A Novel Formulation for Multi-objective Optimization of General Finite Single-Server Queueing Networks

Gabriel L Souza*, Anderson R Duarte†, Gladston J P Moreira* and Frederico R B Cruz‡
*Computing Department, Universidade Federal de Ouro Preto, 35400-000 - Ouro Preto - MG, Brazil
†Department of Statistics, Universidade Federal de Ouro Preto, 35400-000 - Ouro Preto - MG, Brazil
‡Department of Statistics, Universidade Federal de Minas Gerais, 31270-901 - Belo Horizonte - MG, Brazil

***Abstract*—A new mathematical programming formulation is proposed for an optimization problem in queueing networks. The sum of the blocking probabilities of a general service time, single server, finite, acyclic queueing network is minimized, as are the total buffer sizes and the overall service rates. A multi-objective genetic algorithm (MOGA) and a particle swarm optimization (MOPSO) algorithm are combined to solve this difficult stochastic problem. The derived algorithm produces a set of efficient solutions for multiple objectives in the objective function. The implementation of the optimization algorithms is dependent on the generalized expansion method (GEM), a classical tool used to evaluate the performance of finite queueing networks. A set of computational experiments is presented to attest to the efficacy and efficiency of the proposed approach. Insights obtained from the analysis of a complex network may assist in the planning of these types of queueing networks.**

***Index Terms*—Buffer allocation, queueing networks, conflicting objectives, particle swarm optimization.**

I. Introduction

Almost everyone has had the unpleasant experience of spending too much time in queues. This phenomenon occurs in traffic jams, retail checkouts, bank service queues, and in many other situations. Many of these processes can be modeled as queueing systems. In practice, queues occur because the demand for service becomes greater than the ability of the queueing system to meet such a demand. A simplistic solution would be to increase the service capacity to the maximum, but budget and space restrictions usually mean this is not a feasible choice.

A queueing system can be described as customers arriving and waiting for service, then leaving the system after their demand has been met. In other words, queueing systems are present in situations of uncertainty about the flow of products, users, or other items. For example, it is possible to model a process waiting to be processed on a CPU as a queue [1]–[3].

Queues configured in networks, where each queue has an arrival rate λ and a service rate μ, are a natural generalization for various systems of practical interest.

Situations with limited waiting areas (finite buffers) for a given service, result in finite queues. In the case of finite queues with total space for K customers, P_K denotes the probability of finding K customers in the system, including those being served. That is, P_K is the blocking probability. Once a customer arrives in search of the service, and all the servers and the waiting positions are occupied, the customer is blocked by the system. For obvious reasons, high blocking probabilities imply inefficiency of the queueing system [4].

The novelty of this article lies mainly in the new formulation for an optimization problem in finite queueing networks and an effective heuristic method to simultaneously minimize the sum of the blocking probabilities ($\sum_i P_{K_i}$) in acyclic networks of m $M/G/1/K$ queues, that is, in Kendall notation, acyclic networks of Markovian arrivals, general service times, single-serve, finite queues, and the maximum of K customers, including those in service. To obtain the minimum $\sum_i P_{K_i}$, the minimum total capacity $\sum_i K_i$, and the minimum overall service rates $\sum_i \mu_i$ that must be allocated to a queueing network, in a given topology and under a known arrival rate λ_i, the procedure for the minimization process searches for the optimal coordinates of the vectors $\mathbf{P_K} = (P_{K_1}, P_{K_2}, \ldots, P_{K_m})$, $\mathbf{K} = (K_1, K_2, \ldots, K_m)$, and $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_m)$, which determine the optimal configuration for the queueing network.

Optimization in finite-queue network systems concerns many aspects of real-life, with the possibility of helping to understand and improve various systems present in people's daily lives, including industrial processes, health systems, urban traffic, communication systems, and others [5]–[10].

This article focuses on networks of $M/G/1/K$ queues. In particular, the study will address a complex network of queues involving series, splits, and mergers between queues. Entry into the system will be unique, through the first queue of the system, and will continue through the system with a pre-set routing probability vector for split situations. The network configuration under investigation can be seen in Fig. 1.

There is a critical trade-off between buffer allocation, service rates, and blocking probabilities, in each queue of the system. It is reasonable to note that the greater the buffer allocation and the service rate are in the system, the less the blocking probability is for each queue. On the contrary, buffer allocation and service rates are highly costly. The major objective is then to minimize such resources but still to be able to obtain a situation capable of substantially minimizing the blocking probabilities between the queues in the system.

An optimization approach is proposed for the search of a Pareto-optimal solutions range. The method produces a set of efficient solutions for more than one objective in the objective

