Available online at www.elixirpublishers.com (Elixir International Journal)

Mechanical Engineering

Elixir Mech. Engg. 39 C (2011) 4627-4631

ISSN: 2229-712X

Availability redundancy allocation of washing unit in a paper mill utilizing uncertain data

Monica Rani, S.P. Sharma and Harish Garg

Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee -247667, Uttarakhand, India.

ARTICLE INFO

Article history:
Received: 24 July 2011;
Received in revised form:
20 September 2011;
Accepted: 26 September 2011;

Keywords
Artificial bee colony,
Series-parallel system,
Availability optimization,
Redundancy allocation problem.

ABSTRACT

The Availability-Redundancy Allocation Problem (ARAP) is a kind of reliability optimization problems. It involves the selection of components with appropriate levels of redundancy or reliability/availability for maximizing the system availability under some predefined constraints. Identical redundant elements are included in order to achieve a desirable level of availability. Classical mathematical methods fail in handling non-convexities and non-smoothness in optimization problems. This drawback has been removed through meta-heuristics due to their ability of finding an almost global optimal solution in reliability–redundancy optimization problems. Artificial bee colony (ABC) is one of such meta-heuristic algorithm. This paper aims to present an ABC algorithm to search the optimal solution of ARAP with nonlinear resource constraints of a parallel-series system. A washing unit of a paper mill has been taken to illustrate the approach. The experimental results demonstrate that the evolutionary approach can provide more promising solutions in comparison with the widely used single-objective approaches by classical method on parallel-series systems which are frequently studied in the field of reliability optimization.

Introduction

The diversity of system structures and resources' constraints has led to the construction and analysis of several optimization models for reliability improvement. Availability and reliability are good evaluations of a system's performance. Their values depend on the system structure as well as the component availability and reliability. These values decrease as the component ages increase; i.e. their serving times are influenced by their interactions with each other, the applied maintenance policy and their environments [1]. The main requirements for the operation of complex systems are usually specified in terms of cost and availability or equivalently in terms of mean time between failure (MTBF) and/or mean time to repair (MTTR) under a cost constraint. These requirements have to be taken into consideration in the system design stage in order to determine the appropriate reliability and availability of each of the system's components [2].

The importance of designing reliable systems, which normally present high availability, is increasing, due to the engineering requirements of products with better quality and a higher safety level. Improvement of system availability has been the subject of a large volume of research and articles in the area of reliability. There are two ways of increasing the availability of an engineering system: (i) increase the availability of each component or (ii) use redundant components. In the first way, the system availability can be improved to some degree, but the required availability enhancement may never be attainable even though the highest available and reliable components are used. In order to increase the availability of a component, it is possible to work on the improvement of reliability and maintainability. The second way is concerned with choosing the optimal element combination and redundancy-levels. In this way the system availability can be enhanced, but the cost, weight, volume etc will be increased as well. So optimization methods are necessary to determine how many redundancies are necessary in each component or subsystem to maximize availability while taking into account the constraint limits (cost, weight, volume).

Traditional methods, such as the Lagrange multiplier, Dynamic programming etc are inefficient with this kind of problem, because it is necessary to apply complex mathematical fundamentals that make the computational implementation difficult and without flexibility [3]. Some search methods can reach only local optima. However, practical engineering problems may be of large sizes and involve large number of constraints and even multiple-choice components for each subsystem to provide high reliability. Due to the computational difficulty that increases exponentially in terms of problem size, the approximate optimization techniques, or also called heuristics and meta-heuristics, have gained popularity. Recently, the most notable progress in this respect has been in the development of meta-heuristic methods. Heuristic algorithms maintain a trade-off between quality of the obtained solutions and execution time. They approximate the optimal solution rather than finding the mathematical optimum. Meta-heuristics are independent of the type of the problem that is solved and, that is why the word "meta". They offer flexibility and can easily be adapted to solve a wide range of combinatorial optimization problems of practical sizes within reasonable computational time. On the other hand, there are very few works about availability and redundancy allocation. Levitin and Lisnianski [4] have proposed an optimal allocation based on minimizing system cost and considering failure and reparation rates by modifying components replacement frequency and preventive and corrective maintenance policies. Elegbede and Adjallah [2] have developed an availability optimization of series-parallel systems based on genetic algorithms and experience plants. Castro and Cavalca [5] have proposed an

Tele:
E-mail addresses: monicagoyal003@gmail.com

availability allocation based on maintenance policies of series-parallel systems.

