Scientia Iranica E (2022) 29(6), 3523–3541

SCIENTIA IRANICA

Sharif University of Technology
Scientia Iranica
Transactions E: Industrial Engineering
http://scientiairanica.sharif.edu

Reliability optimization of a k-out-of-n series-parallel system with warm standby components

M. Sharifi[a,b,*], **M. Shahriari**[c], **A. Khajehpoor**[b], **and S.A. Mirtaheri**[b]

a. *Department of Mechanical and Industrial Engineering, Ryerson University, Toronto, ON M5B 2K3, Canada.*
b. *Faculty of Industrial & Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran.*
c. *Faculty of Management & Accounting, South Tehran Branch, Islamic Azad University, Tehran, Iran.*

Received 15 June 2020; received in revised form 26 August 2020; accepted 4 January 2021

KEYWORDS
Redundancy allocation problem;
Warm standby;
Reliability;
Meta-heuristic methods;
Imperfect switch.

Abstract. A new hybrid model for the Redundancy Allocation Problem (RAP) in a series-parallel configuration with the k-out-of-n subsystem is presented in this study. The redundancy policy is set to active, warm standby, or no redundancy in the given model. In a warm standby policy, an imperfect switch detects the component failure and replaces the failed component with a new standby. So, the subsystems redundancy policy is one of the model decision variables. We presented a new RAP objective function for calculating the reliability of a system made up of active and warm-standby subsystems. The presented model seeks to determine the subsystems redundancy policy, i.e., the type and number of redundant components required to maximize system reliability within the constraints of system cost, volume, and weight. To solve the proposed model, we used two Genetic Algorithms (GA) and a Hybrid GA (HGA) meta-heuristic algorithm with local search. Because the RPD% of HGA is 2.1% (on average) better than GA in solving ten large-scale instances, the result demonstrates HGA superiority over GA in solving the presented RAP.

1. Introduction

Due to the competitive market, it has needed to have a more reliable design in recent decades. Nowadays, the term reliability includes reliability requirements, reliability design, reliability prediction, reliability modeling, and retrievals. One of the goals of reliability is to design systems with higher quality during their life cycle. Usually, the system reliability improves through the improvement of the reliability of each component or allocating redundant components. This improvement in practice happens by using better materials, better manufacturing processes, or better design principles. Many research methods have been conducted in reliability improvement according to the system structure, problem type, resolve method, objective function, and components' failure distribution [1]. The system's structure can be series, parallel, k-out-of-n [2], and/or a combination of series and parallel [3]. System reliability can be improved by redundancy allocation [3] or reliability allocation [4]. Exact techniques [2,5], approximate methods [6,7], heuristic methods [8,9], and meta-heuristic methods [9–12] are examples of problem-solving methods. The objective function of the Redundancy Allocation Problem (RAP) is usually considered to maximize the system reliability [3] and minimize the system cost [13]. The components

*. *Corresponding author. Tel.: +1 (647) 532 5776*
E-mail addresses: manisharifi@ryerson.ca (M. Sharifi); Shahriari.Mr@Gmail.com (M. Shahriari); Ahmad_Kh1772@Yahoo.com (A. Khajehpoor); ali.mirtaheri@ryerson.ca (S.A. Mirtaheri)

doi: 10.24200/sci.2021.56113.4591

failure rates can be considered constant (exponential distribution) [14], or time-dependent (i.e., Weibull distribution) [15]. In this paper, we worked on a RAP series-parallel system structure and a k-out-of-subsystems configuration. In this paper, the presented RAP aims to optimize the number and type of the redundant components in each subsystem as well as the redundancy strategy of each subsystem to maximize the system reliability under some constraints.

The RAP is divided into two categories based on the allocated redundant components to the subsystems: RAP without Component Mixing (RAPCM) and RAP with a Mix of Components (RAPMC). The subsystems redundancy strategy includes active and standby, and the standby policy has three different types based on the components characteristics: cold standby, warm standby, and hot standby. Misra and Sharma [16] considered the RAP for a series-parallel system with the k-out-of-n subsystem. They used the active redundancy policy without component mixing in their model. They solved the presented model with binary integer programming. Coit and Smith [17] offered a new model for RAPMC and an active redundancy policy. They considered the series-parallel system structure with a k-out-of-n subsystem.

Coit and Liu [18] presented a new RAPCM model for a series-parallel system with k-out-of-n subsystems. For the first time in their model, they considered active and cold standby redundancy policies simultaneously. They assumed the components with a Constant Failure Rate (CFR) and a non-linear model and converted the model to a binary integer program using variable change. Coit [19] presented a new model in which the redundancy policy was considered a model's decision variable. The variable redundancy policy was active, cold standby, and no redundancy. This paper considered the hot standby systems components to bring the problem close to real-world conditions. Since the RAP in computational time is NP-hard problems, we solved the presented model using the meta-heuristic method. A comparative search of recent research (after 2010) related to RAP is shown in Table 1.

In this paper, we aim to fill the literature gap by considering the warm standby redundancy strategy for a RAPCM. The contribution of the current research is as follows:

- Calculating the system reliability with warm standby components;
- Considering the warm standby redundancy strategy for a RAPCM.

The current research methodology is presented in Figure 1.

The rest of the paper is as follows. Section 2 is the problem definition. Section 3 deals with calculating the system and subsystems reliability with active, warm standby, and no redundancy strategy. In Section 4, the solving methodologies are presented. In Section 5, first some instances are solved to determine the algorithms performance. Then the effect of change on the model parameters is investigated using a sensitivity analysis. Finally, the model and algorithm are validated. Section 6 is the conclusion and further studies.

2. Model description

This section discusses the mathematical model of a RAP with a series-parallel structure and a k-out-of-n subsystem. The identical components can be allocated to each subsystem, and the redundancy strategy is the system variable. In the presented model, the problem objective function is to maximize the system reliability under the system cost, volume, and weight constraints.

2.1. Assumptions

The mathematical model of the above-mentioned RAP is established based on the following assumptions:

- Active and warm-standby redundancy policies are considered for each subsystem;
- Different component types are available to allocate to each subsystem;
- All the allocated components to each subsystem must be the same;
- The components cost, weight, volume, and dependability are constant and predefined;
- Components are binary state and have two working or failed states;
- An imperfect switch detects and replaces the failed components;
- Components are non-repairable;
- Components were CFR and failed independently.

The first assumption is the main novelty of the current research, which fills the literature review gap.

2.2. Nomenclatures

i	Subsystems index $(i = 1, ..., s)$
n_i	Number of allocated components to subsystem i
j	Index of the allocated components to each subsystem, $(j = 1, ..., n_i)$
m_i	Number of available components type for subsystem i
z_i	Index of component type which is allocated to subsystem i, $z_i = (1, ..., m_i)$
$R(t)$	System reliability at time t depending on design vectors z and n

Table 1. Related research studies after 2010.

References	Year	Component type	Objectives	Solving algorithm
Beji et al. [20]	2010	Binary	Single	Hybrid particle swarm optimization
Yeh and Hsieh [21]	2011	Binary	Single	Penalty guided artificial bee colony
Hsieh and You [22]	2011	Binary	Single	Immune-based Algorithm
Chambari et al. [23]	2013	Binary	Single	Simulated annealing
Ardakan and Hamadani [24]	2014	Binary	Single	Modified genetic algorithm
Guilani et al. [25]	2014	Multi	Single	Markov process
Zaretalab et al. [26]	2015	Binary	Multi	Knowledge-based archive simulated annealing
Levitin et al. [27]	2015	Binary	Single	Genetic algorithm
Sharifi et al. [28]	2015	Single	Single	Genetic algorithm, memetic algorithm, simulated annealing, and particle swarm optimization
Lai and Yeh [29]	2016	Multi	Single	Two-stage simplified swarm optimization
Teimouri [30]	2016	Binary	Single	Memory-based electromagnetism-like mechanism
Kim and Kim [31]	2017	Binary	Single	Parallel genetic algorithm
Ghavidel et al. [32]	2018	Binary	Single	LJaya-TVAC algorithm
Ardakan and Rezvan [33]	2018	Binary	Multi	NSGA-II
Tavana et al. [34]	2018	Multi	Multi	NSGA-II
Essadqi et al. [35]	2018	Multi	Multi	Effective oriented GA
Peiravi et al. [36]	2018	Single	Single	Genetic algorithm
Hadipour et al. [37]	2019	Binary	Multi	Multi-objectives water flow algorithm, NSGA-II, and NRGA
Ouyang et al. [38]	2019	Binary	Single	Improved particle swarm optimization
Peiravi et al. [39]	2019	Binary	Single	Genetic algorithm
Huang et al. [40]	2019	Binary	Single	Heuristic survival signature-based approach
Sharifi et al. [41]	2019	Binary	Single	Memetic Algorithm
Sun et al. [42]	2019	Multi	Multi	NSGA-II
Sharifi et al. [43]	2019	Multi	Multi	NSGA-II and NRGA

Table 1. Related research studies after 2010 (continued).

