RAIRO-Oper. Res. 60 (2026) 1733–1755
https://doi.org/10.1051/ro/2026038

RAIRO Operations Research
www.rairo-ro.org

A NEW MODEL FOR MULTI-OBJECTIVE REDUNDANCY ALLOCATION PROBLEM WITH NON-IDENTICAL COMPONENTS

Hadi Gholinezhad[1,*] and Ali Zeinal-Hamadani[2]

Abstract. Reliability optimization is a critical area of research, with the Redundancy Allocation Problem (RAP) being one of its central challenges. Since many engineering projects emphasize cost-effectiveness, it is essential to incorporate total system cost as a key optimization objective. This paper proposes a bi-objective RAP that simultaneously maximizes system reliability and minimizes system cost. In the proposed model, the redundancy strategy of each subsystem is treated as a decision variable – selectable among active, standby, or mixed configurations – and component allocation is flexible, allowing a combination of non-identical components within subsystems. To generate a set of optimal trade-off solutions (the Pareto front), an efficient metaheuristic approach known as Fast Non-Dominated Sorting Genetic Algorithm is employed, along with a procedure for identifying the most suitable solution from the Pareto set. The performance of the algorithm is validated using a benchmark problem and a large-scale example. The results show that, despite variations in the initial populations, the final Pareto-optimal sets exhibit minimal differences, demonstrating the robustness of the algorithm. Moreover, the wide variation in cost and reliability among the non-dominated solutions highlights the algorithm's ability to capture diverse optimal trade-offs. Comparative analysis further confirms that the proposed bi-objective model produces solutions with higher reliability and lower cost than existing single-objective approaches, emphasizing its effectiveness in applications where cost efficiency is critical.

Mathematics Subject Classification. 60E05, 90B25, 90C10, 90C29, 90C30.

Received June 29, 2024. Accepted April 4, 2026.

Acronyms

RAP	Redundancy allocation problem
SORAP	Single objective redundancy allocation problem
MORAP	Multi objective redundancy allocation problem
GA	Genetic algorithm
NSGA	Non-dominated sorting genetic algorithm
pdf	Probability density function
cdf	Cumulative distribution function

Keywords. Redundancy allocation problem, redundancy strategy, multi-objective reliability optimization, component mixing, NSGA-II.

[1] Department of Industrial Engineering, University of Gonabad, Gonabad, Iran.

[2] Department of Industrial and Systems Engineering, Isfahan University of Technology, Isfahan, Iran.

*Corresponding author: gholinezhad.hadi@gonabad.ac.ir, gholinezhad.hadi@gmail.com

Notation

S	Number of subsystems
x_{ij}	Number of the jth component allocated to subsystem i as active
y_{ij}	Number of the jth component allocated to subsystem i as standby
$\mathbf{x_i}$, $\mathbf{y_i}$	Vectors of components allocated to subsystem i, ($\mathbf{x_i} = [x_{i1}, x_{i2}, \ldots], \mathbf{y_i} = [y_{i1}, y_{i2}, \ldots]$)
$\mathbf{x}, \mathbf{y}$	Matrices of $[x_{ij}]$ and $[y_{ij}]$, $\left(\mathbf{x} = \begin{bmatrix} x_{11}, x_{12}, \ldots \\ x_{21}, x_{22}, \ldots \\ \vdots \\ x_{s1}, x_{s2}, \ldots \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_{11}, y_{12}, \ldots \\ y_{21}, y_{22}, \ldots \\ \vdots \\ y_{s1}, y_{s2}, \ldots \end{bmatrix}\right)$
n_{ij}	Number of the jth component allocated to subsystem i, ($n_{ij} = x_{ij} + y_{ij}$)
$n_{\max,i}$	Maximum value allowed for n_i
m_i	Number of component choices available for subsystem i
M	Maximum number of component choices available for all the subsystems ($m = \max\{m_i\}$)
$r_{ij}(t)$	Reliability of the jth component available for subsystem i at time t
$R(t; \mathbf{x}, \mathbf{y})$	System reliability at time t due to matrices $\mathbf{x}$ and $\mathbf{y}$
$R_i(t; \mathbf{x_i}, \mathbf{y_i})$	Reliability at time t for subsystem i due to vectors $\mathbf{x_i}$ and $\mathbf{y_i}$
$C(\mathbf{x}, \mathbf{y})$	System cost due to matrices $\mathbf{x}$ and $\mathbf{y}$
w_{ij}, c_{ij}	Weight and cost of the jth component available for subsystem i
$w_{switch,i}, c_{switch,i}$	Weight and cost of switching system for subsystem i (if needed)
W, C	Maximum allowable value for the weight and cost of the system
$\rho_i(t)$	Reliability of switching system for subsystem i at time t (if needed)
z_{il}	Index for lth standby component type in subsystem i, $z_{il} \in \{1, 2, \ldots, m_i\}$
z'_{il}	Index for lth active component type in subsystem i, $z'_{il} \in \{1, 2, \ldots, m_i\}$
λ_{ij}, k_{ij}	Parameters of Erlang distribution for the jth component available for subsystem i
$f_{ij}^{(h_j)}(t)$	pdf of the h_jth failure time for the jth component available for subsystem i
$f_i^{\max}(t)$	pdf of the failure time for the last active component in subsystem i
$f_{ij}(t)$, $F_{ij}(t)$	pdf and cdf of the failure time for the jth component available for subsystem i

1. Introduction

Reliability, an essential attribute in assessing product quality, can be enhanced through various techniques, among which redundancy allocation is particularly effective. The Redundancy Allocation Problem (RAP), a key topic in reliability optimization, involves determining the optimal type and number of components to be allocated to a system's subsystems in order to optimize specific objectives – such as maximizing system reliability – while satisfying given constraints, including system cost and weight. RAP can be classified according to several criteria, including system and component states, redundancy strategy, component repairability, component heterogeneity within subsystems, causes of component failure, and other relevant factors.

In terms of system and components states, RAP can be divided into binary-state and multi-state categories. In the binary-state RAP, the components, as well as the entire system, can be either completely healthy or totally failed, and there is no intermediate state between the two [1–23]. In the multi-state RAP, the system and/or the components can have more than two states [24–28].

With respect to the redundancy strategy, three primary types can be identified: active, standby, and mixed. The redundancy strategy determines how the redundant components are utilized within the subsystems. In an active redundancy strategy, all components operate simultaneously, even though only one is required to function at any given time. Several studies have exclusively examined the active redundancy strategy [1–6]. The standby redundancy strategy can be further categorized into cold, warm, and hot standby configurations [2]. In a cold-standby system, redundant components do not fail before being activated. In a warm-standby system, redundant components fail at a lower rate than those currently in operation. In a hot-standby system, redundant components fail at the same rate as the active component. Numerous studies have focused solely on the standby redundancy strategy [8–12]. In the present paper, the cold-standby redundancy strategy is adopted. Henceforth, any reference to the standby strategy specifically denotes the cold-standby configuration. The mixed redundancy strategy is a versatile approach that encompasses elements of both active and standby strategies, aiming to leverage the benefits of each. Under this strategy, multiple components are operational from the outset, and a set of standby components is activated in the event of the failure of the last active component. There are studies that have explored the mixed redundancy strategy [13–17]. In both standby and mixed redundancy strategies, a switching mechanism is required to activate the redundant components. This mechanism is typically imperfect and characterized by a specific level of reliability. Most previous studies have applied a uniform redundancy strategy across all subsystems within a system [1–12]. However, allowing the subsystems to adopt different strategies – specifically, combining active and standby redundancy – can yield a more flexible and realistic model with markedly higher overall system reliability. Accordingly, several studies have treated the redundancy strategy of each subsystem as a decision variable to achieve improved solutions [18–23].

In the context of redundancy strategies, when employing standby or mixed strategies, the standby components can exist in one of three states: cold, warm, or hot. In the cold standby state, only the active component is susceptible to failure, while the standby components remain unaffected by aging until they are activated. Upon the failure of the active component, the next standby component is activated in a sequential manner [8–10]. In this scenario, the failure rate of each active component is denoted by λ, and the reliability of a subsystem with n cold standby components at time t can be modeled as the cumulative probability of experiencing at most n failures within that time frame, which follows a Poisson process. Conversely, in the warm standby configurations, the active component fails at a rate of λ, while the standby components exhibit a reduced failure rate (typically $\varepsilon\lambda$, where $0 < \varepsilon < 1$) [11,12]. In the hot standby state, all components independently experience failures at a rate of λ. This configuration functions similarly to a parallel system. The cold standby strategy is most appropriate when cost constraints are stringent and when energy consumption or wear-and-tear is a significant concern. In this approach, the failure rate during the standby phase is zero, as the standby components remain inactive and do not experience aging. When the switching is assumed to be perfect, the reliability achieved through the cold standby can surpass that of the active strategy. Cold standby is the most cost-effective option among the various standby types since the standby components do not require any power or maintenance until they are activated. In this strategy, the failure distribution is typically modeled using an exponential failure rate post-activation, with the standby phase contributing no hazard. Common applications of the cold standby include embedded control units, backup generators, and passive systems with low switching frequencies. This paper focuses on the cold standby mode; however, the exploration of warm and hot standby modes is recommended as a potential avenue for future research.

From another perspective, in the RAP, the components within each subsystem may either be identical or non-identical, allowing different component types with varying specifications to be allocated to the subsystems. While some studies have restricted the allocation of identical components [1–3,7,8,12–16,18–22,25], others have allowed non-identical components to be allocated, along with active redundancy strategies [4–6,9–11,24]. There are relatively few studies that consider non-identical components and allow for standby or mixed redundancy strategies [17,23].

Additionally, the components allocated to the subsystems can be either repairable or non-repairable. Non-repairable components cannot be restored to operation once a failure occurs [2–11, 13–23]. Such components are suitable when a system operates in single-use missions or over short durations (*e.g.*, satellites, missiles, temporary field equipment), when field repair is infeasible (*e.g.*, inaccessible locations), or when the cost of repair an infrastructure outweighs its benefit. Some typical applications include aerospace launch systems, military expendable systems, and medical implants. In contrast, repairable components can be fixed and reused after failure [1,12,24]. These components are suitable when the system is maintained in the field or factory, with resources available for repair, when high availability is critical (*e.g.*, 24/7 operations), or when the components can be replaced/swapped during downtime. In this case, downtime and repair times are explicitly considered.

