_1$ and $r_2$ are the roots to the polynomial

$$\lambda - (\lambda + \mu_h + \mu_j)x + \mu_h x^2 = 0, \tag{2.5}$$

with $\lambda = \lambda_j - \lambda_h(1 - p'_K)$, $\lambda_h$ is the actual arrival rate to the artificial holding node, and $\lambda_j$ is the actual arrival rate to the finite node $j$, given by

$$\lambda_j = \tilde{\lambda}_i(1 - p_K) = \tilde{\lambda}_i - \lambda_h. \tag{2.6}$$

(3) $\mu_h$ is calculated as follows using renewal theory:

$$\mu_h = \frac{2\mu_j}{1 + \sigma_j^2 \mu_j^2}, \tag{2.7}$$

where $\sigma_j^2$ is the service time variance.

*Feedback Elimination*

The repeated visits to the holding nodes (due to the feedback) create strong dependence in the arrival process. Therefore, the repeated immediate feedback is eliminated. This is accomplished by giving the customer enough service time during the first passage through the holding node. The adapted service rate for the holding node $\mu'_h$ then becomes

$$\mu'_h = (1 - p'_K)\mu_h. \tag{2.8}$$

*Summary 1.* The goal of GEM is to provide an approximation scheme to update the service rates of upstream nodes that take into account all blocking after service caused by downstream nodes

$$\tilde{\mu}_i^{-1} = \mu_i^{-1} + p_K(\mu'_h)^{-1}. \tag{2.9}$$

For each finite node $j$ in the network succeeding node $i$, we have simultaneous nonlinear equations in variables $p_K$, $p'_K$, and $\mu_h$, along with auxiliary variables such as $\lambda$ and

$\tilde{\lambda}_i$. Solving these equations simultaneously, we can compute all the performance measures of the network

$$\lambda = \lambda_j - \lambda_h\left(1 - p'_K\right), \tag{2.10}$$

$$\lambda_j = \tilde{\lambda}_i\left(1 - p_K\right), \tag{2.11}$$

$$\lambda_j = \tilde{\lambda}_i - \lambda_h, \tag{2.12}$$

$$p'_K = \left(\frac{\mu_j + \mu_h}{\mu_h} - \frac{\lambda\left(\left(r_2^K - r_1^K\right) - \left(r_2^{K-1} - r_1^{K-1}\right)\right)}{\mu_h\left(\left(r_2^{K+1} - r_1^{K+1}\right) - \left(r_2^K - r_1^K\right)\right)}\right)^{-1}, \tag{2.13}$$

$$z = \left(\lambda + 2\mu_h\right)^2 - 4\lambda\mu_h, \tag{2.14}$$

$$r_1 = \frac{\left[\left(\lambda + 2\mu_h\right) - z^{1/2}\right]}{2\mu_h}, \tag{2.15}$$

$$r_2 = \frac{\left[\left(\lambda + 2\mu_h\right) + z^{1/2}\right]}{2\mu_h}, \tag{2.16}$$

$$p_K = \text{ equation (2.1)}. \tag{2.17}$$

Equation (2.10) through (2.13) is related to the arrivals and feedback in the holding node. Equation (2.14) through (2.16) is used to solve (2.13) with $z$ used as a dummy parameter for simplicity. Lastly, (2.1) gives the blocking probability for the $M/G/1/K$ queue. Thus, we essentially have five equations to solve (2.10)–(2.13) and (2.1).

## 2.2. Optimization Algorithm

For the network under consideration, MOEAs appeared to be a suitable choice for the multiobjective maximization of throughput. MOEAs are optimization algorithms that perform an approximate global search based on information obtained from the evaluation of several points in the search space [38, 39]. The population of points that converge to an optimal value are obtained through the application of genetic operators such as *mutation*, *crossover*, *selection*, and *elitism*. Each one of these operators characterizes an instance of a MOEA and can be implemented in several different ways. Additionally, MOEA convergence is guaranteed by assigning a value of fitness to each population member and preserving diversity. In fact, recent successful applications of GAs for single-objective applications were reported by Lin [40] and Calvete et al. [41], whereas Carrano et al. [30] employed GAs for multiple-objective applications. Additionally, the efficiency of GAs is well established for multiobjective problems [42, 43]. Many references are provided by the aforementioned authors.

The instance of MOEA used in this study was based upon the elitist nondominated sorting genetic (NSGA-II) algorithm of Deb et al. [44], which is shown in Algorithm 1 . In the application of GAs for multiobjective optimization, the *selection* operator and *elitism* operator must be specifically structured to correctly identify optimal conditions as shown shortly. Elitism is based on the concept of dominance. Point $\mathbf{x}_i = (x_{i_1}, x_{i_2}, \ldots, x_{i_n})$ dominates point

```
algorithm
  read graph, arrival, service rates, G(N, A), Λ_i ∀ i ∈ N
  P_1 ← GenerateInitialPopulation (popSize)
  for i = 1 until numGen do
    /* generate offspring by crossover and mutation */
    Q_i ← MakeNewPop (P_i)
    /* combine parent and offspring */
    R_i ← P_i ∪ Q_i
    /* find nondominated fronts ℱ = (ℱ_1, ℱ_2, ...)*/
    ℱ ← FastNonDominatedSort (R_i)
    /* find new population by */
    /* the crowding-distance-assignment */
    P_{i+1} ← GenerateNewPopulation (R_i)
  end for
  P_{numGen +1} ← ExtractParetoSet (P_numGen)
  write P_{numGen+1}
end algorithm
```

**Algorithm 1:** Elitist multiobjective genetic algorithm (NSGA-II).

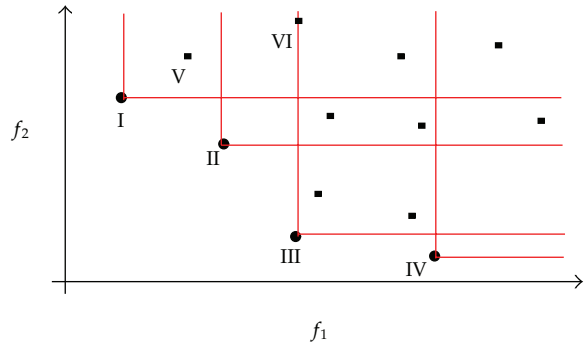



**Figure 4:** Dominated (■) and nondominated (•) points.

$\mathbf{x}_j = (x_{j_1}, x_{j_2}, \ldots, x_{j_n})$ if $\mathbf{x}_i$ is superior to $\mathbf{x}_j$ in one objective ($f_k(\mathbf{x}_i) < f_k(\mathbf{x}_j)$, for minimization) and is not inferior in any other objective ($f_\ell(\mathbf{x}_i) \not> f_\ell(\mathbf{x}_j)$, for minimization).

