Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

Efficient optimization of multi-objective redundancy allocation problems in series-parallel systems

Mina Ebrahimi Arjestan[a*]

[a] *Department of Industrial Engineering, Kharazmi University, Tehran, Iran*

CHRONICLE

Article history:
Received October 3, 2016
Received in revised format: October 22, 2016
Accepted November 25, 2016
Available online November 28 2016

Keywords:
Reliability
Multi-objective optimization
Systems of series-parallel genetic algorithm
Metaheuristic

ABSTRACT

Reliability issues are most important types of optimization problems and they are used in communication, transportation and electrical systems. This paper presents two mathematical models to solve the k-out-of-n redundancy problem where there are two objectives: maximization of reliability and minimization of cost subject to two constraints. Constraints are associated with weight and volume. In addition, strategy of redundancy is intended and ready to go cold and the components of the systems are also identical, because the model is to solve the complex models of the genetic algorithm (GA) and simulated annealing (SA). The proposed study uses NSGAII and MOPSO to solve the proposed studies and compare them using TOPSIS method.

Nomenclature

i	subsystems symbol (i = 1,2,3, ..., s)	j	component of subsystems (j = 1,2,3, ...,)n_i
R (t)	The reliability of the system at time t	n_i	The number of components used in the system i
k_i	The minimum number of members which are working for the system i	z_i	The index is allocated to the system i
t	The time of operation	V	maximum volume
W	The upper limit of weight	v_i	The volume of component j which is allocated to the system i
w_i	The weight of component j which is allocated to the system i	λ_{ij}	The failure rate of component j which is allocated to the system i
S	number of sub-systems (First model)	S	Sets the system which uses cold standby redundancy policy (Second model)
A	collection of subsystems that are active redundancy policy	C	Sets the system that they use cold standby redundancy policy
c_{ij}	The cost of component j which is allocated to the system i	V	maximum volume
W	The upper limit of weight	θ	Life parameter
β	Shape parameter		

[*] Corresponding author.
E-mail address: ebrahimiarjestan@gmail.com (M. Ebrahimi Arjestan)

doi: 10.5267/j.dsl.2016.11.004

1. Introduction

One of the important issues that researchers have to deal with is associated with the optimization of the reliability of the systems. There is an increasing attention on increasing the quality of products through design improvement capabilities. One of the goals of reliability is associated with designing highly reliable systems design. Different models can be used to optimize system reliability. In all existing models, redundancy strategy can improve the reliability of the system. Redundancy allocation problem was first presented by Fyffe et al. (1968). On the issue of series-parallel system architecture subsystems K-out-of-n intends the possibility of allocating identical components to each subsystem and systems for active and standby redundancy. Fyffe et al. (1968) presented the mathematical model of redundancy allocation problem. Their proposed model aimed to maximize the reliability of the system under the weight and fee restrictions. In the proposed series-parallel structure, they considered an active redundancy policy and the possibility of allocating identical units to each subsystem components. They solved this problem by using dynamic programming.

Misra and Sharma (1973) considered the question of redundancy allocation for the series-parallel subsystems k-out-of-n. In their model, active redundancy policy and the possibility of allocating to each sub-system components were assumed to be identical. They benefited of zero-one programming to solve the resulted problem. Chern (1992) proved that redundancy allocation problem in terms of computational time is NP-Hard and proposed appropriate innovative and meta-heuristic methods to solve the problem. Many innovative approaches in the past four decades have been provided for redundancy allocation problem. Redundancy allocation methods to solve public issues were provided by Sharma and Vnkastvran (1971); Aggarwal et al. (1975); Gopal et al. (1978); Misar, (1972); Nakagawa and Nakashyma (1981). Interested readers are referered to Soltani (2014) for a comprehensive review.

In this paper, the issue is intended to form a bi-objective under the policy of k-out-of-n and the structure of the series-parallel is used. Each of the following parts has been formed under different distribution systems, e.g. Erlang or Weibull distribution. The goal here is to maximize reliability and to minimize the cost by considering some constraints.

2. Literature review

Fyffe et al. (1968) were first who presented the mathematical model of redundancy allocation problem. The proposed model aims to maximize the reliability of the system under the weight and cost restrictions. In the proposed series-parallel structure, they considered an active redundancy policy and the possibility of allocating identical unit o each subsystem components. They solved this problem by using dynamic programming. Nakagawa and Miyazaki (1981) solved 33 problems by replacing an exact method and explained that, it would be a better idea to use alternative methods instead of dynamic programming. Bulfin and Liu (1985) discussed allocating redundant components subject to resource constraints to optimize some measure of system performance. Chen and You (2005) considered the series–parallel redundant reliability problems where both the multiple component choices of each subsystem and the redundancy levels of every selected component were taken into account to maximize the system reliability. Coit and Liu (2000) proposed a method to select the components and redundancy levels to maximize system reliability given system-level constraints. Liang and Smith (2004) proposed an ant colony optimization algorithm for the redundancy allocation problem. Nahas et al. (2007) used coupling ant colony and the degraded ceiling algorithm for the redundancy allocation problem of series – parallel systems. Hsieh (2002) used a simple method to solve linear programming redundancy allocation problems. In the first phase of this method, the proposed problem was solved by estimating the objective function and some linear decision variables. This works in the second phase by using the issue of Knapsack problem to improve the results. The allocation for the sub-system redundancy with identical structure was a non-convex nonlinear programming model and Ha and Kuo (2006) presented a branch and bound to solve the problem. Lee et al. (2003) compared the computational complexity of the solutions produced by two methods of Nakagava and Nakashima (1977) with each other. Kulturel-Konak et al. (2006) proposed a tabu search approach to solve multi-objective combinatorial optimization problems. This procedure chooses an objective to become active for a given iteration with a multinomial probability mass function. Kim et al. (2004) used genetic algorithms simulation for solving redundancy allocation problems. Onishi et al. (2007) developed an exact solution to solve a series-parallel problem. As mentioned before, Nakagawa and Miyazaki (1981) solved 33 problems by replacing an exact method but their method could not find optimal solutions for three instances. Onishi et al. (2007) developed an alternative method and could manage to solve all 33 problems, successfully.

