Sharif University of Technology

Scientia Iranica

Transactions on Industrial Engineering

https://scientiairanica.sharif.edu

Reliability-redundancy allocation problem of a queueing system considering energy consumption

Kamyar Sabri-Laghaie [a,*], Mahdi Fathi [b]

a. Department of Industrial Engineering, Urmia University of Technology, Urmia, Iran.
b. Department of Information Technology and Decision Sciences, G. Brint Ryan College of Business, University of North Texas, Denton, Texas, USA.

* Corresponding author: sabri@uut.ac.ir (K. Sabri-Laghaie)

Received 13 April 2021; Received in revised form 12 December 2022; Accepted 16 May 2022

Keywords

Reliability-Redundancy Allocation Problem (RRAP);
Memetic Algorithm (MA);
Queueing system;
Availability;
Markov chain modeling.

Abstract

In Reliability-Redundancy Allocation Problem (RRAP), the reliability and redundancy of components in a given system configuration are determined while considering some problem-specific constraints. RRAP can be applied in various industries. Moreover, queueing systems are among the most common systems in the manufacturing and service industries. Failure in queueing systems can result in unwanted severe damages. Reliability analysis of queueing systems should be conducted concerning their performance measures. Therefore, a RRAP of a queueing system considering queueing costs is studied in this article. The proposed cost function includes queueing, repair, and energy consumption costs. A Memetic Algorithm (MA) is used to obtain optimal redundancy and failure rates of components and the system's service rate, which affects the energy consumption level. Extensive numerical experiments and sensitivity analyses are performed to present the problem's applicability and the proposed algorithm.

1. Introduction

Complicated systems play essential roles in many aspects of the modern world. Failure of these systems can impact our lives and cause significant losses. Therefore, the reliability of these systems should be ensured during their useful life. In this regard, system reliability optimization has gained much more attention throughout the last decades [1]. In general, two main approaches are utilized to improve the system reliability: (i) Enhancing the reliability of system components and (ii) Including redundant components in the subsystems. The first approach cannot always be followed, and there are usually limitations on the degree to which the reliability of components can be increased. The second approach, Redundancy Allocation Problem (RAP), can choose the optimal redundancy and component combination to achieve higher reliabilities. However, some problem-specific constraints like weight, volume, or available budget may be exceeded. In addition to the mentioned approaches, combining these two ways is also an alternative way to increase system reliability [2]. This problem by which redundancy and reliability of components are simultaneously decided is the so-called Reliability-Redundancy Allocation Problem (RRAP) [3].

RRAP is one of the most common problems in the design-for-reliability phase and initial stages of the system design process. In this regard, mixed-integer reliability problems with physical and/or budgetary constraints are solved. RRAP is widely applied in various fields to reduce the impacts of undesired failures [4]. One of the most common systems in service and manufacturing industries for which RRAP can be applied is the queueing system. Failure in queueing systems can impose severe impacts on customer satisfaction and future profitability of industries. Therefore, much more attention should be given to the reliability of queueing systems [5]. In this regard, RRAP can be useful in the design process of these systems.

Failure of queueing systems directly affects their performance measures, such as average customers' waiting time, queue length, and system utilization. Optimizing system reliability without considering a queueing system's performance measures can result in system designs with

To cite this article:
K. Sabri-Laghaie, M. Fathi "Reliability-redundancy allocation problem of a queueing system considering energy consumption", *Scientia Iranica* (2025) **32**(12): 5562. https://doi.org/ 10.24200/sci.2022.58080.5562

2345-3605 © 2025 Sharif University of Technology. This is an open access article under the CC BY-NC-ND license.

higher queueing costs. Improving the reliability of queueing systems regarding the system's available resources and performance measures can be useful in confronting this problem. For instance, Garg and Sharma [6] modeled the RRAP of a pharmaceutical plant, which can be considered a series-parallel system. Although the pharmaceutical plant can be contemplated as a queueing system, Garg and Sharma [6] did not consider the queueing costs in the analysis of the RRAP. Designing this system with respect to queueing costs can reduce the risk of unwanted queueing costs.

Formerly, Sabri-Laghaie and Karimi-Nasab [7] modeled the RAP for a queueing system and developed approximate algorithms to solve this problem. They found the optimum number of redundancies and repairmen by which the queueing system's total cost, including queueing costs, is minimized. It can be observed that there are very few researches in the literature that has modeled RAP and RRAP of queueing systems. As mentioned, the reliability of queueing systems can be improved by considering standby servers. However, increasing the number of servers can impose different expenses, such as energy consumption costs. Therefore, the current research studies the RRAP of a queueing system with maintenance and energy consumption considerations.

RRAP is an NP-hard problem and is supposed to necessitate a huge computational effort to find an optimal solution. RRAP of queueing systems can even be more troublesome to solve. RRAP has been extensively studied under different assumptions and various solution methodologies. Due to the complexity of the RRAP, most of the researches has focused on developing heuristic and metaheuristic approaches. In this regard, we can refer to Simulated Annealing (SA) [8], Genetic Algorithm (GA) [4, 9-12], Particle Swarm Optimization (PSO) [5,6,13-16], artificial bee colony algorithm [3,17], artificial immune search [18], Biogeography-Based Optimization (BBO) [19], fruit fly optimization algorithm [20], Markov decision process [21], Stochastic Fractal Search (SFS) [22], and hybrid algorithms such as SFS-GA [23]. In addition to heuristic and metaheuristic algorithms, simulation-based solution approaches [24] and exact solution methods such as implicit enumeration, branch-and-bound, and dynamic programming have also been used to solve RRAP [23]. GA has been successfully applied to the RRAP [4,9,10]. However, GA lacks a proper local search mechanism around each solution. Therefore, a Memetic Algorithm (MA) that adds the SFS algorithm's diffusion process to the GA is proposed in this research. We use the diffusion process to improve the local search capacity of the GA. Many researchers in the literature have used hybrid versions of GA to improve the quality of solutions. For instance, Sharifi et al. [25] presented a hybrid GA for reliability optimization of a *k*-out-of-*n* series-parallel system with warm standby components. Kim and Kim [10] proposed a parallel GA to solve an advanced RRAP that also considers the optimal redundancy strategy and imperfect switch.

Hence, the RRAP of a queueing system with maintenance and energy consumption considerations is investigated in this article. A MA that incorporates GA with the diffusion process of the SFS algorithm is applied to find the redundancy level and failure rate of components and the system's service rate. Higher service rates lead to better performance measures of the queueing system. However, systems with higher service rates consume more energy. Moreover, increasing the components' reliability can result in fewer system stoppages and improve the queueing system. Also, higher repair rates that incorporate elevated costs can reduce the stoppage time of the system. Optimal values of these variables can be found to realize a trade-off between cost and reliability.

The rest of this article is organized as follows. In Section 2, assumptions and notations are reviewed. Section 3 is devoted to problem formulation and cost analysis. In Section 4, the solution methodology is introduced. In Section 5, a numerical example and sensitivity analysis are given. Finally, concluding remarks and future research directions are presented in Section 6.

2. Problem definition

In this section, the assumptions and notations of the problem are described.

2.1. Assumptions

A queuing server is considered in this research, which serves arriving customers. Failures may occur for a server in exponentially distributed times with rate ν. A failed server is repaired with repair times, that are exponentially random variables with rate δ. Customer arrival and service times also follow an exponential distribution with rates λ and μ, respectively. It is supposed that the service rate is a variable that affects the system's energy consumption. Increasing service rate involves higher energy consumption levels.

By including some redundancy for the main server, one can increase the system's availability. In this case, both cold and hot standbys are considered. A cold standby is inactive and whenever the main server fails becomes active. In other words, the failure rate of a cold standby is zero while it is inactive. On the other hand, a hot standby is always active, and its failure rate is the same as the main server. The system fails whenever all standbys are under repair, and there is no unfailed server to be replaced by the failed one. In this case, all servers are repaired, and the system becomes operational again. Time to repair all servers follows an exponential distribution with rate ω. Here, it is intended to find optimal service and failure rates and the optimal number of redundant servers. The objective function is to minimize the total cost per unit time that includes customers' holding cost, energy consumption cost, repair cost of the standbys, and maintenance and preservation costs of the system. Furthermore, the following assumptions are adopted in this study:

- There is no limitation on the number of servers that could be supplied;
- Failure of a server does not damage other servers;
- Parameters of servers like cost and weight are deterministic and known;
- Hazard rates of servers are constant;
- Individual servers fail independently;
- Failed servers are renewed after repair;
- The switch for server activation in the cold standby strategy is perfect;

Table 1. The notation used for the modeling of the proposed system.

