Expert Systems with Applications 38 (2011) 3640–3646

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Reliability optimization of a series system with multiple-choice and budget constraints using an efficient ant colony approach

Fardin Ahmadizar [a,*], Hiresh Soltanpanah [b]

[a] Department of Industrial Engineering, University of Kurdistan, Pasdaran Boulvard, Sanandaj, Iran
[b] Department of Industrial Engineering, Islamic Azad University, Sanandaj Branch, Sanandaj, Iran

ARTICLE INFO

Keywords:
Series system
Reliability optimization
Multiple-choice
Budget constraint
Ant colony optimization
Fuzzy sets

ABSTRACT

This paper deals with a reliability optimization problem for a series system with multiple-choice and budget constraints. The objective is to choose one technology for each subsystem in order to maximize the reliability of the whole system subject to the available budget. This problem is NP-hard and could be formulated as a binary integer programming problem with a nonlinear objective function. In this paper, an efficient ant colony optimization (ACO) approach is developed for the problem. In the approach, a solution is generated by an ant based on both pheromone trails modified by previous ants and heuristic information considered as a fuzzy set. Constructed solutions are not guaranteed to be feasible; consequently, applying an appropriate procedure, an infeasible solution is replaced by a feasible one. Then, feasible solutions are improved by a local search. The proposed approach is compared with the existing metaheuristic available in the literature. Computational results demonstrate that the approach serves to be a better performance for large problems.

1. Introduction

Reliability is a significant design measure in many industrial environments such as telecommunication systems and manufacturing facilities. The design of such hardware systems, called reliability optimization problem, can usually be based on either maximizing reliability, availability and performance, or minimizing cost. Reliability optimization of a series system has always been a critical matter. Subsystems of a series system are functionally organized such that any failure of each subsystem will cause the failure of the whole system. One of the strategies for increasing the system reliability of these sorts of systems is to use extra units in each subsystem in parallel. In this problem, reliability optimization is concerned with determining the optimal number of redundant units for one component employed in each subsystem. Many algorithms have been developed over the years to solve redundancy allocation problem (e.g. see Chen, 2006; Coit & Smith, 1996; Hsieh, 2003; Ramirez-Marquez & Coit, 2004; Ruan & Sun, 2006; Sung & Lee, 1994; Tavakkoli-Moghaddam et al., 2008; Yeh, 2009; You & Chen, 2005; Zhao & Liu, 2004; Zhao et al., 2007) and in some cases, reliability optimization is concerned with the design of k-out-of-n systems (e.g. see Tan, 2003; Yeh, 2004, 2006). As it is often desired to consider the practical design issue of handling a variety of different component types, this paper deals with a reliability optimization problem with multiple-choice constraints which has not received enough attention.

We consider a series system such that the reliability of the whole system should be maximized subject to multiple-choice and budget constraints. For each subsystem, a range of technologies is available among which only one must be chosen. If there is no constraint in the budget, then the most reliable technologies would be the most favorable. But, the available budget usually is limited and as the more reliable, the more expensive, a strategy is required to identify the optimal combination of technologies. This problem is called the reliability optimization of a series system with multiple-choice and budget constraints. The problem is formulated as a binary integer programming problem with a nonlinear objective function (Ait-Kadi & Nourelfath, 2001; Sung & Cho, 2000), which is equivalent to a knapsack problem with multiple-choice constraints, so that it belongs to the NP-hard class of problems (Garey & Johnson, 1979). Some exact algorithms have been developed to solve such knapsack problems with multiple-choice constraints (Nauss, 1978; Sinha & Zoltners, 1979) or the reliability problem (Sung & Cho, 2000) which are not efficient for large industrial problems because they require a very large amount of computation time to obtain the optimal solution. Therefore, the use of heuristics or metaheuristics is appeared to be necessary to attain optimal or nearly optimal solutions in a little time. Nourelfath and Nahas (2003) have proposed a heuristic approach based on the Hopfield model of neural networks. The approach applies a

* Corresponding author. Tel./fax: +98 871 6660073.
E-mail addresses: f.ahmadizar@uok.ac.ir (F. Ahmadizar), heresh@iausdj.ac.ir (H. Soltanpanah).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.09.018

new model of Hopfield networks, where neurons take quantized values rather than just binary or continuous values. This heuristic is quickly able to obtain optimal or nearly optimal solutions of small problems. The first modern metaheuristic (and the only one based on our knowledge) has been proposed by Nahas and Nourelfath (2005) to solve the problem. In this algorithm, which is an ant system, called AS, a penalty treated in the pheromone trails update is employed for infeasible solutions concerning to the budget constraint. The penalties are proportional to the amount of budget violations. Also, a local search is applied to improve constructed solutions. The AS approach is quickly able to obtain optimal or nearly optimal solutions of large problems.