For instance, Figure 4 displays the points for a two-dimension minimization problem. In the figure, point V is dominated by point I, but not by points II, III, and IV. Point VI is dominated by points I, II, and III, but not by point IV. The best front includes points I through IV and is an approximation for the Pareto set, which is the set of points that are superior to other points. To perform elitism, an algorithm commonly referred to as the fast nondominated sorting algorithm was employed (details may be found in Deb et al. [44]). This algorithm separates the individuals in the population into several layers or fronts $\mathcal{F}_i$, such that the solutions in $\mathcal{F}_1$ are nondominated, and every solution in a given front $\mathcal{F}_i$, $i > 1$, is dominated by at least one solution in $\mathcal{F}_{i-1}$, and not by any solution in $\mathcal{F}_j$, $j \geq i$. This can be achieved in $\mathcal{O}(n \log n)$ time [44].

Selection is performed by sequentially selecting points from each nondominated front $(\mathcal{F}_1, \mathcal{F}_2, \ldots)$ until the number of required individuals for the next iteration is obtained. Criteria must be applied if, after the addition of a group of individuals from $\mathcal{F}_i$, the maximum number

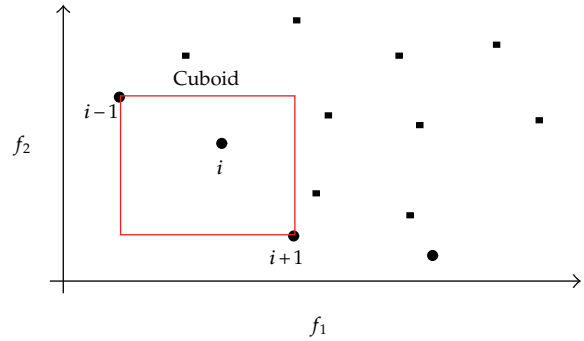



**Figure 5:** Illustration of the crowding distance.

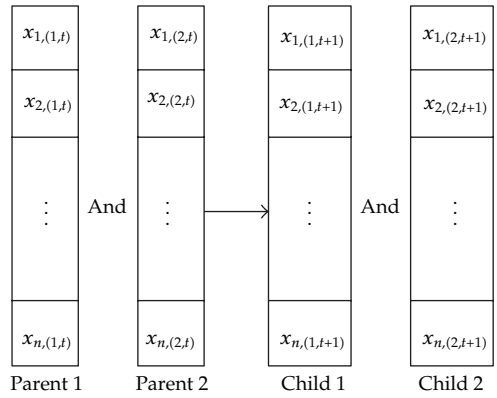



**Figure 6:** Chromosome representation and simulated binary crossover (SBX).

of individuals is exceeded. The algorithm computes a measure of diversity (the crowding distance, as defined by Deb et al. [44]) to ensure the highest possible diversity. Thus, only the points with the largest crowding distance are kept for future iterations, as shown in Figure 5.

*Crossover* and *mutation* are somewhat independent of the multiobjective nature of the problem but are highly dependent on the application. For the problem at hand, a crossover mechanism known as "uniform" was selected [45]. Uniform crossover is popular in multivariable encodings due to its efficiency in identifying, inheriting, and protecting common genes, as well as recombining noncommon genes [46, 47]. In this mechanism, crossovers were performed for each variable with a probability (`rateCro`) that is in accordance with the crossover operator. The crossover operator used in the algorithm was the simulated binary crossover operator (SBX) [48, 49], as illustrated in Figure 6. SBX is quite convenient for real-coded GAs because it is able to simulate binary crossover operators but avoids reencoding the variables. The children $(x_{i,(\cdot,t+1)})$ were calculated from the parents $(x_{i,(\cdot,t)})$ according to the following equation:

$$\begin{aligned} x_{i,(1,t+1)} &= 0.5\left[(1+\beta)x_{i,(1,t)} + (1-\beta)x_{i,(2,t)}\right], \\ x_{i,(2,t+1)} &= 0.5\left[(1-\beta)x_{i,(1,t)} + (1+\beta)x_{i,(2,t)}\right], \end{aligned} \tag{2.18}$$

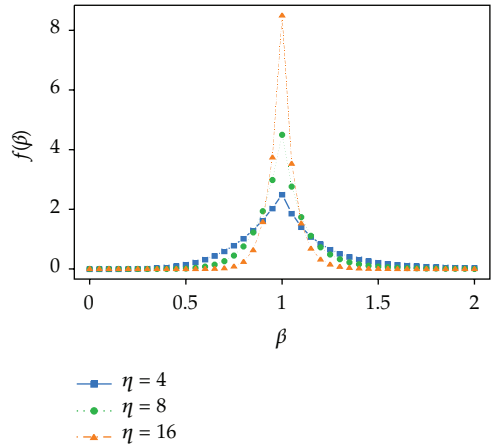



**Figure 7:** Probability density function of $\beta$.

where $\beta$ is a random variable obtained from the following probability distribution function:

$$f(\beta) = \begin{cases} 0.5(\eta+1)\beta^{\eta}, & \text{if } \beta \le 1, \\ 0.5(\eta+1)\dfrac{1}{\beta^{\eta+2}}, & \text{otherwise.} \end{cases} \tag{2.19}$$

The function was designed to create a child solution that possesses a similar search power to a single-point crossover of binary-coded GAs [48]. By adjusting $\eta$, several different weights ($\beta$) can be generated to produce children that are similar to their parents (i.e., large $\eta$) or not (small $\eta$). Several distributions are shown in Figure 7.