Tavakkoli-Moghaddam et al. (2008) used a genetic algorithm to solve the redundancy problem. Chambari et al. (2012) studied a specific type of redundancy allocation problem (RAP) and developed a bi-objective RAP (BORAP). The model includes non-repairable series–parallel systems in which the redundancy strategy was taken into account as a decision variable for individual subsystems. There were two objective functions where the first one was to maximize system reliability and the second one minimized the system cost and there were also some constrains. They proposed non-dominated sorting genetic algorithms (NSGA-II) and multi-objective particle swarm optimization (MOPSO) to solve the problem. The proposed problem of this paper is similar with the model originally developed by Chambari et al. (2012), except that we use different techniques to compare the results.

3. The problem statement

3.1. The proposed model and its assumptions principles

The proposed study of this paper considers the following assumptions:

- Active and standby redundancy are considered cold.
- New and used parts are taken into account.
- Two strategies are intended with size and weight limitations.
- Component failure are independent of each other.
- Components are placed on preventative maintenance.
- Component failure follows Erlang distribution.

General mathematical model (the first model)

$$\max R_i(t) \tag{1}$$

$$\min \; C_i(t) \tag{2}$$

subject to

$$\sum_i w_{i,z_i} n_i \;\; \leq w \tag{3}$$

$$\sum_i v_{i,z_i} n_i \;\; \leq \; v \tag{4}$$

$$n_i \,\epsilon \{1,2,3 \dots \dots \dots \dots, n_{max}\} \; i = 1,2, \dots ., s \tag{5}$$

$$z_i \in \{1,2,3, \dots \dots \dots \dots, m_i\} \; i = 1,2, \dots . ., s \tag{6}$$

As stated in the introduction section, there are two objective functions similar to Chambari et al. (2012) where the first one is associated with reliability and the second one is related to the cost. There are also two types of constrains including weight and volume. We consider 10 sub-systems to calculate the reliability of the system, as follows.

The reliability of the systems that follow the active politics

The number of working parts follows the binomial distribution with $\mathrm{n_i}$ and $\mathrm{R_a}$ parameters in $\mathrm{k-out-of-n\ sub-system}$ with independent and the same parts. $\mathrm{R_a}$ is the reliability of each component in the system *i* and $\mathrm{n_i}$ is the number of parts in the system *i*. So we have:

$$p(\,Exactly\ i\ component\ work) = \ (1-R_a)^{n_i-l}\ R_a^l\binom{n_i}{l} \tag{7}$$

The reliability of sub-system $\mathrm{k-out-of-n}$ is equal to the probability that the number of working parts is greater than or equal *k*. Therefore, the reliability of sub-system $\mathrm{k-out-of-n}$ is equal to Eq. (8) as follows,

$$R(K_i,n_i) = \sum_{l=k_i}^{n_i}\binom{n_i}{l}R_a^l(1-R_a)^{n_i-l}\,, \tag{8}$$

Multiplying the values of the sub-systems that use active policy is as follows,

$$\pi_{i\epsilon A\epsilon}(1\text{-}(1-\exp\left(-T *\ \lambda_{ij}\right).\frac{\sum_{l=0}^{k-1}(T*\lambda_{ij})^l}{l!})^{n_{ij}}\ ) \tag{9}$$

Multiplying the values of the sub-systems that use cool ready-to-work policy, problems in this system is $\mathrm{k_j}$ - 1 - $\mathrm{NK_j}$.

$$\pi_{i\epsilon S}\left(r_{ij}+\rho_i*\exp\left(-T*\lambda_{ij}\right).\sum_{l=k_{ij}}^{NK_{ij-1}}\frac{\left(T*\lambda_{ij}\right)^l}{l!}\right) \tag{10}$$

The reliability of the sub-system without redundancy policy

Multiplying the values of the sub-systems that do not use any police is as follows,

$$\pi_{i\epsilon N}\exp\left(-T\times\lambda_{ij}\right).\sum_{l=0}^{k_j-1}\frac{\left(*\,\lambda_{ij}\right)^l}{l!} \tag{11}$$

If all Eqs. (8-11) are multiplied together and the integral in the period (0 - T) is calculated, the average of moment reliability will be equal to the objective function reliability as follows,

$$R(t;z,n) = (1/n)*\int_0^T\pi_{i\epsilon A}\,(1\quad-\left(1-\exp\left(-T*\lambda_{ij}\right)\quad.\sum_{l=0}^{k-1}\left(T*\lambda_{ij}\right)^l/l!))\right)*\pi_{i\epsilon S}(r_{ij}+\rho_i*\exp\left(-T*\lambda_{ij}\right).\ \sum_{l=k_{ij}}^{NK_{ij-1}}\frac{\left(T*\lambda_{ij}\right)^l}{l!}\Bigg) \tag{12}$$

$$*\,\pi_{i\epsilon N}\exp\left(-T*\lambda_{ij}\right).\sum_{l=0}^{k_j-1}\frac{\left(T*\lambda_{ij}\right)^l}{l!}.$$

$$C = \ \textstyle\sum_{i=1}^{s}C_i\ (t)\ . \tag{13}$$

3.1.1. Solution

In this paper, both genetic algorithm and simulated annealing methods are used to solve the model and the results of both methods will be compared.

