Heliyon 5 (2019) e02346

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.heliyon.com

Research article

Multi-objective Redundancy Allocation Problem with weighted-k-out-of-n subsystems

Mani Sharifi [a,*], Tahmine Ashoori Moghaddam [b], Mohammadreza Shahriari [c]

[a] Faculty of Industrial & Mechanical Engineering, Islamic Azad University, Qazvin Branch, Qazvin, Iran
[b] Faculty of Industrial Engineering, Islamic Azad University, South Tehran Branch, Tehran, Iran
[c] Faculty of Management and Accounting, Islamic Azad University, South Tehran Branch, Tehran, Iran

ARTICLE INFO

Keywords:
Systems engineering
Non-dominated ranked genetic algorithm (NRGA)
Non-dominated sorting genetic algorithms (NSGA-II)
Reliability
Weighted-k-out-of-n

ABSTRACT

Redundancy Allocation Problem (RAP) is one of the most practical problems in the reliability area. Many assumptions have been added to RAP in recent years. The aim was to better represent real-world problems with RAP. One of these assumptions is considering weighted-k-out-of-n sub-systems. This method has been used to model various systems like power and hydro transitions systems. In this paper, we present a new multi-objective RAP (MORAP) model for optimizing the reliability and cost of the weighted-k-out-of-n parallel systems. In our model, the sub-systems are considered as weighted-k-out-of-n. Also, we use the universal generating function and adapt this technique to obtain an exact formula to calculate each sub-system reliability. Since RAP belongs to NP-hard class of problems, we decided to employ the non-dominated sorting genetic algorithm and non-dominated ranked genetic algorithm. Several criteria were used to compare the result of these two algorithms.

1. Introduction

In recent decades, researchers have employed several different optimization methods based on reliability to gain maximum safety and product durability for different systems with different structures. Reliability refers to the probability that a component, instrument, or system must work properly based on defined conditions. Systems may have different structures such as series, parallel, series-parallel, k-out-of-n, weighted-k-out-of-n, complex, and so on. A k-out-of-n: G system is a system with n components that works if at least its k components work correctly. If $k = 1$, then the system is a series system, and if $k = n$, the system is a parallel one. A weighted-k-out-of-n system is a system with n independent components, and each component has a positive coefficient weight w_i. The weight coefficient of the components is the utility of the components, and the total weight (utility) of the system is obtained by $w = \sum_{i=1}^{n} w_i$. The system works when the sum of component weights are equal to or greater than k. The parameter k can be greater than n because k is referred to as weight. A k-out-of-n: G system is a specific type of weighted-k-out-of-n that the weight of all components is equal to 1.

Natvig et al. [1] were the first researchers who worked on the multi-state system's availability. Ushakov [2, 3] optimized a cold-standby system using Universal Generation Function (UGF). Pham presented an optimal design for k-out-of-n systems and then extended it to k-out-of-n sub-systems [4, 5]. Levitin and Lisnianski [6] presented an optimization model for series-parallel systems. They worked on a multi-state Redundancy Allocation Problem (RAP) with the levels of components defined based on the cost and performance of components and used the Genetic Algorithm (GA) to optimize the system. To evaluate the system reliability, Wu and Chen [7] worked on a binary-weighted-k-out-of-n system and in 1994presented a recursive algorithm. Higashiyama [8, 9] presented a new method for a weighted-k-out-of-n system that had fewer steps than Wu and Chen algorithm, but the calculation time and space of both algorithms were approximately equal.

Arulmozhi [9] presented a faster and more efficient algorithm for evaluating the reliability of a k-out-of-n system. Ramirez and Coit [10] presented a heuristic algorithm for solving multi-state series-parallel RAP. The objective function of the problem was to minimize the total system costs. Zuo and Tian [11] presented a new algorithm for evaluating the reliability of a k-out-of-n system that was more flexible than Huang's algorithm [12]. They also compared the optimizing efficiency of GA and Tabu Search (TS) for binary-weighted-k-out-of-n systems using Universal Generating Function (UGF). Their conclusions showed that TS is better than GA. Li and Zuo [13, 14] worked on a multi-state weighted-k-out-of-n using UGF and recursive methods. The UGF method was better for small-

[*] Corresponding author.
E-mail address: m.sharifi@qiau.ac.ir (M. Sharifi).

https://doi.org/10.1016/j.heliyon.2019.e02346
Received 2 November 2018; Received in revised form 1 April 2019; Accepted 16 August 2019

and medium-scale problems that the components were less than 15.

Ebrahimpur and Shabani [15] optimized a series-parallel system with multi-state weighted-k-out-of-n sub-systems by the ant colony algorithm. They first calculated the reliability of sub-systems using the recursive method and then used a Universal Moment Generating Function (UMGF) for calculating the system reliability. Ebrahimpur and Sheikhalishahi [16] presented two fuzzy mathematical methods for optimizing the availability of multi-state weighted-k-out-of-n systems. In their mathematical model, the weight of the components was considered fuzzy weights.

Levitin [17] calculated the reliability of a common bus performance sharing the multi-state system. In their presented model, the exceeded surplus performance of a unit was allowed to transmit to other units. He presented a UGF based algorithm to evaluate the system reliability. Xiao and Peng [18] optimized the allocation and maintenance of a common bus performance sharing for a multi-state element in series-parallel systems. They also considered that the exceeded surplus performance of a unit might be transmitted to other units. Yu et al. [19] proposed an instantaneous availability model for common bus performance sharing repairable multi-state system.

Faghih-Roohi et al. [20] developed a multi-state weighted-k-out-of-n system and presented the optimal component design. They optimized the availability and capacity of the components using GA. Khorshidi et al. [21] used a value-driven approach and optimized a multi-state weighted-k-out-of-n system reliability-RAP. They used GA to solve the presented problem. Levitin et al. [22] optimized the redundancy of a series-parallel phased mission system exposed to random shocks. In their model, the shocks affected all the element's failure rate (increasingly) and modeled by the Poisson process.