For each individual gene (the decision variables $K_i$ and $\mu_i$), the mutation scheme occurs with a specific probability (`rateMut`). As suggested by Deb and Agrawal [48], Gaussian perturbations were added to the decision variables, $K_i + \varepsilon_i$ and $\mu_i + \varepsilon_{N+i}$, for all $i \in N$, with $\varepsilon_i \sim \mathcal{N}(0,1)$, $i \in \{1,2,\ldots,2N\}$.

After crossover and mutation, constraints (1.2) may no longer apply. To guarantee feasibility, the values of integer variables were rounded accordingly and were readjusted by applying reflection operators

$$\begin{aligned} K_{\mathrm{rfl}_i} &= 1 + |K_i - 1|, \\ \mu_{\mathrm{rfl}_i} &= \mu_{\mathrm{lowlim}_i} + \left|\mu_i - \mu_{\mathrm{lowlim}_i}\right|, \end{aligned} \tag{2.20}$$

where 1 is the lower limit of buffer allocation, $\mu_{\mathrm{lowlim}_i}$ is the lower limit of service allocation (to ensure that $\rho < 1$ is applicable), $K_i$ and $\mu_i$ are the resulting values after crossover and mutation, and $K_{\mathrm{rfl}_i}$ and $\mu_{\mathrm{rfl}_i}$ are the results after reflection. The proposed scheme generates feasible solutions without avoiding or favoring any particular solution.

Recently, the stopping criterion of multiobjective optimization evolutionary algorithms has been analyzed in detail (see, e.g., Rudenko and Schoenauer [50] and Martí et al. [51]). Evidently, the maximum number of generations (`numGen`) plays an important role in the quality of the solutions. However, increasing the number of generation may

not be ideal because computational time is wasted when many iterations do not lead to a significant improvement. Thus, Rudenko and Schoenauer [50] suggested that a superior stopping criterion is obtained when a fixed number of iterations are performed without improvement. To demonstrate the complexity of the issue, Rudenko and Schoenauer [50] conducted a comprehensive set of computational experiments. Their results revealed that an obvious stopping criterion, such as the entire population possessing a rank of 1, did not indicate that evolution should be terminated. The authors proposed a local stopping criterion that computes a measure of the stability of nondominated solutions after each iteration. Another global stopping criterion was recently proposed by Martí et al. [51]. This sophisticated method views population evolution as a dynamic system, where the state of the system can be estimated by a Kalman filter. For the sake of simplicity, the criterion of Rudenko and Schoenauer [50] was employed in this study. This criterion is based on the stabilization of the maximal crowding distance, $d_l$, measured over $L$ generations, and is calculated by the following standard deviation:

$$\sigma_L = \sqrt{\frac{1}{L}\sum_{l=1}^{L}\left(d_l - \overline{d}_L\right)^2}. \tag{2.21}$$

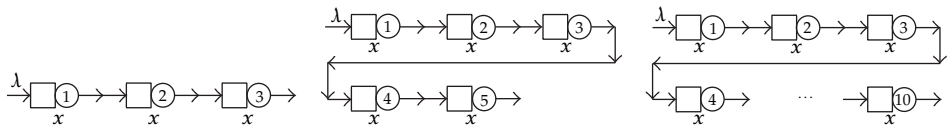
As shown in (2.21), $\overline{d}_L$ is the average of $d_l$ over $L$ generations. Moreover, (2.21) indicates that the MOEA stops when $\sigma_L < \delta_{\text{lim}}$. Rudenko and Schoenauer [50] suggested that $\sigma_L$ does not depend on the actual values of the objective function because crowding distances are normalized. Furthermore, they also suggested that $L$ and $\delta_{\text{lim}}$ should be set to 40 and 0.02, respectively, which leads to a stopping criteria that is $\sigma_{40} \leq 0.02$. According to empirical evidence, these values are compatible with the network under consideration.

## 3. Computational Results and Discussion

To use previous implementations of GEM based on the International Mathematics and Statistics Library (IMSL), the optimization algorithm was implemented in Fortran [31, 33]. The code is available from the corresponding author upon request and must be used for educational and research purposes only. The computational experiments were conducted to discover the suboptimal set of parameters that guarantee rapid convergence. Moreover, the analysis of a large and complex network of finite queues was also achieved with the proposed algorithm.

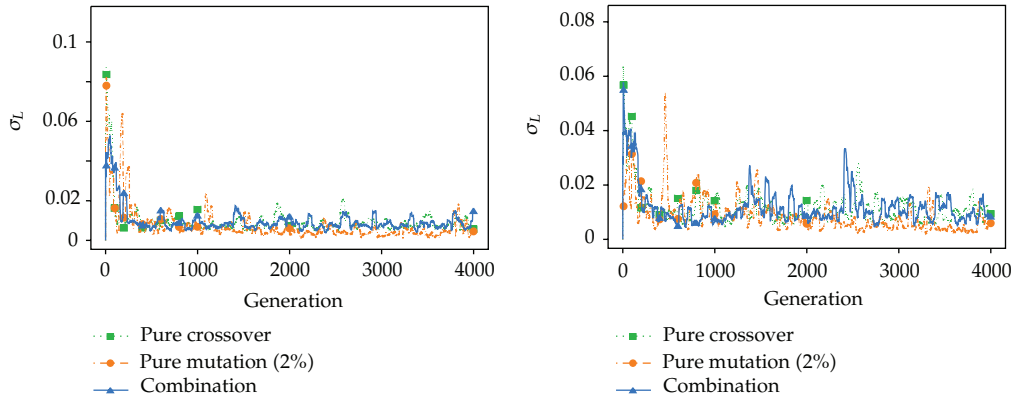
### *3.1. Setting the Parameters*

Unfortunately, to ensure rapid convergence with a minimal amount of computational effort, the suboptimal set of parameters must be determined by trial and error, as indicated by previous studies on GAs. Thus, networks containing 3, 5, and 10 queues were used to set the parameters of MOEA (see Figure 8). For the sake of conciseness, only the results obtained from 3 and 10 nodes are presented (Figures 9–12). Different topologies of acyclic similarly sized networks were also tested, and the results (not presented) were similar to those obtained from series topologies.