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

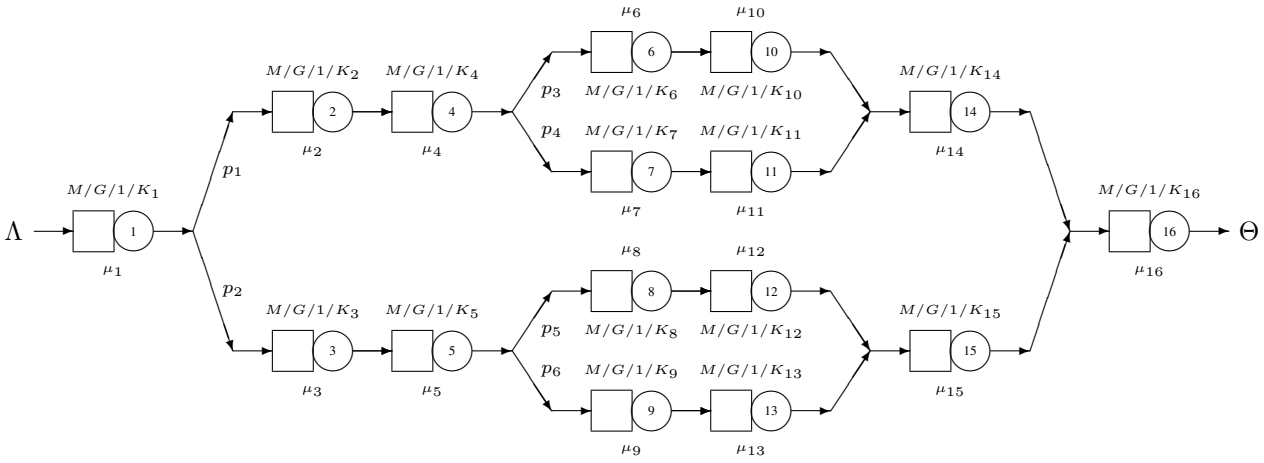


Fig. 1. A mixed-topology network (adapted from MacGregor Smith and Cruz [11])

function. With the approach proposed, the decision-maker can evaluate the effect of each solution. The multi-objective approach also allows the user to increase one objective (e.g., increase buffer allocation) while reducing another objective (e.g., reduce blocking probabilities). A multi-objective genetic algorithm (MOGA) combined with a particle swarm optimization (MOPSO) algorithm was used. Nature-inspired heuristic algorithms are particularly suitable for difficult mono-objective problems, but their multi-objective counterpart has also been shown to be effective in a variety of settings with particularly difficult objective functions and constraints [12]–[17], with great computational performance.

The rest of this article consists of four sections. Section II discusses several mathematical programming formulations for the optimization of queueing networks. A new mathematical programming formulation for optimization of queueing networks and, multi-objective genetic and particle swarm optimization algorithms, specifically developed for multi-objective optimization, are presented in Section III. In Section IV, the results of a comprehensive set of computational experiments are presented to attest to the efficacy and efficiency of the approach. Finally, the article is concluded in Section V, with final remarks and propositions for future research in the area.

II. Background

A. Single-objective Formulations

The objective here is the development of algorithms to optimize (minimize) the sum of the blocking probabilities ($\sum_i P_{K_i}$) of an acyclic network of $M/G/1/K$ queues, while simultaneously optimizing (maximizing) throughput (Θ).

The algorithm is very dependent on the mathematical programming formulation. We begin by describing the formulation of the buffer allocation problem (BAP).

The problem is defined on a digraph $\mathcal{D}(V, A)$ where V is a finite set of m vertexes (queues), and A is a finite set of edges (connections between the queues). The BAP, in its primal formulation [11], is as follows:

$$\text{minimize} \sum_{i=1}^{m} c_i K_i, \tag{1}$$

subject to:

$$\begin{aligned} &\Theta(\mathbf{K}) \geqslant \Theta_{\min}, \\ &K_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\}, \end{aligned} \tag{2}$$

which minimizes the total capacity allocation cost for the network with m queues, subject to a threshold $\Theta_{\min}$ and integer total capacities K_i.

Although quite similar to a linear integer mathematical programming problem, $\Theta(\mathbf{K})$ is difficult to define because it is a function that involves arrival and service rates, the topology of the queueing network, as well as the integer decision variables, K_i.

There is another closely related formulation, a type of dual BAP, which seeks to maximize the throughput, $\Theta(\mathbf{K})$, while constrained to a maximum budget for the total capacity allocation along the network, $\Omega_{\max}$. The dual formulation is a type of stochastic knapsack problem [11], which may be written as follows:

$$\text{maximize } \Theta(\mathbf{K}), \tag{3}$$

subject to:

$$\begin{aligned} &\sum_{i=1}^{m} c_i K_i \leqslant \Omega_{\max}, \\ &K_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\}, \end{aligned} \tag{4}$$

which maximizes the throughput, $\Theta(\mathbf{K})$, subject to a maximum budget, $\Omega_{\max}$, for the capacity allocation along the network with integer capacities, K_i.

Although the BAP formulations just presented could be used as an aid to develop efficient algorithms to solve queueing network design problems, this article considers an algorithm that is based on the following multi-objective formulations.

B. Multi-objective Formulation

The optimization problem of $M/G/1/K$ networks, described in its primal and dual formulations, can be reformulated into a multi-objective mathematical programming formulation, which comprises the minimization of capacities and service rates, simultaneously with maximization of throughput. The multi-objective queueing network problem may be formulated as follows (see [18], [19]):

$$\text{minimize } F(\mathbf{K}, \boldsymbol{\mu}) = \Big[f_1(\mathbf{K}), f_2(\boldsymbol{\mu}), f_3(\mathbf{K}, \boldsymbol{\mu})\Big], \quad (5)$$

subject to:

$$\begin{aligned} &K_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\}, \\ &\mu_i \geqslant 0, \forall i \in \{1, 2, \ldots, m\}, \end{aligned} \quad (6)$$

in which $f_1(\mathbf{K}) = \sum_{i=1}^{m} K_i$ represents the total capacities, $f_2(\boldsymbol{\mu}) = \sum_{i=1}^{m} \mu_i$ represents the overall service rates, and $f_3(\mathbf{K}, \boldsymbol{\mu}) = -\Theta(\mathbf{K}, \boldsymbol{\mu})$ represents the throughput. Note the minus sign associated with throughput, as it is an objective to be maximized.

