Decision Science Letters 4 (2015) 289–302

Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

Multi-objective availability-redundancy allocation problem for a system with repairable and non-repairable components

Hossein Zoulfaghari[a*], Ali Zeinal Hamadani[a] and Mostafa Abouei Ardakan[b]

[a]*Department of Industrial and Systems Engineering, Isfahan University of Technology, 84156-83111 Isfahan, Iran*
[b]*Department of Industrial Engineering, Daneshpajoohan Higher Education Institute, Isfahan, Iran*

C H R O N I C L E

Article history:
Received October 29, 2014
Received in revised format: March 2, 2015
Accepted April 22, 2015
Available online April 25 2015

Keywords:
System availability and reliability
Multi-objective optimization
Redundancy allocation
Repairable and non-repairable components
Genetic algorithm

A B S T R A C T

Reliability is one of the most important characteristics of the electrical and mechanical systems with applications in the space communication industries, internet networks, telecommunication systems, power generation systems, and productive facilities. What adds to the importance of reliability in these systems are system complications, nature of competitive markets, and increasing production costs due to failures. This paper investigates availability optimization of a system using both repairable and non-repairable components, simultaneously. The availability-redundancy allocation problems involve the determination of component availability (i.e., life time and repair time of the components) and the redundancy levels that produce maximum system availability. These problems are often subject to some constraints on their components such as cost, weight, and volume. To maximize the availability and to minimize the total cost of the system, a new Mixed Integer Nonlinear Programming (MINLP) model is presented. To solve the proposed model, an improved version of the genetic algorithm is designed as an efficient meta-heuristic algorithm. Finally, in order to verify the efficiency of the proposed algorithm, a numerical example of a system is presented that consists of both repairable and non-repairable components.

1. Introduction

Reliability optimization is an important topic that has attracted the attention of many researchers. In reliability optimization, the aim is to design a system structure that achieves a higher level of reliability at minimum budget and recourses. In order to improve the reliability of a specific system, one of the following approaches may be adopted: a) increasing the reliability of each component in the system, b) using the redundant components in parallel, c) combining (a) and (b) above, and d) reassignment of interchangeable components (Kuo & Prasad, 2000).

* Corresponding author. Tel:(+98)33915518
E-mail address: hamadani@cc.iut.ac.ir (H. Zoulfaghari)

doi: 10.5267/j.dsl.2015.4.007

Many different types of the reliability optimization problems have been encountered and investigated. However, the two common types include the Redundancy Allocation Problem (RAP) and the Reliability-Redundancy Allocation Problem (RRAP) (Kuo & Prasad, 2000). In RAP, there are discrete component choices with predefined characteristics such as reliability, cost and weight, where the goal is to find the optimal number of redundancies in each subsystem in order to maximize system reliability considering some system constraints. The Reliability-Redundancy Allocation Problem (RRAP) is formulated as a mixed-integer non-linear programming problem. It is a more complicated version compared with the RAP since the reliability and, therefore, other related specifications of the components are not predetermined and considered as decision variables. The purpose of RRAP is to maximize system reliability by selecting component reliability and component redundancy levels, which forms a difficult but realistic optimization problem. Component specifications such as cost, weight, and volume are defined as increasing non-linear functions of component reliability (Coit, 2003). Because of the complex nature of this problem, most classical mathematical methods have failed to yield optimal or near optimal solutions for this problem (Coelho, 2009). All previous studies of RRAP considered the system as a non-repairable one. In this paper, we considered, for the first time, RRAP when the system consists of both repairable and non-repairable components. Therefore, the problem may be renamed as the Availability-Redundancy Allocation Problem (ARAP).

Generally, the reliability of a system (or a component) is the probability that it will adequately perform its specified purpose for a specified period of time under specified environmental conditions (Barlow & Proschan, 1981) and availability is defined as the probability that a system is in its intended functional condition and, therefore, capable of being used in a stated environment (Hamadani, 1980). The main difference between these two concepts is that “availability” is used for repairable components while “reliability” is a functional index for the non-repairable ones. Furthermore, the term “reliability” can be used for the first failure of a repairable component while the term “availability” is used for the entire life of repairable components.

A system usually consists of a number of subsystems in which each subsystem uses several components in parallel. These components are placed in each subsystem according to the system application requirements; and each component has its own predefined availability (reliability), weight, volume, and cost, which should be considered in designing the system and in determining its optimization conditions. Optimization of such a system can be turned into a multi-objective problem due to the presence of several and, sometimes, conflicting objectives such as maximizing availability (or reliability), minimizing system cost, weight, and volume.

Since RAP and RRAP belong to the NP-hard class of optimization problems (Chern, 1992; Ha & Kuo, 2006) they are generally too difficult and time-consuming to solve using traditional optimization methods. More specifically when the problem size is large, most classical mathematical methods have failed to handle these optimization problems properly (Soltani, 2014).

Different methods have been developed for solving the RAPs. Exact optimization methods like dynamic programming (Fyffe et al., 1968; Ng & Sancho, 2001), integer programming (Misra & Sharma, 1991), Lagrangean multipliers (Misra, 1972), and various types of the meta-heuristic algorithms such as genetic algorithm (Hamadani & Khorshidi, 2013; Ardakan & Hamadani, 2014; Ardakan et al., 2015), ant-colony optimization (Chia & Smith, 2004), Immune algorithm (Chen & You, 2004), the surrogate constraint method (Onishi et al., 2007), variable neighborhood search (VNS) algorithms (Liang & Chen, 2007), Tabu search (TS) algorithm (Ouzineb et al., 2008), and particle swarm optimization (Beji et al., 2010; Wu et al., 2011; Garg, 2013) have been used for maximizing system reliability.

