Reliability Engineering and System Safety 193 (2020) 106629

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Redundancy allocation problem with multi-state component systems and reliable supplier selection

Arash Zaretalab[a], Vahid Hajipour[b], Madjid Tavana[c,d,*]

[a] Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran
[b] Department of Industrial Engineering, College of Engineering, West Tehran Branch, Islamic Azad University, Tehran, Iran
[c] Business Systems and Analytics Department, Distinguished Chair of Business Systems and Analytics, La Salle University, Philadelphia, PA 19141, USA
[d] Business Information Systems Department, Faculty of Business Administration and Economics, University of Paderborn, D-33098 Paderborn, Germany

ARTICLE INFO

Keywords:
Reliability
Redundancy allocation problem
Supplier Selection
Memetic algorithm
Applied Intelligence

ABSTRACT

We propose a mathematical model for optimizing multiple redundancy-reliability systems known as mega-systems. The system components are multi-state and the universal generating function (UGF) has been simulated to evaluate the system availability. The components may have minor or major failure, which reduces the components performance rate. We assume the components can be sold to different vendors and a vendor selection component is included in the proposed model to accommodate this assumption. The proposed mathematical model is NP-hard and we use a parameter-tuned memetic algorithm (MA) to solve the problem. We further use a mechanism based on the response surface methodology (RSM) to calibrate the proposed MA. The performance of the proposed MA is compared with a commonly used genetic algorithm (GA).

1. Introduction

Reliability is the ability of a component or system to function at a specific time and it is imperative to study a system's reliability to guarantee its competitiveness [8,9]. In general, there are two methods for maximizing system reliability: (a) increasing the reliability of system components, and (b) finding the optimal number of redundant components used in the system. The optimal location of components is addressed as the redundancy allocation problem (RAP) ([17,20,40]). It has been extensively used in many real-world applications such as electrical power systems, transportation systems, safety systems, telecommunication systems, satellite systems, etc. [31,36,40].

Various structures can be found in the RAP literature depending on the system configurations such as series, parallel, series–parallel, hierarchical series–parallel or complex systems [6,52]. In this paper, we consider the series–parallel structure of the RAP where the system consists of n sub-systems in series. For each sub-system I, there exist a number of components connected in parallel. Chern proved that RAP belongs to NP-hard problems [5]. In recent years, the problem has been tackled using exact approaches such as dynamic programming (DP) [51], branch and bound (B&B) [11] as well as heuristic and meta heuristic approaches such as simulated annealing (SA) [15,56], tabu search (TS) [16], ant colony optimization (ACO) [42], genetic algorithms (GA) [7,44], combination of the greedy method and the GA (Greedy/GA) [54], immune algorithm (IA) [3,4], variable neighborhood descent algorithm (VND) [25], variable neighborhood search (VNS) [24], coupled ant colony with degraded ceiling algorithm (ACO/DC) [35], particle swarm optimization (PSO) [41], artificial bee colony (ABC) [53], electromagnetism-like mechanism (EM) [45].

A latest trend of redundancy allocation research is to consider the problem under the multi-state-system (MSS) framework, which makes it more challenging than under the traditional framework of binary-state systems (BSSs) [19,20]. The complicated deteriorating behaviors of engineered systems beyond binary-state cases can be modeled by considering several states between the perfect functioning state and the completely failed state. Some examples of these systems include power systems [30], networked systems [22,33], municipal infrastructure systems [13], and computing systems [32]. The following methods have been proposed in the literature for analyzing multi-state systems: the extended decision diagram-based method [43], the stochastic process method [28,57], the universal generating function (UGF) method [58], the recursive algorithm method [23], and the simulation-based method [49].

Although the MSS of RAP has not been sufficiently analyzed in comparison with the binary state, it has been getting more attention in recent years. Liu et al. proposed a joint redundancy and imperfect

* Corresponding author at: Business Systems and Analytics Department, Lindback Distinguished Chair of Information Systems and Decision Sciences, La Salle University, Philadelphia, PA 19141, USA.
E-mail addresses: arash_zaretalab@aut.ac.ir (A. Zaretalab), hajipour.v@wtiau.ac.ir (V. Hajipour), tavana@lasalle.edu (M. Tavana).

https://doi.org/10.1016/j.ress.2019.106629
Received 11 March 2019; Received in revised form 13 July 2019; Accepted 31 August 2019
Available online 04 September 2019
0951-8320/

maintenance strategy optimization for multi-state systems [29]. In their model, in addition to determining the optimal redundancy, a component replacement strategy under imperfect repair was also developed. Wang and Li utilized a hybrid algorithm of PSO and local search LS to solve the RAP of series–parallel multi-state systems MSS [50]. Ouzineb et al. solved a multi-state non-homogenous series–parallel RAP using an effective heuristic algorithm comprised of GA and TS [38]. Sharma and Agarwal proposed an ant colony optimization approach to heterogeneous redundancy in multi-state systems with multi-state components [42], in which non-identical components were used. In addition, a UGF was utilized [39] to evaluate system reliability of a repairable multi-state series–parallel RAP. Guilani et al. presented a method for reliability evaluation of non-repairable three-state systems using a Markovian model. They showed their method was more efficient that the UGF approach [10]. Lins and Droguett proposed a multi-objective GA (MOGA) coupled with discrete event simulation to solve redundancy allocation problems in systems subject to imperfect repairs [26]. Recently, Mousavi et al. proposed a multi-objective multi-state homogeneous RAP problem, where performance rates and availability of the components are assumed as fuzzy [34].

Table 1 presents a review of the RAP literature and highlights the differences between this work and other research. As shown in, redundancy allocation is the main approach for increasing the system reliability while the other methods can also be considered to enhance the system reliability. For example, [1] considers a mixed-integer nonlinear optimization of the reliability–redundancy allocation problem (RRAP) to determine simultaneous reliability and redundancy level of components. [47] presented a common reliability-redundancy optimization approach in which the components states distribute under effects of the transition rates. Redundancy allocations are also considered as system variables. Following this, [46] introduced a new approach for increasing reliability through two options: (I) increasing system reliability via effects on operating rates; and (II) increasing system reliability via redundancy allocation. [55] proposed a warm standby repairable system consisting of two dissimilar units and one repairman. In this system, it is assumed that the working time distributions and the repair time distributions of the two units are both exponential, and unit 1 is given priority in use.

Related research shows that improving the performance of a system component can have direct impact on the overall performance and reliability of the entire system. We propose a redundancy allocation problem with multi-state component systems and reliable supplier selection. We use a supplier selection policy with the amount of purchasing components as a system variable and consider optimizing a set of systems called mega-system. The availability of the mega-system is evaluated using the UGF approach. To solve the proposed mathematical model, a memetic algorithm (MA) is proposed and compared with GA. To enhance the performances of the proposed meta-heuristic algorithms, a mechanism based on the Response Surface Methodology (RSM) is considered to calibrate the parameters of the two algorithms. A schematic view of our model is presented in Fig. 1. Many practical examples of minimal repair can be found in Lisnianski and Levitin [20] and Tian et al. [46].

This paper is divided into six sections. In Section 2, the system definition is presented. A numerical example is presented in Section 3, and the solution methodologies are applied in Section 4. Section 5 analyzes the problem solutions of two algorithms. Finally, Section 6 provides conclusions and further studies.

2. Problem definition

Simultaneous optimization of multiple systems with regards to capacity constraints still remains a challenging problem. In this paper, we propose a model that considers the authority of the decision makers in identifying the duration of time in which a system functions faultless, and the minimal expected availability of the sub-systems with regards to their expected performance rates. We propose a mathematical model to optimize multiple redundancy-reliability systems known as mega-systems. The multi-state system components with UGF may have minor or major failure. Moreover, a vendor selection component is used to accommodate component sales to multiple vendors. Parameter-tuned MA and response surface methodology are used to solve and calibrate the proposed model. The performance of parameter-tuned MA is compared with GA. In the next section, we define the problem and then introduce the assumptions, parameters, and decision variables. Finally, we introduce the mathematical model proposed in this study.

2.1. State space diagram

In the proposed model, each component has $(M+1)$ states that are numbered from 0 to M. In classic models, only the transitions between two adjacent states are acceptable. In other words, only one kind of minor failure is available. In this paper, we assume that transitions between all states are acceptable. Failure and repair rates in this study

Table 1
A brief review of relevant RAP literature.

Research	State	Algorithm	Fault elements	Objective	Cost discount strategy	Technical and Organizational activities	Supplier selection
[53]	Binary	ABC	Non-repairable	Single	No	No	No
[14]	Binary	e-constraint	Non-repairable	Multiple	No	No	No
[2]	Binary	SA	Non-repairable	Single	No	No	No
[3]	Binary	IA	Non-repairable	Single	No	No	No
[16]	Binary	TS	Non-repairable	Single	No	No	No
[29]	Multi-state	Imperfect repair model	Repairable	Single	No	No	No
[54]	Binary	Greedy/GA	Non-repairable	Single	No	No	No
[42]	Multi-state	ACO	Non-repairable	Single	No	No	No
[37]	Multi-state	TS	Non-repairable	Single	Yes	No	No
[21]	Multi-state	UGF	Non-repairable	Single	No	No	No
[10]	Multi-state	Markov model	Non-repairable	Single	No	No	No
[26]	Binary	GA	Repairable	Multiple	No	No	No
[24]	Binary	VNS	Non-repairable	Single	No	No	No
[41]	Binary	PSO	Non-repairable	Single	No	No	No
[34]	Multi-state	CE-NRGA	Non-repairable	Multiple	Yes	No	No
[46]	Multi-state	GA	Repairable	Single	Yes	Yes	No
[45]	Binary	EM	Non-repairable	Single	No	No	No
[56]	Binary	SA	Non-repairable	Multiple	No	No	No
Our proposed method	Multi-state	MA	Repairable	Single	Yes	Yes	Yes

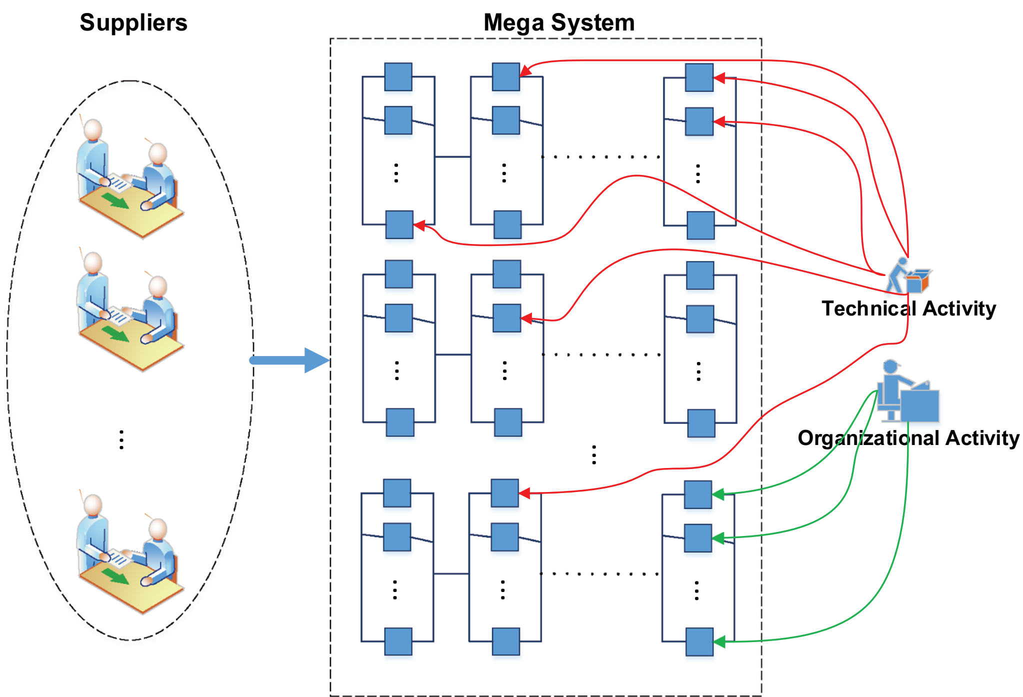

Fig. 1. A schematic view of our proposed framework.