Notation	
M	The transition rates matrix of the system states
n	Number of customers in the system
i	Total number of failed servers in the maintenance center
λ	The arrival rate of customers
δ	Repair rate of a server
ω	Repair rate of the system
π_{ni}	Steady-state probability of the system in state (n, i) with n customers in the system and *i* failed servers
$\boldsymbol{\pi}_n = (\pi_{n0}, \ldots, \pi_{n\beta})$	The probability vector of having *n* customers in the system for $n = 0,1,2,\ldots$
π_i	Steady-state probability of steady-state probability of having *i* failed serves in the system
R_1	The time between the instant when the system is repaired and its next failure instant
R_2	Time to repair of the system
BD	Available budget
W	Maximum allowable weight of the system
C_H	Holding cost of a customer per unit time
C_R	Fixed repair cost of a server per unit time
C_M	Unit time maintenance and stoppage cost of the system
A_M	Unit time fixed cost of system stoppage
C_P	Unit time preservation cost of the system
$C_{EC}(\mu, \beta)$	Energy consumption cost function
$N(t)$	Number of customers that arrive up to time *t*
W_k	Sojourn time of the *k* th customer
T_i	The duration between two successive repairs of a server while there are *i* failed servers in the system
$SC(\beta, \nu)$	The total cost of the servers of the system
$SW(\beta)$	Total weight of the system
$A(t)$	Availability function of the system
Decision variables	
β	Number of redundant servers
ν	Failure rate of a server
μ	Service rate of a server

- Preservation cost corresponds to idle and active servers;
- Energy consumption cost only corresponds to active servers.

2.2. Notations

The notation used for the modeling of the proposed system are defined in Table 1.

3. Problem formulation

This section provides the mathematical formulation of the problem.

3.1. Continuous time Markov chain modeling

A continuous Markov chain is used to model the problem. In this regard, the state of the system is considered as (n, i) in which $n \in \{0,1,\ldots\}$ and $i \in \{0,\ldots,\beta\}$ are total number of customers and failed servers, respectively. The transition rate matrix of the problem is defined as the following:

$$\mathbf{M} = \begin{pmatrix} \mathbf{A}_1 + \mathbf{A}_2 & \mathbf{A}_0 & & \\ \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 & \\ & \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 \\ & & \ddots & \ddots \end{pmatrix},$$

where $\mathbf{A}_0, \mathbf{A}_1$ and $\mathbf{A}_2$ are square matrices with the size of $\beta + 1$. $\mathbf{A}_0$ and $\mathbf{A}_2$ are defined as the following:

$$\mathbf{A}_2 = \begin{pmatrix} \mu & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \mathbf{A}_0 = \begin{pmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \lambda \end{pmatrix}.$$

In the case of cold standby redundancy, $\mathbf{A}_1$ is denoted as:

$$\mathbf{A}_1 = \begin{pmatrix} -(\lambda+\mu+\nu) & \nu & \cdots & 0 & 0 \\ \delta & -(\lambda+\mu+\nu+\delta) & \nu & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -(\lambda+\mu+\nu+\delta) & \nu \\ \omega & 0 & \cdots & 0 & -(\lambda+\omega) \end{pmatrix},$$

and for hot standby redundancy $\mathbf{A}_1$ is stated as:

$$\mathbf{A}_1 = \begin{pmatrix} -(\lambda+\mu+\beta\nu) & \beta\nu & 0 & 0 & 0 \\ \delta & -(\lambda+\mu+(\beta-1)\nu+\delta) & (\beta-1)\nu & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -(\lambda+\mu+\nu+\delta) & \nu \\ \omega & 0 & \cdots & 0 & -(\lambda+\omega) \end{pmatrix}.$$

π_{ni} is the steady state probability of the system in state (n,i). The probability vector of being n customers in the system, $n = 0,1,2,...$, is represented by $\boldsymbol{\pi}_n = (\pi_{n0},...,\pi_{n\beta})$. $\mathbf{A}$ is defined as $\mathbf{A} = \mathbf{A}_0 + \mathbf{A}_1 + \mathbf{A}_2$ and $\boldsymbol{\pi} = (\pi_0,...,\pi_\beta)$ is steady-state probabilities of $\mathbf{A}$. Applying the matrix-geometric approach in [26] to the Continuous-Time Markov Chain (CTMC) with matrix $\mathbf{M}$ results in:

$$\boldsymbol{\pi}_n = \boldsymbol{\pi}(I-\mathbf{S})\mathbf{S}^n, \quad n = 0,1,2,... \tag{1}$$

In Eq. (1), matrix $\mathbf{S}$ is the minimal nonnegative solution to equation:

$$\mathbf{S}^2\mathbf{A}_2 + \mathbf{S}\mathbf{A}_1 + \mathbf{A}_0 = 0. \tag{2}$$

3.2. Availability function

In this subsection, failure time distribution and availability of the system are calculated. Therefore, we divide the time between two successive failures of the system, that is R, into two intervals: (1) R_1, time between the instant when the system is repaired to its failure instant; (2) R_2, time to repair the system.

Calculating the absorption time in the Markov chain with the fundamental matrix $\mathbf{T}$, one can find R_1 . For the cold standby redundancy, matrix $\mathbf{T}$ is defined as:

$$\mathbf{T} = \begin{matrix} 0 \\ 1 \\ \vdots \\ \beta-2 \\ \beta-1 \end{matrix} \begin{bmatrix} -\nu & \nu & & & \\ \delta & -(\nu+\delta) & \nu & & \\ & \ddots & \ddots & \ddots & \\ & & \delta & -(\nu+\delta) & \nu \\ & & & \delta & -(\nu+\delta) \end{bmatrix}.$$

And in the case of warm redundancy is represented by:

$$\mathbf{T} = \begin{matrix} 0 \\ 1 \\ \vdots \\ \beta-2 \\ \beta-1 \end{matrix} \begin{bmatrix} -\beta\nu & \beta\nu & & & \\ \delta & -((\beta-1)\nu+\delta) & (\beta-1)\nu & & \\ & \ddots & \ddots & \ddots & \\ & & \delta & -(2\nu+\delta) & 2\nu \\ & & & \delta & -(\nu+\delta) \end{bmatrix}.$$

The Markov chain equivalent to R_1 can be stated as:

$$\mathrm{Q}' = \begin{bmatrix} \mathrm{T} & \mathrm{t} \\ 0 & 0 \end{bmatrix},$$

where $\mathbf{t} = -\mathbf{T}\mathbf{1}$. Here, $\mathbf{1}$ is a $\beta \times 1$ vector with every element being 1. In fact, R_1 follows a $PH(\tau,\mathbf{T})$ distribution with $\boldsymbol{\tau}_{1\times\beta} = [1,0,\ldots,0]$ and fundamental matrix of $\mathbf{T}$. The absorption time of a Markov chain follows a phase type distribution; therefore, the availability function of the system can be calculated as:

$$A(x) = 1 - F(x) = \boldsymbol{\tau}\exp(\mathbf{T}x)\mathbf{1}. \tag{3}$$

R_2 is the repair time of the system and follows an exponential distribution. Since exponential distribution is a special case of a phase-type family, R_2 follows a $PH(\kappa,K)$ with parameters $K = -\omega$ and $\kappa = 1$. Therefore, the Markov chain of R_2 can be indicated as:

$$\tilde{\mathbf{Q}} = \begin{bmatrix} K & k \\ 0 & 0 \end{bmatrix},$$

where $k = \omega$. The time between two successive failures of the system (shown as R) can be yielded by adding two independent random variables of R_1 and R_2. Using the following property of phase-type distributions [27], the distribution function of R is found.

