ISSN: 2317-0840

Optimal Buffer and Server Allocation in Markovian Queueing Networks by Particle Swarm Algorithm

Marina Campos Oliveira[1] , Josino José Barbosa[1] , Helgem de Souza Ribeiro Martins[1] , Gabriel Lima de Souza[1] , Anderson Ribeiro Duarte[1†].

[1]*Universidade Federal de Ouro Preto, Instituto de Ciências Exatas e Biológicas, Departamento de Estatística;Ouro Preto-MG, Brasil.*

Abstract: *This study optimizes resource allocation in queueing network systems to improve operational efficiency using appropriate algorithms. Specifically, Particle Swarm Optimization (PSO) was applied to the buffer and server allocation problem (BSAP) in queueing networks with Markovian arrivals and service times, multiple servers, and finite buffers. In this context, the buffer and server allocation problem (BSAP) stands out, whose solution methodology can be applied to various real-world situations modeled as queues or queueing networks, such as manufacturing line systems, healthcare services, traffic models, among others. BSAP is computationally challenging as a nonlinear programming problem without a closed-form analytical solution, necessitating derivative-free methods like PSO. The PSO algorithm is considered a promising tool for finding efficient solutions, thus enabling improved resource management. The study examines the algorithm's ability to deliver cost-effective and suitable solutions that accommodate variations in relative costs between servers and buffers, as well as the specific characteristics of each network topology.*

Keywords: *Queueing Theory; Servers; Buffers; Heuristic Optimization; Particle Swarm Optimization.*

Introduction

Optimization procedures are essential in our daily lives, helping us minimize costs and enhance productivity Duarte (2024). It involves converting practical problems into mathematical formulations to maximize or minimize specific functions that are key to achieving our objectives Carter (2018). In applications like queueing systems, optimization can dramatically increase efficiency by adjusting factors such as the number of servers used and the buffer allocation, thereby improving productivity (Gross et al., 2009).

Exact optimization methods seek precise and optimal solutions to specific problems. This search usually occurs through the use of derivatives to progress through a given, well-defined search space. These methods require that the objective function and constraints be continuous and differentiable; it is important to emphasize that these are overly restrictive requirements. Although they guarantee finding the globally optimal solution after a finite number of iterations, in complex or large-scale problems, computational effort can also be a problem Deb (2012). In contrast, heuristic methods do not guarantee the globally optimal solution, but they are practical for everyday decisions, providing fast and often satisfactory solutions with significantly lower computational cost Sharma (2022). This particular study addresses optimization problems solved by heuristic methods, which strike a confident balance between speed and accuracy in decision-making.

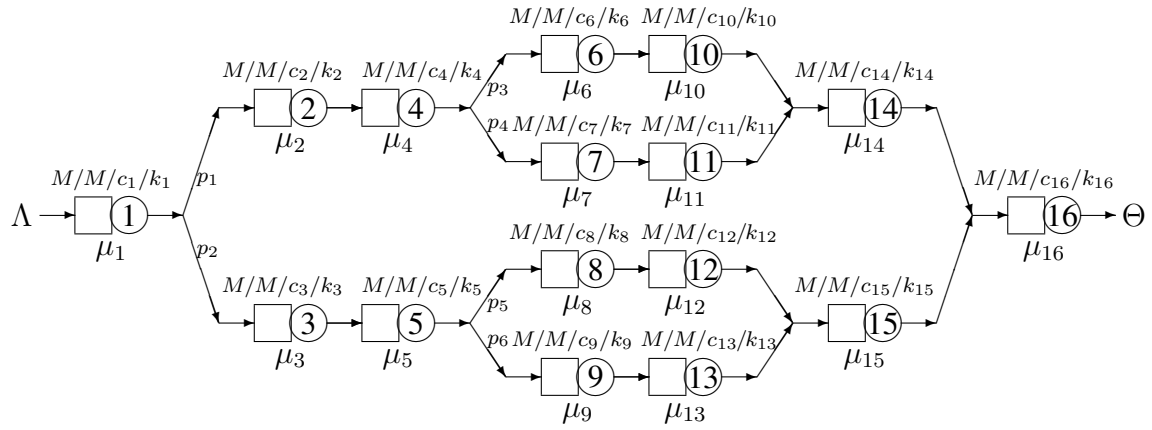
This study presents the optimization framework for queueing networks, where queues arise when service demand exceeds capacity, involving arrival patterns, service times, and waiting buffers. The goal is to optimize resource allocation in these systems to improve their operational efficiency through appropriate optimization algorithms.

[†]Autor correspondente: anderson.duarte@ufop.edu.br
Received: 18/08/2025. Revised: 08/10/2025. Accepted:10/10/2025.

We address here the specific purpose of using the Particle Swarm Optimization (PSO) heuristic to solve a specific optimization problem in queueing networks with Markovian arrivals and service times, multiservers, and finite buffers. The approach focuses on improving the operational efficiency of these queueing systems $M/M/c/k$, in Kendall (1953) notation. Here, $M/M/c/k$ represents the mathematical model to describe arrivals, service times, the number of available servers, and the maximum capacity of the queueing system, respectively. The PSO heuristic is selected as a promising tool for finding solutions close to the global optimum in this specific context, thus enabling improved resource management and reducing user waiting times.

We can illustrate the need to optimize the number of servers and buffers in queueing networks with the example of a medical emergency center. The users often face long waiting queues, resulting in frustration and dissatisfaction. This situation highlights the relevance of studying how to optimize the operation of these systems Zhong (2024). By reducing the number of servers or buffer areas, the service center can save resources, but this can result in even longer wait times. On the other hand, excessively increasing these resources can lead to avoidable costs Miserez (2024). Thus, the existence of a conflict between these two objectives becomes clear. Therefore, it is crucial to find the optimal balance between the number of servers and buffers to ensure operational efficiency and user satisfaction. The complex queueing network, shown in Figure 1, was adapted (Smith; Cruz, 2005).

Figure 1: Queueing network with mixed topology adapted from (Smith; Cruz, 2005).



Source: from the authors (2025).

Figure 1 represents a queueing network suitable for several study experiments. It models customer arrivals, routing to servers, and exit paths. This study aims to discuss the possibilities of mathematically formulating the problem of optimizing resource utilization for queues in acyclic networks, i.e. the Buffers and Servers Alocation Problem (BSAP). In queueing networks, buffers and servers play a crucial role in managing the system. Proper buffers and servers allocation is essential to avoid excessively long queues and long wait times, while excessive buffers and servers allocation can result in wasted resources. The aforementioned conflict becomes evident here. Several formulations have been proposed in the literature, usually problems with queueing networks describe the network using a graph $\mathcal{G}(V, A)$, in which V represents a finite set of m queues (vertices) and A represents a finite set of arcs (connections) between the queues.

