

Research Article

A New Biobjective Model to Optimize Integrated Redundancy Allocation and Reliability-Centered Maintenance Problems in a System Using Metaheuristics

Shima MohammadZadeh Dogahe¹ and Seyed Jafar Sadjadi²

¹Department of Industrial Engineering, Islamic Azad University, Science and Research Branch, Hesarak, Tehran 1477893855, Iran

²Center of Excellence in Advanced Manufacturing and Optimization, School of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

Correspondence should be addressed to Shima MohammadZadeh Dogahe; shima.488@gmail.com

Received 5 February 2015; Revised 27 May 2015; Accepted 17 June 2015

Academic Editor: Babak Shotorban

Copyright © 2015 S. MohammadZadeh Dogahe and S. J. Sadjadi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A novel integrated model is proposed to optimize the redundancy allocation problem (RAP) and the reliability-centered maintenance (RCM) simultaneously. A system of both repairable and nonrepairable components has been considered. In this system, electronic components are nonrepairable while mechanical components are mostly repairable. For nonrepairable components, a redundancy allocation problem is dealt with to determine optimal redundancy strategy and number of redundant components to be implemented in each subsystem. In addition, a maintenance scheduling problem is considered for repairable components in order to identify the best maintenance policy and optimize system reliability. Both active and cold standby redundancy strategies have been taken into account for electronic components. Also, net present value of the secondary cost including operational and maintenance costs has been calculated. The problem is formulated as a biobjective mathematical programming model aiming to reach a tradeoff between system reliability and cost. Three metaheuristic algorithms are employed to solve the proposed model: Nondominated Sorting Genetic Algorithm (NSGA-II), Multiobjective Particle Swarm Optimization (MOPSO), and Multiobjective Firefly Algorithm (MOFA). Several test problems are solved using the mentioned algorithms to test efficiency and effectiveness of the solution approaches and obtained results are analyzed.

1. Introduction and Literature Review

In general, reliability is defined as ability of a system to meet required performance standards under specified conditions during a determined time horizon. It has a significant effect on manufacturing cost, company's fame, production efficiency and environment, and so forth. There are two main approaches to enhance system reliability: implementing a proper maintenance policy and using effective redundancy strategies. Applying these approaches leads to increases in system reliability along with increasing costs of other resources. Thus, reaching a tradeoff between system reliability, cost, volume, weight, and so forth is significantly important [1].

There are two types of maintenance policies: corrective maintenance (CM) and preventive maintenance (PM).

In corrective maintenance, the system is repaired or replaced after failure. However, prescheduled periodic maintenance actions are taken in preventive maintenance. It is obvious that preventive maintenance prevents major failures that impose high costs on the system. In recent years, many authors conducted variety of research on preventive maintenance scheduling problems. A mathematical model has been proposed by Goel and Gupta [2] for a multistate system with repair and replacement policy. Goel et al. [3] presented a formulation for design, production, and maintenance planning to incorporate the reliability allocation problem at the design stage. Then, a simultaneous optimization framework has been employed to solve the proposed model. Tsai et al. [4] studied the preventive maintenance scheduling problem for a multicomponent system by assuming three maintenance actions:

mechanical service, repair, and replacement. Periodic preventive maintenance actions have been considered in order to maximize availability of the system. Mohanta et al. [5] have employed both GA and hybrid GA/SA techniques to optimize maintenance scheduling for a power plant and compared the obtained results by the algorithms.

Martorell et al. [6] proposed a multiobjective model to optimize the maintenance scheduling problem by integrating human and material resources. Reliability, availability, maintainability, and cost were considered as objectives of the model and the Genetic Algorithm was used to solve the problem. Another multiobjective maintenance model for a series-parallel system was investigated by Certa et al. [7]. They implemented periodic PM policy and considered maintenance cost and makespan as objectives of the problem. An effective Pareto optimal frontier approach was applied to solve the multiobjective problem. Moghaddass et al. [8] focused on finding an optimal tradeoff between design of a repairable multistate system with binary components and its maintenance strategy. Also, they considered both active and standby redundancy strategies. Doostparast et al. [9] developed a reliability-based approach to optimize preventive maintenance scheduling in coherent systems. A system is called coherent when its performance is related to all of the components. In other words, for coherent systems each component is relevant and system structure function is monotone and nondecreasing [10]. They studied periodic PM performance in three types of coherent systems and used a Simulated Annealing (SA) algorithm to solve the problem trying to minimize total costs along with meeting the minimum predetermined reliability level. Several other studies dealing with preventive maintenance scheduling problem have been done by [11–13] during the last two decades.