The ABC is a new meta-heuristic approach proposed by Karaboga [6]. Basturk and Karaboga [7] compared the performance of ABC algorithm with the performance of GA. Further it is developed by Karaboga and its co-authors in [8-10]. In [11], they had compared the performance of the ABC algorithm with that of genetic algorithm, particle swarm optimization, differential evolution and evolution strategy algorithms on a large set of unconstrained test functions, and concluded that its performance is better than or similar to that of other algorithms although it uses less control parameters and it can be efficiently used for solving multi-modal and multi-dimensional optimization problems. Recently, [12] have shown that the solution of series-parallel problem found by ABC is better than the other meta-heuristic techniques. Because ABCs have the advantages of memory, multi-character, local search and solution improvement mechanism, it is able to discover an excellent optimal solution. Motivated by this, the present paper considers the availability -redundancy allocation problem of a coherent system which is formulated as a non - convex integer non - linear problem in which we maximize the availability of the given system w.r.t. constraint functions associated with system cost. The technique is explained through an example of washing unit of the paper mill (situated in the northern part of India which produces approximately 200 tons of paper per day). The ARAP model of the system is formulated using the uncertain data while uncertainty is removed by considering the required spread on both side of the data and then problem is solved with the help of one of newly meta-heuristic technique namely ABC.

Mathematical formulation

In this section we present the mathematical formulation of the parallel-series system with the used assumptions and notations.

Assumptions

- The supply of components is unlimited.
- All redundancies are active: hazard function is the same when it is in use or not in use.
- Failures of individual components are s-independent.

Notations

The following notations have been used in the entire paper.

m Number of subsystem in the system

n_i Number of components in the subsystem $i,\ 1 \le i \le m$

n The vector of $(n_1, n_2, ..., n_m)$

λ_i Failure rate of each component in subsystem $i,\ 1 \le i \le m$

λ The vector of $(\lambda_1, \lambda_2, ..., \lambda_m)$

τ_i Repair rate of each components in subsystem $i,\ 1 \le i \le m$

τ The vector of $(\tau_1, \tau_2, ..., \tau_m)$

λ_i^l Lower bound of the failure rate of each component in subsystem i

λ_i^u Upper bound of the failure rate of each component in subsystem i

τ_i^l Lower bound of the repair time of each component in subsystem i

τ_i^l Upper bound of the repair time of each component in subsystem i

A_s The system availability

g_i The i^{th} constraint function

M Number of constraints

C_i Cost of the each component in subsystem $i,\ 1 \le i \le m$

C Maximum budget of the system

$n_{\max}$ Maximum number of redundant component allowed in the subsystem i

S Set of feasible solution

$U[a,b]$ Uniform random number between a and b

Description of availability-redundancy allocation problems

The availability-redundancy allocation problems (ARAP) determine the optimal component availabilities and redundancy level of components in a system to maximize the system availability subject to several resource constraints. The ARAP is formulated as follows:

$$\begin{aligned}
Max \quad & A_s = f(\lambda, \tau, n) \\
s.t. \quad & g_j(\lambda, \tau, n) \le 0 \quad j = 1, 2, ..., M \\
& \lambda_i^l \le \lambda_i \le \lambda_i^u, \quad \lambda_i \in R \\
& \tau_i^l \le \tau_i \le \tau_i^u, \quad \tau_i \in R \\
& 1 \le n_i \le n_{\max}, \quad n_i \in Z^+ \\
& i = 1, 2, ..., m
\end{aligned} \tag{1}$$

where $f(\cdot)$ is availability function, $g_j(\cdot)$ is the j^{th} resource constraint usually associated with system weight, volume and cost, $\lambda = (\lambda_1, \lambda_2, ..., \lambda_m)$ is the vector of the subsystem failure rates for the system, $\tau = (\tau_1, \tau_2, ..., \tau_m)$ is the vector of the subsystem repair times for the system, $n = (n_1, n_2, ..., n_m)$ is the vector of the redundancy allocation for the system; λ_i, τ_i and n_i are the failure rate, repair times and the number of components in the i^{th} subsystem respectively and M is the number of constraints. The goal is to determine the number of component and the components' failure rate, repair time of each subsystem so as to maximize the overall system availability. This problem belongs to the category of constrained nonlinear mixed-integer optimization problems because the number of redundancy n_i is the positive integer values and the component failure rates and repair times are the real values.

