



A note on a reliability redundancy allocation problem using a tuned parameter genetic algorithm

Amirhossain Chambari¹ · Javad Sadeghi² ·
Fakhri Bakhtiari³ · Reza Jahangard⁴

Accepted: 30 September 2015 / Published online: 9 October 2015
© Operational Research Society of India 2015

Abstract This paper presents an improved continuous genetic algorithm (CGA) to optimize the reliability redundancy allocation problem (RRAP) which determines the best redundancy strategies, the number of components, and levels of each subsystem to maximize the system reliability. In this system, both active and cold-standby redundancies can be chosen for individual subsystems. The RRAP belongs to NP-hard problems in the computational complexity theory that is the main reason for employing CGA to solve it. In addition, the response surface methodology (RSM) is used to increase the performance of CGA considering the design of experiments. This algorithm employs a new chromosome so that frees offspring to repair during the evolution process. Considering several numerical examples, the proposed algorithm presents better solutions than the previous studies based on the system reliability. Finally, the conclusion and future research are considered.

Electronic supplementary material The online version of this article (doi:10.1007/s12597-015-0230-9) contains supplementary material, which is available to authorized users.

✉ Javad Sadeghi
Keeyarash@gmail.com; Jsadeghl@binghamton.edu

Amirhossain Chambari
Amir.chambari@gmail.com

Fakhri Bakhtiari
raha19699@gmail.com

Reza Jahangard
jahangard6@gmail.com

- ¹ Young Researchers and Elite Club, Qazvin Branch, Islamic Azad University, Qazvin, Iran
- ² Systems Science and Industrial Engineering, State University of New York at Binghamton, Binghamton, NY 13902, USA
- ³ Department of Business Administration, Faculty of Management, The West Tehran Branch, Payame Noor University, Tehran, Iran
- ⁴ Department of Business Administration, Persian Gulf International Branch, Islamic Azad University, Khorramshahr, Iran

Keywords Redundancy allocation problem · Series–parallel systems · Continuous genetic algorithm · Parameter tuning · Response surface methodology (RSM)

1 Introduction

System reliability is one of the most important issues to design various types of electrical and mechanical systems. In the reliability theory, redundancy is applied to improve system reliability. In addition, reliability redundancy allocation problem (RRAP) is an important class in modeling reliability systems so that belongs to NP-hard problems. In the RRAP literature, there are several considered RRAPs such as series, parallel, series–parallel [1–4], and k-out-of-n [5]. Hence, Fig. 1 shows a series–parallel system in which includes s subsystems and n redundant machines.

There are two categories of RRAPs. The first class is a system including component mixing. The second category is a system without component mixing, which meant that the components allocated in the same subsystem in which should be the same type [6]. Furthermore, active and standby approaches are two main classifications for the redundancy strategies. The situation that all components are operated simultaneously from the time zero is called active redundancy versus standby redundancy that the redundant components are used sequentially during component failure times [7]. Moreover, standby redundancy is categorized into three different types, namely, cold [8–10], hot [11, 12], and warm [13–15] so that frees the component to fail before it operates in the cold standby redundancy versus warm standby that components are more likely to be failed before operating. Further, the failure pattern of components is freed from the idleness or the operation of the components with respect to the hot standby redundancy. In the majority of studies in the RRAP literature, it is assumed that only one type of redundancy can be used to investigate systems: active or cold standby. In the real world, both the redundancy strategies are applied simultaneously in the design of systems. Therefore, this paper studies redundancy strategy with respect to active and cold standby components.

In system-reliability optimization, Kuo and Prasad [16] provided a good overview. However, there are several studies in the scope of cold standby redundancy so that

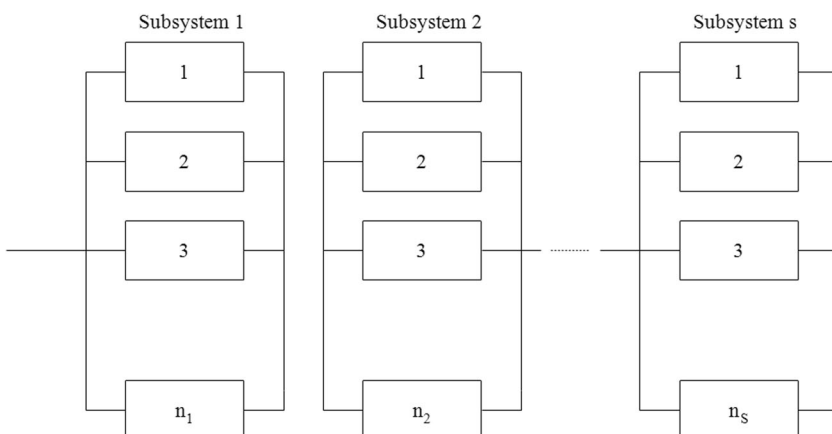


Fig. 1 Graphical representation of series–parallel system

Tillman et al. [17] evidenced in their research; on the other hand, they investigate 100 papers relating to the reliability optimization researches with different types of redundancy. According to strictly cold standby redundancy, Coit [5] discussed a problem that considered imperfect switching and k -Erlang distributed time-to-failures, and then presented an integer programming solution to optimize the RRAP. To consider problems of active and cold standby redundancy is near to real-world situations in which there are few studies because of its computational complexity. Considering multiple k -out-of- n subsystems in the series, Coit and Liu [18] provided a solution methodology to optimize a RRAP for systems so that they assumed that the redundancy strategy is predetermined for each subsystem. Regarding zero–one integer programming method, Coit [19] extended another methodology to maximize system reliability with respect to either active or cold-standby redundancy can be selectively chosen for individual subsystems. Thus, he developed the previous solution methodology by adding several decision variables. Next, Tavakkoli-Moghaddam et al. [20] improved the system reliability the RRAP [19] by employing the genetic algorithm (GA). Tian, et al. [21] extended an approach to optimize systems with multi-state components with respect to the factors that influence system reliability and system life cycle cost. Moreover, Sharma, et al. [22] proposed an efficient algorithm to find a near-optimal solution to minimize the configuration cost of heterogeneous multistate series–parallel systems while a desired reliability index should be satisfied.