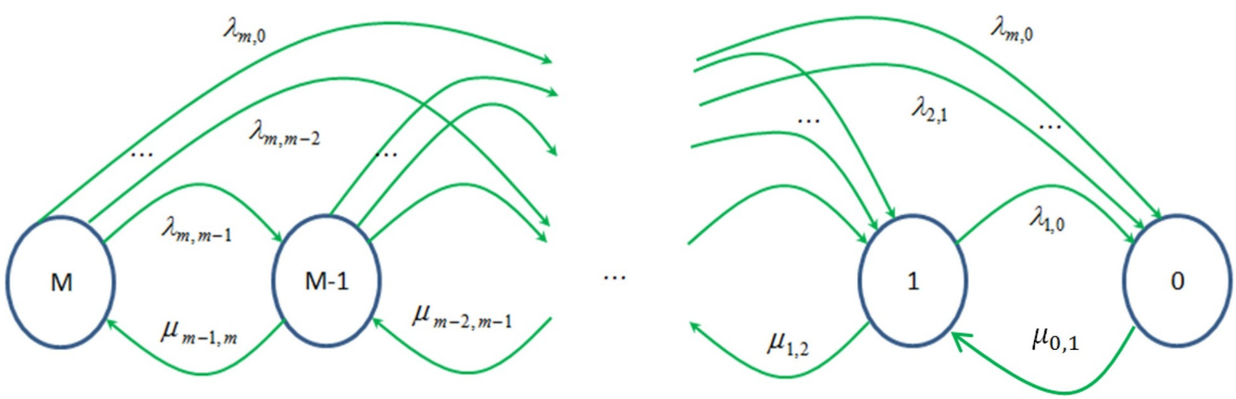

Fig. 2. State space diagram.

are constant and the probability distribution of the states is assumed exponential distribution. Based on this assumption, Markov processes specification is used to compute the probability distribution of the states. The state space diagram of the system components is presented in Fig. 2.

As shown in, transitions from state i to the state j are acceptable for all $j < i$. This means that more than one minor failure may cause for each component in each state, but a transition from state i to state $(i + 1)$ is acceptable when a component repairs. The differential equations of the model are as follows:

$$
\begin{aligned}
&p'_M(t) = -(\textstyle\sum_{i=0}^{M-1} \lambda_{M,i})p_M(t) + \mu_{M-1,M}p_{M-1}(t)\\
&p'_j(t) = -(\textstyle\sum_{i=0}^{j-1} (\lambda_{j,i}) + \mu_{j,j+1})p_j(t) + \sum_{i=j+1}^{M} \{\lambda_{i,j}p_i(t)\} + \mu_{j-1,j}p_{j-1}(t)\,;\\
&j = 1,\ 1, \ldots, M-1\\
&p'_0(t) = -\mu_{0,1}p_0(t) + (\textstyle\sum_{i=1}^{M} \lambda_{i,0})p_i(t)
\end{aligned} \tag{1}
$$

In Eq. (1), $p_i(t)$ is the probability that components are in the state $\{i;\quad i = 1, 2, \ldots, M\}$ at the time t. $\lambda_{i,\,j}$ and $\mu_{i,\,j}$ are the failure and repair rates, respectively. The initial vales are:

$$
\begin{aligned}
&p_i(0) = 0 \quad ; \quad i = 0, 1, \ldots, M-1\\
&p_M(0) = 1
\end{aligned} \tag{2}
$$

2.2. Technical and organizational activities

These activities are divided into two parts: (I) activities in components level that are called technical activities; and (II) activities in subsystems level that are called organizational activities. These activities effect the transition rates and the coefficients of the differential equations and improve system availability. Some real-world applications include [46]: (a) installation of a condition monitoring system, which can monitor the condition of the component and thus reduce certain failure rates; (b) use of a certain maintenance program, which can reduce certain failure rates and increase certain repair rates; (3) doubling the resource and maintenance staff in maintenance of the component, which can increase the repair rates. Let us review several important considerations and motivations:

- Increasing the number of parallel components in a system can promote reliability and availability of the system but it is not sufficient [46]. Related research shows that improving the performance of each component in a system can improve the reliability of the entire system. In addition, if there is a budget constraint, efforts in promoting components' availability leads to less redundancy assignment and vice versa.
- The related research shows optimizing several systems simultaneously with regards to capacity constraints still remains a problem. We propose a model that considers decision makers' authority in determining the system parameters. Considering this authority achieves two objectives. First, the duration time in which a system functions properly can be specified. Second, the minimal expected availability of each system with regards to expected performance rate can be determined.
- The related research does not consider supplier related issues and concerns. In this study, we consider supplier selection problem for each component. This includes determining the amount of components supplied by each supplier and taking into consideration their incentive and discount policies. Therefore, more purchase from a supplier leads to lower prices.
- Many realistic examples of technical and organizational activities are given in Tian et al. [46]. For instance, consider the installation of a vibration monitoring system. The vibration monitoring can monitor the health condition of the fans and suggest preventive replacements to avoid unexpected failures. Before adopting the vibration monitoring system, the reduction in the failure rates of the fans, i.e., the benefit of installing the vibration monitoring system, can be estimated based on the historical failure data. Other examples includes monitoring systems and maintenance planning systems.

In this paper, we use *AC* (Activity in component level) for technical activities and *AS* (Activity in system level) for organizational activities. The technical activities improve the components performance and the cost of these components depends on the number of the components in each sub-system. The organizational activities improve sub-system performance, and the cost of these activities does not depend on the number of the sub-system components. Eq. (3) represents the effect of technical activities:

$$\lambda = (1 - \alpha.\, AC)\lambda \tag{3}$$

In this equation, α determines the maximum effect of doing this activity on the components. For example, assume $\lambda = 0.1,\ \alpha = 0.5$. If these technical activities are done with 100 percent of performance, the failure rate of the sub-system component decreases by 50% ($\lambda = (1 - 0.5 \times 1)\lambda = 0.5\lambda$) and if this type of technical activity is done with 60% of performance, the failure rate of the sub-system component decreases by 30 percent ($\lambda = (1 - 0.5 \times 0.6)\lambda = 0.7\lambda$). *AC* and *AS* are variables between zero and one. Also, for repair rates, the effects of these two activities are:

$$\mu = (1 + \alpha.\, AC)\mu \tag{4}$$

2.3. Universal generating function (UGF)

Ushakov presented the concepts of UGF for evaluating the system reliability [48]. Lisnianski et al. [27] used UGF for reliability evaluation of multi-state serial, parallel and series-parallel systems. For a multi-state system, depending on the components state, the system states increase dramatically, so using classic methods for evaluating the system reliability is impossible. The UGF approach aggregates different state combination along the process of availability calculation to decrease the system states and makes the system availability evaluation easier to carry out.

2.4. Calculating system availability

The following steps can be used to calculate system's availability level based on the state space diagram and the UGF method:

a Obtain the state probability distribution of the components and systems based on the differential equations system. The failure and repair rates under the technical and organizational activities are constantly changing. These changes can affect the probability distribution of the states and systems. Section 2.1 shows the methodology for obtaining differential equation systems as well as state diagrams. These equations can be solved with numerical methods [20]. In this research, the MATLAB software is used to determine the values of technical and organizational activities as well as the failure and repair rates using the relations in Section 2.2.

b Calculate the availability level using the UGF method. Consider a system with n components, where components j ; $j = 1, 2, \ldots, n$, may have k_j different states with certain probabilities of performance rates denoted by an ordering set $g_j = \{g_{j1}, g_{j2}, \ldots, g_{jk_j}\}$ in which g_{ji_j} represents the performance rate of component j in state $i_j \in \{1, 2, \ldots, k_j\}$. The performance rate $G_j(t)$ of component j at time $t \geq 0$ is a random variable taking values in g_j: $G_j(t) \in g_j$. Moreover, let the probabilities associated with different states of component j be the set $P_j = \{p_{j1}, p_{j2}, \ldots, p_{jk_j}\}$. Furthermore, $g_{ji_j} \to P_j$ is often called the probability mass function. As soon as the performance rates of the components are given, the performance rate of a MSS can be determined. Let the system have K different states and g_i be the performance rate of the system in state i; $i = 1, 2, \ldots, K$. Then, the system performance rate at time $t \geq 0$ will be either a random variable or a random vector that takes values in $\{g_1, \ldots g_i, \ldots, g_K\}$. Thus, the space representing all possible combinations of performance rates for all components is $L^n = \{g_{11}, \ldots, g_{1k_1}\} \times \ldots \times \{g_{j1}, \ldots, g_{jk_j}\} \times \ldots \times \{g_{n1}, \ldots, g_{nk_n}\}$ and the space for all possible values of the entire system performance rates is $M = \{g_1, \ldots, g_K\}$. The transformation $\phi(G_1(t) \ldots G_n(t))$: $L^n \to M$ that maps the space of performance rates of system components into the space of system performance rates is called the system structure function [20]. Moreover, the total number of possible states (performance rates) of a MSS is

$$K = \prod_{j=1}^{n} k_j \tag{5}$$

In addition, the probability associated with the state i of the system can be obtained as:

$$P_i = \prod_{j=1}^{n} p_{ji_j} \tag{6}$$

Denoting the MSS performance rate for state i as:

$$g_i = \phi\left(g_{1i_1}, g_{2i_2}, \ldots g_{n\, i_n}\right) \tag{7}$$

The probability distribution of the whole system for K combinations of $i_1, i_2, \ldots, i_n$ is:

$$g_i = \phi\left(g_{1i_1}, g_{2i_2}, ...g_{n\,i_n}\right)\ ;\ P_i = \prod_{j=1}^{n} p_{ji_j} \tag{8}$$

where $1 \le i_j \le k_j$, $(1 \le j \le n)$.