References	Year	Component type	Objectives	Solving algorithm
Yeh [44]	2019	Single	Single	Simplified Swarm Optimization (SSO)
Pourkarim Guilani et al. [45]	2019	Single	Single	Optimization via simulation approach
Juybari et al. [46]	2019	Single	Single	Stochastic fractal search
Sharifi et al. [47]	2020	Multi	Multi	Recursive and genetic algorithms
Sharifi and Taghipour [48]	2020	Binary	Multi	NSGA-II and NRGA
Mellal and Zio [49]	2020	Binary	Binary	Enhanced Nest Cuckoo Optimization Algorithm (ENCOA)
Sharifi et al. [50]	2021	Binary	Single	GA
Borhani-Alamdar and Sharifi [51]	2020	Multi	Single	GA and simulated annealing
Zaretalab et al. [52]	2020	Multi	Single	GA and MA
Current study	2020	Binary	Single	GA and HGA

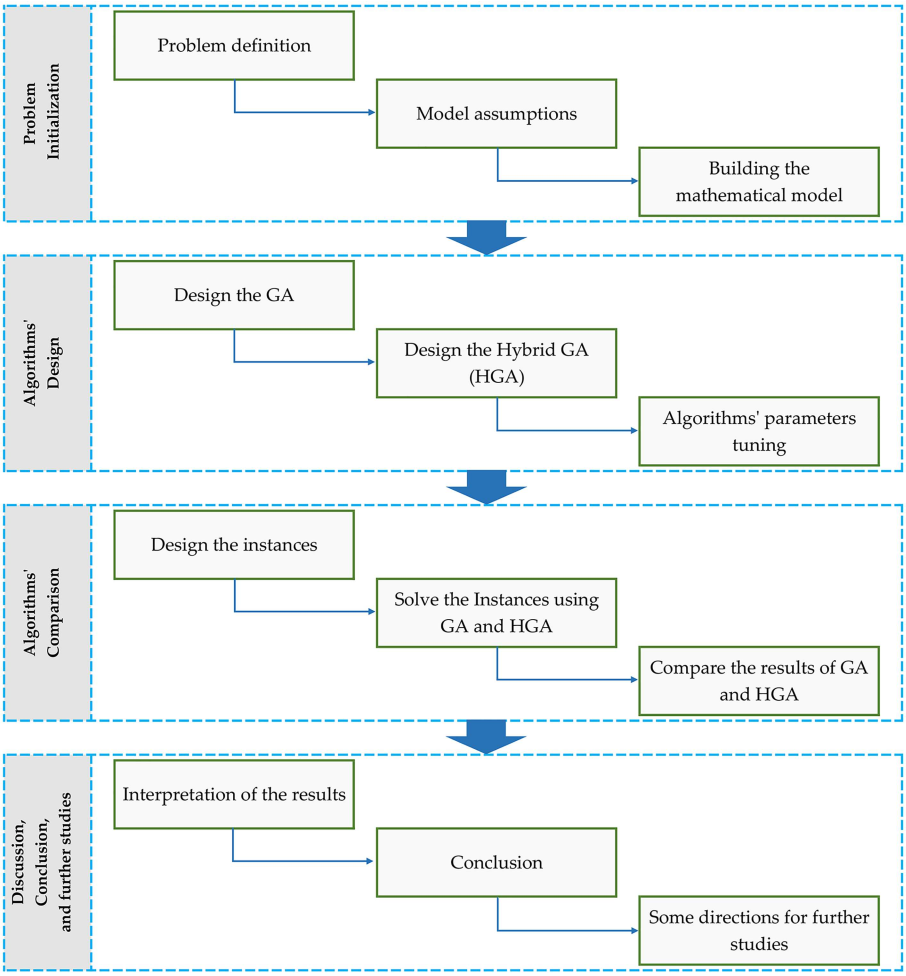

Figure 1. Methodology of the current research (actions, steps, and the order of steps).

k_i	Minimum required number of components for subsystem i
s	Number of system subsystems
t	System's mission time
C, V, W	System-level constraints limits for cost, volume, and weight
$c_{i,j}, w_{i,j}, v_{i,j}$	Cost, volume, and weight of the jth available components type for allocating to subsystem i
$\lambda a_{i,j}$	The failure rate of the jth component's type for allocating to subsystem i, when its working
$\lambda s_{i,j}$	The failure rate of the jth component's type for allocating to subsystem i, when its on standby
ρ_i	Switch success probability at each request for replacement at subsystem i
$R_a(t)$	Reliability of a working component at time t
$R_d(t)$	Reliability of a standby component at time t
$f_a(t)$	p.d.f of a working component at time t
$f_d(t)$	p.d.f of a standby component at time t
$f(t)$	p.d.f of the system at time t
A	Set of all subsystems with active redundancy
S	Set of all subsystems with cold-standby redundancy
N	Set of all subsystems with no redundancy

2.3. Mathematical model

Based on the presented assumptions for the paper, the mathematical RAP model is as follows:

$$\text{Max } R(t) = \Pi_{i=1}^{s} R_i(t, z_i, n_i, k_i), \tag{1}$$

s.t.:

$$\sum_{i=1}^{s} c_{i,z_i} n_i \leq C, \tag{2}$$

$$\sum_{i=1}^{s} v_{i,z_i} n_i \leq V, \tag{3}$$

$$\sum_{i=1}^{s} w_{i,z_i} n_i \leq W, \tag{4}$$

$$n_i \in \{k_i, ..., n_{\max,i}\}, \tag{5}$$

$$z_i \in \{1, ..., m_i\}. \tag{6}$$

Eq.(1) is the model objective function, which aims to maximize system reliability. A description of how to calculate the system reliability will be presented in the next section. Eqs. (2) to (4) are the system cost, volume, and weight constraints. Finally, Eq. (5) determines the minimum and maximum allocated components to each subsystem, and finally, Eq. (6) defines the different available component types for each subsystem.

3. Calculation of the system reliability

If only k component is allocated to a subsystem, the subsystem has no redundancy strategy. If more than k components are allocated to a subsystem, the subsystem can have an active or hot standby redundancy strategy. In this case, the reliability of the subsystem depends on its redundancy strategy. The subsystems reliability with active and warm standby strategies is presented in Subsections 3.1 and 3.3, respectively.

3.1. Subsystems reliability with active redundancy

The reliability of a k-out-of-n subsystem with active redundancy when the components are identical and independent is computed using standard techniques. Therefore, the reliability of ith subsystem with active redundancy is calculated as follows:

$$R_i(t) = \sum_{l=k_i}^{n_i} \binom{n_i}{l} (e^{-\lambda a_{i,zi}.t})^l (1 - e^{-\lambda a_{i,zi}.t})^{n_i - l}. \tag{7}$$

Assume that n_i components of type z_i are allocated to the subsystem i. In Eq. (7), $e^{-\lambda a_{i,zi}.t}$ is the reliability of the component, and $(e^{-\lambda a_{i,zi}.t})^l$ is the probability that l components are working during the mission horizon t. Besides, $(1 - e^{-\lambda a_{i,zi}.t})$ is the failure probability of the components, and $(1 - e^{-\lambda a_{i,zi}.t}))^{n_i - l}$ is the probability that $(n_i - l)$ components fail during the mission horizon t.

3.2. Subsystems reliability with no redundancy

If the model allocates k components to a k-out-of-n subsystem, all k components should start working at the beginning of the mission horizon, and the subsystem has no standby component(s). Therefore, the subsystem has no redundancy strategy. In this case, the subsystem stops working when the first component fails. So, the reliability of the subsystem i, with n_i components of type z_i, is calculated as follows:

$$R_i(t) = (e^{-\lambda a_{i,zi}.t})^{n_i} = (e^{-\lambda a_{i,zi}.t})^{k_i} = e^{-k_i.\lambda a_{i,zi}.t}. \tag{8}$$

3.3. Subsystems reliability with warm standby redundancy

She and Pecht [53] calculated the reliability of a k-out-of-n warm-standby system. In their model, the switching system was perfect. In this paper, a discrete imperfect switch detects the component failure and replaces the failed one with a new one on standby (if it is available). The success probability for each detection and replacement is equal ρ_i. She and Pecht [53] divided the warm standby reliability formula into two parts: fixed coefficients (C-part) and below the integral (I-part):

- **C-part:** The switch starts its function when one of the working components fails, and at least one component is available on standby. When one of the k_i working components fails, the switch failure probability is added to the system probability function. But when one of them $(n_i - k_i)$ on the standby component fails, there is no switch failure probability. Besides, when the system has k_i working components and no component on standby, the switch failure probability is not added to the system probability function. So, the C-part calculates as follows:

$$C\text{-}part = \left[\binom{k_i}{1}\rho_i\lambda a_{i,zi} + \binom{n_i-k_i}{1}\lambda d_{i,zi}\right] \times\left[\binom{k_i}{1}\rho_i\lambda a_{i,zi} + \binom{n_i-k_i-1}{1}\lambda d_{i,zi}\right]\times ... \times\left[\binom{k_i}{1}\rho_i\lambda a_{i,zi} + \binom{1}{1}\lambda d_{i,zi}\right] \times\left[\binom{k_i}{1}\lambda a_{i,zi}\right] = \left[\binom{k_i}{1}\lambda a_{i,zi}\right] \Pi_{i=1}^{n_i-k_i}\left(\rho_i k_i \lambda a_{i,zi} + i\lambda d_{i,zi}\right). \tag{9}$$