(a) A three-node series topology (b) A five-node series topology (c) A ten-node series topology

**Figure 8:** Tested topologies.




(a) 3-node series network (b) 10-node series network

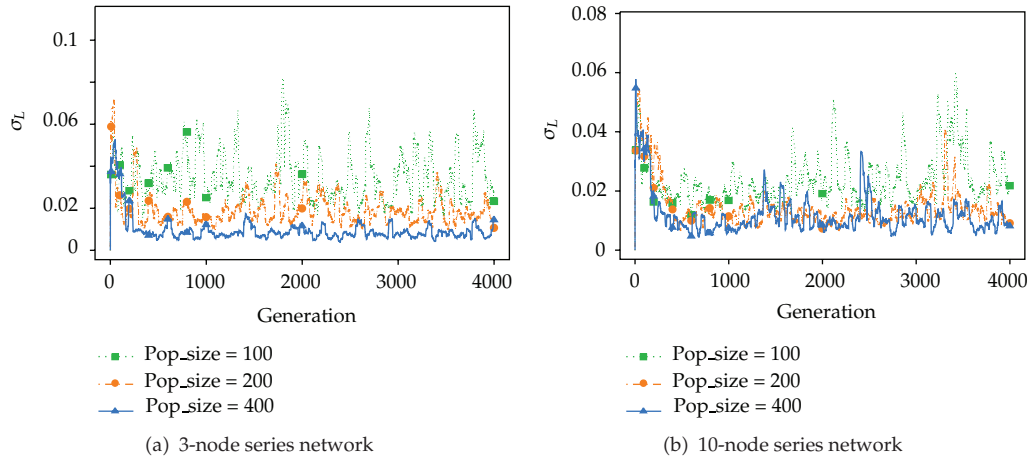
**Figure 9:** Effect of crossover and mutation.

In this study, each factor was analyzed independently; specifically, each factor was varied one at a time while the other parameters were held constant. Montgomery [52] reminds us that the major disadvantage of an independent analysis is that it fails to account for interactions between variables. However, recent experiments reported by Cruz et al. [53] indicated that potential interactions were insignificant; thus, interactions between factors were neglected in this study.

Figure 9 presents the convergence speed (in terms of $\sigma_L$) as a function of the number of generations. It is possible to conclude that *pure* mutation could be used to determine the optimal solution (sometimes pure mutation solves the problem, see Mathieu et al. [54]). However, the SBX operator was also utilized because it removed irregularities from the convergence profile. The combination of pure mutation and SBX provided satisfactory results, regardless of the number of queues in the network.

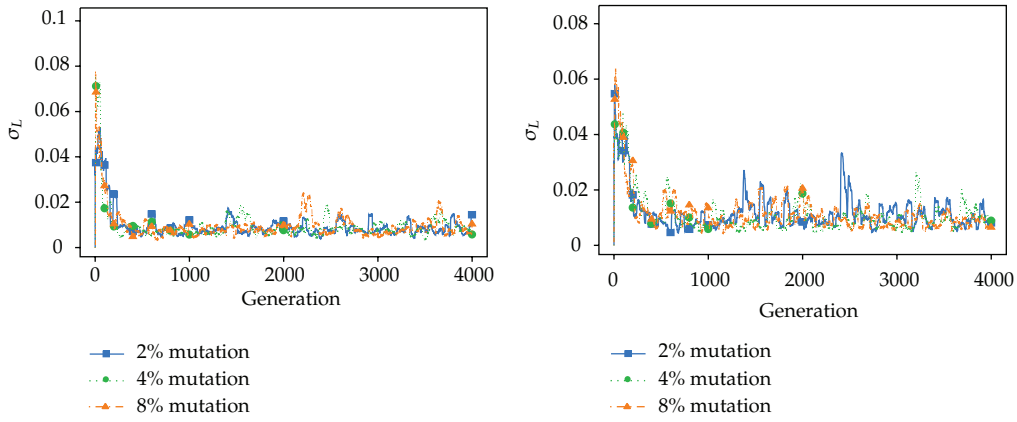
The results in Figure 10 revealed that the population size (`popSize`) had a significant effect on algorithm convergence. However, the population size cannot be arbitrarily increased because the required computational effort would become prohibitive. Moreover, the performance of the algorithm was not affected by an increase in the number of network nodes.

Figure 11 displays the convergence rate as a function of the parameter `rateMut`. The results revealed that an increase in the mutation rate accelerated convergence; however, once a specific rate was attained, a further increase did not lead to improved convergence. Under the experimental conditions, mutation rates between 1 and 2% provided superior results, regardless of the number of nodes.