The z-transform of a random variable $G_j(t)$ represents its probability mass function with $p_j = \{p_{j1}, p_{j2}, ...p_{jk_j}\}$ associated with $g_j = \{g_{j1}, g_{j2}, ...g_{jk_j}\}$. Eq. (9) shows the probability distribution of the component j, called also individual UGF.

$$u(z) = \sum_{i=1}^{k_j} p_{ji_j}.\, z^{g_{ji_j}} \tag{9}$$

To derive the probability distribution of the entire MSS with an arbitrary structure functionϕ, a general composition operator Ω_ϕ is employed on individual UGF of n components as:

$$U(z) = \Omega_\phi\{u_1(z),\ \ldots, u_n(z)\} = \Omega_\phi\left\{\sum_{i_1=1}^{k_1} p_{1i_1}.\ z^{g_{1i_1}},\ \ldots,\ \sum_{i_n=1}^{k_n} p_{ni_n}.\ z^{g_{ni_n}}\right\}$$
$$= \sum_{i_1}^{k_1}\sum_{i_2}^{k_2}\ldots\sum_{i_n}^{k_n}\left(\prod_{j=1}^{n} p_{ji_j}.\ z^{\phi(g_{1i_1},\ldots,g_{ni_n})}\right) \tag{10}$$

Based on the relationship between MSS performance and the demand level ω that is often determined outside the system, the state space of a MSS can be divided into two subsets: acceptable and unacceptable. The relationship usually is determined by the system state adequacy index r_i defined by $r_i = g_i - \omega$. As a result, state i is acceptable if $r_i \ge 0$. The availability of a MSS (reliability of a non-repairable MSS) is defined as the probability the system staying in the subset of acceptable states. Thus, based on the demand level ω the availability of a MSS, $A(\omega)$, is usually defined as the probability the MSS performance rate is greater than ω. In other words:

$$A(\omega) = \sum_{r_i \ge 0} p_i \tag{11}$$

Then, using operator δ_A it becomes

$$A(\omega) = \delta_A(U(z), \omega) = \delta_A\left(\sum_{i=1}^{K} p_i.\, z^{g_i}, \omega\right) = \sum_{i=1}^{K} p_i.\, \alpha_i$$

where

$$\alpha_i = \begin{cases} 1\ , & r_i \ge 0 \\ 0\ , & r_i < 0. \end{cases} \tag{12}$$

In Eq. (12), δ_A is known as the UGF operator. This operator determines the polynomial UGF for a group of components first connected in parallel in a subsystem and then for a group of subsystems in series using simple algebraic operations on the individual UGF of components. In some cases, composition operators can be developed for structures with more complex system structure, such as bridges, as shown by Levitin and Lisnianski [18].

2.5. System assumptions

- Components failures are independent,
- Components are multi-states,
- Components have different performance rates in different states,
- The performance rate of each sub-system is equal to the sum of its component performance rates,
- The performance rate of each system is equal to the minimum performance rate of its sub-systems,
- The technical activities effect all sub-system components,
- The cost of technical activities depends on the number of sub-system components,
- Components have more than one minor failure mode,
- The components may be purchased from different suppliers,
- Suppliers offer discounts to sell more components.

2.6. Nomenclatures

$P^i_{hj}(t)$:	probability that version h of element is in state j at the time t,
g^i_{hj}:	operation rate of components version h in sub-system i and state j,
$u^i_h(t)$:	UGF for components version h in sub-system i,
$\otimes_\varphi$:	UGF operator,
$P^i_j(t)$:	probability of existence sub-system i in the state j at the time t,
g^i_j:	operation rate of components in sub-system i in the state j,
$u^i(t)$:	UGF for sub-system i,
$p_j(t)$:	probability of existence the Super-system in the state j at the time t,
g_j:	operation rate of the Super-system in the state j,
$U(z)$:	UGF of the Super-system,
$A_b(w_{0b})$:	availability of system b under operation rate w_{0b},
A_{b0}	availability constraint of system b,
n_{kbih}:	number of purchased components version h for sub-system i of system b from supplier k,
n_{Max}:	maximum number of allocated components in each sub-system of each system,
AC_{bihv}:	technical activity type v on components type h for sub-system i of system b,
AS_{biw}:	organizational activity w for sub-system i of system b,
α_{bihvcd}:	maximum effects ratio for technical activity type v for transition rate from state c to state d for component version h on sub-system i of system b,
β_{bihwcd}:	maximum effects ratio for organizational activity type w for transition rate from state c to state d for component version h on sub-system i of system b,
λ_{bihcd}:	failure rate in transition from state c to state d, $(d > c)$ for components version h on sub-system i of system b,
μ_{bihcd}:	repair rate in transition from state c to state d, $(d < c)$ for components version h on sub-system i of system b,
CAC_{bihv}:	unique cost for doing technical activity type v on components type h for sub-system i of system b,
$CACO_{bihv}$:	total cost for doing technical activity type v on components type h for sub-system i of system b,
CAS_{biw}:	total cost for doing organizational activity type w for sub-system i of system b,
CO_{kbih}:	fixed cost of purchasing components version h for sub-system i of system b from supplier k,
CC_{kbih}:	per unit cost of purchasing components version h for sub-system i of system b from supplier k,
M_i:	maximum available components state in sub-system i,
M_{si}:	maximum available states in sub-system i,
M:	maximum available states for series-parallel system,
L:	number of suppliers,
B:	number of systems in the Super-system,
s_b:	number of sub-systems in system b,
H_{bi}:	number of different versions in sub-system i of system b,
V_{bi}:	number of technical activities for sub-system i of system b,
W_{bi}:	number of organizational activities for sub-system i of system b,
M_{kih}:	maximum purchased components version h available on sub-system i depend on the supplier k,
x_{kbih}:	a binary variable represents purchasing components version h of sub-system i of system b from supplier k,
σ_k:	amount of money that has been paid to the supplier k,
d_k:	discount ratio of supplier k,
u_{kf}:	binary variable represents the amount of money that is paid to supplier k belong to purchasing level f,
q_f:	minimum necessary expenditure to locate the paid amount of each supplier in f purchase level,
z_f :	discount level related to purchasing level f for each supplier,
F:	purchasing levels,

Table 2
Failure, repair rates, and performance rates of components type in sub-system 1.

Version	g_{h1}^{1}	g_{h2}^{1}	λ_{b1h10}	λ_{b1h21}	λ_{b1h20}	μ_{b1h01}	μ_{b1h12}
1	30	60	0.04	0.05	0.025	0.4	0.6
2	50	100	0.08	0.09	0.045	0.4	0.5
3	60	120	0.05	0.06	0.030	0.5	0.7

Table 3
Fixed cost and unit cost of components in sub-system 1.

Supplier	Version 1		Version 2		Version 3	
	CC_{kb11}	CO_{kb11}	CC_{kb12}	CO_{kb12}	CC_{kb13}	CO_{kb13}
1	18	10	25	15	40	30
2	20	11	22	12	45	33
3	19	10	23	14	42	33

2.7. Developed mathematical model

The mathematical model of the presented problem is as follows:

$$Min \sum_{b=1}^{\mathbf{B}} \sum_{i=1}^{sb} \sum_{h=1}^{H_{bi}} \sum_{v=1}^{V_{bi}} \{AC_{bihv}\{CACO_{bihv} + \sum_{k=1}^{L} (n_{kbih}.\ CAC_{bihv})\}\}+ \sum_{b=1}^{\mathbf{B}} \sum_{i=1}^{sb} \sum_{w=1}^{W_{bi}} \{AS_{biw}.\ CAS_{biw}\} + \sum_{k=1}^{L} (\sigma_k.\ d_k)$$

$s.\ t$:

$$(13-01)\quad \sigma_k = \sum_{b=1}^{B} \sum_{i=1}^{sb} \sum_{h=1}^{H_{bi}} (CO_{kbih}.\ x_{kbih} + n_{kbih}.\ CC_{kbih}) \quad \forall\ k$$
$$(13-02)\quad q_f.\ u_{kf} \le \sigma_k.\ u_{kf} < q_{f+1}.\ u_{kf} \quad \forall\ k, f$$
$$(13-03)\quad \sum_{f=1}^{F} u_{kf} \le 1 \quad \forall\ k$$
$$(13-04)\quad d_k = \sum_{f=1}^{F} (u_{kf}.\ z_f) \quad \forall\ k$$
$$(13-05)\quad A_b(w_0) \ge A_{b_0} \quad \forall\ b$$
$$(13-06)\quad \sum_{k=1}^{L} \sum_{h=1}^{H_{bi}} n_{kbih} \le n_{Max} \quad \forall\ b, i$$
$$(13-07)\quad \sum_{b=1}^{B} n_{kbih} \le M_{kih} \quad \forall\ k, i, h$$
$$(13-08)\quad n_{kbih} \le x_{kbih}.\ n_{Max} \quad \forall\ k, b, i, h$$
$$(13-09)\quad \lambda_{bihcd} = \lambda_{bihcd} - \alpha_{bihvcd}.\ AC_{bihv}.\ \lambda_{bihcd} \quad ; \quad \forall\ b, i, h, v, (c > d)$$
$$(13-10)\quad \mu_{bihcd} = \mu_{bihcd} + \alpha_{bihvcd}.\ AC_{bihv}.\ \mu_{bihcd} \quad ; \quad \forall\ b, i, h, v, (c < d)$$
$$(13-11)\quad \lambda_{bihcd} = \lambda_{bihcd} - \beta_{bihwcd}.\ AS_{biw}.\ \lambda_{bihcd} \quad ; \quad \forall\ b, i, h, w, (c > d)$$
$$(13-12)\quad \mu_{bihcd} = \mu_{bihcd} + \beta_{bihwcd}.\ AS_{biw}.\ \mu_{bihcd} \quad ; \quad \forall\ b, i, h, w, (c < d)$$
$$n_{kbih} \ge 0, \quad x_{xbih} = 0\ Or\ 1, \quad u_{kf} = 0\ Or\ 1, \quad 0 \le AC_{bihv} \le 1, \quad 0 \le AS_{biw} \le 1 \tag{13}$$

The objective function includes: (a) cost of technical activities, (b) cost of organizational activities, and finally (c) cost of purchasing components from suppliers with consideration of discounts. The constraints are defined as follows:

