IJE TRANSACTIONS A: Basics Vol. 28, No. 10, (October 2015) 1476-1485

International Journal of Engineering

Journal Homepage: www.ije.ir

TECHNICAL NOTE

Reliability Optimization for Complicated Systems with a Choice of Redundancy Strategies

S. J. Sadjadi*, A. Makui, E. Zangeneh, S. E Mohammadi

Department of Industrial Engineering, Iran University of Science & Technology, Tehran, Iran

PAPER INFO

Paper history:
Received 02 January 2015
Received in revised form 02 June 2015
Accepted 16 October 2015

Keywords:
Reliability Optimization
Redundancy Allocation Problem
Complicated Systems
Redundancy Strategies
Genetic Algorithm

ABSTRACT

Redundancy allocation is one of the common techniques to increase the reliability of the bridge systems. Many studies on the general redundancy allocation problems assume that the redundancy strategy for each subsystem is predetermined and fixed. In general, active redundancy has received more attention in the past. However, in real world, a particular system design contains both active and cold-standby redundancies, and the choice of the redundancy strategy becomes an additional decision variable. So, the problem is to select redundancy level for each subsystem, component and the best redundancy strategy in order to maximize the system reliability under system-level constraints. This paper presents a new mathematical model for redundancy allocation problem (RAP) for the bridge systems when the redundancy strategy can be selected for individual subsystems. The problem is classified as an NP-hard problem. In this paper, a special version of genetic algorithm (GA) is applied, which has been modified for constrained integer nonlinear problems. Finally, computational results for a typical scenario are presented.

***doi**: 10.5829/idosi.ije.2015.28.10a.11*

1. INTRODUCTION

The main goal of reliability engineering is to optimize the reliability of the system. Bridge system is one of complex configurations which can be used in many areas such as manufacturing systems, computer and electronics, aerospace, telecommunication systems, etc. There are two principle ways to improve the system reliability: Enhancing reliability of the system components and preparing the redundant components in parallel. The latter is called redundancy allocation and is the prevalent one. The redundancy allocation problem (RAP) is simultaneous selection of components, and a system-level design configuration which can collectively satisfy all design constraints in order to optimize some objective functions such as system cost and/or reliability [1]. Fyffe [2] is believed to be the first who introduced RAP and it has been an interesting research topic for more than four decades. Chern et al. [3] demonstrated that RAP is an NP-hard problem. Many models and solution methods have been proposed for this problem.

Mainly, studies on RAP can be categorized in two directions, namely, Mathematical modeling and Solution algorithms. RAP models include variety of system combinations (such as series, parallel, series-parallel, k-out-of-n , complex, etc), different redundancy strategies (active, standby) [4, 5], with various objective functions such as maximizing the reliability of the system, minimizing cost, weight, entropy [6] and so on, as single, bi or multiple objectives with some system level constraints [7]. Binary or multi state component is another aspect of formulating the RAP [8, 9]. For researchers, uncertainty is one of the main concerns in optimization problems and many approaches have been developed to tackle it such as fuzzy, probability or robust optimization methods. Robust optimization is a powerful approach to dispel concerns on uncertainty. There are studies that consider uncertainty in RAP using robust optimization [10, 11]. Recently, there is an attempt to embed discount and ordering policies into RAP [12] and another one, to combine RAP and Supplier Selection

*Corresponding Author's Email: *sjsadjadi@iust.ac.ir* (S. J.sadjadi)

Problem to cope with needs of today's competitive world [13].

Solution methods for RAP include mathematical programming, heuristic and meta-heuristic methods and overview of these methods have been addressed by Prasad and Kuo [14]. Instances for mathematical programming include integer programming [15], dynamic programming [1, 15, 16], branch and bound [17, 18], column generation [19], etc. Instances of heuristics and meta-heuristic approaches include genetic algorithm as the most used method [2, 20-22], ant colony [23], taboo search [24], variable neighborhood search [25], variable neighborhood descend algorithm [26], honey bee mating algorithm [27], partial swarm optimization (PSO) [28], etc.

Even, muli-objective form of RAP has been considered and solved with metaheuristics. For instance, Azizmohammadi et al. [29], proposed a new hybrid multi-objective imperialist competition algorithm (HMOICA) based on imperialist competitive algorithm (ICA) and genetic algorithm (GA) in multi-objective redundancy allocation problems. They consider system reliability be maximized while cost and volume of the system are minimized.

In general, most articles consider only active redundant component as predefined assumption in RAP. Some of them assume components to have cold-standby redundancy. Coit [30] presented a new formulation for RAP as subsystem components by considering either active or cold-standby. However, redundancy strategy for each subsystem remains predetermined. Coit [4] also presented an optimal solution to RAP with additional decision variable that determines the best redundancy strategy for each subsystem when there are some subsystems with active redundancy and others with cold-standby redundancy. Coit [4] applied a zero-one integer programming for it and solved some typical test problems. Tavakkoli-moghaddam et al. [2] developed a genetic algorithm for solving Coit [4] problem and compared its result to optimal results obtained by Coit [4]. Chambari et al. [31] presented an efficient simulated annealing algorithm for Coit [4] problem and compared its results with Coit [4] and Tavakkoli-moghaddam et al. [2]. Sadjadi et al. [32] proposed a nonlinear model for multi-objective form of RAP with choice of redundancy and solved it with compromised programming approach. Soltani et al. [33] applied interval programming for RAP with choice of redundancy strategy under uncertainty and Erlang time to failure distribution. For more information about this topic, the readers can refer to state of the art survey on reliability optimization that recently been published [34].

In this paper, we formulate RAP for bridge complex system that contains five subsystems with parallel components, as depicted in Figure 1, and present a constrained integer nonlinear programming (CINLP) model for it. Both active and cold-standby redundancy are allowed in the system. The objective is to determine optimal number of components, type of component choices and redundancy strategy for each subsystem as reliability of bridge complex system be maximized, while budget, weight and other system constraints are met. For solving this model, MI-LXPM[2] is used. This is a version of genetic algorithm (GA) with special mutation, crossover, and selection operators that are suitable for CINLP models.