With these explanations, the purpose of this article is to develop a better solution methodology to improve the RRAP optimization. In other words, this paper illustrates a continuous GA (CGA) for series–parallel RRAP used in Tavakkoli-Moghaddam et al. [20]. Since the meta-heuristics are impressed by their parameters, we utilize the response surface methodology (RSM) in the design of experiments to tune parameters of the proposed algorithm. As innovations, we can point to CGA that applies a new chromosome so that free offspring from reparation during the generation; in addition, employing RSM to calibrate parameters of CGA is another contribution. Table 1 provides several studies relevant to GA in the literature.

The paper is organized as follows. Section 2 defines the assumptions and the mathematical model. The proposed algorithm is presented in Section 3. Parameter tuning and the comparison of CGA are elaborated in Sections 4 and 5, respectively. Finally, Section 6 provides the conclusion and future research.

Table 1 Several studies relevant to genetic algorithm

Study	Solution algorithm	Tuning parameters
Pasandideh, et al. [23]	GA	No
Sadeghi, et al. [24]	GA	RSM
Nachiappan and Jawahar [25]	GA	No
Tavakkoli-Moghaddam et al. [20]	GA	No
The Proposed algorithm	Tuned-Hybrid GA	RSM

2 Mathematical model

This section presents a RRAP without component mixing for the series–parallel redundant reliability system considering S subsystem so that individual subsystems can select either active or cold standby redundancy strategy. In addition, two separable linear constraints are considered. Several typical assumptions are,

- a. The failed components, which are not repaired, do not damage to the system.
- b. Failures of individual components are statistically-independent.
- c. The states of the elements and the system are either good or have failed (and no other states are considered).

Nomenclatures used in this paper are as follows.

S	Number of subsystems
i	An index used for a subsystem; $i=1,2,3,\dots,S$
n_i	Number of components used in subsystem; $n_i \in \{1,2,\dots,n_{\max,i}\}$
n	(n_1, n_2, \dots, n_s)
m_i	Number of available component choices for a subsystem i
z_i	Index of component choice used for a subsystem i ; $z_i \in \{1,2,\dots,m_i\}$
Z	(z_1, z_2, \dots, z_s)
A	Set of all subsystems using active redundancy
CS	Set of all subsystems using cold standby redundancy
$n_{\max,i}$	Upper bound for $n_i (n_i \leq n_{\max,i})$
$r_{iz_i}(t)$	Reliability at time t for z_i^{th} available component for subsystem i
λ_{iz_i}, k_{iz_i}	Scale and shape parameters the Gamma distribution for z_i^{th} available component for subsystem i
$f_{iz_i}^{(j)}(t)$	pdf of z_i^{th} failure arrival for subsystem i , where is the sum of j iid component failure times
W	System-level constraint limit for weight
c_{ij}, w_{ij}	Cost and weight of the j^{th} available component for the subsystem i , respectively
$\rho_A(t)$	Failure-detection / switching reliability at time t (scenario 1; continual sensing)
ρ_i	Failure-detection/switching reliability success probability (scenario 2; active only in response to a failure state)
$R(t; z, n)$	System reliability at time t for designing vectors z and n
$\tilde{R}(t; z, n)$	Approximation of $R(t; z, n)$
t	Mission time (fixed)

Where the decision variables are: redundancy strategies, selected components, and the number of components for each subsystem.

According to the mentioned nomenclature, the mathematical model of RRAP can be written as follows.

$$\text{Max } z = R(t; z, n) \tag{1}$$

$$\text{s.t,} \quad \sum_i c_{iz_i} n_i \leq C \tag{2}$$

$$\sum_i w_{iz_i} w_i \leq W \tag{3}$$

$$n_i \in \{1, 2, \dots, n_{\max,i}\} \tag{4}$$

$$z_i \in \{1, 2, \dots, m_i\} \tag{5}$$

Equation (3) presents the overall reliability system; in addition, Eqs. (4) and (5) satisfy the constraints on the available cost and weight, respectively. Then, Eqs. (7) and (8) obtain the system reliability as two scenarios [5]:

- Scenario 1: continual detector/switch operation,
- Scenario 2: switch active only in response to a failure,

$$R(t; z, n) = \prod_{i \in A} (1 - (1 - r_{iz_i}(t))^{n_i}) \times \prod_{i \in CS} \left(r_{iz_i}(t) + \sum_{j=1}^{n_i-1} \int_0^t \rho_i^j(u) f_{iz_i}^{(j)}(u) r_{iz_i}(t-u) du \right) \tag{6a}$$

$$R(t; z, n) = \prod_{i \in A} (1 - (1 - r_{iz_i}(t))^{n_i}) \times \prod_{i \in CS} \left(r_{iz_i}(t) + \sum_{j=1}^{n_i-1} \rho_i^j \int_0^t f_{iz_i}^{(j)}(u) r_{iz_i}(t-u) du \right) \tag{6b}$$

In the CS, the subsystem reliability is the sum of the n_i mutually exclusive probabilities. Just after the product for CS, the first term in Eq. (6) is $r_{iz_i}(t)$ that represents the probability that no cold standby redundant components are required during the mission interval. Other subsequent terms represent $n_i - 1$ mutually exclusion probabilities that there are between one and $n_i - 1$ failures. Note that the switch is available to perform its function at the time of the failure, and a redundant component is always operating at time t .

2.1 Approximation of reliability

Regarding Eqs. (7) and (9) obtains the system reliability approximation in which only one difference can be seen in the product term for CS.

$$\tilde{R}(t; z, n) = \prod_{i \in A} (1 - (1 - r_{iz_i}(t))^{n_i}) \times \prod_{i \in CS} \left(r_{iz_i}(t) + \delta_i(t, n_i) \sum_{j=1}^{n_i-1} \int_0^t f_{iz_i}^{(j)}(u) r_{iz_i}(t-u) du \right) \tag{7}$$

Where $\delta_i(t, n_i) = \{ \rho_i(t) \rho_i^{n_i-1}$, as a probability function to determine the probability of the switch at hours for each subsystem, which it is considered 0.99 in this paper.