For solving the RRAPs, numerous meta-heuristic algorithms such as genetic algorithm (Zoulfaghari et al., 2014; Ardakan & Hamadani, 2014), artificial bee colony algorithm (Yeh & Hsieh, 2011), Particle

Swarm Optimization (Coelho, 2009), harmony search (Zou et al., 2011; Wang & Li, 2012; Zou et al., 2012), cuckoo search algorithm (Valian, 2012; Valian et al., 2013), and imperialist competitive algorithm (Afonso et al., 2013) have been widely employed over the past decade.

As already mentioned, this paper deals with the RRAPs whose main difference from similar problems lies in the assumption that the system involves both repairable and non-repairable components; hence it must be considered as ARAP. To solve this problem, a new mixed integer non-linear programming model is introduced and solved by an improved version of Genetic Algorithm (GA). To solve the proposed mathematical model and to show the capability of the proposed GA in handling the problem, a modified problem from the literature is considered and the GA results are compared with those of the Improved Particle Swarm Optimization (IPSO) algorithm (Wu et al., 2011) as one of the best algorithms reported in the literature.

In contrast to the studies conducted on reliability optimization, fewer studies have been devoted to the availability allocation and optimization to investigate the optimal failure and repair rates of each component in a system aimed at maximizing (or minimizing) the objectives. In most cases, the problem of availability allocation and optimization can be defined as a multi-objective optimization problem, which aims to maximize system availability and minimize tits cost (Elegbede & Adjallah, 2003). Levitin and Lisnianski (2001) introduced a model in which the cost of designing the system is fixed and its purpose is to optimize system availability. Also, Zio and Bazzo (2011) presented an analysis on level diagrams of Pareto front for the redundancy allocation problem. Their aims were to maximize system availability and minimize the cost and weight of the whole system. Chiang and Chen (2007) proposed a new multi-objective genetic algorithm, namely the simulated annealing based on a multi-objective genetic algorithm (MOGA), to resolve the availability allocation and optimization problems of a repairable system. In the existing literature, a number of researchers have investigated the theoretical problems of availability modeling (Srivasvata & Fahim, 1988; Zhao, 1994; Lee, 2000; Cao et al., 2002; Sericola, 1999; Ma et al., 2001; Dewinter, 2002). The present article aims at both availability and reliability allocation and availability optimization.

The rest of the paper is organized as follows. In Section 2, the structure of the problem is presented and the proposed model is established. A description of the proposed genetic algorithm is provided in Section 3 and a numerical example for the problem is given Section 4. Finally, the paper concludes with results, conclusions, and some suggestions for future study.

2. Problem definition and the proposed model

Series-parallel systems are used here as a well-known system structure for describing the proposed model. The common structure of a series-parallel system is illustrated in Fig. 1.

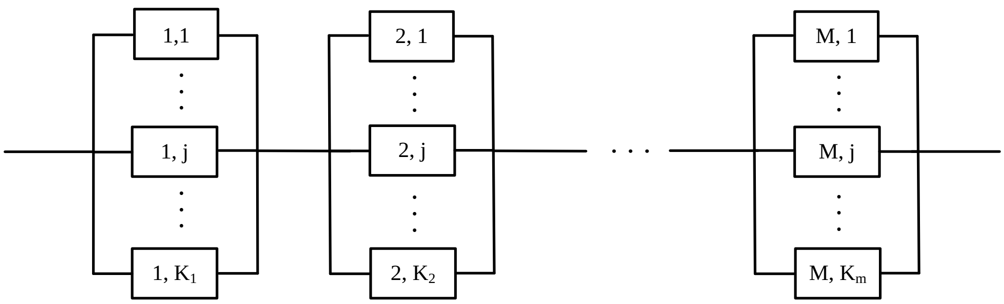

Fig.1. General structure of series-parallel system.

Without loss of generality, it is supposed that all the components in each subsystem are identical (the components have the same reliability and availability). In general, one has two objectives for these types of systems: maximizing system availability and minimizing system cost (Elegbede & Adjallah, 2003).

As already mentioned in the previous sections, most studies on redundancy allocation problems have considered optimization of system reliability and it has been supposed that all the components are non-repairable, at the cost of neglecting the availability and maintainability of components. Furthermore, in cases where system availability is considered, it has been assumed that the system consists of only repairable components. This is while in real world conditions, there are a few systems that are designed to use either repairable or non-repairable components. In fact, most complicated systems consist of both repairable and non-repairable subsystems. Examples include systems composed of electronic and mechanical sections such as automobile motor systems, and airplane systems, production systems, where the electronic sections consist of non-repairable components while the mechanical sections have repairable ones.

Recently, Zoulfaghari et al. (2014) presented a mathematical model for such a system consisting of both repairable and non-repairable components. They developed the model for the case in which the reliability or availability of the components are pre-determined. As an extension to that study, a new model is developed in this paper to consider the problem for the case in which the reliability and availability of the components are not given in advance but they are considered as decision variables. This problem can be called the availability-redundancy allocation problem. It is, therefore, assumed that some subsystems use non-repairable components while others have repairable ones. In this case, it will not be possible to use the reliability formulation for the objective function since some of the components are repairable. It follows then that the modeling should be accomplished in such a way that the objective function is considered for maximizing system availability.