13 − 01: Total cost that is paid to each supplier,
13 − 02: Determine the cost that is paid to each supplier is only applicable to a specific purchase level,
13 − 03: Ensure that the cost that is paid to each supplier lies only in one purchase level,
13 − 04: Determine the discount ratio for each supplier,
13 − 05: Determine that the total availability of each sub-system under performance rate w_0 must be more than the predefined availability of the system,
13 − 06: Determine that the number of components in each system must not be more thann_{Max},
13 − 07: Determine that the sum of the components version h on sub-system i for all systems that have been purchased from each supplier must not be more than the supplier inventory,
13 − 08: Purchase of each component version from each supplier is allowable if the supplier belongs to the supplier's list of the component version,
13 − 09: Represent the effects of technical activities of failure rates,
13 − 10: Represent the effects of technical activities of repair rates,
13 − 11: Represent the effects of organizational activities of failure rates,
13 − 12: Represent the effects of organizational activities of repair rates,

3. Numerical example

A numerical example is considered in this section to illustrate the approach presented in this work. Assume that, we have a mega-system with a two series–parallel system with two parallel sub-systems connected in series. These two systems are identical with regard to different expected availability. For sub-system 1, there exist three different versions of components available so that each version has three different states: 0, 1, and 2. Also, five technical activities and three organizational activities are available. For sub-system 2, there exist four different versions of components so that each version has two different states: 0 and 1. Also, four technical activities and one organizational are available. The failures are multi-stage, but the repair rates are single. The performance rate of each sub-system is equal to the sum of component performance rates and the operation rate of the system is equal to the minimum of sub-system performance rates. w_0 is the expected performance level of all systems, but the minimum availability of the systems is different. The expected availability of system 1 is $A_{1_0} = 0.9$ and expected availability of system 2 is $A_{2_0} = 0.9$ and the working time is considered as 100^h. Other parameters of the example are presented in Tables 2–21.

4. Solution methodologies

In this paper, the proposed MA for solving the problem is presented in Section 4.2. In order to demonstrate the performance of the proposed MA, GA is applied and illustrated in Section 4.3. Before we describe both algorithms, the procedure of the executed RSM is presented in Section 4.1.

4.1. RSM

Both GA and MA have three parameters which must be tuned. The stop condition of these algorithms is 50 rounds. RSM needs 2^3 corner points, 2×3 pivot points and 5 central points. Therefore, for parameter tuning of the algorithm ($2^3 + 2 \times 3 + 5 = 19$) iterations of each algorithm are needed.

4.2. A GA for proposed RAP

Many researchers have used GA in different fields of industrial and operational management such as RAP, inventory control, facility location and so on. Therefore, we have also used this algorithm here as one

Table 4
Technical activities specifications of components version 1 in sub-system 1.

Version	$CACO_{b11v}$	CAC_{b11v}	α_{b11v10}	α_{b11v21}	α_{b11v20}	α_{b11v01}	α_{b11v12}
1	0.1	1.0	0.0	0.1	0.0	0.0	0.0
2	0.4	1.5	0.0	0.2	0.0	0.0	0.0
3	0.8	3.1	0.1	0.2	0.0	0.0	0.0
4	0.0	4.0	0.2	0.3	0.0	0.0	0.0
5	2.0	0.4	0.0	1.0	0.0	0.5	0.0

Table 5
Technical activities specifications of components version 2 in sub-system 1.

Version	$CACO_{b12v}$	CAC_{b12v}	α_{b12v10}	α_{b12v21}	α_{b12v20}	α_{b12v01}	α_{b12v12}
1	0.1	1.0	0.0	0.0	0.0	0.0	0.0
2	0.5	1.5	0.0	0.0	0.0	0.0	0.0
3	0.9	3.1	0.0	0.2	0.0	0.0	0.0
4	0.0	4.0	0.1	0.3	0.0	0.0	0.0
5	2.0	0.4	0.0	0.0	0.0	0.6	0.0

Table 6
Technical activities specifications of components version 3 in sub-system 1.

Version	$CACO_{b13v}$	CAC_{b13v}	α_{b13v10}	α_{b13v21}	α_{b13v20}	α_{b13v01}	α_{b13v12}
1	0.1	1.0	0.0	0.0	0.0	0.0	0.0
2	0.4	1.5	0.0	0.0	0.0	0.0	0.0
3	1.0	3.1	0.0	0.0	0.0	0.0	0.0
4	0.0	4.2	0.1	0.0	0.0	0.0	0.0
5	2.0	0.5	0.0	0.0	0.0	0.4	0.0

Table 7
Organizational activities specifications version 1 in sub-system 1.

Activities	CAS_{b1w}	β_{b11w10}	β_{b11w21}	β_{b11w20}	β_{b11w01}	β_{b11w12}
1	6.4	0.0	0.0	0.0	0.2	0.5
2	8.0	0.0	0.0	0.0	0.5	0.5
3	10.0	0.0	0.0	0.0	0.1	2.0

Table 8
Organizational activities specifications version 2 in sub-system 1.

Activities	CAS_{b1w}	β_{b12w10}	β_{b12w21}	β_{b12w20}	β_{b12w01}	β_{b12w12}
1	6.4	0.0	0.0	0.0	0.2	0.0
2	8.0	0.0	0.0	0.0	0.6	0.2
3	10.0	0.0	0.0	0.0	1.5	1.0

Table 9
Organizational activities specifications version 3 in sub-system 1.

Activities	CAS_{b1w}	β_{b13w10}	β_{b13w21}	β_{b13w20}	β_{b13w01}	β_{b13w12}
1	6.4	0.0	0.0	0.0	0.0	0.2
2	8.0	0.0	0.0	0.0	0.4	0.4
3	10.0	0.0	0.0	0.0	2.0	2.2

Table 10
Failure, repairs rates and performance rates of components type in sub-system 2.

Version	g_{h1}^2	λ_{b2h10}	μ_{b2h01}
1	80	0.05	0.30
2	100	0.06	0.35
3	150	0.03	0.45
4	180	0.02	0.40

Table 11
Fixed cost and unit cost of components in sub-system 2.

Supplier	*Version* 1		*Version* 2		*Version* 3		*Version* 4	
	CC_{kb21}	CO_{kb21}	CC_{kb22}	CO_{kb22}	CC_{kb23}	CO_{kb23}	CC_{kb24}	CO_{kb24}
1	30	20	35	20	60	35	80	40
2	25	18	40	22	55	32	83	42
3	27	19	38	21	57	34	81	41

Table 12
Technical activities specifications of components version 1 in sub-system 2.

Version	$CACO_{b21v}$	CAC_{b21v}	α_{b21v10}	α_{b21v01}
1	0.4	0.8	0.1	0.0
2	0.0	3.2	0.4	0.0
3	1.8	2.4	0.0	1.2

Table 13
Technical activities specifications of components version 2 in sub-system 2.

Version	$CACO_{b22v}$	CAC_{b22v}	α_{b22v10}	α_{b22v01}
1	0.4	1.0	0.1	0.0
2	0.0	3.2	0.4	0.0
3	1.8	2.8	0.0	1.1

Table 14
Technical activities specifications of components version 3 in sub-system 2.

Version	$CACO_{b23v}$	CAC_{b23v}	α_{b23v10}	α_{b23v01}
1	0.4	1.0	0.0	0.0
2	0.0	3.2	0.2	0.0
3	1.8	2.6	0.0	0.6

Table 15
Organizational activities specifications version 1 in sub-system 2.

Version	$CACO_{b24v}$	CAC_{b24v}	α_{b24v10}	α_{b24v01}
1	0.4	1.0	0.0	0.0
2	0.0	3.2	0.0	0.0
3	1.8	2.6	0.0	1.2

Table 16
Organizational activities specifications version 1 in sub-system 2.

Activities	CAS_{b2w}	β_{b21w10}	β_{b21w01}
1	30	0.1	1.4

Table 17
Organizational activities specifications version 2 in sub-system 2.

Activities	CAS_{b2w}	β_{b22w10}	β_{b22w01}
1	30	0.0	1.2

Table 18
Organizational activities specifications version 3 in sub-system 2.

Activities	CAS_{b2w}	β_{b23w10}	β_{b23w01}
1	30	0.0	0.8

Table 19
Organizational activities specifications version 4 in sub-system 2.

Activities	CAS_{b2w}	β_{b24w10}	β_{b24w01}
1	30	0.0	0.6

of the solving methods. The steps of the proposed GA are as follows:

- Chromosome introduction
- Initial population generation
- Offspring creation using operators
- Next generation creations

Table 20
Supplier's inventory.

Supplier	Sub-system 1			Sub-system 2			
	Version 1	Version 2	Version 3	Version 1	Version 2	Version 3	Version 4
1	6	12	7	13	5	8	10
2	5	7	9	4	11	6	7
3	10	5	7	8	9	11	6

Table 21
Purchase levels and discount for each supplier.

Purchase levels	Purchase level 1	Purchase level 2	Purchase level 3
q_f	0	200	400
z_f	1	0.95	0.90

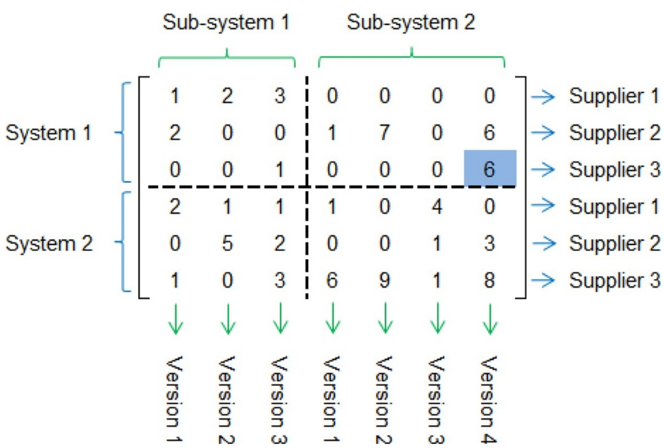

Fig. 3. A scheme of matrix *n*.

These four steps have been done 50 times. The resulting chromosome for GA contains four matrixes:

4.2.1. Matrix n

This matrix represents the number of different versions of the components in each sub-system of each system. The format of matrix *n* is presented in Fig. 3. The first three rows of this matrix represent the purchased components from each supplier for the first system and the second three rows of this matrix represent the purchased components from each supplier for the second system. The first three columns represent the number of each version in sub-system 1 and the second four columns represent the number of each version in sub-system 2. For example, according to matrix 4, in sub-system two of system one, six components of version four are purchased from the third supplier.

4.2.2. Matrices AC1 and AC2

These matrices represent the technical activity intensity of sub-system 1 and sub-system 2. The format of the *AC1* and *AC2* matrices are presented in Figs. 4 and 5. In this problem, five technical activities have been defined for sub-system one and three technical activities have been defined for sub-system two. The first three rows belong to system one and the second three rows belong to system two. For example, in matrix *AC1*, the intensity of the second type technical activity that has been done on components version two in system one is equal to 47.89%.