A specified form of Eq. (9) shall be determined if k -Erlang distribution is considered for component time-to-failure. The k -Erlang distribution as a special form of Gama distribution can model diverse constant and increase hazard functions. According to k -Erlang distribution, the approximation of the system reliability can be calculated as follows.

$$\tilde{R}(t; z, n) = \prod_{i \in A} \left(1 - \left(1 - e^{-\lambda_{i z_i} t} \sum_{l=0}^{k_{i z_i}-1} \frac{(\lambda_{i z_i} t)^l}{l!} \right)^{n_i} \right) \times \prod_{i \in CS} \left[e^{-\lambda_{i z_i} t} \left(\sum_{l=0}^{k_{i z_i}-1} \frac{(\lambda_{i z_i} t)^l}{l!} + \rho_i(t) \sum_{l=k_{i z_i}}^{k_{i z_i} n_i - 1} \frac{(\lambda_{i z_i} t)^l}{l!} \right) \right] \tag{8}$$

3 Solution algorithm: Continuous genetic algorithm (CGA)

There are several methods to optimize NP-hard problems such as dynamic programming [26], integer programming [27], and meta-heuristics like GA [20, 28–30], particle swarm optimization [31–33], the cuckoo search (CS) algorithm [34], the bee algorithm [35]. Thus, we employ an improved meta-heuristic to optimize the model.

GA has been developed as a robust and powerful stochastic search algorithm and works based on Darwin’s theory of evolution [36]. CGA or real-coded GA, which its chromosomes are considered as a floating point number, is a special forming of GAs so that acts better than GA based on a binary method for function optimization problems [37]. There are several superior features of CGA versus the binary GA such as working quickly because of the designed chromosomes, and storing computational less space to run the algorithm. Figure 2 shows the flowchart of the proposed algorithm, which is described as follows.

Fig. 2 The flowchart of the CGA algorithm

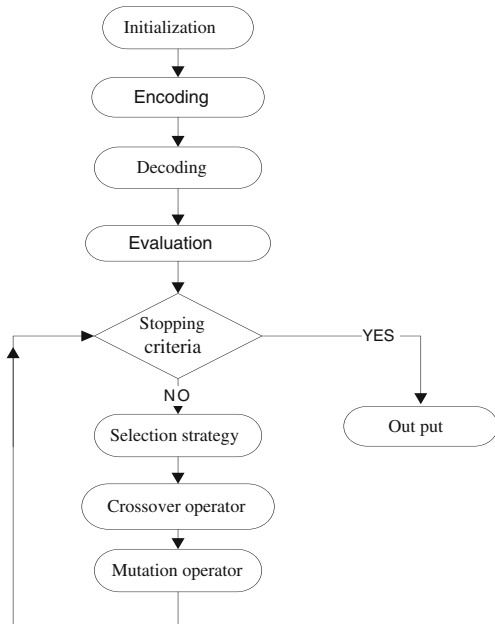


Fig. 3 The proposed chromosome

$$\begin{array}{l}
 1. \left[\begin{array}{cccccc} 0.2 & 0.6 & \dots & 0.89 & 0.65 \end{array} \right] \\
 2. \left[\begin{array}{cccccc} 0.45 & 0.11 & \dots & 0.45 & 0.18 \end{array} \right] \\
 3. \left[\begin{array}{cccccc} 0.52 & 0.25 & \dots & 0.31 & 0.73 \end{array} \right]
 \end{array}$$

3.1 Initialization

To begin the CGA, the parameters of the algorithm should be assigned an initial value. The parameters that have important roles in CGA are: (i) It : Number of iterations of CGA or the number of the generations; (ii) $Npop$: Number of the individual population; (iii) P_c : Crossover operator ratio; (iv) P_m : Mutation operator ratio. Where these parameters are tuned by RSM in Section 4.

3.2 Encoding structure

Encoding solution is the vital component to effective search in the solution area; therefore, we present a new encoding to CGA. Note that a possible solution consists of redundancy strategies, selected components, and the number of components for each subsystem. The proposed encoding presents a feasible solution that does not need to repair during the CGA process. There are three parts in the encoding of solution in which showed in Fig. 3.

Here, the rows, which are a random number between (0~1), are a $1 \times S$ vector so that S denotes number of the subsystem. Thus, these rows obtain a feasible solution after decoding process. The initial population is created randomly considering $Npop$.

3.3 Decoding structure

In order to evaluate the fitness function, it is necessary to decode the encoded solutions. Thus, the decoding process is elaborated considering an example with these explanations:

- S Number of subsystem ($S=6$)
- m_i Number of available component for a subsystem ($m_i=3$)
- $n_{\max,i}$ Constraint on the number of components used in each subsystem ($n_{\max,i}=6$)
- a_i Two redundancy strategies including active and cold standby for each subsystem ($a_i=2$)

The following steps describe decoding process.

- Step 1: Generate randomly a chromosome, (see Fig. 4).
- Step 2: Select the redundancy strategy for each subsystem between active or cold standby by rounding numbers obtained from the first row to nearest integer

Fig. 4 A graphical representation of the chromosome

$$\begin{array}{l}
 1. \left[\begin{array}{cccccc} 0.2 & 0.6 & 0.34 & 0.89 & 0.65 \end{array} \right] \\
 2. \left[\begin{array}{cccccc} 0.45 & 0.11 & 0.78 & 0.45 & 0.18 \end{array} \right] \\
 3. \left[\begin{array}{cccccc} 0.52 & 0.25 & 0.86 & 0.31 & 0.73 \end{array} \right]
 \end{array}$$

Fig. 5 Decoded the chromosome

$$\begin{array}{l}
 1. \\
 2. \\
 3.
 \end{array}
 \left[\begin{array}{ccccc}
 1 & 2 & 1 & 2 & 2 \\
 2 & 1 & 3 & 2 & 1 \\
 4 & 2 & 6 & 2 & 5
 \end{array} \right]$$

like this $\lfloor r_i \times a_i + 1 \rfloor$. Such, active and cold strategies are chosen by 1 and 2, respectively, (see the first row in Fig. 5).