In this paper, we develop an efficient ant colony system, called ACS, for the problem. Ant colony optimization (ACO) (Dorigo, 1992; Dorigo et al., 1996; Dorigo & Stutzle, 2003) is a metaheuristic developed for solving discrete optimization problems. An ACO algorithm is a population-based approach based on the behavior of real ant colonies using pheromones as a communication medium. Real ants are capable of finding the shortest path from their nest to a food source without using visual cues. In the ACS approach, a solution is generated by applying a pseudo-stochastic rule based on a combination of the previous solutions results and the knowledge related to the problem as two fuzzy sets. The unfeasibility of constructed solutions is removed by replacing an infeasible solution by a feasible one based on a neighborhood search procedure. Each solution is then improved by an interesting local search. A set of large problems is used for evaluating the proposed approach.

The remainder of the paper is organized as follows. The next section gives the problem statement as a binary integer programming problem with a nonlinear objective function. The proposed ant colony approach is described in Section 3. Section 4 provides computational experiments and finally, concluding remarks are given in Section 5.

2. Problem formulation

Consider a series system that includes S different subsystems. For subsystem i, there exist N_i available technologies with different characteristics such as cost and reliability. Let C_{ij} and R_{ij} be, respectively, the cost and reliability of subsystem i when technology j is used. Total available amount of budget is B. The optimization problem is to choose only one technology for each subsystem to maximize reliability of the whole system (R_{sys}) subject to the available budget. In order to formulate the problem in mathematical expression, decision variable x_{ij} is addressed as follows:

$$x_{ij} = \begin{cases} 1, & \text{if subsystem } i \text{ uses technology } j \\ 0, \text{otherwise} & \end{cases}$$

Then, the problem is formulated as the following binary integer programming problem with one nonlinear objective function:

$$\text{Max} \quad R_{sys} = \prod_{i=1}^{S} \left(\sum_{j=1}^{N_i} x_{ij} R_{ij} \right)$$

$$\text{s.t.} \quad \sum_{i=1}^{S} \sum_{j=1}^{N_i} x_{ij} C_{ij} \leqslant B \tag{1}$$

$$\sum_{j=1}^{N_i} x_{ij} = 1, \quad \forall\, i = 1, 2, \ldots, S \tag{2}$$

$$x_{ij} = \{0, 1\}, \quad \forall\, i = 1, 2, \ldots, S,\ j = 1, 2, \ldots, N_i \tag{3}$$

Constraint (1) represents the budget constraint, constraint (2) represents the multiple-choice constraint and constraint (3) defines the decision variables.

3. Proposed ant colony approach

In this paper, an ant colony system (Dorigo & Gambardella, 1997a, 1997b) based approach is developed for solving the reliability problem under consideration. To apply an ACO metaheuristic to a combinatorial optimization problem, it is appropriate to represent the problem by a graph $G = (ŋ, ə)$, where ŋ and ə are, respectively, the nodes and edges. To represent the problem as such a graph, two types of nodes are introduced: the set of nodes $ŋ_1$ containing one element for each subsystem and the set of nodes $ŋ_2$ containing one element for each technology. Furthermore, the edges ə connect subsystems to their available technologies, that is, each node in $ŋ_1$ is connected to each of the corresponding nodes in $ŋ_2$ by an edge. In the proposed approach, an ant starts from the first subsystem and chooses (moves to) one of the available technologies for this subsystem. Then, the ant iteratively moves to the next subsystem and chooses a technology. At each step, a technology is chosen by applying a transition rule so-called pseudo-random proportional rule. Note that the generated solution may be infeasible; because constraints (2) and (3) are guaranteed during the construction process, but the total cost of the chosen technologies may be greater than B.

3.1. General structure of the approach

The general structure of the approach can be represented as follows (the next sections provide the details).

Algorithm 1

Step 1. The pheromone trails and the parameters are set.
Step 2. The following procedures are iterated *Max_iter* (an integer parameter) times:
  Step 2.1. The following actions are iterated *Ant_size* (an integer parameter) times:
    A. A solution is constructed by repeatedly applying the transition rule.
    B. If the solution is infeasible, it is replaced by a feasible one using Algorithm 2.
    C. If it is possible, the solution is improved by Algorithm 3, i.e., the local search procedure.
    D. The pheromone trails related to the chosen technologies are finally modified according to the local updating rule.
  Step 2.2. The pheromone trails are modified according to the global updating rule.
Step 3. The best solution found is printed.

