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

A Multiobjective Optimization Approach for General Finite Queueing Networks

N. L. C. Brito, A. R. Duarte, & F. R. B. Cruz

Relatório Técnico RTP-02/2012

Relatório Técnico
Série Pesquisa

A Multiobjective Optimization Approach for General Finite Queueing Networks

N. L. C. Brito · A. R. Duarte · F. R. B. Cruz

Received: date / Accepted: date

Abstract In this paper a multi-objective algorithm to simultaneously optimize the total number of buffers, the overall service rate, and the throughput of a general-service finite queueing network is studied. These conflicting objectives are optimized by means of a multi-objective genetic algorithm, designed to produce solutions for more than one objective. Computational experiments are shown, in order to determine the efficacy and efficiency of the approach. Instigating news insights are given.

Keywords Network of queues · Multi-objective optimization · Throughput maximization · Genetic algorithms

Mathematics Subject Classification (2000) 60K20 · 90B22

1 Introduction

Our focus here is on single-server queueing networks with exponentially distributed inter-arrival times and generally distributed service times, configured in an arbitrary acyclic topology (see Fig. 1). More specifically, the focus is on networks of $M/G/1/K$ queues, which in Kendall (1953) notation stands for **M**arkovian arrivals, **G**enerally distributed service times, a single server, and the total capacity of K items, *including* the item in service.

N. L. C. Brito
Departamento de Ciências Exatas, Universidade Estadual de Montes Claros, 39401-089 - Montes Claros - MG, Brazil
E-mail: nilson.brito@unimontes.br

A. R. Duarte
Departamento de Matemática, Universidade Federal de Ouro Preto, 35400-000 - Ouro Preto - MG, Brazil
E-mail: anderson@iceb.ufop.br

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

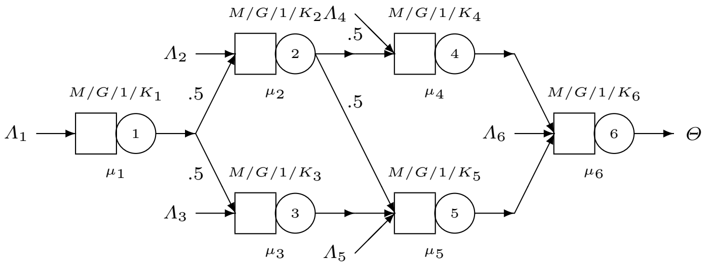


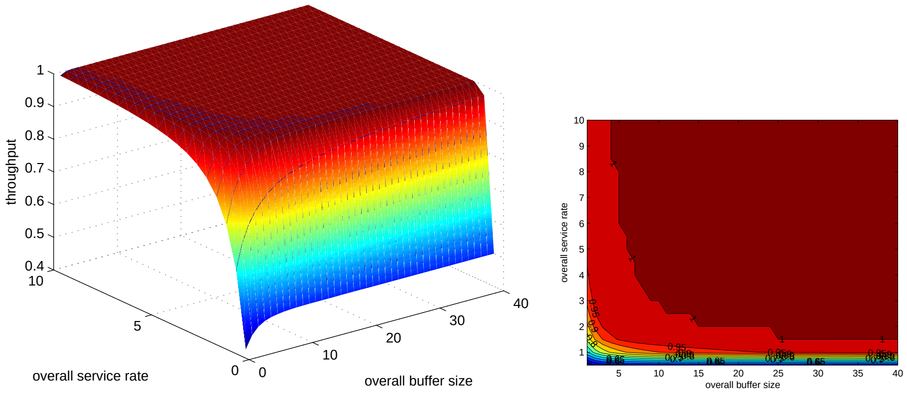
Fig. 1 An $M/G/1/K$ queueing network

Given the topology and the external arrival rates ($\boldsymbol{\Lambda} = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_n\}$), our goal is to obtain the maximum throughput (Θ) by means of the minimum number of buffers ($\mathbf{K} = \{K_1, K_2, \ldots, K_n\}$) and the minimum service rates ($\boldsymbol{\mu} = \{\mu_1, \mu_2, \ldots, \mu_n\}$). Potential users of these queueing models include computer scientists and engineers. Indeed, these models may help to understand and to improve various real-life systems, including manufacturing (Youssef and ElMaraghy 2008), production (Andriansyah et al 2010) and health (Osorio and Bierlaire 2009) systems, urban or pedestrian traffic (Cruz et al 2010), computer and communication systems (Gontijo et al 2011), and web-based applications with tiered configurations (Chaudhuri et al 2007).