Property 1: X and Y are two independent continuous phase-type random variables with n and m phases, respectively. X is denoted by $PH(\boldsymbol{\theta},\mathbf{V})$ with initial probability vector $(\boldsymbol{\theta},\theta_{n+1})$ and Y is represented by $PH(\boldsymbol{\zeta},\mathbf{S})$ with initial probability vector (ζ,ζ_{m+1}). The random variable $X+Y$ is a $PH(\boldsymbol{\gamma},\mathbf{C})$ with $n+m$ phases in which $\gamma = (\boldsymbol{\theta},\theta_{n+1}\zeta)$ and

$$\begin{bmatrix} \mathbf{C} & \mathbf{c} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{V} & \nu\boldsymbol{\zeta} & \zeta_{m+1}\nu \\ 0 & \mathbf{S} & \mathbf{s} \\ & 0 & 0 \end{bmatrix}.$$ ■

According to Property 1, $R = R_1 + R_2$ follows a phase-type distribution as $R \sim PH(\mathbf{r},\mathbf{P})$ where, $\mathbf{r} = (\tau,0)$ and

$$\begin{bmatrix} \mathbf{P} & \mathbf{p} \\ \mathbf{0} & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{T} & \mathbf{t} & 0 \\ 0 & -\omega & \omega \\ & \mathbf{0} & 0 \end{bmatrix}.$$

Accordingly, $E(R)$ which is the expected value of time between two successive failures of the system is found as:

$$E(R) = E(R_1 + R_2) = E(R_1) + E(R_2) = -\boldsymbol{\tau}\mathbf{T}^{-1}\mathbf{1} + \omega^{-1}.$$

The average number of customers in the system is given by:

$$E(N) = \boldsymbol{\pi}\mathbf{S}(\mathbf{I}-\mathbf{S})^{-1}\mathbf{e}. \tag{4}$$

Also, the expected value of operational servers in the system can be obtained as:

$$E(\beta) = \boldsymbol{\pi}\mathbf{B}, \tag{5}$$

where $\mathbf{B}$ is defined as $\boldsymbol{B}_{\beta+1\times1} = [\beta,\beta-1,\ldots,1,0]'$. One can find the steady-state availability of the system according to Eq. (6):

$$\rho = 1 - \pi_\beta. \tag{6}$$

Steady-state probabilities of **A** in cold standby redundancy can be found by solving the following set of equations:

$$\begin{cases} \pi_{\beta-i} = \sum_{j=1}^{i} \dfrac{\delta^{j-1}\omega}{\nu^j}\pi_\beta & i = 1,2,...,\beta \quad (7) \\ \pi_0 + \pi_1 + ... + \pi_\beta = 1 & (8) \end{cases}$$

Therefore, π_β can be given as:

$$\pi_\beta = \frac{1}{1 + \sum_{i=0}^{\beta-1} \dfrac{(\beta - i)\delta^i \omega}{\nu^{i+1}}}. \tag{9}$$

Eq. (7) can be used to give $\pi_{\beta-i}$ for $i = 1,2,\ldots,\beta$. In the same manner, steady-state probabilities of **A** in warm standby redundancy can be found by solving the following set of equations:

$$\begin{cases} \pi_{\beta-i} = \sum_{j=1}^{i} \dfrac{(i-j)!\omega\delta^{j-1}}{i!\nu^j}\pi_\beta & i = 1,2,...,\beta \quad (10) \\ \pi_0 + \pi_1 + ... + \pi_\beta = 1 & (11) \end{cases}$$

Therefore, π_β can be specified as:

$$\pi_\beta = \frac{1}{\sum_{i=1}^{\beta} \sum_{j=1}^{i} \dfrac{(i-j)!\omega\delta^{j-1}}{i!\nu^j} + 1}. \tag{12}$$

Eq (10) can be used to give $\pi_{\beta-i}$ for $i = 1,2,\ldots,\beta$.

3.3. Cost analysis

The cost function of the system is analyzed in this subsection. The cost function is the summation of holding cost of customers, energy consumption cost, repair and maintenance costs, and system preservation costs. The cost function is minimized to obtain the optimal number of redundant servers, and optimal values of failure and service rate of servers. In the following, the elements of the cost function are defined. The maintenance and stoppage cost, C_M, is stated as:

$$C_M = A_M + C_R\beta, \tag{13}$$

where C_R and A_M are respectively repair cost of a server and fixed cost of system stoppage.

To find the average cost of server repairs per unit time, random variable T_i is defined. T_i is the time between two consecutive repairs of a server when there are i failed servers in the maintenance center. It can be shown that cycle T_i is regenerative with regard to the repair process.

In cold and hot standby redundancies, cycle T_i is found based on the absorption time in a Markov chain with fundamental matrices $\mathbf{H}_i$ and $\mathbf{H}_i'$ respectively. Therefore, the expected value of T_i in cold and hot standby redundancies is respectively acquired according to Eqs. (14) and (15):

$$E[T_i] = -\mathbf{e}_1 \mathbf{H}_i^{-1}\mathbf{e}, \tag{14}$$

$$E[T_i] = -\mathbf{e}_1 H_i'^{-1}\mathbf{e}, \tag{15}$$

where $e_1 = [1,0,\ldots,0]$, $\boldsymbol{e} = [1,1,\ldots,1]$ and matrices $\mathbf{H}_i$ and $\mathbf{H}_i'$ are defined as:

$$\mathbf{H}_i = \begin{array}{c} 0 \\ 1 \\ \vdots \\ i \\ \vdots \\ \beta-1 \\ \beta \end{array} \begin{bmatrix} -\nu & \nu & & & & & & \\ \delta & -(\nu+\delta) & \nu & & & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & 0 & -(\nu+\delta) & \nu & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & \delta & -(\nu+\delta) & \nu & \\ \omega & & & & & & & -\omega \end{bmatrix},$$

$$\mathbf{H}_i' = \begin{array}{c} 0 \\ 1 \\ \vdots \\ i \\ \vdots \\ \beta-1 \\ \beta \end{array} \begin{bmatrix} -\beta\nu & \beta\nu & & & & & & \\ \delta & -((\beta-1)\nu+\delta) & (\beta-1)\nu & & & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & 0 & -((\beta-i)\nu+\delta) & (\beta-i)\nu & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & \delta & -(\nu+\delta) & \nu & \\ \omega & & & & & & & -\omega \end{bmatrix}.$$

In each cycle T_i, a repair cost C_R is incurred. Therefore, based on the elementary renewal theorem [28], the average repair cost per unit time is given by:

$$C_R / E[T_i]. \tag{16}$$

There exists $\beta - 1$ states for i. Hence, the average repair cost of the system per unit time can be written as:

$$\sum_{i=1}^{\beta-1} \frac{\pi_i C_R}{E[T_i]}. \tag{17}$$

Similarly, the average maintenance cost of the system per unit time can be found as:

$$C_M / E[R]. \tag{18}$$

Likewise, based on the renewal reward theorem [28], the average customer's holding cost can be obtained as $C_H E(N)$, in which C_H is the holding cost per customer per unit time. Finally, it is assumed that unit time average preservation and energy consumption cost functions of the system are respectively $C_P E(\beta)$ and $C_{EC}(\mu,\beta)$, where C_P is the preservation cost of the system per unit time. $TC(\beta,\upsilon,\mu)$ represents the total cost per unit time of the system.

$$\begin{aligned} \mathrm{TC}(\beta,\nu,\mu) = {} & C_H E(N) + \sum_{i=1}^{\beta-1} \frac{\pi_i C_R}{E[T_i]} + \frac{C_M}{E[R]} \\ & + C_P E(\beta) + C_{EC}(\mu,\beta), \end{aligned} \tag{9}$$

where redundancy level of the system is β, and failure and

service rates of servers are ν and μ, respectively. The optimal redundancy level and values of failure and service rates are obtained such that the total cost function is minimized in both cases of cold and hot standby redundancies.

3.4. RRAP

Generally, RRAP is formulated either to maximize system reliability or minimize system costs. In this study, the average total cost of the system per unit time is to be minimized by determining the number of redundant servers as well as servers' failure and service rates such that two constraints are satisfied: (1) The total cost of the servers in the system should not exceed a given budgetary level BD; and (2) The total system weight should not exceed a given upper value W.

$$\text{Min TC}(\beta,\nu,\mu), \tag{20}$$

s.t.,

$$SC(\beta,\nu) \le BD, \tag{21}$$

$$SW(\beta) \le W, \tag{22}$$

$$\beta \in \mathbb{N},\ \nu,\mu \in \mathbb{R}_{\ge 0}. \tag{23}$$

The objective function of the problem is defined as Eq. (20). According to this relation, unit time costs of the system are summed. Inequalities (21) and (22) refer to the system's available budget and maximum allowed weight, respectively. In the abovementioned model, $SC(\beta,\nu)$ and $SW(\beta)$ are total cost of servers (procurement cost) and total weight of the system, respectively. Correspondingly, BD, and $W$are available budget and maximum allowed weight of the system, respectively. $\mathbb{N}$ and $\mathbb{R}_{\ge 0}$ denote the natural number set and the set of nonnegative real numbers, respectively.