3.2. Pseudo-random transition rule

Artificial ants probabilistically build solutions by iteratively choosing technologies by taking into account both the heuristic information on the problem and the (artificial) pheromone trails which change dynamically at run-time. An ant chooses one of the available technologies to assign to the current subsystem as follows: with probability q_0 an ant k for subsystem i selects the technology j for which the product between the pheromone trail and the heuristic information is maximum, that is,

$$j = \arg\max[\tau_{ij}(\eta_{ij})^{\beta}] \tag{4}$$

where τ_{ij} and η_{ij} are, respectively, the pheromone trail and heuristic information between subsystem i and technology j – denoted by edge (i, j). Also, β is a positive parameter denoting the relative importance of the heuristic information versus the pheromone trail.

While with probability $1 - q_0$, the ant selects a technology j according to the probability distribution given in the following equation:

$$p_{ij}^k = \frac{\tau_{ij}(\eta_{ij})^\beta}{\sum_{l=1}^{N_i} \tau_{il}(\eta_{il})^\beta} \quad (5)$$

As seen, η_{ij} shows the desirability of selecting technology j for subsystem i. Therefore, the heuristic information related to one subsystem can be considered as a fuzzy set. We present two priority rules in order to calculate the heuristic information as follows.

Rule 1: (based on the objective function) If a technology is reliable, it must be chosen with a high desirability.
Rule 2: (based on the budget constraint) If a technology is cheap, it must be chosen with a high desirability.

Let $F_{ij}^{(1)}$ and $F_{ij}^{(2)}$ be, respectively, the priority grade, i.e., the grade of membership, of technology j for choosing in subsystem i related to the sets of reliable (say fuzzy set 1) and cheap (say fuzzy set 2) technologies for this subsystem. According to Rules 1 and 2 and based on both $F_{ij}^{(1)}$ and $F_{ij}^{(2)}$, the heuristic information can be calculated in several ways. In this research, four operators are considered as the aggregation operator:

$$\begin{aligned} &O1)\eta_{ij} = F_{ij}^{(1)} F_{ij}^{(2)} \\ &O2)\eta_{ij} = \text{Average}\left(F_{ij}^{(1)}, F_{ij}^{(2)}\right) \\ &O3)\eta_{ij} = \text{Min}\left(F_{ij}^{(1)}, F_{ij}^{(2)}\right) \\ &O4)\eta_{ij} = \text{Max}\left(F_{ij}^{(1)}, F_{ij}^{(2)}\right) \end{aligned} \quad (6)$$

Moreover, $F_{ij}^{(1)}$ and $F_{ij}^{(2)}$ can be calculated in several manners. In this research, three methods are defined as follows.

In method 1, the relative values of the reliability and the cost of a technology are considered according to the given fuzzy sets as follows:

$$F_{ij}^{(1)} = \frac{R_{ij}}{\sum_{l=1}^{N_i} R_{il}}, \quad F_{ij}^{(2)} = \frac{1/C_{ij}}{\sum_{l=1}^{N_i} 1/C_{il}} \quad (7)$$

While in methods 2 and 3, the reliability and the cost of a technology are indirectly considered. Assume that, based on fuzzy set 1, the available technologies for subsystem i are sequenced in decreasing order of reliability. Let $rank_{ij}^1$ be the rank of technology j which can be between 1 (related to the most reliable technology) and N_i (related to the most unreliable technology). Also, suppose that, based on fuzzy set 2, these technologies are rearranged in increasing order of cost. Let $rank_{ij}^2$ be the rank of technology j which can be between 1 (related to the cheapest technology) and N_i (related to the most expensive technology). Then, according to methods 2 and 3, $F_{ij}^{(1)}$ and $F_{ij}^{(2)}$ are, respectively, formulated as (8) and (9).

$$F_{ij}^{(1)} = \frac{N_i + 1 - rank_{ij}^1}{N_i}, \quad F_{ij}^{(2)} = \frac{N_i + 1 - rank_{ij}^2}{N_i} \quad (8)$$

$$F_{ij}^{(1)} = \frac{1}{rank_{ij}^1}, \quad F_{ij}^{(2)} = \frac{1}{rank_{ij}^2} \quad (9)$$

Note that all of the three methods guarantee that the higher reliability, the greater $F_{ij}^{(1)}$ and the smallest cost, the greater $F_{ij}^{(2)}$ which, respectively, agree with fuzzy sets 1 and 2. In addition, $F_{ij}^{(1)}$ and $F_{ij}^{(2)}$ in (7) are greater than 0 and smaller than 1, while in both (8) and (9) are limited to the interval $[1/N_i,\ 1]$.

3.3. Dealing with infeasibility

As mentioned before, a constructed solution may be infeasible because the total cost of the chosen technologies may be greater than B (that is, the violation of constraint (1)). Thus, we develop a convenient procedure where an infeasible solution is replaced by a feasible one. This mechanism is based on a neighborhood search as follows. (Let TC be the total cost of the current solution.)