Song et al. [23] made a tradeoff between the cost and reliability value for a k-out-of-n: G (F) majority voter to pursue the most desirable design and discussed the relationship between the cost and corresponding component parameters. They proposed a new evaluation standard for the most cost-effective design and then presented the optimal designs under different standards.

Franko et al. [24] defined optimal system configurations caused by minimizing the overall system costs in a weighted k-out-of-n: G system consisting of two-component types. Zhang [25] optimized the active redundant components and generalized some known policies for weighted k-out-of-n systems. Meenkashi and Singh [26] used Markov stochastic process and analyzed the reliability of the multi-state complex system with multi-state weighted sub-systems and. Table 1 shows a list of recent studies on the reliability field, and at the end of this table, our present work is highlighted.

In this paper, we explore a multi-objective multi-state weighted-k-out-of-n system. Our objectives are maximizing the system availability and minimizing the system costs. Non-dominated Sorting Genetic Algorithm (NSGA-II) and Non-Dominated Ranked Genetic Algorithm (NRGA) were used to optimize the system.

This paper is composed of five sections. In Section 2, we present the UGF method and its mathematical basis. The third section deals with the mathematical model. The NSGA-II and NRGA algorithms are presented in Section 4. A numerical example is presented in Section 5 to compare the performance of two methods, and the final part is devoted to the conclusion and further studies.

2. Materials

Ushakov presented the basis of this method in 1980. Then he presented the UGF in 1986 [2,3]. After 1980, this method has been widely used in availability problems. Lisnianski et al. [38] used this method for solving the multi-state RAP in 1994, and after that, so many studies have been made using this methodology.

2.1. Reliability evaluation of a weighed-k-out-of-n system

UGF method uses the system operation states and its corresponding probabilities. Consider a system with n multi-state components. Also, consider the j^{th} component has m_j different working states with corresponding performance levels. The performance level set of this component is shown in the g_j set as follows:

$$g_j = \left\{ g_{j1}, g_{j2}, \ldots, g_{jm_j} \right\} \tag{1}$$

and UGF of the j^{th} component is presented as follows:

$$u_j(z) = \sum_{i=1}^{m_j} p_{ji}.z^{g_{ji}} \tag{2}$$

In Eq. (2), p_{ji} is the correspondence probability of g_{ji}. According to Eq. (1), all possible combinations for system performance levels are defined as follows:

$$L^N = \prod_{j=1}^{N} \left\{ g_{j1}, \ldots, g_{jm_j} \right\} \tag{3}$$

and the maximum possible system states are:

$$M = \prod_{j=1}^{N} m_j \tag{4}$$

The set of system performance levels is shown as follows:

$$M = \{ g_1, \ldots, g_K \} \tag{5}$$

So, for the UGF system, we use the operator Ω_φ as follows:

Table 1
Some of the recent studies on reliability field.

Authors	Year	Components state type	Elements type	Algorithm	Objective	Parameter tuning	Failure rate
Ouzineb et al. [27]	2008	Multi	Homos	TS	Single	No	Constant
Sharma and Agarwal [28]	2009	Multi	Hetero	ACO	Single	No	Constant
Ouzineb et al. [29]	2011	Multi	Hetero	GA	Single	No	Constant
Lins and Droguett [30]	2011	Multi	Hetero	GA	Multiple	No	Constant
Levitin et al. [31]	2013	Multi	Hetero	GA	Single	No	Constant
Maatouk et al. [32]	2013	Multi	Hetero	GA	Single	No	Constant
Liu et al. [33]	2013	Multi	Hetero	Imperfect repair model	Single	No	Constant
Guilani et al. [34]	2014	Multi	Homos	Markov model	Single	-	Constant
Sharifi et al. [35]	2015	Binary	Hetero	GA, MA	Single	RSM	Time-dependent
Mousavi et al. [36]	2015	Multi	Homos	CE-NRGA	Multiple	Taguchi	Constant
Zaretalab et al. [37]	2015	Multi	Homos	MOSA	Multiple	-	Constant
Current Research	—	Multi	Homos	NSGA-II/NRGA	Single	RSM	Constant

$$
\begin{aligned}
U(z) &= W_j\{u_1(z),\ldots,u_N(z)\} \\
&= W_j\left\{\sum_{i_1=1}^{m_1} p_{1i_1}.z^{g_{1i_1}},\ldots,\sum_{i_N=1}^{m_N} p_{Ni_N}.z^{g_{Ni_N}}\right\} \\
&= \sum_{i_1}^{m_1}\sum_{i_2}^{m_2}\cdots\sum_{i_N}^{m_N}\left\{\prod_{j=1}^{N} p_{ji_j}.z^{j\left(g_{1i_1},\ldots,g_{Ni_N}\right)}\right\}
\end{aligned}
\tag{6}
$$

Now assume a system with s independent sub-systems. For each sub-system, if w_j is the weight of j^{th} component, the UGF of this component is calculated as follows:

$$
U_j(z) = p_j z^{w_j} + \left(1-p_j\right)z^0 \tag{7}
$$

If the i^{th} sub-system has n_i parallel components; the UGF of this sub-system is calculated using Eq. (12) as follows:

$$
\begin{aligned}
&U_{s_i}(z) = \Omega\{U_1(Z),\ U_2(Z),\ \ldots,U_{n_i}(Z)\} \\
&= \Omega\left(p_1 z^{w_1} + (1-p_1)z^0,\ldots,p_{n_i}z^{w_{n_i}} + \left(1-p_{n_i}\right)z^0\right) \\
&\sum_{l_1=0}^{1}\cdots\sum_{l_{n_i}=0}^{1}\left\{\prod_{j=1}^{n_i}\left\{p_j^{l_j}\left(1-p_j\right)^{1-l_j}\right\}.z^{\sum_{j=1}^{n_i}\left(l_j.w_j^{l_j}\right)}\right\}
\end{aligned}
\tag{8}
$$