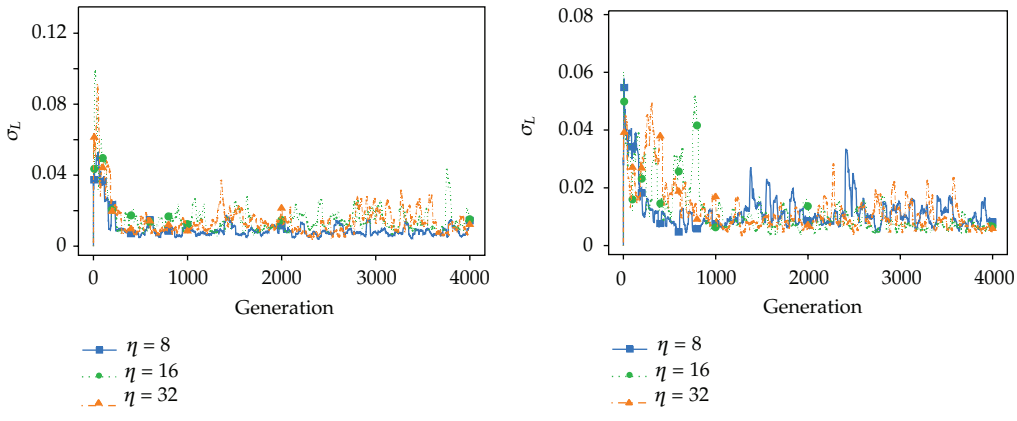

(a) 3-node series network (b) 10-node series network

**Figure 10:** Effect of population size.



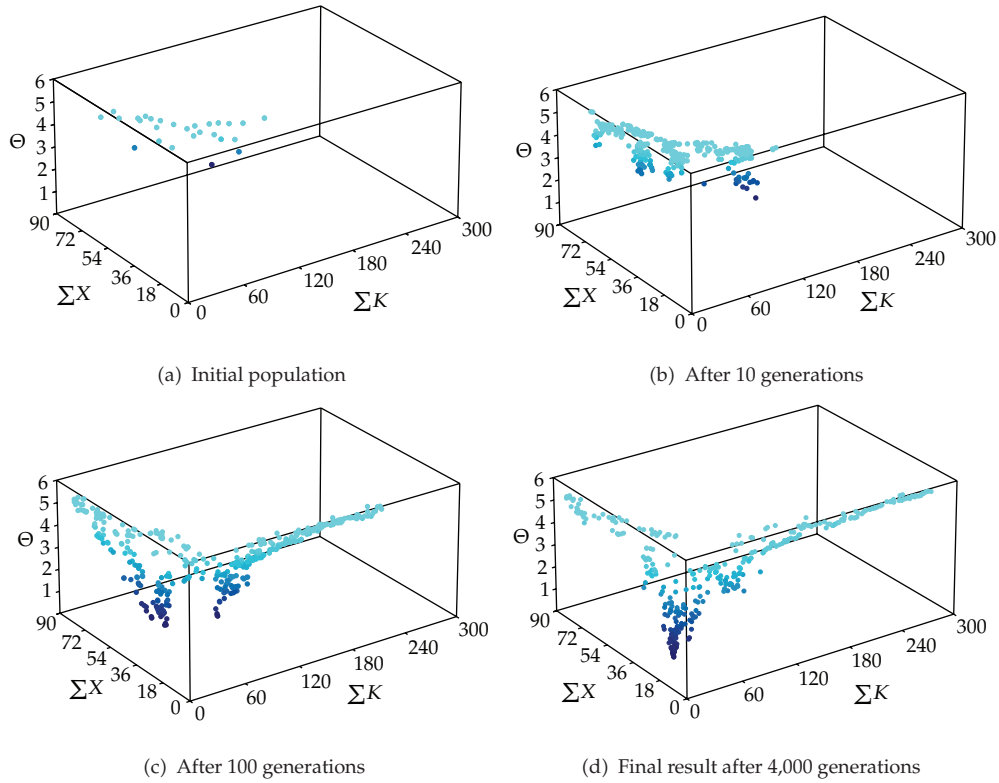

(a) 3-node series network (b) 10-node series network

**Figure 11:** Effect of the mutation rate.




(a) 3-node series network (b) 10-node series network

**Figure 12:** Effect of $\eta$.




(a) Initial population

(b) After 10 generations

(c) After 100 generations

(d) Final result after 4,000 generations

**Figure 13:** Population evolution of a 3-node network.

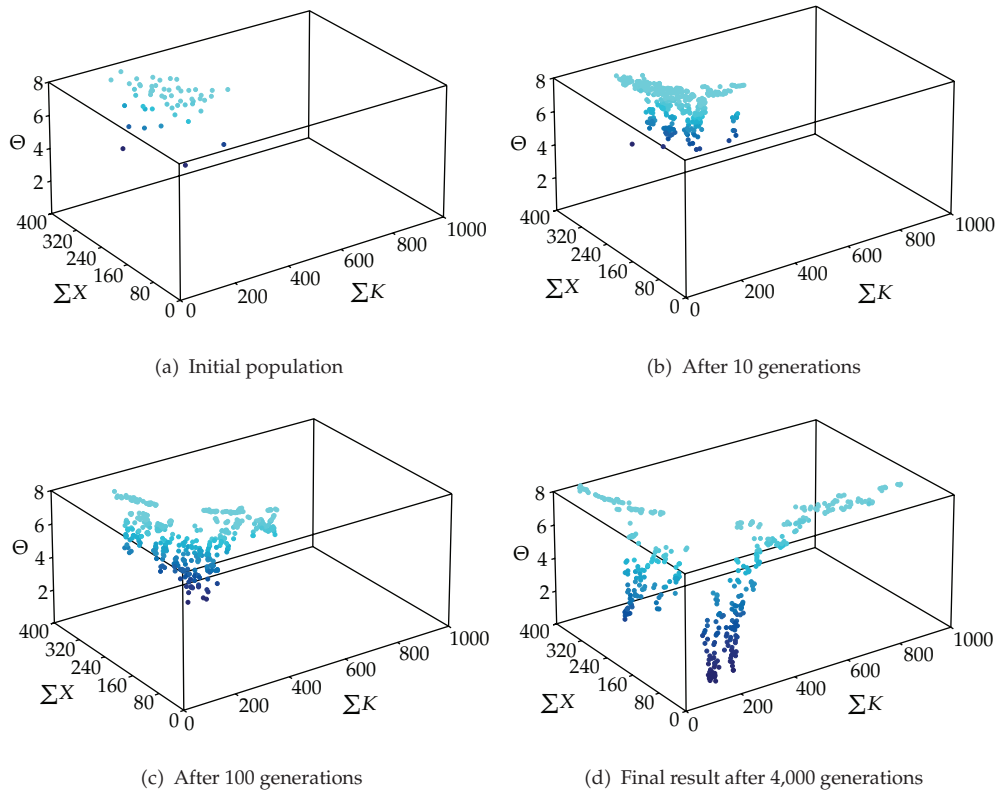
Figure 12 displays the convergence as a function of $\eta$, which controls the dispersion of $\beta_q$ in the SBX operator, (2.19). A further improvement in the convergence speed was not observed for values of $\eta$ greater than 8.

Finally, Figures 13 and 14 illustrate the population evolution, from the starting point to the final generation. They show the population spreading over time to cover an increasing proportion of objective space.