- **I-part:** She and Pecht [53] calculated the I-part as follows:

$$I\text{-}part = \int_t^\infty \left[\int_{t_{n_i-k_i=0}}^t \Pi_{i=1}^{n_i-k_i-1} \int_{t_i=0}^{t_i+1} e^{-k_i\lambda a_{i,zi}t - \sum_{j=1}^{n_i-k_i}\lambda d_{i,zi}t_j}\Pi_{i=1}^{n_i-k_i}dt_i\right]dt. \tag{10}$$

With simplification and integration, the I-part is simplified as follows:

$$I\text{-}part = \int_t^\infty e^{-k_i\lambda a_{i,zi}t}\left[\sum_{i=0}^{n_i-k_i}(-1)^i\frac{e^{-i\lambda d_{i,zi}t}}{i!(n_i-k_i-i)!\lambda d_{i,zi}^{n_i-k_i}}\right]dt. \tag{11}$$

Finally, with the integration of Eq. (11), I-part is obtained as follows:

$$I\text{-}part = \frac{1}{\lambda d_{i,zi}^{n_i-k_i}}\sum_{i=0}^{n_i-k_i}\frac{(-1)^i}{i!(n_i-k_i-i)!(k_i\lambda a_{i,zi}+i\lambda d_{i,zi})} e^{-(k_i\lambda a_{i,zi}+i\lambda d_{i,zi})t}. \tag{12}$$

Now, with multiplying the C-part and I-part (Eqs. (9) and (12)), the subsystem reliability is calculated as follows:

$$R_i(t) = (\text{C-part}) \times (\text{I-part}) \rightarrow R_i(t) = \left\{\left[\binom{k_i}{1}\lambda a_{i,zi}\right]\Pi_{i=1}^{n_i-k_i}(\rho_i k_i\lambda a_{i,zi} + i\lambda d_{i,zi})\frac{1}{\lambda d_{i,zi}^{n_i-k_i}}\right\} \times\left\{\sum_{i=0}^{n_i-k_i}\frac{(-1)^i}{i!(n_i-k_i-i)!(k_i\lambda a_{i,zi}+i\lambda d_{i,zi})} e^{-(k_i\lambda a_{i,zi}+i\lambda d_{i,zi})t}\right\}. \tag{13}$$

3.4. System reliability

In a series-parallel system structure, the subsystems are connected serially, when in each subsystem, the components are parallel. So, the system reliability is calculated by multiplying the subsystems reliabilities as follows:

$$R(t) = \Pi_{i\in A}\left\{\sum_{l=k_i}^{n_i}\binom{n_i}{l}(e^{-\lambda a_{i,zi}t})^l(1-e^{-\lambda a_{i,zi}t})^{n_i-l}\right\} \times\Pi_{i\in S}\left\{\left\{\left\{\binom{k_i}{1}\lambda a_{i,zi}\Pi_{i=1}^{n_i-k_i} (\rho_i k_i\lambda a_{i,zi} + i\lambda d_{i,zi})\right\}.\frac{1}{\lambda d_{i,zi}^{n_i-k_i}}\right\} \times\left\{\sum_{i=0}^{n_i-k_i}\frac{(-1)^i}{i!(n_i-k_i-i)!(k_i\lambda a_{i,zi}+i\lambda d_{i,zi})} e^{-(k_i\lambda a_{i,zi}+i\lambda d_{i,zi})t}\right\}\right\} \times \Pi_{i\in N}e^{-k_i\lambda a_{i,zi}t}. \tag{14}$$

4. Solving methods

Chern [54] proved that RAP belongs to the NP-hard category of problems, so we used two metaheuristic algorithms to solve the presented model. The first algorithm is the Genetic Algorithm (GA), a wild application for addressing the RAP (Table 1). The second one is a Hybrid GA (HGA), which combines the GA with a local search to improve the GA's performance.

4.1. Genetic Algorithm (GA)

The GA has a wide range of applicability in different

```
Set Npop, p_c, p_m, p_e and number of algorithm's iteration's Noi
Generate the initial generation,
    Create random individuals,
    Evaluate the individuals
        Calculate individuals' reliability
        Calculate individuals' penalty functions
        Calculate individuals' fitness function
For i = 1 to Noi do
    Procedure: Create the next generation
        Create the pool containing the previous generation individuals
        Operators
            Parents selection: roulette wheel
            Crossover %p_c
            Mutation %p_m
        Procedure: Evaluate the individuals
        Next-generation
            Elitism %p_e
            Roulette wheel selection %(1 − p_e)
End if
Report: The best individual
```

Figure 2. Pseudo-code of the proposed GA.

Subsystem	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Redundancy strategy:	A	S	S	N	N	A	A	N	S	S	N	S	N	A
Components' type:	3	3	4	1	3	3	1	2	3	1	3	3	2	1
Number of allocated component(s):	2	1	4	2	2	3	5	5	2	1	6	4	2	4

Figure 3. A sample for the solution encoding.

engineering optimization problems. This algorithm is a population-based algorithm that starts from an initial population and, with the inspiration of natural genetics, moves to the global optimal solution. GA begins with a set of solutions called the initial population (initial generation), shown through the chromosome structure. Then generate the next generation, using some operators like crossover, mutation, and elitism. The new generations at least have the characteristics of the previous generation. The pseudo-code of the proposed GA is presented in Figure 2.

4.1.1. Solution encoding

Each solution (chromosome) of the presented model is coded as a $3 \times s$ matrix [55]. On this chromosome, s is the number of the system subsystems. The first, second, and third rows of the chromosome represent the type of redundancy strategy, the type of selected components, and the number of allocated components to each subsystem. As for this chromosome and the first row, three choices are available as A: Active strategy, S: Standby strategy, and N: No redundancy strategy. The values of the second row of the chromosome vary from 1 to $m_i; (i = 1, ..., s)$, and the values of the third row vary from k_i to $n_{\max,i}$. A sample of chromosomes for a system with 14 subsystems is shown in Figure 3.

As is presented in Figure 3, the first subsystem redundancy strategy is active, and two components of type 3 are allocated to the subsystem.

4.1.2. Initial population

The initial population is generated randomly.

4.1.3. Fitness function

The objective function of the presented model is to maximize the system reliability. Since the initial population is generated randomly, some of the generated chromosomes are not feasible. We used a penalty function to give a better chance to the feasible solutions for the algorithm operators. The fitness function of the model is presented in Eq. (15) as follows:

$$F = R/(b \times pf). \tag{15}$$

In Eq. (15), F is the chromosome fitness function, R is the chromosome reliability, and pf is the penalty function. The value of pf depends on the cost, volume, and weight of the chromosome and is calculated as follows:

$$pf = \Pi_{i=1}^{3} pf_i = pf_1 \times pf_2 \times pf_3, \tag{16}$$

$$pf_1 = \max\left(\frac{\sum_{i=1}^{s} c_{i,z_i} n_i}{C}, 1\right), \tag{17}$$

$$pf_2 = \max\left(\frac{\sum_{i=1}^{s} v_{i,z_i} n_i}{V}, 1\right), \tag{18}$$

$$pf_2 = \max\left(\frac{\sum_{i=1}^{s} w_{i,z_i} n_i}{W}, 1\right). \tag{19}$$

For a chromosome, if all constraints are satisfied, the chromosome is feasible and $pf_1 = pf_2 = pf_3 = 1$ and the value of the fitness function is equal to the chromosome reliability. But if at least one of the constraints is not satisfied, the chromosome is not feasible. So $pf_1 \times pf_2 \times pf_3 > 1$ and the value of the chromosome fitness function is less than its reliability.

4.1.4. Parents selection strategy

We used a roulette wheel selection strategy for selecting the parents for the operators. This method gives more chances to the chromosomes with better fitness function.

4.1.5. Crossover operator

In this research, we used the uniform crossover. In this type of crossover operator, first we select two chromosomes using a roulette wheel. We then generate a random chromosome whose genomes have a binary random value (e.g., 0 or 1). The size of the random chromosome is equal to the size of the problem chromosomes. For each genome of the chromosome, if the genome value is equal to one, the correspondence genome of the parents replaces each other. The crossover procedure is shown in Figure 4.

4.1.6. Mutation operator

For mutation, one parent is selected using the roulette wheel. Then we generate a random chromosome whose genomes have a real random value between 0 and 1. For each genome, if the genome value is less than a pre-defined value (mutation rate), the corresponding genome in the parent chromosome will mutate. For the first row of the chromosome, the parent genome is equal to N, A, or S. For mutation, each genome will change randomly to two other redundancy strategies. For example, if the redundancy strategy is A, it will be changed randomly to N, or S. The genome value for the second and third rows of the chromosome will be increased or decreased by one unit at random. Figure 5 shows the procedure of the mutation operator.