The reminder of this paper is structured as follows: section 2 describes the problem under study and the related mathematical model. Applied solution algorithm and its features is presented in section 3, while sensitivity analysis and experimental results are presented in section 4. Finally, the paper is concluded in section 5.

2. PROBLEM DESCRIPTION

2. 1. General Description of Problem and Assumptions

In this paper, a complex bridge system is considered that contains five subsystems with parallel components under some constraints such as cost, weight, etc. This combination is presented in Figure 1. One of the important decision variables is the choice of redundancy strategy (active or cold-standby). Tow case is considered in cold standby redundancy strategy. In case 1, system performance is continually monitored by the failure detection and switching hardware (or software) to detect the failure and to activate the redundant component. For this case, switch failure can occur at any time and switch reliability, $\rho_i(t)$, is a non-increasing function of time (and not the number of required switches).

In case 2, when the switch is required, failure can be possible. At any time the switch is required, there is a constant probability, ρ_i, that the switching will be successful [4]. The probability of system failure caused by detection and switching in response to the x^{th} component failure can be considered as a geometric random variable with probability mass function of $p_i^{x-1}(1-p_i)$ [30]. The two cases are designated as S1 and S2.

The system and subsystem do not fail as consequence of a switch failure, unless the switch is actually required subsequently during the mission life. For example, if an initial operating component does not fail during its life, then the subsystem will successfully operate even if the switch has failed [4].

The goal is to design a complex bridge system with maximum reliability to meet the constraints. In addition, following assumptions are made:

[2] Mix Integer – Laplace crossover Power Mutation

- Two redundancy strategies (i.e., active and cold-standby) are considered.
- There are only two states for each element: good and fail.
- Failures of components are statistically independent.
- The failed components do not damage the system and are not repaired.
- Mixing of components is not allowed but there are multiple choices for each subsystem component.
- There is no limitation on supply of components.

There is imperfect switching for the cold standby redundancy strategy.

2. 2. Problem Formulation In [35] the bridge network system as shown in Figure 2, was considered. The system has five components, each having component reliability $R_i,\ i = 1,2,\ldots,5$.

The reliability of the system (**R**), which is the probability of success of the system, is given by:

$$\begin{aligned}\boldsymbol{R} &= R_1R_4 + R_2R_5 + R_2R_3R_4 + R_1R_3R_5 + 2R_1R_2R_3R_4R_5 \\ &-R_1R_2R_4R_5 - R_1R_2R_3R_4 \\ &-R_1R_3R_4R_5 - R_2R_3R_4R_5 - R_1R_2R_3R_5\end{aligned} \tag{1}$$

So, **R** is a nonlinear function.

In this paper, each component in Figure 2 is considered as a subsystem with parallel component itself. So, the resulting combination is same as Figure 1. Each subsystem can apply two types of redundancy strategies for its components, active or cold-standby, but only one of them could be used at the same time; i.e. at a special time, all components of each subsystem would be either active, or cold-standby. If components of subsystem i have active redundancy, the reliability of the subsystem is given by:

$$R_i = 1-\left(1-r_{i,z_i}(t)\right)^{N_i}, \tag{2}$$

and, if components of subsystem i have cold-standby redundancy, the reliability of the subsystem is given by:

$$R_i = r_{i,z_i}(t) + \delta_j(t,j) \times exp\left(-\lambda_{i,z_i}t\right) \times \sum_{l=\alpha_{i,z_i}}^{\alpha_{i,z_i}N_i-1} \frac{\left(\lambda_{i,z_i}t\right)^l}{l!}, \tag{3}$$

where approximated by Coit [4]. In (2) and (3), $r_{i,z_i}(t)$ is:

$$r_{i,z_i}(t) = \exp\left(-\lambda_{i,z_i}t\right) \sum_{l=0}^{\alpha_{i,z_i}-1} \frac{\left(\lambda_{i,z_i}t\right)^l}{l!}, \tag{4}$$

That is, (4) is the reliability function of a component when the time-to-failure for the component follows Erlang (Gamma) distribution with λ and α as scale and shape parameters respectively. So, the reliability of subsystem i in general is given by:

$$R_i = \sum_{z_1=1}^{m_i} \left[\begin{array}{l} \left[1-\left(1-r_{i,z_i}(t)\right)^{N_i}\right]^{\Omega_{Ai}} \times \\ \left[\begin{array}{l} r_{i,z_i}(t) + \delta_j(t,j) \times \\ \exp\left(-\lambda_{i,z_i}t\right) \sum_{l=\alpha_{i,z_i}}^{\alpha_{i,z_i}N_i-1} \frac{\left(-\lambda_{i,z_i}t\right)^l}{l!} \end{array} \right]^{\Omega_{CSi}} \end{array} \right] y_{i,z_i} \tag{5}$$

where $r_{i,z_i}(t)$ is (4), Ω_{Ai} and Ω_{CSi} are zero-one variables that determine redundancy strategy. $y_{i,z_i},\ z_i = 1,2,\ldots,m_i$ are zero-one variables that determine type of component choice when there are m_i component choices.

So, the mathematical model can be presented in following section.