However, selecting a proper maintenance policy is not all we can do to maximize system reliability. Identifying and implementing the best redundancy strategy is another way to optimize system reliability. One of the famous problems in field of reliability optimization is redundancy allocation problem (RAP). Redundant components are incorporated into the system to back up different parts of the system and prevent system breakdown under different redundancy strategies. There are two main redundancy strategies: (1) active redundancy, in which all redundant components are implemented in a parallel structure together from time zero and only one component is required to work at any given time, and (2) standby redundancy, in which a sequential order is determined for using the redundant components at component failure time. Three variants of the standby redundancy strategy are called cold, warm, and hot. Each strategy can be implemented in a different part of a system. RAPs are proved to be NP-hard by Chern [14]. Therefore, metaheuristic algorithms have been widely used in the literature to solve such problems. Literature of the redundancy allocation problem (RAP) could be reviewed from several points of view. In this paper, literature of the problem has been accurately reviewed by assuming three main characteristics: objectives of the problem, applied solution algorithms, and considered redundancy strategies.

Coit [15] studied the redundancy optimization problem using integer programming approach and logarithm function to develop an equivalent formulation of the problem and obtained high quality solutions. He assumed nonconstant component hazard functions, Erlang distribution for component time-to-failure, imperfect switching, and multiple component choices for each subsystem. Cold standby redundancy strategy is considered for a nonrepairable series-parallel system. Zhao and Liu [16] proposed a stochastic model for the redundancy optimization problem aiming at maximizing system lifetime or system reliability by considering both active and standby redundancy strategies. They used stochastic simulation, Genetic Algorithm (GA), and Neural Network (NN) to develop a hybrid intelligent algorithm to solve the problem. Liang and Smith [17] proposed an Ant Colony Optimization (ACO) Algorithm to solve the redundancy allocation problem (RAP) for a series-parallel system. The objective is to maximize system reliability when active redundancy strategy has been implemented in the system. Restrictions are set on system cost and system weight, in addition. They found the ACO algorithm to be very effective and efficient for solving NP-hard reliability design problems because it brings GA flexibility, robustness, and ease of implementation along with improving its random behavior. Tavakkoli-Moghaddam et al. [18] studied RAP for a series-parallel system by considering both active and standby redundancy strategies. They formulated the problem as a nonlinear integer programming model and used a Genetic Algorithm to solve the NP-hard problem and maximize system reliability. Sadjadi and Soltani [19] proposed a heuristic and a hybrid GA for the RAP in a series-parallel system to maximize its reliability. Parameters of the proposed hybrid GA are calibrated using Taguchi's robust design method to enhance efficiency and effectiveness of the algorithm. Solving numerical examples indicated that the proposed heuristic method is time-efficient and produces comparable solutions to the hybrid GA in terms of quality. Kumar et al. [20] studied a multiobjective multilevel RAP and proposed multiobjective hierarchical Genetic Algorithms to solve two numerical examples. They integrated the hierarchical genotype encoding scheme with two multiobjective Genetic Algorithms. Beji et al. [21] proposed a hybrid metaheuristic algorithm based on Particle Swarm Optimization (PSO) and local search for RAP in a series-parallel system and tried to maximize system reliability.

A large number of studies on redundancy allocation problem have been conducted after 2010. Among those who studied multiobjective RAP (MORAP), Zio and Bazzo [22] used a level diagram analysis of Pareto solutions to assist the decision maker in selecting his/her preferred system design in terms of reliability and availability. Soylu and Ulusoy [23] applied UTADIS sorting procedure to categorize the Pareto solutions obtained by augmented epsilon constraint method into preference ordered classes. They considered maximization of the minimum system reliability along with minimization of the overall system cost and weight as objectives. Safari [24] studied a MORAP by considering system reliability and overall system cost as objectives and both active and standby strategies as candidate redundancy strategies. He used a Non-dominated Sorting Genetic Algorithm (NSGA-II) to solve