Genetic algorithm

In this section, we explain the proposed genetic algorithm to solve the problem. Population-based genetic algorithm is an algorithm to solve a variety of optimization problems. The problem is to study an optimization problem is non-binding. A chromosome is defined according to the statement of the problem carried out in the previous section as follows,

Type of Strategy	1	2	2	1	2	1	1
Type of the surplus	2	2	1	3	2	3	1
The number of surplus	3	2	4	3	2	3	2

Fig. 1. Chromosomes

After generating a new chromosome, the chromosome must be evaluated, the total cost of the response must be specified. This is implemented by simply using the formula mentioned for the objective function to be performed in the third quarter. Binary tournament is proposed in selection operator in genetic algorithm. In addition to maintain the best answer obtained, so far, 10 percent of the best solutions in the current population are passed directly to the population. In this study, a three-strand structure is used to display chromosomes. For all three strands of the chromosome, after selecting two parents using the method mentioned in the previous section, for each subsystem, the corresponding number in the house is chosen randomly from one parent. Fig 2 shows an example of this type of intersection which contains 4 subsystems. Random string determines the child to inherit their genes from each parent. For example, to determine what sort of additional components is assigned to the first subsystem, referred to a random string, the corresponding number in the string is equal to one, therefore, the child, the type of subsystem components is used in its first surplus inherited from the first parent. Also, to determine what sort of additional components are assigned to the second subsystem, referred to a random string, the corresponding number in the string is equal to two, therefore, the child, and the type of subsystem components are used in its first surplus inherited from the first parent. By following this process, the child is shown in the figure is obtained.

The first parrent	1	3	1	2
The second parrent	2	2	2	1
Random string	1	2	1	1
Child	1	2	1	2

Fig. 2. The process of intersection

This method is used for the production of disciplines related to allocation strategies surplus and a surplus component. After the chromosomes crossover operator, mutation operator is applied on them. Mutation prevents the trapping algorithm from the optimal position. Operator mutation considered operator as a basis for maintaining population distribution. For the algorithm presented in this study, the maximum number of 100 iterations of the algorithm is intended as a termination condition. Two-parent crossover operator as input characteristics are combined to produce a child. Consequently, the children produce their properties inherited from their parents.

Genetic algorithm parameters

Here Pm, Pe and Psize represent mutation probability, the possibility of elitism and the number of population, respectively. In this section, operators and the appropriate parameters are selected using experimental design. These operations and parameters are given in Table 1 in the form of a number of factors.

Table 1
Factor and their levels

Factor	Levels	Number
The first population size for genetic algorithm (Pop_size)	{100,150,200,250,300}	5
Percent of elitism (elite rate)	{0.05,0.1,0.15,0.2,0.25}	5
The probability of mutation (mut_rate)	{0.05,0.1,0.15,0.2 }	4

Due to the large number of parameters in the proposed algorithm, finding the right combinations of parameters that improves the algorithm performance is of great importance. Fractional factorial designs are used to tune the parameters (Taguchi, 1986). A family of matrices developed that reduces the number of tests. In Taguchi method, orthogonal arrays are used to study a large number of decision variables by a small number of experiments. Degrees of freedom for factors considered in this study is equal to 11, so Taguchi design selected should be at least 11 rows and 3 columns. Taguchi test is available from standard designs selected design L25 (53) as an appropriate plan. But this plan needs to be improved. In this study, there is one factor that has 4 levels. For a perfect match with the chosen design, this factor must be level 5 and to remedy this gap for each factor deficient levels. Table 2 shows 25 tests for modified design L25 (53).

Table 2
Adjusted design L25 (53)

pop size	elit_rate	mut_rate
100	0.05	0.05
100	0.1	0.1
100	0.15	0.15
100	0.2	0.2
100	0.25	0.2
150	0.05	0.1
150	0.1	0.15
150	0.15	0.2
150	0.2	0.2
150	0.25	0.05
200	0.05	0.15
200	0.1	0.2
200	0.15	0.2
200	0.2	0.05
200	0.25	0.1
250	0.05	0.2
250	0.1	0.2
250	0.15	0.05
250	0.2	0.1
250	0.25	0.15
300	0.05	0.2
300	0.1	0.05
300	0.15	0.1
300	0.2	0.15
300	0.25	0.2

After obtaining the results, Taguchi, S / N ratios are calculated using Minitab software. The results are shown in Table 3, as it is clear from this table, the most important factor in the performance of genetic algorithm is proposed for size of the population.

Table 3
The results

Source	DF	Seq SS	Adj SS	Adj MS	F	P
pop size	4	0.000873	0.000873	0.000218	9.48	0.001
elit_rate	4	0.000166	0.000166	0.000041	1.8	0.189
mut_rate	3	0.000185	0.000185	0.000062	2.67	0.091
Residual Error	13	0.000299	0.000299	0.000023		
Total	24	0.001523				

Fig. 3 determines the optimum level for each factor used for the major effect diagram.

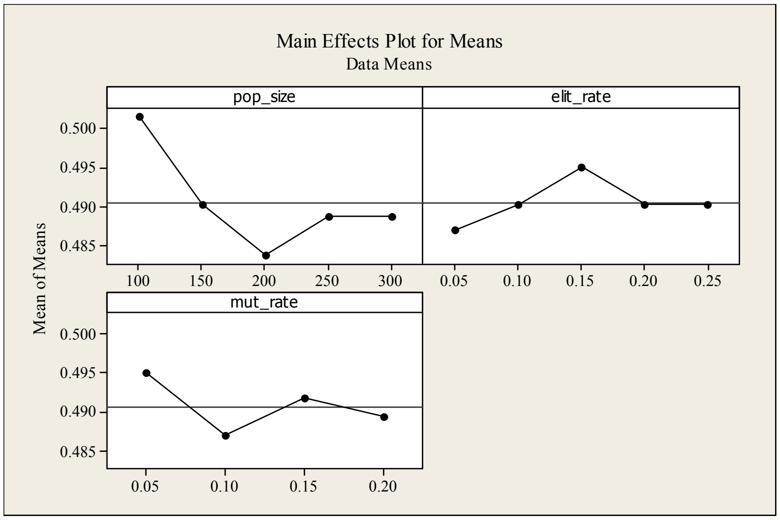

Fig. 3. The major effects of unpaired Reply

Based on the response variable defined, the values of these variables are less and the algorithm had a better performance. Therefore, the optimal values for each level will be determined based on Table 4.