Algorithm 2

Step 1. One of the subsystems is chosen; if it is possible, the current technology of this subsystem is replaced as follows: among the available technologies which have smaller cost than the current one, (if the given set is not empty) the most reliable technology is selected.
Step 2. TC is calculated. If TC is not greater than B, the procedure is terminated, and otherwise, it is repeated from Step 1.

In Step 1, a subsystem can be chosen in a random way or in a purposeful manner based on cost (that is, the subsystem with the greatest cost).

3.4. Local search

When an ant colony algorithm is coupled with a local search procedure, the performance could be greatly improved (Dorigo & Stutzle, 2003). A local search is performed based on a neighborhood search in order to find a better solution. Therefore, if the entire available budget is used by a solution, it may not be improved by a local search which is not very deep. It is noteworthy that if a technology is more reliable than another, it also has greater cost, and otherwise, the last one will not be chosen in the optimal solution of the problem and hence, it should be eliminated from the list of available technologies in the beginning – this proposition has been proved in Sung and Cho (2000). In order to achieve the best performance, the following local search procedure is developed to improve each solution which has not used the entire available budget.

Algorithm 3

Step 1. One of the subsystems is chosen (say i); if it is possible, the current technology of subsystem i (let us say technology j) is replaced as follows: among the available technologies which have higher reliability than technology j and which $C_{i\cdot} - C_{ij} \leqslant B - TC$, (if the given set is not empty) the most reliable technology is selected.
Step 2. TC is calculated. If TC is smaller than B, the procedure is continued from the next step, and otherwise, i.e., if $TC = B$, it is terminated.
Step 3. One of the subsystems is chosen (say i). The available technologies which have higher reliability than the current one (say technology j) and which $C_i - C_{ij} \leqslant B - TC$ are specified. If this set is empty, the procedure is terminated, and otherwise, technology j is replaced by the most reliable one among the specified technologies.
Step 4. TC is calculated. If TC is smaller than B, the procedure is continued from Step 1, and otherwise, i.e., if $TC = B$, it is terminated.

In Steps 1 and 3, a subsystem can be selected in a random way or in a purposeful manner based on reliability (that is, the subsystem with the lowest reliability).

3.5. Local updating of the pheromone trails

While constructing a solution, an ant changes the pheromone intensity on each edge (i, j) related to its chosen technologies by applying the local updating rule as follows:

Table 1
Data for example 5.