the multiobjective RAP. Khalili-Damghani and Amiri [25] applied three solution methods, epsilon constraint, multistart partial bound enumeration algorithm, and Data Envelopment Analysis (DEA) to optimize system reliability, cost, and weight in a RAP for a series-parallel system. Chambari et al. [26] studied a biobjective RAP trying to maximize system reliability and minimize overall cost along with making decision about using active and/or standby redundancy strategies for a system with nonrepairable components. They proposed two metaheuristics, NSGA-II and MOPSO, to solve the problem. However, Zoufaghari et al. [27] considered system reliability and availability as objectives of a RAP with both repairable and nonrepairable components and proposed a mixed integer nonlinear programming (MINLP) model for the problem. Cao et al. [28] used a decomposition-based exact approach to solve a multiobjective RAP of a mixed system trying to optimize system reliability, cost, and weight. Garg and Sharma [29] studied a multiobjective RAP with nonstochastic uncertain parameters by considering system reliability and cost as objectives. They formulated the problem as a fuzzy multiobjective optimization problem.

Firefly Algorithm as a new metaheuristic optimization method was introduced in 2008 by Yang [30]. dos Santos Coelho et al. introduced a modified FA approach combined with chaotic sequences to optimize reliability-redundancy problem [31]. Many other authors proposed metaheuristic algorithms for RAP. Sadjadi and Soltani [32] developed a heuristic method and a honey bee mating algorithm to solve the large-scale RAP. Hsieh and Yeh [33] applied a penalty guided bee colony algorithm for RAP in a series-parallel system. Several other metaheuristic algorithms have been proposed by other researchers in [34–38].

In this paper, a novel mathematical model of a system of repairable and nonrepairable components is formulated. The model contains two objectives: firstly, it aims to select a proper redundancy strategy for nonrepairable part of the system and secondly, it offers a maintenance policy for repairable part of the system. Minimizing net present value of total cost and maximizing system reliability are objectives of the problem. In addition, different types of redundancy strategies, repair, and replacement actions are considered in order to model the problem as realistic as possible. Other practical constraints such as available budget for purchasing redundant components, volume, weight, and maximum allowed failure rate in each inspection period are taken into account. Due to NP-hardness of the problem, the authors tried to employ metaheuristic methods to solve proposed model. Three common solution approaches called NSGA-II, MOPSO, and MOFA were selected based on the Vanoye and Parra classification. Ruiz-Vanoye and Díaz-Parra [39] classified metaheuristics into three groups: metaheuristics based on gene transfer (like Genetic Algorithms), metaheuristics based on interactions among individual insects (e.g., Ant Colony, Honey Bees, and Firefly Algorithms) and metaheuristics based on biological aspects of alive beings (such as Simulated Annealing, Tabu Search, and Particle Swarm Optimization Algorithms).

Remainder of the paper is organized as follows. In Section 2, mathematical formulation of the problem is proposed

followed by detailed explanation of objective functions and constraints. Three metaheuristic algorithms are presented in Section 3 to solve the proposed model. A set of numerical examples have been solved using the metaheuristics in Section 4. Then, obtained results are indicated and computational analysis is carried out. Finally, a summary of the paper and conclusions have been presented in Section 5.

2. Problem Formulation

In this section, a new integrated mathematical model is proposed for redundancy allocation and reliability-centered maintenance problems. Objective of the reliability problems could be one or a set of the following objectives: maximizing system reliability and minimizing cost, weight, and volume of the system. In this paper, system reliability and costs including maintenance and operational costs are considered as objectives.

In most articles, the system under study includes either repairable or nonrepairable components. However, systems usually consist of repairable and nonrepairable components simultaneously in real world [27]. Generally, components of electronic devices are not repairable and should be replaced by new ones after failure. However, components of mechanical systems are usually repairable and repairing or replacing the broken component after failure brings the system back to the normal condition.

In this paper, a system of electronic and mechanical components has been considered. Figure 1 represents configuration of the system. Two approaches are applied to achieve the highest possible system reliability: (1) maximizing reliability of each component by using a diverse set of high quality and reliable redundant parallelized components (heterogeneous redundancy); (2) choosing optimal maintenance policies. It is obvious that the first approach can be used for nonrepairable electronic components and the second one is applied on repairable mechanical components.

Assumptions. The following assumptions have been taken into account in proposing mathematical formulation of the problem under study:

- (i) The system is comprised of two subsystems in series: mechanical components and electronic components (Figure 1). Mechanical components are repairable while electronic components are nonrepairable.
- (ii) Selecting optimal maintenance policy for mechanical components is considered in order to maximize system reliability.
- (iii) Selecting a proper redundancy strategy, active or cold standby, and determining the number of redundant components in the electronic section is taken into account aimed at improving system reliability. Selecting active redundancy strategy adds operational costs to the system cost while selecting cold standby redundancy strategy threatens system performance by imperfect switching.