Table 4
Appropriate levels

Factor	The optimum level
The first population size for genetic algorithm (Pop size)	{200}
Percent of elitism (elite rate)	{0.05}
The probability of mutation (mut_rate)	{0.1 }

Simulated Annealing (SA)

The concept of simulated annealing (SA) was first proposed by Kirkpatrick et al. (1983). Simulated annealing algorithm proposes structure similar to what was used in the suggested genetic algorithm. At each step, the SA heuristic considers some neighboring state s' of the current state s, and probabilistically makes a decision between moving the system to state s' or remaining in state s. These probabilities eventually lead the system to lead to states of lower energy and this step is repeated until the system approaches to a state that is good enough for the application, or until a predetermined computation budget is reached. For the algorithm presented in this study, the maximum number of 200 iterations of the algorithm is intended as a termination condition. In this section we determine parameters and operators of simulated annealing algorithm. The number of operators and parameters are given in the Table 5.

Table 5
Factor and their levels

Factor	Levels	Number
Inception temperature (int temp)	{100,150,350}	3
Cooling coefficient (q)	{0.7,0.8,0.9}	3
The number of iteration at each temperature (it)	{50,100,150}	3
Number of iterations to increase diversity strategy (dn)	{20, 50,100 }	3

Due to a large number of parameters in the proposed algorithm, we find the right combinations of parameters that improves the performance of the algorithm performance. Here, 4 factors determine the characteristics and parameters of the proposed algorithm. These factors include: Start temperature, coefficients cooling, temperature and number of repetitions for strategy to increase the number of repetitions in the each variety. The total number of possible combinations of these factors is equal to 81 and this number can be reduced to 27 by using the Taguchi method. The algorithm uses Taguchi method to reduce the number of performances which, as mentioned above, can be reduced to a quarter of the required performances. This is desirable to reduce the number of repetitions. Degrees of freedom for factors considered in this study is equal to 7, so Taguchi design selected should be at least 9 rows and 5 columns. Taguchi tests available from the standard designs are selected design L27 (34) as appropriate plan. Table 6 shows 27 tests for the modified design L27 (34).

Table 6
Adjusted plan L27

Inti_temp	q	Iter	dn	Inti_temp	q	Iter	dn
1	1	1	1	2	2	3	1
1	1	1	1	2	3	1	2
1	1	1	1	2	3	1	2
1	2	2	2	2	3	1	2
1	2	2	2	3	1	3	2
1	2	2	2	3	1	3	2
1	3	3	3	3	1	3	2
1	3	3	3	3	2	1	3
1	3	3	3	3	2	1	3
2	1	2	3	3	2	1	3
2	1	2	3	3	3	2	1
2	1	2	3	3	3	2	1
2	2	3	1	3	3	2	1
2	2	3	1				

After obtaining the results of Taguchi, S / N ratio is calculated using Minitab software. The results are shown in Table 7, as it is clear from this table, the most important factor in the performance of the algorithm determines the size of the population.

Table 7
The results

Source	DF	Seq SS	Adj SS	Adj MS	F	P
init_tmp	2	19.2646	19.2646	9.6323	8.98	0.23
q	2	14.1486	14.1486	7.0743	6.6	0.265
iter	2	13.8338	13.8338	6.9169	6.45	0.268
dn	1	0.5031	0.5031	0.5031	0.47	0.618
Residual Error	1	1.0725	1.0725	1.0725		
Total	8	48.8226				

Fig. 4 determines the optimum level for each factor used the major effect diagram.

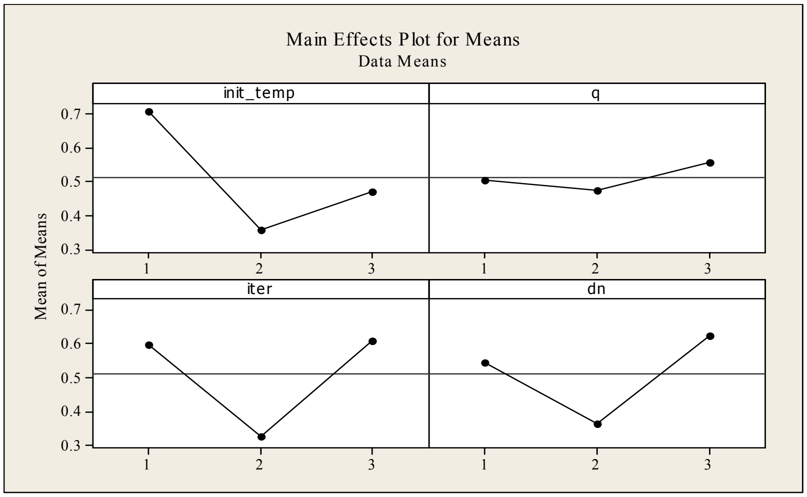

Fig. 4. The major effects of unpaired Reply

Based on the response variable defined, the values of these variables are less and the algorithm had a better performance. Considering this, the optimal values for each level will be determined according to Table 4 as follows.