Subsystem		Tech 1	Tech 2	Tech 3	Tech 4	Tech 5	Tech 6	Tech 7	Tech 8
1	Reliability	0.9	0.99	0.999	0.9999	0.99999	0.999999	0.9999999	0.99999999
	Cost ($)	20	40	60	80	100	120	140	180
2	Reliability	0.85	0.9775	0.9966	0.9995	0.9999	–	–	–
	Cost ($)	30	60	90	120	150			
3	Reliability	0.8	0.96	0.99	0.998	0.9997	0.9999	0.99999	0.999999
	Cost ($)	20	40	60	80	100	120	140	160
4	Reliability	0.75	0.938	0.98	0.999	0.9999	–	–	–
	Cost ($)	30	40	50	60	70			
5	Reliability	0.85	0.99	0.999	0.9999	0.99998	0.999998	0.9999998	0.99999998
	Cost ($)	20	40	65	80	100	120	140	155
6	Reliability	0.9	0.95	0.999	0.9999	0.99999	–	–	–
	Cost ($)	25	30	50	70	90			
7	Reliability	0.95	0.99	0.997	0.9997	0.99997	0.999997	0.9999997	0.99999997
	Cost ($)	40	60	80	100	120	140	160	180
8	Reliability	0.85	0.995	0.999	0.9999	0.99999	–	–	–
	Cost ($)	10	30	60	80	120			
9	Reliability	0.9	0.95	0.995	0.9995	0.99995	0.999995	0.9999995	0.99999995
	Cost ($)	30	50	70	90	110	130	150	170
10	Reliability	0.99	0.999	0.9999	0.99999	0.999999	0.9999999	–	–
	Cost ($)	15	40	70	100	130	160		
11	Reliability	0.95	0.999	0.9998	0.99999	0.999998	0.9999999	0.99999997	0.999999999
	Cost ($)	20	40	60	80	100	120	140	160
12	Reliability	0.8	0.9	0.99	0.999	0.9999	–	–	–
	Cost ($)	40	60	85	110	130			
13	Reliability	0.75	0.85	0.99	0.999	0.9996	0.99996	0.999996	0.9999996
	Cost ($)	30	50	80	100	120	140	160	180
14	Reliability	0.8	0.95	0.99	0.999	0.9999	–	–	–
	Cost ($)	10	30	40	60	80			
15	Reliability	0.99	0.999	0.9999	0.99999	0.999999	0.9999999	0.99999998	0.999999995
	Cost ($)	50	80	110	140	160	180	200	220
16	Reliability	0.9	0.99	0.999	0.9999	0.99999	0.999999	0.9999999	0.99999999
	Cost ($)	20	40	60	80	100	120	140	180
17	Reliability	0.85	0.9775	0.9966	0.9995	0.9999	–	–	–
	Cost ($)	30	60	90	120	150			
18	Reliability	0.8	0.96	0.99	0.998	0.9997	0.9999	0.99999	0.999999
	Cost ($)	20	40	60	80	100	120	140	160
19	Reliability	0.75	0.938	0.98	0.999	0.9999	–	–	–
	Cost ($)	30	40	50	60	70			
20	Reliability	0.85	0.99	0.999	0.9999	0.99998	0.999998	0.9999998	0.99999998
	Cost ($)	20	40	65	80	100	120	140	155
21	Reliability	0.9	0.95	0.999	0.9999	0.99999	–	–	–
	Cost ($)	25	30	50	70	90			
22	Reliability	0.95	0.99	0.997	0.9997	0.99997	0.999997	0.9999997	0.99999997
	Cost($)	40	60	80	100	120	140	160	180
23	Reliability	0.85	0.995	0.999	0.9999	0.99999	–	–	–
	Cost ($)	10	30	60	80	120			
24	Reliability	0.9	0.95	0.995	0.9995	0.99995	0.999995	0.9999995	0.99999995
	Cost ($)	30	50	70	90	110	130	150	170
25	Reliability	0.99	0.999	0.9999	0.99999	0.999999	0.9999999	–	–
	Cost ($)	15	40	70	100	130	160		
26	Reliability	0.95	0.99	0.999	0.9995	0.99999	0.999995	0.9999999	0.99999998
	Cost ($)	25	35	55	70	95	115	140	160
27	Reliability	0.85	0.97	0.997	0.9995	0.9999	–	–	–
	Cost ($)	40	60	90	120	145			
28	Reliability	0.85	0.96	0.99	0.998	0.9998	0.99995	0.99999	0.999999
	Cost ($)	25	45	60	85	100	125	150	170
29	Reliability	0.8	0.9	0.98	0.998	0.9995	–	–	–
	Cost ($)	30	45	60	70	85			
30	Reliability	0.8	0.98	0.995	0.9995	0.99995	0.999995	0.9999995	0.99999995
	Cost ($)	20	40	60	80	100	120	140	160
31	Reliability	0.85	0.9	0.99	0.999	0.9999	–	–	–
	Cost ($)	20	30	50	70	90			
32	Reliability	0.9	0.97	0.997	0.9997	0.99997	0.999997	0.9999997	0.99999997
	Cost($)	30	50	70	90	110	130	150	170
33	Reliability	0.85	0.95	0.995	0.9995	0.99995	–	–	–
	Cost ($)	15	30	60	85	110			
34	Reliability	0.9	0.95	0.995	0.9995	0.99995	0.999995	0.9999995	0.99999995
	Cost ($)	25	45	65	85	105	125	145	165
35	Reliability	0.95	0.999	0.9995	0.99999	0.999995	0.9999999	–	–
	Cost ($)	20	45	70	100	140	170		
36	Reliability	0.99	0.998	0.9998	0.99998	0.999998	0.9999998	0.99999998	0.999999998
	Cost ($)	30	40	60	80	100	120	140	160
37	Reliability	0.8	0.9	0.99	0.999	0.9999	–	–	–
	Cost ($)	30	50	80	115	130			

(continued on next page)

Table 1 (*continued*)

Subsystem		Tech 1	Tech 2	Tech 3	Tech 4	Tech 5	Tech 6	Tech 7	Tech 8
38	Reliability	0.75	0.85	0.95	0.996	0.9996	0.99996	0.999996	0.9999996
	Cost ($)	20	40	75	100	115	140	155	175
39	Reliability	0.75	0.9	0.99	0.999	0.9995	–	–	–
	Cost($)	15	30	40	60	80			
40	Reliability	0.99	0.999	0.9999	0.99999	0.999999	0.9999999	0.99999998	0.999999995
	Cost ($)	40	70	100	130	160	185	210	225

Tech = Technology.

$$\tau_{ij} = (1-\rho')\tau_{ij} + \rho'\tau_0 \tag{10}$$

where τ_0 is the initial value of the pheromone trails and ρ', a parameter between 0 and 1, is the local pheromone trail evaporation rate. The effect of this updating is to make the desirability of edges change dynamically in order to explore different paths by the next ants in the colony. That is, the technologies in one ant's solution will be chosen with a lower probability in constructing other ants' solutions.