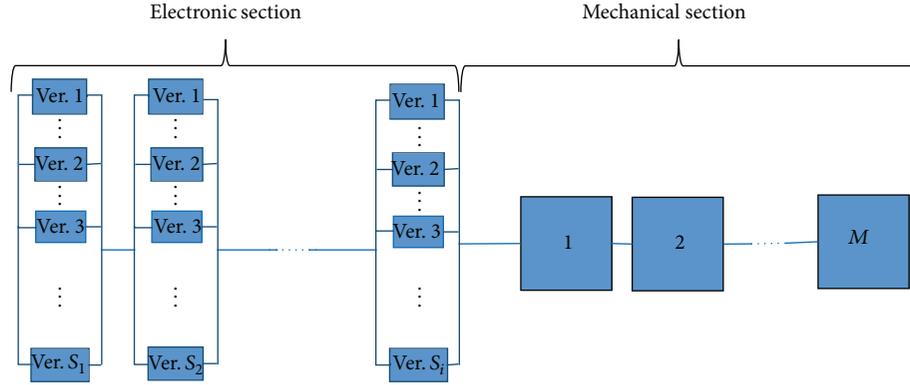


FIGURE 1: A system of mechanical and electronic subsystems.

- (iv) It is possible to use different type of components with different initial and operational costs and failure rates for the electronic subsystem.
- (v) Since unstable market and economic conditions may have serious effects on results, inflation rate and time value of money are considered in computations.
- (vi) Required resources such as financial resources, human resources, volume, and weight are known deterministically.
- (vii) Time-to-failure distributions of components are independent.
- (viii) A fixed amount of budget is available at time zero to purchase electronic components (initial cost).
- (ix) Secondary cost of the system is calculated by taking operational costs of the electronic subsystem and maintenance costs of the mechanical subsystem into account during the system running period (mission time).
- (x) The system mission time is finite.
- (xi) Repair and replacement times and restoration times are calculated as the system downtime cost.
- (xii) In cold standby strategy, redundant components do not fail before their activation. In addition, failure rates in active redundancy strategy are larger than cold standby strategy because active redundant components are exposed to the operational stresses. (i.e., $\lambda_{ij1} \geq \lambda_{ij2}$)
- (xiii) Failure detection mechanism and switching are imperfect.

2.1. Mathematical Model. The constraints of the problem can be formulated as follows:

$$\sum_{k=1}^2 \sum_{j=1}^{S_i} \sum_{i=1}^E w_{ij} \cdot x_{ijk} \leq W, \quad (1)$$

$$\sum_{k=1}^2 \sum_{j=1}^{S_i} \sum_{i=1}^E v_{ij} \cdot x_{ijk} \leq V, \quad (2)$$

$$\sum_{k=1}^2 \sum_{j=1}^{S_i} x_{ijk} \leq N_{\max i} \quad \forall i = 1, \dots, E, \quad (3)$$

$$x_{ijk} \leq M y_{ik} \quad \forall i = 1, \dots, E, \quad j = 1, \dots, S_i, \quad k = 1, 2, \quad (4)$$

$$\sum_{k=1}^2 y_{ik} = 1 \quad \forall i = 1, \dots, E, \quad (5)$$

$$\sum_{l=1}^L x m_{rlt} + \sum_{u=1}^{S_r} x r_{rut} \leq 1 \quad (6)$$

$$\forall r = 1, \dots, M, \quad t = 1, \dots, mT,$$

$$\sum_{r=1}^M \left(\sum_{l=1}^L h_{rl} \cdot x m_{rlt} + \sum_{u=1}^{S_r} h_{ru} \cdot x r_{rut} \right) \leq H_o \quad (7)$$

$$\forall t = 1, \dots, mT,$$

$$\lambda_{rt} = (\lambda_{rt-1} + \eta_r) \left(1 - \left(\sum_{l=1}^L x m_{rlt} + \sum_{u=1}^{S_r} x r_{rut} \right) \right) \quad (8)$$

$$+ \sum_{l=1}^L x m_{rlt} \bar{\lambda}_{rl} + \sum_{u=1}^{S_r} x r_{rut} \bar{\lambda}_{ru}$$

$$\forall r = 1, \dots, M, \quad t = 1, \dots, mT,$$

$$\lambda_{rt} \leq \lambda_{\max, r} \quad \forall r = 1, \dots, M, \quad t = 1, \dots, mT, \quad (9)$$

$$\sum_{i=1}^E \sum_{j=1}^{S_i} \sum_{k=1}^2 x_{ijk} c_{ij} \leq B_0, \quad (10)$$

$$x m_{rlt} \in \{0, 1\},$$

$$x r_{rut} \in \{0, 1\},$$

$$y_{ik} \in \{0, 1\},$$

$$x_{ijk} \in Z^+.$$

(11)