In the next subsections, notations and relations used in the proposed mixed integer non-linear programming model and the mathematical representation of the model are presented.

2.1 Notation

$A_{sys}(t)$	Availability of the system at time t,
C_{sys}	Cost of the system,
$R_i(\mathrm{t})$	Reliability of the component used in the subsystem i at time t,
$A_i(\mathrm{t})$	Availability of the component used in the subsystem i at time t,
$Av_i(\mathrm{t})$	Availability of the subsystem i at time t,
s	Number of total subsystems,
s_1	Number of subsystems that include non-repairable components,
A	Set of subsystems including repairable components,
R	Set of subsystems including non-repairable components,
n_i	Number of components in subsystem i,
α_i, β_i	Constants representing the physical characteristic of each component in subsystem i,
T	Operating time during which the component must not fail,
$a_i, b_i, q_i \geq 0$ *and* $p_i \leq 0$	Real numbers used to calculate the cost of repairable component in subsystem i,
$E(X), E(Y)$	Average life time and repair time of repairable components,

$w_i, w_i v_i^2$	Real numbers used to calculate the weight and volume of components in subsystem i,
V	Maximum permitted volume of the system,
W	Maximum permitted weight of the system,
N_i	Maximum permitted number of components in subsystem i,
P_i	Minimum permitted number of components in subsystem i
L_{R_i}, U_{R_i}	Lower and upper bounds of reliability in subsystem i,
L_{A_i}, U_{A_i}	Lower and upper bounds of availability in subsystem i.

3. The mathematical model

In this section, a bi-objective mathematical model is developed and presented for the problem. The proposed model is as follows:

$$f_1: \; Max \;\; A_{sys}(t) = Max\left\{\prod_{i}\left(1-\left(1-Av_i(t)\right)^{n_i}\right)\right\} = Max\left\{\prod_{i\in R}\left(1-\left(1-R_i(t)\right)^{n_i}\right)\times\prod_{i\in A}\left(1-\left(1-A_i(t)\right)^{n_i}\right)\right\} \tag{1}$$

$$f_2: \; Min \;\; C_{sys} = \sum_{i\in R}\alpha_i\left(-\frac{T}{\ln R_i}\right)^{\beta_i}\left(n_i + e^{\frac{n}{s_1}}\right) + \sum_{i\in A} n_i\left(a_i\left(\frac{1}{E(X)}\right)^{p_i} + b_i\left(\frac{1}{E(Y)}\right)^{q_i}\right) \tag{2}$$

s.t:

$$\sum_{i\in(A\cup R)} w_i n_i e^{\frac{n_i}{s_1}} \le W \tag{3}$$

$$\sum_{i\in(A\cup R)} w_i v_i^2 n_i^2 \le V \tag{4}$$

$$p_i \le n_i \le N_i \;, \quad \forall i \in (A\cup R) \tag{5}$$

$$n_i \in Z^+ \;, \quad \forall i \in (A\cup R) \tag{6}$$

$$L_{R_i} \le R_i \le U_{R_i} \;, \; \forall i \in R \tag{7}$$

$$L_{A_i} \le A_i \le U_{A_i} \;, \; \forall i \in A \tag{8}$$

In this model, Eq. (1) denotes the first objective function for maximizing system availability. The function is a multiplication of two parts: the first maximizes the reliability of non-repairable subsystems while the second maximizes the availability of the subsystems with repairable components. Eq. (2) is the second objective function which is related to the total cost of the system. This function also consists of two parts: the first one calculates the cost of repairable components. In this part, $e^{\frac{n_i}{s_1}}$ accounts for the interconnecting hardware, α_i and β_i are the constants representing the physical characteristic of each component in subsystem $i \in R$, and T is the operating time during which the component must not fail (Dhingra, 1992). The second part calculates the cost of repairable components, where a_i, b_i, and q_i are positive real numbers, while p_i is negative, $(\forall i \in A)$ (Chiang & Chen ,2007). Eq. (3) shows the constraint on maximum weight, while Eq. (4) indicates the constraint of maximum volume for the system. Constraint (5) is related to the maximum and minimum numbers of permitted components in each subsystem and constraint (6) shows the conditions of the decision variables. Constraint (7) denotes the domain of reliability for non-repairable components and constraint (8) denotes the domain of availability for repairable components. For the purposes of system availability analysis and calculation, we assume that the life time and repair time are exponential distributions. Therefore:

$$f_1 : Max\, A_{sys}(t) = Max\left\{\prod_i (1-(1-Av_i(t))^{n_i})\right\}$$

$$= \prod_{i\in R}(1-(1-e^{-\lambda_i t})^{n_i}) \times \prod_{i\in A}(1-(1-\frac{\lambda_i}{\lambda_i+\mu_i}-\frac{\mu_i}{\lambda_i+\mu_i}e^{-t(\frac{1}{\lambda_i}+\frac{1}{\mu_i})})^{n_i} \tag{9}$$

$$f_2 : Min\, C_{sys} = \sum_{i\in R}\alpha_i\left(-\frac{T}{\lambda_i t}\right)^{\beta_i}\left(n_i + e^{\frac{n_i}{s_1}}\right) + \sum_{i\in A} n_i(a_i(\lambda_i)^{p_i} + b_i(\mu_i)^{q_i}) \tag{10}$$