There is a trade-off between the overall number of buffers, the service rates, and the resulting throughput. Because buffers and services can be very expensive, the overall buffer and service capacity should not be large. On the other hand, the highest possible network throughput should be reached. Unfortunately, the throughput is directly affected by the number of buffers allocated and the service rates. Indeed, if the buffer and service capacity reduces there will be in general an undesirable reduction in the throughput. In Fig. 2 it is possible to observe this behavior , which shows Θ for a single $M/G/1/K$ queue with $cv^2 = 1.5$ (squared coefficient of variation of the service time) and $\Lambda = 1$ users per time unit (external arrival rate), as a function of several values for buffer size, K, and service rate, μ (see Equations 4 and 10), as well as the respective contour plot.

Similar throughput behavior is also observed in a network of queues, as we shall show shortly. Notice that the surface of the plot shown in Fig. 2 is smooth. Also suggested is convexity. Similar results were reported for simple queueing networks (Meester and Shanthikumar 1990). However, the top surface flatness represents trouble for the traditional optimization algorithms. Indeed, Smith and Cruz (2005) reported a successful optimization algorithm based on Powell method, coupled with multiple starts to avoid premature convergence to local optima.

In this study, an optimization approach is presented to simultaneously optimize the total number of buffers, the overall service rate, and the throughput of networks of $M/G/1/K$ queues. The proposed method produces a set of efficient solutions for more than one objective in the objective function (Chankong and Haimes 1983). With the proposed approach, the decision maker is able to evaluate the effect of solution replacement. Moreover, the multi-objective 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) throughput *versus* service rate and buffer size

(b) contour plot

Fig. 2 Behavior of a single $M/G/1/K$ queue with an arrival rate $\Lambda = 1$ user per time unit

This paper is organized as follows. A multi-objective evolutionary algorithm specifically developed to multi-objective optimization is presented in Sec. 2, along with the GEM, a performance evaluation tool used to approximate the throughput. In Sec. 3, the results of a comprehensive set of computational experiments are presented to show the efficiency of the approach. Finally, Sec. 4 concludes this paper with final remarks and suggestions for future research in the area.

2 Algorithms

2.1 Mathematical Programming Formulation

From a modeling point of view, the throughput maximization problem can be defined by a mixed-integer mathematical programming formulation, in which the total buffer and server costs are minimized and the throughput is maximized subject to integer buffer allocations and non-negative service rates. By defining a queueing network as a digraph $G(N, A)$, where N is a finite set of nodes (queues) and A is a finite set of arcs (pair of connected queues), a possible formulation is:

$$\text{minimize}\, F(\mathbf{K}, \boldsymbol{\mu}), \tag{1}$$

subject to

$$K_i \in \{1, 2, \ldots\},\ \forall i \in N, \tag{2}$$

$$\mu_i \geq 0,\ \forall i \in N, \tag{3}$$

where the decision variables K_i and μ_i indicate the total capacity of the service and the service rate for the ith $M/G/1/K$ queue, respectively. The objective functions, $F(\mathbf{K}, \boldsymbol{\mu}) \equiv \Big(f_1(\mathbf{K}), f_2(\boldsymbol{\mu}), -f_3(\mathbf{K}, \boldsymbol{\mu}\Big)$, 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, $f_3(\mathbf{K}, \boldsymbol{\mu}) = \Theta(\mathbf{K}, \boldsymbol{\mu})$.

Such formulation (1)–(3) was successfully used by Cruz et al (2012). However, notice that in the literature the throughput is commonly modeled as a constraint that must be greater than a threshold $\Theta_\mathcal{T}$ value rather than as an objective that must be maximized (see, Andriansyah et al (2010), for instance). The problem is that to solve this single-objective version of the problem the throughput constraint must be relaxed and to establish an appropriate $\Theta_\mathcal{T}$ is not a trivial task. Moreover, often a small decrease in the throughput results in a significant reduction in the buffer and service allocation. Such a trade-off between throughput and the number of buffers and service rates unfortunately will not be apparent in an *equivalent* single-objective formulation (which usually combines the multiple-objective formulation into a single-objective formulation by means of a vector of weights, $\boldsymbol{\omega}$). Additionally the determination of vector $\boldsymbol{\omega}$ is difficult and often leads to arbitrary single-objective formulations.