Usually, in the literature, the throughput is modeled as a constraint. One drawback of this approach is that the throughput restriction must be relaxed. However, finding a suitable threshold is not a trivial task.

C. Multi-Objective Optimization

Multi-objective optimization addresses the problem of searching for optimal solutions, when multiple, competing objective functions are interacting. A multi-objective optimization problem comprises a pair of objects $\mathcal{X}$ and F, defined as:

$$\begin{aligned} &\text{minimize } F(\mathbf{x}) = \Big[f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_\ell(\mathbf{x})\Big], \\ &\text{subject to: } \mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathcal{X}, \end{aligned} \quad (7)$$

wherein $\mathbf{x} \in \mathcal{X}$ is the *decision variable*, $\mathcal{X}$ is the *feasible solutions set*, and $\mathcal{F} = F(\mathcal{X})$ is the *objective space*, with ℓ objective functions. The goal in multi-objective optimization is to find the solutions that "minimize" $F(\mathbf{x})$ in the Pareto-optimality context [20].

Given $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ and the relation $\prec$, defined as

$$F(\mathbf{x}) \prec F(\mathbf{x}') \iff F(\mathbf{x}) \leq F(\mathbf{x}') \text{ and } F(\mathbf{x}) \neq F(\mathbf{x}'),$$

wherein $F(\mathbf{x}) \leq F(\mathbf{x}')$ if and only if $f_i(\mathbf{x}) \leq f_i(\mathbf{x}')$, for all $i = 1, \ldots, \ell$, and $F(\mathbf{x}) \neq F(\mathbf{x}')$ if and only if $\exists i \in \{1, \ldots, \ell\}$, such that $f_i(\mathbf{x}) \neq f_i(\mathbf{x}')$, a feasible solution $\mathbf{x}^* \in \mathcal{X}$ is a *Pareto optimal solution* of the multi-objective optimization problem given by Eq. (7) if there is no $\mathbf{x} \in \mathcal{X}$ such that $F(\mathbf{x}) \prec F(\mathbf{x}^*)$. The range of $\mathbf{x}^*$, $F(\mathbf{x}^*)$, is called a *non-dominated point*. The set of all Pareto-optimal solutions is called the *Pareto-optimal set*, and the set of all non-dominated points is called the *Pareto front*. It is said that solution $\mathbf{x}$ dominates solution $\mathbf{x}'$ and/or that $F(\mathbf{x})$ dominates $F(\mathbf{x}')$ if $F(\mathbf{x}) \prec F(\mathbf{x}')$.

III. A Novel Multi-objective Formulation

A. Blocking Probabilities Optimization

The literature presents several possible formulations for the optimization problem based on the throughput of the system, (Θ) [4], [18], [19], [21]–[28].

The new mathematical formulation proposed here for optimization focuses on the blocking probabilities of the queueing system. This investigation prioritizes minimizing the sum of blocking probabilities in the system while minimizing the total capacity allocation and the overall service allocation. The rationale behind this choice is a more intuitive nature of blocking probability. That is, when looking at the overall throughput, there is no clear idea of the interrelation between the queues in the network. Prioritizing blocking probability values ensures a greater degree of decoupling between the different queues of the network. With a low blocking probability, the lines suffer less interlocking, which occurs when the downstream flow blocks the upstream flow.

Given that the decision variables K_i and μ_i indicate the total capacity of the system and service rate for the ith $M/G/1/K$ queue, respectively, the optimization problem under study can be formulated as:

$$\text{minimize } F(\mathbf{K}, \boldsymbol{\mu}) = \Big[f_1(\mathbf{K}), f_2(\boldsymbol{\mu}), f_3(\mathbf{K}, \boldsymbol{\mu})\Big], \quad (8)$$

subject to:

$$\begin{aligned} &K_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\}, \\ &\mu_i \geqslant 0, \forall i \in \{1, 2, \ldots, m\}, \end{aligned} \quad (9)$$

in which $f_1(\mathbf{K}) = \sum_{i=1}^{m} K_i$ represents the total capacity allocated; $f_2(\boldsymbol{\mu}) = \sum_{i=1}^{m} \mu_i$ represents the service rates, and $f_3(\mathbf{K}, \boldsymbol{\mu}) \sum_{i=1}^{m} P_{K_i}$ represents the sum of blocking probabilities.

B. Performance Evaluation

In *single $M/G/1/K$ queues*, the estimate of the blocking probability P_K can be obtained through a computationally efficient and accurate closed-form. The method, proposed by MacGregor Smith [29], is based on a two-moment approximation of Kimura [30]:

$$P_K = \frac{\rho^{\left(\frac{2+\sqrt{\rho}s^2-\sqrt{\rho}+2(K-1)}{2+\sqrt{\rho}s^2-\sqrt{\rho}}\right)}(\rho - 1)}{\rho^{\left(2\frac{2+\sqrt{\rho}s^2-\sqrt{\rho}+(K-1)}{2+\sqrt{\rho}s^2-\sqrt{\rho}}\right)} - 1}, \quad (10)$$

in which $\rho < 1$ must hold, where ρ is the system utilization, defined as the ratio between the total arrival rate and the service rate, $\rho = \lambda/\mu$, and, $s^2 = \text{Var}(T_s)/\,\mathbb{E}^2(T_s)$ is the squared coefficient of variation of the service time (T_s). Several previous studies confirm that the approximation of P_K is accurate for a wide range of values [4], [11], [31].