2. 3. Mathematical Model The mathematical model of the bridge redundant reliability system which includes five subsystems with parallel components in each subsystem is presented as the following integer nonlinear programming problem. This model includes two nonlinear constraints and ten linear constraints. In this model components within the same subsystem are of the same type.

maximize

$$\begin{aligned} R(t;\boldsymbol{z},\boldsymbol{n}) = &\prod_{i=1,4}\sum_{z_i=1}^{m_i} \left[\begin{array}{l} \left[1-\left(1-r_{i,z_i}(t)\right)^{N_i}\right]^{\Omega_{Ai}} \\ \times\left[r_{i,z_i}(t) + \delta_i(\mathrm{t,i}) \times \exp\left(-\lambda_{i,z_i}t\right) \sum_{l=\alpha_{i,z_i}}^{\alpha_{i,z_i}N_i-1} \frac{\left(\lambda_{i,z_i}t\right)^l}{l!} \right]^{\Omega_{CSi}} \end{array} \right] y_{i,z_i} + \ldots - \\ &\prod_{i=1}^{5}\sum_{z_i=1}^{m_i} \left[\begin{array}{l} \left[1-\left(1-r_{i,z_i}(t)\right)^{N_i}\right]^{\Omega_{Ai}} \\ \times\left[r_{i,z_i}(t) + \delta_i(\mathrm{t,i}) \times \exp\left(-\lambda_{i,z_i}t\right) \sum_{l=\alpha_{i,z_i}}^{\alpha_{i,z_i}N_i-1} \frac{\left(\lambda_{i,z_i}t\right)^l}{l!} \right]^{\Omega_{CSi}} \end{array} \right] y_{i,z_i} \end{aligned} \tag{6}$$

Subject to

$$\sum_{i=1}^{5} N_i \left[\sum_{z_i=1}^{m_i} C_{i,z_i} \times y_{i,z_i} \right] \le B \tag{7}$$

$$\sum_{i=1}^{5} N_i \left[\sum_{z_i=1}^{m_i} w_{i,z_i} \times y_{i,z_i} \right] \le W \tag{8}$$

$$\sum_{z_i=1}^{m_i} y_{i,z_i} = 1 \qquad \text{for } i = 1,\ldots,5 \tag{9}$$

$$\Omega_{Ai} + \Omega_{CSi} = 1 \qquad \text{for } i = 1,\ldots,5 \tag{10}$$

$$N_i \quad integer \qquad \text{for } i = 1,\ldots,5 \tag{11}$$

$$\Omega_{Ai}, \Omega_{CSi}, y_{i,z_i} \in \{0,1\} \quad \text{for } i = 1,\ldots,5 \text{ and } z_i = 1,\ldots,m_i \tag{12}$$

where in R_i is (5) and r_{i,z_i} is (4). In the above model, with substitution of (4) in (1), the Equation (6) is resulted and maximizes reliability of the bridge system, determines redundancy strategy, component type and quantity of components in each subsystem as objective function. Constraints (7) and (8), address the available cost and weight, respectively. Constraint set (9) determines the type of component choices for each subsystem. Constraint set (10) chooses optimal redundancy strategy for each subsystem. Constraint sets (11) and (12) are integer and zero-one restrictions for variables.

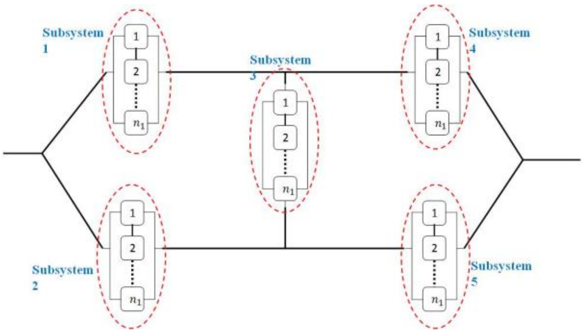

Figure 1. Complex bridge system with five subsystems

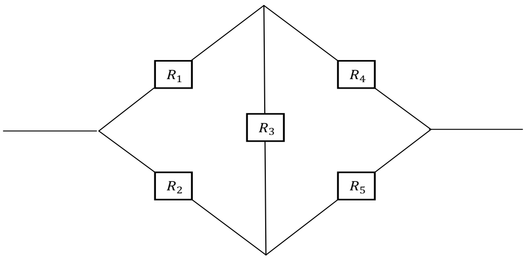

Figure 2. Complex bridge network system

3. SOLUTION APPROACH

3. 1. Genetic Algorithm Genetic algorithm (GA) is one of the probabilistic search methods for solving optimization problems. Holland [36] was the first person who develop GA and applied it for combinatorial problems. In a GA, chromosomes (individuals), a population of solutions, are evolved over successive generations using a set of genetic operators called mutation, crossover and selection operators. First of all, based on some criteria, every chromosome is assigned a fitness value, and then a selection operator is applied to choose relatively 'fit' chromosome to be part of the reproduction process [37]. In reproduction process, new individuals are created using crossover and mutation operators. The genetic information is blended between chromosomes by crossover operator to explore the search space, whereas mutation operator maintains adequate diversity in the population of chromosomes to avoid premature convergence [38]. For more information on applications of GAs to reliability optimization problem, you can refer to Gen et al. [39]. In this paper, for solving this problem, a version of GA with special selection, mutation and crossover operators is applied that is called MI-LXPM algorithm and has been proposed by Deep [38]. MI-LXPM is proposed for solving integer and mix integer constrained problems efficiently, and in fact, it is an expansion of LXPM algorithm. In MI-LXPM, Power mutation and Laplace crossover were expanded to consider integer variables. In addition, for compensation of integer restriction on decision variables a truncation procedure has been applied. Constraint handling is considered through a 'parameter free' penalty approach in MI-LXPM algorithm. More details of these operators are brought in subsequent subsections.

3. 2. Solution Encoding The number of subsystem components N_i, redundancy strategies, and type of selected component determine solution (phenotype) to this problem. In each subsystem, components can be chosen in one type amongst the m_i available components. The solution encoding is a vector that contains three parts. Figure 4 shows this solution encoding. The first part, or part 1, includes five genes that each of them shows the number of component in corresponding subsystem, respectively from subsystem 1 to subsystem 5. The second part, or part 2, includes another five genes, from gen 1 to gen 5 corresponding to subsystem1 to subsystem 5: in each of them, A/CS shows the active or cold-Standby strategy for related subsystem. As stated earlier, there are multiple component choices for each subsystem, the third part, or part 3, determines the type of selected components. All genes in this part will have only 0 or 1 value. For example, if there is 3 component choices for subsystem 1, and in the first triple genes only gen 2 has value of 1 and the two others have value of 0, it means that component type 2 is selected for subsystem 1. The number of cells in part 3 depends on number of available component choices for subsystems and varies from one case to another. Figure 3 shows an example of encoding solution with 4 component choices for subsystems 1, and 2 components choice for subsystems 2, 3, 4, and 5. This solution represents a prospective solution with nine of the third component used in parallel with Cold-standby redundancy for the first subsystem; five of the first components used in parallel with active redundancy for the second subsystem, etc.