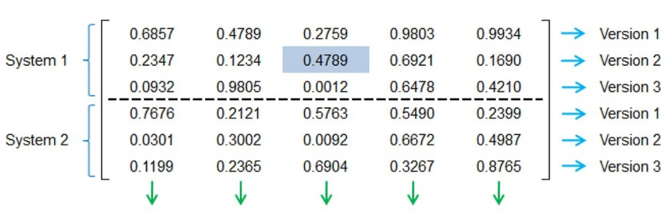

Fig. 4. Matrix *AC1*.

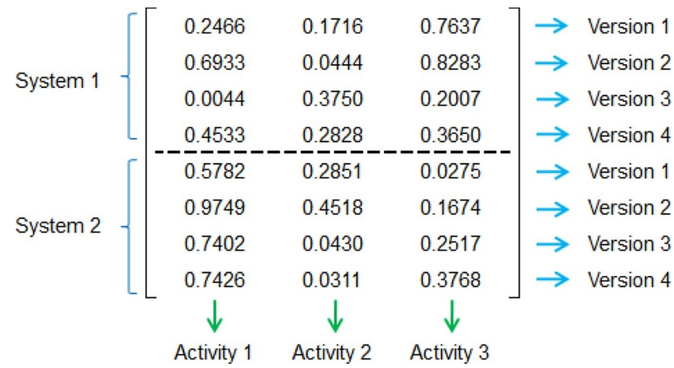

Fig. 5. Matrix *AC2*.

4.2.3. Matrix AS

This matrix represents the intensity of organizational activities on the sub-systems of both systems. The format of the matrix *AS* is presented in Fig. 6. The first row belongs to the first system, and the second row belongs to the second sub-system. The first three columns represent the intensity of the organizational activities on sub-system 1 and the next column represents the intensity of the organizational activities on sub-system 2.

4.2.4. GA operators

For the crossover operator, two random chromosomes are selected as parents and by using uniform crossover; two offspring are generated ([59,12]). For the mutation operator, one chromosome is selected randomly, and a random binary number created. If the number is equal to one, the corresponding element in the selected matrix randomly mutates. The crossover and mutation have been made on all matrices.

4.2.5. Parameter tuning of GA

The boundaries of tuned parameters are presented in Table 22. In this table, *npop* is the population size, p_c is the probability of crossover, and p_m is the probability of mutation. The result of the RSM implementation of GA is given in Tables 23–25. The final output including the optimal values of the GA parameters is represented in Fig. 7. Based on the results of RSM, the optimal parameters of GA are *npop* = 100,

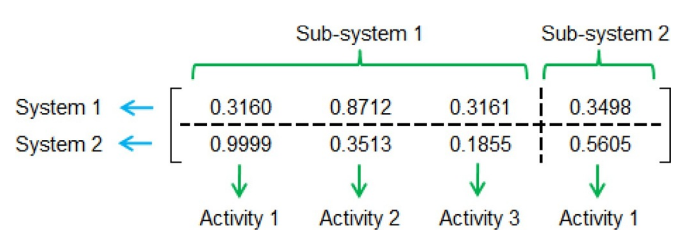

Fig. 6. Matrix AS2.

Table 22
Boundaries of GA tuned parameters.

GA Parameters	Lower value	Upper value
npop	50	100
p_c	0.3	0.7
p_m	0.1	0.3

Table 23
Computational results of GA.

Iteration	*npop*	p_c	p_m	Cost
1	50	0.3	0.1	1182.34
2	100	0.3	0.1	1043.20
3	50	0.7	0.1	1015.45
4	100	0.7	0.1	992.95
5	50	0.3	0.3	1097.69
6	100	0.3	0.3	1031.13
7	50	0.7	0.3	1054.03
8	100	0.7	0.3	967.87
9	50	0.5	0.2	960.21
10	100	0.5	0.2	958.91
11	75	0.3	0.2	1057.05
12	75	0.7	0.2	979.05
13	75	0.5	0.1	1019.60
14	75	0.5	0.3	1035.10
15	75	0.5	0.2	1051.80
16	75	0.5	0.2	1024.82
17	75	0.5	0.2	963.37
18	75	0.5	0.2	1054.12
19	75	0.5	0.2	1072.67

Table 24
GA estimated regression coefficients for cost.

Term	Coef	SE Coef	T	P-Value
Constant	1391.39	253.71	5.484	0
N_{pop}	5.36	6.73	0.797	0.446
P_c	-1157.22	727.12	-1.592	0.146
P_m	-1836.87	1236.51	-1.485	0.172
N_{pop}*N_{pop}	-0.05	0.04	-1.238	0.247
P_c * P_c	636.47	666.96	0.954	0.365
P_m * P_m	3475.65	2667.84	1.303	0.225
N_{pop} * P_c	2.43	3.12	0.778	0.457
N_{pop} * P_m	0.45	6.24	0.072	0.945
P_c * P_m	688.87	779.57	0.884	0.4

Table 25
GA analysis of variance for cost.

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Regression	9	37,210.0	37,210.0	4134.4	2.13	0.138
Linear	3	26,587.6	11,888.5	3962.8	2.04	0.179
N_{pop}	1	9964.0	1235.0	1235.0	0.64	0.446
P_c	1	16,164.9	4925.9	4925.9	2.53	0.146
P_m	1	458.6	4287.0	4287.0	2.20	0.172
Square	3	7916.9	7916.9	2639.0	1.36	0.317
N_{pop} * N_{pop}	1	14.6	2981.6	2981.6	1.53	0.247
P_c * P_c	1	4601.5	1771.0	1771.0	0.91	0.365
P_m * P_m	1	3300.8	3300.8	3300.8	1.70	0.225
Interaction	3	2705.5	2705.5	901.8	0.46	0.715
N_{pop} * P_c	1	1177.1	1177.1	1177.1	0.61	0.457
N_{pop} * P_m	1	9.9	9.9	9.9	0.01	0.945
P_c * P_m	1	1518.5	1518.5	1518.5	0.78	0.400
Residual	9	17,502.7	17,502.7	1944.75		
Lack-of-Fit	5	10,214.9	10,214.9	2043.0	1.12	0.469
Pure	4	7287.8	7287.8	1821.95		
Total	18	54,712.8				

$p_c \approx 0.6$ and $p_m \approx 0.2$.

4.3. MA for the proposed RAP

In the proposed MA, we added a hybridized local search to GA. The Pseudo-code of the proposed MA is presented in Fig. 8. It is to be mentioned that the chromosome structure and operators of the proposed MA are like GA. The procedure of the proposed MA is illustrated in the following subsections.

4.3.1. Local search structure

In the local search, at first one of the sub-systems is considered by implementing three steps as follows:

a The systems that satisfy their minimum availability constraints are sent to the second step. If in a solution, none of the systems satisfies its minimum availability constraints, the local search stops at this step.
b For each system the value of J_b is calculated as follows:

$$J_b = A_b(w_0) - A_{b0} \tag{14}$$

c The system with the largest amount of J_b is chosen.

Using these steps, the system with the largest distance and the least minimum availability has been chosen for neighborhood creation. For example, if the availability of a system in a solution is $A_1(w_0) = 0.94$ and $A_2(w_0) = 0.86$ and the minimum availability of two systems are $A_{b1} = 0.9$ and $A_{b2} = 0.8$, then $(J_1 = 0.94 - 0.9 - 0.04)$ and $(J_1 = 0.86 - 0.8 - 0.06)$ so the second system will be chosen for neighborhood creation. After selecting the system, two different neighborhoods have been created as follows:

(I) First neighborhood

In this neighborhood, a row of matrix n is randomly selected, and one unit is subtracted from the largest number of this row. For example, if the third row for the second system has been selected randomly, the largest number in this row is 9 and this number changed to (9-1 = 8). This neighborhood creation is illustrated in Fig. 9.

(II) Second neighborhood

This neighborhood is created by changing the activities level. In this type of neighborhood, a random row of *AC1, AC2,* and *AS* matrices is selected, and the largest value of this row is replaced by a random number between zero and one. Fig. 10 illustrates this type of neighborhood creation. In this figure, the second row of system 2 has been selected randomly.

After creation of the neighborhoods, the objective function is calculated for each neighborhood. Then a comparison will be made between the results that have been sent to the neighborhood and the two created neighbors. The result that has the minimum objective function will be selected as the output of the local search.

4.3.2. Parameter tuning of Memetic

The boundaries of the tuned parameters are presented in Table 26. In this table, *npop* is the population size, p_c is the probability of crossover and p_m is the probability of mutation. The result of RSM implementation of MA is in Tables 27–29. The final output including the optimal values of the MA parameters is represented in Fig. 11. Based on the results of RSM, the optimal parameters of MA are *npop* = 100, $p_c \approx 0.3$ and$p_m \approx 0.3$.

5. Result comparison

The optimal solutions for GA and MA are reported in Tables 30–33. In Table 30, the optimal values for the version of the components and the number of the suppliers are presented. Also, Tables 31–33 present the optimal values of the technical and organizational activities for

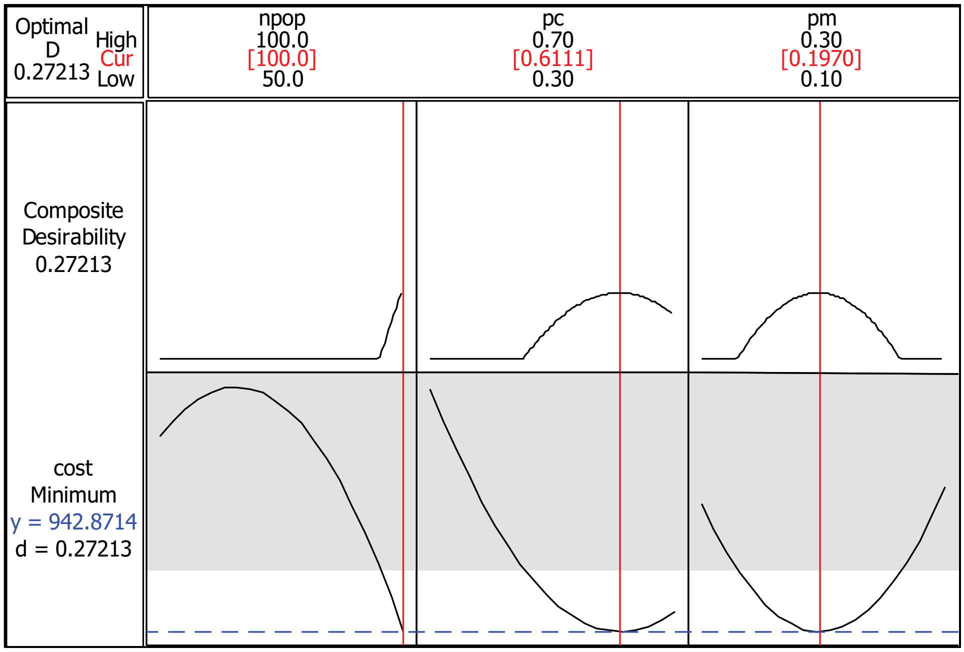

Fig. 7. GA optimal parameters.