4.1.7. The algorithm criterion for stopping

The pre-defined maximum generation is the algorithm stopping criteria.

4.2. HGA with adaptive local search

The GA searches through all feasible and insensible solutions at random. In many problems, most of the time, a considerable part of the random initial populations is not feasible. Since one of the most critical factors in GA for finding an optimal (or near-optimal) solution is the quality of the initial population, using a random initial population decreases the chance of finding the right answers. To eliminate these weaknesses, many different methods combine with GA. One of these methods is a local search algorithm that leads the reliability optimization problems to a better result [55]. Local search is a technique to search near the generated random solution to find potential better solutions, so it improves the GA performance. Yun [56] presented the adaptive local search, which searched for the solutions neighborhood in each iteration of the GA. Using adaptive local search decreases the local solution trap in GA and leads GA to the optimal global solution. In this paper, we present the HGA with an adaptive local search for solving the presented RAP.

4.2.1. Adaptive local search scheme

The adaptive local search which we applied in this paper uses the average fitness function values of two consecutive generations as follows:

$$Fvr(g) = \frac{Afv(g)}{Afv(g-1)}, \tag{20}$$

Parent 1:

Subsystem	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	A	S	S	N	N	A	A	N	S	S	N	S	N	A
	3	3	4	1	3	3	1	2	3	1	3	3	2	1
	2	1	4	2	2	3	5	5	2	1	6	4	2	4

Parent 2:

Subsystem	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	S	S	N	A	N	S	S	N	A	S	S	N	A	S
	3	4	1	1	3	3	2	1	4	3	2	2	1	3
	5	2	2	3	2	1	5	5	3	4	7	4	2	4

Random chromosome:

Subsystem	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	0	0	1	0	0	0	1	0	1	0	0	0	1	0
	0	1	1	0	0	0	0	0	1	1	0	1	0	1
	1	0	0	1	1	0	0	0	1	0	0	0	1	0

Offspring 1:

Subsystem	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	A	S	N	N	N	A	S	N	A	S	N	S	A	A
	3	4	1	1	3	3	1	2	4	3	3	2	2	3
	5	1	4	3	2	3	5	5	3	1	6	4	2	4

Offspring 2:

Subsystem	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	S	S	S	A	N	S	A	N	S	S	S	N	N	S
	3	3	4	1	3	3	2	1	3	1	2	3	1	1
	2	2	2	2	2	1	5	5	2	4	7	4	2	4

Figure 4. Uniform crossover operator of the model.

Parent:

	Subsystem													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Parent:	A	S	S	N	N	A	A	N	S	S	N	S	N	A
	3	3	4	1	3	3	1	2	3	1	3	3	2	1
	2	1	4	2	2	3	5	5	2	1	6	4	2	4

	Subsystem													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Random chromosome: Mutation rate = 0.10	0.05	0.94	0.67	0.12	0.19	0.43	0.03	0.86	0.97	0.93	0.21	0.03	0.66	0.23
	0.18	0.21	0.33	0.18	0.19	0.86	0.02	0.05	0.99	0.54	0.48	0.64	0.02	0.18
	0.15	0.93	0.56	0.86	0.21	0.07	0.18	0.21	0.23	0.13	0.05	0.08	0.18	0.21

	Subsystem													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Offspring:	S	S	S	N	N	A	S	N	S	S	N	A	N	A
	3	3	4	1	3	3	2	3	3	1	3	3	1	1
	2	1	4	2	2	4	5	5	2	1	1	3	2	4

Figure 5. Mutation operator.

$$\begin{cases} \text{if } \ Fvr(g) > 1: & \text{Applying GA to local search in the iteration} \\ \text{if } \ Fvr(g) \leq 1: & \text{Only applying GA in the iteration} \end{cases} \quad (21)$$

In Eq. (20), $Afv(g)$ is the average fitness function values of the best population based on an elitist selection strategy at generation g, $Afv(g-1)$ is the average fitness function values of the best population based on an elitist selection strategy at generation $(g-1)$, and $Fvr(g)$ is the fitness function value ratio at generation g.

4.2.2. HGA with local search

In this proposed HGA, we used the hill-climbing (HL) local search method. Firstly, we apply the HL local search for each of the chromosomes selected by the elitist selection strategy for the next generation. The new chromosomes are then obtained from the HL local search algorithm, replaced by the old chromosomes, and moved to the next generation. The HL local search algorithm includes the following steps:

Step 1: Select one of the chromosomes that are selected by the elitist selection strategy for the next generation;

Step 2: Randomly generate some neighborhoods for the selected chromosomes and calculate their fitness function. The number of generated chromosomes neighborhoods is equal to the problem population size;

Step 3: Determine which neighborhood has the best fitness function;

Step 4: If the fitness function of the neighborhood chosen in Step 3 is better than the fitness function of the chosen chromosome, replace the chromosome with the neighborhood and proceed to Step 2;

```
Set n_e = p_e × popsize
For i = 1 to n_e do
        Select individual i by elitism selection strategy
        Set BestInd = Individual i
        Set Best ff = ff_individula i
        For j = 1 to popsize do
                Randomly generate neighborhood j
                Calculate ff_neighborhood j
                If ff_neighborhood j > BestInd
                        BestInd = neighborhood j
                End If
        End For
End For
Move BestInd to the next generation
```

Figure 6. Pseudo-code of the presented HL local search.

Step 5: Repeat Steps 1–4 for each chromosome selected using the elitist selection strategy.

How to generate solution encoding, generate the initial population, parents selection mechanism, calculate fitness function, perform the crossover and mutation operators, selection strategy of the next generation, and stop condition are precise as the presented GA. The pseudo-code of the proposed HL local search is presented in Figure 6.

4.3. Parameters tuning

The results of the metaheuristic algorithms depend on the input parameters. So, we used the response surface methodology [57] for the algorithms parameters tuning. The range of the algorithms parameters is presented in Table 2.

In Table 2, *popsize* defines the algorithms population size, p_c is the crossover probability, p_m is the mutation probability, b is the penalty constant, and *maxgen* is the maximum number of algorithms generations. The optimal values for both algorithms are presented in Table 3.

Table 2. The range of the algorithms parameters.

Parameter	Range	Lower level	Middle level	High level
$popsize$	30–100	30	65	100
p_c	0.60–1.00	0.60	0.80	1.00
p_m	0.01–0.30	0.01	0.155	0.3
b	5–50	5	34.5	50
$maxgen$	20–80	20	45	80

Table 3. The optimum value of the algorithms input parameters.

	Optimal value	
Parameter	**GA**	**HGA**
$popsize$	100	81.45
p_c	1.00	1.00
p_m	0.22	0.30
b	34.50	5.00
$maxgen$	80	61

5. Numerical analysis

Firstly, we solve ten different instances to have a comparison between metaheuristics. Then the effect of changing the parameters of the objective functions is investigated in the sensitivity analysis section. Next, the model and algorithms are validated by comparing them with other research. Finally, some managerial insights are presented.

5.1. Numerical example

For comparison of the proposed algorithms, we used a numerical instance presented by Fyffe et al. [3]. The instance contains a system with a k-out-of-n series-parallel structure and 14 subsystems. In each subsystem, three or four different component types are available. Other instance parameters are presented in Table 4. The probability of switch success is 0.999, and the mission horizon is 100 hours. The maximum number of components for each subsystem is six, and the constraints' right-hand sides are equal to $C = 130$, $V = 110$, and $W = 170$. The number of unique solutions to the problem is 7.996×10^{23}.

The proposed GA and HGA are both coded using MATLAB R2019b. The results of GA and HGA are presented in Tables 5 and 6.

The results in Tables 5 and 6 show the superiority of the HGA in comparison to the GA. To better compare these two algorithms, we selected ten problems from the 33 presented by Nakagawa and Miyazaki [5] and solved them using both algorithms. These problems are quite similar to the solved instance except that the weight constraint (right-hand side of the weight constraint) varies from 166 to 175. Each algorithm is run five times, and then we report the best, the average, and the standard deviation of the system reliability within these runs. The results for these ten instances are presented in Table 7, and Table 8 shows the PDA% of the algorithms.

The result of PDA% in Table 8 shows that HGA has better performance for best-case and average-case results for all instances. The best-case and average-case results of GA are 2.41% and 2.1% (on average) less than HGA, respectively.

To illustrate the significant differences between the results obtained by the proposed HGA and the GA, a two-sample T-test was performed using Minitab 17, and the result is presented in Table 9 and Figure 7.

These results prove that the HGA algorithm is

Table 4. The instance input parameters.