ARAP formulation

Availability of a system

Consider a series reliability structure consisting of m subsystems whose failure rate and repair time are λ_i, τ_i respectively in which each subsystem $i = 1, 2, ..., m$ consists of n_i identical units in parallel. Redundant units are operating actively in parallel and thus are subject to failures. Each identical unit in subsystem i has mean time between failure is

$MTBF_i$ and mean time to repair is $MTTR_i$ and inherent availability is given by

$$Av_i = \frac{MTBF_i}{MTBF_i + MTTR_i} \quad (2)$$

For the parallel-series systems, the asymptotic availability is given by

$$A_s = \prod_{i=1}^{m}\left[1-(1-Av_i)^{n_i}\right] \quad (3)$$

Expression for cost of the components:

The Manufacturing cost depends on product specification. If MTBF of any component is longer, the failure rate will be lower, indicating that the cost of the component is likely to be higher and consequently the component will be highly reliable. Thus longer MTBF leads to a sharp increase in the manufacturing cost [13]. The MTBF of a component and manufacturing cost are related to each other and the relation can be expressed mathematically as [14]:

$$CMTBF_i = \alpha_i (MTBF)^{\beta_i} + \gamma_i \quad (4)$$

where $CMTBF_i$ and $MTBF_i$ respectively, represent the manufacturing cost and MTBF of the i^{th} component , while α_i, β_i and γ_i are the constants, representing the physical properties of the i^{th} component and β_i is greater than one.

As the failure of any component will reduce the output or even impair the efficiency of the complete system. In an effort to avoid such occurrence, it is necessary to repair the faulty components of the system. It is always intended for recovery as soon as possible in an event of system failure. To facilitate the repair within a limited time frame, experienced staff may be required to work overtime or repair using the state-of-the-art equipments. Assuming a linear relationship between $MTTR_i$ and repairing cost of individual components $CMTTR_i$, with the relation represented mathematically as [15]:

$$CMTTR_i = a_i - b_i \cdot MTTR_i \quad (5)$$

while a_i and b_i are constants depending upon the i^{th} component.

Based on equations (4) and (5), the cost of the i^{th} component can be written as

$$C_i = \sum_{i=1}^{m}(\alpha_i (MTBF)^{\beta_i} + \gamma_i) + \sum_{i=1}^{m}(a_i - b_i MTTR_i) \quad (6)$$

Optimization model

Using the achieved cost and availability of the system, the optimization model is formulated as

$$\begin{aligned} & Max\ A_s = \prod_{i=1}^{m}\left[1-(1-Av_i)^{n_i}\right] \\ & s.t. \quad \sum_{i=1}^{m} C_i\left(n_i + \exp(n_i/4)\right) \le C \qquad (7) \\ & \quad X_k^l \le X_k \le X_k^u \\ & \quad 1 \le n_i \le n_{\max} \qquad n_i \in Z^+ \\ & \quad X = [\lambda_1, \lambda_2, \ldots, \lambda_m, \tau_1, \tau_2, \ldots, \tau_m]^T \\ & \quad i = 1, 2, \ldots, m \end{aligned}$$

where X_k^l and X_k^u are the lower and upper bounds of decision vector X_k which are nothing but the components failure rate and repair time respectively. The factor $\exp(n_i/4)$ accounts for the interconnecting hardware and C is the upper limit on the cost of the system. The optimization model (7) is a constraint non-linear optimization problem.

The main task while solving the constraint optimization problem is to handle the constraints relating to the problem because the feasible solution of this problem is not easy to find due to the presence of both types of constraints in the form of equalities as well as inequalities. So, firstly the constrained problem is converted into the unconstrained one by using penalty-based methods. So the unconstrained function is given as

$$F(x_i) = \begin{cases} f(x_i) & if\ \ x_i \in S \\ f(x_i) - penality(x_i) & if\ \ x_i \notin S \end{cases} \quad (8)$$

The penalty function is usually based on a distance measured to the nearest solution in the feasible region *S* or to the effort to repair the solution. The goal of the algorithm is to adapt the unfeasible antibodies to the feasible antigen(s), so as to reduce the constraint violations of the search for obtaining the optimal or near optimal solutions.