In this paper, a multi-objective evolutionary algorithm (MOEA) is used in combination with a generalized expansion method (GEM), which is a well-known method for obtaining accurate approximations of queueing network performance (Kerbache and Smith 1987). MOEAs are particularly suitable for multi-objective problems and have been shown to perform well in similar multi-objective problems of networks (*e.g.*, see Carrano et al 2006, and references therein). The algorithms will be presented in two parts. Initially, the performance evaluation algorithm will be described. Then, the proposed optimization algorithm will be detailed.

2.2 Performance Evaluation - Single Queues

When the interest is on single queues (not exactly the case here), the throughput $\Theta(\mathbf{K}, \boldsymbol{\mu})$ is:

$$\Theta(\mathbf{K}, \boldsymbol{\mu}) = \lambda(1 - p_K), \tag{4}$$

where λ is the external arrival rate and p_K is the called blocking probability, which is the probability that an item finds the system full (that is, the number of items in the systems is equal to the total capacity K). Thus, the problem of finding $\Theta(\mathbf{K}, \boldsymbol{\mu})$ reduces to determining p_K.

For the special case of pure Markovian systems (*i.e.*, $M/M/1/K$ queues), the blocking probability expression may be easily found in the queueing theory literature (*e.g.,* Gross et al 2009):

$$p_K = \frac{(1-\rho)\rho^K}{1-\rho^{K+1}}. \tag{5}$$

valid for $\rho < 1$, where $\rho \equiv \lambda/\mu$ is the system utilization. Relaxing the integrality constraint of K, it is possible to express in closed form the optimal buffer allocation for $M/M/1/K$ queues in terms of ρ and p_K:

$$K_\mathrm{M} = \left\lceil \frac{\ln\left(\frac{p_K}{1-\rho+p_K\rho}\right)}{\ln(\rho)} \right\rceil, \tag{6}$$

where $\lceil x \rceil$ is the smallest integer not superior to x. Consequently, it is possible to derive the optimal buffer allocation for $M/M/1/K$ queues:

$$x_{\mathrm{M}} = K_{\mathrm{M}} - 1. \tag{7}$$

For general-service multi-server queues $M/G/c/K$, the blocking probability must be derived by approximate techniques. In particular, Smith and Cruz (2005) have shown in a previous paper that a two-moment approximation based on the Markovian expression, Eq. (7), is quite effective:

$$x_{\epsilon}(cv^2) = x_{\mathrm{M}} + \mathrm{INT}\left[\frac{(cv^2-1)\sqrt{\rho}}{2} x_{\mathrm{M}}\right], \tag{8}$$

where INT[x] is the integer part of x. In particular, for single-server queues, $M/G/1/K$, given ρ and cv^2, the optimal buffer allocation may be written as:

$$x_{\epsilon} = \frac{\left[\ln\left(\frac{p_K}{1-\rho+p_K\rho}\right) + \ln(\rho)\right]\left(2 + \sqrt{\rho}cv^2 - \sqrt{\rho}\right)}{2\ln(\rho)}. \tag{9}$$

Finally, one can isolate p_K and determine a closed-form expression for the blocking probability in $M/G/1/K$ queues, as a function of K (note that for $M/G/1/K$ queues, $K = 1 + x_{\epsilon}$):

$$p_K = \frac{(1-\rho)\rho^{\left(\frac{2+\sqrt{\rho}cv^2-\sqrt{\rho}+2(K-1)}{2+\sqrt{\rho}cv^2-\sqrt{\rho}}\right)}}{1-\rho^{\left(2\frac{2+\sqrt{\rho}cv^2-\sqrt{\rho}+(K-1)}{2+\sqrt{\rho}s^2-\sqrt{\rho}}\right)}}. \tag{10}$$

2.3 Performance Evaluation - Networks of Queues

For networks of queues, the estimation of the throughput is made by means of the generalized expansion method (GEM), which is an algorithm that has been successfully used to estimate the performance of arbitrarily configured, finite queueing, acyclic networks (Kerbache and Smith 1987). The method is a combination of node-by-node decomposition and repeated trials, in which each queue is analyzed separately, and corrections are made to account for interrelated effects between network queues.

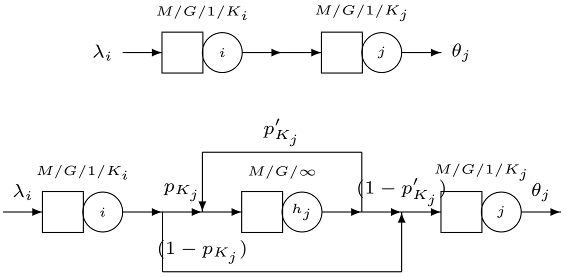


Fig. 3 Generalized expansion method

As described in details byKerbache and Smith (1987), the GEM creates for each finite node j an auxiliary vertex (h_j) that is modeled as a $M/G/\infty$ queue, as seen in Fig. 3. For each entity placed into the system, vertex j may be blocked (with probability p_{K_j}), or may be unblocked (with probability $1 - p_{K_j}$). When blocking occurs, the entities are rerouted to vertex h_j and are delayed while node j is busy. Vertex h_j records the time an entity has to wait before entering vertex j and computes the effective arrival rate to vertex j.