In summary all of the attempted problems could be successfully solved by employing the following combination: a combined use of mutation and SBX, a population size of 400 individuals, a mutation rate of 2%, and a dispersion parameter ($\eta$) of 8. Moreover, the convergence seemed to be fairly independent of the topology (results not shown, for split topologies, merge topologies, and so on), the external arrival ($\mathbf{\Lambda}$), the squared coefficient of variation of the service time ($s^2$), and the number of nodes of the network. Additionally, to ensure that the computation is complete within a finite amount of time, the maximum number of generations (`numGen`) must be predefined. In this study, `numGen` was set to 4,000 generations.

### 3.2. *Analysis of a Large Complex Network*

The complex network presented in Figure 1 was extracted from the literature [20] and analyzed with the proposed method. Three different squared coefficients of variation were




(a) Initial population

(b) After 10 generations

(c) After 100 generations

(d) Final result after 4,000 generations

**Figure 14:** Population evolution of a 10-node network.

analyzed ($s^2 = \{0.5, 1.0, 1.5\}$) at an arrival rate ($\Lambda_1$) of 5.0. First, the convergence speed of the genetic algorithm was confirmed to be robust for this type of problem. The experimental setup was identical to the previous analysis. However, the results indicated that convergence was stable at 2,000 iterations. Moreover, as shown in Figure 15, the convergence was largely independent of the squared coefficient of variation.

The corresponding profiles are shown in Figure 16, including the contour plot and final surface after convergence. For comparison, an exact contour plot of a single-node queue is presented in Figure 2(b), and the resemblance between the two graphs was encouraging. However, the behavior of a given network cannot be predicted without the use of an algorithmic approach such as the one proposed here. A detailed analysis of the results in Figure 16 revealed that many different pairs of buffers and service rates can be selected for a given throughput. Additionally, it is possible to evaluate the results when the buffer size or service rate is so high that it does not have an effect on the throughput (i.e., when the respective contour lines are parallel to the axes). Moreover, with the proposed algorithm, important insights related to the target throughput are obtained. For example, the results in Figure 16(d) suggest that it is easier to increase the throughput from 2.6 to 3.1 (20% improvement) than 4.1 to 4.5 (10% improvement). Contour lines that are far apart indicate that further improvements can be achieved only by dramatically increasing the buffer size and service rate.

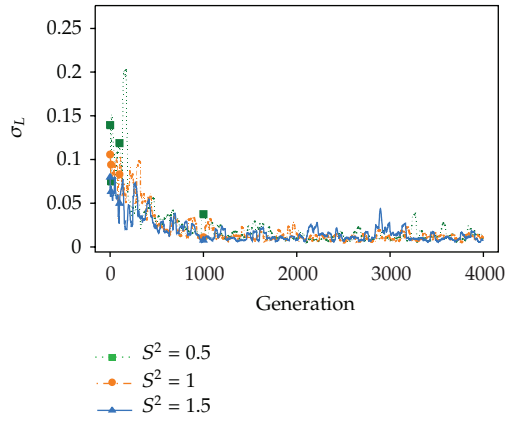
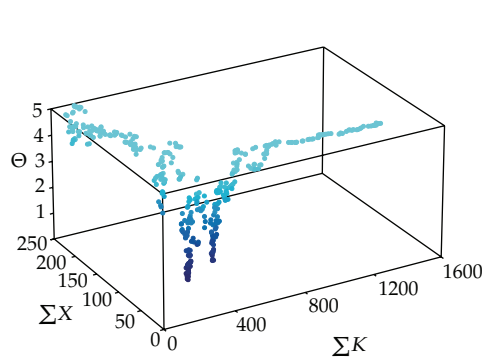
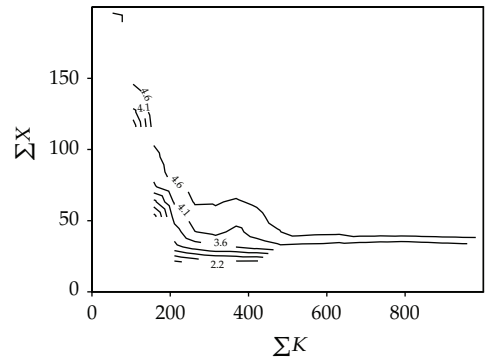


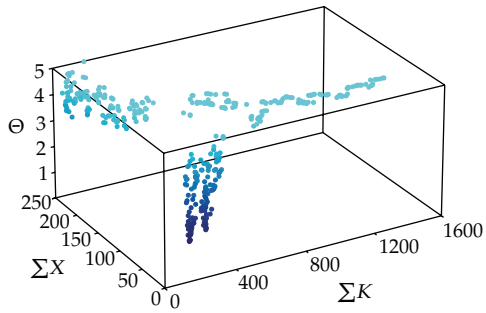

**Figure 15:** Convergence of the 16-node network from Figure 1.



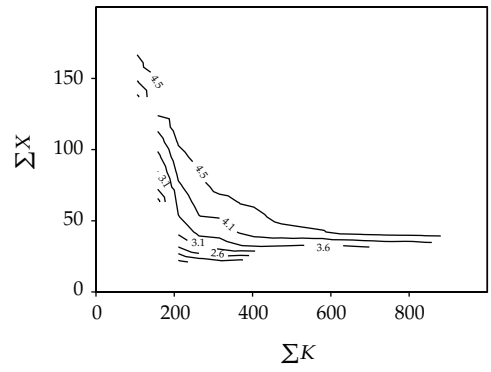

(a) Final surface for $s^2 = 0.5$




(b) Contour plot for $s^2 = 0.5$




(c) Final surface for $s^2 = 1.5$




(d) Contour plot for $s^2 = 1.5$

**Figure 16:** Final results for the 16-node network from Figure 1.

## 4. Conclusions and Final Remarks

In this study, a multiobjective approach was developed to maximize the throughput of single server, general queueing networks. By combining the generalized expansion method (GEM) with a multiobjective evolutionary algorithm (MOEA), insightful Pareto curves were obtained. These curves display the tradeoff between throughput, total buffer allocation, and overall service allocation.