Optimization using ABC algorithms

Inspired from the intelligent for-aging behavior of honey-bee swarm, [7] developed a meta-heuristic technique which is a population based approach and named it as ABC. It has been an effective technique to search for optima of optimization problems. This algorithm is based on swarm intelligence and social insects. A swarm is a group of multi-agent system such as bees, in which simple agents coordinate their activities to solve the complex problems.

This approach is inspired by the intelligent foraging behavior of honey-bee swarm. The colony of artificial bees contains three groups of bees: employed bees, onlooker bees and scouts. A bee waiting on the dance area for making decision to choose a food source, is called an onlooker and a bee going to food source visited by itself previously is named an employed bee. A bee carrying out random search is called a scout. Onlooker bees which watch numerous dances before choosing a food source tend to choose a food source according to the probability proportional to the quality of that food source. Therefore, the good food sources attract more bees than the bad ones. Whenever a bee, whether it is scout or onlooker, finds a food source it becomes employed. Whenever a food source is exploited fully, all the employed bees associated with it abandon it, and may again become scouts or onlookers. Scout bees can be visualized as performing the job of exploration, whereas employed and onlooker bees can be visualized as performing the job of exploitation.

ABC is an iterative algorithm. In this, the first half comprises employed bees, whereas the latter half contains the onlookers. The ABC algorithm assumes that there is only one employed bee for every food source. The employed bee of an abandoned food source becomes a scout and as soon as it finds a new food source it again becomes employed. At the initialization, the ABC generates a randomly distributed population of *N* employed bees solutions representing the food source positions, where *N* is size of the population. Based on these initializations each fitness value can be determined by a fitness function whose value is equal to the value of the objective function.

During iteration, if the fitness amount is better than that of its currently associated food source then that bee moves to this new food source by leaving the old one otherwise it retains its old one. When all bees have finished this process, then these bees share the information with onlookers, each of whom select a food source according to the probability proportional to the fitness amount of that source which is given as

$$p_i = \frac{fit_i}{\sum_{i=1}^{N} fit_i} \tag{9}$$

where fit_i is the fitness of the solution represented by the food source i. After all onlookers have selected their food sources, each of them determines a food source in the neighborhood of his chosen food source and computes its fitness. The best food source among all the neighboring food sources determined by the onlookers associated with a particular food source i itself, will be the new location of the food source i.

Local search for the improvement in solution

Local search process is used to improve the generated solution carried out by onlookers and employed bees. If the fitness of the source is better than that of the present one then the bees moves to the new source. Suppose each solution consists of d parameters and let $X_h = (X_{h1}, X_{h2}, ..., X_{hd})$ be a solution with parameter values $X_{h1}, X_{h2}, ..., X_{hd}$. To determine a solution Z_h in the neighbourhood of X_h, a solution parameter j and another solution $X_k = (X_{k1}, X_{k2}, ..., X_{kd})$ are selected randomly. Except for the value of the selected parameter j, all other parameter values of Z_h are same as X_h, i.e., $Z_h = (X_{h1}, X_{h2}, ..., X_{h(j-1)}, Z_{hj}, X_{h(j+1)}, ..., X_{hd})$. The value Z_{hj} of the selected parameter j in Z_h is determined using the following formula:

$$Z_{hj} = X_{hj} + u(X_{hj} - X_{kj}) \tag{10}$$

where u is an uniform variable in [-1, 1]. If the resulting value falls outside the acceptable range for parameter j, it is set to the corresponding extreme value in that range.

If a particular food source solution does not improve for a predetermined iteration number then a new food source will be searched out by its associated bee and it becomes a scout. In ABC, providing that a position cannot be improved further through a predetermined number of cycles, then that food source is assumed to be abandoned. The value of predetermined number of cycles is an important control parameter of the ABC algorithm, which is called "limit" for abandonment. Assume that the abandoned source is X_h, and $j \in \{1, 2, ..., d\}$ then the scout discovers a new food source to be replaced with X_h as $X_h^j = X_{\min}^j + r_1(X_{\max}^j - X_{\min}^j)$ where r_1 is the random number between 0 and 1. So this randomly generated food source is equally assigned to this scout and changing its status from scout to employed and hence other iteration of ABC algorithm begins until the termination condition is not satisfied.