The ultimate goal of GEM is to provide an approximation procedure that updates the service rates of upstream nodes and takes into account blocking in services caused by downstream nodes. The approximation below is based on the corrected blocking probabilities ($\tilde{p}_{K_i}$) of all nodes, which will provide an accurate estimation for the overall throughput (Θ).

$$\tilde{\mu}_i^{-1} = \mu_i^{-1} + p_{K_j}(\mu_h')^{-1}. \tag{11}$$

Notice that the performance evaluation process must be conducted in a specific order. The performance evaluation of the network under study, defined as digraph $G(N, A)$, is presented in Fig. 4. The algorithm accounts for blocked services at upstream nodes, resulting in effective service rates that are reduced, in accordance with Eq. (11). Note that the performance evaluation algorithm is a variant of Dijkstra's labeling algorithm for the determination of shortest paths (Dijkstra 1959). For instance, in the network illustrated in Fig.1, a valid evaluation sequence is 1 → 2 → 4 → 3 → 5 → 6. Specifically, the sequence must make it sure a node only will be assessed after all of its predecessors. Assuming that circuits are not present in $G(N, A)$, the GEM has a running time complexity of $\mathcal{O}(N^2)$, which is in accordance with Dijkstra's algorithm.

```
algorithm
    read graph, $G(N, A)$
    read routing probabilities, $p_{[ij]}, \forall (i, j) \in A$
    read external arrival rates and service rates, $\Lambda_i, \mu_i, \forall i \in N$
    initialize set of labeled nodes, $P \leftarrow \emptyset$
    while $P \neq V$
        choose $j$ such that $(j \in N)$ and $(j \notin P)$
        if $\{i | (i, j) \in A\} \subseteq P$ then
            /* compute performance measures */
            compute $p_{K\,j}\ \theta_j$
            /* forward information to successors */
            for $\forall\, k \in \{k' | (j, k') \in A\}$ then
                $\lambda_k \leftarrow \lambda_k + \theta_j p_{[jk]}$
            end for
            /* label node as pre-evaluated */
            $P \leftarrow P \cup \{j\}$
        end if
    end while
end algorithm
```

Fig. 4 Performance evaluation algorithm

2.4 Optimization Algorithm

For the network under consideration, MOEAs seems to be a suitable choice for the multi-objective 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 (Deb 2001). The population of points that converge to an optimal value are obtained through the application of the genetic operators, *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 were reported for single-objective applications (Lin 2008) and for multiple-objective applications (Carrano et al 2006). The instance of MOEA used in this study is based upon the elitist non-dominated sorting genetic (NSGA-II) algorithm of Deb et al (2002), which is shown in Fig. 5. In the application of GAs for multi-objective optimization, the selection and elitism operators must be specifically structured to correctly identify optimal conditions as it will be shown shortly.

```
algorithm
    read graph, arrival, service rates, $G(N, A), \Lambda_i \; \forall \, i \in N$
    $P_1 \leftarrow$ GenerateInitialPopulation(popSize)
    for $i = 1$ until numGen do
        /* generate offspring by crossover and mutation */
        $Q_i \leftarrow$ MakeNewPop($P_i$)
        /* combine parent and offspring */
        $R_i \leftarrow P_i \cup Q_i$
        /* find non-dominated fronts $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \ldots)$ */
        $\mathcal{F} \leftarrow$ FastNonDominatedSort($R_i$)
        /* find new population by */
        /* the crowding-distance-assignment */
        $P_{i+1} \leftarrow$ GenerateNewPopulation($R_i$)
    end for
    $P_{\texttt{numGen}+1} \leftarrow$ ExtractParetoSet($P_{\texttt{numGen}}$)
    write $P_{\texttt{numGen}+1}$
end algorithm
```

Fig. 5 Elitist multi-objective genetic algorithm (NSGA-II)

Elitism is based on the concept of dominance. Point $\mathbf{x}_i = (x_{i_1}, x_{i_2}, \ldots, x_{i_n})$ dominates point $\mathbf{x}_j = (x_{j_1}, x_{j_2}, \ldots, x_{j_n})$ 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). To perform elitism, the fast non-dominated sorting algorithm is employed (Deb et al 2002). 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 non-dominated, 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 (Deb et al 2002).