4. Solution methodology

This section presents a GA-based MA for solving the RRAP model. Also, a procedure based on the Sequential Quadratic Programming (SQP) algorithm, as an exact method for convex optimization problems, is proposed to validate the main solution approach.

4.1. GA

The structural elements of the proposed GA are explained as the following. The fitness function is the total cost Function (19). Therefore, the best chromosomes transferred to the next generation should have the lowest fitness function values. A successful implementation of GA depends on how chromosomes are encoded. In this matter, a simple encoding scheme is considered as $X = (\beta,\nu,\mu)$. Two-parent solutions are taken in the crossover process, and a child is produced from them. Based on this operation, parents are recombined to yield new offspring. The proposed crossover operation is performed in four steps:

(i). A pair of chromosomes are randomly selected, shown as $p_1 = (p_{11}, p_{12}, p_{13})$ and $p_2 = (p_{21}, p_{22}, p_{23})$, for the mating.

(ii). The first cell of the parents p_{11} and p_{21}, along the string are swapped.

(iii). The remaining cells of the parents are merged to produce new offsprings. In this regard, a single offspring variable value, p_{newi} for $i = 2,3$, is the combination of parents as:

$$p^1_{newi} = p_{1i} - \alpha\left(p_{1i} - p_{2i}\right),$$

$$p^2_{newi} = p_{1i} + \alpha\left(p_{1i} - p_{2i}\right),$$

where $\alpha \in [0,1]$ is a random number.

(iv). The new offsprings are

$$\text{offspring}\,1 = \left(p_{21}, p^1_{new2}, p^1_{new3}\right),$$

$$\text{offspring}\,2 = \left(p_{11}, p^2_{new2}, p^2_{new3}\right).$$

To produce off springs, different random numbers can also be used. In this process, crossover probability, p_c, determines the number of offsprings that are produced based on each population in each iteration of GA. In other words, $p_c \cdot N_P$ offsprings are produced in each generation of the algorithm [29]. By mutation, the structure of the chromosome is randomly modified. To that end, each chromosome is usually mutated with a small probability, p_m, which is called mutation probability [30]. In this research, random cells of chosen chromosomes are changed randomly.

4.2. MA

In this subsection, a MA based on the GA and diffusion process of the SFS algorithm [31] is proposed. Juybari et al. [22] successfully applied an SFS algorithm to the RRAP. Also, Dobani et al. [23] formulated a new RRAP and solved it with an SFS-GA hybrid algorithm. They showed that their proposed algorithm outperforms some of the previously implemented algorithms. In this regard, we utilize a MA that incorporates GA and diffusion process operator of the SFS algorithm to solve the proposed RRAP.

In the diffusion process, points are diffused around their current positions to generate similar structures. A Gaussian walk based on Eq. (24) is used for the diffusion process:

$$GW = \psi + \left(\varepsilon \cdot X_{BP} - \varepsilon' \cdot X_i\right), \tag{24}$$

where ψ is a random number following $N(\mu_{BP},\sigma)$, ε and ε' are random numbers following a uniform distribution between 0 and 1. Also, X_{BP} and X_i represent the positions of the best point and the ith point in the group, respectively. In addition, μ_{BP} is the absolute value of X_{BP}. Furthermore, σ denotes the standard deviation of the normal distribution, which is given as:

```
Pseudo code
Input: {N_G, N_P, N_D, p_m and p_c}
For i=1 to N_P
    Generate a random solution X_i
    Evaluate the random solutions based on fitness function in Equation (19),
End for
Find the best point, X_BP, from the population based on their fitness values
For g=1 to N_G
    For i=1 to N_P
        For j=1 to N_D
            Based on Equation (24) generate a new point, X_{i,j}^{DF}, from X_i,
            If TC(X_i^{DF}) < TC(X_i)
                Set X_i = X_{i,j}^{DF}
            End if
        End for
    End for
    Generate P_c.N_P new solutions by performing crossover
    Generate P_c.N_P new solutions by performing mutation
    Sort all solutions (i.e., current solutions as well as new solutions) based on the fitness
      function in Equation (19)
    Select the best N_P solutions and discard others
    Find the best solution, X_BP, from the population
    For i=1 to N_P
        For j=1 to N_D
            Based on Equation (24) generate a new point, X_{i,j}^{DF}, from X_i
            If TC(X_{i,j}^{DF}) < TC(X_i)
                Set X_i = X_{i,j}^{DF}
            End if
        End for
    End for
    Sort all solutions based on the fitness function in Equation (19)
    Find the best solution, X_BP, from the population
End for
Output: {the best solution X_BP}.
```

Figure 1. Pseudo code of the MA.

$$\sigma = \left| \frac{\log(g)}{g} \times (X_i - X_{BP}) \right|, \tag{25}$$

where g is the iteration number. The term $\frac{\log(g)}{g}$ is used for a more localized search as the number of iterations increases and the algorithm gets closer to the final solution. Therefore, the neighborhood of point X_i is searched more intensively in higher iterations [31]. We use the diffusion process to enhance the local search capability of GA. The parameters used for the MA are listed in Table 2.

Also, the pseudo-code of the proposed MA is illustrated in Figure 1.

4.3. SQP-based algorithm

In this subsection, an SQP-based algorithm is used to validate the results of the proposed metaheuristics. The SQP algorithm is an iterative method that can be used for solving constrained nonlinear optimization problems. In this algorithm, the objective function and constraints should be twice continuously differentiable. The SQP algorithm can find the global optimum in convex optimization problems. However, the objective function of our problem is very complicated and its convexity is not clear.

Table 2. MA parameters.

Parameter	Description
N_G	Number of generations
N_P	Population number
N_D	Maximum number of diffusions
p_m	Mutation probability
p_c	Crossover probability

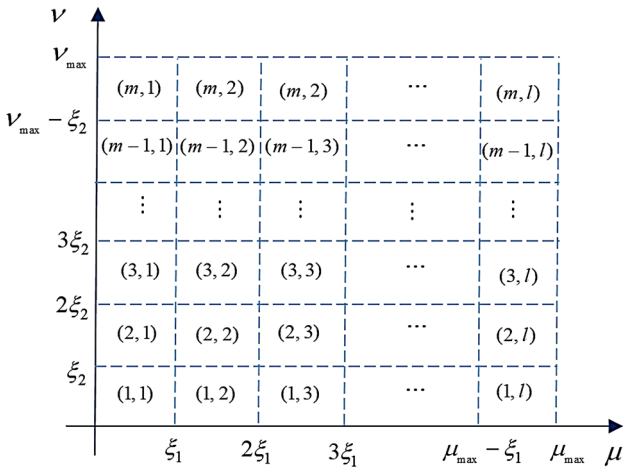

Figure 2. Dividing the solution space to grids.

This objective function is formed based on the multiplication of several matrices, in which the dimensions of the matrices are dependent on the value of variable β. Also, taking the definite derivative of the objective function is very hard. Therefore, using traditional exact optimization methods for solving this problem is not straightforward. In this regard, to optimize the objective function, we can fully enumerate the possible values for the integral variable β. Since the original upper bound of the variable β is infinity, its full enumeration is not possible. Therefore, a maximum value for this variable is assumed as $\beta_{\max}$. Then, the SQP method is used to find the optimal values of υ and μ. Given an integral value of $\beta \in \{0,\ldots,\beta_{\max}\}$, we use the numerical differentiation of the objective function and constraints to implement the SQP algorithm. To do so, we divide the solution space of variables ν and μ to several small grids and implement the SQP method in each grid (see Figure 2). This approach breaks the solution space of the original problem into several small grids (i.e., sub-problems), where the SQP method is supposed to be able to find the global optimum. It should be noted that the search space of each sub-problem/grid is very small and hence, can be assumed to be an almost convex space. In other word, if the grids are small enough, we can be sure that the local optimum that is found by the SQP method is not too far from the global optimum of a grid.