Step 3: Calculate $\lfloor m_i \times c_i + 1 \rfloor$ to allocate available component for subsystem i so that 1, 2, and 3 are their number, (see the second row in Fig. 5).

Step 4: Obtain $\lfloor n_{\max,i} \times n_i + 1 \rfloor$ to select the number of the components used in subsystem i , (see the third row in Fig. 5).

3.4 CGA operators

In order to satisfy the constraints described in Eqs. (4) and (5), we employ one of the penalty methods, called the death penalty [36]. Moreover, individuals with higher qualification should have a greater chance of selection than those with lower fitness. Hence, we apply the roulette wheel selection to select the chromosomes. GA generates a new population with crossover and mutation operators. This paper utilizes a specific version of the continuous uniform crossover used by Radcliffe [38] as follows. Two chromosomes, which called parents, are selected between the populations. Next, a row similar to one of the parent is generated, namely β (with values between zero and one). Afterwards, offspring are created considering Eqs. (11) and (12). Figure 6 elaborates the process of the crossover operator for the 1th row of chromosomes.

$$\text{offspring}_1 = \beta \times \text{parent}_1 + (1-\beta)\text{parent}_2 \tag{9}$$

$$\text{offspring}_2 = \beta \times \text{parent}_2 + (1-\beta)\text{parent}_1 \tag{10}$$

The second operator in GA, which called mutation, prevents the algorithm to converge on a local optimal solution. The uniform mutation operator of Radcliff [38] is used in this paper. Figure 7 explains mutation operator; such, a chromosome is selected randomly, and then exchanged one of its cells with a random number between zero and one.

4 Parameter tuning: Response Surface Methodology

There are two main approaches to calibrate the parameters used in meta-heuristics: a trial-and-error procedure, and the statistical methods. The parameters of meta-heuristics

Fig. 6 An example of crossover operator

$$\left. \begin{array}{l}
 \text{Parent}_1, \quad 0.65 \quad 0.04 \quad 0.56 \quad 0.1 \\
 \text{Parent}_2, \quad 0.78 \quad 0.54 \quad 0.16 \quad 0.43 \\
 \beta \quad \quad \quad 0.3 \quad 0.11 \quad 0.76 \quad 0.11
 \end{array} \right\} \rightarrow \begin{array}{l}
 \text{Offspring}_1, \quad 0.42 \quad 0.48 \quad 0.25 \quad 0.13 \\
 \text{Offspring}_2, \quad 0.68 \quad 0.09 \quad 0.46 \quad 0.39
 \end{array}$$

Fig. 7 An example of mutation operator

	Parent	→	Offspring
1.	0.2 0.6 <u>0.34</u> 0.89 0.65		0.2 0.6 0.12 0.89 0.65
2.	0.45 0.11 0.78 <u>0.45</u> 0.18		0.45 0.11 0.78 0.33 0.18
3.	<u>0.52</u> 0.25 0.86 0.31 0.73		0.92 0.25 0.86 0.31 0.73

impress the quality of the solutions; hence, we employ one of the statistical methods in design of experiments, called the Response Surface Methodology (RSM). The RSM uses a collection of useful mathematical and statistical techniques to model and analysis of problems [39]. Considering a response that here is the system reliability, the RSM determines the best levels of variables in a problem. There are four variables in which are presented in Table 2 that shows their levels and coded factor.

In the beginning of RSM, a trial-and-error procedure is used to initialize the variables. Considering the obtained responses, a first-order model is fitted; and performed a test of lack of fit so that this test determines the sufficiency; otherwise, a second-order model is tested [40]. There are several the designs to fit a second-order model such as the central composite design (CCD) in which includes 2^k factorial points, n_c central points, and $2k$ axial points. The second-order model of CCD can be written as,

$$E(Y) = \beta_0 + \sum_{i=1}^k \beta_i X_i + \sum_{i=1}^k \beta_i X_i^2 + \sum_{i < j}^k \sum_{i < j}^k \beta_{ij} X_i X_j \quad (11)$$

Where $E(Y)$ is the expected value of the response variable in which is the system reliability, and β_0, β_i and β_{ij} are fixed coefficients of the second-order model. In addition, X_i and X_j are the input variables in k number.

4.1 Experiments

To perform RSM, we run 33 test problems in which have 14 subsystems with varying decreasingly weights W from 191 to 159, and the available budget restricted to 130 ($C=130$).

Note that the values of $-1, 0,$ and 1 are low, middle, and high levels of parameters, respectively. To remove the influence of results obtained from the test problems with different size, the mean ratio of the CGA solution (Y as the response) is defined as the performance index of the CGA. Table 3 shows the CCD that consists of eight factorial

Table 2 The parameters and their levels as the input variables of RSM

Parameters	Coded factor	Range	Levels		
			Low	Middle	High
N_{pop}	X_1	20–70	20	45	70
P_c	X_2	0.7–0.95	0.7	0.82	0.95
P_m	X_3	0.01–0.3	0.01	0.15	0.3
It	X_4	100–400	100	250	400

Table 3 The central composite design (CCD) with the responses

Runs	Input variable				Responses
	X_1	X_2	X_3	X_4	Y
1	0	0	0	0	0.94
2	0	0	0	-1	0.9
3	0	0	1	0	0.94
4	1	1	1	1	0.96
5	0	1	0	0	0.95
6	1	-1	-1	1	0.98
7	-1	1	-1	1	0.95
8	1	-1	1	-1	0.93
9	-1	1	-1	-1	0.85
10	-1	1	1	-1	0.78
11	-1	0	0	0	0.91
12	1	0	0	0	0.96
13	0	0	0	0	0.94
14	0	0	0	0	0.94
15	0	0	0	1	0.95
16	0	0	0	0	0.94
17	0	-1	0	0	0.95
18	0	0	-1	0	0.97
19	1	1	-1	-1	0.97
20	-1	-1	1	1	0.92

points, eight axial points, and four central points in twenty experiments. Considering the CCD, we run CGA to obtain the responses (Y). Since the proposed model belongs to the maximization models, a response (Y) with higher values presents better performances of the CGA. Regarding the analysis of variance (ANOVA) test for the response

Table 4 Analysis of variance for the accuracy performance index (Y)