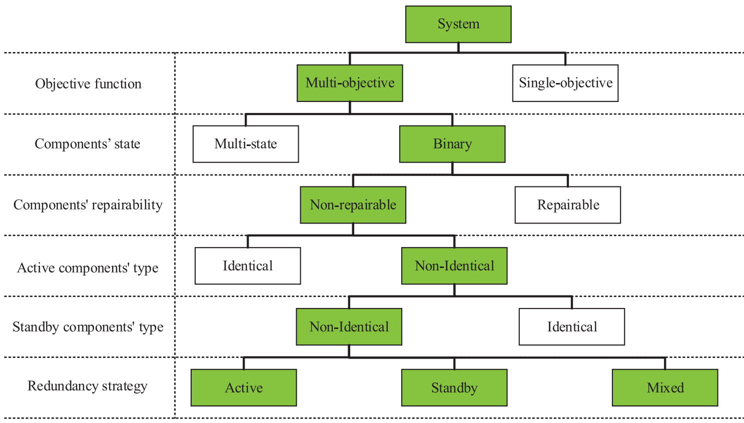

FIGURE 1. The position of the RAP presented in the current research.

Some typical applications include industrial machinery, power plants, data centers and communication networks, and transportation systems.

When components are non-repairable, lifetime modeling focuses on the failure distribution (*e.g.*, exponential, Weibull) and reliability block diagrams or probability theory are used as analysis tools. However, when the components are repairable, lifetime modeling focuses on the failure-repair process (*e.g.*, Markov, Renewal) and Markov processes or queuing models are used as analysis tools. Incorporating repairability would substantially modify both the structure and interpretation of the model. In repairable systems, the system's performance over time is governed by its availability rather than its mission reliability. Consequently, cost considerations would extend to include repair and downtime costs. The optimization model would therefore need to balance reliability design decisions with maintenance-related factors, leading to a more dynamic and time-dependent formulation. In terms of applicability, integrating repairability would enhance the model's relevance to long-life or continuously operating systems, such as manufacturing lines, telecommunication networks, or power systems, where maintenance actions restore functionality.

Most reliability optimization studies have focused on a single objective function, such as maximizing system reliability [4, 8, 10, 13–21, 23] or minimizing system cost [11, 24]. In a Single Objective Redundancy Allocation Problems (SORAP), presenting only one solution may limit the decision-making process. To address this, a Multi-Objective Redundancy Allocation Problem (MORAP) can offer a range of non-dominated solutions [1, 3, 5, 6, 9, 12, 22]. A variety of meta-heuristic algorithms have been proposed to solve both SORAPs, including genetic algorithm [14, 16, 17, 19, 21], particle swarm optimization [20, 24], bat algorithm [4, 27], water cycle algorithm [15], and MORAPs, including non-dominated sorting genetic algorithm [12, 22], multi-objective harmony search [9], artificial bee colony algorithm [1], swarm optimization [3], bat-based algorithm [6], and others [5, 22].

This paper presents a bi-objective model for the RAP aimed at generating a set of optimal solutions. The proposed model permits unrestricted allocation of components of any type and quantity to the subsystems, allowing the integration of non-identical components and treating the redundancy strategy – active, standby, or mixed – as a decision variable. The position of the RAP presented in the current research, is displayed in green in Figure 1.

Some examples of real-world applications of the model presented in this paper are as follows:

– **Satellite and space systems: Onboard computer and power management modules:** Satellites are equipped with multiple processing units, such as flight computers, which are essential for orbit control, navigation, and communication. Some computers can operate actively, managing real-time data and control, while others can remain in standby, powered off or partially powered to conserve energy [29]. Given the extreme limitations of energy resources in space, maintaining full active redundancy is often economically unfeasible. Moreover, the harsh conditions of radiation and temperature can degrade electronic components; thus, standby units are preserved to extend mission longevity. These standby units are intentionally preserved to mitigate the cumulative radiation damage and thermal degradation, thereby extending mission lifetime through fault-tolerant distributed computing architectures [30]. The mixed redundancy approach effectively mitigates risk without exhausting the limited energy and heat dissipation capacity available. It also minimizes the simultaneous exposure of all units to potential radiation damage. The rationale for adopting a mixed redundancy strategy in this context is as follows:

- Energy limitations prevent full activation of all units.
- Switchover delays are critical but can be mitigated by partially active units.
- Radiation-hardened components are expensive; a mix of active and dormant units balances cost and reliability.

– **Aerospace: Flight control systems:** Aircrafts depend on multiple flight control computers (FCCs) to interpret pilot commands and stabilize flight. In this context, several FCCs may operate concurrently in an active state, while one or more may remain in standby mode, receiving data but not actively controlling. The active units facilitate real-time fault masking, allowing for immediate detection and response to issues. The standby units are poised to take over seamlessly if any faults are identified [31,32]. The mixed strategy ensures fault tolerance without excessive interference between control units. The motivation for employing a mixed redundancy strategy in this context arises from the following considerations:

- Ensures high availability with minimal performance degradation in case of partial failure.
- Standby units reduce electrical and thermal stress.
- Safety regulations require immediate fault tolerance with verifiable fallback.

– **Data centers: Uninterruptible power supply (UPS) modules and cooling systems:** Continuous power supply to servers is paramount. UPS units are modular in design; a subset can operate actively in a load-sharing capacity, while the others remain in standby mode, powered but not supplying load unless a failure occurs [33]. The mixed redundancy strategy mitigates the inefficiencies associated with low-load operations and enhances operational flexibility while managing power losses and cooling requirements effectively. Several factors justify the use of a mixed redundancy strategy in this context, as outlined below:

- Balances load efficiency with protection against cascading failures.
- Reduces energy consumption and wear in the backup units.
- Enables maintenance without system shutdown.

– **Railway systems: Train control system units:** Safety-critical systems play a vital role in determining train speeds and routing. In these systems, the main processing units may operate actively while a backup safety processor remains in standby mode, often isolated from potential faults in the primary unit. This isolation prevents fault propagation and helps maintain safety certification by limiting electromagnetic interference from the redundant systems [34, 35]. The implementation of mixed redundancy reduces interference and failure correlation, which is crucial in tightly coupled safety systems. The rationale for adopting a mixed redundancy strategy in this context can be summarized as follows:

- Ensures fail-safe operation with minimal latency.
- Standby controllers are isolated until failure detection to prevent fault propagation.
- Full activation is not allowed due to strict electromagnetic compatibility constraints.

– **Medical devices: Ventilators in intensive care:** Life-support systems, such as ventilators, cannot afford any failures during operation. In these critical systems, the primary controllers can operate actively while the secondary units remain on standby, continuously updated with patient data but not executing control functions. This arrangement enables instant switchover without risking patient safety and reduces component fatigue in the backup systems [36]. Regulatory standards often mandate multiple independent failover pathways; mixed redundancy meets these requirements without introducing conflicting commands or resource inefficiencies. The underlying reasons for selecting a mixed redundancy strategy in this context are summarized as follows:

- Critical for patient safety–failures must be mitigated instantly.
- Warm standby ensures rapid switchover without excessive energy use or wear.
- Full active redundancy may introduce system instability (*e.g.*, conflicting commands).

– **Telecommunication networks: Core network control planes:** The stability of network control is essential for effective routing and session management. To achieve this, some control nodes operate actively while the others remain in standby mode, activated dynamically during fault occurrences or traffic surges. In this configuration, the active nodes maximize throughput while the standby nodes contribute to reduced energy consumption and thermal output [37]. This dynamic load balancing enhances scalability and represents a contemporary, cloud-inspired application of mixed redundancy. The justification for employing a mixed redundancy strategy in this context is presented as follows:

- Enables dynamic load balancing while preserving quick recovery paths.
- Cold standby saves power in low-traffic periods.
- Fully active systems might lead to packet collisions or overload under certain configurations.

The subsequent sections will delve into the problem formulation (Sect. 2), the solution method (Sect. 3), the application of the proposed model to numerical examples (Sect. 4), and conclude with insights and future research directions (Sect. 5).

2. Problem formulation

The system described in this paper is based on a binary-state MORAP with non-repairable components and multiple redundancy strategies for subsystems. Additionally, the model allows non-identical components to be mixed within subsystems. Figure 2 shows a general system configuration with the assumptions considered in this paper.

2.1. Assumptions

In the MORAP considered in this paper, it is assumed that:

– The entire system and all components are considered in binary form.
– Independent component failures do not immediately cause system failure.
– Components characteristics including cost and weight as well as the failure distribution parameters are predetermined.
– The redundancy strategies considered in the model are active, standby, mixed, or none and are treated as decision variables.
– The model accounts for imperfect switching systems in standby and mixed redundancy strategies.
– Mixing non-identical components within subsystems is allowed.
– The allocated components are assumed to be non-repairable and no preventive maintenance is considered.

The following subsection presents a mathematical model of the problem under the above assumptions.

2.2. Mathematical modeling

The proposed bi-objective model can be written as follows:

$$\max\ R(t;\mathbf{x},\mathbf{y}) \tag{1}$$

$$\min\ C(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{s}\sum_{j=1}^{m_i} c_{ij}n_{ij} + \sum_{i\in\{standby,mixed\}} c_{switch,i} \tag{2}$$

s.t.