In Eq. (8) p_j is the probability that sub-system is in state j and $\sum_{j=1}^{n_j}(l_j.w_j^{l_j})$ is the system performance in the state$j$. Assume that after calculating Eq. (8), we realized that sub-system i has only m_i stated with a corresponding weight g_i. Thus, we can show the UGF function for this sub-system as follows:

$$
U_{s_i}(z) = \sum_{m=1}^{m_i} p_m z^{g_m} \tag{9}
$$

For calculating the reliability of the j^{th} weighted-k-out-of-n sub-system, we have:

$$
\begin{aligned}
&A_{S_i}(k) = \delta_A\{U_{S_i}(z),k\} = \\
&\delta_A\left\{\sum_{m=1}^{m_i} p_m z^{g_m},k\right\} = \sum_{m=1}^{m_i} p_m\alpha(g_m - k)
\end{aligned}
\tag{10}
$$

In Eq. (10), $\alpha(x)$ works as follows:

$$
\alpha(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{11}
$$

3. Model

In this section, we present our mathematical model. The evaluated system is a system with serial sub-systems. The sub-systems are k-out-of-n, and the components in each sub-system have weight. In each sub-system, different types of components are available. The components are non-repairable with Constant Failure Rate (CFR). The objectives of this model are determining the type and the number of components in each sub-system.

Initially, the model assumptions and parameters are presented, and then the mathematical model is analyzed for simplicity and transparency.

3.1. Model assumptions

The model assumptions are as follows:

- All the sub-systems are weighted-k-out-of-n,
- The sub-systems are serial,
- All the failure rates, component costs and weight, and other system parameters are constant,
- All components may have only two states: working and failed,
- The components are non-repairable,
- The components failure does not affect the overall status of the system,
- The components have CFRs.

3.2. Mathematical model

The general mathematical model is as follows:

$$
Max:\quad R(n,t) \tag{12}
$$

$$
Min:\quad \sum_{i=1}^{n}\sum_{j=1}^{m}\left(c_{i\,j}\times n_{i\,j}\right) \tag{13}
$$

$$
s.t:\quad \sum_{i=1}^{n} n_{i\,j} \leq m_{Max,j} \tag{14}
$$

$$
\sum_{j=1}^{m} n_{i\,j} \leq n_{Max,i} \tag{15}
$$

Eq. (12) demonstrates the system reliability. In this equation, $R(n,t)$ is the system's reliability at time t. Eq. (13) is the model's second objective function, which aims to minimize the system's cost. In this equation, n_{ij} and c_{ij} are the number and cost of component's type j is subsystem i, respectively ($i = 1,\ldots,n; j = 1,\ldots,m$). Eq. (14) presents a limit for allocating component's type j to all the system's subsystems. In this equation, $m_{Max,j}$ is the maximum available component's type j. Eq. (15) is an upper limit for allocating redundant components to each subsystem ($n_{Max,j}$). The reliability calculations of a weighted-k-out-of-n system were presented in Section 2. After determining the reliability of each sub-system, the reliability of the whole system can be calculated using Eq. (16).

$$
r(t) = \prod_{i=1}^{s} r_i(t,k_i) \tag{16}
$$

In Eq. (16), $r_i(t,\ k_i)$ is the reliability of sub-system i at the time t with the minimum utility of k_i.

4. Methods

For solving a model with more than one inconsistent objective, multi-objective optimization methods are used. The results of this type of problem are not a single solution. In this situation, a set of solutions is available, called Pareto solutions. In these problems, the results are multiple and diverse.

In this paper, two GA-based algorithms, NSGA-II and NRGA, are used to obtain the result of the presented model.

4.1. NSGA-II algorithm

One of the most famous and efficient multi-objective optimization meta-heuristic algorithms is NSGA-II. Deb et al. [39] presented this very efficient algorithm. It considers the rank of each solution and the distance of the solutions at the same time. In this method, non-dominated solutions are classified and ranked, and the best-ranked solutions are selected.

The operators of this algorithm are as follows.

4.1.1. Initializations

The population size (Pop size), crossover operator probability (p_c), mutation operator probability (p_m), and the stop criterion is the initial information of the algorithm. In this paper, population size is created randomly. Crossover operator probability is $0.65 \leq p_c \leq 0.85$, and mutation operator probability is $0.05 \leq p_m \leq 0.20$.

$$\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \end{array} \begin{bmatrix} 3 & 0 & 0 & 1 \\ 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 3 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 3 & 1 & 0 & 0 \end{bmatrix}$$

Fig. 1. Chromosome structure.

4.1.2. *Chromosome*

Each solution chromosome is considered as a $s \times m$ matrix, where s is the number is sub-systems, and m is the number of components types. The matrix elements are integer numbers. Fig. 1 presents the chromosome structure of a system with 14 sub-systems and four different types of components.

4.1.3. *Chromosome evaluation*

After producing a chromosome, a fitness value must be allocated to the chromosome. One of the most important things in using a genetic algorithm for a problem is how the algorithm faces the constraints. The genetic algorithm operators may cause the infeasible solution. In this paper, the repairing strategy has been used to change an infeasible solution to a feasible one.

4.1.4. *Next-generation creation*

For the creation of the next generation, two different operators, mutation and crossover, have been used.

4.1.4.1. Mutation. The simplest operation for the next created generation is the mutation. The mutation probability for each gene is p_m. For mutation of a chromosome, an element of chromosome matrix is selected randomly, and the value of that element is changed. If the value is not equal to zero, it is changed to zero, and if the value is zero, a random integer number between 1 and $Min(n_{Max}, m_{Max})$ is allocated to this element. The mutation operator is illustrated in Fig. 2.

4.1.4.2. Crossover. The second method for changing a chromosome is the crossover. In this operator, every two parents make two offspring. Some different crossover operators are available. In this paper, a single-point crossover has been used. In this kind of crossover operator, one of the parent's matrix array is randomly selected, and the matrix is divided into two parts, then the first part of the first parent is replaced by the first part of the second parent. Fig. 3 illustrates the crossover operator when the third array of each parent is randomly selected for crossover.