3.6. Global updating of the pheromone trails

Once all ants in the colony have constructed their solutions, the amount of pheromone on each edge (i, j) related to the global-best solution, that is, the best solution constructed so far, is modified by applying the global updating rule as follows:

$$\tau_{ij} = (1-\rho)\tau_{ij} + \rho z(R_{gb}/TC_{gb}) \tag{11}$$

where ρ, a parameter between 0 and 1, is the global pheromone trail evaporation rate, z is a positive parameter and R_{gb} and TC_{gb} are, respectively, the reliability and total cost of the global-best solution. The global updating rule is intended to provide a greater amount of pheromone to better solutions in order to make the search more directed.

4. Computational results

In order to evaluate the approach developed for the given reliability problem, five large examples are used. Examples 1, 2, 3 and 4 presented in Nahas and Nourelfath (2005) are used for comparing ACS with the previous metaheuristic (i.e., AS) and example 5 is considered for an additional evaluation. Example 1 has 15 subsystems and 60 decision variables, example 2 has 15 subsystems and 80 decision variables, example 3 has 15 subsystems and 100 decision variables and example 4 has 25 subsystems and 166 decision variables (for details see Nahas & Nourelfath, 2005). The data for example 5 are shown in Table 1. For this example, which has 40 subsystems and 266 decision variables, the available budget is 2700$ and the search space size is larger than 3.039×10^{32}.

The proposed algorithms have been coded in Visual C++6.0 and run on a Pentium 4, 2 GHz PC with 256 MB memory under Windows XP. In order to test the effect of the local search technique, ACS is considered both without the local search and coupled with it. Furthermore, as the heuristic information can be calculated based on (7)–(9), eight cases are studied as follows:

Case 1: Without the local search and without the heuristic information.
Case 2: Without the local search and with the heuristic information based on (6) and (7).
Case 3: Without the local search and with the heuristic information based on (6) and (8).
Case 4: Without the local search and with the heuristic information based on (6) and (9).
Case 5: With the local search and without the heuristic information.
Case 6: With the local search and with the heuristic information based on (6) and (7).
Case 7: With the local search and with the heuristic information based on (6) and (8).
Case 8: With the local search and with the heuristic information based on (6) and (9).

Note that in order to test the effect of the heuristic information, in Cases 1 and 5 the approach is considered without using the heuristic information, while in the other ones it is employed. When the heuristic information is not used, (4) changes to $j = \arg\max[\tau_{ij}]$ and (5) is also modified as follows:

$$p_{ij}^k = \frac{\tau_{ij}}{\sum_{l=1}^{N_i} \tau_{il}} \tag{12}$$

In the preliminary experiment, some values have been tested for the numeric parameters. Four different values of *Max_iter* (1000, 1500, 2000 and 3000), five different values of *Ant_size* (5, 10, 20, 30 and 50), various values of q_0 (0.75, 0.8, 0.85, 0.9, 0.95 and 0.97), different values of β (0.01, 0.1, 0.5, 1, 1.5 and 2), a range of values of ρ and ρ' (0.01, 0.025, 0.05, 0.075, 0.1 and 0.15) and three different values of z (1, $S/2$ and S) have been considered. We set $\tau_0 = 10^{-6}$. This value should always be chosen so little that $\tau_0 < z(R_{gb}/TC_{gb})$. This issue imposes that (11) causes an increase in the related τ_{ij} (and then (10) causes a decrease in the related τ_{ij}; of course, τ_0 is a lower bound of the pheromone trails). With different combinations of the parameter values, each operator in (6) has been evaluated and the aggregation operator $O3$, the Min operator,

Table 2
Results for example 1.

Case	Time	Reliability			
		Minimum	Average	Std. dev.	Maximum
1	1.53	0.850098	0.855895	0.002072	0.857054
2	1.31	0.857054	0.857054	0	0.857054
3	1.29	0.850098	0.855963	0.002100	0.857054
4	1.3	0.856108	0.856659	0.000427	0.857054
5	3.75	0.856108	0.856959	0.000299	0.857054
6	4.14	0.857054	0.857054	0	0.857054
7	2.69	0.856108	0.856770	0.000457	0.857054
8	4.21	0.857054	0.857054	0	0.857054

Table 3
Results for example 2.

Case	Time	Reliability			
		Minimum	Average	Std. dev.	Maximum
1	2.01	0.915042	0.915042	0	0.915042
2	1.71	0.915042	0.915042	0	0.915042
3	1.7	0.915042	0.915042	0	0.915042
4	1.7	0.915042	0.915042	0	0.915042
5	2.33	0.915042	0.915042	0	0.915042
6	2.44	0.915042	0.915042	0	0.915042
7	2.72	0.915042	0.915042	0	0.915042
8	2.01	0.915042	0.915042	0	0.915042

Table 4
Results for example 3.