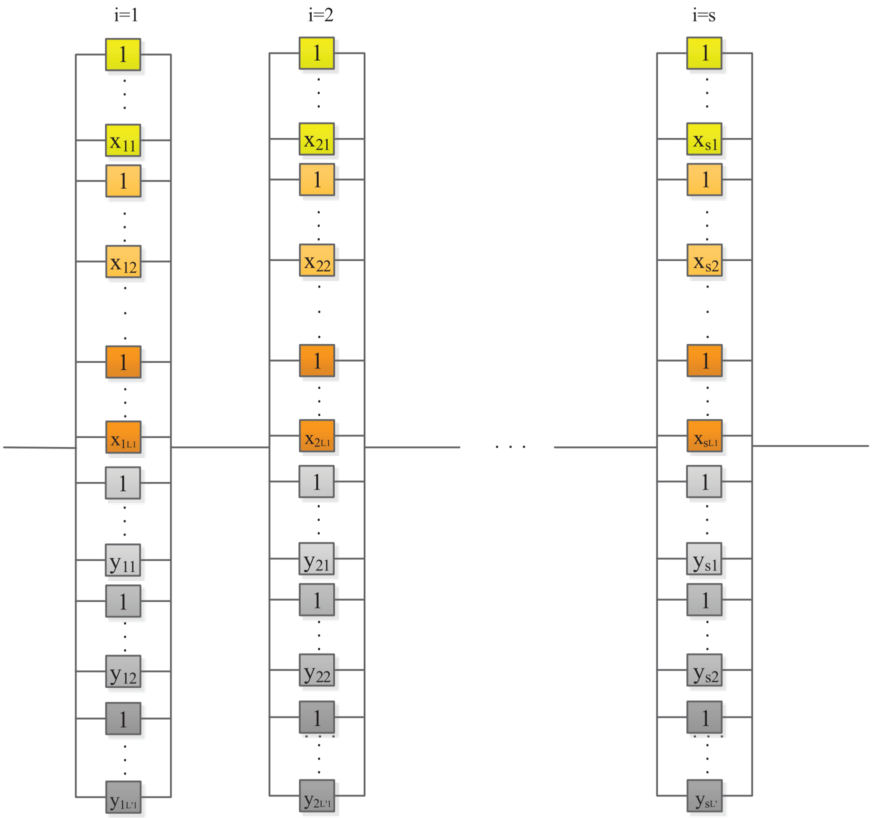

FIGURE 2. A general form of a series–parallel system with component mixing.

$$\sum_{i=1}^{s}\sum_{j=1}^{m_i} w_{ij}n_{ij} + \sum_{i\in\{standby,mixed\}} w_{switch,i} \leq W \tag{3}$$

$$n_{ij} = x_{ij} + y_{ij}; \qquad \forall i, j \tag{4}$$

$$\sum_{j=1}^{m_i} x_{ij} \geq 1; \qquad \forall i \tag{5}$$

$$\sum_{j=1}^{m_i} n_{ij} \leq n_{\max,i}; \qquad \forall i. \tag{6}$$

The mathematical model proposed in the paper aims to optimize the allocation of active and standby components in each subsystem to maximize system reliability (Eq. (1)) while minimizing system cost (Eq. (2)). The proposed model includes constraints related to weight limits, component allocation, and redundancy strategy selection. Constraint (3) enforces the available weight limit. Equation (4) states that the number of allocated components of type j in a subsystem is equal to the sum of the number of active and standby components of this type. Constraint (5) ensures that at least one component is active. Constraint (6) takes into consideration the maximum allowed value for the number of components allocated to each subsystem.

The MORAP system described in this paper offers a range of redundancy strategies for each subsystem, including active, standby, mixed, or none. Accordingly, the first objective function can be expressed as follows:

$$\begin{aligned}
R(t;\mathbf{x},\mathbf{y}) = {} & \prod_{i\in NR}\left(r_{iz'_{i1}}(t)\right) \times \prod_{i\in active}\left(1-\prod_{j\in z'_i}(1-r_{ij}(t))^{x_{ij}}\right)\\
& \times \prod_{i\in standby}\left(r_{iz'_{i1}}(t)+\int_0^t \rho_i(u) f_{iz'_{i1}}(u) r_{iz_{i1}}(t-u)\,\mathrm{d}u + \sum_{j\in z_i}\sum_{h_j=0}^{y_{ij}-1}\int_0^t \rho_i(u) f_{ij}^{(h_j)}(u) r_{ij}(t-u)\,\mathrm{d}u\right.\\
& \left. - \int_0^t \rho_i(u) f_{iz_{i1}}^{(0)}(u) r_{iz_{i1}}(t-u)\,\mathrm{d}u\right)\\
& \times \prod_{i\in mixed}\left(\left(1-\prod_{j\in z'_i}(1-r_{ij}(t))^{x_{ij}}\right)+\int_0^t \rho_i(u) f_i^{\max}(u) r_{iz_{i1}}(t-u)\,\mathrm{d}u\right.\\
& + \sum_{j\in z_i}\sum_{h_j=0}^{y_{ij}-1}\int_0^t\int_{t_1}^t \rho_i(u) f_i^{\max}(v) f_{ij}^{(h_j)}(u-v) r_{ij}(t-u)\,\mathrm{d}u\,\mathrm{d}v\\
& \left. - \int_0^t\int_{t_1}^t \rho_i(u) f_i^{\max}(v) f_{iz_{i1}}^{(0)}(u-v) r_{iz_{i1}}(t-u)\,\mathrm{d}u\,\mathrm{d}v\right).
\end{aligned} \tag{7}$$

The reliability of a subsystem without any redundant components (denoted by "NR") is equivalent to the reliability of the single allocated component. In this context, z'_{i1} denotes the index of the sole allocated component within a given subsystem. For instance, if a component of type 3 is allocated to subsystem i, then $z'_{i1}=3$ and then $r_{iz'_{i1}}(t)=r_{i3}(t)$.

For a subsystem with an active redundancy strategy (denoted by "*active*"), reliability is based on the probability that at least one active component remains operational until time t. In this context, z'_i denotes the index set of active components allocated to subsystem i. For instance, if components of types 1, 2, and 4 are allocated to subsystem i, then $z'_i=\{1,2,4\}$. Consequently, the reliability of subsystem i is calculated as follows:

$$1-\prod_{j\in z'_i}(1-r_{ij}(t))^{x_{ij}} = 1-\prod_{j\in\{1,2,4\}}(1-r_{ij}(t))^{x_{ij}} = 1-(1-r_{i1}(t))^{x_{i1}}\times(1-r_{i2}(t))^{x_{i2}}\times(1-r_{i4}(t))^{x_{i4}}. \tag{8}$$

The reliability of a subsystem employing a standby strategy (denoted by "*standby*") can be analyzed through several parts. The first part pertains to the reliability of the sole active component, which is defined as the probability that this active component will remain operational until time t, thereby negating the need for any standby components. The second part involves the probability that the active component fails prior to time t, specifically at time u. At this point, the switching system remains functional, allowing the first standby component to be activated and to maintain its operational status until time t. The third part, which comprises two nested summations, represents the probabilities associated with the failure of the h_jth standby component of type j at a time prior to t, denoted as u, with the switching system being operational at this moment, allowing for the activation of the subsequent standby component, and its remaining functional until time t. Since the probability of the first standby component being activated and remaining functional until time t was determined in the second part, it should not be included in the summation of the third part. Consequently, the fourth part adjusts this probability to account for its exclusion in the third part. In this context, z'_{i1} denotes the index of the sole active component allocated to subsystem i, while z_{i1} represents the index of the first standby component allocated to subsystem i. For instance, if a component of type 3 is allocated as the active component to subsystem i, and the first standby component is of type 4, then $z'_{i1}=3$ and $z_{i1}=4$. Consequently, the first and second parts of the reliability calculation for subsystem i are expressed as follows:

$$r_{iz'_{i1}}(t)+\int_0^t \rho_i(u) f_{iz'_{i1}}(u) r_{iz_{i1}}(t-u)\,\mathrm{d}u = r_{i3}(t)+\int_0^t \rho_i(u) f_{i3}(u) r_{i4}(t-u)\,\mathrm{d}u. \tag{9}$$

The reliability of a subsystem employing a mixed redundancy strategy (denoted by "*mixed*") comprises several parts. The first part pertains to the reliability of the active components, which is defined as the probability that at least one active component remains functional until time t, thereby negating the need for any standby components. The second part addresses the probability that the last active component fails at a time prior to t (denoted as time u), at which point the switching system is operational, allowing the first standby component to be activated and to remain functional until time t. The third part involves a more complex calculation, represented by two nested summations. This accounts for the probability that the last active component fails at a time before t (specifically at time v). Following this failure, the standby components are activated sequentially and may fail until the h_jth standby component of type j fails at a time between v and t (for instance, at time u). During this interval, the switching system should remain operational, allowing for the activation of the next standby component, so that it remains functional until time t. It is important to note that the probability of failure of the last active component and the subsequent activation and survival of the first standby component has already been accounted for in the second part. Therefore, this probability should not be included in the summation of the third part. Consequently, the fourth part adjusts this probability to avoid double-counting in the overall reliability assessment. In this context, z' and z_{i1} represent, respectively, the set of indices of the actively allocated components and the index of the first standby component, as previously mentioned. In equation (9), $f_i^{\max}(t)$ is calculated as follows:

$$f_i^{\max}(t) = \sum_{j \in z_i'} x_{ij} \left(\prod_{\substack{k \in z_i' \\ k \neq j}} (F_{ik}(t))^{x_{ik}} \right) \cdot (F_{ij}(t))^{x_{ij}-1} \cdot f_{ij}(t) \tag{10}$$

where, $f_{ij}(t)$ and $F_{ij}(t)$ represent the probability density function (pdf) and cumulative distribution function (cdf) of the time-to-failure of the jth component available for subsystem i, respectively. Equation (10) is a summation over the last component that fails. For example, if subsystem i has 2 components of type 2 and 1 component of type 3 as active components, $f_i^{\max}(t)$, therefore, will be as follows:

$$f_i^{\max}(t) = F_{i2}(t) \cdot F_{i2}(t) \cdot f_{i3}(t) + F_{i2}(t) \cdot F_{i3}(t) \cdot f_{i2}(t) + F_{i2}(t) \cdot F_{i3}(t) \cdot f_{i2}(t). \tag{11}$$

Equation (11) comprises three parts. The first part calculates the probability of type 2 components failing before time t, while a type 3 component fails exactly at time t. The second and third parts follow a similar structure, except in the second part, the last failing component is the first type 2 component, and in the third part, it is the second type 2 component. Equation (11) can be alternatively expressed as equation (12), which bears resemblance to equation (10).

$$f_i^{\max}(t) = 2F_{i3}^{1}(t) \cdot F_{i2}^{(2-1)}(t) \cdot f_{i2}(t) + 1F_{i2}^{2}(t) \cdot F_{i3}^{(1-1)}(t) \cdot f_{i3}(t). \tag{12}$$

It is important to note that the mixed redundancy strategy serves as a comprehensive approach encompassing other strategies, suitable for all scenarios. In the mixed redundancy strategy, when $\sum_{j=1}^{m_i} y_{ij} = 0$, it indicates an active or no redundancy strategy selection, and when $\sum_{j=1}^{m_i} x_{ij} = 1$, it signifies a standby or no redundancy strategy selection. However, active and standby strategies are generally preferred due to their simplicity.