Future investigations should be conducted to determine the applicability of this methodology for the determination of other optimal conditions in finite queueing networks. For instance, this method could be applied to optimize throughput in finite general, multiserver queueing networks or queueing networks with loops. Thus, the method could be used to model systems that lead to a reverse stream of products. Moreover, future research should be conducted to evaluate the algorithms in real-life situations.

## Acknowledgments

The research of Frederico Cruz has been partially funded by CNPq (*Conselho Nacional de Desenvolvimento Científico e Tecnológico*; Grants, 201046/1994-6, 301809/1996-8, 307702/2004-9, 472066/2004-8, 304944/2007-6, 561259/2008-9, 553019/2009-0, 550207/2010-4, 501532/2010-2, 303388/2010-2), by CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior; grant BEX-0522/07-4), and by FAPEMIG (Fundação de Amparo à Pesquisa do Estado de Minas Gerais; Grants, CEX-289/98, CEX-855/98, TEC-875/07, CEX-PPM-00401/08, CEX-PPM-00390-10). The Brazilian government funding agencies mentioned above had no role in the study.

## References

[1] J. Li, E. Enginarlar, and S. M. Meerkov, "Conservation of filtering in manufacturing systems with unreliable machines and finished goods buffers," *Mathematical Problems in Engineering*, vol. 2006, Article ID 27328, 12 pages, 2006.

[2] A. B. Hu and S. M. Meerkov, "Lean buffering in serial production lines with Bernoullimachines," *Mathematical Problems in Engineering*, vol. 2006, Article ID 17105, 24 pages, 2006.

[3] A. M. A. Youssef and H. A. ElMaraghy, "Performance analysis of manufacturing systems composed of modular machines using the universal generating function," *Journal of Manufacturing Systems*, vol. 27, no. 2, pp. 55–69, 2008.

[4] I. Dimitriou and C. Langaris, "A repairable queueing model with two-phase service, start-up times and retrial customers," *Computers and Operations Research*, vol. 37, no. 7, pp. 1181–1190, 2010.

[5] F. R. B. Cruz, F. S. Q. Alves, H. C. Yehia, L. A. C. Pedrosa, and L. Kerbache, "Upper bounds on performance measures of heterogeneous *M/M/c* queues," *Mathematical Problems in Engineering*, vol. 2011, Article ID 702834, 18 pages, 2011.

[6] J. H. Harris and S. G. Powell, "An algorithm for optimal buffer placement in reliable serial lines," *IIE Transactions*, vol. 31, no. 4, pp. 287–302, 1999.

[7] R. Andriansyah, T. van Woensel, F. R. B. Cruz, and L. Duczmal, "Performance optimization of open zero-buffer multi-server queueing networks," *Computers and Operations Research*, vol. 37, no. 8, pp. 1472–1487, 2010.

[8] J. M. Smith, F. R. B. Cruz, and T. van Woensel, "Topological network design of general, finite, multi-server queueing networks," *European Journal of Operational Research*, vol. 201, no. 2, pp. 427–441, 2010.

[9] N. Koizumi, E. Kuno, and T. E. Smith, "Modeling patient flows using a queuing network with blocking," *Health Care Management Science*, vol. 8, no. 1, pp. 49–60, 2005.

[10] A. M. de Bruin, A. C. van Rossum, M. C. Visser, and G. M. Koole, "Modeling the emergency cardiac in-patient flow: an application of queuing theory," *Health Care Management Science*, vol. 10, no. 2, pp. 125–137, 2007.
[11] C. Osorio and M. Bierlaire, "An analytic finite capacity queueing network model capturing the propagation of congestion and blocking," *European Journal of Operational Research*, vol. 196, no. 3, pp. 996–1007, 2009.
[12] F. R. B. Cruz, J. M. Smith, and D. C. Queiroz, "Service and capacity allocation in *M/G/c/c* state-dependent queueing networks," *Computers and Operations Research*, vol. 32, no. 6, pp. 1545–1563, 2005.
[13] F. R. B. Cruz, T. van Woensel, J. M. Smith, and K. Lieckens, "On the system optimum of traffic assignment in *M/G/c/c* state-dependent queueing networks," *European Journal of Operational Research*, vol. 201, no. 1, pp. 183–193, 2010.
[14] N. U. Ahmed and X. H. Ouyang, "Suboptimal RED feedback control for buffered TCP flow dynamics in computer network," *Mathematical Problems in Engineering*, vol. 2007, Article ID 54683, 17 pages, 2007.
[15] J. Chen, C. Hu, and Z. Ji, "An improved ARED algorithm for congestion control of network transmission," *Mathematical Problems in Engineering*, vol. 2010, Article ID 329035, 14 pages, 2010.
[16] L. Tang, H.-S. Xi, J. Zhu, and B.-Q. Yin, "Modeling and optimization of m/g/1 -type queueing networks: an efficient sensitivity analysis approach," *Mathematical Problems in Engineering*, vol. 2010, Article ID 130319, 16 pages, 2010.
[17] G. M. Gontijo,, G. S. Atuncar, F. R. B. Cruz, and L. Kerbache, "Performance evaluation and dimensioning of *GIX/M/c/N* systems through kernel estimation," *Mathematical Problems in Engineering*, vol. 2011, Article ID 348262, 20 pages, 2011.
[18] K. Chaudhuri, A. Kothari, R. Pendavingh, R. Swaminathan, R. Tarjan, and Y. Zhou, "Server allocation algorithms for tiered systems," *Algorithmica*, vol. 48, no. 2, pp. 129–146, 2007.
[19] D. A. Menascé, "QoS issues in web services," *IEEE Internet Computing*, vol. 6, no. 6, pp. 72–75, 2002.
[20] J. M. Smith and F. R. B. Cruz, "The buffer allocation problem for general finite buffer queueing networks," *IIE Transactions*, vol. 37, no. 4, pp. 343–365, 2005.
[21] D. G. Kendall, "Stochastic processes occurring in the theory of queues and their analysis by the method of imbedded Markov chains," *Annals Mathematical Statistics*, vol. 24, pp. 338–354, 1953.
[22] J. G. Shanthikumar and D. D. Yao, "Optimal server allocation in a systemofmulti-server stations," *Management Science*, vol. 33, no. 9, pp. 1173–1180, 1987.
[23] L. E. Meester and J. G. Shanthikumar, "Concavity of the throughput of tandem queueing systems with finite buffer storage space," *Advances in Applied Probability*, vol. 22, no. 3, pp. 764–767, 1990.
[24] F. R. B. Cruz, "Optimizing the throughput, service rate, and buffer allocation in finite queueing networks," *Electronic Notes in Discrete Mathematics*, vol. 35, pp. 163–168, 2009.
[25] F. R. B. Cruz, A. R. Duarte, and T. van Woensel, "Buffer allocation in general single-server queueing networks," *Computers and Operations Research*, vol. 35, no. 11, pp. 3581–3598, 2008.
[26] V. Chankong and Y. Y. Haimes, *Multiobjective Decision Making: Theory and Methodology*, Elsevier, Amsterdam, The Netherlands, 1983.
[27] L. Kerbachea and J. M. Smith, "The generalized expansion method for open finite queueing networks," *European Journal of Operational Research*, vol. 32, no. 3, pp. 448–461, 1987.
[28] L. Kerbache and J. M. Smith, "Asymptotic behavior of the expansion method for open finite queueing networks," *Computers and Operations Research*, vol. 15, no. 2, pp. 157–169, 1988.
[29] L. Kerbache and J. M. Smith, "Multi-objective routing within large scale facilities using open finite queueing networks," *European Journal of Operational Research*, vol. 121, no. 1, pp. 105–123, 2000.
[30] E. G. Carrano, L. A. E. Soares, R. H. C. Takahashi, R. R. Saldanha, and O. M. Neto, "Electric distribution network multiobjective design using a problem-specific genetic algorithm," *IEEE Transactions on Power Delivery*, vol. 21, no. 2, pp. 995–1005, 2006.
[31] J. M. Smith, "Optimal design and performance modelling of *M/G/1/K* queueing systems," *Mathematical and Computer Modelling*, vol. 39, no. 9-10, pp. 1049–1081, 2004.
[32] T. Kimura, "A transform-free approximation for the finite capacity *M/G/s* queue," *Operations Research*, vol. 44, no. 6, pp. 984–988, 1996.
[33] J. M. Smith, "*M/G/c/K* blocking probability models and system performance," *Performance Evaluation*, vol. 52, no. 4, pp. 237–267, 2003.
[34] D. Gross, J. F. Shortle, J. M. Thompson, and C. M. Harris, *Fundamentals of Queueing Theory*, Wiley-Interscience, New York, NY, USA, 4th edition, 2009.
[35] J. Y. Cheah and J. M. Smith, "Generalized *M/G/C/C* state dependent queueing models and pedestrian traffic flows," *Queueing Systems*, vol. 15, no. 1-4, pp. 365–386, 1994.