3. 3. Initial Population Each cell of the chromosome has a discrete random value which is selected regarding to its bounds and 10 times of variable numbers is considered as population size. Then, initial population is created in randomly manner.

3. 4. Laplace Crossover In the genetic algorithm, crossover operators such as Laplace Crossover, combine two parents to generate new offspring for the next generation. Recently a new parameter was added to the laplace crossover which is allowed to this operator for participating integer variables in offspring generation process. Working of the extended laplace crossover is described below. Two parents, $x^1=\left(x_1^1,x_2^1,\ldots,x_n^1\right)$ and $x^2=\left(x_1^2,x_2^2,\ldots,x_n^2\right)$ combines with each other and create two offsprings, $y^1=\left(y_1^1,y_2^1,\ldots,y_n^1\right)$ and $y^2=\left(y_1^2,y_2^2,\ldots,y_n^2\right)$. First, a random number β_i which satisfies the laplace distribution is generated as:

$$\beta_i=\begin{cases} a-b\log(u_i), r_i\le 1/2;\\ a+blog(u_i), r_i>1/2;\end{cases} \tag{13}$$

where a is location parameter and $b>0$ is scaling parameter and $u_i, r_i\in[0,1]$ are uniform random numbers. For integer variables, $b=b_{int}$, otherwise $b=b_{real}$, i.e., scaling parameter (b) is different for integer and real cases. Now two offsprings can be produced as under:

$$\begin{aligned} y_i^1&=x_i^1+\beta_i\left|x_i^1-x_i^2\right|,\\ y_i^2&=x_i^2+\beta_i\left|x_i^1-x_i^2\right|.\end{aligned} \tag{14}$$

3. 5. Power Mutation The mutation operators produce random changes on some chromosomes in order to jump outside local optima and protect from premature intensification. The Power mutation is based on power distribution and recently a new parameter is added to this operator for decision variables that have a restriction to be integer. The extended Power mutation works as follows: A parent solution $\bar{x}$ is changed and muted solution x is produced as under:

$$x=\begin{cases}\bar{x}-s\left(\bar{x}-x^l\right), t<r;\\ \bar{x}+s\left(x^u-\bar{x}\right), t\ge r;\end{cases} \tag{15}$$

where s is a random number which follows the power distribution $s=(s_1)^p$, s_1 and r a uniform random number between 0 and 1, and p the index of mutation and governs the strength of perturbation of power mutation. $t=\frac{\bar{x}-x^l}{x^u-\bar{x}}$, x^l and x^u are the lower and upper bounds on the value of the decision variable and depend on integer or real nature of the variables, $p=p_{real}$ or $p=p_{int}$ can be used.

3. 6. Selection Selection operator inserts individuals into mating pool in Genetic algorithms. Individuals from the mating pool breed new offsprings, and the next generation is based on newcomers. In MI-LXPM algorithm, tournament selection operator is used as reproduction operator. In this operator, k solutions (k is tournament size) compete with each other in k tournaments and the best ones are chosen and placed in the mating pool. This process is repeated and the better solutions are placed in another section in the mating pool. The best solution in a population can win all the k tournaments, and similarly, the worst solution may lose in all the k tournaments and are eliminated from the population. The user specifies the size of the tournament set as a percentage of the total population. In this study, tournament selection operator with tournament size three is used.

9	5	6	6	14	CS	A	CS	A	A	0	0	1	0	1	0	0	1	0	1	1	0

Figure 3. Encoding solution as a chromosome representation

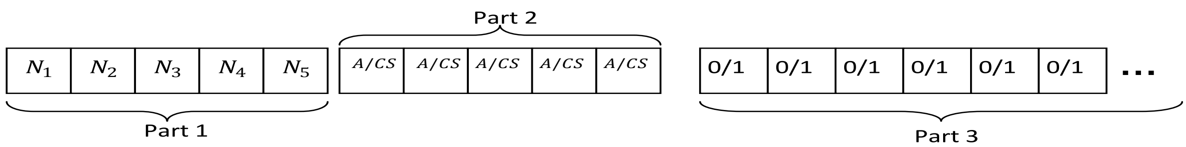

Figure 4. Solution encode

3. 7. Truncation Procedure for Integer Restrictions In MI-LXPM, the integer restrictions are satisfied by truncation procedure as follows: Namely, for $\forall i \in I, x_i$ is truncated to integer value $\bar{x}_i$ by the rule:

- if x_i is integer then $\bar{x}_i = x_i$, otherwise
- $\bar{x}_i$ is equal to either $[\bar{x}_i]$ or $[\bar{x}_i]+1$ each with probability 0.5, ($[x_i]$ is the integer part of x_i).

This guarantees greater randomness in the set of solutions being created and avoids from generating same integer values, whenever a real value lying between the same two consecutive integers is truncated.

3. 8. Constraint Handling Approach Constraint handling is one of the important challenges in optimization. Parameter free, penalty function approach based on feasibility approach proposed by Deb [40] is used in MI-LXPM algorithm. Fitness value, fitness (X_i) of and i^{th} individual is evaluated as:

$$fitness\,(X_i) = \begin{cases} f\,(X_i), & \text{if } X_i \text{ feasible} \\ f_{worst} + \sum_{j=1}^{m} |\phi_j\,(X_i)| & \text{otherwise} \end{cases} \tag{16}$$

where f_{worst} is the objective function value of the worst feasible solution currently available in the population. Thus, the fitness of an infeasible solution not only depends on the amount of constraint violation, but also on the population of solutions at hand. However, the fitness of a feasible solution is always fixed and is equal to its objective function value. $\phi_j(X_i)$, is associated with value of the left hand side of the inequality constraints (equality constraints are also transformed to inequality constraints using a tolerance). If there are no feasible solutions in the population, then f_{worst} is set to zero. The use of constraint violation in the comparisons aim to push infeasible solutions towards the feasible region (In a real life optimization problem, the constraints are often non-commensurable, i.e., they are expressed in different units. Therefore, constraints are normalized to avoid any sort of bias).