For *single queues*, a fraction P_K of the arrivals cannot join the system. Thus, P_K represents the probability that a customer arrives when there is no more waiting space. Therefore, only the fraction $(1 - P_K)$ of the arrivals can be served by the queue [32], resulting in a throughput of $\lambda(1 - P_K)$. That is, the fraction of customers, arriving at a rate of λ, who did not find the system blocked, will be the throughput of this single queue. The throughput is approximately Markovian, that is, the inter-arrival times approximately follow an exponential distribution (see [11]).

Investigations of queueing network problems are addressed from many perspectives [27], [33]–[35]. Approaches using optimization methods are quite common, for example, Powell's method [28], Genetic algorithms [18], and Simulated Annealing [19] have been used. These approaches use the throughput (Θ), which is generally obtained by using an approximate performance evaluation method, namely the generalized expansion method (GEM) [36].

GEM is an algorithm that has been successfully used to estimate the performance of arbitrarily configured, finite queueing, acyclic networks that updates system performance measures over repeated trials. The method considers the delay effect generated by several possible blockages occurring in the flow of customers along the queueing network. GEM solves a set of simultaneous nonlinear equations through iterative procedures. This leads to considerable improvement in the precision of the estimation of the performance measures of the queueing network. 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.

As described in detail by Kerbache and MacGregor Smith [36], GEM creates, for each finite node j, an auxiliary vertex (h_j) that is modeled as an $M/G/\infty$ queue, as shown in Fig. 2. For each entity placed in 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 that an entity has to wait, with a service rate μ'_h, given by GEM, before entering vertex j, and updates accordingly the effective arrival rate coming from vertex i to vertex j, $\lambda_{\text{eff}} = \lambda_i(1 - P_{K_i})$.

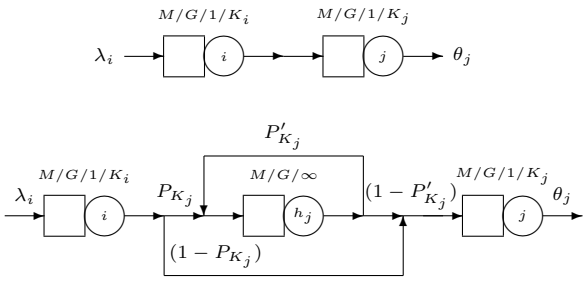


Fig. 2. Generalized expansion method for a tandem network

The ultimate goal of GEM is to provide updates of the service rates of the nodes as follows:

$$\tilde{\mu}_i^{-1} = \mu_i^{-1} + P_{K_j}(\mu'_h)^{-1}, \tag{11}$$

in such a way that from Eq. (11) services rates μ_i can be updated for all nodes along ρ and consequently P_K, from Eq. (10).

The GEM iteratively calculates updates of the system performance measures. The method takes into account the delay effect generated by several possible blockages occurring in the flow of customers along the queues. This paper does not consider the throughput (Θ) as its optimization objective. Instead, the sum of the blocking probabilities in each queue is considered. It is important to note that the computation of P_K, even though the approximation proposed in Eq. (10), is dependent on the knowledge of the queue arrival rate, λ. For the front queue of a network (see Fig. 1), the arrival rate, λ, is known. However, what are the arrival rates for subsequent queues? The procedure applied to obtain these arrival rates in many studies also takes into account approximations produced through the use of GEM.

Note that P_{K_i} is calculated by GEM [29] and is dependent on λ_i, μ_i, and K_i. The μ_i and K_i values are decision variables of the optimization problem, but arrival rate λ_i is dependent on the throughput rate of the previous queue. Without loss of generality, in a tandem queueing network, the computations performed in this study always assume that the arrival rate on the ith queue is dependent on the previous $(i-1)$th queue, given by:

$$\lambda_i = \lambda_{i-1}(1 - P_{K_{i-1}}), \tag{12}$$

in which $i \in \{2, \ldots, m\}$ and λ_1 is considered to be the external queueing network arrival, that is, $\lambda_1 = \Lambda$.

C. Multi-objective Approach

A multi-objective evolutionary algorithm (MOEA) is adapted for the optimization problem given by Eq. (8) and (9). An MOEA is an optimization algorithm that approximately performs global searches based on the information that is obtained from the evaluation of several points in the search space [37]. The population converges to a mutually non-dominated approximation set of the Pareto front by the application of the genetic operators of *mutation*, *crossover*, *selection*, and *elitism*. The MOEA used here is the elitist non-dominated sorting genetic (NSGA-II) algorithm, which is state of the art for multi-objective optimization of finite queueing networks [19]. Details will not be given here due to space considerations, but can be found easily (e.g., see [19]).

Following NSGA-II optimization, a multi-objective particle swarm optimization (MOPSO) algorithm is applied to improve the solutions provided by the NSGA-II algorithm. Given the newly introduced mathematical programming formulation, the convergence of NSGA-II might be greatly improved by a post-processing algorithm such as a MOPSO. The proposed MOPSO extends the single-objective PSO algorithm from Kennedy & Eberhart [38].

Each particle should represent a possible solution for the resource allocation (capacities and service rates) that optimize the finite-queueing network under study. Hence, in this particular formulation, each particle can be represented by variables $(x_1, \ldots, x_\ell) = (K_1, K_2, \ldots, K_m, \mu_1, \mu_2, \ldots, \mu_m)$, with $\ell = 2m$.

It is important to highlight here that the multi-objective optimization problem being addressed is a mixed-integer problem. Thus, a particle repair strategy must be defined. Indeed, changes to capacities are performed and then integer values are used, as $K_i \geqslant 1$ is always respected. Similarly, the restrictions associated with service rates are also respected, because it is necessary to guarantee that $\rho < 1$. That is, the queue arrival rate must be strictly less than the service rate μ. These considerations guarantee the feasibility of the investigated solutions.