4.1.5. *Selection based on the rapid ordering of non-dominant and congestion distance*

Now, the parents, the offspring, and the new chromosomes produced through mutation operators are combined. Therefore, we have a population less than twice of the primary population. So, an ordering method of non-dominant is used to categorize the population. In this method, a common comparison between the members of the offspring and the parents are made. Then different non-dominants queues based on the priority have been formed and filled one by one. The filling of the queues is started with the best non-dominant queue and continued until the population of the next generation is completed. The remaining population is then disposed of.

The stop criterion of the algorithm is a constant number of generations.

4.2. *Non-dominated ranked genetic algorithms (NRGA)*

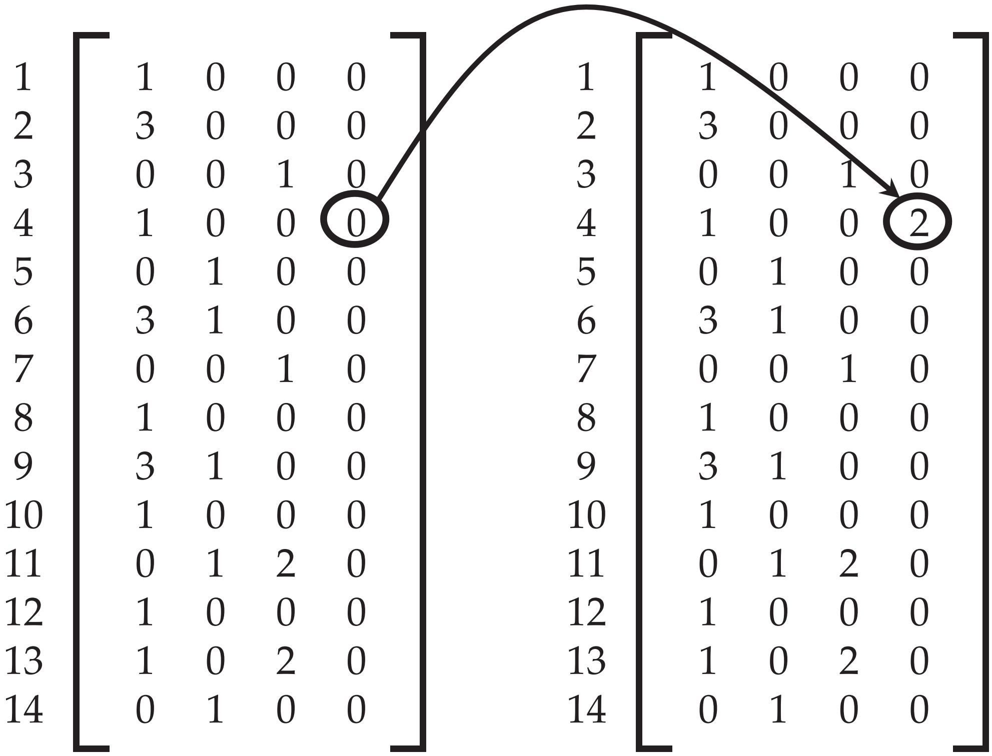
This algorithm is presented for the first time in 2008 by Al Jaddan et al. [40] for optimizing non-convex, nonlinear, and discrete functions. They expand a new approach combining the ranked based roulette wheel selection and population ranking algorithm based on Pareto, called NRGA algorithm. In this combination, a two-layer ranking method was presented based on the roulette wheel that randomly selected the new generation according to the best solutions. Compared with other multi-objective evolutionary algorithms, this algorithm can obtain a wider solution in the Pareto boundaries and more rapid convergence to optimal boundaries of Pareto. The basic difference between NRGA and NSGA-II lies in their selection strategies. In the selection based on the roulette wheel, at first, all the solutions in non-dominant boundaries are ranked so that the first boundary has the best solution among the populations. Then, if we have five non-dominant boundaries, the first boundary takes number 5, and the fifth boundary takes number 1. After ranking the boundaries, the solutions inside each boundary are ranked based on the congestion distance. After ranking the solutions in each boundary, the solution with the highest congestion distance takes the highest rank, and the solution with the lowest congestion distance takes the lowest rank. Thus, each solution has a two-layer rank. The first rank indicates the rank of the solution in non-dominant solution boundaries, and the second rank is the congestion distance-based rank inside the boundary.

In roulette wheel selection, between two different solutions in two different boundaries, the solution in the better rank boundary has more selection chance. Of two different solutions in a non-dominant boundary, the solution with more congestion distance has more selection chance.

5. Example

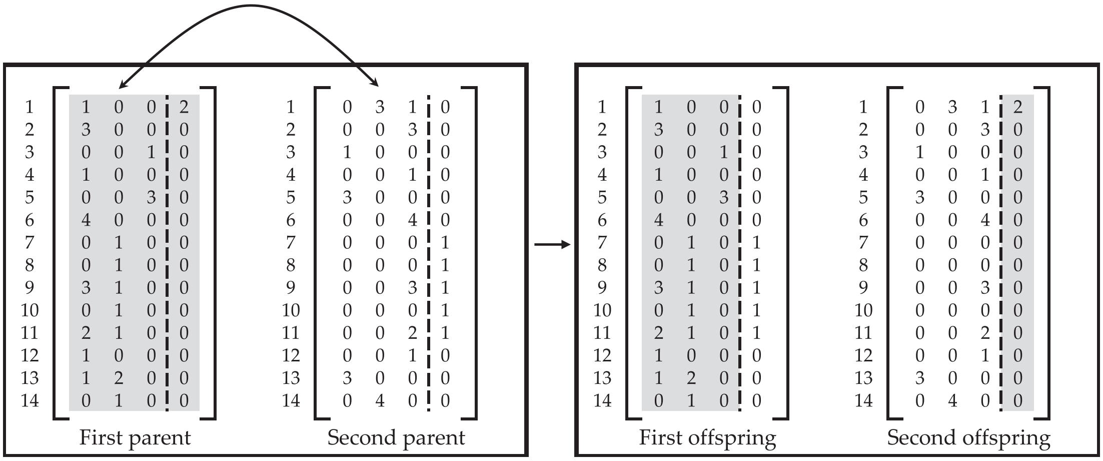
In this section, the validity of the two discussed algorithms is checked using a numerical example. Then after tuning the parameters of the algorithms, the solutions are categorized and presented.