subject to

$$\sum_{i\in(A\cup R)} w_i n_i e^{\frac{n_i}{s_1}} \le W \tag{11}$$

$$\sum_{i\in(A\cup R)} w_i v_i^2 n_i^2 \le V \tag{12}$$

$$p_i \le n_i \le N_i\ , \quad \forall i \in (A\cup R) \tag{13}$$

$$n_i \in Z^+\ , \qquad \forall i \in (A\cup R) \tag{14}$$

$$L_{\lambda_i} \le \lambda_i \le U_{\lambda_i}\ , \quad \forall i \in (A\cup R) \tag{15}$$

$$L_{\mu_i} \le \mu_i \le U_{\mu_i}\ , \quad \forall i \in A \tag{16}$$

Chern (1992) proved that the redundancy allocation problem in its simplest form of series system was an NP-hard problem. Therefore, the proposed model which is more complicated than the model given by Chern would also be NP-hard. In order to maximize the objective function of this model, it is reasonable to use a meta-heuristic method such as GA as used in this paper.

4. The proposed genetic algorithm

Genetic algorithm (GA) was first proposed by Holland (1975) and has been one of the most applicable meta-heuristic methods for solving combinatorial optimization problems over the past three decades. Generally, GA is employed for solving models with one objective function. In this paper, this algorithm is used to solve the proposed bi-objective model. In the first step, the ε-constrained method is used to determine the optimized value for the second objective function (cost). This value is then used in the problem constraints, and the single-objective problem is solved by the genetic algorithm. Since the value obtained for the second objective function is the minimum value of the system cost, by the optimal value for the first objective function can also be obtained by the gradual increase of this value in the constraint. It is clear that whenever the cost increases, due to releases in the added constraint, availability of the problem should be better than before. This trend produces different solutions for the problem for different levels of costs and availability, which makes the decision makers able to select appropriate solutions by considering different criteria. Below is presented a complete description of the proposed genetic algorithm.

4.1 Chromosome definition

For the proposed GA, each chromosome includes $3\times s$ genes where the first row presents the number of components used in subsystems, the second row presents the life times of the components in each subsystem and the third row shows the repair times of the components in the repairable subsystems. These genes are randomly produced at given intervals. In other words, the value for each gene should be verified in constraints (13), (15), and (16). Fig. 2 represents the chromosome structure considered for this problem.

	Non-Repairable					Repairable				
Components level	n_1	n_2	...	n_{s_1-1}	n_{s_1}	n_{s_1+1}	n_{s_1+2}	...	n_{s-1}	n_s
Life time	λ_1	λ_2	...	λ_{s_1-1}	λ_{s_1}	λ_{s_1+1}	λ_{s_1+2}	...	λ_{s-1}	λ_s
Repair Time	0	0	0	0	0	μ_{s_1+1}	μ_{s_1+2}	...	μ_{s-1}	μ_s

Fig. 2. Chromosome (solution representation)

4.2 Fitness function

As mentioned above, to solve the proposed bi-objective model by GA, the second objective is considered as a constraint so that the model becomes a single-objective one. Fitness function (*ff*) is the value of the first objective function (system availability) plus the penalty for constraint violation. In other words, the problem constraints are added to the objective function in such a way that if one solution goes beyond the constraints, a relatively large amount of penalty is added to the objective function. This penalty keeps the feasibility of the final solution while it also provides the search in the infeasible space of the problem. The search in the infeasible space leads to an appropriate diversity for the genetic algorithm.

4.3 Initial population

In order to produce an initial population, *Pop* chromosomes are randomly generated. In this paper, population size (*Pop*) is equal to 100. This number of population has been used in previous studies such as Safari (2012) and Debb et al. (2002). Safari (2012) states that in problems with a big solution space, the number of primary population should be more than 100.

4.4 Selection

In order to select the required chromosomes in the crossover operation, the following steps need to be taken. The fitness function (*ff*) is calculated for all the existing chromosomes (*Pop*) in the present population. Then, from *Pop* present chromosomes, k chromosomes are randomly selected and sorted based on *ff*. The chromosome with the largest fitness functions (availability-penalties) is selected as the parent for generating a new population. This process will be repeated *Pop* times until *Pop* parents are finally selected for the crossover and mutation operators.

4.5 Crossover

Crossover takes place at a certain rate. Using the crossover operation, six offspring are generated from each two parents. The two parents and the six offspring create eight chromosomes and the two premier chromosomes based on *ff* are selected to transfer to the next generation. As a result, there will be *Pop* population at the end of the crossover operations. In order to produce these six offspring from the two selected parents, the following steps are taken:

Step 1: Two random numbers $(m_1,\ m_2)$ are selected such that m_1 is in the interval (1 to s_1-1) and m_2 is selected from s_1+1 to $s-1$.

Step 2: The genes in the interval (1 to m_1) for each parent are exchanged to produce two offspring.

Step 2: The genes in the interval ($s_1 + 1$ to m_2) for each parent are exchanged to produce two other offspring.

Step 3: The genes in the interval (1 to m_1) and those in the interval ($s_1 + 1$ to m_2) for each parent are exchanged at the same time to produce the last two offspring.