Case	Time	Reliability			
		Minimum	Average	Std. dev.	Maximum
1	2.57	0.958163	0.963905	0.002086	0.965134
2	2.17	0.964070	0.965028	0.000336	0.965134
3	2.11	0.964070	0.964708	0.000549	0.965134
4	2.15	0.964070	0.964496	0.000549	0.965134
5	5.23	0.964070	0.965028	0.000336	0.965134
6	5.13	0.965134	0.965134	0	0.965134
7	4.05	0.965134	0.965134	0	0.965134
8	4.34	0.964070	0.964815	0.000514	0.965134

Table 5
Results for example 4.

Case	Time	Reliability			
		Minimum	Average	Std. dev.	Maximum
1	4.27	0.864660	0.865127	0.000402	0.865439
2	3.62	0.864660	0.865127	0.000402	0.865439
3	3.68	0.855926	0.864176	0.002925	0.865439
4	3.63	0.864660	0.865127	0.000402	0.865439
5	4.73	0.865439	0.865439	0	0.865439
6	3.91	0.865439	0.865439	0	0.865439
7	4.04	0.865439	0.865439	0	0.865439
8	3.79	0.865439	0.865439	0	0.865439

Table 6
Results for example 5.

Case	Time	Reliability			
		Minimum	Average	Std. dev.	Maximum
1	6.69	0.910796	0.913103	0.001608	0.914872
2	5.83	0.912396	0.913920	0.001106	0.914895
3	5.82	0.910730	0.913798	0.001793	0.914895
4	5.91	0.910727	0.914063	0.001389	0.914895
5	7.12	0.911551	0.913441	0.001384	0.914895
6	6.15	0.914045	0.914794	0.000263	0.914895
7	5.96	0.912487	0.914553	0.000771	0.914895
8	5.83	0.911639	0.914221	0.001203	0.914895

has been yielded the best results (while the Max operator has been the worst). Therefore, (6) is set according to *O*3. In addition, the randomly selection of a subsystem in Algorithms 2 and 3 has been yielded better performance (the purposefully selections have been yielded an increase in the CPU time without any significant improvement in the objective function). Then, the best performance of ACS has been obtained with $Max\_iter = 2000$, $Ant\_size = 20$, $q_0 = 0.9$, $\beta = 1$, $\rho = \rho' = 0.1$ and $z = S/2$.

To evaluate the given cases, each of the problem instances has been tested for 10 trials. The summarized results of examples 1–5 are, respectively, shown in Tables 2–6, which give comparisons between the eight different cases. The time representing the average computational time is in seconds. As seen, the performances of the cases have been the same in example 2. In view of the objective function, Case 6 has been superior in all of the examples compared to the other cases. Furthermore, when the local search has been applied, the ACS approach has been enhanced in case of the same heuristic information (Case 5 in comparison with Case 1, Case 6 in comparison with Case 2 and so on). Of curse, using the local search has often caused an increase in the time, but the CPU times of the different cases for the same examples have been so close that could be ignored. Because the sizes of the examples are large enough to conclude, using the local search is suggested. Moreover, Case 5 in comparison with Cases 6, 7 and 8 has not been better. Consequently, the ACS approach coupled with the local search and with the heuristic information based on (6) and (7) is recommended.

Finally, Table 7 gives a comparison between the results of ACS (Case 6) and AS for examples 1–4. The proposed approach has been the same as AS in examples 1 and 2, but superior in examples 3 and 4. In addition, it is worth to point out that the results of AS have not been generated using unique parameter values for all of the four examples (four set of numeric parameters have been used; for details see (Nahas & Nourelfath, 2005). This implies that what set of parameter values should be used to solve another example is not specified. Therefore, the parameters of AS should be set again and again and this may be a weak point, whereas the numeric parameters prepared for ACS are unique and because of the variety of the five examples, it is assumed that they can be employed to solve any other example. On the whole, ACS has outperformed AS and has been able to get very good solutions for large problems at a reasonable CPU time.

5. Conclusions

In this paper, an ant colony approach is presented for reliability optimization of a series system with multiple-choice and budget constraints. Each artificial ant constructs a solution by iteratively applying a pseudo-random transition rule based on both the heuristic information and the pheromone trails. The heuristic information is calculated based on an aggregation of two fuzzy sets. The generated solution may be infeasible; in other words, the total cost of the chosen technologies may be greater than the available budget. An infeasible solution is replaced by a feasible one using a neighborhood search procedure which randomly searches and finds a feasible solution with nearly highest reliability. The solution is then improved by an efficient local search method. Finally, the ant changes the pheromone intensity on each edge related to its chosen technologies using the local updating rule. Once all ants have built their solutions, the pheromone trails are globally modified in order to make the search more directed. To evaluate the performance of the developed approach, it has been compared with the only available algorithm. Our algorithm has effectively been able to obtain optimal or near optimal solutions for large problems. Computational experiments are given to show the superiority of the proposed ant colony approach.