We used the example of Fyffe et al. [41] with some changes. This example is a system with 14 series weighed-k-out-of-n sub-systems. The components in each sub-system have an active strategy. In each sub-system, more than one kind of components is available. All



$$\begin{array}{r}1\\2\\3\\4\\5\\6\\7\\8\\9\\10\\11\\12\\13\\14\end{array}\left[\begin{array}{cccc}1&0&0&0\\3&0&0&0\\0&0&1&0\\1&0&0&\boxed{0}\\0&1&0&0\\3&1&0&0\\0&0&1&0\\1&0&0&0\\3&1&0&0\\1&0&0&0\\0&1&2&0\\1&0&0&0\\1&0&2&0\\0&1&0&0\end{array}\right]\qquad\begin{array}{r}1\\2\\3\\4\\5\\6\\7\\8\\9\\10\\11\\12\\13\\14\end{array}\left[\begin{array}{cccc}1&0&0&0\\3&0&0&0\\0&0&1&0\\1&0&0&\boxed{2}\\0&1&0&0\\3&1&0&0\\0&0&1&0\\1&0&0&0\\3&1&0&0\\1&0&0&0\\0&1&2&0\\1&0&0&0\\1&0&2&0\\0&1&0&0\end{array}\right]$$

Fig. 2. Mutation operator.



$$\begin{array}{r}1\\2\\3\\4\\5\\6\\7\\8\\9\\10\\11\\12\\13\\14\end{array}\left[\begin{array}{ccc:c}1&0&0&2\\3&0&0&0\\0&0&1&0\\1&0&0&0\\0&0&3&0\\4&0&0&0\\0&1&0&0\\0&1&0&0\\3&1&0&0\\0&1&0&0\\2&1&0&0\\1&0&0&0\\1&2&0&0\\0&1&0&0\end{array}\right]\qquad\begin{array}{r}1\\2\\3\\4\\5\\6\\7\\8\\9\\10\\11\\12\\13\\14\end{array}\left[\begin{array}{ccc:c}0&3&1&0\\0&0&3&0\\1&0&0&0\\0&0&1&0\\3&0&0&0\\0&0&4&0\\0&0&0&1\\0&0&0&1\\0&0&3&1\\0&0&0&1\\0&0&2&1\\0&0&1&0\\3&0&0&0\\0&4&0&0\end{array}\right]$$

First parent Second parent

$$\begin{array}{r}1\\2\\3\\4\\5\\6\\7\\8\\9\\10\\11\\12\\13\\14\end{array}\left[\begin{array}{ccc:c}1&0&0&0\\3&0&0&0\\0&0&1&0\\1&0&0&0\\0&0&3&0\\4&0&0&0\\0&1&0&1\\0&1&0&1\\3&1&0&1\\0&1&0&1\\2&1&0&1\\1&0&0&0\\1&2&0&0\\0&1&0&0\end{array}\right]\qquad\begin{array}{r}1\\2\\3\\4\\5\\6\\7\\8\\9\\10\\11\\12\\13\\14\end{array}\left[\begin{array}{ccc:c}0&3&1&2\\0&0&3&0\\1&0&0&0\\0&0&1&0\\3&0&0&0\\0&0&4&0\\0&0&0&0\\0&0&0&0\\0&0&3&0\\0&0&0&0\\0&0&2&0\\0&0&1&0\\3&0&0&0\\0&4&0&0\end{array}\right]$$

First offspring Second offspring

Fig. 3. Crossover operator.

Table 2
Input parameters of the example.

i	$j = 1$			$j = 2$			$j = 3$			$j = 4$			k_j	n_j
	λ_{i1}	k_{i1}	c_{i1}	λ_{i2}	k_{i2}	c_{i2}	λ_{i3}	k_{i3}	c_{i3}	λ_{i4}	k_{i4}	c_{i4}		
1	0.00532	2	1	0.00073	1	1	0.00499	2	2	0.00818	3	2	8	4
2	0.00818	3	2	0.00062	1	1	0.00431	2	1	—	—	—	6	3
3	0.01330	3	2	0.01100	3	3	0.01240	3	1	0.00466	2	4	11	4
4	0.00741	2	3	0.01240	3	4	0.00683	2	5	—	—	—	7	3
5	0.00619	1	2	0.00413	2	2	0.00818	3	3	—	—	—	6	3
6	0.00436	3	3	0.00567	3	3	0.00268	2	2	0.00041	1	2	9	4
7	0.01050	3	4	0.00466	2	4	0.00394	2	5	—	—	—	7	3
8	0.01050	3	3	0.00105	1	5	0.01050	3	6	—	—	—	7	3
9	0.00268	2	2	0.00010	1	3	0.00408	1	4	0.00094	1	3	5	4
10	0.01410	3	4	0.00683	2	4	0.00105	1	5	—	—	—	6	3
11	0.00394	2	3	0.00355	2	4	0.00314	2	5	—	—	—	6	3
12	0.00236	1	2	0.00769	2	3	0.01330	3	4	0.01100	3	5	9	4
13	0.00215	2	2	0.00536	3	3	0.00665	3	2	—	—	—	8	3
14	0.01100	3	4	0.00834	1	4	0.00355	2	5	0.00436	3	6	8	4
m_i	17			18			17			8				

Table 3
Results of the NSGA-II algorithm with initial inputs.