Computational steps of the proposed MI-LXPM algorithm are [29]:

1) Generate a suitably large initial set of random points within the domain prescribed by the bounds on variable i.e., points satisfying $x_i^L \le x_i \le x_i^U, i = 1,2,\ldots,n$ for variables which are to have real values and $y_i^L \le y_i \le y_i^U, y_i$ integer for variables which are to have integer values.
2) Check the stopping criteria. If satisfied stop; else go to 3.
3) Apply tournament selection procedure on initial (old) population to make mating pool.
4) Apply Laplace crossover and power mutation to all individuals in mating pool, with probability of crossover (P_c) and probability of mutation (P_m), respectively, to make new population.
5) Apply integer restrictions on decision variables where necessary and evaluate their fitness values.
6) Increase generation++; old population← new population; go to 2.

4. NUMERICAL EXAMPLES AND SENSITIVITY ANALYSIS

4. 1. Sensitivity Analysis on Gamma Distribution Parameter In this paper, time-to-failure for available components are distributed according to k-Erlang distribution with parameters (λ_{ij}) and (k_{ij}), and reliability of a subsystem is considered as (5). A sensitivity analysis on (λ_{ij}), as scale parameter, has been executed. It is considered that λ_{ij} ranges from 0.001 to 0.25, and k=3, t=100h, N=5 , δ=0.99 are other required elements in (5). It is assumed that only one component choice be available. Figure 5 shows the reliability of a subsystem based on different λ_{ij} . As shown in this figure, reliability of a subsystem, and consequently reliability of total system, is highly dependent on λ parameter, especially where λ is very low, in both redundancy strategy.

4. 2. Numerical Example In this section, we took a typical example from [4] and made some slight changes. First, parameter λ in typical example is 10 times less than λ in our example. Second, in typical example the series-parallel system is connected by 14 parallel subsystems, while in our example, the problem has only five subsystems in bridge combination. We select subsystems 1 to 5 from typical example and use their data in our example. Each subsystem has three or four components of choice. Component cost, weight and other parameter are given in Table 1. The Objective is maximizing the system reliability at t=100h, given the constraints for the system cost (B=130) and the system weight (W=170). For each subsystem, active or cold standby redundancy strategy can be used, the switch operates as first case (S1), and the reliability of the switch (at 100 h) is 0.99 for all subsystems.

Because of the stochastic nature of GA, this example was solved four times and the best solution amongst the four has been considered as the final solution. The solution is given in Table 2. In the next step, 33 test problems from the literature that are often applied for series-parallel combination were chosen. These test problems introduced by Fyffe [2] and used in many articles, with little changes. A modified version of these 33 test problems has been used to establish a metric for future comparison on this bridge combination case. The data sets of these problems are shown in Table 1 where various weights of the available resource from 159 to 191 have been considered. For each weight, the problem was solved four times and the best solution and its CPU time considered as the final solution. The computational results are shown in Table 3. The standard deviation of each solution has also been calculated and is depicted in Figure 6. MI-LXPM was implemented using MATLAB R2013a on PC with 8GB of RAM, and 3.30 GHz AMD phenom II X6 1100T processor.

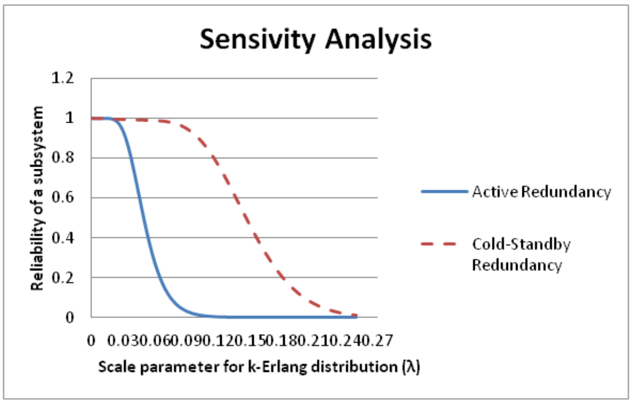

Figure 5. Relationship between reliability of a subsystem and Scale parameter for k-Erlang distribution.

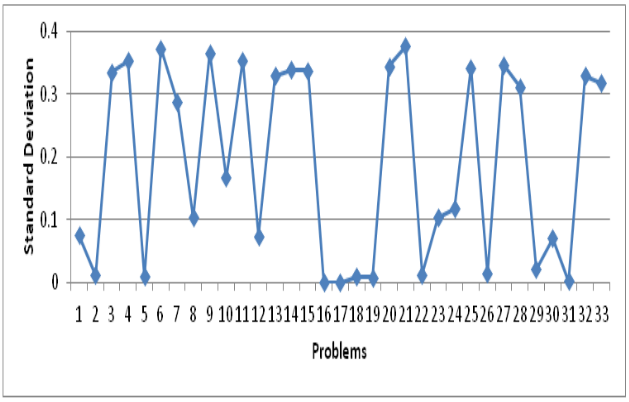

Figure 6. Standard deviation of the fitness function

TABLE 1. Parameter setting for the given problem

i	Choice 1 (j = 1)				Choice 2 (j=2)				Choice 2 (j=2)				Choice 2 (j=2)			
	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	0.0532	2	1	3	0.00726	1	1	4	0.0499	2	2	2	0.0818	3	2	5
2	0.0818	3	2	8	0.00619	1	1	10	0.0431	2	1	9	-	-	-	-
3	0.133	3	2	7	0.11	3	3	5	0.124	3	1	6	0.0466	2	4	4
4	0.0741	2	3	5	0.124	3	4	6	0.0683	2	5	4	-	-	-	-
5	0.0619	1	2	4	0.0431	2	2	3	0.0818	3	3	5	-	-	-	-