Since μ and ν are unbounded positive variables, a reasonable maximum value is assumed for each of them so that the number of grids becomes finite. In Figure 2, $\mu_{\max}$ and ν_{max} are maximum values of variables μ and ν. Also, ξ_1 and ξ_2 are minimum length and width of a grid, respectively. We assign small values to ξ_1 and ξ_2 so that $\mu_{\max} \bmod \xi_1 = 0$ and $\nu_{\max} \bmod \xi_2 = 0$. Accordingly, l and m are integer values, where $l \times m$ determines the number of grids. In Figure 2, the ordered pair (i,j) for $i \in \{1,\ldots,m\}$ and $j \in \{1,\ldots,l\}$ is used to characterize the grids. Thus, the algorithm if Figure 3 is proposed to solve the problem.

Pseudo code

Input: $\{\beta_{\max}, \mu_{\max}, \nu_{\max}, l, \text{and } m\}$
Set $TC = TC^W = \infty$
For $\beta = 1$ to $\beta_{\max}$
  For $i = 1,\ldots,m$
    For $j = 1,\ldots,l$
      Use the SQP algorithm to find the optimum values of the grid (i,j), μ_{ij}^* and ν_{ij}^*,
      If $TC(\beta,\nu_{ij}^*,\mu_{ij}^*) < TC$
        $TC = TC(\beta,\nu_{ij}^*,\mu_{ij}^*)$
        $\nu = \nu_{ij}^*$, $\mu = \mu_{ij}^*$
      End if
    End for
  End for
  If $TC < TC^W$
    $TC^W = TC$
    $\nu^* = \nu$, $\mu^* = \mu$
  End if
End for
Output: $\{\nu^*, \mu^* \text{ and } TC^W\}$

Figure 3. Pseudo code of the SQP based algorithm.

In the SQP algorithm, we use the center point of the grid as the initial point. For instance, the initial point for grid (1,1) is considered as $(\frac{\xi_1}{2},\frac{\xi_2}{2})$. Also, the SQP solver in MATLAB R2018b is utilized to implement the SQP algorithm.

5. Numerical experiments

In this section, the proposed approach is applied to a sample problem. Then, the mathematical model of the problem is formulated. Afterward, the computational results and sensitivity analysis for the proposed problem and solution approach are presented. In this regard, the RRAP model and the proposed solution approach are illustrated through the reliability-redundancy allocation of a workstation in a job-shop plant, because it is the bottleneck of the plant. It is supposed that this workstation gives service to incoming Work-In-Process (WIP) parts. The optimal redundancy and failure rate of the components of this workstation and its service rate is to be determined through an RRAP model.

5.1. Mathematical model of the RRAP

In this problem, the objective function is the sum of unit time costs of the system. Also, there are constraints on the available budget and weight of the system. It is supposed that the available budget and maximum allowed weight of the system are 300 and 50, respectively. Moreover, the arrival rate of customers is $\lambda = 2.5$. The servers and the system are repaired with rates $\delta = 0.8$ and $\omega = 0.05$, respectively. Also, customers' holding cost per unit time, fixed repair cost of a server, and preservation cost of the system are respectively $C_H = 50$, $C_R = 100$, and $C_P = 5$. Additionally, it is assumed that maintenance and cost function of the system is $C_M = 1000 + 100\beta$. Energy consumption cost function is defined as $C_{EC}(\mu,\beta) = 100\beta^x + 20\mu$ where $x = 0$ stands for the cold standby redundancy and $x = 1$ stands for the hot standby redundancy. It is also supposed that the cost of a system with β servers and failure rate ν is $SC(\beta,\nu) = \gamma\nu^{-\theta}\left(\beta + exp\left(\frac{\beta}{4}\right)\right)$. Similarly, the weight of a

system with β servers is $SW(\beta) = w\left(\beta + exp\left(\frac{\beta}{4}\right)\right)$. The parameters γ and θ are the physical feature (shaping and scaling factor) of the cost-reliability curve of each server. Parameter w is the weight of each server. The factor $exp\left(\frac{\beta}{4}\right)$ accounts for the additional cost due to the interconnection between the parallel components [32]. For this example, these parameters are set as $\gamma = 5 \times 10^{-5}$, $\theta = 4.5$, and $w = 5$.

5.2. Computational results

In this section, the problem is solved with the proposed metaheuristics and the SQP based algorithm. Therefore, in Subsection 5.2.1, the parameter setting of the algorithms is defined. Then, in Subsection 5.2.2, the results of the problem are provided. In this regard, MATLAB R2018b installed on a Laptop with a Core i7 CPU@2.50 GHz and 8 GB RAM is used to solve the problem.

5.2.1. Parameter setting

Here, the parameters of the MA, GA, and SQP-based algorithms are set. For the MA and GA, the best combination of the parameters is set by trial and error. Also, the algorithm is executed several times for each set of parameters and the best solution is reported. However, the results show that the variation of results among several proposed algorithm runs is insignificant. The maximum number of iterations and population size are respectively set to 300 and 100. The mutation probability is set to 0.15. Constraints are added to the fitness function as penalties. For the SQP based algorithm, β_{max}, μ_{max}, and ν_{max} are set to 10, 10, and 2, respectively. Also, a sensitivity analysis is performed on the parameters l and m.

5.2.2. Results

To evaluate the performance of the proposed metaheuristics and the SQP-based algorithm, 10 test problems are randomly generated as shown in Table 3. In this table, C_M and C_{EC} in Subsection 5.1 are respectively multiplied by factors F_M and F_{EC}. To perform the SQP-based algorithm, we should decide on the number of grids. A small number of grids can result in local optimums. On the other hand, dividing the solution space to a large number of grids can increase the chance of convergence to the global optimum, and accordingly increase the time to find the optimum solution significantly. Therefore, a tradeoff between the solution quality and time to implement the algorithm should be taken into account. To do so, we change the number of grids and investigate the solution quality and run time for 3 test problems {1, 5, and 10} from Table 3. In this regard, we implement the SQP-based algorithm on the mentioned test problems for $(l, m) = (10,20)$, $(50,100)$, $(100,200)$ and $(150,300)$. The results are presented in Figure 4. It can be observed that the optimal average total cost, for both cold and hot standby redundancies, converge at $(l, m) = (100,200)$. This means that dividing the solution space to grids based on $(l, m) > (100, 200)$ does not necessarily improve the total cost, but may increase the run time significantly. Therefore, we set $(l, m) = (100,200)$ to implement the SQP-based algorithm.

The proposed metaheuristics and the SQP-based algorithm are implemented on the test problems and the results are presented in Table 3. In this table, the average run time of the MA and GA for 5 implementations of these algorithms, and 1 implementation of the SQP-based algorithm on cold and hot standby redundancies, is reported. It can be observed that the MA and SQP-based algorithm result in the same solution for different combination of parameters. Also, results of the MA and SQP-based algorithm mostly outperform the GA. However, our experiments represent that the time to perform these algorithms are not comparable. In more details, implementation of the SQP-based algorithm is very time-consuming compared to the other two algorithms. On the other hand, time to implement the GA is to some extent less than the MA. However, the results show that the GA cannot find the best solution in some cases. Therefore, it can be concluded that the MA outperforms the other two algorithms. In the following, the MA is used to solve the problem described in Subsection 5.1 in more details.

The outputs of the MA for the given problems are summarized in Tables 4 and 5. The optimal number and failure and service rates of servers for cold and hot standby redundancies are represented in Table 4. In this table, unit time system costs and the weight and procurement cost of the system are reported. By reducing the unit time costs of the system, availability of the system is increased. Therefore, it can be observed that the repair costs are insignificant compared to the other costs. Queueing performance measures of the system are also presented in Table 5. The steady-state availabilities of cold and hot standby redundancies are 0.9998 and 0.9868, respectively. Due to higher energy consumption costs in hot standby redundancy, fewer redundant servers are allocated to the system. Therefore, the average number of customers and the average number of safe (i.e., unfailed) servers in the system are higher and less in hot standby redundancy, respectively. Also, the system's total cost with hot standby redundancy is higher than the cold standby redundancy. Generally, when the server activation switch in cold standby redundancy is perfect, the cold standby strategy outperforms the hot standby strategy. However, due to some limitations, the cold standby strategy cannot be considered for all cases.

The total cost is plotted in Figures 5 and 6 versus decision variables for both cold and hot standby redundancies. In each figure, one variable is set to its optimal value, and other variables are changed. These figures show the relation between the total cost and decision variables. Generally, augmentation. In more details, the following observations

Table 3. Comparison between the MA, GA, and SQP-based algorithm.