If s denotes the size of the swarm (population of particles), each particle $1 \leq i \leq s$ has the following attributes:

- Position, $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,\ell})$;
- Velocity, $v_i = (v_{i,1}, v_{i,2}, \ldots, v_{i,\ell})$;
- Personal best position, p_i;

- Global best position, g_i.

The proposed MOPSO approach for queue network optimization is to execute the following sequence of steps:

step 1: Initialize:
- The population of particles, x_i;
- The velocities of particles, v_i;
- The best position, $p_i = x_i$;
- Particle's global best positions, $g_i = x_i$;
- Iteration counter;

step 2: Store the non-dominated particles of x_i into external archive A;

step 3: Compute the crowding distance of particles stored in A and sort them in descending values;

step 4: Randomly select a solution from the non-dominated particles from A and store the position to g_i for each particle of the population;

step 5: Update the velocity and position of the particles according to Eq. (13) and (14), respectively:

$$v_i^{t+1} = w^t + r_1\left(p_i - x_i^t\right) + r_2\left(g_i - x_i^t\right), \quad (13)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}; \quad (14)$$

For the integer variables, the position must be updated accordingly, Eq. (15), that is:

$$x_i^{t+1} = \text{int}\left(x_i^t + v_i^{t+1}\right); \quad (15)$$

step 6: Update the particle's best position, p_i; if the current position dominates p_i in memory, then reset p_i to current position.

step 7: Go to step 2 until a criterion is satisfied.

The parameters and their values were defined as follows: r_1 and r_2 are positive random numbers with uniform distribution belonging to the interval $[0, 1.0]$, $w = 0.4$ is the inertia weight.

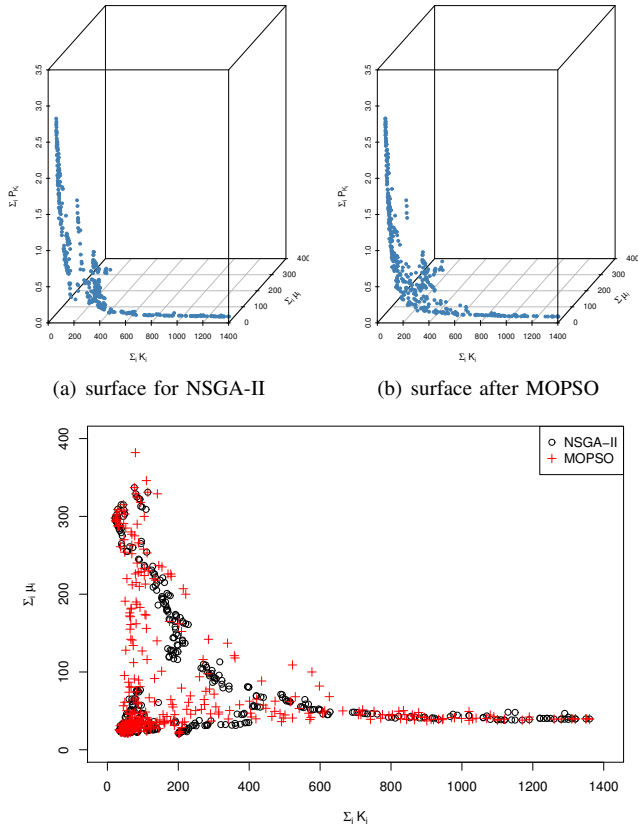
Further details about the implementation of MOPSO algorithms can be found in the literature. The MOPSO just described is an adaptation of the classical implementation described by Coello-Coello and Lechunga [39]. However simplified versions may be found [40] and more sophisticated and improved versions [41], [42] as well, including mixed-integer mathematical programming formulations [43].

IV. Computational Results

The optimization algorithms were implemented in FORTRAN. The NSGA-II code used here was provided by Cruz et al. [19]. For educational and research purposes, the code is available from the authors upon request. The execution environment of the computational experiments was conducted on Intel(R) Core(TM) i3-2310M 2.10 GHz running Windows 10 Pro 64 bits, with 6.00GB of RAM.

The mixed-topology network presented in Fig. 1 was adapted from the literature [11] and analyzed with the proposed method. Three different squared coefficients of variation were analyzed, $s^2 = \{0.5, 1.0, 1.5\}$, to characterize systems that are hypo-exponential, exponential (Markovian), and hyper-exponential, respectively. The external arrival rate was $\Lambda = 5.0$. For comparison purposes, the above experiments match those previously performed by Cruz et al. [19].

The number of particles in the GA and swarm was defined as 400, and the maximum number of interactions of the algorithm was defined as 4,000. To examine the solutions in greater detail, Fig. 3, 4 and 5 present the results obtained for all squared coefficients of variation tested. In each of these figures, sub-optimal solutions are presented for, (a) a three-dimensional space, provided by the NSGA-II, (b) by MOPSO algorithm, and (c) the points projected into the two-dimensional space $\sum_i K_i \times \sum_i \mu_i$. This projection leaves a false impression of the existence of dominated points but, in fact, they are all non-dominated points in the three-dimensional space. In the graph representing the projection, the points obtained by NSGA-II are shown, as are the points obtained by MOPSO post-processing.