TABLE 2. Numerical example results

i	MI-LXPM (GA)		
	N_i	z_i	Redundancy
1	9	2	Active
2	5	1	Cold-Standby
3	3	4	Cold-Standby
4	10	2	Cold-Standby
5	7	2	Active
System reliability		0.9939449	
Resource consumed		Weight	169
		Cost	85

TABLE 3. MI-LXPM performance for problems taken from Fyffe [2]

Problem	Weight constraint	Trial 1				Trial 2				Trial 3				Trial 4				Standard Deviation	Best feasible Solution	CPU Time
		Reliability	Weight	Cost	CPU Time (s)	Reliability	Weight	Cost	CPU Time (s)	Reliability	Weight	Cost	CPU Time (s)	Reliability	Weight	Cost	CPU Time (s)			
1	159	0.9996	146	57	4872	0.8164	150	130	7022	0.989	155	98	5416	0.9792	159	58	5456	0.0752	0.9996	4872
2	160	0.9716	154	73	6467	0.9997	157	62	4383	0.9957	138	69	6466	0.9992	159	82	4806	0.0116	0.9997	4383
3	161	0.9714	159	70	6529	0.9811	128	45	5965	0.2076	161	93	4980	0.9998	160	77	5848	0.3362	0.9986	5848
4	162	0.9715	159	73	6529	0.9715	159	73	6563	0.1659	144	129	6510	0.999	156	59	6893	0.353	0.999	6893
5	163	0.9714	163	76	6552	0.9897	155	82	6377	0.9714	163	76	6494	0.9897	155	82	6360	0.0091	0.9897	6360
6	164	0.9824	163	89	5963	0.2113	163	99	4856	0.98	154	59	6654	0.2591	143	59	6493	0.3734	0.9824	5963
7	165	0.1896	164	74	8026	0.7285	162	130	5451	0.9716	167	95	6445	0.7409	165	130	5873	0.287	0.9716	6444
8	166	0.9716	165	77	6484	0.7245	160	125	5854	0.8622	159	102	5409	0.9783	123	59	6244	0.1031	0.9783	6244
9	167	0.9719	167	79	6409	0.9734	168	81	3621	0.9907	163	47	5532	0.1336	143	106	5794	0.366	0.9907	5532
10	168	0.9997	161	89	5844	0.9942	168	80	5039	0.6133	168	102	6198	0.9998	168	74	4538	0.1665	0.9998	4538
11	169	0.9884	168	92	5836	0.9975	147	61	6433	0.1755	169	126	6345	0.9935	168	100	6059	0.3541	0.9975	6433
12	170	0.9805	167	68	5845	0.9939	169	85	6452	0.8152	163	129	4897	0.9812	163	60	6340	0.0738	0.9939	6452
13	171	0.982	171	84	4090	0.9365	168	115	7041	0.9836	162	59	5711	0.2068	166	88	5425	0.3299	0.9836	5711
14	172	0.9712	171	75	6301	0.6703	171	69	9256	0.1516	150	72	6278	0.99	171	78	6387	0.3388	0.99	6386
15	173	0.9895	169	86	5310	0.6241	173	73	5926	0.981	173	47	6774	0.1636	169	54	6536	0.3375	0.9895	5310
16	174	0.9712	171	75	6215	0.9712	171	75	6244	0.9712	171	75	6225	0.9712	171	75	6310	0	0.9712	6214
17	175	0.9718	163	77	6274	0.9718	163	77	6379	0.9718	163	77	6357	0.9718	163	77	6305	0	0.9718	6274
18	176	0.9721	176	85	6217	0.9721	176	85	6226	0.9721	176	85	6258	0.9937	176	103	5879	0.0094	0.9937	5879
19	177	0.9724	174	81	6071	0.9873	177	69	6136	0.9724	174	81	6154	0.9873	177	69	6179	0.0075	0.9873	6136
20	178	0.1766	144	130	6824	0.9724	174	81	6079	0.9724	174	81	6141	0.9724	174	81	6181	0.3446	0.9724	6078
21	179	0.9908	178	100	5805	0.9724	174	81	6170	0.2252	176	130	6751	0.2279	179	100	4554	0.3776	0.9908	5805
22	180	0.9725	178	83	6075	0.9725	178	83	6112	0.9997	178	120	5872	0.9949	173	94	6206	0.0125	0.9997	5872
23	181	0.9725	178	83	6113	0.7339	181	85	6331	0.9918	179	86	5569	0.86	162	130	6188	0.103	0.9918	5568
24	182	0.9725	178	83	6118	0.7079	62	43	6434	0.9901	178	120	4958	0.9725	173	94	6104	0.1173	0.9901	4958
25	183	0.1955	183	76	7927	0.9907	180	63	5718	0.9727	183	85	6130	0.9965	181	92	5482	0.3427	0.9965	5482
26	184	0.9727	183	85	6574	0.9995	183	84	7788	0.9727	183	85	6089	0.9995	183	84	7761	0.0134	0.9995	7761
27	185	0.1417	185	110	6315	0.9903	182	80	6167	0.6306	185	77	5824	0.99	169	66	6762	0.348	0.9903	6167
28	186	0.9869	183	93	6115	0.9448	186	90	5213	0.2538	185	88	7416	0.989	178	97	4872	0.3122	0.989	4872
29	187	0.9723	186	85	5751	0.9997	179	68	5883	0.9421	183	79	6414	0.9793	181	95	5421	0.0207	0.9997	5883
30	188	0.9725	188	86	5764	0.9810	188	101	5318	0.9863	183	69	6113	08174	160	119	6886	0.0705	0.9863	6113
3	189	0.9845	171	67	6353	0.9811	188	110	6133	0.9849	171	67	7342	0.9813	187	88	5778	0.0018	0.9849	7342
32	190	0.1335	190	102	6215	0.9908	184	75	6168	0.4527	189	58	6133	0.8093	170	128	5905	0.3301	0.9908	6167
33	191	0.9896	161	91	6175	0.2518	182	92	6010	0.971	182	79	5302	0.9998	191	96	5599	0.3184	0.9998	5599