No.	Parameters										Standby strategy								Algorithm	Average run time (s)
											Cold				Hot					
	λ	δ	ω	C_P	C_H	C_R	F_M	F_{EC}	C	W	β	ν	μ	TC	β	ν	μ	TC		
1	1.25	0.45	0.01	10	20	20	20	1.8	150	50	5	0.066	2.028	349.33	4	0.059	4.724	1436.74	GA	163.5
											5	0.058	2.149	341.75	4	0.056	4.817	1360.92	MA	209.3
											5	0.058	2.149	341.75	4	0.056	4.817	1360.92	SQP-based	4325.3
2	2.5	0.55	0.08	2	10	60	30	1.6	250	40	4	0.051	3.186	308.33	4	0.049	2.337	355.74	GA	158.6
											4	0.049	3.390	306.34	4	0.049	2.337	355.74	MA	214.8
											4	0.049	3.390	306.34	4	0.049	2.337	355.74	SQP-based	4125.7
3	1.5	0.5	0.04	1	40	20	70	1.2	50	15	1	0.057	15.42	21965.04	1	0.056	16.812	21953.32	GA	174.6
											1	0.055	15.284	21908.07	1	0.055	15.284	21908.07	MA	211.3
											1	0.055	15.284	21908.07	1	0.055	15.284	21908.07	SQP-based	4412.6
4	2	0.1	0.09	1	45	10	10	0.8	75	55	6	0.074	4.931	261.11	2	0.056	8.541	830.13	GA	168.4
											6	0.072	4.865	253.28	3	0.061	7.669	813.97	MA	205.9
											6	0.072	4.865	253.28	3	0.061	7.669	813.97	SQP-based	4236.1
5	1	0.45	0.03	3	15	100	20	1.2	100	45	5	0.064	1.799	198.45	3	0.057	2.970	612.67	GA	172.8
											5	0.064	1.799	198.45	3	0.057	2.825	612.41	MA	216.7
											5	0.064	1.799	198.45	3	0.057	2.825	612.41	SQP-based	4681.8
6	0.75	0.15	0.07	5	50	10	30	1.2	225	50	4	0.051	2.099	264.81	3	0.049	2.763	875.35	GA	159.4
											5	0.053	2.041	241.58	3	0.048	2.745	856.14	MA	223.2
											5	0.053	2.041	241.58	3	0.048	2.745	856.14	SQP-based	4005.4
7	2.75	0.4	0.09	6	30	60	90	0.4	50	20	2	0.062	8.996	1014.84	2	0.062	63.393	2088.43	GA	177.5
											2	0.062	8.996	1014.84	2	0.062	10.648	1718.26	MA	227.4
											2	0.062	8.996	1014.84	2	0.062	10.648	1718.26	SQP-based	4124.9
8	0.5	0.35	0.02	8	50	60	40	1.4	275	45	5	0.051	1.456	248.52	3	0.046	2.568	857.53	GA	165.5
											4	0.048	1.505	252.43	4	0.048	2.124	853.61	MA	206.2
											4	0.048	1.505	252.43	4	0.048	2.124	853.61	SQP-based	4469.7
9	1.75	0.2	0.07	1	45	80	70	1.4	150	45	5	0.062	3.396	328.29	3	0.052	5.159	1439.32	GA	170.3
											5	0.058	3.498	318.23	3	0.052	5.159	1439.32	MA	219.1
											5	0.058	3.498	318.23	3	0.052	5.159	1439.32	SQP-based	4706.2
10	2.5	0.2	0.1	3	10	50	30	2	175	45	4	0.054	3.360	404.74	2	0.047	4.356	1118.15	GA	167.5
											5	0.056	3.314	388.11	2	0.046	4.452	1111.96	MA	212.5
											5	0.056	3.314	388.11	2	0.046	4.452	1111.96	SQP-based	4923.4

Table 4. Optimal results of the problem.

Redundancy type	β	ν	μ	Costs					Servers cost	System weight
				Customers' holding	Server repairs	System repair	Preservation	Energy consumption		
Cold	4	0.0476	5.012	50.671	0.252	0.0125	19.680	200.237	300	33.6
Hot	3	0.045	5.776	97.869	1.564	0.8701	13.958	418.513	300	25.6

Table 5. Queueing performance measures of the problem.

Redundancy type	π_0	π_1	π_2	π_3	π_4	$E(\beta)$	ρ	$E(N)$
Cold	0.9404	0.0559	0.0033	0.0002	0.0002	3.9361	0.9998	1.0133
Hot	0.8329	0.1391	0.0147	0.0132	-	2.7917	0.9868	1.9574

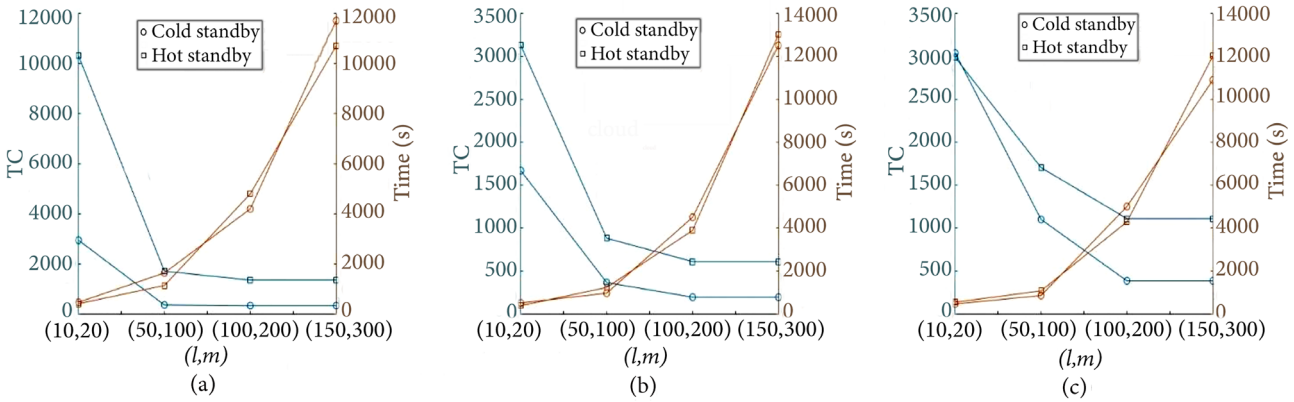

Figure 4. Solution and run time of the SQP based algorithm for test problems; (a) 1, (b) 5, and (c) 10.

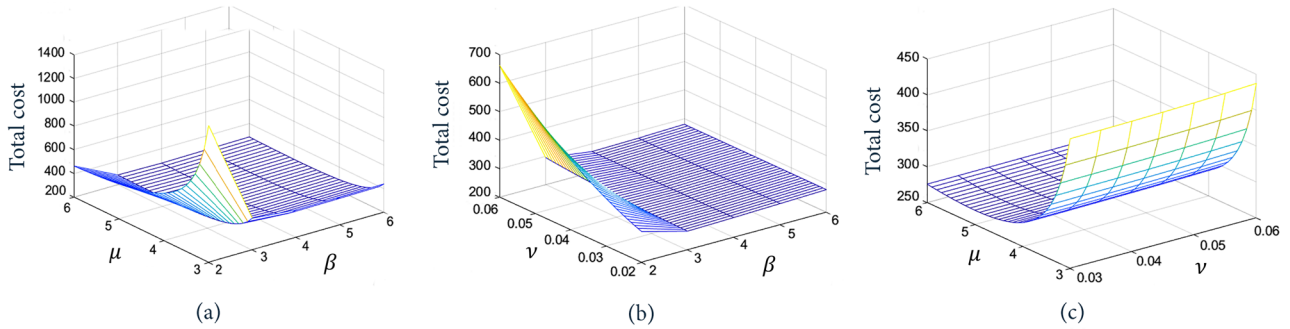

Figure 5. Total cost of the cold standby versus; (a) μ and β for $\nu = 0.0476$; (b) ν and β for $\mu = 5.012$; and (c) ν and μ for $\beta = 4$.