Procedure memetic algorithm;
Begin
→ *Initialize population;*
For each individual do local-search individual;
Repeat
For individual =1 to crossovers do
→ *Select two parents Individual 1, Individual 2 \ in population randomly;*
→ *Individual 3:= crossover (Individual 1, Individual 2);*
→ *Individual 3:= local-search (Individual 3);*
Add Individual 3 to population;
End for;
For individual =1 to mutations do
→ *Select an Individual of population randomly;*
→ *Individual (m}:= mutate (Individual);*
→ *Individual {m}:= local-search (Individual {m});*
Add Individual {m} to population;
End for;
Population:=select (population);
If population converged then
For each individual of best populations do
Individual: = local-search (mutate (individual));
End if
Until terminate=true;
End

Fig. 8. The Pseudo-code of the proposed MA.

different values of w_0. In these tables, the numbers between parenthesis represent the number of technical and organizational activities and {} represents that no technical activity has been done on components. The convergence diagrams for these two algorithms are represented in Figs. 12–14. In these tables and figures, w_0 is considered as 300, 500 and 700.

The results from the performance comparison of GA and MA under different indices is reported in Table 34. According to these results, despite the fact that the computational time of MA is somewhat higher than GA, the objective functions obtained from MA are better than GA, and the best value and the variance for the objective function values in MA are less than GA.

The result of these analyses shows that the proposed MA works better than GA in producing initial solutions. Moreover, the MA has better performance in terms of the quality of the obtained solutions especially when w_0 increases.

6. Managerial insights

Increasing the number of parallel components in a system can stimulate the reliability and availability of the system but it is not sufficient. The related research shows improving the performance of each component in a system can improve the reliability of the entire system. Therefore, in this paper, both criteria are employed to improve the performance of the system. It should be noted that the required parameters for designing the model are formed based on the effects of each

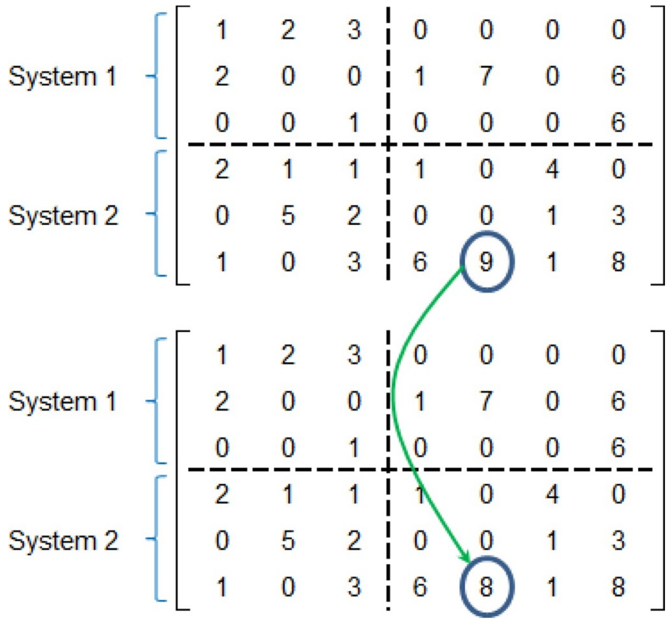

Fig. 9. The first type of neighborhood structure.

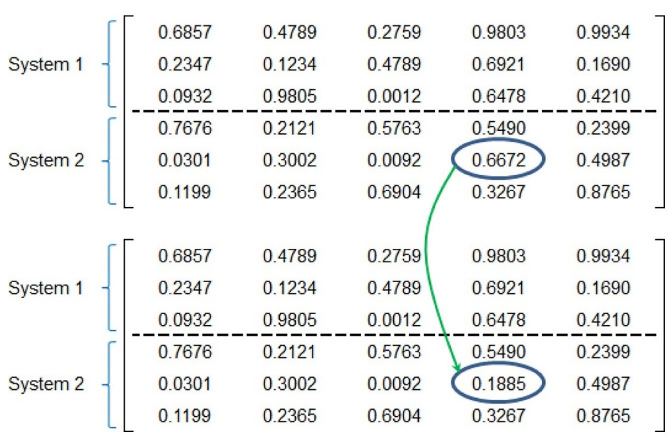

Fig. 10. Second type of neighborhood creation.

Table 26
Boundaries of memetic tuned parameters.

MA parameters	Lower value	Upper value
npop	50	100
p_c	0.3	0.7
p_m	0.1	0.3

approach and their related costs. In other words, if there is a budget capacity constraint, effort in promoting components' availability leads to a less redundancy assignment and vice versa.

The related research indicates that optimizing several systems simultaneously with regards to capacity constraints remains as a problem. The model proposed in this study considers the decision makers' authority in determining the model parameters. Using this authority, the decision makers can specify the faultless performance of the system and the minimal expected availability of each system with regards to expected performance rate. We also formulate a supplier selection model for each component and determine the amount of components supplied by each supplier by taking into account their incentive and discount policies. We discussed several examples for technical and organizational activities including the installation of a vibration monitoring system.

Table 27
Computational results of MA.

Iteration	*npop*	p_c	p_m	*Cost*
1	50	0.3	0.1	990.91
2	100	0.3	0.1	886.20
3	50	0.7	0.1	943.60
4	100	0.7	0.1	912.71
5	50	0.3	0.3	928.70
6	100	0.3	0.3	901.21
7	50	0.7	0.3	951.00
8	100	0.7	0.3	909.12
9	50	0.5	0.2	958.47
10	100	0.5	0.2	885.09
11	75	0.3	0.2	919.74
12	75	0.7	0.2	891.55
13	75	0.5	0.1	913.73
14	75	0.5	0.3	906.22
15	75	0.5	0.2	896.98
16	75	0.5	0.2	912.12
17	75	0.5	0.2	910.79
18	75	0.5	0.2	907.93
19	75	0.5	0.2	911.37

Table 28
Memetic estimated regression coefficients for cost.

Term	Coef	SE Coef	T	*P*-value
Constant	1307.13	79.523	16.437	0.000
N_{pop}	-6.42	2.109	-3.042	0.014
P_c	-187.35	227.909	-0.822	0.432
P_m	-636.30	387.572	-1.642	0.135
N_{pop}*N_{pop}	0.03	0.013	1.943	0.084
P_c * P_c	2.77	209.053	0.013	0.990
P_m * P_m	444.07	836.212	0.531	0.608
N_{pop} * P_c	1.49	0.977	1.520	0.163
N_{pop} * P_m	3.31	1.955	1.694	0.125
P_c * P_m	318.81	244.350	1.305	0.224

Table 29
Memetic analysis of variance for cost.

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Regression	9	11,108.3	11,108.3	1234.26	6.46	0.005
Linear	3	8042.2	4011.3	1337.09	7.00	0.010
N_{pop}	1	7747.9	1768.3	1768.32	9.26	0.014
P_c	1	35.3	129.1	129.11	0.68	0.432
P_m	1	259.1	515.0	514.98	2.70	0.135
Square	3	1751.1	1751.1	583.69	3.05	0.084
N_{pop} * N_{pop}	1	1687.7	721.1	721.14	3.77	0.084
P_c * P_c	1	9.5	0.0	0.03	0.00	0.990
P_m * P_m	1	53.9	53.9	53.88	0.28	0.608
Interaction	3	1315.0	1315.0	438.35	2.29	0.147
N_{pop} * P_c	1	441.5	441.5	441.49	2.31	0.163
N_{pop} * P_m	1	548.3	548.3	548.30	2.87	0.125
P_c * P_m	1	325.3	325.3	325.25	1.70	0.224
Residual	9	1719.6	1719.60	191.06		
Lack-of-Fit	5	1562.1	1562.1	312.43	7.94	0.033
Pure	4	157.4	157.40	39.36		
Total	18	12,827.9				

7. Conclusion and future studies

In this paper, we have developed the range of optimization parameters to model the RAP as close to the real world as possible and determine the following variables:

- Number of assigned components in each sub-system of each system
- The version of assigned components in each sub-system of each system
- The level of purchasing components from each supplier

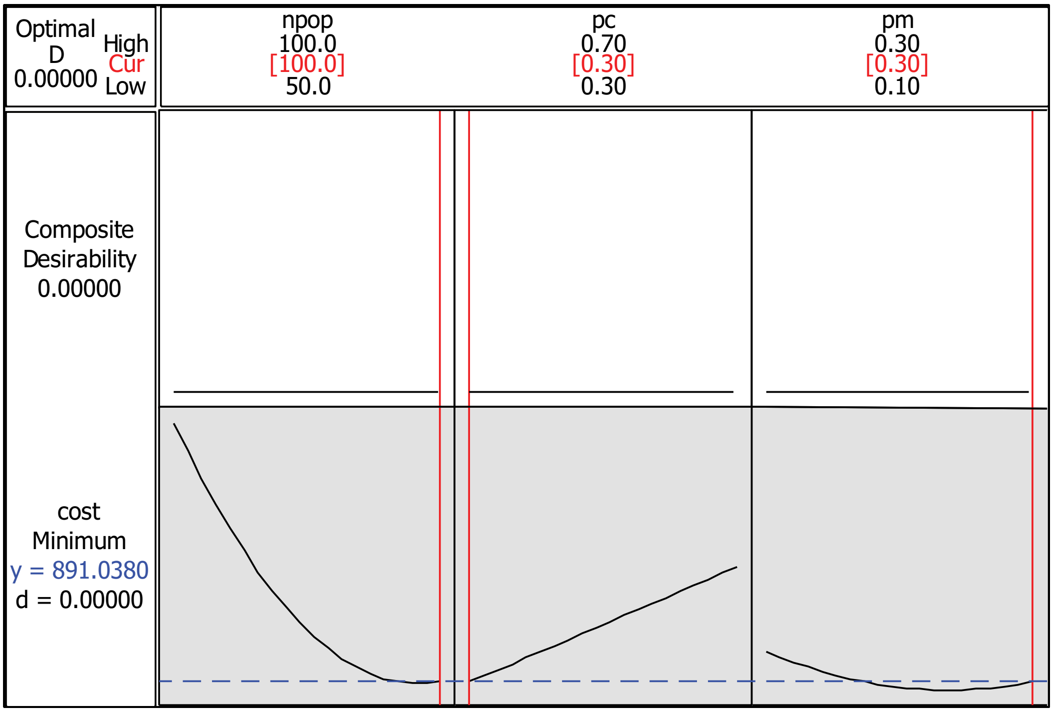

Fig. 11. Memetic optimal parameters.