[36] F. R. B. Cruz, P. C. Oliveira, and L. Duczmal, "State-dependent stochastic mobility model in mobile communication networks," *Simulation Modelling Practice and Theory*, vol. 18, no. 3, pp. 348–365, 2010.
[37] J. Labetoulle and G. Pujolle, "Isolation method in a network of queues," *IEEE Transactions on Software Engineering*, vol. 6, no. 4, pp. 373–381, 1980.
[38] K. Deb, *Multi-Objective Optimisation Using Evolutionary Algorithms*, Wiley, 2001.
[39] C. A. Coello Coello, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer, 2002.
[40] F. T. Lin, "Solving the knapsack problem with imprecise weight coefficients using genetic algorithms," *European Journal of Operational Research*, vol. 185, no. 1, pp. 133–145, 2008.
[41] H. I. Calvete, C. Galé, and P. M. Mateo, "A new approach for solving linear bilevel problems using genetic algorithms," *European Journal of Operational Research*, vol. 188, no. 1, pp. 14–28, 2008.
[42] C. A. Coello Coello, "An updated survey of GA-based multiobjective optimization techniques," in *Proceedings of the ACM Computing Surveys*, vol. 32, pp. 109–143, 2000.
[43] C. M. Fonseca and P. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary Computing*, vol. 3, no. 1, pp. 1–16, 1995.
[44] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
[45] T. Bäck, D. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*, Institute of Physics Publishing and Oxford University Press, 1997.
[46] X. B. Hu and E. Di Paolo, "An efficient Genetic Algorithm with uniform crossover for the multi-objective airport gate assignment problem," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '07)*, pp. 55–62, Singapore, September 2007.
[47] G. Sywerda, "Uniformcrossover in genetic algorithms," in *Proceedings of the 3rd International Conference on Genetic Algorithms*, pp. 2–9, Morgan Kaufmann Publishers, San Francisco, Calif, USA, 1989.
[48] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, pp. 115–148, 1995.
[49] K. Deb and H.-G. Beyer, "Self-adaptive genetic algorithms with simulated binary crossover," Tech. Rep. CI-61/99, Department of Computer Science/XI, University of Dortmund, Dortmund, Germany, 1999.
[50] O. Rudenko and M. Schoenauer, "A steady performance stopping criterion for Pareto-based evolutionary algorithms," in *Proceedings of the 6th International Multi-Objective Programming and Goal Programming Conference*, Hammamet, Tunisia, 2004.
[51] L. Martí, J. García, A. Berlanga, and J. M. Molina, "A cumulative evidential stopping criterion for multiobjective optimization evolutionary algorithms," in *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO '07)*, pp. 2835–2842, ACM, New York, NY, USA, 2007.
[52] D. C. Montgomery, *Design and Analysis of Experiments*, John Wiley & Sons, New York, NY, USA, 6th edition, 2004.
[53] F. R. B. Cruz, T. van Woensel, and J. M. Smith, "Buffer and throughput trade-offs in *M/G/1/K* queueing networks: a bi-criteria approach," *International Journal of Production Economics*, vol. 125, no. 2, pp. 224–234, 2010.
[54] R. Mathieu, L. Pittard, and G. Anandalingam, "Genetic algorithms based approach to bi-level linear programming," *Recherche Opérationnelle/Operations Research*, vol. 28, no. 1, pp. 1–21, 1994.