In equations (7) and (9), the time to failure for each component type within a subsystem may follow any arbitrary distribution. Accordingly, for subsystems employing standby or mixed redundancy strategies, $f_{ij}(t)$ and $F_{ij}(t)$ represent the pdf and cdf of any chosen distribution. Specifically, $f_{ij}^{(hj)}(t)$ denotes the pdf of the time to the h_jth failure of the jth component type in subsystem i. This function corresponds to the pdf of a random variable defined as the sum of several other random variables, each potentially following a different distribution. Under certain conditions, an explicit form of $f_{ij}^{(hj)}(t)$ can be derived. For example, if all component lifetimes within a subsystem are exponentially distributed with identical parameters, $f_{ij}^{(hj)}(t)$ follows an Erlang distribution, representing the sum of multiple exponential variables. Similarly, if all component lifetimes are normally distributed, $f_{ij}^{(hj)}(t)$ also follows a normal distribution, as it represents the sum of normally distributed variables. However, when the components' lifetimes follow arbitrary or unknown distributions without a closed-form expression for the sum of multiple variables, calculating $f_{ij}^{(hj)}(t)$ requires multiple nested integrals, resulting in significant computational complexity.

3. Solution method

Since the RAP is classified as NP-hard, a meta-heuristic algorithm is developed in this study to search for near-optimal solutions. The Genetic Algorithm (GA), a widely recognized and effective meta-heuristic technique, is employed for this purpose due to its proven capability in solving complex optimization problems [38]. Several GA-based methods have been developed to address multi-objective optimization problems, among which the Fast Non-Dominated Sorting Genetic Algorithm (NSGA-II) is recognized as one of the most efficient and powerful approaches [39]. Comparative studies have demonstrated the superior performance of NSGA-II in reliability optimization problems, confirming its suitability for such applications [40]. Therefore, this study employs NSGA-II as the primary solution method for identifying the optimal set of solutions. The pseudo-code of the algorithm is provided below.

- $it \leftarrow 0$.
- Generate the initial population ($P(0)$).
- Calculate the reliability and cost of the system for each individual in $P(0)$ (the system reliability value is defined as Eq. (7) minus a coefficient of the amount of violation of the constraints (if violated)).
- Sort all individuals in $P(0)$ based on rank and crowding distance.
- Termination flag $\leftarrow$ False.
- While termination flag is False, do:
  - $it \leftarrow it + 1$.
  - Select parents for reproduction using a selection operator (*e.g.*, roulette wheel selection).
  - Generate offspring using crossover and mutation operators on the selected parents.
  - Calculate the reliability and cost of the system for each individual in the offspring population (the system reliability value is defined as Eq. (7) minus a coefficient of the amount of violation of the constraints (if violated)).
  - Combine the parents and offspring populations.
  - Sort all individuals in the combined population based on rank and crowding distance and determine all non-dominated fronts ($F1, F2, \ldots$).
  - Select the non-dominated solutions ($F1$) as the Pareto front.
  - Select the first $nPop$ sorted individuals as the new population ($P(t)$).
  - If a stopping criterion is met (*e.g.*, maximum generations reached), set termination flag to True.

In the presented pseudocode, offspring solutions are generated using two main genetic operators: crossover and mutation. To promote solution diversity, multiple crossover techniques are employed, including single-point, double-point, uniform, and max–min crossover. Likewise, mutation is performed using operators such as random mutation and max–min mutation. For each offspring, the reliability of every subsystem is evaluated based on the corresponding part of equation (7), which depends on the redundancy strategy determined by the number of active and standby components within each subsystem. To ensure feasibility, a large penalty proportional to the degree of constraint violation is subtracted from the system reliability, thereby reducing the likelihood of selecting infeasible solutions in subsequent generations. Finally, any infeasible solutions remaining in the obtained Pareto front are eliminated. The NSGA-II algorithm described here is employed to solve the proposed MORAP. A brief overview of the key components constituting the NSGA-II algorithm is provided below.

$$
\begin{array}{rc}
 & \begin{array}{cccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \end{array} \\
\begin{array}{r} \textit{type}\,1 \\ \textit{type}\,2 \\ \textit{type}\,3 \\ \textit{type}\,4 \\ \textit{type}\,1 \\ \textit{type}\,2 \\ \textit{type}\,3 \\ \textit{type}\,4 \end{array} &
\begin{pmatrix}
1 & 0 & 3 & 0 & 1 & 0 & 0 & 2 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 2 & 1 & 0 & 2 & 0 & 1 & 2 & 1 \\
1 & 0 & 0 & 1 & 0 & 2 & 0 & 0 & 2 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 2 & 1 & 2 & 0 & 0 & 1 \\
0 & 0 & 0 & 2 & 2 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 2 & 1 \\
1 & 2 & 0 & 0 & 1 & 2 & 1 & 2 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
3 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 1
\end{pmatrix}
\end{array}.
$$

FIGURE 3. An example for encoded solution (chromosome).

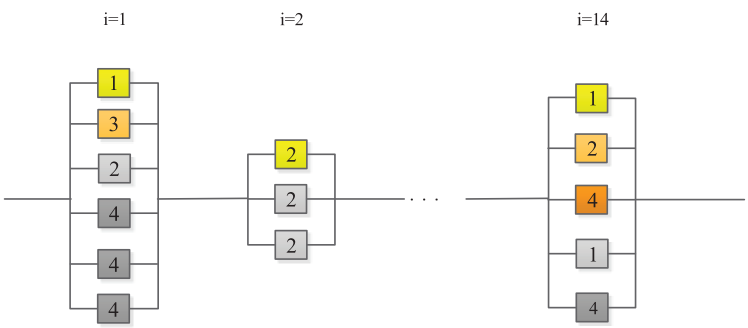

FIGURE 4. The detailed structure of chromosome illustrated in Figure 3.

3.1. Solution encoding (Chromosome)

In GA, each encoded solution, referred to as a chromosome, is typically represented as a matrix or a vector. In the algorithm developed in this study, each chromosome is structured as a matrix representation. The matrix contains s columns corresponding to the number of subsystems and $m = \max\{m_i\}$ rows, where m_i denotes the number of available component types for subsystem i. The first m rows represent the number of active components of each type, while the subsequent m rows indicate the number of standby components of each type within a subsystem. For any subsystem with fewer available component types than m (*i.e.*, when $m_i < m$), the final $(m - m_i)$ rows corresponding to the active and standby components are set to zero.

An example of the solution encoding for a system consisting of 14 subsystems with $m = 4$ is illustrated in Figure 3, and the corresponding configuration is depicted in Figure 4. In this example, subsystem 1 employs non-identical components, including one active component of type 1, one active component of type 3, one standby component of type 2, and three standby components of type 4, thereby implementing a mixed redundancy strategy. Similarly, subsystems 2 and 3 utilize standby and active redundancy strategies, respectively, with the former comprising identical components and the latter incorporating non-identical components.

3.2. Initial population

In this study, the initial population consists of *nPop* randomly generated solution matrices, where *nPop* denotes the population size.

3.3. Selection

To apply the mutation and crossover operators, parent chromosomes are selected using the tournament selection method. In this approach, k chromosomes are randomly chosen, and the one with the highest rank among them is selected as a parent. If multiple chromosomes share the highest rank, the chromosome with the largest crowding distance is chosen. Following the selection process, the mutation and crossover operators are applied to generate offspring.

3.4. Crossover

The crossover operator is applied at a predefined rate rc. To enhance population diversity, several crossover techniques are employed, including single-point, double-point, uniform, and max–min crossover. In the single-point crossover, two parent chromosomes are selected, and a crossover point (column) between 1 and $(s-1)$ is randomly chosen. All corresponding genes after this point are then exchanged between the parents. In the double-point crossover, two crossover points are randomly selected between 1 and s, and all corresponding genes between these two points are swapped. In the uniform crossover, the genes of each column (subsystem) are exchanged between the parents with a probability of 0.5; in other words, each column's genes are randomly assigned to one of the offspring. In the max–min crossover, the columns corresponding to the subsystems with the lowest and highest reliabilities in the parent chromosomes are exchanged.

3.5. Mutation

The mutation operator is applied at a predefined rate rm. In this study, two types of mutation operators are employed: random mutation and max–min mutation. In the random mutation, the values of the genes in each column (subsystem) is altered with a probability pr_m. In the max–min mutation, the columns corresponding to the subsystems with the lowest and highest reliabilities are subjected to mutation.

3.6. Selection

After applying the mutation and crossover operators, the previous population and the newly generated solutions are combined and then sorted based on their rank and crowding distance. The top *nPop* solutions are selected to form the new generation, from which the non-dominated solutions constitute the Pareto front.

3.7. Stopping condition

The algorithm terminates after reaching the predefined maximum number of iterations (*MaxIt*).

4. Numerical example

As an illustration, the newly proposed MORAP is tested on two systems. The first example is a well-known benchmark problem considered in previous studies [2, 4, 8, 13, 19, 20]. The example is a system composed of 14 subsystems, each offering 3 or 4 component choices for selection, each with its own cost and weight. Component failure times follow an Erlang distribution. The specifications of the components are presented in Table 1. The objective functions in this numerical example are defined to maximize system reliability at the mission time $t = 100$ and to minimize the corresponding total system cost.

The maximum allowable system weight is set to $W = 170$. For all subsystems employing mixed or standby redundancy strategies, the switching system is assumed to have a reliability of $\rho_i(t) = 0.99$ at the mission time, while its cost and weight are considered negligible. Furthermore, it is assumed that each subsystem can accommodate a maximum of six components ($n_{\max,i} = 6$).

By trial-and-error, the parameters for the application of the proposed NSGA-II are set to $nPop = 1200$, $rc = 0.25$, $rm = 0.3$ and $pr_m = 0.2$. Moreover, a maximum of 100 iterations is set as *MaxIt*.

Table 1. Specifications of components for the first numerical example.