Source	DF	Seq SS	Adj SS	Adj MS	F	P-value
Regression	11	0.040605	0.040605	0.003691	41.87	0
Linear	4	0.029778	0.029778	0.007444	84.44	0
Square	4	0.004183	0.004183	0.001046	11.86	0.002
Interaction	3	0.006644	0.006644	0.002215	25.12	0
Residual Error	8	0.000705	0.000705	0.000088		
Lack-of-Fit	5	0.000702	0.000702	0.00014	145.73	0.001
PureError	3	0.000003	0.000003	0.000001		
Total	19	0.04131				

S=0.009389; R-Sq=98.3 %; R-Sq(adj)=95.9 %

Table 5 The optimal value of input variables

Parameter	Value
$Npop$	56
P_c	0.89
P_m	0.18
It	321

in Table 4, there is a statistically significant curvature on the response surfaces; therefore, a second order model is employed to analysis experiments. The responses in Table 3 are used to estimate the second order models.

The best values of parameters are given by solving a single objective problem in Eq. (12), which is obtained by the estimated regression coefficients in the second-order model.

$$Y = 0.95 + 0.039X_1 - 0.0015X_2 - 0.021X_3 + 0.032X_4 - 0.0149X_1^2 + 0.0027X_2^2 + 0.0084X_3^2 - 0.0244X_4^2 + 0.007X_1X_2 + 0.007X_1X_3 - 0.0271X_1X_4 \quad (12)$$

Finally, the optimal values of parameters are illustrated in Table 5.

5 Numerical example

In this section, an example [5] is used to evaluate the performance of the proposed algorithm. There are 14 subsystems with three or four component selections; in

Table 6 Component data for example

i	Choice 1(j=1)				Choice 2(j=2)				Choice 3(j=3)				Choice 4(j=4)			
	λ_{ij}	k_{ij}	c_{ij}	w_{ij}	λ_{ij}	k_{ij}	c_{ij}	w_{ij}	λ_{ij}	k_{ij}	c_{ij}	w_{ij}	λ_{ij}	k_{ij}	c_{ij}	w_{ij}
1	0.00532	2	1	3	0.000726	1	1	4	0.00499	2	2	2	0.00818	3	2	5
2	0.00818	3	2	8	0.000619	1	1	10	0.00431	2	1	9	–	–	–	–
3	0.0133	3	2	7	0.0110	3	3	5	0.0124	3	1	6	0.00466	2	4	4
4	0.00741	2	3	5	0.0124	3	4	6	0.00683	2	5	4	–	–	–	–
5	0.00619	1	2	4	0.00413	2	2	3	0.00818	3	3	5	–	–	–	–
6	0.00436	3	3	5	0.00567	3	3	4	0.00268	2	2	5	0.000408	1	2	4
7	0.0105	3	4	7	0.00466	2	4	8	0.00394	2	5	9	–	–	–	–
8	0.0105	3	3	4	0.00105	1	5	7	0.0105	3	6	6	–	–	–	–
9	0.00268	2	2	8	0.000101	1	3	9	0.00408	1	4	7	0.000943	1	3	8
10	0.0141	3	4	6	0.00683	2	4	5	0.00105	1	5	6	–	–	–	–
11	0.00394	2	3	5	0.00355	2	4	6	0.00314	2	5	6	–	–	–	–
12	0.00236	1	2	4	0.00769	2	3	5	0.0133	3	4	6	0.0110	3	5	7
13	0.00215	2	2	5	0.00536	3	3	5	0.00665	3	2	6	–	–	–	–
14	0.0110	3	4	6	0.00834	1	4	7	0.00355	2	5	6	0.00436	3	6	9

Table 7 The result of comparison the proposed algorithm

<i>i</i>	Fyffe et al. [6]			Coit [5]			Coit [19]			Tavakkoli-Moghaddam et al. [20]			Proposed algorithm		
	<i>z_i</i>	<i>n_i</i>	R	<i>z_i</i>	<i>n_i</i>	R	<i>z_i</i>	<i>n_i</i>	R	<i>z_i</i>	<i>n_i</i>	R	<i>z_i</i>	<i>n_i</i>	R
1	3	3	A	3	3	C	3	4	A	1	2	C	3	4	A
2	1	2	A	1	2	C	1	2	C	1	2	A	1	2	C
3	4	3	A	4	3	C	4	3	A	4	3	A	4	3	A
4	3	3	A	3	3	C	3	3	C	3	3	C	3	3	C
5	2	3	A	2	3	C	2	3	A	2	2	A	2	3	A
6	2	2	A	2	2	C	2	2	C	2	2	A	2	2	C
7	1	2	A	1	2	C	1	2	C	1	2	C	1	2	C
8	1	4	A	3	2	C	3	2	C	1	3	C	1	3	C
9	3	2	A	2	2	C	1	2	C	1	2	A	1	2	C
10	2	3	A	2	3	C	2	3	C	1	2	C	2	3	C
11	1	2	A	3	2	C	3	2	C	1	4	C	3	2	C
12	1	4	A	4	2	C	4	2	C	1	3	C	4	2	C
13	2	2	A	2	2	C	2	2	A	3	2	C	2	2	A
14	3	2	A	3	2	C	3	2	C	3	2	A	3	2	C
Reliability	0.9700			0.9863			0.9875			0.9705			0.9872		
Weight	170			170			170			170			170		
Cost	119			123			123			104			120		

* R=type of redundancy strategy

addition, the maximum number of components is six ($n_{max,i}=6$). Moreover, it is considered 0.99 as the reliability of a switch at 100 h for each subsystem.