Subsys.[a]		Component type 1					Component type 2					Component type 3					Component type 4				
i	k_i	λa_{i1}	λs_{i1}	c_{i1}	w_{i1}	v_{i1}	λa_{i2}	λs_{i2}	c_{i2}	w_{i2}	v_{i2}	λa_{i3}	λs_{i3}	c_{i3}	w_{i3}	v_{i3}	λa_{i4}	λs_{i4}	c_{i4}	w_{i4}	v_{i4}
1	1	0.001054	0.000100	1	3	5	0.000726	0.000040	1	4	4	0.000943	0.000080	2	2	3	0.000513	0.000025	2	5	2
2	2	0.000513	0.000025	2	8	2	0.000619	0.000032	1	10	1	0.000726	0.000040	1	9	2	–	–	–	–	–
3	1	0.001625	0.000425	2	7	4	0.001054	0.000100	3	5	4	0.001393	0.000708	1	6	2	0.000834	0.000042	4	4	3
4	2	0.001863	0.000538	3	5	3	0.001393	0.000708	4	6	2	0.001625	0.000425	5	4	3	–	–	–	–	–
5	1	0.000619	0.000032	2	4	5	0.000726	0.000040	2	3	4	0.000513	0.000025	3	5	5	–	–	–	–	–
6	2	0.000101	0.000010	3	5	4	0.000202	0.000015	3	4	4	0.000305	0.000020	2	5	3	0.000408	0.000023	2	4	3
7	1	0.000943	0.000080	4	7	3	0.000834	0.000042	4	8	2	0.000619	0.000032	5	9	4	–	–	–	–	–
8	2	0.002107	0.000720	3	4	1	0.001054	0.000100	5	7	1	0.000943	0.000080	6	6	2	–	–	–	–	–
9	3	0.000305	0.000020	2	8	5	0.000101	0.000010	3	9	3	0.000408	0.000023	4	7	4	0.000943	0.000080	3	8	5
10	3	0.001863	0.000550	4	6	3	0.001625	0.000415	4	5	2	0.001054	0.000100	5	6	1	–	–	–	–	–
11	3	0.000619	0.000032	3	5	4	0.000513	0.000025	4	6	3	0.000408	0.000023	5	6	3	–	–	–	–	–
12	1	0.002357	0.000835	2	4	4	0.001985	0.000605	3	5	3	0.001625	0.000708	4	6	4	0.001054	0.000100	5	7	2
13	2	0.000202	0.000015	2	5	5	0.000101	0.000010	3	5	5	0.000305	0.000020	2	6	3	–	–	–	–	–
14	3	0.001054	0.000100	4	6	4	0.000834	0.000042	4	7	2	0.000513	0.000025	5	6	2	0.000101	0.000010	6	9	4

[a]: Subsys.: Subsystem.

Table 5. Results of the GA and HGA.

Subsystem	**GA**			**HGA**		
i	z_i	n_i	**Redundancy strategy**	z_i	n_i	**Redundancy strategy**
1	3	2	Warm standby	3	2	Warm standby
2	1	2	No redundancy	1	2	No redundancy
3	4	2	Warm standby	4	1	No redundancy
4	3	3	Warm standby	3	3	Warm standby
5	1	1	No redundancy	2	1	No redundancy
6	2	2	No redundancy	2	2	No redundancy
7	3	1	No redundancy	2	1	No redundancy
8	1	3	Warm standby	1	3	Warm standby
9	3	3	No redundancy	3	3	No redundancy
10	2	4	Warm standby	2	4	Warm standby
11	1	4	Warm standby	1	4	Warm standby
12	1	2	Warm standby	1	2	Warm standby
13	2	2	No redundancy	2	2	No redundancy
14	3	3	No redundancy	3	4	Warm standby

Table 6. Comparison between the computational results of GA and HA.

Algorithm	**GA**	**HGA**
System reliability	0.4269	0.4403
Resources consumed cost	118	118
Resources consumed weight	170	170
Resources consumed volume	105	101

preferred at a confidence level of 95%. The difference between the obtained results of both algorithms under the statistical test presented in Eq. (22) is investigated. Table 9 shows the results of the T-test for the above comparison. The $P - value = 0.000$ indicates that there is a significant difference between these two algorithms. After normalizing the data, the following typical hypothesis test is run:

$$\begin{cases} \mu_{HGA} = \mu_{GA} \\ \mu_{HGA} \neq \mu_{GA} \end{cases} \tag{22}$$

The box-plot shown in Figure 7 also supports a significant difference between the mean of the results obtained from the HGA algorithm and the GA algorithm.

5.2. Sensitivity analysis

For sensitivity analysis, different values for C, W, and V are considered to investigate the effect of changing these parameters on the optimal system reliability. Since the HGA has superiority in solving the instances, we only solve sensitivity analysis instances using the HGA. Moreover, we consider that the maximum allocatable components for each subsystem is equal to 4.

Regarding the system cost (C), the system weight

Table 7. Results for the ten instances.

		GA			**HGA**		
Problem	W	**Best**	**Average**	**SD**	**Best**	**Average**	**SD**
1	166	0.3913	0.3828	0.0081	0.3975	0.3907	0.0085
2	167	0.3974	0.3942	0.0031	0.4108	0.4025	0.0091
3	168	0.4172	0.4125	0.0081	0.4211	0.4156	0.0064
4	169	0.4219	0.4199	0.0030	0.4355	0.4283	0.0068
5	170	0.4269	0.4221	0.0044	0.4403	0.4395	0.0014
6	171	0.4331	0.4262	0.0070	0.4499	0.4432	0.0060
7	172	0.4468	0.4423	0.0040	0.4547	0.4475	0.0063
8	173	0.4611	0.4591	0.0018	0.4713	0.4656	0.0057
9	174	0.4656	0.4642	0.0013	0.4765	0.4692	0.0084
10	175	0.4705	0.4664	0.0037	0.4816	0.4799	0.0024

Table 8. PDA% of the algorithms.

Problem	W	GA		HGA	
		Best	Average	Best	Average
1	166	1.56	2.02	**0.00**	**0.00**
2	167	3.26	2.06	**0.00**	**0.00**
3	168	0.93	0.75	**0.00**	**0.00**
4	169	3.12	1.96	**0.00**	**0.00**
5	170	3.04	3.96	**0.00**	**0.00**
6	171	3.73	3.84	**0.00**	**0.00**
7	172	1.74	1.16	**0.00**	**0.00**
8	173	2.16	1.40	**0.00**	**0.00**
9	174	2.29	1.07	**0.00**	**0.00**
10	175	2.30	2.81	**0.00**	**0.00**
Average		2.41	2.10	**0.00**	**0.00**

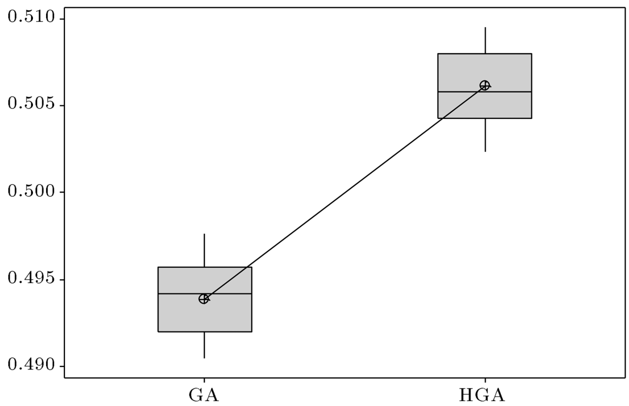

Figure 7. Box plots of the statistical test on HGA and GA performance.

and volume constraints are relaxed, and the value of C increases from 130 to 220 by steps of 10. The results are presented in Table 10.

In Table 10, the system with $C = 130$ is considered the main system, and for other values of C, the changes are highlighted as **bold** and underlined letters and numbers. When the value of C increases, firstly, the model allocates more components to the subsystems with the minimum allocated components (i.e., the subsystem with $n = k$). The model increases the number of allocated components for each subsystem. When $C = 180$, all subsystems have four components, which is the maximum allocatable component for each subsystem. In this case, the redundancy strategy of all subsystems is changed to warm standby. After that, by increasing the value of C, the model allocates the components with better performance. Thus, by an increase in the value of C from 190 to 220, only the types of the components were changed. By increasing the value of C from 130 to 220, the system reliability increases from 0.5039 to 0.7643, which shows a 51.77% increase.

Regarding the system weight (W), the system cost and volume constraints are relaxed, and the value of W increases from 170 to 350 by steps of 20. The results are presented in Table 11.

In Table 11, the system with $W = 170$ is considered the main system, and for other values of W, the changes are highlighted as **bold** and underlined letters and numbers. When the value of W increases, the model allocates more components to the subsystems with the minimum allocated components (i.e., the subsystem with $n = k$). The model increases the number of allocated components for each subsystem. When $C = 290$, all subsystems have four components, which is the maximum allocatable component for each subsystem. In this case, the redundancy strategy of all subsystems is changed to warm standby. After that, by increasing the value of W, the model allocates the components with better performance. Thus, by an increase in the value of W from 290 to 350, only the types of the components were changed. By increasing the value of W from 170 to 290, the system's reliability increases from 0.4403 to 0.7626, which shows a %73.20 increase.

Regarding the system volume (V), the system cost and weight constraints are relaxed, and the value of V increases from 110 to 200 by steps of 20. The results are presented in Table 12.