This study aims to present a specific formulation for BSAP, fine-tune the classic PSO algorithm for the problem under investigation, and discuss the results obtained by the optimization algorithm for complex queueing networks. Modeling queueing networks has well-defined economic impacts on management in various sectors, illustrating the applied nature of this research. Furthermore, the problem is firmly situated within the realm of Queueing Theory and Stochastic Processes, ensuring a relevant and robust theoretical framework.

Methodological Tools

According to Yang (2010), optimization studies encompass a wide variety of applications and contexts. Any problem that involves searching for a specific level of optimality falls into the category of optimization problems.

Some specific concepts for approaching optimization problems with varied strategies were listed by Deb (2002), Collette (2004), and Ringuest (2012). The objective function is the central target of the optimization problem, representing the metric of interest to be improved or optimized. It is generally expressed as a mathematical formulation that relates the system variables. System variables directly influence the objective function. Identifying these variables is crucial to understanding their impact on system performance.

Linked to the system variables, the search space is defined as the set of all possible combinations of the system variables Sharma (2022). Potential constraints delimit the search space. These constraints are the problem specifications that limit the feasible values for the system variables and the objective function. These constraints can be related to server capacity, buffer capacity, or response time requirements, for example Cruz (2012). From a functional perspective in the mathematical conception, the system variables determining the elements of the domain of the objective function, thereby informing us about the system's structure. It is also important to understand the role of the elements within this function's range, specifically the optimization objectives (Deb, 2012).

The objective space is the set of all possible values of the objective function associated with the elements of the search space. In short, the objective space corresponds to the image of the objective function. Understanding this space contributes to the visualization of achievable and desirable targets. Among the elements belonging to the objective space are the local and global optimal solutions. A local optimal solution is the best solution within a specific region of the search space. On the other hand, a global optimal solution is the best possible solution in the entire search space, regardless of its location. Once the concept of optimization and the optimization problem is well established, a discussion about possible solution strategies for these problems is paramount (Deb, 2012).

Optimization in Queueing Networks

Abensur (2011) addressed Queue Networks in a study associated with the queues present in people's daily lives, such as healthcare queues, traffic congestion, or even virtual systems, such as call center waiting times. The reason queues form is directly related to the server's service capacity and the demand to be met. For almost obvious reasons, it is easy to understand that situations in which demand for a given service exceeds service capacity lead to queue formation.

According to Kendall (1953) notation, a queueing system can be composed of four basic components: the user arrival model, the service model, the number of servers, and capacity. This study focuses on solving optimization problems in queueing systems, aiming to improve

their operational efficiency. Optimization involves the process of determining the most efficient resource utilization configuration for the system, aiming to minimize or maximize specific performance metrics. These metrics can involve discrete or continuous variables, such as the number of servers, capacity, service rates, among others.

When solving optimization problems in queueing systems, it is essential to follow a systematic approach, which involves identifying the objective function, system variables, search space, and constraints involved. Furthermore, the selection and implementation of appropriate optimization algorithms play an important role in finding optimal solutions. By applying these concepts and techniques, it is possible to significantly improve the efficiency and performance of queueing systems. This study is particularly interested in applying the classic PSO algorithm.

PSO Introduction

Nature commonly reveals strategies based on the concepts of cooperation and sociability among individuals of the same species, as well as among different species within the same ecosystem. Species typically benefit from improved foraging strategies, increased chances of success in defense against predators, and many other advantages. Humans themselves seek to benefit from social interaction. It is easy to see that this coexistence strategy occurs intuitively among species due to the possibility of improving expressed objectives, i.e., it is an optimization strategy.

The term "swarm" clearly serves to define the concept of a collective, for example, a swarm of bees, an ant colony, among other possible metaphorical expressions. The PSO algorithm is a bioinspired optimization algorithm, that is, inspired by the strategies of living beings, originally proposed by Kennedy (1995). The algorithm aims to computationally replicate the movement of flocks, herds, or swarms of various animals as they search for food and migrate to warmer or colder regions. Leaders usually guide these movements, but they also involve a collaborative process executed by each of the other members of the swarm. The ultimate goal is to guide the entire swarm toward a better position for the overall well-being of the species.

The widespread PSO algorithm utilizes mathematical expressions to simulate swarm movements and cause a set of points to move in search of an optimal position. The concepts of cooperation, individuality, and sociability are fundamental to the algorithm's operation. Essentially, the memory of previously visited positions composes the set of information to guide the search for optimal solutions. Initially, we randomly positioned the particles within the feasible solution space. At each iterative step of the algorithm, we performed successive particle movements (according to the information contained in the particles themselves), guided by the objective function. The movement of each point is determined by calculating the magnitude and direction of each movement. This operation is treated as the velocity of a particle in this swarm (Jain et al., 2022).

This study requires clarification of three specific methodological points. Because it discusses specific applications in Queueing Theory, a review of queueing studies is essential. Furthermore, the objective is to address a queueing problem from an optimization perspective, which requires a discussion of the mathematical formulations associated with optimization problems in Queueing Theory. Finally, for the optimization problem under discussion, the focus is on adopting a metaheuristic optimization approach. Therefore, a methodological discussion of the optimization strategy is also crucial (Cruz et al., 2012; Cruz et al., 2018; Souza et al., 2020).

Queueing Theory

Queueing theory studies initially emerged in a more theoretical context, in studies of stochastic processes. However, its broad applicability quickly led to applications in statistics, applied mathematics, operations research, and other fields. Queueing research focuses on the functioning of systems in which demand tends to exceed the capacity of the service supply. Queueing theory studies employ a specific nomenclature and conceptual framework. The terms customers, servers, buffers, and rates, among others, are paramount (Gross et al., 2009).

Customers are usually those seeking the service, not necessarily individuals; in practice, any entity waiting for service is considered a customer. Servers are those who provide the service in question, just as customers, not necessarily individuals, can be attendants, electronic terminals, computer processors, among many other possible examples. It is essential to note that servers are system resources, meaning they are entities directly linked to system performance and incur a usage cost. Buffers, or waiting areas, are spaces designated for the waiting process for service. In some contexts, they are unlimited and not relevant to performance. However, in some specific contexts, particularly in this study, buffers are limited and constitute important system resources. Rates determine the frequency of occurrences of interest in the system. The arrival rate (λ) represents the frequency with which customers arrive seeking service. The service rate (μ) represents the frequency of services provided per unit of time (Ghimire et al., 2017).

Server Allocation Problem (SAP) Formulation

The first description is a mono-objective formulation, a representation of SAP in a basic formulation by (Duarte, 2024):

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

s.t.:

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

where w_i represents the weight of the cost associated with the number of servers c_i for the i^{th} queue in the queueing network, $\mathbf{C}$ is the vector of server numbers, and $\Theta(\mathbf{C})$ is the service rate of the queueing network, that is, the rate at which customers leave the queueing system. The primary objective is to minimize the cost of servers in a network with m interconnected queues, subject to a minimum threshold for the service rate, $\Theta_{\min}$.