Thus, Table 6 illustrates the component cost, the weight and *k*-Erlang distribution parameters of the example. The aim of solving is to maximize system reliability

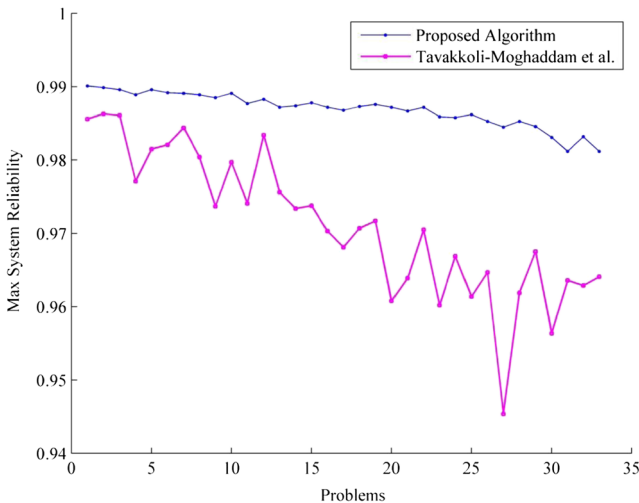


Fig. 8 Performance of CGA vs. Tavakkoli-Moghaddam et al. [20]

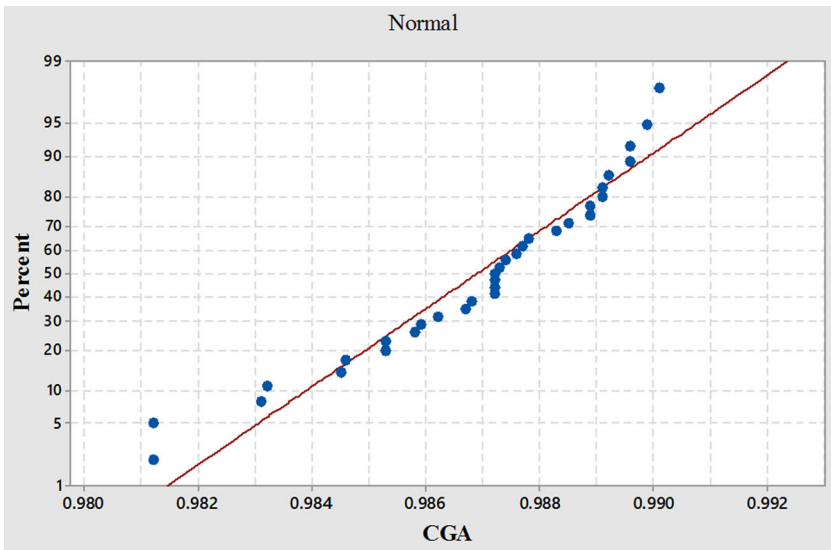


Fig. 9 The normal probability plot of the best costs provided by the proposed CGA

considering two constraints of the system cost ($C=130$) and the weight ($W=170$) in 100 h. There are two redundancy strategies for each subsystem: active or cold standby. The switch operates and fails similar to Scenario 1 in the cold standby redundancy strategy. Therefore, a continuous decreasing function would be suggested for switch reliability.

Table 7 makes a comparison between three solution approaches. The results explain that the proposed algorithm optimizes better the problem versus four studies. The

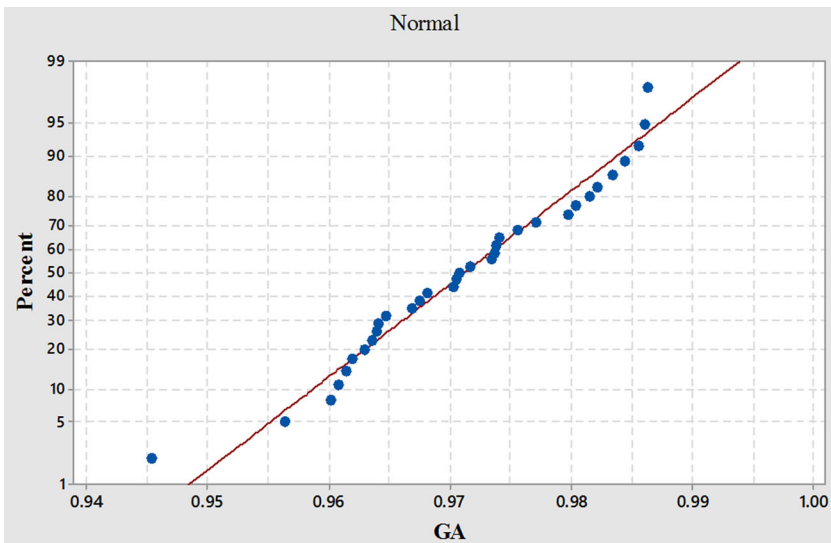


Fig. 10 The normal probability plot of the best costs provided by GA [20]

reliability (0.9872) presented by the proposed algorithm, for example, is better than the one presented by Coit [5].

In the second part of comparison, we resolve 33 problems used in Tavakkoli-Moghaddam et al. [20] considering varying the available weight from 191 to 159 and the constrained cost to 130. It can be understood from Table 8 that the proposed

Table 8 The result of comparison of the proposed algorithm with Tavakkoli-Moghaddam et al. [20]

Problem	W	C	Proposed algorithm			Tavakkoli- Moghaddam et al. [20]		
			Reliability	C	W	Reliability	C	W
1	191	130	0.9901	130	191	0.9856	123	189
2	190	130	0.9899	129	190	0.9863	128	190
3	189	130	0.9896	130	189	0.9861	120	189
4	188	130	0.9889	127	188	0.9771	112	187
5	187	130	0.9896	130	187	0.9815	115	187
6	186	130	0.9892	123	185	0.9821	116	186
7	185	130	0.9891	130	184	0.9844	120	184
8	184	130	0.9889	129	184	0.9804	105	184
9	183	130	0.9885	126	183	0.9737	112	183
10	182	130	0.9891	129	182	0.9797	102	178
11	181	130	0.9877	123	180	0.9741	106	180
12	180	130	0.9883	122	180	0.9834	113	179
13	179	130	0.9872	119	179	0.9756	100	176
14	178	130	0.9874	120	177	0.9734	107	178
15	177	130	0.9878	129	177	0.9738	114	177
16	176	130	0.9872	119	176	0.9703	102	176
17	175	130	0.9868	123	175	0.9681	105	174
18	174	130	0.9873	127	174	0.9707	109	174
19	173	130	0.9876	128	173	0.9717	102	172
20	172	130	0.9872	122	171	0.9608	107	172
21	171	130	0.9867	123	170	0.9639	98	171
22	170	130	0.9872	120	170	0.9705	104	170
23	169	130	0.9859	126	169	0.9602	102	168
24	168	130	0.9858	117	168	0.9669	92	166
25	167	130	0.9862	124	167	0.9614	91	167
26	166	130	0.9853	116	166	0.9647	100	165
27	165	130	0.9845	113	165	0.9454	97	164
28	164	130	0.9853	115	164	0.9619	96	164
29	163	130	0.9846	118	163	0.9675	104	163
30	162	130	0.9831	112	162	0.9564	93	162
31	161	130	0.9812	112	160	0.9636	95	161
32	160	130	0.9832	116	160	0.9629	93	159
33	159	130	0.9812	117	159	0.9641	95	159