Constraints (1) and (2) ensure that total weight and volume of the electronic components are lower than the maximum allowed amounts. Constraint (3) represents the maximum allowed number of redundant components in each subsystem. Constraints (4) and (5) ensure that each subsystem uses only one redundancy strategy. Constraint (6) states that components of the mechanical section can be repaired or replaced only at inspection points. Constraint (7) ensures that the number of required operators to perform maintenance actions does not exceed the number of available operators. Constraint (8) calculates failure rates of mechanical components in each period. Failure rate is increased by η_r if no repair or replacement action is taken on the component at period t and is changed into $\bar{\lambda}_{rl}$ or $\bar{\lambda}_{ru}$ if the component is repaired or replaced at period t , respectively. Constraint (9) ensures that failure rates of mechanical components in each period do not exceed the maximum allowed amount. Constraint (10) represents the maximum available budget for purchasing electronic components at the beginning of the mission. Finally, constraint (11) represents domain restrictions of the decision variables.

2.2. Objective Functions. The proposed model contains two objectives:

- (1) Maximizing system reliability in each period by selecting optimal redundancy strategy and maintenance policy.
- (2) Minimizing secondary cost of the system including operational costs of the electronic subsystem and maintenance costs of the mechanical subsystem during the system mission.

In the following of this section, detailed explanations on the introduced objectives have been proposed.

2.2.1. System Reliability. The system under study consists of two electronic and mechanical subsystems connected in series according to Figure 1. Thus, system reliability is equal to $R_S(t) = R_E(t) \times R_M(t)$, where $R_E(t)$ and $R_M(t)$ are reliability of electronic and mechanical subsystem, respectively.

Electronic Subsystem. The electronic part consists of E subsystems connected in series. Each subsystem can use either active or cold standby redundancy strategy to maximize reliability of the subsystem. As discussed previously, electronic components are nonrepairable and failed components can only be replaced by redundant components until all redundant components are used. Generally, cold standby redundancy leads to higher system reliability in comparison with active redundancy [15].

Let H subsystems use active redundancy and C ($C = E - H$) subsystems use cold standby redundancy. Thus, reliability of the electronic subsystem is calculated according to the following equation:

$$R_E(t) = R_{\text{Hot}}(t) R_{\text{Cold}}(t). \quad (12)$$

$R_{\text{Hot}}(t)$ is reliability of the subsystem under active redundancy strategy that is calculated by considering it as

a subsystem with a number of redundant components and calculated as follows:

$$R_{\text{Hot}}(t) = \prod_{i \in H} \left(1 - \left(1 - r_{ij}(t) \right)^{x_{ijk}} \right), \quad (13)$$

where $r_{ij}(t)$ is the reliability of component j in the subsystem i at period t . Also, reliability of a system under cold standby redundancy is calculated as (14)

$$R_{\text{Cold}}(t) = \prod_{i \in C} \left(r_{ij}(t) + \sum_{n=1}^{x_{ijk}-1} \int_0^T \rho_i(u) f_{ij}^n(u) r_{ij}(T-u) du \right), \quad (14)$$

where $f_{ij}^n(t)$ is the density function of n th failure for type j of the subsystem i . Also, $\rho_i(u)$ refers to switch reliability. Coit [15] calculated a lower bound for $R_{\text{Cold}}(t)$ and proposed an approximation of the system reliability by considering imperfect switching using the following equation:

$$\bar{R}_{\text{Cold}}(t) = \prod_{i \in C} \left(r_{ij}(t) + \rho_i(t) \sum_{n=1}^{x_{ijk}-1} \int_0^T f_{ij}^n(u) r_{ij}(T-u) du \right). \quad (15)$$