Despite its similarity to an integer linear optimization problem, $\Theta(\mathbf{C})$ is difficult to define, since it is a function dependent on the arrival and service rates and also on the queueing network topology.

Buffer Allocation Problem (BAP) Formulation

Again, a description with a mono-objective formulation is presented, now representing the BAP by (Smith; Cruz, 2005):

$$\text{minimize } \sum_{i=1}^{m} w_i b_i, \tag{3}$$

s.t.:

$$\begin{aligned} &\Theta(\mathbf{B}) \geqslant \Theta_{\min}, \\ &b_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\}, \end{aligned} \tag{4}$$

where w_i represents the weight associated with the number of buffers b_i for the i^{th} queue in the queueing network, $\mathbf{B}$ is the vector of buffer allocations, and $\Theta(\mathbf{B})$ is the service rate of the queueing network. The central objective here is to minimize the total buffers allocated in a network with m interconnected queues, subject to a minimum service rate threshold $\Theta_{\min}$.

Buffer and Server Area Allocation Problem (BSAP) Formulation

Finally, a description for Buffer and Server Area Allocation Problem in a mono-objective formulation is presented by (van Woensel et al., 2010):

$$\text{minimize } \sum_{i=1}^{m} \alpha b_i + (1 - \alpha) c_i, \tag{5}$$

s.t.:

$$\begin{aligned} &\Theta(\mathbf{B}, \mathbf{C}) \geqslant \Theta_{\min}, \\ &0 \leqslant \alpha \leqslant 1, \forall i \in \{1, 2, \ldots, m\}, \\ &b_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\}, \\ &c_i \in \mathbb{N}, \forall i \in \{1, 2, \ldots, m\}, \end{aligned} \tag{6}$$

where b_i represents the number of buffers allocated and c_i represents the number of servers allocated to the i^{th} queue in the queueing network, $\mathbf{B}$ is the vector of buffer allocations, $\mathbf{C}$ is the vector of server allocations, the constants α and $1 - \alpha$ define the proportionality relationship between the cost of buffers and servers, and $\Theta(\mathbf{B}, \mathbf{C})$ is the service rate of the queueing network. The objective function is the function associated with the resource consumption in buffers and servers in a network with m queues, subject to a service threshold $\Theta_{\min}$.

This last formulation is indeed the direct interest of this study. Otherwise, the objective is to minimize resource consumption while ensuring that some minimum throughput is met.

Fine-tuning the PSO Algorithm for Optimization in Queueing Networks

The initial step in proposing a PSO algorithm is to define the particle representation for the problem under study explicitly. In this context, each particle must have its position well determined. The position of the i^{th} particle in the search space is represented by $\mathbf{x_i} = (x_{i1}, x_{i2}, \ldots, x_{in})$. Particularly for a network composed of m rows, in this study $n = 2m$ and $\mathbf{x_i} = (b_{i1}, b_{i2}, \ldots, b_{im}, c_{i1}, c_{i2}, \ldots, c_{im})$. The particle velocity is represented by $v_i = (v_{i1}, v_{i2}, \ldots, v_{in})$. According to the proposal of (Souza et al., 2020), let s be the size of the particle population (swarm), then each particle i, with $1 \leqslant i \leqslant s$, has the following attributes:

- particle position $\mathbf{x_i} = (x_{i1}, x_{i2}, \ldots, x_{in})$;
- particle velocity $\mathbf{v_i} = (v_{i1}, v_{i2}, \ldots, v_{in})$;
- personal best position p_i^{best};
- global best position g^{best}.

The best position of the i^{th} particle during the searches is given by $p_i^{best} = (p_{i1}, p_{i2}, \ldots, p_{in})$. The velocity and position of the particles are updated from iteration t to iteration $t+1$ according to the equations:

$$v_i^{t+1} = w^t + r_1\big(p_i^{best} - x_i^t\big) + r_2\big(g^{best} - x_i^t\big), \quad (7)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}; \quad (8)$$

The parameters of the PSO algorithm were defined by r_1 and r_2 which are positive random numbers with uniform distribution $\mathcal{U} \sim [0, 1]$, $w(t)$ is the inertia weight. The inertia weight was set $w(t) = 0.4$. The PSO described here is an adaptation of the classical implementation by (Coello; Lechuga, 2002). In this study, the decision variables are integers and must be updated accordingly, Eq. (9):

$$x_i^{t+1} = \text{int}\Big(x_i^t + v_i^{t+1}\Big); \quad (9)$$

The choice of the best position of the i^{th} particle (p_i^{best}) is made at each iteration, highlighting the continuous improvement and refinement of the process. If the new position is superior to the position p_i^{best}, it is updated by the new position x_i^{t+1}. If the current position is inferior to the position p_i^{best}, the position p_i^{best} is preserved. Suppose p_i^{best} is neither superior nor inferior (the same objective function value in a different allocation) to the current position x_i^{t+1}. In that case, the choice is made randomly between p_i^{best} and x_i^{t+1}. The global best position (g^{best}) is chosen as the best position p_i^{best} among all particles. Algorithm 1 presents the pseudo-code of the implementation used in this study.

Algorithm 1 PSO

Require: $\mathcal{G}(V, A), \lambda_i \; \forall i \in V$
1: $X^0 \leftarrow$ **GenerateInitialSwarm**(`swarmSize`)
2: $P \leftarrow X^0$
3: **for** $t = 0$; $t <$ `maxIter`, t++ **do** ▷ /* *begin move swarm* */
4: $g^t \leftarrow$ **SelectBest**(P)
5: **for** $i = 1$; $i \leqslant$ `swarmSize`; i++ **do**
6: $v_i^{t+1} \leftarrow$ **Speed**(x_i^t, p_i^t, g^t)
7: $x_i^{t+1} \leftarrow$ **NewPosition**(x_i^t, v_i^t)
8: **if** x_i^{t+1} *overcomes* p_i^t **then** $p_i^t \leftarrow x_i^{t+1}$
9: **else**
10: **if** p_i^t *overcomes* x_i^{t+1} **then** $p_i^t \leftarrow p_i^t$
11: **else** $p_i^t \leftarrow$ **Rand**(x_i^{t+1}, p_i^t)
12: **end if**
13: **end if**
14: **end for**
15: $P \leftarrow X^{t+1}$
16: **end for** ▷ /* *end move swarm* */
17: **write** $X^{\texttt{maxIter}}$ ▷ /* *write final solution* */

Numerical Results and Insights

The optimization algorithm used in this study was implemented using R statistical software R (R Core Team, 2021). Additionally, the R packages `igraph` and `queueing` were used in the implementation. The codes implemented in this study are available for research and educational purposes upon prior request. The execution environment for the computational experiments was an 11th Gen Intel® Core™ i7-1165G7 processor, running at 2.80 GHz, with Windows 11 Home Single Language 64-bit and 8 GB of RAM.