Cost	41	44	46	47	51	52	56	60	63	65	66
R(100)	0.4475	0.8560	0.8829	0.8948	0.9173	0.9284	0.9497	0.9604	0.9608	0.9653	0.9681
Cost	67	69	73	75	79	82	85	86	92	103	
R(100)	0.9692	0.9716	0.9752	0.9789	0.9834	0.9836	0.9846	0.9871	0.9896	0.9923	

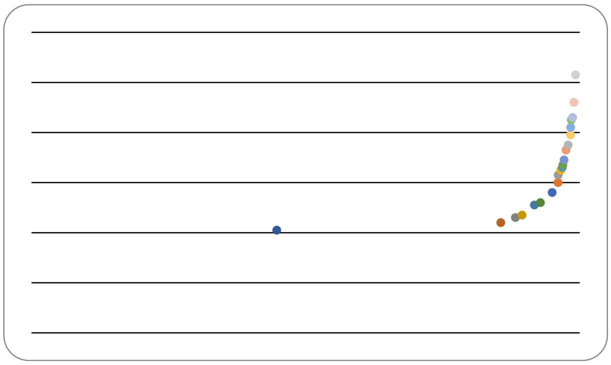


Fig. 4. Pareto curve of NSGA-II algorithm with initial inputs.

components are CFR type, so the component life is exponential. The system parameters such as components' weight, cost, failure rate, and so on are listed in Table 2. The objectives of the example are maximizing the reliability of the system and minimizing the system costs after 100 h working.

This example was solved with NSGA-II and NRGA algorithms. The initial input parameters are $Npop = 150$, $P_c = 0.65$, and $P_m = 0.05$. The results for NSGA-II algorism for Pareto solution set are presented in Table 3 and displayed in Fig. 4, and the results for NRGA algorithm for Pareto solution set are presented in Table 4 and displayed in Fig. 5.

5.1. Parameter tuning

For tuning the parameters of two algorithms, we used the Response Surface Methodology (RSM). Three parameters in each algorithm are needed to be tuned. Each parameter is considered at three different levels. These values, as well as optimal parameter values for both algorithms, are presented in Table 5.

The optimal parameter values for both algorithms are the same.

5.2. Computational results

After tuning the parameters of two algorithms, the example was solved again with the new parameters, and the results of the two algorithms are presented in Tables 6 and 7.

To compare the results of two algorithms, for all common costs, the obtained reliability of two systems have been considered. The common costs of two systems are 40, 45, 51, 54, 57, 61, 63, 67, 74, 81, 92, 93, 94, and 96. Regarding all 14 common costs, the NRGA has yielded better solutions than NSGA-II and it means that for this problem, the NRGA algorithm is more powerful than NSGA-II.

The system components allocations are presented in Table 8 for a random common cost of 93.

For evaluation of the quality and dispersal of Pareto results, some different indexes are available. In this paper, two indexes are used:

5.2.1. Distance index

This index calculates the uniformity of the non-dominant results of each algorithm. This index is calculated as follows:

$$S = \left\{ \frac{1}{N-1} \sum_{i=1}^{N} \left(\overline{d} - d_i\right)^2 \right\}^{1/2} \tag{17}$$

In Eq. (14), d_i is the Euclidean distance between two consecutive results in optimal boundaries obtained from each algorithm and $\overline{d}$ is the mean of d_i.

5.2.2. Dispersal index

This index indicates the dispersal of non-dominant results on the optimal boundaries of solutions. This index is calculated as follows:

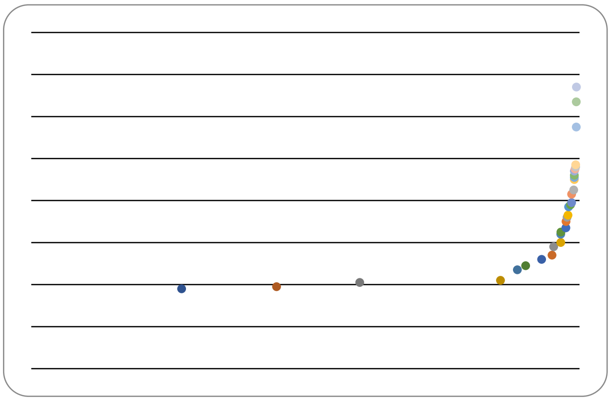


Fig. 5. Pareto curve of NRGA algorithm with initial inputs.

Table 5
Parameter levels.

Algorithms parameters	Lower limit	Medium	Upper limit	Upper limit
Npop	100	150	200	100
P_c	0.65	0.75	0.85	0.65
P_m	0.05	0.125	0.20	0.20

Table 4
Results of NRGA algorithm with initial inputs.

Cost	38	39	41	42	47	49	52	54	60
R(100)	0.274446	0.447535	0.599157	0.856035	0.886744	0.901939	0.931154	0.950049	0.965976
Cost	64	65	67	70	72	73	77	78	79
R(100)	0.966054	0.966129	0.97537	0.975425	0.977443	0.979079	0.980359	0.983707	0.98575
Cost	83	85	90	91	92	94	95	96	97
R(100)	0.985814	0.989707	0.990552	0.990571	0.990595	0.990678	0.991975	0.993176	0.993237

Table 6
The results for the NSGA-II algorithm with tuned parameters.

Cost	40	42	45	47	51	54	55	57
$R(100)$	0.447535	0.856035	0.895545	0.915402	0.933448	0.938274	0.950288	0.959544
Cost	61	63	67	72	73	74	79	81
$R(100)$	0.968174	0.972444	0.978322	0.980083	0.982544	0.983308	0.984664	0.985901
Cost	83	87	92	93	94	96	101	103
$R(100)$	0.987792	0.990396	0.991176	0.992267	0.992605	0.992895	0.994045	0.994529

Table 7
The results for NRGA algorithm with tuned parameters.