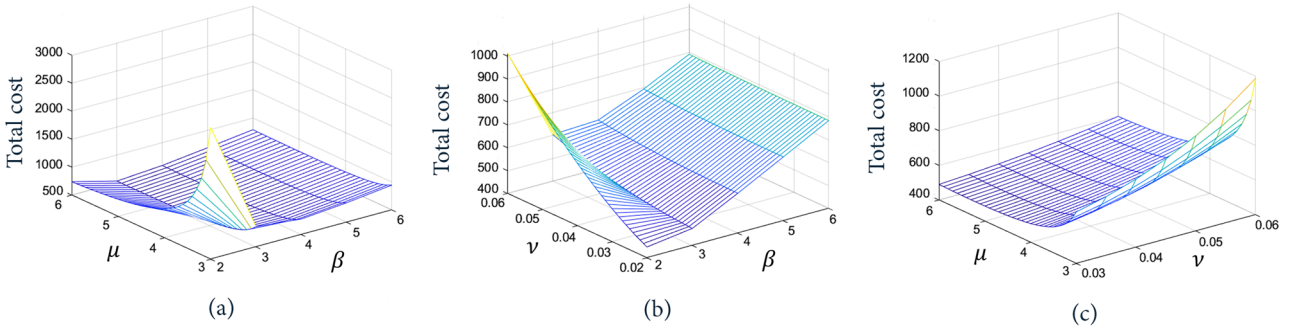

Figure 6. Total cost of the hot standby versus: (a) μ and β for $\nu = 0.045$; (b) ν and β for $\mu = 5.776$; and (c) ν and μ for $\beta = 3$.

can be listed for both types of standby redundancy:

- The total cost increases when β and μ decrease and ν is fixed (Figures 5(a) and 6(a)),
- The total cost increases when ν increases, β decreases, and μ is fixed. Also, for hot standby redundancy, the total cost decreases when ν and β decrease, and μ is fixed (Figures 5(b) and 6(b)),
- The total cost increases when ν increases, μ decreases, and β is fixed (Figures 5(c) and 6(c)).

Figures 5 and 6 imply that when a variable is fixed, the solution space of other variables is apparently convex. However, our experiments indicate that the solution space of the problem is not convex and traditional convex optimization methods cannot converge to the global optimum. Therefore, the proposed metaheuristics and an SQP-based algorithm are used to solve the non-convex problem.

5.3. Sensitivity analysis

In this subsection, a sensitivity analysis on the parameters of the problem is performed. The results of the sensitivity analysis are obtained by changing the values of λ, δ, ω, C_P, C_H, C_R, F_M, F_{EC}, W and C. For the sensitivity analysis and in order to analyze the results with regard to change in C_M and C_{EC}, these costs are respectively multiplied by factors F_M and F_{EC}. Tables S1 to S5 in the Supplementary file, represent the results of sensitivity analysis. According to these tables, the following results can be observed:

- **Observation 1:** Increase in λ mostly results in service rate increase in both cases of cold and hot standby redundancies. Due to budget and weight constraints, number of redundant servers and their failure rates do not change for higher values of λ.
- **Observation 2:** Increase in δ impact β and ν in cold standby redundancy and μ in hot standby redundancy. In other words, lower repair rates are respectively compensated with higher number of redundant servers and higher service rates for cold and hot standby redundancies.
- **Observation 3:** Increase in ω results in β and ν decrease in both cold and hot standby redundancies and μ decrease in hot standby redundancy.
- **Observation 4:** Increase in C_P only impacts cold standby redundancy by reducing β and ν and increasing μ. In hot stand by redundancy, preservation cost is insignificant in comparison to energy consumption and customers' holding costs.
- **Observation 5:** Increasing C_H mostly results in service rate increase.
- **Observation 6:** Increasing C_R does not affect optimal solutions. Repair costs are insignificant in comparison to other costs of the system.
- **Observation 7:** Increasing F_M does not impact cold standby redundancy but increases β and ν in hot standby redundancy.
- **Observation 8:** Increasing F_{EC} mostly leads to service rate decrease in both cold and hot standby redundancies. However, in hot standby redundancy for some cases server failure rate is reduced to compensate the growth of energy consumption cost.
- **Observation 9:** Increasing C results in failure rate and service rate decrease. Number of redundant servers is

not sensitive to C. On the other hand, procurement cost of each server increases by reducing server failure rates.

- **Observation 10:** Results are more sensitive to W when it is compared to C. Weight constraint is only dependent on β. Therefore, for lower values of W fewer number of redundant servers should also be used. In this regard, failure and service rates should change to increase the availability of the system.

Generally, when the activation server switch in cold standby redundancy is perfect, this redundancy strategy outperforms the hot standby redundancy. Due to higher preservation, repair and energy consumption costs in hot standby strategy, it is more sensitive to parameter variations. Also, the results show that the queueing parameters like λ, μ, and C_H can impact reliability and redundancy allocations. Thus, in reliability and redundancy allocation of a queueing system, performance measures of these systems should also be considered.

6. Conclusion

Reliability-Redundancy Allocation Problem (RRAP) in a queueing system is investigated in this study. A Memetic Algorithm (MA) is used to find the system's optimal redundancy level and failure rates of components and service rate. It is supposed that higher service rates increase energy consumption level of the system. Optimal value of the decision variables is obtained so that the unit time cost of the system is minimized. The cost function comprises queueing, repair, and energy consumption costs. Results show that the queueing parameters can impact the reliability and redundancy allocations. In other words, the reliability of queueing systems not only affects the failure and maintenance costs, but also impacts the queueing measures of the system. Therefore, these parameters should be considered when the reliability of a queueing system is evaluated. The proposed model is advantageous because it can incorporate all the queueing and reliability costs of the system.

It is shown that the redundancy level of a queueing system can directly affect the queueing costs of the system. Therefore, reliability and redundancy allocation of queueing systems should be studied with regard to queueing measures of these systems. The proposed model of RRAP can determine the optimal service rate of the system that supposedly impacts the energy consumption level of the system. The experiments show that the solution space is not convex, therefore, a MA is developed to find the optimal solution(s). Although the objective function is complex, but the proposed algorithm is able to find the optimal or near-optimal solutions for a reasonable computational cost.

In this research, we analyzed a single queueing server. However, queueing systems are sometimes a network of single servers that are connected in different configurations. Tandem queueing network is the simplest queueing network that is used in many applications, e.g., assembly lines. RRAP of this system that also considers queueing costs can be a topic for future research. To study more complex queueing networks, simulation-based optimization approaches can be implemented. Also, RRAP of queueing systems can be investigated under other assumptions like the choice of redundancy strategy or uncertain parameters. Finally, the RRAP of queueing systems which benefit other types of manufacturing paradigms, such as Reconfigurable Manufacturing System (RMS) [33], can be studied as another topic for future researches.

Acknowledgments

The authors are thankful to the review team of the journal for providing constructive comments on an earlier version of this paper. Also, Dr. Mehdi Karimi-Nasab helped the authors in different aspects, including the idea of the SQP-based algorithm for searching for the optimal policy in an exact manner.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for profit sectors.

Conflicts of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Authors contribution statement

First author: Conceptualization; Investigation; Methodology; Software; Visualization; Formal analysis; Writing-original draft

Second author: Validation; Conceptualization: Supervision; Writing-Review and Editing

References

1. Teimouri, M., Zaretalab, A., Niaki, S.T.A., et al. "An efficient memory-based electromagnetism-like mechanism for the redundancy allocation problem", *Applied Soft Computing*, **38**, pp. 423-436 (2016). https://doi.org/10.1016/j.asoc.2015.09.043

2. Sharifi, M., Saadvandi, M., and Shahriari, M.R. "Presenting a series-parallel redundancy allocation problem with multi-state components using recursive algorithm and meta-heuristic", *Scientia Iranica*, **27**(2), pp. 970-982 (2020). https://doi.org/10.24200/sci.2018.51176.2040

3. Yeh, W.-C. and. Hsieh, T.-J "Solving reliability redundancy allocation problems using an artificial bee colony algorithm", *Computers & Operations Research*, **38**(11), pp. 1465-1473 (2011). https://doi.org/10.1016/j.cor.2010.10.028

4. Kanagaraj, G., Ponnambalam, S., and Jawahar, N. "A hybrid cuckoo search and genetic algorithm for reliability-redundancy allocation problems", *Computers & Industrial Engineering*, **66**(4), pp. 1115-1124 (2013). https://doi.org/10.1016/j.cie.2013.08.003

5. Sabri-Laghaie, K., Eshkevari, M., Fathi, M., et al. "Redundancy allocation problem in a bridge system with dependent subsystems", *Proceedings of the*

Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, **233**(4), pp. 658-669 (2019). https://doi.org/10.1177/1748006X18814627

6. Garg, H. and Sharma, S. "Multi-objective reliability-redundancy allocation problem using particle swarm optimization", *Computers & Industrial Engineering*, **64**(1), pp. 247-255 (2013). https://doi.org/10.1016/j.cie.2012.09.015

7. Sabri-Laghaie, K. and Karimi-Nasab, M. "Random search algorithms for redundancy allocation problem of a queuing system with maintenance considerations", *Reliability Engineering & System Safety*, **185**, pp. 144-162 (2019). https://doi.org/10.1016/j.ress.2018.12.010

8. Najafi, A.A., Karimi, H., Chambari, A., et al. "Two metaheuristics for solving the reliability redundancy allocation problem to maximize mean time to failure of a series-parallel system", *Scientia Iranica*, **20**(3), pp. 832-838 (2013). https://doi.org/10.1016/j.scient.2012.12.022

9. Roy, P., Mahapatra, B., Mahapatra, G., et al. "Entropy based region reducing genetic algorithm for reliability redundancy allocation in interval environment", *Expert Systems with Applications*, **41**(14), pp. 6147-6160 (2014). https://doi.org/10.1016/j.eswa.2014.04.016

10. Kim, H. and Kim, P. "Reliability-redundancy allocation problem considering optimal redundancy strategy using parallel genetic algorithm", *Reliability Engineering & System Safety*, **159**, pp. 153-160 (2017). https://doi.org/10.1016/j.ress.2016.10.033

11. Sharifi, M. and Taghipour, S. "Optimizing a redundancy allocation problem with open-circuit and short-circuit failure modes at the component and subsystem levels", *Engineering Optimization*, **53**(6), pp. 1064-1080 (2020). https://doi.org/10.1080/0305215X.2020.1771704

12. Pourkarim Guilani, P., Zaretalab, A., Niaki, S.A., et al. "A bi-objective model to optimize reliability and cost of *k*-out-of-*n* series-parallel systems with tri-state components", *Scientia Iranica*, **24**(3), pp. 1585-1602 (2017). https://doi.org/10.24200/sci.2017.4137

13. Ouyang, Z., Liu, Y., Ruan, S.-J., et al. "An improved particle swarm optimization algorithm for reliability-redundancy allocation problem with mixed redundancy strategy and heterogeneous components", *Reliability Engineering & System Safety*, **181**, pp. 62-74 (2019). https://doi.org/10.1016/j.ress.2018.09.005

14. Huang, C.-L. "A particle-based simplified swarm optimization algorithm for reliability redundancy allocation problems", *Reliability Engineering & System Safety*, **142**, pp. 221-230 (2015). https://doi.org/10.1016/j.ress.2015.06.002

15. Zhang, E. and Chen, Q. "Multi-objective reliability redundancy allocation in an interval environment using particle swarm optimization", *Reliability Engineering & System Safety*, **145**, pp. 83-92 (2016). https://doi.org/10.1016/j.ress.2015.09.008

16. Hsieh, T.-J. "A simple hybrid redundancy strategy accompanied by simplified swarm optimization for the reliability–redundancy allocation problem", *Engineering Optimization*, **54**(3), pp. 369-386 (2021). https://doi.org/10.1080/0305215X.2020.1862826

17. Garg, H., Rani, M., and Sharma, S. "An efficient two phase approach for solving reliability-redundancy allocation problem using artificial bee colony technique", *Computers & Operations Research*, **40**(12), pp. 2961-2969 (2013). https://doi.org/10.1016/j.cor.2013.07.014

18. Hsieh, Y.C. and You, P.S. "An effective immune based two-phase approach for the optimal reliability-redundancy allocation problem", *Applied Mathematics and Computation*, **218**(4), pp. 1297-1307 (2011). https://doi.org/10.1016/j.amc.2011.06.012

19. Garg, H. "An efficient biogeography based optimization algorithm for solving reliability optimization problems", *Swarm and Evolutionary Computation*, **24**, pp. 1-10 (2015). https://doi.org/10.1016/j.swevo.2015.05.001

20. Mousavi, S.M., Alikar, N., and Niaki, S.T.A. "An improved fruit fly optimization algorithm to solve the homogeneous fuzzy series-parallel redundancy allocation problem under discount strategies", *Soft Computing*, **20**, pp. 2281-2307 (2016). https://doi.org/10.1007/s00500-015-1641-5

21. Maksoud, E.Y.A. and Moustafa, M.S. "A semi-Markov decision algorithm for the optimal maintenance of a multistage deteriorating two-unit standby system", *Operational Research*, **9**(2), pp. 167-182 (2009). https://doi.org/10.1007/s12351-008-0022-6

22. Juybari, M.N., Abouei Ardakan, M., and Davari-Ardakani, H. "A penalty-guided fractal search algorithm for reliability-redundancy allocation problems with cold-standby strategy", *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, **233**(5), pp. 775-790 (2019). https://doi.org/10.1177/1748006X19825707

23. Dobani, E.R., Ardakan, M.A., Davari-Ardakani, H., et al. "RRAP-CM: A new reliability-redundancy allocation problem with heterogeneous components", *Reliability Engineering & System Safety*, **191**, 106563 (2019). https://doi.org/10.1016/j.ress.2019.106563

24. Azimi, P., M. Hemmati, and A. Chambari, "Solving the redundancy allocation problem of *k*-out-of-*n* with non-exponential repairable components using optimization via simulation approach", *Scientia Iranica*, **24**(3), pp. 1547-1560 (2017). http://doi.org/10.24200/sci.2017.4134

25. Sharifi, M., Shahriyari, M., Khajepour, A., et al. "Reliability optimization of a *k*-out-of-*n* series-parallel system with warm standby components", *Scientia Iranica*, **29**(6), pp. 3523-3541 (2021). http://doi.org/10.24200/sci.2021.56113.4591

26. Neuts, M.F. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*, Dover Pubns (1981).

27. Buchholz, P., Kriege, J., and Felko, I. *Input Modeling with Phase-type Distributions and Markov Models: Theory and Applications*, Springer (2014).

28. Medhi, J., *Stochastic Models in Queueing Theory*, Elsevier (2002).

29. Gendreau, M. and Potvin, J.-Y. *Handbook of Metaheuristics*, 2nd Ed., Springer (2010).

30. Liu, B. and Liu, B., *Theory and Practice of Uuncertain Programming*, 2nd Ed., Springer (2009).

31. Salimi, H., "Stochastic fractal search: A powerful metaheuristic algorithm", *Knowledge-Based Systems*, **75**, pp. 1-18 (2015). https://doi.org/10.1016/j.knosys.2014.07.025

32. Kuo, W. and. Prasad, V.R "An annotated overview of system-reliability optimization", *IEEE Transactions on Reliability*, **49**(2), pp. 176-187 (2000). https://doi.org/10.1109/24.877336

33. Kazemisaboor, A., Aghaie, A., and Salmanzadeh, H. "A simulation-based optimisation framework for process plan generation in Reconfigurable Manufacturing Systems (RMSs) in an uncertain environment", *International Journal of Production Research*, **60**(7), pp. 2067-2085 (2021). https://doi.org/10.1080/00207543.2021.1883762

Biographies

Kamyar Sabri-Laghaie received his BS degree in Industrial Engineering from the Khajeh Nasir Toosi University of Technology, Tehran, Iran, in 2008, and the MS and PhD degrees in Industrial Engineering from Iran University of Science and Technology, Tehran, Iran, in 2011 and 2015, respectively. He is currently an Assistant Professor of Industrial Engineering at Urmia University of Technology, Urmia, Iran. His research interests include quality engineering, reliability engineering, and data analytics.

Mahdi Fathi received his BS and MS degrees from Amirkabir University of Technology, Tehran, Iran, in 2006 and 2008, and his PhD degree from Iran University of Science and Technology in 2013. He has been a Visiting Scholar at the University of Florida, National Tsing Hua University, and Tecnológico de Monterrey. He is currently an Assistant Professor at the University of North Texas. His research interests include operations research, data science, AI in business, cybersecurity, energy systems, and healthcare. He has held three postdoctoral fellowships in France, Belgium, and the USA. He serves as an Associate Editor for several journals and is editing the Handbook of Smart Energy Systems.