Different complex queueing network topologies were employed in the experiments, aiming to evaluate the adaptability of the proposed algorithm to various computational scenarios. To maintain the generalizability of the study, all experiments considered servers with the same workload, that is, the same service rate μ. However, it is important to note that there are no restrictions on conducting new experiments with servers of different capacities or other variations. For all topological structures investigated the arrival rate was always fixed at $\lambda = 5$.

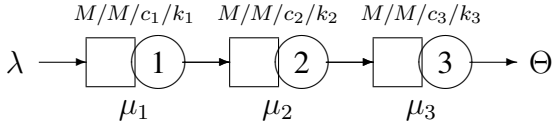
In all executions, the initial solutions were generated completely randomly. However, this condition is not guaranteed after the solutions have evolved using the PSO algorithm. The goal is to produce solutions that are more efficient in terms of total cost and that do not compromise performance in terms of service rate.

The algorithm was configured with a maximum number of cycles fixed at $\texttt{numIter} = 2000$. The number of initial solutions `swarmSize` was set to 50, all randomly generated. In the problem formulation presented in equations (5) and (6), one of the constraints is to ensure that $\Theta(\mathbf{B}, \mathbf{C}) \geqslant \Theta_{\text{min}}$. In this study, Θ_{min} is the Θ value of the initial solution. This fact means that the algorithm must be able to reduce resource costs (buffers and servers) while still meeting the constraint of preserving or increasing the queueing network's service rate.

It is essential to note that this formulation aims to ensure that reducing the budgetary resources required for the queueing network does not compromise performance. Specific results for each topology under investigation will be presented below.

Several procedures of practical interest are composed of series-coupled queues; many production processes feature these topologies. Figure 2 shows a specific experimental series topology analyzed in this study.

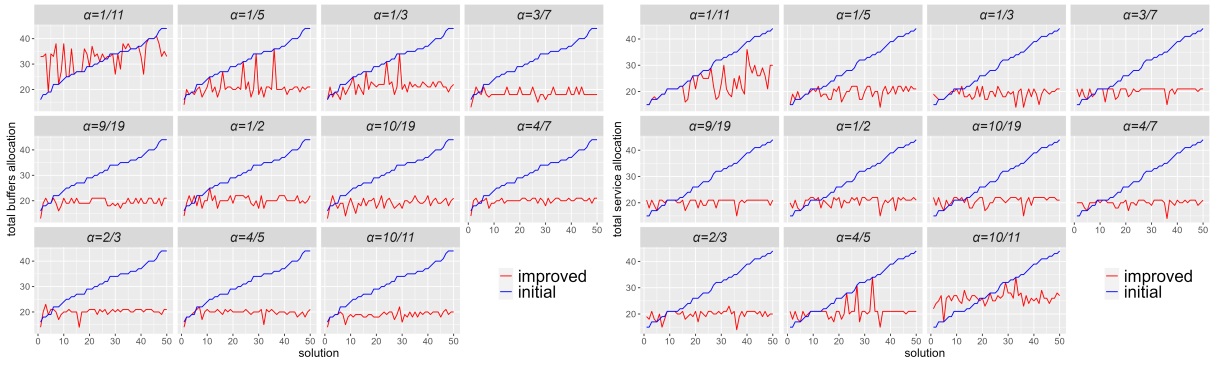
Figure 2: Experimental topology with 3-node serial queueing network.



Source: from the authors (2025).

The experimental results for the queueing network presented in Figure 2 are detailed in Figures 4(a), 4(b), and 4. The graphical analyses were performed with the aid of the graphics package `ggplot2` by Wickham (2016). Figure 4 uses the optimization formulation described in Eqs. (5) and (6). The values α represent the difference in the cost ratio between the allocated buffers and the allocated servers. For example, if $\alpha = {}^{1}/_{11}$, then $1 - \alpha = {}^{10}/_{11}$, which indicates that a server is 10 times more expensive than a buffer. Several cost relationships between buffers and servers were investigated.

Figure 3: Resource allocation for different α values in the queueing network of Figure 2



(a) Capacity allocation (buffers).

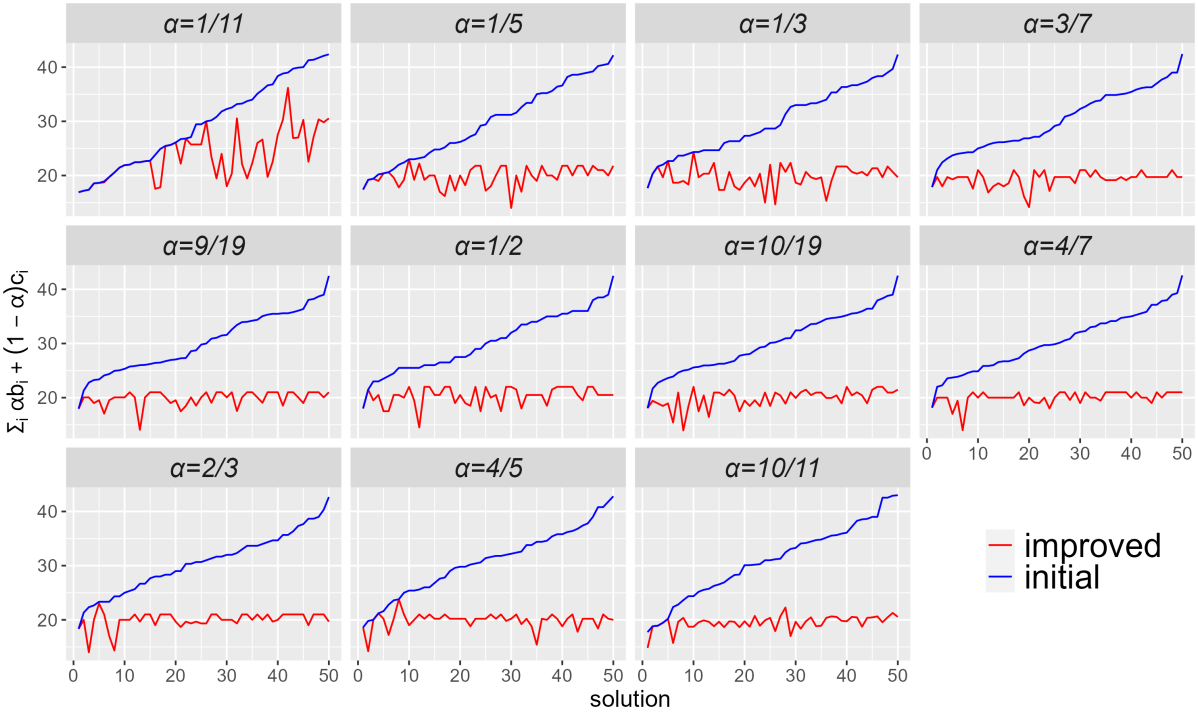
(b) Total server allocation.