(a) surface for NSGA-II (b) surface after MOPSO

(c) two-dimensional projection for solutions

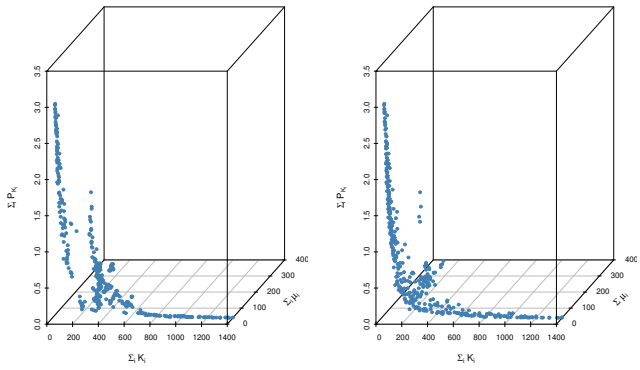
Fig. 3. Solutions for $s^2 = 0.5$.

From the results for the hypo-exponential system ($s^2 = 0.5$) shown in Fig. 3, including the final surface in two-dimensional space $\sum_i K_i \times \sum_i \mu_i$, after convergence it is possible to see that the behavior of a given queueing network cannot be predicted without the use of an algorithmic approach such as the one proposed here. A detailed analysis of the results in Fig. 3, revealed that many different pairs of capacities and service rates can be selected for a given sum of blocking probabilities. It is noteworthy that the MOPSO better distributes the points and improves the representation of the Pareto surface using its

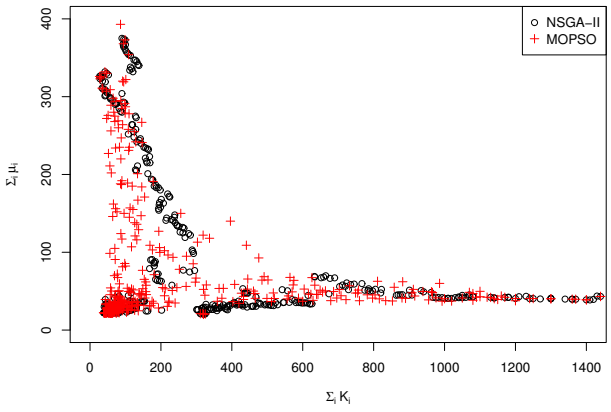
discrete points so that more diverse solutions are available for the analysis to choose from.

The two-dimensional space $\sum_i K_i \times \sum_i \mu_i$ projection, shown in Fig. 3-(c), resulting from the NSGA-II and MOPSO approach, reveals several solutions with a low overall service allocation, but which are still efficient to solve the problem under investigation. Even with low capacity allocation, the algorithm can produce promising solutions. It is possible to observe a large number of solutions with the capacity allocation between approximately 50 and 150, while the overall service allocation is less than 250, better than some solutions previously provided the NSGA-II. Fig. 3-(c) illustrates that these solutions have an acceptable sum of blocking probabilities for solving the problem. This analysis confirms that, for hypo-exponential systems, the proposed approach is capable of delivering many efficient solutions.

Results for the exponential system ($s^2 = 1.0$) are shown in Fig. 4. Again, the analysis shows that several different pairs of service rates and capacities may be chosen for a given sum of blocking probabilities. The MOPSO post-processing produced solutions that are more broadly spread in the region compared to the solutions previously produced by the NSGA-II alone.



(a) surface for NSGA-II (b) surface after MOPSO



(c) two-dimensional projection for solutions

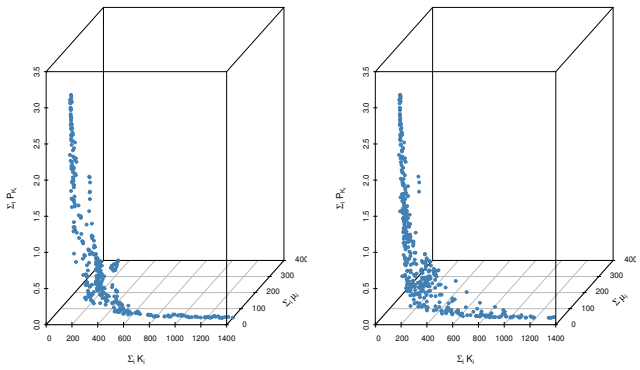
Fig. 4. Solutions for $s^2 = 1.0$.

The two-dimensional space $\sum_i K_i \times \sum_i \mu_i$ projection, Fig. 4-(c), from NSGA-II and MOPSO, again shows many similar points for exponential systems. The MOPSO approach was able to better represent the Pareto surface by utilizing more representative discrete points that were more broadly distributed.

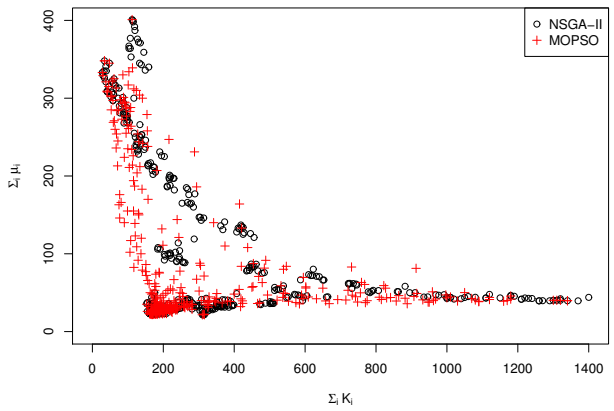
It is noticeable that MOPSO uses solutions provided by the NSGA-II with an overall service allocation above 200, and replaces them with other solutions when there is a lower overall service allocation, even while preserving the capacity allocation of previous solutions. Such solutions are less costly compared to the previous, while still providing effective solutions regarding blocking probabilities.