	Choice 1 ($j = 1$)				Choice 2 ($j = 2$)				Choice 3 ($j = 3$)				Choice 4 ($j = 4$)			
i	λ_{ij}	k_{ij}	c_{ij}	w_{ij}	λ_{ij}	k_{ij}	c_{ij}	w_{ij}	λ_{ij}	k_{ij}	c_{ij}	w_{ij}	λ_{ij}	k_{ij}	c_{ij}	w_{ij}
1	0.00532	2	1	3	0.000726	1	1	4	0.00499	2	2	2	0.00818	3	2	5
2	0.00818	3	2	8	0.000619	1	1	10	0.00431	2	1	9	–	–	–	–
3	0.0133	3	2	7	0.011	3	3	5	0.0124	3	1	6	0.00466	2	4	4
4	0.00741	2	3	5	0.0124	3	4	6	0.00683	2	5	4	–	–	–	–
5	0.00619	1	2	4	0.00431	2	2	3	0.00818	3	3	5	–	–	–	–
6	0.00436	3	3	5	0.00567	3	3	4	0.00268	2	2	5	0.000408	1	2	4
7	0.0105	3	4	7	0.00466	2	4	8	0.00394	2	5	9	–	–	–	–
8	0.015	3	3	4	0.00105	1	5	7	0.0105	3	6	6	–	–	–	–
9	0.00268	2	2	8	0.000101	1	3	9	0.000408	1	4	7	0.000943	1	3	8
10	0.0141	3	4	6	0.00683	2	4	5	0.00105	1	5	6	–	–	–	–
11	0.00394	2	3	5	0.00355	2	4	6	0.00314	2	5	6	–	–	–	–
12	0.00236	1	2	4	0.00769	2	3	5	0.0133	3	4	6	0.011	3	5	7
13	0.00215	2	2	5	0.00436	3	3	5	0.00665	3	4	6	–	–	–	–
14	0.011	3	4	6	0.00834	1	4	7	0.00355	2	2	6	0.00436	3	6	9

The proposed NSGA-II was implemented in the MATLAB environment on a computer with an Intel® Core^{TM} i5 CPU @ 2.67 GHz and 4 GB of RAM running a 64-bit Windows operating system. Due to the stochastic nature of NSGA-II, the example underwent 5 algorithm trials.

The results obtained for the first example are presented in Figure 5. Furthermore, given the large number of non-dominated solutions generated in each trial, a summary of these solutions is provided in Table 2. In Figure 5, the initial randomly created population and the last generated Pareto front are shown. Notably, despite variations in the initial populations across trials, there is minimal disparity in the final Pareto optimal set, showcasing the algorithm's robustness in this regard. Furthermore, the standard deviation values of the cost and reliability for the non-dominated solutions obtained from various trials of the NSGA-II (shown in Tab. 2) demonstrate a commendable diversity among the solutions. This indicates the algorithm's capability to identify a broad spectrum of diverse solutions, thereby enhancing the decision maker's capacity to make informed decisions.

Among the five runs, the run in which the largest system reliability is achieved is selected for further investigation. In run #3 of NSGA-II for the first example, 103 non-dominated solutions are obtained. A summary of some solutions is provided Table 3. The solution with the highest reliability in the Pareto set is presented in the first row, with a reliability value of 0.992664. As shown in Table 3, several solutions simultaneously achieve higher system reliability and lower system cost compared to those reported in the previous studies. In other words, these solutions dominate all previously obtained results. The identification of such solutions – unattainable through single-objective formulations – highlights one of the key advantages of employing a bi-objective optimization approach. In SORAPs, the primary goal is to maximize system reliability, while the total system cost is typically treated as a constraint rather than an objective. Consequently, the cost is not necessarily minimized and may even exceed that of prior studies, which can be undesirable, particularly when cost considerations play a critical role.

The first 22 solutions in Table 3 dominate the best solutions reported in the previous studies, except those employing mixed redundancy strategies [2,8,18]. Furthermore, two solutions outperform the best result obtained by the mixed redundancy strategy without component mixing [13]. These solutions, along with the best results from other studies, are presented in Figure 6 and Table 4 to facilitate a clear comparison. As can be deduced, the proposed model in this paper can find a variety of good solutions that allow to the decision maker to choose appropriately. It is important to note that the best solutions obtained by previous research corresponds to the works employing a single-objective model, in which the optimal solution is defined as the one achieving the highest system reliability. The purpose of comparing the non-dominated solutions obtained in this study with the best solutions reported in previous research is to demonstrate that the proposed bi-objective model

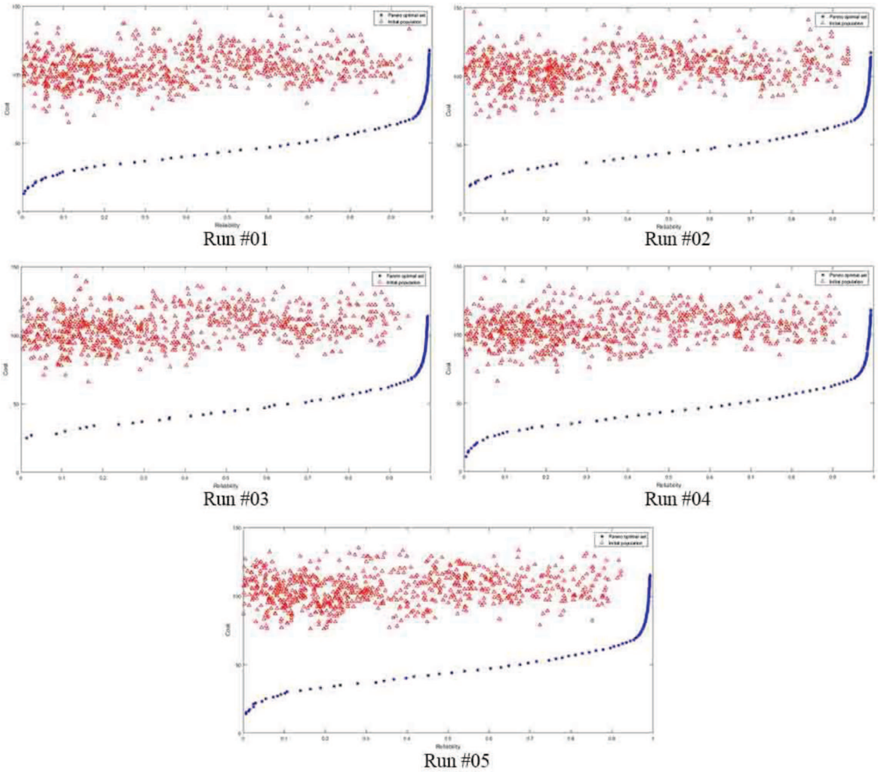

FIGURE 5. Initial population and Pareto optimal set derived through NSGA-II for the first example.

can identify solutions superior to those of earlier single-objective models in terms of both reliability and cost, highlighting the limitations of single-objective approaches in capturing such solutions.

The second numerical example involves a larger-scale system containing 15 subsystems, each providing 10 component choices for selection. The purpose of the second example is to provide an example with a larger scale to demonstrate that the presented solution method can also be used for such problems. The components' failure times in this example follow Erlang distributions. The specifications of the components for the second example are presented in Table A.1 located in Appendix A. Similar to the previous example, this numerical example aims to maximize the system reliability at the mission time $t = 100$ while minimizing the total system cost. The system is constrained by a maximum allowable weight of $W = 400$. For all subsystems employing mixed or standby redundancy strategies, the reliability of the switching mechanism at the mission time is assumed to be $\rho_i(t) = 0.99$, and its cost and weight are considered negligible. Each subsystem is also assumed to be capable of containing up to six components ($n_{\max,i} = 6$).

Through a trial-and-error procedure, the parameters of the proposed NSGA-II algorithm were calibrated as follows: $nPop = 800$, $rc = 0.3$, $rm = 0.4$ and $pr_m = 0.2$. The maximum number of iterations was set to $MaxIt = 100$. Owing to the stochastic nature of the NSGA-II algorithm, five independent runs were performed for this example to ensure the consistency of the results.

The results obtained for the second example are presented in Figure 7, and a summary of non-dominated solutions is provided in Table 5. Figure 7 illustrates the initial randomly generated population along with the final Pareto front. The results of the second example further demonstrate that, despite differences in the initial populations across runs, the resulting Pareto-optimal sets show only minor variations, confirming the robustness

TABLE 2. Summary of non-dominated solutions achieved in different trials of NSGA-II; first example.

	run #1		run #2		run #3		run #4		run #5	
	Cost	Reliability	Cost	Reliability	Cost	Reliability	Cost	Reliability	Cost	Reliability
1	118	0.9925445	117	0.9925213	118	0.9926643	115	0.99246923	114	0.99244606
10	108	0.9917125	106	0.9913748	108	0.9917321	106	0.99137475	105	0.99078161
20	98	0.9896516	96	0.9891482	98	0.9887177	96	0.9891482	95	0.98891883
30	88	0.9860826	86	0.9844317	87	0.9844948	86	0.98435019	85	0.98395601
40	78	0.9770431	76	0.9749722	77	0.9763614	76	0.97495466	75	0.9719936
50	68	0.9532926	66	0.934488	67	0.9442636	66	0.93521679	65	0.92558344
60	58	0.8185601	58	0.827185	57	0.8103909	56	0.79472173	61	0.878483
70	49	0.6481963	48	0.6120354	47	0.6038223	46	0.57226316	45	0.52186723
80	39	0.3624088	38	0.3412399	37	0.3233988	36	0.2794793	35	0.23904663
90	29	0.0979725	27	0.0646752	–	–	26	0.07326356	–	–
100	18	0.0132514	–	–	–	–	–	–	–	–
Last solution	13	0.0031792	20	0.0143009	36	0.2824421	14	0.00681842	25	0.01384865
Average	66.33	0.7084122	67.78	0.7382116	76.38	0.860167	65.83	0.7102245	70.91	0.7856738
Standard deviation	29.49	0.3566795	27.33	0.3343039	23.82	0.1989327	28.86	0.3549181	25.27	0.2900902
Number of solutions	103		97		81		99		87	

TABLE 3. Some of non-dominated solutions achieved in run #3 of NSGA-II for the first example.