Table 8
Appropriate levels

Factor	The optimum level
Inception temperature (int temp)	{150}
Cooling coefficient (q)	{0.8}
The number of iteration at each temperature (it)	{100}
The number of iteration for the strategy (dn)	{50}

3.2. Second mathematical model

The second model considers the following assumptions:

- ✓ The policies of active redundancy, cold standby and no redundancy policy for each subsystem are used,
- ✓ The possibility of allocating new components is used, simultaneously,
- ✓ Size and weight restrictions are intended,
- ✓ Component failure are independent of each other,
- ✓ The components are not preventive maintenance,
- ✓ Component failure follows a Weibull distribution,
- ✓ The following systems are under k- out - of –n.

$$max\, R_i\,(t) \tag{14}$$

$$min\;\; C_i(\mathrm{t}) \tag{15}$$

subject to

$$\sum w_{i,z_i} n_i \;\; \leq \mathrm{w} \tag{16}$$

$$\sum \mathrm{v}_{\mathrm{i},\mathrm{z}_\mathrm{i}} \mathrm{n}_\mathrm{i} \;\; \leq \; v \tag{17}$$

$$n_\mathrm{i} \in \{1,2,3 \dots\dots\dots\dots, n_{max}\} \;\; i = 1,2, \dots .\,, s \tag{18}$$

$$z_\mathrm{i} \in \{1,2,3, \dots\dots\dots\dots, m_i\} \;\; i = 1,2, \dots .\,.\,, s \tag{19}$$

We consider 10 sub-system to calculate the reliability of the system, as follows

The reliability of the systems that follow the active politics

The number of working parts follows the binomial distribution with $\mathrm{n_i}$ and $\mathrm{R_a}$ parameters in $\mathrm{k - out - of - n\ sub - system}$ with independent and the same parts. $\mathrm{R_a}$ is the reliability of each component in the system *i* and $\mathrm{n_i}$ is the number of parts in the system *i*. So we have:

$$p(exactly, works\; l\; component\,) = \; (1 - R_a)^{n_{i-l}}\; R_a^l \binom{n_i}{l} \tag{20}$$

The reliability of sub-system $\mathrm{k-out-of-n}$ is equal to the probability that the number of working parts is greater than or equal *k*. Therefore, the reliability of sub-system $\mathrm{k-out-of-n}$ is equal to Eq. (21) as follows,

$$R(K_i, n_i) = \sum_{l=k_i}^{n_i} \binom{n_i}{l} R_a^l (1-R_a)^{n_i-l} \, . \tag{21}$$

Multiplying the values of the sub-systems that use active policy is as follows,

$$\pi_{i\epsilon A\epsilon}(1\text{-}(1-\exp(-T*\lambda_{ij}).\frac{\sum_{l=0}^{k-1}(T*\lambda_{ij})^l}{l!})^{n_{ij}}\ ) \, . \tag{22}$$

Multiplying the values of the sub-systems that use cool ready-to-work policy is $\mathrm{k_j}$ - 1 - $\mathrm{NK_j}$. This indicates the number of parts assigned to each sub-system.

$$\pi_{i\epsilon S}\left(r_{ij} + \rho_i * \exp(-T*\lambda_{ij}).\sum_{\mathrm{l=k_{ij}}}^{\mathrm{NK_{ij-1}}} \frac{(\mathrm{T}*\lambda_{\mathrm{ij}})^{\mathrm{l}}}{\mathrm{l!}}\right). \tag{23}$$

Therefore,

$$C = \sum_{i=1}^{s} C_i\ (t), \tag{24}$$

$$R_i(t) = \sum_{l=k_i}^{n_i} \binom{n_i}{l} \left(e^{-\frac{t}{\theta ij}}\right)^{\beta_{ij}} \left(1-e^{-(\frac{t}{\theta ij})^{\beta_{ij}}}\right)^{n_{i-l}} \times \pi_{i\epsilon S}\left(e^{-\frac{t}{\theta ij}}\right)^{\beta_{ij}}$$
$$\times \pi_{i\epsilon N} e^{-\frac{1}{\theta ij^{\beta ij}}}[(T_0+t)^{\beta ij} - T_0^{\beta ij}] \times \pi_{i\epsilon N}[\left(e^{-\frac{t}{\theta ij}}\right)^{\beta_{ij}}]^{K_i}], \tag{25}$$

We consider this model for both objectives and use meta-heuristic algorithms to solve them.

3.2.1. Solution

Multi-objective optimization problem

We first propose nondominated solution genetic algorithms (NSGA II) as a solution strategy. The proposed study of this paper has adopted the method given by Chambar et al. (2012). Next, we use multi-objective particle swarm algorithm (MOPSO) as an alternative solution strategy. The implementation of the MOPSO is similar to the method developed by Chambar et al. (2012). Like other algorithms, the performance optimization algorithm becomes suitable when we decide how to set parameters regarding the characteristics of the issue, therefore topic has been studied very carefully to set the parameters. For this purpose, a statistical method known as response surface methodology (RSM) is used to determine the optimal levels of significant factors. NSGAII setting values for the parameters of the algorithm in which the number and mutation probability and probability of crossover and maximum number of repetitions are $\mathrm{n_{pop}}$, $\mathrm{p_m}$ *and* $\mathrm{p_c}$, respectively are stated in Table 9 and Table 10. In addition, Table 11 and Table 12 present details of parameter setting for MOPSO.

Table 9
Coding parameters and levels tested for NSGAII

Parameter	-1	0	1
$\mathrm{n_{pop}}$	50	75	100
$\mathrm{p_m}$	0.01	0.03	0.05
$\mathrm{p_c}$	0.5	0.65	0.8
max it	100	200	300

Table 10
Parameters setting values

Factor	$\mathbf{n_{pop}}$	$\mathbf{p_m}$	$\mathbf{p_c}$	$\mathbf{max\,it}$	**R**	**C**
The optimal value, coded	1	-1	-0.25	1		
The actual optimum value	100	0.01	0.6125	300	0.804897	0.797

Table 11
Coding parameters and levels tested for MOPSO

Parameter	-1	0	1
n_{pop}	50	75	100
Repository	50	75	100
m_u	0.1	0.15	0.2
$\max it$	100	200	300

Optimized values for MOPSO algorithm in which the population $n_{pop}$and p_m

Table 12
Parameters setting values for MOPSO

Factor	n_{pop}	Rep	m_u	$\max it$	R	C
The optimal value, coded	0.47	-1	0.07	-1		
The actual optimum value	74.8825	50	0.1535	100	0.79924	0.869

According to the results of parameter setting, NSGAII algorithm shows better results.