This kind of crossover leads to in the enhanced capability of the algorithm for finding better solutions. Here, an illustrative example is used to explain the crossover operation. Suppose that for the case in Fig. 3, there is a system with 6 subsystems, 3 of which include non-repairable components and 3 include repairable ones while: $m = 2$, $m' = 4$; $1 \le n_i \le 5 \quad \forall i = 1, 2, ..., 6$; $0.0004 \le \lambda_i \le 0.002 \quad \forall i = 1, 2, ..., 6$; $0.02 \le \mu_i \le 0.35 \quad \forall i = 4, 5, 6$.

Now, there are eight chromosomes and we should select two superior ones. For this selection, *ff* is calculated for all the eight chromosomes and they are compared with each other. Finally, two chromosomes with the highest value of *ff* are selected.

4.6 Mutation

The mutation operator is also used at a certain rate which is less than that of the crossover operator. The main purpose of applying the mutation operator is to increase diversity and to avoid trapping in the local optimization. In this operator, one offspring is randomly selected from among two chromosomes produced by the crossover operator.

	Non-repairable			repairable		
First Parent	2	3	1	3	4	1
	0.001	0.004	0.002	0.0005	0.0004	0.002
	0	0	0	0.2	0.27	0.34
Second Parent	1	4	1	2	4	2
	0.001	0.0005	0.0004	0.001	0.002	0.001
	0	0	0	0.3	0.31	0.29
First Child	1	4	1	3	4	1
	0.001	0.0005	0.002	0.0005	0.0004	0.002
	0	0	0	0.2	0.27	0.34
Second Child	2	3	1	2	4	2
	0.001	0.0004	0.0004	0.001	0.002	0.001
	0	0	0	0.3	0.31	0.29
Third Child	2	3	1	2	4	1
	0.001	0.0004	0.002	0.001	0.0004	0.002
	0	0	0	0.3	0.27	0.34
Fourth Child	1	4	1	3	4	2
	0.001	0.0005	0.0004	0.0005	0.0002	0.0001
	0	0	0	0.2	0.31	0.29
Fifth Child	1	4	1	2	4	1
	0.001	0.0005	0.001	0.001	0.0004	0.002
	0	0	0	0.3	0.27	0.34
Sixth Child	2	3	1	3	4	2
	0.001	0.0004	0.0004	0.0005	0.002	0.001
	0	0	0	0.2	0.31	0.29

Fig. 3. Crossover Operation

Two random numbers (m'_1, m'_2) are considered such that m'_1 is selected from (1 to s_1) and m'_2 is selected from ($s_1 + 1$ to s) and the values of these two genes are exchanged. Then, *ff* is calculated for the muted offspring and compared with the value for *ff* of the pre-mutation chromosome. If the value for *ff* of the new offspring is greater than that of the previous one, it will then be replaced by the newly generated offspring. Otherwise, the previous offspring remains as the superior ones. For example, suppose that in Fig. 4, the offspring has been selected for mutation, $m'_1 = 3$ and $m'_2 = 5$. Fig. 4 represents the mutation operator for these random values.

	Non-repairable			repairable		
Child before mutation	1	4	1	2	4	1
	0.001	0.0005	0.002	0.001	0.0004	0.002
	0	0	0	0.3	0.27	0.34
Child after mutation	1	4	2	2	3	1
	0.001	0.0005	0.004	0.001	0.00035	0.002
	0	0	0	0.3	0.24	0.34

Fig. 4. Mutation Operation

4.7 Stopping criteria

The GA process will continue until a predefined number of iterations (*Gen*). In this paper, the number of iterations is set equal to 500 generations.

5. A numerical example

This part of the paper includes an example whose data is a combination of those applied in Zou et al. (2011) and Chiang and Chen (2007). In this example, the system includes 10 subsystems where subsystems 1 to 5 have non-repairable components while subsystems 6 to 10 have repairable components. Maximum allowable weight and volume for the system are 300 (units of weight) and 380 (units of volume), respectively. Maximum and minimum numbers of allowable components in each subsystem have been considered as 5 and 1, respectively. Other details are presented in Table 1.

To solve this problem, the improved genetic algorithm proposed in this paper has been used. The improved GA designed here has been coded by MATLAB software and run on a computer with 2G of RAM. In this paper, some preliminary experiments were used and the crossover and mutation rates were set to 0.9 and 0.3, respectively. Also, the population size and maximum generations were taken to be 100 and 500, respectively.

Table 1
details of problem

		w_i	$w_i v_i^2$	$\alpha_i \times 10^5$	β_i	a_i	b_i	p_i	q_i	L_{λ_i}	U_{λ_i}	L_{μ_i}	U_{μ_i}
non-repairable	Sub.1	7	1	2.33	1.5	-	-	-	-	0.0004	0.002	-	-
	Sub.2	8	2	1.45	1.5	-	-	-	-	0.0005	0.002	-	-
	Sub.3	8	3	0.541	1.5	-	-	-	-	0.0004	0.002	-	-
	Sub.4	6	4	8.05	1.5	-	-	-	-	0.0005	0.002	-	-
	Sub.5	9	2	1.95	1.5	-	-	-	-	0.0003	0.002	-	-
Repairable	Sub.6	5	4	-	-	0.040	0.04	-0.32	0.34	0.0004	0.002	0.20	0.340
	Sub.7	5	1	-	-	0.20	0.02	-0.16	0.17	0.0005	0.002	0.35	0.595
	Sub.8	6	2	-	-	0.50	0.10	-0.80	0.85	0.0004	0.002	0.40	0.680
	Sub.9	7	2	-	-	0.80	0.08	-0.64	0.68	0.0005	0.002	0.45	0.765
	Sub.10	7	3	-	-	0.12	0.12	-0.96	1.02	0.0003	0.002	0.35	0.595