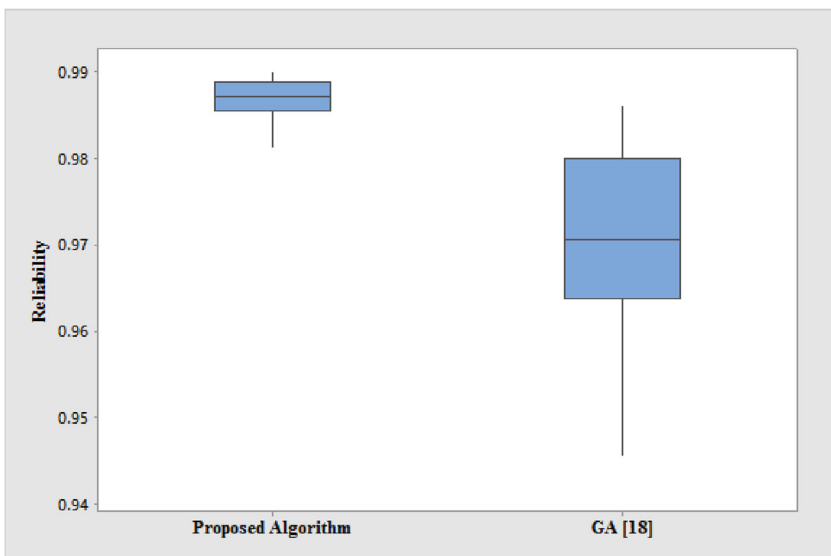
Table 9 T-Test for the best costs

	Number	Mean	StDev	SE Mean
The proposed CGA	33	0.98690	0.00235	0.00041
GA [20]	33	0.97116	0.00977	0.00170

T-Test of difference=0 (vs>): P-Value=0.000.

algorithm acts better than the solution algorithm of Tavakkoli-Moghaddam et al. [20] for all the problems. Figure 8 shows the domination of the proposed algorithm versus solution algorithm Tavakkoli-Moghaddam et al. [20].

Figures 9 and 10 shows normal probability plots of the best results obtained (Table 8) using the proposed CGA and the GA [20], respectively, to statistically compare the results obtained. Table 9 presents the result of a t-test for the alternative “The proposed CGA<The GA [20]” as the null hypothesis. As a significant difference is not shown (p -value=0.000<0.5), there is no presumption against the null hypothesis. Thus, the proposed CGA provides the better solution than the GA [20] on the average. Table 9 shows more details, in which SE Mean and StDev abbreviate the standard error of the mean and the standard deviation, respectively. Moreover, Fig. 11 presents the boxplot of comparison between the two methods, in which the reliability provided by the proposed algorithm is better than GA [20]. Note that the obtained solutions are the best among 20 runs of the algorithm. Notice also that a PC with 2.2 GHz Intel Core 2 Duo CPU, and 4 GB of RAM memory calculates all calculations in this paper; moreover, CGA is run in MATLAB 2009a.

**Fig. 11** The Boxplot of Proposed Algorithm and GA [18]

6 Conclusion

This paper has considered the optimization of a series–parallel reliability redundancy allocation problem (RRAP) in which subsystems can select active or cold standby strategies. The aim of this problem was to choose the best redundancy strategies, components, and levels for each subsystem for maximizing the system reliability. Since the RRAP is an NP-hard problem, we employed a continuous genetic algorithm (CGA), which applied a new method in the production of chromosomes, to solve the model. In addition, the performance of CGA was improved by parameter tuning that one of statistical methods namely the response surface methodology (RSM) was used. According to the comparison of the proposed algorithm with the last solution algorithms, it can be concluded that the CGA has an efficient performance to optimize the RRAP problems. In other words, the presented method can probably be the best available algorithms in the literature to solve the considered problem. Finally, the future works of this study are as follows. (1) Investigate fuzzy condition in modeling. (2) Develop the objectives to consider the cost. (3) Study other redundancy strategies as decision variables.

References

1. Yalaoui, A., Chu, C., Châtelet, E.: Reliability allocation problem in a series–parallel system. *Reliab. Eng. Syst. Saf.* **90**, 55–61 (2005)
2. Yalaoui, A., Chatelet, E., Chengbin, C.: A new dynamic programming method for reliability & redundancy allocation in a parallel–series system. *IEEE Trans. Reliab.* **54**, 254–261 (2005)
3. Yeh, W.-C.: Orthogonal simplified swarm optimization for the series–parallel redundancy allocation problem with a mix of components. *Knowl.-Based Syst.* **64**, 1–12 (2014)
4. Huang, C.-L.: A particle-based simplified swarm optimization algorithm for reliability redundancy allocation problems. *Reliab. Eng. Syst. Saf.* **142**, 221–230 (2015)
5. Coit, D.W.: Cold-standby redundancy optimization for nonrepairable systems. *IIE Trans.* **33**, 471–478 (2001)
6. Fyffe, D.E., Hines, W.W., Lee, N.K.: System reliability allocation and a computational algorithm. *IEEE Trans. Reliab.* **17**, 64–69 (1968)
7. Ebeling, C. E. :An introduction to reliability and maintainability engineering: McGraw Hill (1997)
8. Franko, C., Ozkut, M., Kan, C.: Reliability of coherent systems with a single cold standby component. *J. Comput. Appl. Math.* **281**, 230–238 (2015)
9. Grabski, F.: 6 - SM models of renewable cold standby system. In: Grabski, F., (ed.). *Semi-Markov Processes: Applications in System Reliability and Maintenance*, pp. 99–118. Elsevier, (2015)
10. Levitin, G., Xing, L., Dai, Y.: Optimal component loading in 1-out-of-N cold standby systems. *Reliab. Eng. Syst. Saf.* **127**, 58–64 (2014)
11. Levitin, G., Xing, L., Dai, Y.: Cold vs. hot standby mission operation cost minimization for 1-out-of-N systems. *Eur. J. Oper. Res.* **234**, 155–162 (2014)
12. Mukherjee, A., Dhar, A.S.: Real-time fault-tolerance with hot-standby topology for conditional sum adder. *Microelectron. Reliab.* **55**, 704–712 (2015)
13. Huang, W., Loman, J., Song, T.: A reliability model of a warm standby configuration with two identical sets of units. *Reliab. Eng. Syst. Saf.* **133**, 237–245 (2015)
14. Wells, C.E.: Reliability analysis of a single warm-standby system subject to repairable and nonrepairable failures. *Eur. J. Oper. Res.* **235**, 180–186 (2014)
15. Levitin, G., Xing, L., Dai, Y.: Optimal sequencing of warm standby elements. *Comput. Ind. Eng.* **65**, 570–576 (2013)