In Table 12, the system with $V = 110$ is considered the main system, and for other values of V, the changes are highlighted as **bold** and underlined letters and numbers. When the value of V is equal to 110, the system allocates the components with the highest performance to each subsystem. So, by increasing the value of V, the components type doesn't change, and only the number of allocated components to each subsystem increases. By increasing the value of V from 110 to 180, the system reliability increases from 0.6286 to 0.7741, which shows a 23.14% increase.

The results of the sensitivity analysis demonstrated that the system is more sensitive to the value of W then to the value of C or finally to the value of V.

Table 9. Two-sample T-test for HGA and GA performance.

Algorithm	Number of test problem	Mean	Standard deviation	Degree of freedom	*T-value*	*P-value*
HGA	10	0.49388	0.00223	18	–12.25	0.000
GA	10	0.50612	0.00223			

Table 10. Sensitivity analysis of the system's available budget (C).

No.	C		Subsystems														System reliability
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	130	z	3	1	4	3	1	2	2	1	2	2	1	1	2	3	0.5039
		n	2	2	2	3	2	2	2	4	3	4	4	3	2	4	
		S	W	N	A	W	A	N	W	W	N	W	W	W	N	W	
2	140	z	3	1	4	3	1	2	2	1	2	2	1	1	2	3	0.5386
		n	**3**	**3**	2	3	2	**3**	2	4	3	4	4	3	**3**	4	
		S	W	**W**	A	W	A	**A**	W	W	N	W	W	W	**A**	W	
3	150	z	**2**	1	4	3	1	2	2	1	**1**	2	1	1	2	3	0.5580
		n	**3**	**4**	**3**	3	**3**	**3**	**3**	4	**4**	4	4	**4**	**3**	4	
		S	W	**W**	A	W	A	**A**	W	W	**W**	W	W	W	**A**	W	
4	160	z	**2**	1	4	3	1	2	2	1	**1**	2	1	1	2	3	0.5631
		n	**4**	**4**	**3**	3	**4**	**3**	**4**	4	**4**	4	4	**4**	**4**	4	
		S	W	**W**	A	W	A	**A**	W	W	**W**	W	W	W	**W**	W	
5	170	z	**2**	1	4	3	1	2	2	1	**1**	2	1	1	2	3	0.5954
		n	**4**	**4**	**4**	**4**	**4**	**3**	**4**	4	**4**	4	4	**4**	**4**	4	
		S	W	**W**	**W**	W	A	**A**	W	W	**W**	W	W	W	**W**	W	
6	180	z	**4**	1	4	3	1	2	2	1	2	2	1	1	2	3	0.6209
		n	**4**	**4**	**4**	**4**	**4**	**4**	**4**	4	**4**	4	4	**4**	**4**	4	
		S	W	**W**	**W**	W	**W**	**A**	W	W	**W**	W	W	W	**W**	W	
7	190	z	**4**	1	4	3	3	1	3	1	2	2	1	1	2	3	0.6333
		n	**4**	**4**	**4**	**4**	**4**	**4**	**4**	4	**4**	4	4	**4**	**4**	4	
		S	W	**W**	**W**	W	**W**	**W**	W	W	**W**	W	W	W	**W**	W	
8	200	z	**4**	1	4	3	**3**	**1**	**3**	**2**	2	2	1	1	2	**4**	0.6847
		n	**4**	**4**	**4**	**4**	**4**	**4**	**4**	4	**4**	4	4	**4**	**4**	4	
		S	W	**W**	**W**	W	**W**	**W**	W	W	**W**	W	W	W	**W**	W	
9	210	z	**4**	**1**	4	3	**3**	**1**	**3**	**3**	2	**3**	1	1	2	**4**	0.7219
		n	**4**	**4**	**4**	**4**	**4**	**4**	**4**	4	**4**	4	4	**4**	**4**	4	
		S	W	**W**	**W**	W	**W**	**W**	W	W	**W**	W	W	W	**W**	W	
10	220	z	**4**	1	4	3	**3**	**1**	**3**	**3**	2	**3**	1	**4**	2	**4**	0.7643
		n	**4**	**4**	**4**	**4**	**4**	**4**	**4**	4	**4**	4	4	**4**	**4**	4	
		S	W	**W**	**W**	W	**W**	**W**	W	W	**W**	W	W	W	**W**	W	

5.3. Model and algorithms validation

For model validation, we relaxed the volume constraint and reduced the switch success probability to 0.99. Then we multiply the values of the components warm standby failure rate by γ and reduce the value of γ from one to zero by steps of 0.2. Changing the value of γ does not affect the number and type of the allocated components to each subsystem as well as the redundancy strategy of each subsystem. Only the value of the system reliability increased smoothly as we

Table 11. Sensitivity analysis of the system maximum acceptable weight (W).

No.	W		Subsystems 1	2	3	4	5	6	7	8	9	10	11	12	13	14	System reliability
1	170	n	3	1	4	3	2	2	2	1	3	2	1	1	2	3	0.4403
		z	2	2	1	3	1	2	1	3	3	4	4	2	2	4	
		S	W	N	N	W	N	N	N	W	N	W	W	W	N	W	
2	190	n	3	1	4	3	2	2	2	1	3	2	1	1	2	3	0.4696
		z	**4**	2	**2**	3	**2**	2	**2**	3	3	4	4	2	2	4	
		S	W	N	**W**	W	**W**	N	**A**	W	N	W	W	W	N	W	
3	210	n	3	1	4	3	2	2	2	1	3	2	1	**2**	2	3	0.5006
		z	**4**	2	**3**	3	**2**	**3**	**2**	3	3	4	4	**3**	**3**	4	
		S	W	N	**W**	W	**W**	**W**	**A**	W	N	W	W	W	**A**	W	
4	230	n	3	1	4	3	2	2	2	1	3	2	1	2	2	3	05319
		z	**4**	**4**	**3**	3	**3**	**3**	**2**	3	3	4	4	**3**	**3**	4	
		S	W	**W**	**W**	W	**W**	**W**	**A**	W	N	W	W	W	**A**	W	
5	250	n	3	1	4	3	3	2	2	1	3	2	1	2	2	3	0.5759
		z	**4**	**4**	**3**	**4**	**3**	**3**	**3**	**4**	3	4	4	**3**	**3**	4	
		S	W	**W**	**W**	W	**W**	**W**	**W**	W	N	W	W	W	**A**	W	
6	270	n	3	1	4	3	**3**	2	2	1	3	2	**2**	**2**	2	3	0.6210
		z	**4**	**4**	**3**	**4**	**3**	**4**	**3**	**4**	**4**	4	4	**3**	**4**	4	
		S	W	**W**	**W**	W	**W**	**W**	**W**	W	**W**	W	W	W	**W**	W	
7	290	n	**3**	**1**	4	3	3	2	2	1	3	2	**2**	**2**	2	3	0.6310
		z	**4**	**4**	**4**	**4**	**3**	**4**	**4**	**4**	**4**	4	4	4	**4**	4	
		S	W	**W**	**W**	W	**W**	**W**	**W**	W	**W**	W	W	W	**W**	W	
8	310	n	**4**	1	4	3	**3**	2	2	1	**1**	2	**2**	**2**	2	3	0.6412
		z	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**	4	4	4	**4**	4	
		S	W	**W**	**W**	W	**W**	**W**	**W**	W	**W**	W	W	W	**W**	W	
9	330	n	**4**	1	4	3	**3**	**1**	**3**	**3**	**2**	2	**2**	**2**	2	3	0.6952
		z	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**	4	4	**4** **4**	4		
		S	W	**W**	**W**	W	**W**	**W**	**W**	W	**W**	W	W	W	**W**	W	
10	350	n	**4**	**1**	4	3	**3**	**1**	**3**	**3**	**2**	**3**	**3**	**4**	2	3	0.7626
		z	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**	4	4	**4**	**4**	4	
		S	W	**W**	**W**	W	**W**	**W**	**W**	W	**W**	W	W	W	**W**	W	

expected. The system reliability for different values of γ is presented in Table 13.

The system reliability for $\gamma = 0$ is equal to 0.4505. when the value of γ is equal to zero. The standby components failure rates are equal to zero, so the model is turned into a system with cold standby components. The result for $\gamma = 0$ in terms of the subsystems allocated components, the type of the allocated components to each subsystem, the redundancy strategy of the subsystems, and the system reliability, the result

Table 12. Sensitivity analysis on the system maximum acceptable volume (V).