Solution no.	System characteristics			L2	Solution no.	System characteristics			L2
	Reliability	Cost	Weight			Reliability	Cost	Weight	
1	0.992664	118	170	1.000000	17	0.989697	101	170	0.841127
2	0.992641	117	170	0.990654	18	0.989574	100	170	0.831782
3	0.992494	115	170	0.971963	19	0.989424	99	170	0.822436
4	0.992471	114	170	0.962617	20	0.988718	98	170	0.813094
5	0.992320	113	170	0.953271	21	0.988531	97	170	0.803749
6	0.992197	112	170	0.943925	22	0.988486	96	170	0.794404
7	0.992119	111	170	0.934580	30	0.984495	87	170	0.710329
8	0.991968	110	170	0.925234	40	0.976361	77	166	0.617043
9	0.991817	109	170	0.915888	50	0.944264	67	148	0.525654
10	0.991732	108	170	0.906543	60	0.810391	57	126	0.467851
11	0.991582	107	170	0.897197	**63**	**0.762819**	**54**	**121**	**0.464399***
12	0.991394	106	170	0.887851	70	0.603822	47	110	0.517906
13	0.990382	105	170	0.878508	80	0.323399	37	88	0.719938
14	0.990221	104	170	0.869162	90	0.134664	30	73	0.886763
15	0.990071	103	170	0.859817	100	0.017334	17	56	0.989201
16	0.990069	102	170	0.850471	103	0.005098	11	41	1.000000

*The best solution on the Pareto front based on the L_2 norm criterion.

of the algorithm. Furthermore, the standard deviation values of the cost and reliability for the non-dominated solutions obtained from different trials of the NSGA-II (as reported in Tab. 5) reveal considerable variation among the solutions. This finding further confirms the algorithm's capability to identify a wide range of diverse and well-distributed solutions.

In the second example, the highest system reliability was obtained in the first run, which yielded 74 solutions. A summary of these solutions is provided in Table 6. The solution with the greatest reliability within the Pareto set is listed in the first row, with a reliability value of 0.9985001.

The wide range of Pareto-optimal solutions provides the decision maker with substantial flexibility in selecting the most appropriate design. For instance, the non-dominated solutions obtained in the first run indicate that achieving the highest system reliability of 0.9985001 requires an expenditure of 232 cost units, whereas a reliability level of 0.9625664 can be achieved at a cost of only 82. The preferable decision among these alternatives depends on several contextual factors. For example, in the case of a critical system where failure could lead

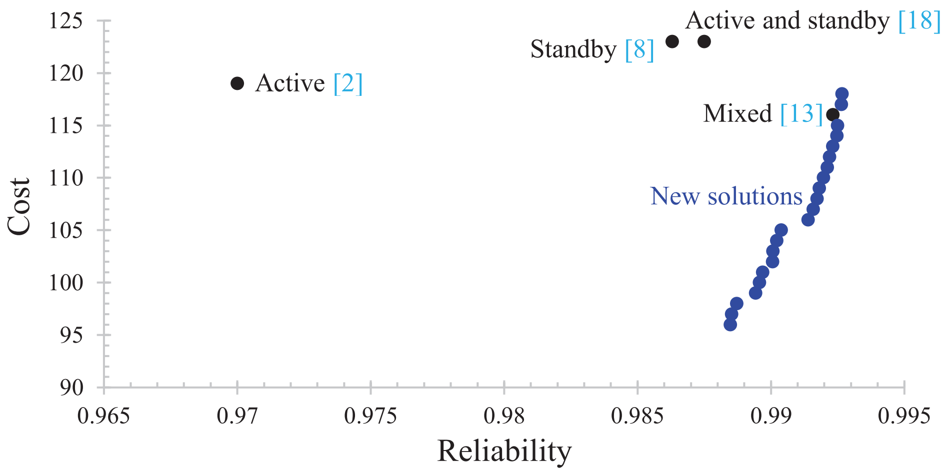

FIGURE 6. Non-dominated solutions derived through the proposed model and the best solutions achieved by the previous researches.

TABLE 4. Details of non-dominated solutions derived through the proposed model and the best solutions achieved by the previous researches – first example.

Solution no.	Reliability	Cost	Comment
1	0.992664	118	New solution
2	0.992641	117	New solution
3	0.992494	115	New solution
4	0.992471	114	New solution
	0.99233	116	Best solution with mixed redundancy strategy [13]
5	0.992320	113	New solution
6	0.992197	112	New solution
7	0.992119	111	New solution
8	0.991968	110	New solution
9	0.991817	109	New solution
10	0.991732	108	New solution
11	0.991582	107	New solution
12	0.991394	106	New solution
13	0.990382	105	New solution
14	0.990221	104	New solution
15	0.990071	103	New solution
16	0.990069	102	New solution
17	0.989697	101	New solution
18	0.989574	100	New solution
19	0.989424	99	New solution
20	0.988718	98	New solution
21	0.988531	97	New solution
22	0.988486	96	New solution
	0.9875	123	Best solution with redundancy strategy as a decision variable (active and standby) [18]
	0.9863	123	Best solution with standby redundancy strategy [8]
	0.97	119	Best solution with active redundancy strategy [2]

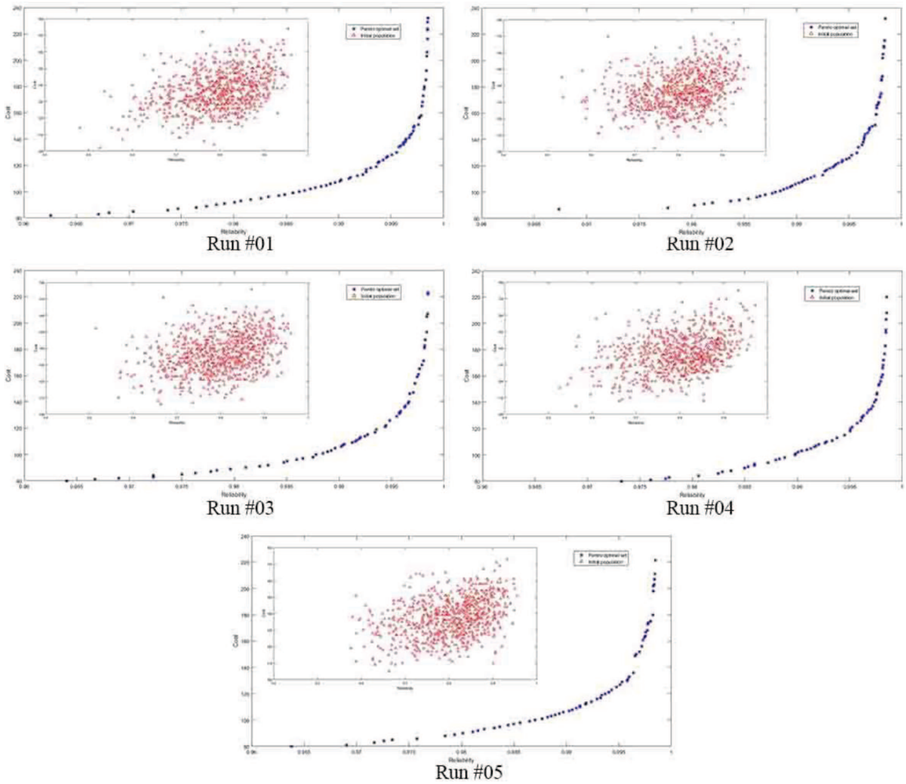

FIGURE 7. Initial population and Pareto optimal set derived through NSGA-II for the second example.

TABLE 5. Summary of non-dominated solutions achieved in different trials of NSGA-II; second example.

	run #1		run #2		run #3		run #4		run #5	
	Cost	Reliability	Cost	Reliability	Cost	Reliability	Cost	Reliability	Cost	Reliability
1	232	0.9985001	242	0.9984726	251	0.9984961	220	0.998501	234	0.9984912
10	180	0.9981671	185	0.998082	181	0.9981162	166	0.9981449	171	0.9983551
20	149	0.9971287	151	0.9974753	139	0.9965923	137	0.9971774	143	0.9978744
30	135	0.9958872	136	0.9962365	119	0.9935504	120	0.9950734	126	0.9966963
40	118	0.9930752	122	0.9939836	107	0.9903446	105	0.9912842	114	0.9935483
50	106	0.989219	110	0.9910124	95	0.9850428	93	0.9859284	106	0.9905067
60	96	0.9833192	100	0.9879872	84	0.9723124	81	0.9760023	97	0.985742
70	86	0.9736912	90	0.9802854	–	–	–	–	87	0.9770377
Last solution	82	0.9625664	86	0.9452374	78	0.9416643	75	0.8812292	77	0.9301408
Average	131.04	0.9905011	135.01	0.9922273	127.67	0.9888943	122.39	0.9891656	124.81	0.9890168
Standard deviation	39.28	0.0087349	37.93	0.0080631	43.03	0.0112344	36.46	0.0158324	37.87	0.0130488
Number of solutions	74		73		66		66		79	

to severe or irreparable consequences, allocating more cost to achieve maximum reliability would be justified. Conversely, if system failures incur relatively minor costs and can be remedied through simple repairs, opting for a lower-cost solution with slightly reduced reliability would be more reasonable. Furthermore, the scale or unit of cost may also influence the decision regarding whether the additional expenditure for higher reliability is warranted.

TABLE 6. Some of non-dominated solutions achieved in run #1 of NSGA-II for the second example.

Solution no.	System characteristics			L2	Solution no.	System characteristics			L2
	Reliability	Cost	Weight			Reliability	Cost	Weight	
1	0.9985001	232	398	1.00000	30	0.9958872	135	248	0.36074
2	0.9984794	229	395	0.98000	40	0.9930752	118	229	0.28353
3	0.9984756	224	392	0.94667	**43**	**0.9922954**	**113**	**245**	**0.26931***
4	0.9984569	223	389	0.94000	50	0.9892190	106	217	0.30383
5	0.9984510	216	397	0.89333	60	0.9833192	96	203	0.43266
10	0.9981671	180	316	0.65340	70	0.9736912	86	196	0.69092
20	0.9971287	149	272	0.44829	74	0.9625664	82	192	1.00000

*The best solution on the Pareto front based on the L_2 norm criterion.