16. Kuo, W., Prasad, V.R.: An annotated overview of system-reliability optimization. *IEEE Trans. Reliab.* **49**, 176–187 (2000)
17. Tillman, F.A., Hwang, C.-L., Kuo, W.: Optimization techniques for system reliability with redundancy: a review. *IEEE Trans. Reliab.* **26**, 148–155 (1977)
18. Coit, D.W., Liu, J.C.: System reliability optimization with k-out-of-n subsystems. *Int. J. Reliab. Qual. Saf. Eng.* **07**, 129–142 (2000)
19. Coit, D.W.: Maximization of system reliability with a choice of redundancy strategies. *IEE Trans.* **35**, 535–543 (2003)
20. 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.* **93**, 550–556 (2008)
21. Tian, Z., Levitin, G., Zuo, M.J.: A joint reliability–redundancy optimization approach for multi-state series–parallel systems. *Reliab. Eng. Syst. Saf.* **94**, 1568–1576 (2009)
22. Sharma, V.K., Agarwal, M., Sen, K.: Reliability evaluation and optimal design in heterogeneous multi-state series–parallel systems. *Inf. Sci.* **181**, 362–378 (2011)
23. Pasandideh, S.H.R., Niaki, S.T.A., Nia, A.R.: A genetic algorithm for vendor managed inventory control system of multi-product multi-constraint economic order quantity model. *Expert. Syst. Appl.* **38**, 2708–2716 (2011)
24. Sadeghi, J., Sadeghi, A., Saidi-Mehrabad, M.: A parameter-tuned genetic algorithm for vendor managed inventory model for a case single-vendor single-retailer with multi-product and multi-constraint. *J Optim. Ind. Eng.* **4**, 57–67 (2011)
25. Nachiappan, S.P., Jawahar, N.: A genetic algorithm for optimal operating parameters of VMI system in a two-echelon supply chain. *Eur. J. Oper. Res.* **182**, 1433–1452 (2007)
26. Ng, K.Y.K., Sancho, N.G.F.: A hybrid ‘dynamic programming/depth-first search’ algorithm, with an application to redundancy allocation. *IEE Trans.* **33**, 1047–1058 (2001)
27. Gen, M., Ida, K., Lee, J.-U.: A computational algorithm for solving 0–1 goal programming with GUB structures and its application for optimization problems in system reliability, *Electron. Commun. Jpn. (Part III: Fundam Electron Sci)*, **73**, 88–96, (1990).
28. Sahoo, L., Bhunia, A.K., Kapur, P.K.: Genetic algorithm based multi-objective reliability optimization in interval environment. *Comput. Ind. Eng.* **62**, 152–160 (2012)
29. Sadeghi, J., Sadeghi, S., Niaki, S.T.A.: A hybrid vendor managed inventory and redundancy allocation optimization problem in supply chain management: an NSGA-II with tuned parameters. *Comput. Oper. Res.* **41**, 53–64 (2014)
30. Paul, R.J., Chanev, T.S.: Optimising a complex discrete event simulation model using a genetic algorithm. *Neural Comput. & Applic.* **6**, 229–237 (1997)
31. Pereira, C.M.N.A., Lapa, C.M.F., Mol, A.C.A., da Luz, A.F.: A Particle Swarm Optimization (PSO) approach for non-periodic preventive maintenance scheduling programming. *Prog. Nucl. Energy* **52**, 710–714 (2010)
32. Pai, P.-F., Chen, C.-T., Hung, Y.-M., Hung, W.-Z., Chang, Y.-C.: A group decision classifier with particle swarm optimization and decision tree for analyzing achievements in mathematics and science, *Neural Comput. Applic.*, 1–13, (2014)
33. Jiang, S., Yi, D., Ju, X., Wang, L. Liu, Y.: An approach for test data generation using program slicing and particle swarm optimization. *Neural Comput. Applic.* 1–9 (2014)
34. Ouaraab, A., Ahiod, B., Yang, X.-S.: Discrete cuckoo search algorithm for the travelling salesman problem. *Neural Comput. & Applic.* **24**, 1659–1669 (2014)
35. Tsai, H.-C.: Integrating artificial bee colony and bees algorithm for solving numerical function optimization. *Neural Comput. & Applic.* **25**, 635–651 (2014)
36. Gen, M., Cheng, R.: *Genetic algorithms and engineering design*, 1st edn. Wiley, New York (2000)
37. Randy, S.E.H., Haupt, L.: *Practical genetic algorithms*. Wiley & Sons, New York (2004)
38. Reeves, C.R.: *Genetic algorithms and grouping problems*. *IEEE J. Magazines* **5**, 297–298 (2001)
39. Raymond, D.C.M., Myers, H., Anderson-Cook, C.M.: *Response surface methodology: process and product optimization using designed experiments*. John Wiley & Sons, New York (2009)
40. Montgomery, D.C.: *Design and analysis of experiments*, 5th edn. John Wiley and Sons, New York (2001)