Table 7
Performance comparison.

Example	ACS				AS			
	Minimum	Average	Std. dev.	Maximum	Minimum	Average	Std. dev.	Maximum
1	0.857054	0.857054	0	0.857054	0.85705	0.85705	0	0.85705
2	0.915042	0.915042	0	0.915042	0.91504	0.91504	0	0.91504
3	0.965134	0.965134	0	0.965134	0.96406	0.96439	0.00050	0.96513
4	0.865439	0.865439	0	0.865439	0.86465	0.86491	0.00038	0.86543

References

Ait-Kadi, D., & Nourelfath, M. (2001). Availability optimization of fault-tolerant systems. In *International conference on industrial engineering and production management (IEPM'2001), Quebec*.

Chen, T. C. (2006). IAs based approach for reliability redundancy allocation problems. *Applied Mathematics and Computation, 182*, 1556–1567.

Coit, D., & Smith, A. (1996). Reliability optimization of series–parallel systems using a genetic algorithm. *IEEE Transactions on Reliability, 45*(2), 254–260.

Dorigo, M. (1992). *Optimization, learning and natural algorithm*. PhD thesis, DEI, Politecnico di Milano, Itally (in Italian).

Dorigo, M., & Gambardella, L. M. (1997a). Ant colonies for the traveling salesman problem. *BioSys, 43*, 73–81.

Dorigo, M., & Gambardella, L. M. (1997b). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation, 1*, 53–66.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems man and Cybernetics – Part B, 26*, 29–41.

Dorigo, M., & Stutzle, T. (2003). The ant colony optimization metaheuristic: algorithms, applications and advances. In F. Glover & G. Kochenberger (Eds.), *Handbook of metaheuristics* (pp. 251–285). Norwell, MA: Kluwer Academic Publishers.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability*. San Francisco: Freeman.

Hsieh, Y. C. (2003). A linear approximation for redundant reliability problems with multiple component choices. *Computers & Industrial Engineering, 44*, 91–103.

Nahas, N., & Nourelfath, M. (2005). Ant system for reliability optimization of a series system with multiple-choice and budget constraints. *Reliability Engineering & System Safety, 87*, 1–12.

Nauss, R. M. (1978). The 0–1 knapsack problem with multiple choice constraints. *European Journal of Operational Research, 2*(2), 121–131.

Nourelfath, M., & Nahas, N. (2003). Quantized Hopfield networks for reliability optimization. *Reliability Engineering & System Safety, 81*, 191–196.

Ramirez-Marquez, J. E., & Coit, D. W. (2004). A heuristic for solving the redundancy allocation problem for multi-state series–parallel systems. *Reliability Engineering & System Safety, 83*, 341–349.

Ruan, N., & Sun, X. (2006). An exact algorithm for cost minimization in series reliability systems with multiple component choices. *Applied Mathematics and Computation, 181*, 732–741.

Sinha, P., & Zoltners, A. A. (1979). The multiple choice knapsack problem. *Operations Research, 27*(3), 503–515.

Sung, C. S., & Cho, Y. K. (2000). Reliability optimization of a series system with multiple-choice and budget constraints. *European Journal of Operational Research, 127*, 159–171.

Sung, C. S., & Lee, H. K. (1994). A branch-and-bound approach for spare unit allocation in a series system. *European Journal of Operational Research, 75*(1), 217–232.

Tan, Z. (2003). Minimal cut sets of s–t networks with k-out-of-n nodes. *Reliability Engineering & System Safety, 82*, 49–54.

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

Yeh, W. C. (2004). A simple algorithm for evaluating the k-out-of-n network reliability. *Reliability Engineering & System Safety, 83*, 93–101.

Yeh, W. C. (2006). A new algorithm for generating minimal cut sets in k-out-of-n networks. *Reliability Engineering & System Safety, 91*, 36–43.

Yeh, W. C. (2009). A two-stage discrete particle swarm optimization for the problem of multiple multi-level redundancy allocation in series systems. *Expert Systems with Applications, 36*, 9192–9200.

You, P. S., & Chen, T. C. (2005). An efficient heuristic for series–parallel redundant reliability problems. *Computers & Operations Research, 32*, 2117–2127.

Zhao, R., & Liu, B. (2004). Redundancy optimization problems with uncertainty of combining randomness and fuzziness. *European Journal of Operational Research, 157*, 716–735.

Zhao, J. H., Liu, Z., & Dao, M. T. (2007). Reliability optimization using multiobjective ant colony system approaches. *Reliability Engineering & System Safety, 92*, 109–120.