Cost	39	40	41	43	45	46	47	49	50
$R(100)$	0.447535	0.579204	0.856035	0.895545	0.897738	0.915402	0.925803	0.927578	0.928852
Cost	51	52	53	54	56	57	59	60	61
$R(100)$	0.939908	0.941345	0.942377	0.950288	0.958336	0.962522	0.965533	0.966139	0.973378
Cost	62	63	64	66	67	69	70	74	75
$R(100)$	0.974015	0.97721	0.977977	0.981542	0.981867	0.984731	0.984999	0.986807	0.988971
Cost	76	78	80	82	84	86	88	89	92
$R(100)$	0.989165	0.989261	0.990471	0.991072	0.992226	0.992405	0.993605	0.993717	0.994474
Cost	93	94	95	96	98	99	100	114	116
$R(100)$	0.994572	0.995381	0.995462	0.995502	0.996009	0.996079	0.996462	0.996856	0.996884

Table 8
The system components allocations for a random common cost of 93.

Number of sub-systems		Types of components				Number of sub-systems		Types of components			
		1	2	3	4			1	2	3	4
The designed system by NSGA-II algorithm	1	0	1	0	0	The designed system by NRGA algorithm	1	1	0	0	0
	2	3	0	0	0		2	3	0	0	0
	3	0	0	1	0		3	0	0	1	0
	4	1	0	0	0		4	1	0	0	0
	5	0	1	2	0		5	0	0	3	0
	6	3	1	0	0		6	4	0	0	0
	7	0	1	0	0		7	1	0	0	0
	8	1	0	0	0		8	1	0	0	0
	9	3	1	0	0		9	3	1	0	0
	10	1	0	0	0		10	0	1	0	0
	11	2	1	0	0		11	2	1	0	0
	12	1	0	0	0		12	1	0	0	0
	13	1	0	2	0		13	0	0	3	0
	14	0	0	0	4		14	0	0	1	3
$R(100) = 0.9923$						$R(100) = 0.9936$					

Table 9
Comparison of the results of different indexes for NSGA-II and NRGA algorithms.

Index	Mean value for NRGA	Mean value for NSGA-II
Number of Pareto solutions	43	28
Distance index	0.579	0.394
Dispersal index	53.3471	37.2867

$$D=\left\{\sum_{i=1}^{N} Max\left\|x_t^i-y_t^i\right\|\right\}^{½} \tag{18}$$

In this index, $\|x_t^i - y_t^i\|$ illustrates the Euclidian distance between adjacent x_t^i and y_t^i results on optimal boundaries of solutions.

The number of Pareto results are used to compare the two algorithms in addition to the two presented indexes. Based on the better solution of each algorithm, five iterations of each algorithm are considered to compare the performance of the algorithm. The results are presented in Table 9.

According to Table 9, the number of Pareto results for NRGA is more than that in NSGA-II. However, the dispersal and distance index for NSGA-II results are better (less than) than those in NRGA.

6. Discussion & conclusion

In this paper, we worked on a multi-objective RAP with weighted-k-out-of-n sub-systems. The objective functions of our model were maximizing the system reliability and minimizing the system cost. We considered that the sub-systems configurations are weighted k-out-of-n. This type of configuration is applicable in many systems like water and hydro transition systems. We used the UGF approach and presented some new formulas for calculating weighted k-out-of-n reliability. Since RAP belongs to NP-hard class of problems, we used NSGA-II and NRGA algorithms to solve the presented problem and compared the results of these two algorithms.

We propose further studies in two different directions. The first direction deals with changing the components characteristics. For drawing this problem closer to real-world situations, we can consider uncertain or fuzzy parameter types for components failure rate. Also, we can consider more than one working state with different performance rate components. Considering components short and open-circuit failures as well as common cause failures, is a novel idea.

The second direction is about system characteristics. One can consider technical and organizational activities to reduce the system failure rate and cost.

Declarations

Author contribution statement

Mani Sharifi: Conceived and designed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Tahmine Ashoori Moghaddam, Mohammadreza Shahriari: Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data.

Funding statement

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

Competing interest statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

References

[1] B. Natvig, et al., Multistate reliability theory—a case study, Adv. Appl. Probab. 18 (4) (1986) 921–932.

[2] I. Ushakov, Universal generating function, Sov. J. Comput. Syst. Sci. 24 (5) (1986) 118–129.

[3] I.A. Ushakov, Optimal standby problems and a universal generating function, Sov. J. Comput. Syst. Sci. 25 (5) (1987) 79–82.

[4] H. Pham, Optimal design of k-out-of-n redundant systems, Microelectron. Reliab. 32 (1-2) (1992) 119–126.

[5] H. Pham, On the optimal design of k-out-of-n: G subsystems, IEEE Trans. Reliab. 41 (4) (1992) 572–574.

[6] G. Levitin, et al., Redundancy optimization for series-parallel multi-state systems, IEEE Trans. Reliab. 47 (2) (1998) 165–172.

[7] J.-S. Wu, R.-J. Chen, An algorithm for computing the reliability of weighted-k-out-of-n systems, IEEE Trans. Reliab. 43 (2) (1994) 327–328.

[8] Y. Higashiyama, A factored reliability formula for weighted-k-out-of-n system, Asia Pac. J. Oper. Res. 18 (1) (2001) 61.

[9] G. Arulmozhi, Exact equation and an algorithm for reliability evaluation of k-out-of-n: G system, Reliab. Eng. Syst. Saf. 78 (2) (2002) 87–91.

[10] J.E. Ramirez-Marquez, D.W. Coit, A heuristic for solving the redundancy allocation problem for multi-state series-parallel systems, Reliab. Eng. Syst. Saf. 83 (3) (2004) 341–349.

[11] M.J. Zuo, Z. Tian, Performance evaluation of generalized multi-state k-out-of-n systems, IEEE Trans. Reliab. 55 (2) (2006) 319–327.

[12] J. Huang, M.J. Zuo, Y. Wu, Generalized multi-state k-out-of-n: G systems, IEEE Trans. Reliab. 49 (1) (2000) 105–111.