4. Summary and Conclusions

Meta-heuristic methods need to be examined for performance testing. But the important issue here is to determine the algorithm with better performance. To answer this question it is necessary to use several methods to assess the suitability of the results they achieved an overall results. In this section, to assess the effectiveness of the proposed method, we compared the results using some statistical tests.

4.1. Comparison of metaheuristic using hypothesis testing

In this section for a more accurate comparison of the proposed algorithms, we used the hypothetical tests. Each test was conducted with the percentage of 95%. For tests, we have used the software Minitab version 16.2 for all tests and the assumptions considered are as follows:

$$H_0: \mu_{SA} - \mu_{GA} = 0$$

$$H_1: \mu_{SA} - \mu_{GA} \neq 0$$

If the P-value is achieved by the application of meaningful level, we rejected the null hypothesis. This means that there is a significant difference between the two algorithms $(\alpha = 1 - 0.95 = 0.05)$.

Table 13
The results achieved by the algorithm criteria of SA and GA

Conclusion	p-value	The amount of t	Comparison criteria	Exam Number
Rejecting the null hypothesis	0.000	6.81	CPU	1
The null hypothesis is not rejected	0.562	0.59	OBJ	2

As is clear from the table, in terms of the execution time (CPU), that the performance of the methods is significantly different, because the P-value achieved is less than the significance level. Fig. 3 shows the performance of these two algorithms. SA has a running time of less than GA.

Fig. 5 and Fig. 6 show the performance of these two algorithms in terms of running time and objective function, respectively. Simulated annealing algorithm has a running time of less than GA.

As a result of the implementation of algorithms, simulated annealing algorithm has a better performance and this performance is statistically significant. In terms of value of the objective function, simulated annealing algorithm has a better performance, but the difference was not statistically significant. So finally, the use of simulated annealing algorithm is recommended.

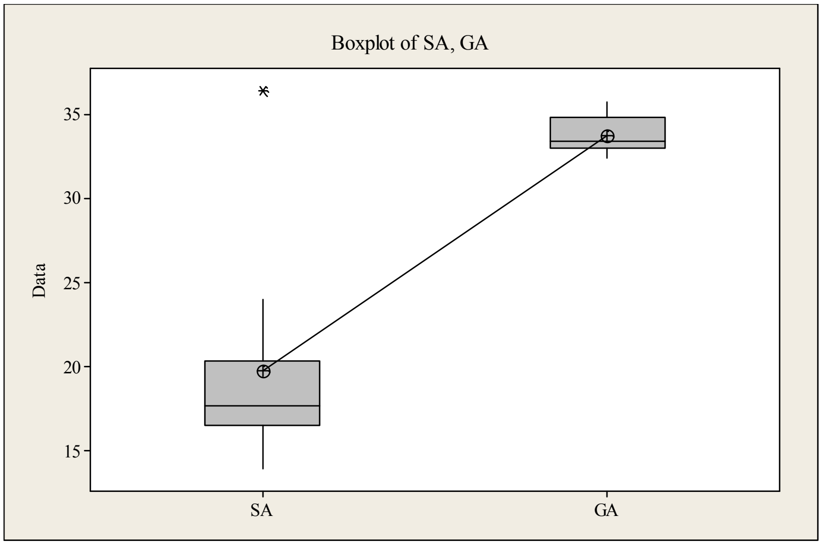

Fig. 5. Comparison of the performance in terms of runtime

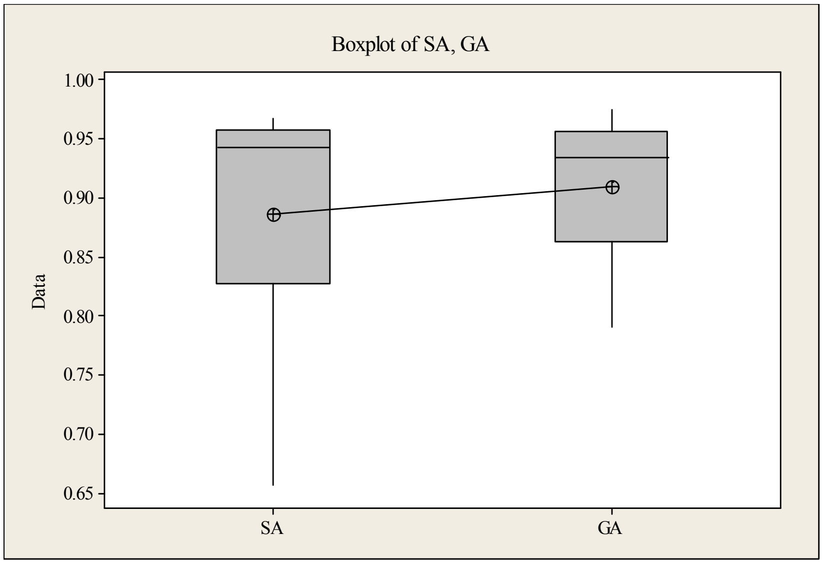

Fig. 6. The performance of the algorithms in terms of the objective function

We now compare the performance of the proposed MOPSO and NSGA II in terms of different factors. Tables 14-17 demonstrate the results of ANOVA test.