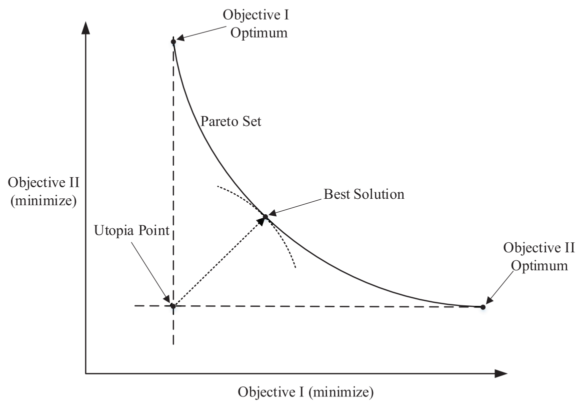

FIGURE 8. Optimal solution using L_2 norm technique.

The main objective in multi-objective optimization problems is to identify the Pareto-optimal set. A secondary goal involves selecting a specific solution from this set. A widely used approach for this purpose is the application of L_p norms, which determine the solution closest to the ideal (utopia) point. The optimal solution can be obtained using the following expression:

$$\text{Minimize}\left(\sum_{i=1}^{numf}\left(func_i(x) - func_i^*\right)^p\right)^{\frac{1}{p}} \tag{13}$$

where $numf$ denotes the number of objective functions, $func_i$ represents the ith objective function, and $func_i^*$ corresponds to the best (ideal) value of the ith objective function among the Pareto-optimal solutions.

In the L_p norm method, the parameter p typically takes values of 1, 2, or ∞. In this study, the L_2 norm ($p = 2$) is applied to determine the optimal solution. This solution, illustrated in Figure 8, represents the point with the minimum Euclidean distance from the utopia point.

In the L_p norm approach, all objective functions are typically expressed in minimization form. Therefore, in this study, the first objective is reformulated as the minimization of system unreliability. Subsequently, the objective functions are normalized using the following equations:

$$func_c = \frac{C(\mathbf{x},\mathbf{y}) - C^{\min}(\mathbf{x},\mathbf{y})}{C^{\max}(\mathbf{x},\mathbf{y}) - C^{\min}(\mathbf{x},\mathbf{y})} \tag{14}$$

$$func_{ur} = \frac{UR(t;\mathbf{x},\mathbf{y}) - UR^{\min}(t;\mathbf{x},\mathbf{y})}{UR^{\max}(t;\mathbf{x},\mathbf{y}) - UR^{\min}(t;\mathbf{x},\mathbf{y})}. \tag{15}$$

In equations (14) and (15), $C^{\min}(\mathbf{x},\mathbf{y})$, $UR^{\min}(t;\mathbf{x},\mathbf{y})$, $C^{\max}(\mathbf{x},\mathbf{y})$, and $UR^{\max}(\mathbf{x},\mathbf{y})$ denote the minimum and maximum values of the system's cost and unreliability among the Pareto-optimal solutions. The best solution within the Pareto set is the one that minimizes $\sqrt{func_c^2 + func_{ur}^2}$.

Some non-dominated solutions of the first and second examples, along with their corresponding L_2 norms, are presented in Tables 3 and 6, respectively. According to the L_2 norm criterion, the best solutions within the Pareto-optimal sets are identified as Solution No. 63 in Table 3 ($L_2 = 0.464399$) for the first example and Solution No. 43 in Table 6 ($L_2 = 0.26931$) for the second example.

5. Conclusion and future research

This study presented a novel multi-objective redundancy allocation problem aimed at simultaneously maximizing system reliability and minimizing total system cost. A mathematical model was developed that accommodates non-identical components and flexible redundancy strategies – active, cold-standby, or mixed – without restrictions on component types or quantities within subsystems. Owing to the NP-hard nature of the problem, an NSGA-II metaheuristic algorithm was employed. The best compromise solution from the obtained Pareto set was then identified using the L_p norm method. The proposed model was evaluated using two numerical examples. In the first, a benchmark problem, the results demonstrated that the proposed multi-objective model can concurrently improve both objectives compared with previous studies. Specifically, the obtained solutions achieved higher reliability and lower cost, which is attributed to the model's flexibility in allowing non-identical component allocation and mixed redundancy strategies across subsystems. A larger-scale example was also examined to illustrate the scalability and efficiency of the proposed algorithm. The results indicated that the algorithm consistently produces nearly identical Pareto frontiers across multiple runs, confirming its robustness and stability. Moreover, the diverse set of non-dominated solutions generated provides decision-makers with a broad range of trade-off options, enhancing the practical applicability of the model in complex system design scenarios.

Future research directions include exploring the impact of component activation sequences on subsystem reliability when non-identical components are involved. Additionally, investigating scenarios in which multiple components can activate simultaneously upon a failure could lead to an extended model for enhanced system reliability. Another avenue for exploration involves determining the optimal number and activation timing of components operating concurrently in standby and mixed redundancy strategies. Extending the model to incorporate repairability also provides a natural direction for future research aimed at optimizing reliability and maintainability jointly. Additionally, future research could explore the allocation of warm and hot standby components within subsystems.

Author contribution statement

Conceptualization: Hadi Gholinezhad and Ali Zeinal-Hamadani; Methodology: Hadi Gholinezhad; Formal analysis and investigation: Hadi Gholinezhad and Ali Zeinal-Hamadani; Writing - original draft preparation: Hadi Gholinezhad; Writing - review and editing: Ali Zeinal-Hamadani; Software: Hadi Gholinezhad; Resources: Hadi Gholinezhad; Supervision: Ali Zeinal-Hamadani.

References

[1] Q.F. He, Z.T. Wang, M.F. Zheng and L. Yan, An artificial bee colony algorithm for multiobjective redundancy allocation problem in repairable system. *Appl. Mech. Mater.* **519** (2014) 1401–1404.

[2] D.E. Fyffe, W.W. Hines and N.K. Lee, System reliability allocation and a computational algorithm. *IEEE Trans. Reliab.* **17** (1968) 64–69.

[3] W.C. Yeh, Y.Z. Su, X.Z. Gao, C.F. Hu, J. Wang and C.L. Huang, Simplified swarm optimization for bi-objection active reliability redundancy allocation problems. *Appl. Soft Comput.* **106** (2021) 107321.

[4] T.P. Talafuse and E.A. Pohl, A bat algorithm for the redundancy allocation problem. *Eng. Optim.* **48** (2016) 900–910.

[5] A. Salmasnia, E. Ameri and S.T. Niaki, A robust loss function approach for a multi-objective redundancy allocation problem. *Appl. Math. Model.* **40** (2016) 635–645.
[6] W.C. Yeh, BAT-based algorithm for finding all Pareto solutions of the series–parallel redundancy allocation problem with mixed components. *Reliab. Eng. Syst. Saf.* **228** (2022) 108795.
[7] S.A. Mousavi, A.Z. Hamadani and H. Gholinezhad, Reliability modeling for degrading systems subject to random direct shocks transmitted between components. *Qual. Reliab. Eng. Int.* **41** (2025) 3736–3745.
[8] D.W. Coit, Cold-standby redundancy optimization for nonrepairable systems. *IIE Trans.* **33** (2001) 471–478.
[9] M.R. Valaei and J. Behnamian, Allocation and sequencing in 1-out-of-N heterogeneous cold-standby systems: multi-objective harmony search with dynamic parameters tuning. *Reliab. Eng. Syst. Saf.* **157** (2017) 78–86.
[10] T.J. Hsieh, Component mixing with a cold standby strategy for the redundancy allocation problem. *Reliab. Eng. Syst. Saf.* **206** (2021) 107290.
[11] G. Levitin, L. Xing and Y. Dai, Heterogeneous non-repairable warm standby systems with periodic inspections. *IEEE Trans. Reliab.* **65** (2015) 394–409.
[12] F. Kayedpour, M. Amiri, M. Rafizadeh and A.S. Nia, Multi-objective redundancy allocation problem for a system with repairable components considering instantaneous availability and strategy selection. *Reliab. Eng. Syst. Saf.* **160** (2017) 11–20.
[13] M.A. Ardakan and A.Z. Hamadani, Reliability optimization of series–parallel systems with mixed redundancy strategy in subsystems. *Reliab. Eng. Syst. Saf.* **130** (2014) 132–139.
[14] E. Zio and H. Gholinezhad, Redundancy allocation of components with time-dependent failure rates. *Mathematics* **11** (2023) 3534.
[15] N. Mahdavi-Nasab, M.A. Ardakan and A. Peiravi, A new model for the reliability-redundancy allocation problem with the mixed redundancy strategy. *J. Stat. Comput. Simul.* **92** (2022) 2956–2979.
[16] N.H. Zavieh, M.A. Ardakan and H. Davari-Ardakani, A new model for the reliability-redundancy allocation problem under the K-mixed redundancy strategy. *J. Stat. Comput. Simul.* **92** (2022) 3542–3560.
[17] H. Gholinezhad, A new model for reliability redundancy allocation problem with component mixing. *Reliab. Eng. Syst. Saf.* **242** (2024) 109815.
[18] D.W. Coit, Maximization of system reliability with a choice of redundancy strategies. *IIE Trans.* **35** (2003) 535–543.
[19] M. Aghaei, A. Zeinal Hamadani and M. Abouei Ardakan, Redundancy allocation problem for k-out-of-n systems with a choice of redundancy strategies. *J. Ind. Eng. Int.* **13** (2017) 81–92.
[20] X. Kong, L. Gao, H. Ouyang and S. Li, Solving the redundancy allocation problem with multiple strategy choices using a new simplified particle swarm optimization. *Reliab. Eng. Syst. Saf.* **144** (2015) 147–158.
[21] M. Abouei Ardakan, M. Sima, A. Zeinal Hamadani and D.W. Coit, A novel strategy for redundant components in reliability–redundancy allocation problems. *IIE Trans.* **48** (2016) 1043–1057.
[22] S. Azizi and M. Mohammadi, Strategy selection for multi-objective redundancy allocation problem in a k-out-of-n system considering the mean time to failure. *Opsearch* **60** (2023) 1021–1044.
[23] H. Gholinezhad and R. Khosravi, A new approach to improve system reliability by eliminating early failures. *J. Data Anal. Intell. Decis.-Making* **1** (2025) 1–12.
[24] C.M. Lai and W.C. Yeh, Two-stage simplified swarm optimization for the redundancy allocation problem in a multi-state bridge system. *Reliab. Eng. Syst. Saf.* **156** (2016) 148–158.
[25] H. Zhang, M. Sun and Y.F. Li, Reliability–redundancy allocation problem in multi-state flow network: minimal cut-based approximation scheme. *Reliab. Eng. Syst. Saf.* **225** (2022) 108564.
[26] Y.F. Li and H. Zhang, The methods for exactly solving redundancy allocation optimization for multi-state series–parallel systems. *Reliab. Eng. Syst. Saf.* **221** (2022) 108340.
[27] Y. Xu, D. Pi, S. Yang and Y. Chen, A novel discrete bat algorithm for heterogeneous redundancy allocation of multi-state systems subject to probabilistic common-cause failure. *Reliab. Eng. Syst. Saf.* **208** (2021) 107338.
[28] J. Li, G. Wang, H. Zhou and H. Chen, Redundancy allocation optimization for multi-state system with hierarchical performance requirements. *Proc. Inst. Mech. Eng. Part O: J. Risk Reliab.* **237** (2023) 1031–1047.
[29] Z. Chen, H. Zhang, X. Wang, J. Yang and H. Dui, Reliability analysis and redundancy design of satellite communication system based on a novel Bayesian environmental importance. *Reliab. Eng. Syst. Saf.* **243** (2024) 109813.
[30] M. Fayyaz and T. Vladimirova, Survey and future directions of fault-tolerant distributed computing on board spacecraft. *Adv. Space Res.* **58** (2016) 2352–2375.
[31] J. Downer, When Failure is an Option: Redundancy, Reliability and Regulation in Complex Technical Systems (No. DP 53). ESRC Centre for Analysis of Risk and Regulation (2009).
[32] I. Moir, A. Seabridge and M. Jukes, Civil Avionics Systems, 2nd edition. Wiley (2019).