Selection is performed by sequentially choosing points from each non-dominated front $(\mathcal{F}_1, \mathcal{F}_2, \ldots)$ until the number of required individuals for the next iteration is obtained. Some decision must be made if the maximum number of individuals is exceeded after the addition of a group of individuals from front $\mathcal{F}_i$. One possibility is to compute a measure of diversity, such as the crowding distance defined by Deb et al (2002), to ensure the highest diverse population. Thus, only the points with the largest crowding distance are kept for future iterations.

Crossover and mutation are dependent on the application, as well known. For the problem at hand, the *uniform crossover* mechanism was selected (Bäck et al 1997), which is popular in multivariable encodings due to its efficiency in identifying, inheriting, and protecting common genes, as well as re-combining non-common genes (Hu and Di Paolo 2007). In this mechanism, crossover is performed for each variable with a probability (`rateCro`), in accordance with the crossover operator. The crossover operator used in the algorithm is the *simulated binary crossover operator* (SBX) (Deb and Beyer 1999). SBX is quite convenient for real-coded GAs because of its ability to simulate *binary crossover operators* avoiding re-encoding the variables. The children ($x_{i,(\bullet,t+1)}$) are calculated from the parents ($x_{i,(\bullet,t)}$) according to the following equations

$$x_{i,(1,t+1)} = 0.5\Big[(1+\beta)x_{i,(1,t)} + (1-\beta)x_{i,(2,t)}\Big], \tag{12}$$

$$x_{i,(2,t+1)} = 0.5\Big[(1-\beta)x_{i,(1,t)} + (1+\beta)x_{i,(2,t)}\Big], \tag{13}$$

where β is a random variable with the following density function:

$$f(\beta) = \begin{cases} 0.5(\eta+1)\beta^{\eta}, & \text{if } \beta \leq 1, \\ 0.5(\eta+1)\frac{1}{\beta^{\eta+2}}, & \text{otherwise}, \end{cases} \tag{14}$$

noticing that Equations (12) and (13) are designed to create children solutions that posses a similar search power to a single-point crossover of binary-coded GAs Deb and Agrawal (1995). By adjusting η, several different weights (β) can be generated to produce children that are more (small η) or less (large η) similar to their parents.

For each individual gene (each decision variables K_i or μ_i), the mutation scheme occurs with a specific probability (`rateMut`). As suggested by Deb and Agrawal (1995), Gaussian perturbations were added to the decision variables, $K_i + \varepsilon_i$ and $\mu_i + \varepsilon_{N+i}$, for all $i \in N$, with $\varepsilon_i \sim \mathcal{N}(0,1)$, $i \in \{1, 2, \ldots, 2N\}$.

Finally, to ensure feasibility of constraints (2) and (3) after crossover and mutation, the integer variables values must be rounded accordingly and all the variables readjusted by applying reflection operators as follows

$$K_{\text{rfl}_i} = K_{\text{lowlim}} + |K_i - K_{\text{lowlim}}|, \tag{15}$$

and

$$\mu_{\text{rfl}_i} = \mu_{\text{lowlim}_i} + |\mu_i - \mu_{\text{lowlim}_i}|, \tag{16}$$

where K_{lowlim} is the lower limit of buffer allocation (*i.e.*, $K_{\text{lowlim}} = 1$) and μ_{lowlim_i} is the lower limit of service allocation (to ensure that $\rho < 1$ holds). Notice that K_i and μ_i are the resulting values after crossover and mutation, and K_{rfl_i} and μ_{rfl_i} are the results after reflection. The proposed scheme always generates feasible solutions without avoiding or favoring any particular solution.

2.5 Convergence Issues

Recently, the stopping criterion of multi-objective optimization evolutionary algorithms has been analyzed in detail. 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 (2004) 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 (2004) 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. Rudenko and Schoenauer (2004) proposed a local stopping criterion that computes a measure of the stability of non-dominated solutions after each iteration based on the stabilization of the maximal crowding distance, d_l, measured over L generations and calculated by the following standard deviation:

$$\sigma_L = \sqrt{\frac{1}{L}\sum_{l=1}^{L}(d_l - \bar{d}_L)^2}, \tag{17}$$

in which $\bar{d}_L$ is the average of d_l over L generations and the criterion $\sigma_L < \delta_{\text{lim}}$ should indicate when MOEA should stop. Rudenko and Schoenauer (2004) suggested that L and δ_{lim} should be set to 40 and 0.02, respectively, which leads to a stopping criteria that is $\sigma_{40} \leq 0.02$.