Table 14
The summary of the results of one-way ANOVA test on NSGA II versus MOPSO for small sizes

			MSI			
Conclusion	P	F	MS	SS	DF	Source
Equality hypothesis is confirmed	0.197	1.80	83589	83589	1	Factor
			46560	838088	18	Error
				921677	19	Total
			NPI			
Equality hypothesis is rejected	P	F	MS	SS	DF	Source
	0.0	45.67	561.8	561.8	1	Factor
			12.3	22.14	18	Error
				783.2	19	Total
			CPU TIME			
Equality hypothesis is rejected	P	F	MS	SS	DF	Source
NSGAII	**0.00**	2234.89	14679.11	14679.11	1	Factor
Better			6.57	118.23	18	Error
				14797.34	19	Total
			SI			
Equality hypothesis is rejected	P	F	MS	SS	DF	Source
	0.003	11.36	11.62	11.62	1	Factor
			1.02	18.42	18	Error
				30.04	19	Total

Table 15
The summary of the results of one-way ANOVA test on NSGA II versus MOPSO for medium sizes

			MSI			
Conclusion	P	F	MS	SS	DF	Source
Equality hypothesis is confirmed	0.094	3.13	138917	138917	1	Factor
			44435	799823	18	Error
				938739	19	Total
			NPI			
Equality hypothesis is rejected	P	F	MS	SS	DF	Source
	0.00	27.95	510.0	510.0	1	Factor
			18.3	328.5	18	Error
				838.5	19	Total
			CPU TIME			
Equality hypothesis is rejected	P	F	MS	SS	DF	Source
NSGAII	**0.00**	3733.77	14429.41	14429.41	1	Factor
Better			3.86	69.56	18	Error
				14498.97	19	Total
			SI			
Equality hypothesis is rejected	P	F	MS	SS	DF	Source
	0.008	9.01	1.955	1.955	1	Factor
			0.217	3.905	18	Error
				5.860	19	Total

Table 16
The summary of the results of one-way ANOVA test on NSGA II versus MOPSO for large sizes

			MSI			
Conclusion	P	F	MS	SS	DF	Source
Equality hypothesis is rejected	0.033	5.36	115236	115236	1	Factor
			21493	386869	18	Error
				502	19	Total
			NPI			
Equality hypothesis is rejected	P	F	MS	SS	DF	Source
	0.00	35.88	405.0	405.0	1	Factor
			11.3	203.2	18	Error
				608.2	19	Total
			CPU TIME			
Equality hypothesis is rejected	P	F	MS	SS	DF	Source
NSGAII	**0.00**	1030.48	13677.0	13677.0	1	Factor
Better			13.3	238.9	18	Error
				13915.9	19	Total
			SI			
Equality hypothesis is rejected	P	F	MS	SS	DF	Source
	0.002	9.01	11.676	11.676	1	Factor
			0.866	15.588	18	Error
				15.588	19	Total

Table 17
The summary of the results

			MSI			
Conclusion	P	F	MS	SS	DF	Source
Equality hypothesis is rejected	0.000	111.78	1470.2	1470.2	1	Factor
			1470.2	762.8	58	Error
				2233.0	59	Total
			NPI			
Equality hypothesis is rejected	P	F	MS	SS	DF	Source
	0.00	111.78	1470.2	1470.2	1	Factor
			13.2	762.8	58	Error
				2233.0	59	Total
			CPU TIME			
Equality hypothesis is rejected	P	F	MS	SS	DF	Source
NSGAII	**0.00**	5571.02	42775.90	42775.90	1	Factor
Better			7.68	445.34	58	Error
				43221.25	59	Total
			SI			
Equality hypothesis is rejected	P	F	MS	SS	DF	Source
	0.000	29.79	22.549	22.549	1	Factor
			0.757	43.895	58	Error
				66.444	59	Total

In order to compare the performance of the proposed studies we have used TOPSIS method |(Yoon & Hwang, 1981) and the result indicates that NSGAII performs better than MOPSO. Details of the computations are given in Appendix.

5. Conclusion

In this paper, we have presented an empirical investigation to propose a mathematical model for k-out-of-n redundancy problem. The proposed study of this paper has been formulated by considering two objectives; namely reliability and cost subject to volume and cost as constraints. Since the resulted model was classified as NP-Hard problem, we developed several methaheuristics to solve the resulted model and the performance of these method have been compared using different statistical tests. The preliminary results have indicated that NSGAII outperformed MOPSO in terms of different criteria. The proposed study of this paper can be extended for problems with more than two constraints and we leave it as a future research for interested researchers.

References

Aggarwal, K. K., Gupta, J. S., & Misra, K. B. (1975). A new heuristic criterion for solving a redundancy optimization problem. *IEEE Transactions on Reliability*, 24, 86-87.

Bulfin, R. L., & Liu, C .Y. (1985). Optimal allocation of redundant components for large systems. *IEEE Transactions on Reliability*, *34*, 241-7.

Chambari, A., Rahmati, S. H. A., & Najafi, A. A. (2012). A bi-objective model to optimize reliability and cost of system with a choice of redundancy strategies. *Computers & Industrial Engineering*, *63*(1), 109-119.

Chen, T., & You, P. (2005). Immune algorithms – based approach for redundant reliability problems with multiple component choices. *Computers in Industry*, 56, 195-205.

Chern, M. S. (1992). On the Computational Complexity of Reliability Redundancy Allocation in a Series System. *Operation Research Letters*, 11, 309-315.

Coit, D. W., & Liu, J. (2000). System reliability optimization with k- out – of – n subsystems. *International Journal of Reliability, Quality & Safety Engineering*, 7, 129-143.

Fyffe, D. E., Hines, W. W., & Lee, N. K. (1968). System Reliability Allocation and a Computational Algorithm. *IEEE Transactions on Reliability*, 17, 64-69.

Gopal, k., Aggarwal, K. K., & Gupta, J. S. (1978). An improved algorithm for reliability optimization. *IEEE Transactions on Reliability*, 27, 325-328.

Ha, C., & Kuo, W. (2006). Reliability redundancy allocation: an improved realization for non – convex nonlinear programming problem. *European Journal of Operational Research*, 171, 24-38.