5. CONCLUSION

In this paper we have performed a study on complicated bridge systems and presented a new mathematical model for this type of combination where active/ cold-standby redundancy can be used for individual subsystems. These reliability design problems are usually formulated as a nonlinear integer programming model under a number of constraints. In general, these problems are not easy to solve for real-world case studies, especially for large systems. Therefore, we have applied genetic algorithms (GAs) to effectively determine some near optimal solutions. For establishing a measure for future comparisons, 33 modified typical test problems were solved and the results were presented. For future work on this study, new heuristic and meta-heuristic approaches can be implemented to solve problems and improved efficiency of proposed algorithm. Furthermore, simplifying assumptions considered in this paper such as bi-state components can be replaced with three or more states to make the model more realistic. Considering dependent components in bridge networks can be a major interest in this study.

6. REFERENCES

1. Tavakkoli-Moghaddam, R., Safari, J. and Sassani, F., "Reliability optimization of series-parallel systems with a choice of redundancy strategies using a genetic algorithm", ***Reliability Engineering & System Safety***, Vol. 93, No. 4, (2008), 550-556.
2. Fyffe, D.E., Hines, W.W. and Lee, N.K., "System reliability allocation and a computational algorithm", ***Reliability, IEEE Transactions on***, Vol. 17, No. 2, (1968), 64-69.
3. Chern, M.-S., "On the computational complexity of reliability redundancy allocation in a series system", ***Operations Research Letters***, Vol. 11, No. 5, (1992), 309-315.
4. Coit, D.W., "Maximization of system reliability with a choice of redundancy strategies", ***IIE Transactions***, Vol. 35, No. 6, (2003), 535-543.
5. Coit, D.W. and LIU, J.C., "System reliability optimization with k-out-of-n subsystems", ***International Journal of Reliability, Quality and Safety Engineering***, Vol. 7, No. 02, (2000), 129-142.
6. Soltani, R., Sadjadi, S.J. and Tavakkoli-Moghaddam, R., "Entropy based redundancy allocation in series-parallel systems with choices of a redundancy strategy and component type: A multi-objective model", ***Applied Mathematics***, Vol. 9, No. 2, (2015), 1049-1058.
7. Hsieh, Y.C. and You, P.S., "An effective immune based two-phase approach for the optimal reliability–redundancy allocation problem", ***Applied Mathematics and Computation***, Vol. 218, No. 4, (2011), 1297-1307.
8. Zhigang, T., Zuo, M.J. and Hongzhong, H., "Reliability-redundancy allocation for multi-state series-parallel systems", ***Reliability, IEEE Transactions on***, Vol. 57, No. 2, (2008), 303-310.
9. Pourkarim Guilania, P., Sharifia, M., Niaki, S.T.A. and A., Z., "Redundancy allocation problem of a system with three-state components: A genetic algorithm", ***International Journal of Engineering***, Vol. 27, No. 11 (B), (2014).
10. Soltani, R. and Sadjadi, S.J., "Reliability optimization through robust redundancy allocation models with choice of component type under fuzziness", ***Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability***, Vol., No., (2014), 1748006X14527075.
11. Soltani, R., Sadjadi, S.J. and Tavakkoli-Moghaddam, R., "Robust cold standby redundancy allocation for nonrepairable series–parallel systems through min-max regret formulation and benders' decomposition method", ***Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability***, Vol., No., (2013), 1748006X13514962.
12. Soltani, R., Sadjadi, S.J. and Tofigh, A.A., "A model to enhance the reliability of the serial parallel systems with component mixing", ***Applied Mathematical Modelling***, Vol., No. 0, (2013).
13. Soltani, R., Tofigh, A.A. and Sadjadi, S.J., "Redundancy allocation combined with supplier selection for design of series-parallel systems", ***International Journal of Engineering***, Vol. 28, No. 5, (2015), 730-737.
14. Prasad, V.R. and Way, K., "Reliability optimization of coherent systems", ***Reliability, IEEE Transactions on***, Vol. 49, No. 3, (2000), 323-330.
15. Misra, K.B., "Dynamic programming formulation of the redundancy allocation problem", ***International Journal of Mathematical Education in Science and Technology***, Vol. 2, No. 3, (1971), 207-215.
16. Woodhouse, C.F., "Optimal redundancy allocation by dynamic programming", ***Reliability, IEEE Transactions on***, Vol. R-21, No. 1, (1972), 60-62.
17. Nakagawa, Y., Nakashima, K. and Hattori, Y., "Optimal reliability allocation by branch-and-bound technique", ***Reliability, IEEE Transactions on***, Vol. R-27, No. 1, (1978), 31-38.
18. Sung Chang, S. and Cho Yong, K., "Branch-and-bound redundancy optimization for a series system with multiple-choice constraints", ***Reliability, IEEE Transactions on***, Vol. 48, No. 2, (1999), 108-117.
19. Zia, L. and Coit, D.W., "Redundancy allocation for series-parallel systems using a column generation approach", ***Reliability, IEEE Transactions on***, Vol. 59, No. 4, (2010), 706-717.
20. Sadjadi, S.J. and Soltani, R., "An efficient heuristic versus a robust hybrid meta-heuristic for general framework of serial–parallel redundancy problem", ***Reliability Engineering & System Safety***, Vol. 94, No. 11, (2009), 1703-1710.
21. Coit, D.W. and Smith, A.E., "Penalty guided genetic search for reliability design optimization", ***Computers & Industrial Engineering***, Vol. 30, No. 4, (1996), 895-904.
22. Coit, D.W. and Smith, A.E., "Reliability optimization of series-parallel systems using a genetic algorithm", ***Reliability, IEEE Transactions on***, Vol. 45, No. 2, (1996), 254-260, 266.
23. Yun-Chia, L. and Smith, A.E., "An ant system approach to redundancy allocation", in Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on. Vol. 2, No. Issue, (1999), 1484 Vol. 1482.
24. Kulturel-Konak, S., Smith, A.E. and Coit, D.W., "Efficiently solving the redundancy allocation problem using tabu search", ***IIE Transactions***, Vol. 35, No. 6, (2003), 515-526.
25. Liang, Y.-C. and Chen, Y.-C., "Redundancy allocation of series-parallel systems using a variable neighborhood search algorithm", ***Reliability Engineering & System Safety***, Vol. 92, No. 3, (2007), 323-331.