[13] W. Li, M.J. Zuo, Y. Ding, Optimal design of binary weighted k-out-of-n systems, Int. J. Reliab. Qual. Saf. Eng. 15 (05) (2008) 425–440.

[14] W. Li, M.J. Zuo, Reliability evaluation of multi-state weighted k-out-of-n systems, Reliab. Eng. Syst. Saf. 93 (1) (2008) 160–167.

[15] V. Ebrahimipur, A. Shabani, Optimization multi-state series weighted k-out-of-n systems by ant colony algorithm, in: Industrial Engineering and Engineering Management, 2009. IEEM 2009. IEEE International Conference on. 2009. IEEE.

[16] V. Ebrahimipur, et al., Reliability optimization of multi-state weighted k-out-of-n systems by fuzzy mathematical programming and genetic algorithm, Int. J. Syst. Assurance Eng. Manage. 2 (4) (2011) 312–318.

[17] G. Levitin, Reliability of multi-state systems with common bus performance sharing, IIE Trans. 43 (7) (2011) 518–524.

[18] H. Xiao, R. Peng, Optimal allocation and maintenance of multi-state elements in series–parallel systems with common bus performance sharing, Comput. Ind. Eng. 72 (2014) 143–151.

[19] H. Yu, J. Yang, H. Mo, Reliability analysis of repairable multi-state system with common bus performance sharing, Reliab. Eng. Syst. Saf. 132 (2014) 90–96.

[20] S. Faghih-Roohi, et al., Dynamic availability assessment and optimal component design of multi-state weighted k-out-of-n systems, Reliab. Eng. Syst. Saf. 123 (2014) 57–62.

[21] H.A. Khorshidi, I. Gunawan, M.Y. Ibrahim, A value-driven approach for optimizing reliability-redundancy allocation problem in multi-state weighted k-out-of-n system, J. Manuf. Syst. 40 (2016) 54–62.

[22] G. Levitin, M. Finkelstein, Y. Dai, Redundancy optimization for series-parallel phased mission systems exposed to random shocks, Reliab. Eng. Syst. Saf. 167 (2017) 554–560.

[23] X. Song, et al., Optimal design of the $ k $-Out-of-$ n $: G (F) majority voter, IEEE Access 5 (2017) 22647–22656.

[24] C. Franko, G.Y. Tütüncü, S. Eryilmaz, Reliability of weighted k-out-of-n: G systems consisting of two types of components and a cold standby component, Commun. Stat. Simulat. Comput. 46 (5) (2017) 4067–4081.

[25] Y. Zhang, Optimal allocation of active redundancies in weighted k-out-of-n systems, Stat. Probab. Lett. 135 (2018) 110–117.

[26] K. Meenkashi, S. Singh, A. Kumar, Reliability analysis of multi-state complex system with multi-state weighted subsystems, Int. J. Qual. Reliab. Manag. 36 (4) (2019) 552–568.

[27] M. Ouzineb, M. Nourelfath, M. Gendreau, Tabu search for the redundancy allocation problem of homogenous series–parallel multi-state systems, Reliab. Eng. Syst. Saf. 93 (8) (2008) 1257–1272.

[28] V.K. Sharma, M. Agarwal, Ant colony optimization approach to heterogeneous redundancy in multi-state systems with multi-state components, in: Reliability, Maintainability and Safety, 2009. ICRMS 2009. 8th International Conference on. 2009. IEEE.

[29] M. Ouzineb, M. Nourelfath, M. Gendreau, A heuristic method for non-homogeneous redundancy optimization of series-parallel multi-state systems, J. Heuristics 17 (1) (2011) 1–22.

[30] I.D. Lins, E.L. Droguett, Redundancy allocation problems considering systems with imperfect repairs using multi-objective genetic algorithms and discrete event simulation, Simul. Model. Pract. Theory 19 (1) (2011) 362–381.

[31] G. Levitin, et al., Reliability of series-parallel systems with random failure propagation time, IEEE Trans. Reliab. 62 (3) (2013) 637–647.

[32] I. Maatouk, E. Châtelet, N. Chebbo, Availability maximization and cost study in multi-state systems, in: Reliability and Maintainability Symposium (RAMS), 2013 Proceedings-Annual, 2013. IEEE.

[33] Y. Liu, et al., A joint redundancy and imperfect maintenance strategy optimization for multi-state systems, IEEE Trans. Reliab. 62 (2) (2013) 368–378.

[34] P.P. Guilani, et al., Reliability evaluation of non-reparable three-state systems using Markov model and its comparison with the UGF and the recursive methods, Reliab. Eng. Syst. Saf. 129 (2014) 29–35.

[35] M. Sharifi, et al., Reliability optimization of a series-parallel k-out-of-n system with failure rate depends on working components, Int. J. Indus. Eng.: Theory, Applications and Practice 22 (4) (2015).

[36] S.M. Mousavi, et al., Two tuned multi-objective meta-heuristic algorithms for solving a fuzzy multi-state redundancy allocation problem under discount strategies, Appl. Math. Model. 39 (22) (2015) 6968–6989.

[37] A. Zaretalab, et al., A knowledge-based archive multi-objective simulated annealing algorithm to optimize series–parallel system with choice of redundancy strategies, Comput. Ind. Eng. 80 (2015) 33–44.

[38] A. Lisnianski, H. Ben Haim, D. Elmakis, Redundancy optimization for power station, in: Proceeding of the 10th International Conference of the Israel Society for Quality, 1994, pp. 313–319. Jerusalem.

[39] K. Deb, et al., A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, in: International Conference on Parallel Problem Solving from Nature, Springer, 2000.

[40] O. Al Jadaan, L. Rajamani, C. Rao, NON-DOMINATED ranked genetic algorithm for solving multi-objective optimization problems: NRGA, J. Theor. Appl. Inf. Technol. 4 (1) (2008).

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