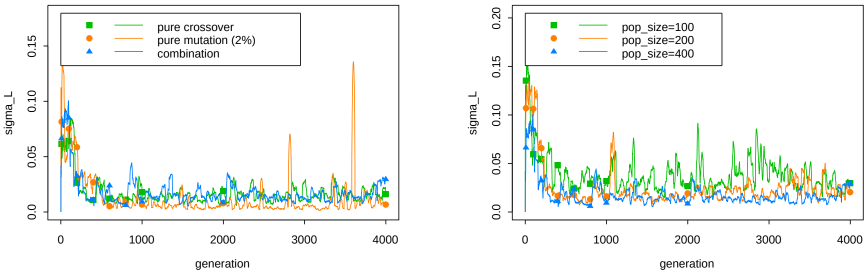
3 Results and Discussion

In order to use a previous implementation of the GEM algorithm that was based on the International Mathematics and Statistics Library (IMSL), the optimization algorithm was coded in FORTRAN. Upon request directly from the authors all codes are available for educational and research purposes. Firstly, the computational experiments were conducted to discover a sub-optimal set of parameters that guarantee rapid convergence. Finally, a detailed analysis of a queueing network was performed.

3.1 Parameter Setup

As indicated by previous studies on GAs, a sub-optimal set of parameters to ensure rapid convergence with a minimal amount of computational effort may be determined without trouble by trial and error. Only results obtained for the network from Fig. 1 are presented, although different topologies of acyclic similarly sized networks were also tested. The results (not presented) were similar.

Convergence is monitored in Fig. 6-a in terms of σ_L along the generations, for combinations of crossover and mutation. It is remarkable that pure mutation could solve the problem but using SBX operator (crossover) may remove instabilities from σ_L. Thus, combining mutation and SBX may be advantageous regardless of the number of queues in the network. Figure 6-b discloses that the population size (`popSize`) affects algorithm convergence, although the population size cannot be arbitrarily increased, as it may risk the processing time.

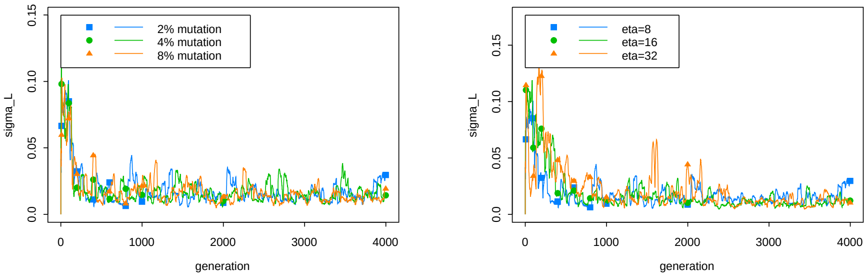


(a) crossover and mutation

(b) population size

Fig. 6 Effects of crossover and mutation and population size

The standard deviation as function of `rateMut` is displayed in Fig. 7-a disclosing that an increase in the mutation rate may speed up the convergence but after a certain rate no further improvements are obtained. Then, mutation rates between up to 2% seemed to respond for the best results, as shown by the experimental results provided. In Fig. 7-b the standard deviation is presented as a function of parameter η which responds for the SBX operator performance. From these experiments, it is possible to conclude that values above 8 are not effective in terms of stability.