Source: from the authors (2025).

Figure 3(a) compares the buffer allocation in the optimized solutions with the initial solutions. For smaller values of α, where buffers are less expensive, the algorithm does not prioritize buffer savings. However, as α increases, reflecting the higher relative cost of buffers compared to servers, the algorithm begins to offer solutions that emphasize savings in buffer allocation.

Figure 3(b) shows the Total server allocation in the optimized solutions compared to the initial proposed solutions. For low values of α where servers are more expensive, the algorithm strongly prioritizes reducing server costs. As α increases, the algorithm provides solutions that allow for greater server resource consumption.

Figure 4: Improvements by values α in the topology of Figure 2.

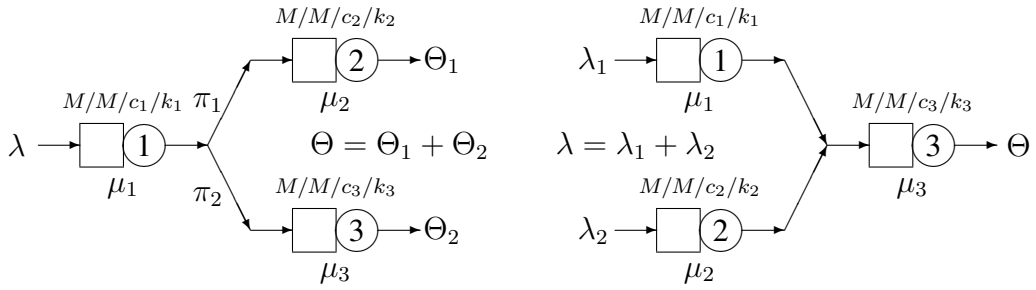


Source: from the authors (2025).

The PSO algorithm demonstrated a significant ability to reduce the investment required in queueing networks while maintaining compliance with throughput constraints (see Figure 4). The algorithm's effectiveness in obtaining more economical solutions increases as servers become less costly relative to buffers. Regardless of the cost ratio between buffers and servers, PSO was able to impressively reduce the resource consumption allocated to queueing network operations, achieving substantial savings.

Queueing networks, a relevant part of many practical procedures, often feature task arrangements that go beyond the simple series structure. Figure 5 illustrates two such diverse topologies for queueing networks.

Figure 5: Experimental topologies with queueing networks in merging and splitting.



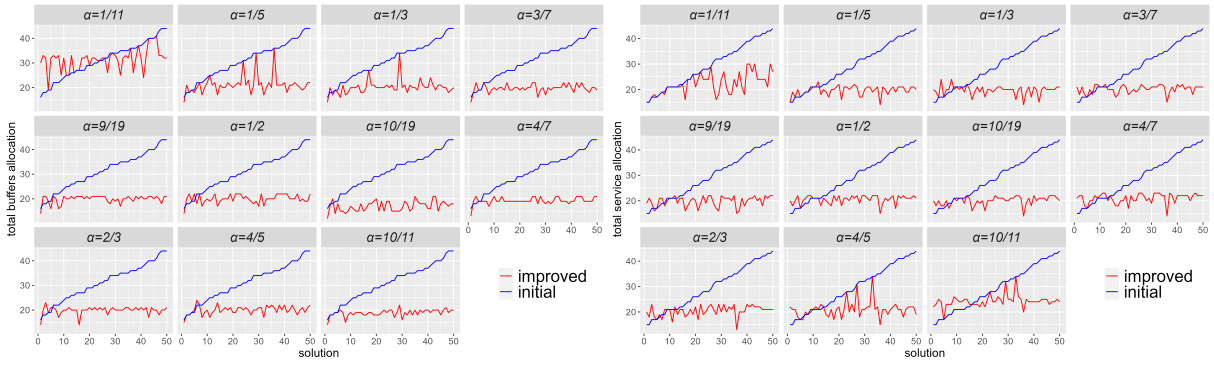
(a) 3-node queueing network (split). (b) 3-node queueing network (merge).

Source: from the authors (2025).

While series arrangements are common in industrial settings, it's important to note that diverse topologies are also prevalent. These structures, which are found in processes related to in-person and remote services, production, traffic, and more, involve the merging and splitting of queues within the system, reflecting the real-world complexity of these operations.

The results obtained for the queueing network in Figure 6(a) with the split topological structure are presented in Figures 7(a), 7(b), and 7. The graphical presentation uses the same pattern previously presented.

Figure 6: Resource allocation for different α values in the queueing network of Figure 6(a).



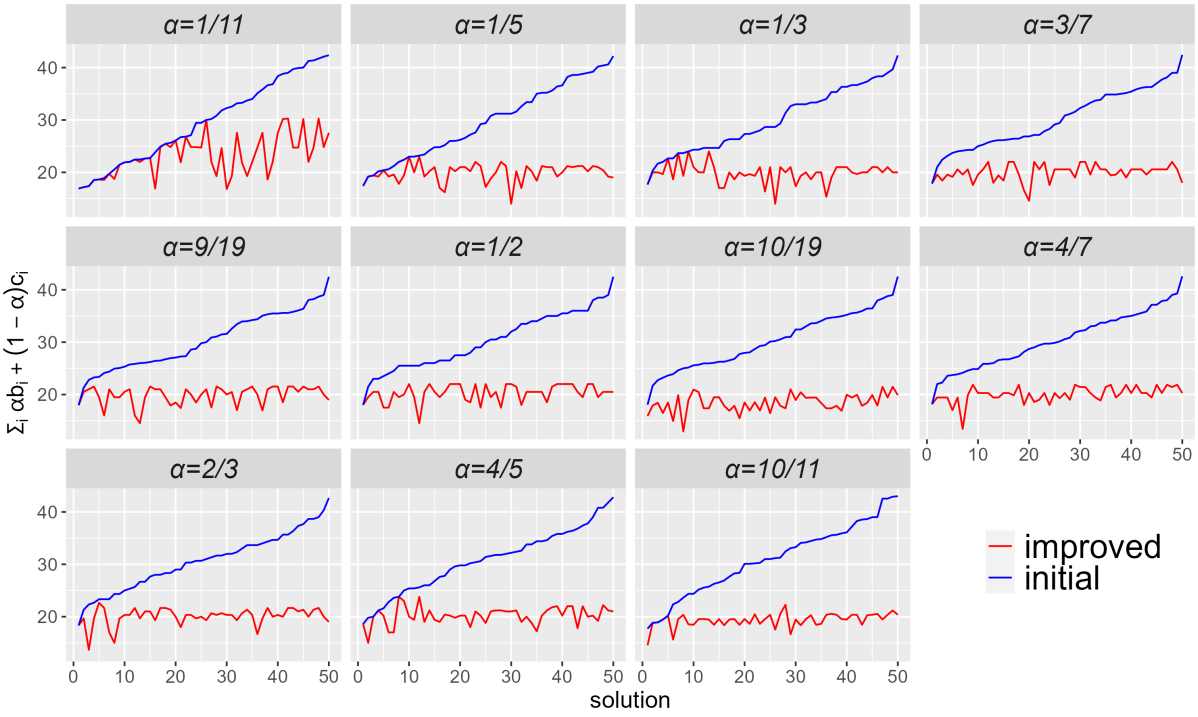
(a) Capacity allocation (buffers). (b) Total server allocation.