Hsieh, Y – C. (2002). A Two – Phase Liner programming Approach for redundancy Allocation Problem. *Yugoslav Journal of Operations Research*, 12 (2), 227-236.

Kim, H., Bae, C., & Park, S. (2004). Simulated annealing algorithm for redundancy optimization with multiple component choices in advanced reliability modeling proceeding of the Asian international workshop. World Scientific, pp. 237- 244.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simmulated annealing. *science*, *220*(4598), 671-680.

Kulturel-Konak, S., Smith, A. E., & Norman, B. A. (2006). Multi-objective tabu search using a multinomial probability mass function. *European Journal of Operational Research*, *169*(3), 918-931.

Lee, H., Kuo, W., & Ha, c. (2003). Comparison of max – min approach and NN method for reliability optimization of series – parallel system. *Journal of Systems Science and Systems Engineering*, 12 (1), 39-48.

Liang, Y-C., & Smith, A. (2004). An ant colony optimization algorithm for the redundancy allocation problem (RAP). *IEEE Transactions on Reliability*. 53, 417- 423.

Misar, K.B. (1972). Reliability optimization of series – parallel system. *IEEE Transactions on Reliability*. 21, 230-238.

Misra, K. B., & Sharma, J. (1973). Reliability optimization of a system by zero-one programming. *Microelectronics Reliability*, *12*(3), 229-233.

Nahas, N., Nourelfath, M., & Ait – kadi, D. (2007). Coupling ant colony and the degraded ceiling algorithm for the redundancy allocation problem of series – parallel systems. *Reliability Engineering and System Safety*, 92, 211-222.

Nakagawa, Y., & Miyazaki, S. (1981). Surrogate Constraints Algorithm for Reliability Optimization Problems with Two Constraints. *IEEE Transactions on Reliability*, 30, 175-180.

Nakagava, Y., & Nakashima, K. (1977). A heuristic method for determining optimal reliability allocation. *IEEE Transactions on Reliability*, 26, 156-161.

Onishi, J., Kimura, S., James, R. J. W., & Nakagawa, Y. (2007). Solving the redundancy allocation problem with a mix of components using the improved surrogate constraint method. *IEEE Transactions on Reliability*, 56 (1), 94-101.

Sharma, J., & Yenkateswarn, K. V. (1971). A direct method for maximizing the system reliability. *IEEE Transactions on Reliability*, 20, 256-259.

Soltani, R. (2014). Reliability optimization of binary state non-repairable systems: A state of the art survey. *International Journal of Industrial Engineering Computations*, *5*(3), 339.

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

Taguchi, G. (1986). *Introduction to quality engineering: designing quality into products and processes*.

Yoon, K., & Hwang, C. L. (1981). TOPSIS (technique for order preference by similarity to ideal solution)–a multiple attribute decision making, w: Multiple attribute decision making–methods and applications, a state-of-the-at survey.

Appendix

Table A.1 demonstrates the results of the implementation of TOPSIS method.

Table A.1
The results of the implementation of TOPSIS method for comparing the performance of NSGA II versus MOPSO

	MSI		NPI		CPU		SI	
	NSGAII	MOPSO	NSGAII	MOPSO	NSGAII	MOPSO	NSGAII	MOPSO
1	196.1593	529.1037	11	9	2.2557	51.621	0.6067	0.663
2	484.0251	196.1938	17	7	3.3622	53.2074	0.4503	2.0319
3	961.3218	225.0222	26	5	2.2432	52.9653	0.3842	2.2214
4	576.0495	748.238	24	7	2.2933	52.1329	0.0035	5.1387
5	324.1314	441.1062	17	11	2.2399	59.8833	0.2327	0.9794
6	289.0494	169.1948	16	4	2.2689	58.5299	0.0057	2.3118
7	400.2498	289.2939	19	9	2.2645	60.6635	0.00238	1.5738
8	225.2351	256.1343	15	10	2.2458	59.5928	0.2452	0.483
9	529.0996	64.0694	16	8	2.2567	57.8256	0.7782	0.3368
10	626.0663	400.0553	20	5	2.2509	59.0917	0.0147	2.2316
11	784.0633	484.2341	22	9	2.2523	62.7554	0.3391	1.0397
12	196.1221	64.071	12	7	2.3199	54.82	0.0128	0.3418
13	361.0417	289.1113	20	9	2.2848	56.7112	0.008	0.6562
14	361.2663	121.2381	17	8	2.2544	56.0795	0.4659	0.3934
15	1.444	121.2381	29	8	2.2556	55.3229	0.4221	0.3934
16	729.2085	256.0581	20	9	2.2584	53.5956	0.4572	0.4788
17	484.0307	36.0168	18	5	2.2491	52.4892	0.4118	0.7937
18	36.054	169.1034	7	7	2.3242	56.6152	0.0278	2.4303
19	441.1031	361.4744	18	10	2.2378	56.9673	0.0207	1.1585
20	256.0613	81.0156	16	6	2.2543	54.5386	0.0188	0.7514
21	529.0722	361.1497	16	8	2.2517	55.6355	0.6569	2.9534
22	266.0634	144	15	4	2.2815	42.463	0	3.8971
23	121.0935	169.4744	9	10	2.668	56.9673	0.9426	1.1585
24	450.2314	161.8754	17	10	2.5412	57.6635	0.2452	0.3338
25	529.2324	400.0551	18	9	2.3433	57.3367	0.0654	0.2367
26	653.3349	335.8945	22	11	2.0012	51.0874	0.0071	1.7635
27	451.9913	371.8819	17	9	2.8961	53.1432	0.2457	2.8263
28	678.5942	487.8075	24	7	2.4589	58.7139	0.1494	2.8713
29	331.7563	225.2983	18	11	2.2679	60.4526	0.0384	0.6149
30	496.7691	332.5702	22	9	2.0146	53.2718	0.1972	1.1739