26. Liang, Y.C., "A variable neighbourhood descent algorithm for the redundancy allocation problem", ***Industrial Engineering & Management Systems*** , Vol. 4, No. 1, (2005), 94-101.

27. Sadjadi, S.J. and Soltani, R., "Alternative design redundancy allocation using an efficient heuristic and a honey bee mating algorithm", ***Expert Systems with Applications***, Vol. 39, No. 1, (2012), 990-999.

28. Beji, N., Jarboui, B., Eddaly, M. and Chabchoub, H., "A hybrid particle swarm optimization algorithm for the redundancy allocation problem", ***Journal of Computational Science***, Vol. 1, No. 3, (2010), 159-167.

29. Azizmohammadi, R., Amiri, M., Tavakkoli-Moghaddam, R. and Mohammadi, M., "Solving a redundancy allocation problem by a hybrid multi-objective imperialist competitive algorithm", ***International Journal of Engineering***, Vol. 26, No. 9 (C), (2013).

30. Coit, D.W., "Cold-standby redundancy optimization for nonrepairable systems", ***IIE Transactions***, Vol. 33, No. 6, (2001), 471-478.

31. Chambari, A., Najafi, A.A., Rahmati, S.H.A. and Karimi, A., "An efficient simulated annealing algorithm for the redundancy allocation problem with a choice of redundancy strategies", ***Reliability Engineering & System Safety***, Vol. 119, No. 0, (2013), 158-164.

32. Sadjadi, S.J., Tofigh, A.A. and Soltani, R., "A new nonlinear multi-objective redundancy allocation model with the choice of redundancy strategy solved by the compromise programming approach", ***International Journal of Engineering***, Vol. 27, No. 4 (A), (2014).

33. Soltani, R., Sadjadi, S.J. and Tavakkoli-Moghaddam, R., "Interval programming for the redundancy allocation with choices of redundancy strategy and component type under uncertainty: Erlang time to failure distribution", ***Applied Mathematics and Computation***, Vol. 244, No., (2014), 413-421.

34. Soltani, R., "Reliability optimization of binary state non-repairable systems: A state of the art survey", ***International Journal of Industrial Engineering Computations***, Vol. 5, No. 3, (2014), 339-364.

35. Mohan, C. and Shanker, K., "Reliability optimization of complex systems using random search technique", ***Microelectronics Reliability***, Vol. 28, No. 4, (1988), 513-518.

36. Holland, J.H., "Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence, U Michigan Press, (1975).

37. Soak, S.-M. and Ahn, B.-H., *New genetic crossover operator for the tsp*, in *Artificial intelligence and soft computing-icaisc 2004*. 2004, Springer. p. 480-485.

38. Deep, K., Singh, K.P., Kansal, M. and Mohan, C., "A real coded genetic algorithm for solving integer and mixed integer optimization problems", ***Applied Mathematics and Computation***, Vol. 212, No. 2, (2009), 505-518.

39. Gen, M. and Cheng, R., "Foundations of genetic algorithms", ***Genetic Algorithms and Engineering Design***, Vol., No., (1997), 1-41.

40. Deb, K., "An efficient constraint handling method for genetic algorithms", ***Computer Methods in Applied Mechanics and Engineering***, Vol. 186, No. 2, (2000), 311-338.

Reliability Optimization for Complicated Systems with a Choice of Redundancy Strategies

TECHNICAL NOTE

S. J. Sadjadi, A. Makui, E. Zangeneh, S. E Mohammadi

Department of Industrial Engineering, Iran University of Science & Technology, Tehran, Iran

P A P E R I N F O

Paper history:
Received 02 January 2015
Received in revised form 02 June 2015
Accepted 16 October 2015

Keywords:
Reliability Optimization
Redundancy Allocation Problem
Complicated Systems
Redundancy Strategies
Genetic Algorithm

چکیده

تخصیص افزونگی یکی از روش‌های رایج برای افزایش قابلیت اطمینان در سیستم‌های پُل است. بسیاری از مطالعات در مورد مسئله تخصیص افزونگی استراتژی افزونگی برای هر زیرسیستم را ثابت و از پیش تعیین شده فرض می‌کنند. به طور معمول، افزونگی فعال بیشتر از سایر استراتژی‌ها در گذشته مورد توجه قرار داشته است. با این حال، در دنیای واقعی، یک سیستم می‌تواند شامل هر دو نوع استراتژی افزونگی فعال و آماده به کار سرد باشد. بنابراین، انتخاب نوع استراتژی افزونگی خود یک متغیر تصمیم است. در نتیجه، مسئله انتخاب میزان افزونگی، نوع مؤلفه به کار رفته و نوع استراتژی افزونگی برای هر زیرسیستم است به نحوی که قابلیت اطمینان سیستم تحت محدودیت‌هایشبیشینه شود. این مقاله یک مدل ریاضی جدید از مسئله تخصیص افزونگی (RAP) در سیستم های پُل ارائه می‌کند که در آن نوع استراتژی افزونگی برای هر زیرسیستم قابل انتخاب است. این مسئله در گروه مسائل با پیچیدگی NP-hard طبقه بندی می‌شود. در این مقاله، نسخه ویژه‌ای از الگوریتم ژنتیک (GA) به کار گرفته می‌شود که برای مسائل غیرخطی عدد صحیح دارای محدودیت توسعه یافته است. درنهایت، نتایج محاسباتی برای یک سناریو معمول ارائه می‌شود.

doi*: 10.5829/idosi.ije.2015.28.10a.11*