Table 30
Optimal solutions for GA and MA.

w_0	Algorithm	System	Sub-system	Version	Supplier	n_{kbih}
300	MA	1	1	3	2	5
			2	1	1	5
		2	1	3	1	6
			2	3	3	2
	GA	1	1	1	2	4
			2	3	1	5
		2	1	1	2	5
			2	3	1	5
500	MA	1	1	2	2	7
			2	1	1	7
				2	2	1
		2	1	2	2	8
			2	1	1	7
				2	2	1
	GA	1	1	2	2	8
			2	2	1	4
				3	2	5
		2	1	3	1	7
				3	2	2
			2	2	2	6
700	MA	1	1	1	2	3
				2	2	6
			2	2	3	6
		2	1	1	2	3
				2	2	7
			2	2	3	6
	GA	1	1	3	1	1
				1	2	1
				2	2	7
				1	3	1
			2	1	1	5
				1	2	5
		2	1	3	1	4
				1	3	1
				3	3	5
			2	2	2	1
				3	2	7

Table 31
Optimal values of technical and organizational activities for $w_0 = 300$.

Algorithm	System	Sub-system	Version	AC_{bihv}	AS_{biw}
MA	1	1	2	0.0240(1)	
				0.0227(2)	0.6174(1)
				0.0375(3)	0.4305(2)
				0.0212(4)	0.3731(3)
				0.0034(5)	
		2	–	{}	0.5298(1)
	2	1	1	0.1072(1)	
				0.0774(2)	0.4282(1)
				0.0707(3)	0.6117(2)
				0.1024(4)	0.4618(3)
				0.0375(5)	
		2	–	{}	0.4806(1)
GA	1	1	2	0.0646(1)	
				0.0726(2)	0.5833(1)
				0.2397(3)	0.7142(2)
				0.4185(4)	0.7161(3)
				0.0266(5)	
		2	–	{}	0.8636(1)
	2	1	2	0.2216(1)	
				0.0125(2)	0.4897(1)
				0.0196(3)	0.6114(2)
				0.0295(4)	0.3809(3)
				0.3945(5)	
		2	–	{}	0.34(1)

- The type of technical and organizational activities
- Intensity of technical and organizational activities

To solve the developed mathematical model, we proposed an MA to determine the optimal solutions. The performance of proposed algorithm is compared with GA. Both algorithms are tuned by RSM to enhance accuracy and precision of the algorithms. The results illustrate that MA has a better performance than GA in terms of the objective value and also the production of initial solutions. The comparisons between the performance of the algorithms show that the performance of MA in generating initial solutions and finding the final solution is better than GA. By increasing the minimum performance level to reach

Table 32
Optimal values of technical and organizational activities for $w_0 = 500$.

Algorithm	System	Sub-system	Version	AC_{bihv}	AS_{biw}
MA	1	1	2	0.0069(3)	0.6434(1)
					0.7025(2)
					0.5769(3)
		2	–	{}	0.6678(1)
	2	1	–	{}	0.7763(1)
					0.2947(1)
					0.5102(3)
		2	1	0.0421(1)	0.2275(1)
				0.0348(2)	
				0.0203(3)	
GA	1	1	2	0.1742(1)	
				0.1491(2)	0.4103(1)
				0.1077(3)	0.4581(2)
				0.5127(4)	0.3585(3)
				0.3487(5)	
		2	–	{}	0.3531(1)
	2	1	1	0.4212(1)	
				0.5386(2)	
				0.3762(3)	
				0.1124(4)	0.9430(1)
				0.4308(5)	0.5613(2)
			2	0.0863(1)	0.5333(3)
				0.1438(2)	
				0.0.611(3)	
				0.0189(4)	
				0.0878(5)	
		2	–	{}	0.3727(1)

Table 33
Optimal values of technical and organizational activities for $w_0 = 700$.

Algorithm	System	Sub-system	Version	AC_{bihv}	AS_{biw}
MA	1	1	–	{}	0.5546(1)
					0.6461(2)
					0.6135(3)
		2	–	{}	0.3410(1)
	2	1	2	0.3040(1)	
				0.1451(2)	0.6699(1)
				0.0046(3)	0.4756(2)
				0.0120(4)	0.4760(3)
				0.3551(5)	
		2	–	{}	0.3292(1)
GA	1	1	1	0.0690(1)	0.5619(1)
				0.0326(2)	0.3525(2)
				0.0212(3)	0.3252(3)
				0.1178(4)	
				0.0177(5)	
			2	0.0352(1)	
				0.2294(2)	
				0.1556(3)	
				0.4308(4)	
				0.0621(5)	
			3	0.5719(1)	
				0.4688(2)	
				0.5035(3)	
				0.3110(4)	
				0.5013(5)	
		2	–	{}	0.6377(1)
	2	1	1	0.0867(1)	0.6655(1)
				0.0225(2)	0.4741(2)
				0.0488(3)	0.3485(3)
				0.0291(4)	
				0.3321(5)	
			3	0.0160(1)	
				0.4252(2)	
				0.1331(3)	
				0.0055(4)	
				0.0017(5)	
		2	–	{}	0.5150(1)

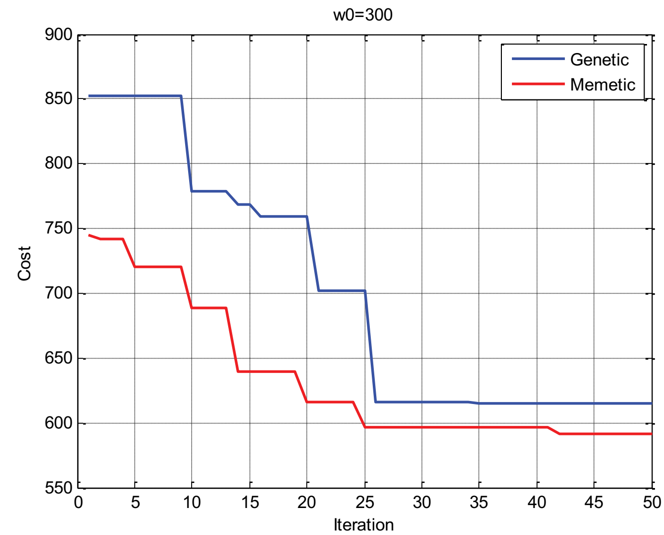

Fig. 12. Graphical comparison for $w_0 = 300$.

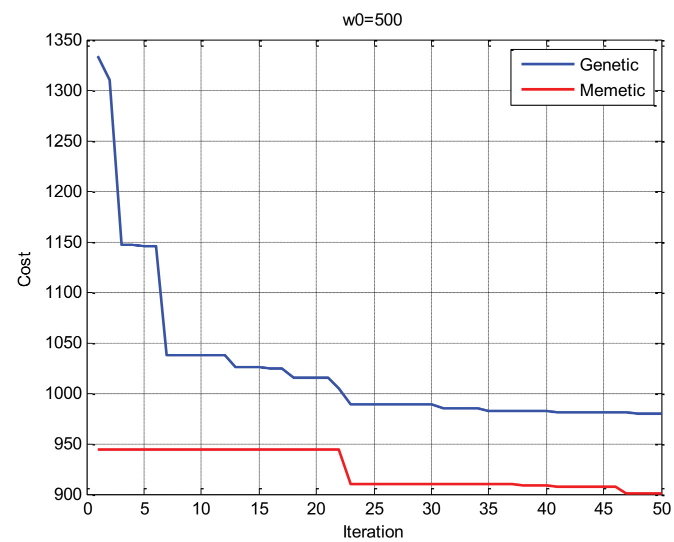

Fig. 13. Graphical comparisons for $w_0 = 500$.

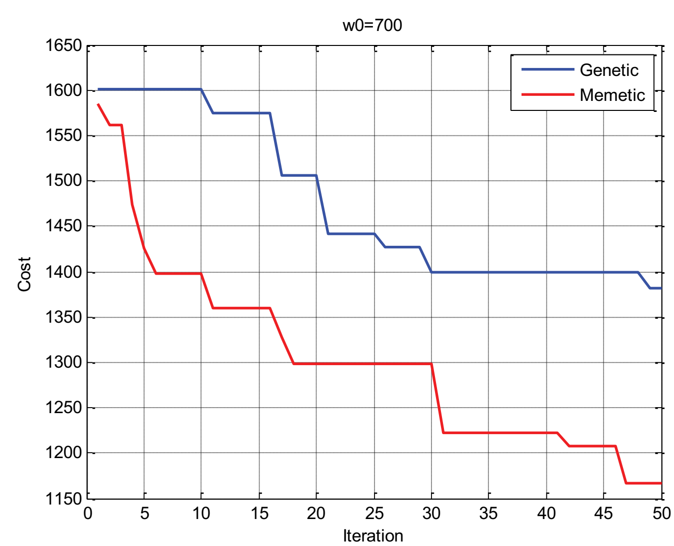

Fig. 14. Graphical comparison for $w_0 = 700$.

Table 34
Performance comparison of GA and MA under different indices.

Minimum performance rate	300		500		700	
Algorithm	GA	MA	GA	MA	GA	MA
Best objective value	614.4950	591.8912	980.1612	901.2137	1381.5859	1167.3857
Worst objective value	617.5290	593.3130	984.8653	903.0981	1389.7509	1171.2417
Variance of objective values	3.7214	1.0052	5.06527	1.5921	13.7428	6.1792
CPU time (s)	122	134	129	138	137	149

the minimum desired availability, the difference between the final solutions of the two algorithms becomes more tangible. The reason for this is, at the low performance levels, it is easier to find solutions that satisfy the constraints of the minimum availability level of system but by increasing the minimum performance level, the feasible solution space will be limited, and it will be difficult to achieve efficient and feasible solutions.

The most important limitation of the present research is the inability to use exact solution methods for the optimization model. Given that the systems availability level in this study is calculated using the UGF method, the model proposed in this study is nonlinear. In addition to the computational complexity of the redundancy allocation problem with multi-state components, the main reason for using the meta-heuristic algorithms and optimizing the multi-state systems is that it is not possible to calculate system availability with a mathematical function.

For further studies, one can consider different efficient algorithms or hybridize them to achieve better solutions. The other types of RAP like weighted k-out-of-n systems or multi-state k-out-of-n systems can also be considered as the basic problem.

Acknowledgment

The authors would like to thank the anonymous reviewers and the editor for their insightful comments and suggestions. Dr. Madjid Tavana is grateful for the partial support he received from the Czech Science Foundation (GA ČR19-13946S) for this research.