No.	V		Subsystems **1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**System reliability**
1	110	n	4	1	4	3	3	1	3	3	2	3	3	4	2	4	0.6286
		z	2	2	2	3	2	2	2	3	3	4	4	2	2	4	
		S	W	N	A	W	A	N	W	W	N	W	W	W	N	W	
2	120	n	4	1	4	3	3	1	3	3	2	3	3	4	2	4	0.6391
		z	2	2	2	3	2	**3**	2	3	3	4	4	2	**3**	4	
		S	W	N	A	W	A	**W**	W	W	N	W	W	W	**W**	W	
3	130	n	4	1	4	3	3	1	3	3	2	3	3	4	2	4	0.7047
		z	2	**3**	**3**	3	2	**3**	**3**	3	3	4	4	**3**	**3**	4	
		S	W	**W**	**W**	W	A	**W**	W	W	N	W	W	W	**W**	W	
4	140	n	4	1	4	3	3	1	3	3	2	3	3	4	2	4	0.7269
		z	**3**	**3**	**3**	3	**3**	**3**	**3**	3	3	4	4	**4**	**3**	4	
		S	W	**W**	**W**	W	**W**	**W**	W	W	N	W	W	W	**W**	W	
5	150	n	4	1	4	3	3	1	3	3	2	3	3	4	2	4	0.7443
		z	**4**	**4**	**3**	3	**3**	**4**	**3**	3	**4**	4	4	**4**	**3**	4	
		S	W	**W**	**W**	W	**W**	**W**	W	W	**W**	W	W	W	**W**	W	
6	160	n	4	1	4	3	3	1	3	3	2	3	3	4	2	4	0.7674
		z	**4**	**4**	**3**	**4**	**4**	**4**	**3**	**4**	**4**	4	4	**4**	**3**	4	
		S	W	**W**	**W**	W	**W**	**W**	W	W	**W**	W	W	W	**W**	W	
7	170	n	4	1	4	3	3	1	3	3	2	3	3	4	2	4	0.7732
		z	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**	4	4	**4**	**3**	4	
		S	W	**W**	**W**	W	**W**	**W**	W	W	**W**	W	W	W	**W**	W	
8	180	n	4	1	4	3	3	1	3	3	2	3	3	4	2	4	0.7741
		z	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**	4	4	**4**	**4**	4	
		S	W	**W**	**W**	W	**W**	**W**	W	W	**W**	W	W	W	**W**	W	

Table 13. System's reliability for different values of γ.

γ	1.00	0.80	0.60	0.40	0.20	0.00
System reliability	0.4303	0.4440	0.4467	0.4489	0.4499	0.4505

is the same as the results of Aghaei et al. [58]. It shows the presented RAP ability to deal with warm and cold standby components and demonstrates the solving methodologies are precisely designed.

Moreover, Table 13 shows that the presented model is applicable to cold and warm standby components simultaneously. For this reason, and for the subsystems with cold standby components, the warm standby failure rates should be set to zero.

5.4. Managerial insights

The presented model will help the managers and system designers optimize the redundant systems in terms of reliability when the components are warm. Using the results of the presented models leads the managers to operate the systems at a lower cost and the system designers to a beneficial trade-off between the system's reliability and cost. The systems that use warm standby components like batteries (i.e., UPSs) and radioactive components (i.e., nuclear power plants and nuclear submarines) and the electricity transmission systems may use the result of the presented model to design and operate more reliable systems.

The results of Tables 10–12 show that increasing the right-hand-side of the model constraints first leads the model to allocate more components to the subsystems. Then the models use the components with higher performance to increase the system reliability. It means that the model is more sensitive to the number of subsystems components than to their type.

6. Conclusions and recommendations for future research

In most of the research conducted on Redundancy Allocation Problem (RAP), the subsystems components are considered cold standby. But in real-world systems such as UPSs, nuclear power plants, and nuclear submarines, the components are in warm standby. So, it is essential to present the new practical models to figure out the reliability of these systems. This paper presents a new Hybrid GA (HGA) for solving the RAP without Component Mixing (RAPCM) with k-out-of-n subsystems configuration and warm standby components with Constant Failure Rate (CFR). In this model, the redundancy of the subsystems was considered as the model decision variable. Since the proposed model is an NP-hard non-linear programming model, we solve the presented model with an HGA and compare the results with a Genetic Algorithm (GA). The results show the superiority of the HGA compared to GA, and the HGA achieves results on average 2.1% better than GA in terms of the system reliability for ten different large-scale problems. Moreover, the results show that the model is more sensitive to the number of the allocated components to the subsystem compared to the type of the allocated components. By changing the values of the warm standby components failure rates, we showed that the presented model is applicable for systems with cold and warm standby components simultaneously.

Future studies may have two directions. The first direction deals with the model assumptions. Considering the systems with repairable components makes the problem more realistic. Besides, the structure of the current research may apply to a RAP with a Mix of Components (RAPMC). Finally, considering the multi-state warm standby components is a proper way to draw the problem close to real-world conditions. The second direction is using different solving methodologies. Considering the multi-objective RAP with the current assumptions brings more options for decision-makers.

References

1. Kuo, W., Prasad, V.R., Tillman, F.A., et al. "Optimal reliability design: fundamentals and applications", Cambridge University Press (2001).
2. Pham, H. "Optimal design of k-out-of-n redundant systems", *Microelectronics Reliability*, **32**(1–2), pp. 119–126 (1992).
3. Fyffe, D.E., Hines, W.W., and Lee, N.K. "System reliability allocation and a computational algorithm", *IEEE Transactions on Reliability*, **17**(2), pp. 64–69 (1968).
4. Yalaoui, A., Chu, C., and Chatelet, E. "Reliability allocation problem in a series-parallel system", *Reliability Engineering & System Safety*, **90**(1), pp. 55–61 (2005).
5. Nakagawa, Y. and Miyazaki, S. "Surrogate constraints algorithm for reliability optimization problems with two constraints", *IEEE Transactions on Reliability*, **30**(2), pp. 175–180 (1981).
6. Tillman, F.A., Hwang, C.L., and Kuo, W. "System effectiveness models: an annotated bibliography", *IEEE Transactions on Reliability*, **29**(4), pp. 295–304 (1980).
7. Prasad, V.R. and Kuo, W. "Reliability optimization of coherent systems", *IEEE Transactions on Reliability*, **49**(3), pp. 323–330 (2000).
8. You, P.S. and Chen, T.C. "An efficient heuristic for series-parallel redundant reliability problems", *Computers & Operations Research*, **32**(8), pp. 2117–2127 (2005).
9. Coit, D.W. and Smith, A.E. "Reliability optimization of series-parallel systems using a genetic algorithm", *IEEE Transactions on Reliability*, **45**(2), pp. 254–260 (1996).
10. Kulturel-Konak, S., Smith, A.E., and Coit, D.W. "Efficiently solving the redundancy allocation problem using tabu search", *IIE Transactions*, **35**(6), pp. 515–526 (2003).
11. Kim, H.G., Bae, C.O., and Park, S.Y. "Simulated annealing algorithm for redundancy optimization with multiple component choices", *Advanced Reliability Modeling, Proceedings of the 2004 Asian International Workshop (AIWARM 2004)*, Hiroshima, Japan, 26 –27 August, pp. 237–244 (2004).
12. Liang, Y.C. and Smith, A.E. "An ant colony optimization algorithm for the redundancy allocation problem (RAP)", *IEEE Transactions on Reliability*, **53**(3), pp. 417–423 (2004).
13. Coit, D.W. and Smith, A.E. "Solving the redundancy allocation problem using a combined neural network genetic algorithm approach", *Computers & Operations Research*, **23**(6), pp. 515–526 (1996).

14. Coit, D.W. and Smith, A.E. "Stochastic formulations of the redundancy allocation problem", In *Proceedings of the Fifth Industrial Engineering Research Conference*, Minneapolis (1996).

15. Amari, S.V. and Dill, G. "A redundancy optimization problem with warm-standby redundancy", In *2010 Proceedings of IEEE Annual Reliability and Maintainability Symposium (RAMS)* (2010).

16. Misra, K.B. and Sharma, U. "An efficient algorithm to solve integer-programming problems arising in system-reliability design", *IEEE Transactions on Reliability*, **40**(1), pp. 81–91 (1991).

17. Coit, D.W. and Smith, A.E. "Optimization approaches to the redundancy allocation problem for series-parallel systems", In *Fourth Industrial Engineering Research Conference Proceedings* (1995).

18. Coit, D.W. and Liu, J.C. "System reliability optimization with k-out-of-n subsystems", *International Journal of Reliability, Quality and Safety Engineering*, **7**(02), pp. 129–142 (2000).

19. Coit, D.W. "Maximization of system reliability with a choice of redundancy strategies", *IIE Transactions*, **35**(6), pp. 535–543 (2003).

20. Beji, N., Jarboui, B., Eddaly, M., et al. "A hybrid particle swarm optimization algorithm for the redundancy allocation problem", *Journal of Computational Science*, **1**(3), pp. 159–167 (2010).

21. 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).

22. 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).

23. Chambari, A., Najafi, A.A., Rahmati, S.H.A., et al. "An efficient simulated annealing algorithm for the redundancy allocation problem with a choice of redundancy strategies", *Reliability Engineering & System Safety*, **119**, pp. 158–164 (2013).

24. Ardakan, M.A. and Hamadani, A.Z. "Reliability-redundancy allocation problem with cold-standby redundancy strategy", *Simulation Modelling Practice and Theory*, **42**, pp. 107–118 (2014).

25. Guilani, P.P., Sharifi, M., Niaki, S., et al. "Reliability evaluation of non-reparable three-state systems using Markov model and its comparison with the UGF and the recursive methods", *Reliability Engineering & System Safety*, **129**, pp. 29–35 (2014).