(a) mutation rate

(b) η

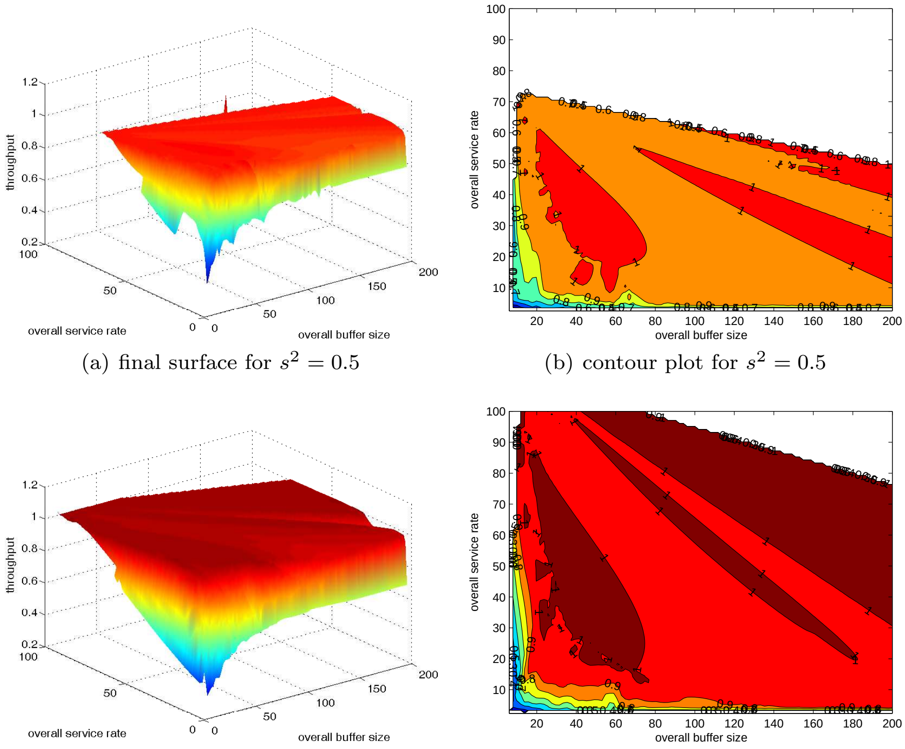
Fig. 7 Effect of the mutation rate and η

As a final word concerning the best group of parameters for the algorithm, one could use the following combination: (i) combined use of SBX and mutation, with

(ii) a mutation rate below 2%, (iii) although greater the better the population, 400 individuals seem to be enough, and (iv) the dispersion parameter, η, should not go above 8. To ensure a finite computation time, a maximum number of generations numGen was set to 4,000. Fortunately, MOEAs are robust enough to perform well in a broad range of problems, as confirmed by the experiments run (not shown).

3.2 Network Analysis

The network presented in Fig. 1 was analyzed with the proposed method. Two different squared coefficients of variation were analyzed, $cv^2 = 0.5$ and 1.5, with arrival rate ($\Lambda_1 = 1.0$). First, the convergence speed of the genetic algorithm was confirmed to be robust for this type of problem. The experimental set-up was identical to the previous analysis. However, the results indicated that convergence was stable at 2,000 iterations.



(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 $s^2 = 1.5$

Fig. 8 Final results for the network from Fig. 1

Figure 8 shows the results. It is possible to see the final population and the respective contour plot. It is remarkable the resemblance between these two contour plots and the exact contour plot for a single queue, Fig. 2-b. The results suggest that the *networks* of queues seem to behave such as an equivalent *single* queue. Unfortunately, it is unknown whether or not it would be possible to derive some sort of algorithm to predict the parameters of such an equivalent single queue.

Additionally note that when the squared coefficient of variation of the service time cv^2 is equal to 0.5, then the contour lines are closer to the origin point (0,0) than when cv^2 is equal to 1.5. Such a behavior is expected since a smaller cv^2 means less variability in the service time. The methodology is shown therefore consistent. Another interesting point is that the contour lines help identify the points from which further increase is not worth in the buffer spaces (or in the service rates) since beyond these points only negligible gains will be reached in the throughput.

Table 1 presents some Pareto efficient solutions for a more detailed analysis. Note that, with this multiobjective methodology, it is possible to identify points from which there is no more interest in increasing the spend on buffer sizes or service rates because the gain in the throughput will be rather narrow. For example, for a $cv^2 = 0.5$, keeping the overall service rate approximately constant one had to increase the overall buffer size by 22% to produce a gain of only 0.01% in the throughput. Similarly, there may be a similar point for the overall service rate. In fact, it can be seen that for an increase of 12% in the overall service rate, an increase of only 0.01% is produced in the throughput, which may be considered negligible. Note also that with $cv^2 = 1.5$ such a phenomenon can occur even more pronounced. It is observed that it may be necessary to increase by 36% the overall buffer size to reach an increase of only 0.6% on the throughput. It is therefore more advantageous to maintain a system with an allocation that produces on output of 99.99% of the input (that is, 0.9999/1.000) than spending 70% more in service rate to raise the output by only 0.01% (ie, raising it to 100% of the arrival rate). These are just some examples of the analyzes that can be done in finite general service queueing networks via the multiobjective methodology.