[33] Q. Zhang, M. Chen, L. Li and W. Yi, Reliability modeling and evaluation of modular UPS systems in data centers. *IEEE Trans. Reliab.* **67** (2018) 1615–1627.

[34] D. Xun, C. Xin and J. Wenyi, The analysis of GSM-R redundant network and reliability models on high-speed railway, in 2010 International Conference on Electronics and Information Engineering. Vol. 2. IEEE (2010) V2-154.

[35] I.A. Hansen and J. Pachl, Railway Timetabling and Operations: Analysis, Modelling, Optimisation. Eurailpress (2014).

[36] International Electrotechnical Commission, IEC 60601-1: Medical electrical equipment – Part 1: General requirements for basic safety and essential performance. IEC (2012).

[37] B. Han, V. Gopalakrishnan, L. Ji and S. Lee, Network function virtualization: challenges and opportunities for innovations. *IEEE Commun. Mag.* **53** (2015) 90–97.

[38] J.H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. MIT Press (1992).

[39] K. Deb, A. Pratap, S. Agarwal and T.A. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6** (2002) 182–197.

[40] D. Salazar, C.M. Rocco and B.J. Galván, Optimization of constrained multiple-objective reliability problems using evolutionary algorithms. *Reliab. Eng. Syst. Saf.* **91** (2006) 1057–1070.

Appendix A.

Table A.1. Specifications of components for the second numerical example.

Subsystem	Component choice	λ_{ij}	k_{ij}	c_{ij}	w_{ij}
$i=1$	$j=1$	0.014	3	4	2
	$j=2$	0.034	5	4	3
	$j=3$	0.014	2	3	6
	$j=4$	0.052	7	5	4
	$j=5$	0.048	8	2	6
	$j=6$	0.013	2	3	4
	$j=7$	0.021	3	4	2
	$j=8$	0.045	7	5	5
	$j=9$	0.023	4	5	8
	$j=10$	0.071	7	2	4
$i=2$	$j=1$	0.027	3	5	3
	$j=2$	0.017	2	3	3
	$j=3$	0.105	8	2	4
	$j=4$	0.019	3	2	6
	$j=5$	0.029	3	3	4
	$j=6$	0.049	6	4	5
	$j=7$	0.053	5	2	6
	$j=8$	0.046	7	5	8
	$j=9$	0.017	2	5	5
	$j=10$	0.023	3	4	3

TABLE A.1. continued.

Subsystem	Component choice	λ_{ij}	k_{ij}	c_{ij}	w_{ij}
$i = 3$	$j = 1$	0.071	8	2	8
	$j = 2$	0.018	2	2	8
	$j = 3$	0.022	2	5	2
	$j = 4$	0.034	4	2	4
	$j = 5$	0.021	2	5	8
	$j = 6$	0.019	4	5	4
	$j = 7$	0.071	8	3	7
	$j = 8$	0.051	6	4	2
	$j = 9$	0.063	7	5	6
	$j = 10$	0.018	2	4	2
$i = 4$	$j = 1$	0.021	2	2	7
	$j = 2$	0.014	3	3	6
	$j = 3$	0.025	5	2	3
	$j = 4$	0.032	5	2	5
	$j = 5$	0.047	4	3	2
	$j = 6$	0.017	3	2	7
	$j = 7$	0.049	4	5	6
	$j = 8$	0.051	8	3	8
	$j = 9$	0.021	3	2	7
	$j = 10$	0.043	7	4	6
$i = 5$	$j = 1$	0.073	8	5	3
	$j = 2$	0.048	6	3	2
	$j = 3$	0.022	3	2	3
	$j = 4$	0.055	6	3	8
	$j = 5$	0.054	8	5	7
	$j = 6$	0.017	2	2	8
	$j = 7$	0.032	4	3	2
	$j = 8$	0.063	9	4	4
	$j = 9$	0.051	6	4	8
	$j = 10$	0.073	9	2	2
$i = 6$	$j = 1$	0.016	2	2	6
	$j = 2$	0.027	5	3	8
	$j = 3$	0.012	2	2	3
	$j = 4$	0.009	2	4	7
	$j = 5$	0.051	7	5	7
	$j = 6$	0.029	4	4	3
	$j = 7$	0.027	3	3	7
	$j = 8$	0.017	2	4	7
	$j = 9$	0.021	2	3	4
	$j = 10$	0.031	4	3	5
$i = 7$	$j = 1$	0.032	6	4	7
	$j = 2$	0.022	3	3	6
	$j = 3$	0.053	7	4	5
	$j = 4$	0.017	2	5	5
	$j = 5$	0.029	4	3	3
	$j = 6$	0.034	7	4	8
	$j = 7$	0.037	5	4	6
	$j = 8$	0.051	8	3	8
	$j = 9$	0.024	2	2	4
	$j = 10$	0.015	2	3	5
$i = 8$	$j = 1$	0.018	2	3	3
	$j = 2$	0.023	3	3	6
	$j = 3$	0.012	2	2	4
	$j = 4$	0.024	5	3	2
	$j = 5$	0.02	3	5	3
	$j = 6$	0.058	7	2	5
	$j = 7$	0.061	8	2	4
	$j = 8$	0.024	3	2	6
	$j = 9$	0.068	7	3	2
	$j = 10$	0.073	8	2	7

TABLE A.1. continued.

Subsystem	Component choice	λ_{ij}	k_{ij}	c_{ij}	w_{ij}
$i = 9$	$j = 1$	0.049	5	5	7
	$j = 2$	0.068	8	2	7
	$j = 3$	0.024	4	3	5
	$j = 4$	0.019	2	3	7
	$j = 5$	0.024	3	5	5
	$j = 6$	0.049	6	3	5
	$j = 7$	0.035	6	5	5
	$j = 8$	0.025	3	3	6
	$j = 9$	0.049	6	2	8
	$j = 10$	0.039	5	4	5
$i = 10$	$j = 1$	0.028	7	4	2
	$j = 2$	0.048	5	4	2
	$j = 3$	0.019	3	3	4
	$j = 4$	0.015	2	3	4
	$j = 5$	0.018	2	5	5
	$j = 6$	0.019	2	2	3
	$j = 7$	0.058	8	3	8
	$j = 8$	0.016	2	4	4
	$j = 9$	0.051	9	2	6
	$j = 10$	0.065	8	2	3
$i = 11$	$j = 1$	0.033	4	3	3
	$j = 2$	0.027	3	3	7
	$j = 3$	0.035	5	2	8
	$j = 4$	0.009	2	5	8
	$j = 5$	0.038	6	2	3
	$j = 6$	0.035	5	3	3
	$j = 7$	0.029	6	2	3
	$j = 8$	0.022	3	2	3
	$j = 9$	0.058	7	2	8
	$j = 10$	0.062	8	2	2
$i = 12$	$j = 1$	0.022	2	5	6
	$j = 2$	0.041	7	2	6
	$j = 3$	0.037	4	3	4
	$j = 4$	0.023	2	3	4
	$j = 5$	0.032	4	2	4
	$j = 6$	0.059	7	5	3
	$j = 7$	0.038	6	5	4
	$j = 8$	0.054	8	2	7
	$j = 9$	0.057	5	2	3
	$j = 10$	0.015	2	3	8
$i = 13$	$j = 1$	0.047	5	2	7
	$j = 2$	0.019	2	5	5
	$j = 3$	0.053	7	4	8
	$j = 4$	0.039	4	5	3
	$j = 5$	0.019	2	3	2
	$j = 6$	0.048	6	2	6
	$j = 7$	0.023	3	4	8
	$j = 8$	0.034	5	3	7
	$j = 9$	0.072	9	2	8
	$j = 10$	0.051	5	4	3
$i = 14$	$j = 1$	0.019	2	5	5
	$j = 2$	0.038	4	5	7
	$j = 3$	0.045	7	5	6
	$j = 4$	0.033	4	2	7
	$j = 5$	0.045	6	4	2
	$j = 6$	0.024	3	4	5
	$j = 7$	0.023	3	2	4
	$j = 8$	0.048	7	3	3
	$j = 9$	0.024	3	5	8
	$j = 10$	0.041	6	4	7
$i = 15$	$j = 1$	0.021	2	4	2
	$j = 2$	0.041	5	5	3
	$j = 3$	0.043	5	4	8
	$j = 4$	0.025	3	3	3
	$j = 5$	0.012	2	5	6
	$j = 6$	0.039	4	5	4
	$j = 7$	0.047	5	2	6
	$j = 8$	0.013	2	2	6
	$j = 9$	0.015	2	4	4
	$j = 10$	0.044	8	2	3