26. Zaretalab, A., Hajipour, V., Sharifi, M., et al. "A knowledge-based archive multi-objective simulated annealing algorithm to optimize series-parallel system with choice of redundancy strategies", *Computers & Industrial Engineering*, **80**, pp. 33–44 (2015).

27. Levitin, G., Xing, L., Peng, S., et al. "Optimal choice of standby modes in 1-out-of-N system with respect to mission reliability and cost", *Applied Mathematics and Computation*, **258**, pp. 587–596 (2015).

28. Sharifi, M., Cheragh, G., Maljaii, K.D., et al. "Reliability optimization of a series-parallel k-out-of-n system with failure rates depends on working components of system", *International Journal of Industrial Engineering*, **22**(4), pp. 438–453 (2015).

29. Lai, C.-M. and Yeh, W.-C. "Two-stage simplified swarm optimization for the redundancy allocation problem in a multi-state bridge system", *Reliability Engineering & System Safety*, **156**, pp. 148–158 (2016).

30. Teimouri, M., Zaretalab, A., Niaki, S., et al. "An efficient memory-based electromagnetism-like mechanism for the redundancy allocation problem", *Applied Soft Computing*, **38**, pp. 423–436 (2016).

31. 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).

32. Ghavidel, S., Azizivahed, A., and Li, L. "A hybrid Jaya algorithm for reliability-redundancy allocation problems", *Engineering Optimization*, **50**(4), pp. 698–715 (2018).

33. Ardakan, M.A. and Rezvan, M.T. "Multi-objective optimization of reliability-redundancy allocation problem with cold-standby strategy using NSGA-II", *Reliability Engineering & System Safety*, **172**, pp. 225–238 (2018).

34. Tavana, M., Khalili-Damghani, K., Di Caprio, D., et al. "An evolutionary computation approach to solving repairable multi-state multi-objective redundancy allocation problems", *Neural Computing and Applications*, **30**(1), pp. 127–139 (2018).

35. Essadqi, M., Idrissi, A., and Amarir, A. "An Effective oriented genetic algorithm for solving redundancy allocation problem in multi-state power systems", *Procedia Computer Science*, **127**, pp. 170–179 (2018).

36. Peiravi, A., Karbasian, M., and Abouei Ardakan, M. "K-mixed strategy: A new redundancy strategy for reliability problems. Proceedings of the Institution of Mechanical Engineers", *Part O: Journal of Risk and Reliability*, **232**(1), pp. 38–51 (2018).

37. Hadipour, H., Amiri, M., and Sharifi, M. "Redundancy allocation in series-parallel systems under warm standby and active components in repairable subsystems", *Reliability Engineering & System Safety*, **192**, 106048 (2019).

38. 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).

39. Peiravi, A., Karbasian, M., Ardakan, M.A., et al. "Reliability optimization of series-parallel systems with K-mixed redundancy strategy", *Reliability Engineering & System Safety*, **183**, pp. 17–28 (2019).

40. Huang, X., Coolen, F.P., and Coolen-Maturi, T. "A heuristic survival signature based approach for reliability-redundancy allocation", *Reliability Engineering & System Safety*, **185**, pp. 511–517 (2019).

41. Sharifi, M., Shahriari, M., and Zaretalab, A. "The effects of technical and organizational activities on redundancy allocation problem with choice of selecting redundancy strategies using the memetic algorithm", *International Journal of Industrial Mathematics*, **11**(3), pp. 165–176 (2019).

42. Sun, M.X., Li, Y.F., and Zio, E. "On the optimal redundancy allocation for multi-state series-parallel systems under epistemic uncertainty", *Reliability Engineering & System Safety*, **192**, 106019 (2019).

43. Sharifi, M., Moghaddam, T.A., and Shahriari, M. "Multi-objective redundancy allocation problem with weighted-k-out-of-n subsystems", *Heliyon*, **5**(12), e02346 (2019).

44. Yeh, W.C. "Solving cold-standby reliability redundancy allocation problems using a new swarm intelligence algorithm", *Applied Soft Computing*, **83**, 105582 (2019).

45. Pourkarim Guilani, P., Azimi, P., Sharifi, M., et al. "Redundancy allocation problem with a mixed strategy for a system with k-out-of-n subsystems and time-dependent failure rates based on Weibull distribution: An optimization via simulation approach", *Scientia Iranica*, **26**(2), pp. 1023–1038 (2019).

46. 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).

47. 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).

48. 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. 1–17 (2020).

49. Mellal, M.A. and Zio, E. "System reliability-redundancy optimization with cold-standby strategy by an enhanced nest cuckoo optimization algorithm", *Reliability Engineering & System Safety*, **201**, 106973 (2020).

50. Sharifi, M., Cheragh, G., Maljaii, K.D., et al. "Reliability and cost optimization of a system with k-out-of-n configuration and choice of decreasing the components failure rates", *Scientia Iranica*, **28**(6), pp. 3602–3616 (2021).

51. Borhani Alamdari, A.H. and Sharifi, M. "Solving a joint availability-redundancy optimization model with multi-state components with meta-heuristic", *International Journal of Industrial Mathematics*, **12**(1), pp. 59–70 (2020).

52. Zaretalab, A., Hajipour, V., and Tavana, M. "Redundancy allocation problem with multi-state component systems and reliable supplier selection", *Reliability Engineering & System Safety*, **193**, 106629 (2020).

53. She, J. and Pecht, M.G. "Reliability of a k-out-of-n warm-standby system", *IEEE Transactions on Reliability*, **41**(1), pp. 72–75 (1992).

54. Chern, M.S. "On the computational complexity of reliability redundancy allocation in a series system", *Operations Research Letters*, **11**(5), pp. 309–315 (1992).

55. Tavakkoli-Moghaddam, R., Safari, J., and Sassani, F. "Reliability optimization of series-parallel systems with a choice of redundancy strategies using a genetic algorithm", *Reliability Engineering & System Safety*, **93**(4), pp. 550–556 (2008).

56. Yun, Y. "Hybrid genetic algorithm with adaptive local search scheme", *Computers & Industrial Engineering*, **51**(1), pp. 128–141 (2006).

57. Montgomery, D.C., *Design and Analysis of Experiments*, 6th Edn., John Wiley and Sons New York, NY (2005).

58. Aghaei, M., Hamadani, A.Z., and Ardakan, M.A. "Redundancy allocation problem for k-out-of-n systems with a choice of redundancy strategies", *Journal of Industrial Engineering International*, **13**(1), pp. 81–92 (2017).

Biographies

Mani Sharifi was born in 1973 in Tehran, Iran. Mani is a post-doctoral research fellow at Ryerson University, Faculty of Computer Science, and the Reliability, Risk, and Maintenance Research Laboratory (RRMR Lab). He holds a BSc degree from Qazvin Islamic Azad University, an MSc degree from the south Tehran branch of Islamic Azad University, and a PhD degree from Tehran Research and Science Islamic Azad University in industrial engineering. He was the Managerial editor of the Journal of Optimization in Industrial Engineering. He joined Ryerson University in November 2018 as a researcher and instructor for grad courses. His area of interest includes reliability engineering, combinatorial optimization, statistical optimization, and production scheduling. He has published a number of papers in journals such as Reliability Engineering and Safety System, Computers and Industrial Engineering, Applied Soft Computing, Journal of Manufacturing Systems, Journal of Cleaner Production, Journal of Advanced Manufacturing Technology, Scientia Iranica, and among others.

Mohammadreza Shahriari received his PhD in Operational Research from the Islamic Azad University in 2009. He worked for more than 2 years to find out he how can present and solve a Time -Cost Trade-off problem based on the time value of money, considering each crash. He was also six years in charge of chancellor of Islamic Azad University in Dubai (UAE) for more than one year in the same position at Islamic Azad University in Oxford (UK). He has had another executive academic mandate, like, establishing an Iranian University in Germany, Italy, Russia, and Oman. After almost running the abovementioned mandate, he again backed his academic and teaching activities in the university. Now he is busy teaching OR (1, 2, and 3) and advanced OR and Reliability at Bachelor and Master levels and teaching the innovative field of Decision Theory. Now, he is an Associate Professor at Azad University and is busy with some PhD and master dissertations in the field of decision theory and reliability theory.

Ahmad Khajepour was born in 1985 in Behbahan, Iran. He received his Master of Science degree in industrial engineering from Islamic Azad University (IAU), Science and Research Branch, Qazvin, Iran, in 2010. His research interests are mainly Reliability Optimization, Maintenance, and Planning & Control projects. He is currently working at TTPC related to the Iranian National Petrochemical company.

Seyed Ali Mirtaheri was born in 1985 in Qazvin, Iran. He is an MEIE student at Ryerson University in Toronto, Canada. He received an MEng degree in manufacturing engineering from University Malaya (UM), Kuala Lumpur, Malaysia, in 2012. His research interests include reliability redundancy allocation problems. He holds a BSc degree in industrial engineering from Qazvin Islamic Azad University in 2008.