References

[1] Ardakan MA, Hamadani AZ. Reliability–redundancy allocation problem with cold-standby redundancy strategy. Simul Model Pract Theory 2014;42:107–18.
[2] Chambari A, Najafi AA, Rahmati SHA, Karimi A. An efficient simulated annealing algorithm for the redundancy allocation problem with a choice of redundancy strategies. Reliab Eng Syst Saf 2013;119:158–64.
[3] Chen TC, You PS. Immune algorithms-based approach for redundant reliability problems with multiple component choices. Comput Ind 2005;56(2):195–205.
[4] Chen TC. IAs based approach for reliability redundancy allocation problems. Appl Math Comput 2006;182(2):1556–67.
[5] Chern MS. On the computational complexity of reliability redundancy allocation in a series system. Oper Res Lett 1992;11(5):309–15.
[6] Coit DW, Liu JC. System reliability optimization with k-out-of-n subsystems. Int J Reliab Qual Saf Eng 2000;7(02):129–42.
[7] Coit DW, Smith AE. Penalty guided genetic search for reliability design optimization. Comput Ind Eng 1996;30(4):895–904.
[8] Di Bona G, Falcone D, Silvestri A, Forcina A. IFM target 2.0: an innovative method to define reliability target for prototype systems. Int J Adv Manuf Technol 2018;95(9–12):3349–67.
[9] Di Bona G, Silvestri A, Forcina A, Petrillo A. Total efficient risk priority number (TERPN): A new method for risk assessment. J Risk Res 2018;21(11):1384–408.
[10] Guilani PP, Sharifi M, Niaki STA, Zaretalab A. 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 2014;129:29–35.
[11] Ha C, Kuo W. Reliability redundancy allocation: An improved realization for non-convex nonlinear programming problems. Eur J Oper Res 2006;171(1):24–38.
[12] Hajipour V, Khodakarami V, Tavana M. The redundancy queuing-location-allocation problem: a novel approach. IEEE Trans Eng Manag 2014;61(3):534–44.
[13] Hong L, Ouyang M, Peeta S, He X, Yan Y. Vulnerability assessment and mitigation for the Chinese railway system under floods. Reliab Eng Syst Saf 2015;137:58–68.
[14] Khalili-Damghani K, Abtahi AR, Tavana M. A decision support system for solving multi-objective redundancy allocation problems. Qual Reliab Eng Int 2014;30(8):1249–62.
[15] Kim HG, Bae CO, Park DJ. Reliability-redundancy optimization using simulated annealing algorithms. J Qual Maint Eng 2006;12(4):354–63.
[16] Kulturel-Konak S, Smith AE, Coit DW. Efficiently solving the redundancy allocation problem using tabu search. IIE Trans 2003;35(6):515–26.
[17] Kuo W, Prasad VR. An annotated overview of system-reliability optimization. IEEE Trans Reliab 2000;49(2):176–87.
[18] Levitin G, Lisnianski A. Survivability maximization for vulnerable multi-state systems with bridge topology. Reliab Eng Syst Saf 2000;70(2):125–40.
[19] Levitin G, Lisnianski A. A new approach to solving problems of multi-state system reliability optimization. Qual Reliab Eng Int 2001;17(2):93–104.
[20] Lisnianski A, Levitin G. Multi-state system reliability. Assessment, optimization and applications. Series on Quality, Reliability and Engineering Statistics 6. (Ed) World Scientific; 2003. p. 28.
[21] Levitin G, Xing L, Ben-Haim H, Dai Y. Reliability of series-parallel systems with random failure propagation time. IEEE Trans Reliab 2013;62(3):637–47.
[22] Levitin G, Xing L, Johnson BW, Dai Y. Mission reliability, cost and time for cold standby computing systems with periodic backup. IEEE Trans Comput 2014;64(4):1043–57.
[23] Li W, Zuo MJ. Reliability evaluation of multi-state weighted k-out-of-n systems. Reliab Eng Syst Saf 2008;93(1):160–7.
[24] Liang YC, Chen YC. Redundancy allocation of series-parallel systems using a variable neighborhood search algorithm. Reliab Eng Syst Saf 2007;92(3):323–31.
[25] Liang YC, Wu CC. A variable neighbourhood descent algorithm for the redundancy allocation problem. Ind Eng Manag Syst 2005;4(1):94–101.
[26] Lins ID, Droguett EL. Redundancy allocation problems considering systems with imperfect repairs using multi-objective genetic algorithms and discrete event simulation. Simul Model Pract Theory 2011;19(1):362–81.
[27] Lisnianski A, Levitin G, Ben-Haim H, Elmakis D. Power system structure optimization subject to reliability constraints. Electr Power Syst Res 1996;39(2):145–52.
[28] Liu Y, Chen Y, Jiang T. On sequence planning for selective maintenance of multi-state systems under stochastic maintenance durations. Eur J Oper Res 2018;268(1):113–27.
[29] Liu Y, Huang HZ, Wang Z, Li Y, Yang Y. A joint redundancy and imperfect maintenance strategy optimization for multi-state systems. IEEE Trans Reliab 2013;62(2):368–78.
[30] Liu Y, Liu Q, Xie C, Wei F. Reliability assessment for multi-state systems with state transition dependency. Reliab Eng Syst Saf 2019;188:276–88.
[31] Lyu MR, Rangarajan S, Van Moorsel AP. Optimal allocation of test resources for software reliability growth modeling in software development. IEEE Trans Reliab 2002;51(2):183–92.
[32] Mo Y, Cui L, Xing L, Zhang Z. Performability analysis of large-scale multi-state computing systems. IEEE Trans Comput 2017;67(1):59–72.
[33] Mo Y, Xing L, Zhong F, Zhang Z. Reliability evaluation of network systems with dependent propagated failures using decision diagrams. IEEE Trans Dependable Secure Comput 2015;13(6):672–83.
[34] Mousavi SM, Alikar N, Niaki STA, Bahreininejad A. Two tuned multi-objective meta-heuristic algorithms for solving a fuzzy multi-state redundancy allocation problem under discount strategies. Appl Math Model 2015;39(22):6968–89.
[35] Nahas N, Nourelfath M, Ait-Kadi D. Coupling ant colony and the degraded ceiling algorithm for the redundancy allocation problem of series–parallel systems. Reliab Eng Syst Saf 2007;92(2):211–22.
[36] Ouiddir R, Rahli M, Meziane R, Zeblah A. Ant colony optimization for new redesign problem of multi-state electrical power systems. J Electr Eng Bratisl 2004;55(3/4):57–63.
[37] Ouzineb M, Nourelfath M, Gendreau M. Tabu search for the redundancy allocation problem of homogenous series–parallel multi-state systems. Reliab Eng Syst Saf 2008;93(8):1257–72.
[38] Ouzineb M, Nourelfath M, Gendreau M. A heuristic method for non-homogeneous redundancy optimization of series-parallel multi-state systems. J Heuristics 2011;17(1):1–22.
[39] Pandey M, Zuo MJ, Moghaddass R. Selective maintenance modeling for a multistate system with multistate components under imperfect maintenance. IIE Trans 2013;45(11):1221–34.
[40] Ravi V, Reddy PJ, Zimmermann HJ. Fuzzy global optimization of complex system reliability. IEEE Trans Fuzzy Syst 2000;8(3):241–8.
[41] Safaei N, Tavakkoli-Moghaddam R, Kiassat C. Annealing-based particle swarm optimization to solve the redundant reliability problem with multiple component choices. Appl Soft Comput 2012;12(11):3462–71.
[42] Sharma VK, Agarwal M. Ant colony optimization approach to heterogeneous redundancy in multi-state systems with multi-state components. Proceedings of the eighth international conference on reliability, maintainability and safety, ICRMS. IEEE; 2009. p. 116–21.

[43] Shrestha A, Xing L, Coit DW. An efficient multistate multivalued decision diagram-based approach for multistate system sensitivity analysis. IEEE Trans Reliab 2010;59(3):581–92.
[44] Tavakkoli-Moghaddam R, Safari J, Sassani F. Reliability optimization of series-parallel systems with a choice of redundancy strategies using a genetic algorithm. Reliab Eng Syst Saf 2008;93(4):550–6.
[45] Teimouri M, Zaretalab A, Niaki STA, Sharifi M. An efficient memory-based electromagnetism-like mechanism for the redundancy allocation problem. Appl Soft Comput 2016;38:423–36.
[46] Tian Z, Levitin G, Zuo MJ. A joint reliability-redundancy optimization approach for multi-state series-parallel systems. Reliab Eng Syst Saf 2009;94(10):1568–76.
[47] Tian Z, Zuo MJ, Huang H. Reliability-redundancy allocation for multi-state series-parallel systems. IEEE Trans Reliab 2008;57(2):303–10.
[48] Ushakov IA. Optimal standby problems and a universal generating function. Soviet J Comput Syst Sci 1987;25(4):79–82.
[49] Wang W, Di Maio F, Zio E. Three-loop Monte Carlo simulation approach to multi-state physics modeling for system reliability assessment. Reliab Eng Syst Saf 2017;167:276–89.
[50] Wang Y, Li L. Heterogeneous redundancy allocation for series-parallel multi-state systems using hybrid particle swarm optimization and local search. IEEE Trans Syst Man Cybern Part A Syst Hum 2012;42(2):464–74.
[51] Yalaoui A, Châtelet E, Chu C. A new dynamic programming method for reliability & redundancy allocation in a parallel-series system. IEEE Trans Reliab 2005;54(2):254–61.
[52] Yalaoui A, Chu C, Châtelet E. Reliability allocation problem in a series-parallel system. Reliab Eng Syst Saf 2005;90(1):55–61.
[53] Yeh WC, Hsieh TJ. Solving reliability redundancy allocation problems using an artificial bee colony algorithm. Comput Oper Res 2011;38(11):1465–73.
[54] You PS, Chen TC. An efficient heuristic for series-parallel redundant reliability problems. Comput Oper Res 2005;32(8):2117–27.
[55] Yuan L, Meng XY. Reliability analysis of a warm standby repairable system with priority in use. Appl Math Model 2011;35(9):4295–303.
[56] Zaretalab A, Hajipour V, Sharifi M, Shahriari MR. A knowledge-based archive multi-objective simulated annealing algorithm to optimize series-parallel system with choice of redundancy strategies. Comput Ind Eng 2015;80:33–44.
[57] Zhao X, Wang S, Wang X, Cai K. A multi-state shock model with mutative failure patterns. Reliab Eng Syst Saf 2018;178:1–11.
[58] Zhao X, Wu C, Wang S, Wang X. Reliability analysis of multi-state k-out-of-n: G system with common bus performance sharing. Comput Ind Eng 2018;124:359–69.
[59] Gen M, Cheng R. Genetic Algorithms & Engineering Design Hoboken: John Wiley & Sons, Inc.; 1997.