In order to show the capability of the genetic algorithm, the problem has also been solved by the Improved Particle Swarm Optimization (IPSO) algorithm proposed in Wu et al. (2011), which is considered as one of the best algorithms in RAP so far. They demonstrated that IPSO is an algorithm with a great capability for solving these problems. Therefore, this algorithm was selected as suitable for making comparisons. Also, for the IPSO, the population size was selected to be $PS = 100$, maximal number of iterations was set at K=500, and the mutation probability to $p_m = 0.05$. The two algorithms were run 20 times for each value of cost moving from 1000 to 4500 and the results were presented in Table 2. In this Table, SD represents standard deviation which is based on the 20 converging values of the objective function. SD is expressed as follows:

$$SD = \sqrt{\frac{1}{20-1}\sum_{k=1}^{20}(f_k - \overline{f})^2} \tag{17}$$

where, f_k is the k^{th} converging value of the objective function, and $\overline{f}$ represents the average value (median) of the objective function. Based on the four criteria (Best, Worst, Median, and SD) in Table 2, the improved GA proposed here outperformed IPSO for all groups of cost values. The outperformance of the improved GA is due to the use made of the crossover and mutation operations designed here. These results also show that the availability of the system increases with increasing cost. The solutions thus obtained are shown in Table 2 and the detailed solution for the cost value of 1000 is presented in Table 3.

Table 2
Comparison of results for example between GA and IPSO

Solution number	Cost	Algorithm	Best	Worst	Median	SD
1	1000	GA	**0.99875429**	**0.99412757**	**0.99674852**	6.819e-004
		IPSO	0.99377264	0.98675214	0.99474125	5.437e-005
2	1500	GA	**0.99928294**	**0.99884234**	**0.99914824**	5.932e-005
		IPSO	0.99919742	0.99823549	0.99861254	3.833e-004
3	2000	GA	**0.99959213**	**0.99908954**	**0.99938541**	5.937e-005
		IPSO	0.99937157	0.99842713	0.99903917	2.048e-005
4	2500	GA	**0.99975197**	**0.99929437**	**0.99957136**	6.328e-004
		IPSO	0.99961843	0.99908291	0.99947126	7.994e-004
5	3000	GA	**0.99989534**	**0.99942671**	**0.99982591**	3.092e-005
		IPSO	0.99980216	0.99934901	0.99965731	3.296e-004
6	3500	GA	**0.99991759**	**0.99957593**	**0.99989315**	3.909e-005
		IPSO	0.99987174	0.99940917	0.99969427	6.393e-004
7	4000	GA	**0.99996429**	**0.99970286**	**0.99990973**	2.061e-004
		IPSO	0.99990197	0.99951024	0.99971907	8.001e-004
8	4500	GA	**0.999989271**	**0.99990814**	**0.99995716**	1.043e-005
		IPSO	0.999941826	0.99962794	0.99989716	3.073e-004

Fig. 5 shows the convergence of the objective function value in each generation. This solution belongs to one of the 20 iterations for a cost equal to 2000. The near-optimal solution (objective function value = 0.99935684) was achieved after approximately 425 generations.

Table 3
Details of solution number 1with cost=1000

	Sub.1	**Sub.2**	**Sub.3**	**Sub.4**	**Sub.5**	**Sub.6**	**Sub.7**	**Sub.8**	**Sub.9**	**Sub.10**
n_i	3	3	4	3	4	3	2	2	2	2
λ_i	0.000883	0.000796	0.001521	0.001251	0.001621	0.0004	0.0003	0.0005	0.0004	0.0003
μ_i	-	-	-	-	-	0.3285	0.5950	0.7650	0.6800	0.5950

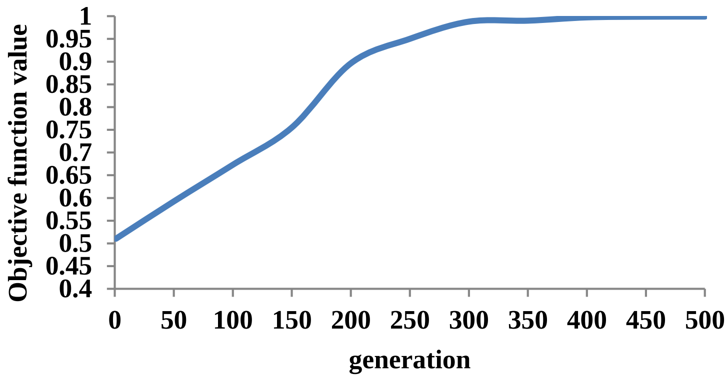

Fig. 5. Objective function value convergence

The Pareto front results are illustrated in Fig. 6 by using the median of availability. This Figure shows that for each value of cost, the availability obtained by GA is better than that obtained by IPSO.