Finally, the results of hyper-exponential systems ($s^2 = 1.5$) are highlighted in Fig. 5, including the final surface and the swarm in two-dimensional space $\sum_i K_i \times \sum_i \mu_i$ after final processing. Once again, the analysis of the results in Fig. 5 shows that several different pairs of service rates and capacities may be achieved for a given sum of blocking probabilities. This finding is very important in multi-objective approaches.

The two-dimensional projection of the space $\sum_i K_i \times \sum_i \mu_i$, Fig. 5-(c), by NSGA-II and MOPSO, again shows a good resemblance between the two sets of points. Similar to the hypo-exponential and exponential systems, the MOPSO approach is capable of achieving many of the solutions previously obtained. Post-processing with MOPSO also identifies new solutions with capacity allocation between 70 and 150, but with overall service allocation notably lower than the previous solutions provided by NSGA-II.



(a) surface for NSGA-II (b) surface after MOPSO



(c) two-dimensional projection for solutions

Fig. 5. Solutions for $s^2 = 1.5$.

Two specific factors affect the computational efficiency of the algorithm in obtaining solutions for the queueing network optimization problem in hand. First, the choice of MOPSO for post-processing after NSGA-II and, second, the new mathematical programming formulation that prioritizes the blocking probability, a performance measure that makes more sense given the behavior of the system being modeled, because it shows the proportion of items that are not served by the respective queue. For the comparisons to be adequate, the same population size in the NSGA-II was also used as the particle number in the swarm for the MOPSO algorithm. Further, the number of generations chosen for the MOPSO algorithm was identical to the number of interactions used with the NSGA-II algorithm.

Considering the three different squared coefficients of variation analyzed $s^2 = \{0.5, 1.0, 1.5\}$, the CPU times of the NSGA-II were 527.0, 529.4 and 539.8 seconds, respectively. The CPU times of the MOPSO algorithm were 478.1 sec, 475.5 and 495.95 seconds, respectively. This result confirms that the new approach provides efficient solutions in a reasonable amount of time.

V. Conclusion

In this article, a novel mathematical programming formulation was proposed for an optimization problem in queueing networks. The sum of the blocking probabilities of a general-service time, single-server, finite, acyclic queueing network was minimized along with the total capacity and the overall service rate.

A combination of multi-objective genetic and particle swarm optimization algorithms were developed and employed. Insightful Pareto curves were obtained. In this new approach, the use of a classical tool to approximately evaluate the performance of finite queueing networks, namely GEM, was used and very efficient solutions were obtained.

Concerning CPU time, the new approach is comparable to the previous approach. Thus, future experiments can be performed to evaluate this approach's flexibility regarding the number of interactions and swarm sizes, its use under different queueing networks, with different topologies (e.g., series, merges, and splits), and with various numbers of nodes, arrival rates, and service time variability.

The modifications in the mathematical programming formulation of this stochastic optimization problem in queueing networks, and the change in the optimization heuristic applied, brought improvements to the area. The combination of NSGA-II and MOPSO was successful. The novel mathematical programming formulation also made it possible to produce an optimization approach that performed well under GEM updates.

Future investigations should be executed to determine the applicability of this approach for the determination of other optimal conditions in queueing networks. For instance, this method could be applied to optimize general, multi-server finite queueing networks. Moreover, future research should be conducted to evaluate the algorithms in real-life situations.

Acknowledgment

This research is partially supported by the Brazilian Agencies, FAPEMIG (Fundação de Amparo à Pesquisa do Estado de Minas Gerais, grants APQ-00379-17 and CEX-PPM-00564-17), CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico, grant 305515/2018-7), and by UFOP (Universidade Federal de Ouro Preto).

References

[1] 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, no. Article ID 54683, p. 17, 2007.

[2] J. Chen, C. Hu, and Z. Ji, "An improved ARED algorithm for congestion control of network transmission," *Mathematical Problems in Engineering*, vol. 2010, no. Article ID 329035, p. 17, 2010.

[3] V. Inzillo, F. De Rango, and A. A. Quintana, "A self clocked fair queuing MAC approach limiting deafness and round robin issues in directional MANET," in *2019 Wireless Days (WD)*. IEEE, 2019, pp. 1–6.

[4] F. R. B. Cruz, A. R. Duarte, and T. van Woensel, "Buffer allocation in general single-server queueing networks," *Computers & Operations Research*, vol. 35, no. 11, pp. 3581 –3598, 2008.

[5] 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.

[6] F. S. Q. Alves, H. C. Yehia, L. A. C. Pedrosa, F. R. B. Cruz, and L. Kerbache, "Upper bounds on performance measures of heterogeneous $M/M/c$ queues," *Mathematical Problems in Engineering*, vol. 2011, no. Article ID 702834, p. 18, 2011.

[7] J. MacGregor 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.

[8] 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.

[9] 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.

[10] D. A. Menasc, "QoS issues in web services," *IEEE Internet Computing*, vol. 6, no. 6, pp. 72–75, 2002.

[11] J. MacGregor 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.

[12] W. Leong and G. G. Yen, "PSO-based multiobjective optimization with dynamic population size and adaptive local archives," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 5, pp. 1270–1293, 2008.

[13] V. Hajipour and S. H. R. Pasandideh, "Proposing an adaptive particle swarm optimization for a novel bi-objective queuing facility location model," *Economic Computation and Economic Cybernetics Studies and Research*, vol. 46, no. 3, pp. 223–240, 2012.