Source: from the authors (2025).

Once again, the PSO algorithm was able to reduce resource allocation even under the constraint imposed on the throughput. Figure 7(a) shows the total buffer allocation. For lower values of α, the PSO algorithm uses many buffers, due to the lower cost associated with these buffers. Figure 7(b) shows the Total server allocation. When α is lower, the algorithm prioritizes reducing server allocation; predictably, this strategy is adjusted as the value of α increases.

Figure 7 shows the reductions achieved for the different values α tested for the objective function of the mathematical formulation of the proposed problem. It is possible to verify that, regardless of the value α, the algorithm is capable of providing reductions in resource allocation.

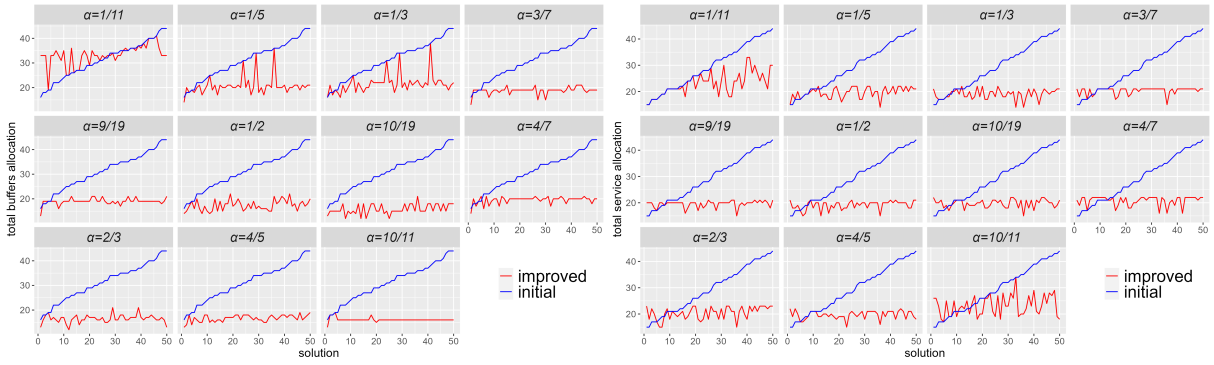
Figure 7: Improvements by values α in the topology of Figure 6(a).



Source: from the authors (2025).

The results obtained for the queueing network in Figure 6(b) with the merge topological structure are presented in Figures 9(a), 9(b), and 9.

Figure 8: Resource allocation for different α values in the queueing network of Figure 6(b).

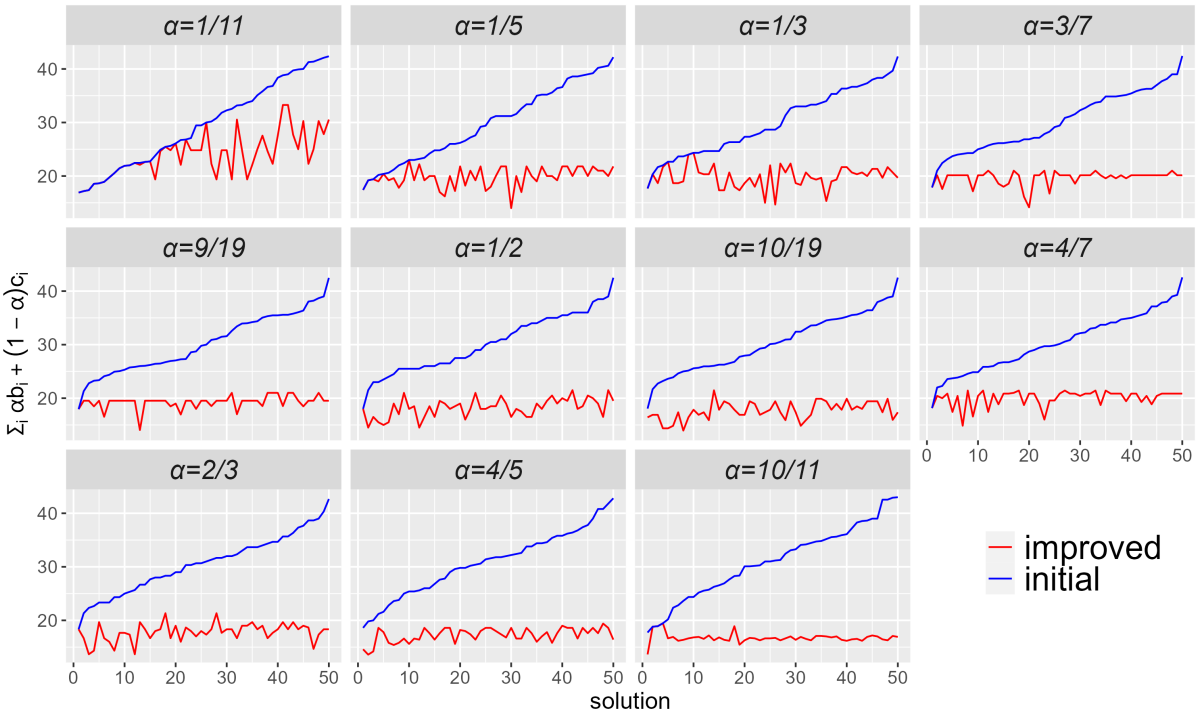


(a) Capacity allocation (buffers). (b) Total server allocation.

Source: from the authors (2025).

As previously verified for split-topology queueing networks, the PSO algorithm successfully reduced resource allocation in queueing networks, even under the constraint imposed on queueing network service rates. Figure 9(a) details the total buffer allocation in the queueing network. As the value of α increases, the PSO algorithm increases the number of buffers allocated, since the cost associated with buffers is lower. Figure 9(b) shows the total server allocation with similar effects to those of buffers. The PSO algorithm prioritizes reducing server allocation as the value of α increases, and the PSO algorithm increases server consumption. Figure 9 presents the effects obtained on the objective function by the PSO algorithm for the different α values investigated.

Figure 9: Improvements by values α in the topology of Figure 6(b).