Numerical Example

The above mentioned technique for solving AAP is illustrated through the problem of optimization of reliability of a washing subunit of a paper mill. The brief description of the system is given below.

System Description

Industrial systems are complex engineering systems [16] arranged in complex configuration, and so it is very difficult to analyze their behavior and predict their failure pattern. They can be arranged in parallel, series and in hybrid configuration. The proposed approach has been demonstrated through the washing unit in a paper mill. The paper mill is one of the large capital oriented engineering systems, comprising of units/subsystems namely, feeding, pulping, washing, screening, bleaching, forming, dryer and press, arranged in predefined configuration. Washing unit [16-17] is an important functionary part of the paper mill which has a dominant role in production of the paper. The Washing of prepared pulp (from the pulping unit) is done in three to four stages, to get it free from blackness and to prepare the fine fibers of the pulp. The considered system consists of four main subsystems, defined as Filter, cleaner, screener and decker all are arranged in series. In here, filter is used to drain black liquor from the cooked pulp. Cleaner is used to clean the pulp by centrifugal action. Screener unit is used to remove oversized, uncooked and odd shaped fibers from pulp through straining action while the function of decker is to reduce the blackness of pulp.

Mathematical model of the system

The failure rates and repair times are given in Table 1 [16-17]. The data as collected from historical records and opinion of field experts is imprecise and vague so it is represented with $\pm 15\%$ spread suggested by system expertise. Based on that the ARAP model is formulated for the given system as follow

$$\begin{aligned}
&Max \quad A_s = \prod_{i=1}^{4}\left[1-(1-Av_i)^{n_i}\right] \\
&\text{where } \; Av_i = \frac{MTBF_i}{MTBF_i + MTTR_i} = \frac{1}{1+\lambda_i \tau_i} \\
&s.t. \quad \sum_{i=1}^{4} C_i\left(n_i + \exp(n_i/4)\right) \le C \\
&\qquad (1-s)X_k \le X_k \le (1+s)X_k \\
&\qquad 1 \le n_i \le 6 \qquad n_i \in Z^+ \\
&\qquad X = [\lambda_1, \lambda_2, ..., \lambda_4, \tau_1, \tau_2, \; ...., \; \tau_4]^T \\
&\qquad i = 1, \; 2, \; ...., \; 4 \\
&\qquad s = 0.15 \quad \text{(considered uncertainity level)}
\end{aligned} \tag{11}$$

where C_i is given by equation (6). The parameters α_i, β_i and γ_i are taken to be U[0.91, 0.94], U[1.95, 1.98] and U[1500,1550] whereas a_i and b_i are assumed to be U[19150,19550] and U[50,65] respectively.

Results

In order to solve the optimization problem (11), the meta-heuristic technique ABC as described above has been used. The particles of the bees uses the variable vectors λ_i, τ_i and n_i. During the evolution process, the integer variable n_i is treated as real variables, and in evaluating the objective functions, the real values are transformed to the nearest integer values. The optimization method is implemented in Matlab (MathWorks) and the program is run on a T6400 @ 2GHz Intel Core(TM) 2 Duo processor with 2GB of Random Access Memory (RAM). In order to eliminate stochastic discrepancy, 15 independent runs are made and involving 15 different initial trial solutions with population size 100. The termination criterion has been set either to a maximum number of 2000 generations or the order of relative error equal to , whichever is achieved first..

Based on these selected parameters, the optimal results of the ABC scheme for the given system are reported in Table 2.

Conclusion

In this paper, the application of ABC to the redundancy allocation problem of a coherent system is formulated as non-convex integer non-linear problem with monotonically increasing objective and constraint functions. The constraint functions are handled with the penalty method and the resulting problem is solved with one of the evolutionary algorithms namely ABC. The optimization problem in redundant component allocation for maximizing the availability of a washing unit of a paper mill with the constraint associated with budget on the redundant component has been solved. The decision variable corresponding to the washing unit of a plant are reported which may be targeted so that optimum system performance could be achieved by using the discussed approach. System reliability engineers/analysts may use these results to set the future targets of their interest.