[14] M. Sharafi and T. Y. ELMekkawy, "Multi-objective optimal design of hybrid renewable energy systems using PSO-simulation based approach," *Renewable Energy*, vol. 68, pp. 67–79, 2014.

[15] W. Deng, H. Zhao, X. Yang, J. Xiong, M. Sun, and B. Li, "Study on an improved adaptive PSO algorithm for solving multi-objective gate assignment," *Applied Soft Computing*, vol. 59, pp. 288–302, 2017.

[16] P. Azimi and A. Asadollahi, "Developing a new bi-objective functions model for a hierarchical location-allocation problem using the queuing theory and mathematical programming," *Journal of Optimization in Industrial Engineering*, vol. 12, no. 2, pp. 149–154, 2019.

[17] D. R. X. Oliveira, G. J. P. Moreira, A. R. Duarte, A. L. F. Cançado, and E. Luz, "Spatial cluster analysis using particle swarm optimization and dispersion function," *Communications in Statistics - Simulation and Computation*, pp. 1–18, 2019.

[18] F. R. B. Cruz, G. Kendall, L. While, A. R. Duarte, and N. C. L. Brito, "Throughput maximization of queueing networks with simultaneous minimization of service rates and buffers," *Mathematical Problems in Engineering*, vol. 2012, no. Article ID 692593, p. 19, 2012.

[19] F. R. B. Cruz, A. R. Duarte, and G. L. Souza, "Multi-objective performance improvements of general finite single-server queueing networks," *Journal of Heuristics*, vol. 24, no. 5, pp. 757–781, 2018.

[20] G. Moreira and L. Paquete, "Guiding under uniformity measure in the decision space," in *Proceedings of the 2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, 2019, pp. 1–6.

[21] L. Kerbache and J. MacGregor 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.

[22] 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.

[23] T. van Woensel, R. Andriansyah, F. R. B. Cruz, M. J, and L. Kerbache, "Allocation in general multi-server queueing networks," *International Transactions in Operational Research*, vol. 17, no. 2, pp. 257–286, 2010.

[24] ——, "Buffer and server allocation in general multi-server queueing networks," *International Transactions in Operational Research*, vol. 17, no. 2, pp. 257–286, 2010.

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

[26] F. R. B. Cruz, T. van Woensel, and J. MacGregor 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.

[27] T. van Woensel and F. R. B. Cruz, "Optimal routing in general finite multi-server queueing networks," *PLoS ONE*, vol. 9, no. 7, p. e102075, 07 2014. [Online]. Available: http://dx.doi.org/10.1371%2Fjournal.pone.0102075

[28] H. S. R. Martins, F. R. B. Cruz, A. R. Duarte, and F. L. P. Oliveira, "Modeling and optimization of buffers and servers in finite queueing networks," *OPSEARCH*, vol. 56, no. 1, pp. 123–150, 2019.

[29] J. MacGregor 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.

[30] T. Kimura, "A transform-free approximation for the finite capacity $M/G/s$ queue," *Operations Research*, vol. 44, no. 6, pp. 984–988, 1996.

[31] J. MacGregor Smith, "$M/G/c/K$ blocking probability models and system performance," *Performance Evaluation*, vol. 52, no. 4, pp. 237–267, 2003.

[32] D. Gross, J. F. Shortle, T. J. M, and H. C. M, *Fundamentals of Queueing Theory*, 4th ed. New York, NY: Wiley - Interscience, 2009.

[33] S. Chowdhury and S. P. Mukherjee, "Estimation of traffic intensity based on queue length in a single $M/M/1$ queue," *Communications in Statistics - Theory and Methods*, vol. 42, no. 13, pp. 2376–2390, 2013.

[34] D. Qi, Z. Li, X. Zi, and Z. Wang, "Weighted likelihood ratio chart for statistical monitoring of queueing systems," *Quality Technology & Quantitative Management*, vol. 14, no. 1, pp. 19–30, 2017.

[35] F. R. B. Cruz, R. C. Quinino, and L. L. Ho, "Control charts for traffic intensity monitoring of Markovian multiserver queues," *Quality and Reliability Engineering International*, vol. 36, no. 1, pp. 354–364, 2020.

[36] L. Kerbache and J. MacGregor Smith, "The generalized expansion method for open finite queueing networks," *European Journal of Operational Research*, vol. 32, pp. 448–461, 1987.

[37] D. Kalyanmoy, *Multi-objective Optimisation using Evolutionary Algorithms*. New York, NY: John Wiley & Sons, Inc., 2001.

[38] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, 1995, pp. 1942–1948.

[39] C. A. Coello Coello and M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, vol. 2, 2002, pp. 1051–1056.

[40] V. Trivedi, P. Varshney, and M. Ramteke, "A simplified multi-objective particle swarm optimization algorithm," *Swarm Intelligence*, vol. (in press), pp. 1–34, 2019.

[41] Z. Fan, T. Wang, Z. Cheng, G. Li, and F. Gu, "An improved multiobjective particle swarm optimization algorithm using minimum distance of point to line," *Shock and Vibration*, vol. 2017, pp. 1–16, 2017.

[42] C. Jia and H. Zhu, "An improved multiobjective particle swarm optimization based on culture algorithms," *Algorithms*, vol. 10, no. 2, p. 46, 2017.

[43] X. Zhao, Y. Jin, H. Ji, J. Geng, X. Liang, and R. Jin, "An improved mixed-integer multi-objective particle swarm optimization and its application in antenna array design," in *2013 5th IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*, 2013, pp. 412–415.