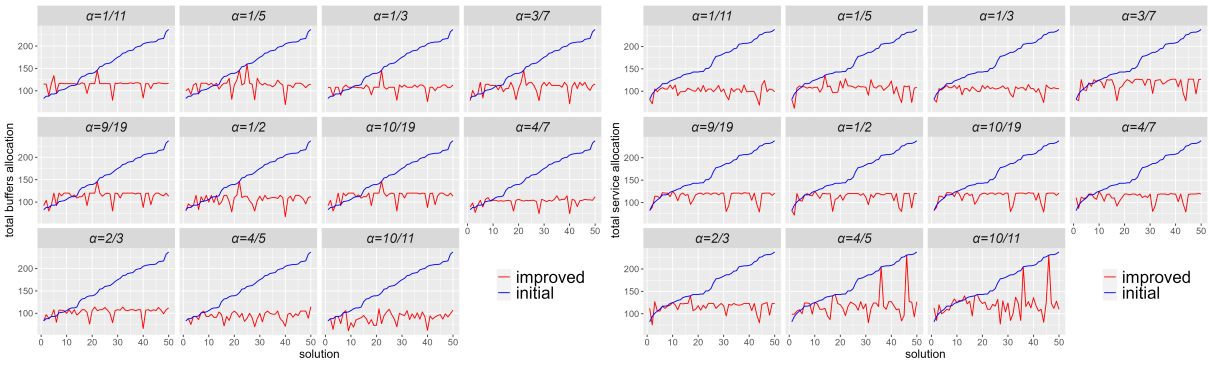


Source: from the authors (2025).

It can be seen that, unlike the split topology, for smaller α values, the PSO algorithm achieved smaller reductions in the merge topology. In contrast, the opposite effect occurred for larger α values, that is, the PSO algorithm achieved greater reductions in the weighted cost function for buffers and servers.

Finally, we explore possibilities involving mixed queueing networks, which present a higher level of complexity in their topological structure. Many real-world problems of practical interest often involve situations that combine series, splits, and merges. Figure 1 presents a topology in this context. We present the results of using the PSO algorithm for this queueing network in Figures 11(a), 11(b), and 11.

Figure 10: Resource allocation for different α values in the queueing network of Figure 1.



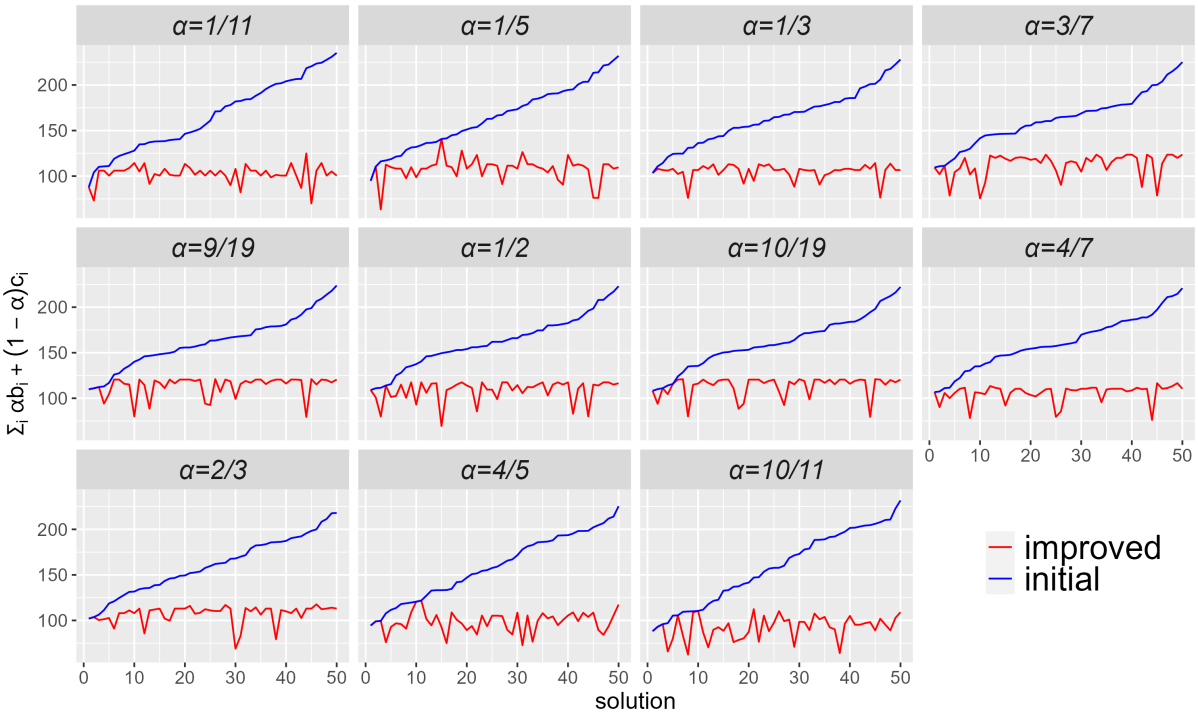
(a) Capacity allocation (buffers).

(b) Total server allocation.

Source: from the authors (2025).

Again, the PSO algorithm provided promising results regardless of the value of α. Particularly for $\alpha \in \{4/5, 10/11\}$, the variability in the total server allocation among the proposed solutions increased significantly. Despite this, in all contexts, the PSO algorithm consistently provided improvements over the initially proposed solutions. The results for the objective function are presented in Figure 11.

Figure 11: Improvements by values α in the topology of Figure 1.



Source: from the authors (2025).

When considering a larger and more complex queueing network (see Figure 1), the PSO algorithm also demonstrates its effectiveness in reducing resource allocation in queueing networks, even with throughput constraints. Figure 11 confirms this analysis for all investigated α values.

Final Remarks

This study proposes an analysis of the buffer and server allocation problem (BSAP). It includes a up-to-date literature review of recent studies on resource allocation in queueing networks. Specifically for BSAP, a formulation in terms of a mathematical programming model is presented.

We examined BSAP using a bioinspired optimization algorithm, the classic Particle Swarm Optimization (PSO). A practical and efficient fine-tuning for this specific problem was presented through a coded implementation for use in the statistical software R.

Among the contributions, a performance evaluation of the PSO algorithm was presented in different topological instances of queueing networks. These topologies, which include series, merge, split, and a more general mixed topology, represent practical scenarios. The PSO algorithm demonstrated efficiency in reducing total costs and maintaining the quality of service in terms of throughput. The effectiveness of the algorithm was demonstrated by its ability to optimize server and buffer allocation even when pre-conditioning throughput constraints.

The PSO algorithm significantly reduced the required costs. This budget reduction effect was especially noticeable when the relative cost of servers compared to buffers was higher. By reducing allocated resources without compromising throughput, we confirmed the robustness of the PSO algorithm. It reliably deals with variations in the cost ratio between buffer and server

allocation, providing a sense of reassurance in its performance.

The ability of the algorithm to adapt its allocation strategy according to the value of α (the cost ratio between buffers and servers) reflected the flexibility and efficiency of the PSO algorithm in managing resources in complex queueing networks.