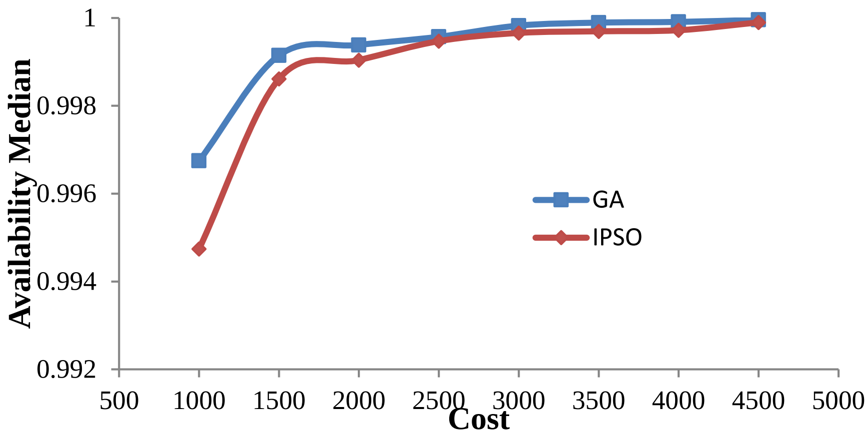

Fig. 6. Pareto front of results for GA and IPSO

These results also demonstrated that the convergence and stability of the proposed GA are better than those of the IPSO algorithm. Fig. 7 shows that in 7 out of 8 cases, the value of SD for GA was smaller than that for IPSO. The precision of the genetic algorithm is also observed to be higher than that of the IPSO algorithm. These indicate that the proposed GA is a robust optimization algorithm.

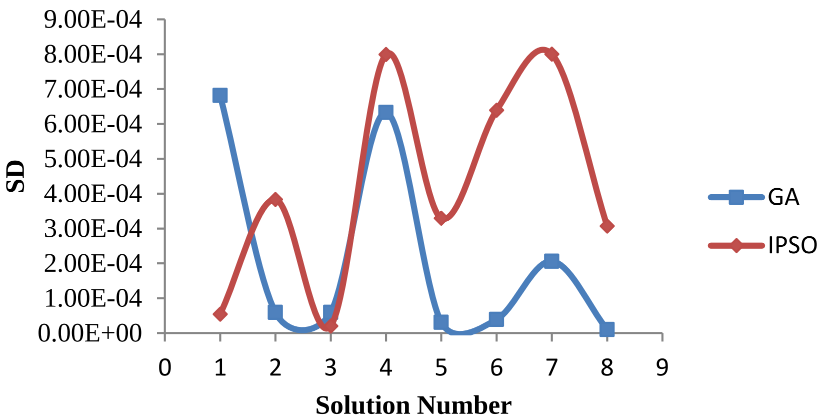

Fig. 7. Compare stability algorithms

6. Summary and Conclusions

In redundancy allocation problems (RAPs), it is commonly assumed that the system consists of either only repairable or non-repairable components. As an extension to this assumption, a system consisting of both repairable and non-repairable components was considered in this paper and a new mathematical model was developed for the system. The problem has been formulated as a nonlinear integer programming model subject to a number of given constraints. Since the RAPs belong to the NP-hard class of optimization problems, it is not easy to solve the proposed model in real cases, especially for large systems. Therefore, meta-heuristic methods are suggested for solving such a hard and complex problem. In this paper, an improved genetic algorithm (GA) was developed as an effective meta-heuristic algorithm for solving the RAP. The results obtained by the genetic algorithm showed the satisfactory and appropriate availability of the system. In addition, the precision of the genetic algorithm was shown to be high when compared with one of the best algorithms reported in the literature. For future work, the authors are investigating the extension of the proposed model by introducing fuzzy numbers.

References

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

Ardakan, M. A., & Hamadani, A. Z. (2014). Reliability optimization of series–parallel systems with mixed redundancy strategy in subsystems. *Reliability Engineering & System Safety*, *130*, 132-139.

Ardakan, M. A., Hamadani, A. Z., & Alinaghian, M. (2015). Optimizing bi-objective redundancy allocation problem with a mixed redundancy strategy. *ISA transactions*, *55*, 116–128.

Afonso, L. D., Mariani, V. C., & dos Santos Coelho, L. (2013). Modified imperialist competitive algorithm based on attraction and repulsion concepts for reliability-redundancy optimization. *Expert Systems with Applications*, *40*(9), 3794-3802.

Barlow RE, Proschan F. (1981).Statistical theory of reliability and life testing probability model. Silver spring.

Beji, N., Jarboui, B., Eddaly, M., & Chabchoub, H. (2010). A hybrid particle swarm optimization algorithm for the redundancy allocation problem. *Journal of Computational Science*, *1*(3), 159-167.

Cao, Y., Sun, H., Trivedi, K. S., & Han, J. J. (2002). System availability with non-exponentially distributed outages. *Reliability, IEEE Transactions on*, *51*(2), 193-198.

Chen, T. C., & You, P. S. (2005). Immune algorithms-based approach for redundant reliability problems with multiple component choices. *Computers in Industry*, *56*(2), 195-205.

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

Liang, Y. C., & Smith, A. E. (2004). An ant colony optimization algorithm for the redundancy allocation problem (RAP). *Reliability, IEEE Transactions on*,*53*(3), 417-423.

Chiang, C. H., & Chen, L. H. (2007). Availability allocation and multi-objective optimization for parallel–series systems. *European journal of operational research*, *180*(3), 1231-1244.

dos Santos Coelho, L. (2009). An efficient particle swarm approach for mixed-integer programming in reliability–redundancy optimization applications.*Reliability Engineering & System Safety*, *94*(4), 830-837.

Coit, D. W. (2003). Maximization of system reliability with a choice of redundancy strategies. *IIE transactions*, *35*(6), 535-543.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, *6*(2), 182-197.