Table 1 Pareto efficient solutions selected from the computational experiments

cv^2	$\sum_i K_i$	$\Delta\%$	$\sum_i \mu_i$	$\Delta\%$	Θ	$\Delta\%$
0.5	18	-	51.0	-	0.9999	-
	22	22%	51.4	0.8%	1.0000	0.01%
	20	-	46.6	-	0.9999	-
	20	0%	52.1	12%	1.0000	0.01%
1.5	14	-	60.4	-	0.9944	-
	19	36%	61.1	1.1%	0.9999	0.6%
	19	-	61.1	-	0.9999	-
	19	0%	104.0	70%	1.0000	0.01%

4 Conclusions

In order to optimize the throughput, the buffer sizes, and the service rates of single server, general-service queueing networks, a multi-objective approach was presented. The generalized expansion method (GEM) was coupled with a multi-objective genetic algorithm (MOGA) to make it possible to derive insightful Pareto curves displaying the trade-off between throughput and the allocation of buffers and service rates.

Topics for future investigation in this area include extensions to networks of multi-server queues and networks of general-arrival queues, possibly by means of kernels Gontijo et al (2011). Also interesting is to consider different performance measures, such as the WIP, sojourn time, and so on. These are only few examples of possible topics for research.

Acknowledgements This research is supported by the Brazilian agencies, 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), CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior; grant BEX-0522/07-4), and 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, and CEX-PPM-00071-12).

References

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

Bäck T, Fogel D, Michalewicz Z (eds) (1997) Handbook of Evolutionary Computation. Institute of Physics Publishing and Oxford University Press

Carrano EG, Soares LAE, Takahashi RHC, Saldanha RR, Neto OM (2006) Electric distribution network multiobjective design using a problem-specific genetic algorithm. IEEE Transactions on Power Delivery 21(2):995–1005

Chankong V, Haimes YY (1983) Multiobjective Decision Making: Theory and Methodology. Elsevier, Amsterdam, The Netherlands

Chaudhuri K, Kothari A, Pendavingh R, Swaminathan R, Tarjan R, Zhou Y (2007) Server allocation algorithms for tiered systems. Algorithmica 48(2):129–146

Cruz FRB, van Woensel T, Smith JM, Lieckens K (2010) On the system optimum of traffic assignment in $M/G/c/c$ state-dependent queueing networks. European Journal of Operational Research 201(1):183–193

Cruz FRB, Kendall G, While L, Duarte AR, Brito NLC (2012) Throughput maximization of queueing networks with simultaneous minimization of service rates and buffers. Mathematical Problems in Engineering 2012(Article ID 348262):19 pages

Deb K (2001) Multi-objective Optimisation using Evolutionary Algorithms. John Wiley & Sons, Inc., New York, NY

Deb K, Agrawal RB (1995) Simulated binary crossover for continuous search space. Complex Systems 9:115–148

Deb K, Beyer HG (1999) Self-adaptive genetic algorithms with simulated binary crossover. Technical report no. CI-61/99, Department of Computer Science/XI, University of Dortmund, 44221 Dortmund, Germany

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2):182–197

Dijkstra EW (1959) A note on two problems in connection with graphs. Numerical Mathematics 1:269–271

Gontijo GM, Atuncar GS, Cruz FRB, Kerbache L (2011) Performance evaluation and dimensioning of $GIX/M/c/N$ systems through kernel estimation. Mathematical Problems in Engineering 2011(Article ID 348262):20 pages

Gross D, Shortle JF, Thompson JM, Harris CM (2009) Fundamentals of Queueing Theory, 4th edn. Wiley-Interscience, New York, NY, USA

Hu XB, Di Paolo E (2007) An efficient genetic algorithm with uniform crossover for the multi-objective airport gate assignment problem. In: IEEE Congress on Evolutionary Computation, CEC 2007, Singapore, pp 55–62

Kendall DG (1953) Stochastic processes occurring in the theory of queues and their analysis by the method of embedded Markov chains. Annals Mathematical Statistics 24:338–354

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

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

Meester LE, Shanthikumar JG (1990) Concavity of the throughput of tandem queueing systems with finite buffer storage space. Advances in Applied Probability 22(3):764–767

Osorio C, Bierlaire M (2009) An analytic finite capacity queueing network model capturing the propagation of congestion and blocking. European Journal of Operational Research 196(3):996–1007

Rudenko O, Schoenauer M (2004) 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

Smith JM, Cruz FRB (2005) The buffer allocation problem for general finite buffer queueing networks. IIE Transactions 37(4):343–365

Youssef AM, ElMaraghy HA (2008) Performance analysis of manufacturing systems composed of modular machines using the universal generating function. Journal of Manufacturing Systems 27(2):55–69