The results obtained indicate that the PSO algorithm is a powerful and effective tool for solving optimization problems in queueing networks, regardless of the complexity of the topologies involved. The PSO algorithm's ability to provide more economically adaptable solutions, even while subject to performance constraints, is a clear indication of its effectiveness. A trend toward improved PSO performance was observed with the reduction of the relative costs of servers compared to buffers; however, results with the opposite cost ratio are still effective. This effectiveness instills confidence in the PSO algorithm's capabilities.

Although the results are quite promising, there is still room for improvement and further exploration. It is recommended that future research introduce additional variables, such as different customer arrival rates or priority policies, to test the robustness of the PSO algorithm in even more specific scenarios. Furthermore, applying the PSO algorithm to optimization problems involving queueing networks with variable demand characteristics or additional constraints can provide valuable insights into the adaptability of the PSO algorithm to problems of this nature, thereby encouraging further investigation.

Future research also includes assessing the quality of estimating other network queue performance measures, such as the probability of server idleness, the system waiting time (W), and the average queue time (W_q). The potential for further investigations in this field is vast, including the study of queues with different structures, such as multi-server Markovian queues without buffer constraints ($M/M/c$), among other possibilities. These are just some of the many exciting topics for future work in this field.

Another possibility for further research is comparing the PSO algorithm with other optimization algorithms. The aim is to evaluate relative performance in various optimization contexts. Comparative analysis can help identify the specific advantages and limitations of the PSO algorithm in comparison to other approaches.

Conflicts of interest and the use of artificial intelligence

The authors declare that there is no conflict of interest in conducting this research, and the authors also confirm that no AI resources were used.

Acknowledgements

The authors would like to thank the Universidade Federal de Ouro Preto for partially supporting the development of this study.

References

ABENSUR, E. O. Banking operations using queuing theory and genetic algorithms. **Produto & Produção**, v. 12, n. 2, 2011.

CARTER, M.; PRICE, C. C.; RABADI, G. **Operations research: a practical introduction**. [S.l.]: Chapman and Hall/CRC, 2018.

COELLO, C. A. C.; LECHUGA, M. S. MOPSO: A proposal for multiple objective particle swarm optimization. In: **Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)**. [S.l.: s.n.], 2002. v. 2, p. 1051–1056.

COLLETTE, Y.; SIARRY, P. Multiobjective optimization: principles and case studies. **OR/MS Today**, Institute for Operations Research and the Management Sciences, v. 31, n. 1, p. 60–61, 2004.

CRUZ, F. R. B.; DUARTE, A. R.; SOUZA, G. L. Multi-objective performance improvements of general finite single-server queueing networks. **Journal of Heuristics**, Springer, v. 24, n. 5, p. 757–781, 2018.

CRUZ, F. R. B.; KENDALL, G.; WHILE, L.; DUARTE, A. R.; BRITO, N. C. L. Throughput maximization of queueing networks with simultaneous minimization of service rates and buffers. **Mathematical Problems in Engineering**, Hindawi, v. 2012 - Article ID 692593, p. 19 pages, 2012.

DEB, K. **Optimization for engineering design: Algorithms and examples**. [S.l.]: PHI Learning Pvt. Ltd., 2012.

DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: Nsga-ii. **IEEE transactions on evolutionary computation**, IEEE, v. 6, n. 2, p. 182–197, 2002.

DUARTE, A. R. The server allocation problem for Markovian queueing networks. **International Journal of Services and Operations Management**, v. 48, n. 2, p. 256–271, 2024.

DUARTE, A. R.; CRUZ, F. R. B.; SOUZA, G. L. A greedy post-processing strategy for multi-objective performance optimization of general single-server finite queueing networks. **Soft Computing**, Springer, v. 28, n. 17, p. 9483–9494, 2024.

GHIMIRE, S.; THAPA, G. B.; GHIMIRE, R. P.; SILVESTROV, S. A survey on queueing systems with mathematical models and applications. **American Journal of Operation Research**, v. 7, n. 1, p. 1–14, 2017.

GROSS, D.; SHORTLE, J. F.; M., T. J.; M., H. C. **Fundamentals of Queueing Theory**. 4th edition. ed. New York, NY: Wiley - Interscience, 2009.

JAIN, M.; SAIHJPAL, V.; SINGH, N.; SINGH, S. B. An overview of variants and advancements of pso algorithm. **Applied Sciences**, MDPI, v. 12, n. 17, p. 8392, 2022.

KENDALL, D. G. Stochastic processes occurring in the theory of queues and their analysis by the method of embedded Markov chains. **Annals Mathematical Statistics**, v. 24, p. 338–354, 1953.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: **Neural Networks, 1995. Proceedings., IEEE International Conference on**. [S.l.: s.n.], 1995. v. 4, p. 1942–1948.

MISEREZ, J.; COLLE, D.; PICKAVET, M.; TAVERNIER, W. Exploiting queue information for scalable delay-constrained routing in deterministic networks. **IEEE Transactions on Network and Service Management**, IEEE, 2024.

R CORE TEAM. **R: A Language and Environment for Statistical Computing**. Vienna, Austria, 2021. Disponível em: ⟨https://www.R-project.org/⟩.

RINGUEST, J. L. **Multiobjective optimization: behavioral and computational considerations**. [S.l.]: Springer Science & Business Media, 2012.

SHARMA, S.; KUMAR, V. A comprehensive review on multi-objective optimization techniques: Past, present and future. **Archives of Computational Methods in Engineering**, Springer, v. 29, n. 7, p. 5605–5633, 2022.

SMITH, J. M.; CRUZ, F. R. B. The buffer allocation problem for general finite buffer queueing networks. **IIE Transactions**, v. 37, n. 4, p. 343–365, 2005.

SOUZA, G. L.; DUARTE, A. R.; MOREIRA, G. J. P.; CRUZ, F. R. B. A novel formulation for multi-objective optimization of general finite single-server queueing networks. In: IEEE. **Proceedings of the 2020 Congress on Evolutionary Computation. CEC'20**. [S.l.], 2020. p. 1–8.

VAN WOENSEL, T.; ANDRIANSYAH, R.; CRUZ, F. R. B.; J., M. S.; KERBACHE, L. Buffer and server allocation in general multi-server queueing networks. **International Transactions in Operational Research**, Wiley Online Library, v. 17, n. 2, p. 257–286, 2010.

WICKHAM, H. **ggplot2: Elegant Graphics for Data Analysis**. Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. Disponível em: ⟨https://ggplot2.tidyverse.org⟩.

YANG, X. S. **Engineering Optimization: An Introduction with Metaheuristic Applications**. 1st. ed. [S.l.]: Wiley Publishing, 2010. ISBN 0470582464, 9780470582466.

ZHONG, Z.; CAO, P.; HUANG, J.; ZHOU, S. X. Capacity allocation and scheduling in two-stage service systems with multiclass customers. **Manufacturing & Service Operations Management**, INFORMS, v. 26, n. 5, p. 1842–1859, 2024.