DeWinter, F. A., Paes, R., Vermaas, R., & Gilks, C. (2002). Maximizing large drive availability. *Industry Applications Magazine, IEEE*, *8*(4), 66-75.

Dhingra, A. K. (1992). Optimal apportionment of reliability and redundancy in series systems under multiple objectives. *Reliability, IEEE Transactions on*,*41*(4), 576-582.

Elegbede, C., & Adjallah, K. (2003). Availability allocation to repairable systems with genetic algorithms: a multi-objective formulation. *Reliability Engineering & System Safety*, *82*(3), 319-330.

Fyffe, D. E., Hines, W. W., & Lee, N. K. (1968). System reliability allocation and a computational algorithm. *Reliability, IEEE Transactions on*, *17*(2), 64-69.

Garg, H. (2013). Performance analysis of complex repairable industrial systems using PSO and fuzzy confidence interval based methodology. *ISA transactions*, *52*(2), 171-183.

Ha, C., & Kuo, W. (2006). Reliability redundancy allocation: An improved realization for non-convex nonlinear programming problems. *European Journal of Operational Research*, *171*(1), 24-38.

Hamadani, A. Z., & Khorshidi, H. A. (2013). System reliability optimization using time value of money. *The International Journal of Advanced Manufacturing Technology*, *66*(1-4), 97-106.

Hamadani A.Z. (1980). Availability and Reliability Modeling, Ph.D. thesis, University of Bradford, England.

Holland JH. (1975). Adaptation in natural and artificial systems. Michigan: University of Michigan Press.

Kuo, W., & Prasad, V. R. (2000). An annotated overview of system-reliability optimization. *Reliability, IEEE Transactions on*, *49*(2), 176-187.

Lee, K. W. (2000). Stochastic models for random-request availability. *Reliability, IEEE Transactions on*, *49*(1), 80-84.

Levitin, G., & Lisnianski, A. (2001). A new approach to solving problems of multi-state system reliability optimization. *Quality and reliability engineering international*, *17*(2), 93-104.

Liang, Y. C., & Chen, Y. C. (2007). Redundancy allocation of series-parallel systems using a variable neighborhood search algorithm. Reliability Engineering & System Safety, 92(3), 323-331.

Ma, Y., Han, J. J., & Trivedi, K. S. (2001). Composite performance and availability analysis of wireless communication networks. *Vehicular Technology, IEEE Transactions on*, *50*(5), 1216-1223.

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

Misra, K. B. (1972). Reliability optimization of a series-parallel system. *Reliability, IEEE Transactions on*, *21*(4), 230-238.

Ng, K. Y., & Sancho, N. G. F. (2001). A hybrid dynamic programming/depth-first search algorithm, with an application to redundancy allocation. *IIE Transactions*, *33*(12), 1047-1058.

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

Ouzineb, M., Nourelfath, M., & Gendreau, M. (2008). Tabu search for the redundancy allocation problem of homogenous series–parallel multi-state systems. *Reliability Engineering & System Safety*, *93*(8), 1257-1272.

Safari, J. (2012). Multi-objective reliability optimization of series-parallel systems with a choice of redundancy strategies. *Reliability Engineering & System Safety*, *108*, 10-20.

Sericola, B. (1999). Availability analysis of repairable computer systems and stationary detection. *Computers, IEEE Transactions on*, *48*(11), 1166-1172.

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), 339-364.

Srivastava, V. K., & Fahim, A. (1988). k-out-of-m system availability with minimum-cost allocation spares. *Reliability, IEEE Transactions on*, *37*(3), 287-292.

Valian, E., & Valian, E. (2013). A cuckoo search algorithm by Lévy flights for solving reliability redundancy allocation problems. *Engineering Optimization*,*45*(11), 1273-1286.

Valian, E., Tavakoli, S., Mohanna, S., & Haghi, A. (2013). Improved cuckoo search for reliability optimization problems. *Computers & Industrial Engineering*, *64*(1), 459-468.

Wang, L., & Li, L. P. (2012). A co-evolutionary differential evolution with harmony search for reliability–redundancy optimization. *Expert Systems with Applications*, *39*(5), 5271-5278.

Wu, P., Gao, L., Zou, D., & Li, S. (2011). An improved particle swarm optimization algorithm for reliability problems. *ISA transactions*, *50*(1), 71-81.

Yeh, W. C., & Hsieh, T. J. (2011). Solving reliability redundancy allocation problems using an artificial bee colony algorithm. *Computers & Operations Research*, *38*(11), 1465-1473.
Zhao, M. (1994). Availability for repairable components and series systems. *Reliability, IEEE Transactions on*, *43*(2), 329-334.
Zio, E., & Bazzo, R. (2011). Level Diagrams analysis of Pareto Front for multi-objective system redundancy allocation. *Reliability Engineering & System Safety*, *96*(5), 569-580.
Zou, D., Gao, L., Wu, J., Li, S., & Li, Y. (2010). A novel global harmony search algorithm for reliability problems. *Computers & Industrial Engineering*, *58*(2), 307-316.
Zou, D., Gao, L., Li, S., & Wu, J. (2011). An effective global harmony search algorithm for reliability problems. *Expert Systems with Applications*, *38*(4), 4642-4648.
Zoulfaghari, H., Hamadani, A. Z., & Ardakan, M. A. (2014). Bi-objective redundancy allocation problem for a system with mixed repairable and non-repairable components. *ISA transactions*, *53*(1), 17-24.