Acknowledgment

Corresponding author (Monica Rani) and Harish Garg acknowledges the University Grants Commission (UGC), India, New Delhi for all financial support to carry out this work. (UGC), India, New Delhi for all financial support to carry out this work.

References

[1] Samrout, M., Yalaoui, F., Chatelet, E., Chebbo, N.: New methods to minimize the preventive maintenance cost of series-parallel systems using ant colony optimization, Reliability Engineering and System Safety 89, 346-354 (2005).

[2] Elegbede, C., Adjallah, K.: Availability allocation to repairable systems with genetic algorithms: A multi-objective formulation, Reliability Engineering and System Safety 82, 319-330 (2003).

[3] Kuo, W., Prasad, V. R., Tillman, F. A., Hwang, C.,: Optimal Reliability Design - Fundamentals and Applications, Cambridge University Press, Cambridge, UK, 2001.

[4] Levitin, G., Lisnianski, A.: Joint redundancy and maintenance optimization for multistate series-parallel systems, Reliability Engineering and System Safety 64, 33-42 (1999).

[5] Castro, H. F. D., Cavalca, K. L.: Availability optimization with genetic algorithm, International Journal of Quality and Reliability Management 20 (7), 847-863 (2003).

[6] Karaboga, D.: An idea based on honey bee swarm for numerical optimization, Tech. rep., TR06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005).

[7] Basturk, B., Karaboga, D.: An artificial bee colony (ABC) algorithm for numeric function optimization, In IEEE Swarm Intelligence Symposium Indianapolis, Indiana, USA, 2006.

[8] Karaboga, D., Basturk, B.: On the performance of artificial bee colony (abc) algorithm, Applied Soft Computing 8 (1), 687-697 (2008).

[9] Bahriye, A., Karaboga, D.: A modified artificial bee colony algorithm for real-parameter optimization, Information Science doi:10.1016/j.ins.2010.07.015 (2010).

[10] Karaboga, D., Ozturk, C.: A novel clustering approach: Artificial bee colony (abc) algorithm, Applied Soft Computing 11 (1), 652-657 (2011).

[11] Karaboga, D., Akay, B.: A comparative study of artificial bee colony algorithm, Applied Mathematics and Computation 214 (1), 108- 132 (2009).

[12] Yeh, W. C., Hsieh, T. J.: Solving reliability redundancy allocation problems using an artificial bee colony algorithm, Computer and operational research 38 (11), 1465-1473 (2011).

[13] Li, Z. L.: Availability allocation of series-parallel system solved from object-oriented planning, Master's thesis, Feng-Chia University, Taichung, Taiwan (2001).

[14] Tillman, F. A., Hwang, C. L., Kuo, W.: Optimization of Systems Reliability, New York: Marcel Dekker, 1980.

[15] Juang, Y. S., Lin, S. S., Kao, H. P.: A knowledge management system for series-parallel availability optimization and design, Expert Systems with Applications 34, 181-193 (2008).

[16] Kumar, D.: Analysis and optimization of systems availability in sugar, paper and fertilizer industries, Ph.D. thesis, University of Roorkee (Presently IIT Roorkee), India (1991).

[17] Komal, Sharma S.P., Kumar, D.: Complex repairable industrial system reliability analysis using GABLT technique, In: 32th National Systems Conference (NSC-2008), Electrical Engineering Department, IIT Roorkee, 617-622, December 17-19, 2008.

Table 1: Failure rate and repair time data for washing system

Components	Failure rate (λ_i) (failures/hrs)	Repair time (τ_i) (hrs)
Filter	1×10^{-3}	3
Cleaner	3×10^{-3}	2
Screener	5×10^{-3}	3
Decker	5×10^{-3}	3

Table 2: Optimal Result for the washing unit

Subunits	Failure rate	Repair time	n_i
Filter	0.00109849	2.55000000	3
Cleaner	0.00328566	1.71309718	5
Screener	0.00432753	2.87821328	5
Decker	0.00524142	2.55000000	5
System Availability	0.999999977517749		
Slack Cost	114.502983		