According to Coit's approximation [26] and since most electronic devices have exponential failure distributions and system working time is equal to the sum of working times of components, time-to-failure distribution function of each component is Erlang (λ, k) ; thus, reliability of the electronic subsystem is calculated as follows:

$$R_E(t) = \prod_{i \in H} \left(1 - \prod_{j=1}^{S_i} \left(1 - e^{-\lambda_{ijk} t} \sum_{l=0}^{k_{ij}-1} \frac{(\lambda_{ijk} t)^l}{l!} \right)^{x_{ijk}} \right) \prod_{i \in C} \left(1 - \prod_{j=1}^{S_i} \left(1 - e^{-\lambda_{ij} t} \left(\sum_{l=0}^{k_{ij}-1} \frac{(\lambda_{ijk} t)^l}{l!} + \rho_i(t) \sum_{l=k_{ij}}^{k_{ij} x_{ijk}-1} \frac{(\lambda_{ijk} t)^l}{l!} \right) \right) \right). \quad (16)$$

Mechanical Subsystem. The mechanical subsystem includes M devices connected in series. Therefore, all devices should be in the working condition to have an active subsystem during the mission time. In order to prevent system shutdown, scheduled maintenance actions should be taken. Reliability of the mechanical subsystem is calculated by multiplying reliability of all components as follows:

$$R_M(t) = \prod_{r=1}^M p_r(t), \quad (17)$$

where $p_r(t)$ indicates reliability of the component r at period t . Each time period is divided into m equal intervals for inspection. Thus, we have $m \times T$ inspections during the time horizon. All components are inspected in each interval and one of the following maintenance actions is performed on each component: (1) inspection, (2) preventive repair, and (3) preventive replacement [9]. Each action has a special effect on

reliability of the component with a different cost dependent on the amount of required resources. Lifetime of the mechanical devices has Weibull $(\alpha, 1/\lambda)$ distribution function. Thus, reliability of the mechanical subsystem is calculated as follows:

$$R_M(t) = \prod_{r=1}^M e^{-(\lambda_r t)^\alpha}. \quad (18)$$

In order to calculate reliability of the system in each period, a few points should be considered: (1) each component deteriorates after activation with an increase in its failure rate; (2) failure rate of a component is reduced or fixed by performing maintenance actions [3].

2.2.2. System Cost. Total cost of the system (C_T) could be divided into two parts: initial costs and secondary costs. Initial costs (C_I) include purchasing costs of the electronic devices while secondary costs (C_S) refer to operational costs of the electronic subsystem (C_S^O) [40] plus inspection and maintenance costs of the mechanical subsystem (C_S^M). The objective is to minimize secondary costs ($C_S = C_S^O + C_S^M$) according to restrictions on the initial budget for purchasing costs and other constraints.

Initial Cost. Initial cost is calculated based on the purchase cost and number of the components implemented in electronic subsystem at the beginning of running the system

$$C_I = \sum_{i=1}^E \sum_{j=1}^{S_i} \sum_{k=1}^2 x_{ijk} c_{ij}. \quad (19)$$

Secondary Cost. As mentioned before, secondary cost is sum of the operational costs of the electronic components and maintenance and inspection costs of the mechanical components. Operational cost is calculated by multiplying the number of operating components by unit operational cost (\hat{c}_{ij}). Number of operating components in each period is determined according to the redundancy strategy used in the system. If active redundancy is used, operational cost is calculated for all working components because all redundant components are operating from time zero in this strategy. However under cold standby redundancy, operational cost is only calculated for the operating redundant component because one component is required to be operating in this strategy.

It should be noted that, in active redundancy, expected value for failure of each component in each period (λ_{ijk}) is deducted from the total amount. Using compound interest rate, operational costs of different periods are converted to the present time and net present value of the total operational cost is calculated using the following equation:

$$C_S^O = \sum_{t=1}^{mT} \sum_{i=1}^E \sum_{j=1}^{S_i} \sum_{k=1}^2 c'_{ij} \left((x_{ijk} - [(t-1)\lambda_{ijk}]) y_{ik} + (1 - y_{ik}) \right) \cdot (1+I)^t. \quad (20)$$

Maintenance costs of the mechanical components consist of repair/replacement costs, plus system downtime cost due to performing maintenance actions. Repairing the component adds a repair cost while replacing it adds a purchasing cost to the secondary cost. System downtime cost is calculated based on the selected maintenance. Present value of maintenance costs is calculated using (21) and added to the secondary costs

$$C_S^M = \sum_{t=1}^{mT} \sum_{r=1}^M \left(\sum_{l=1}^L (c_{rl}^1 + c_{rl}^3) x m_{rlt} + \sum_{u=1}^{S_r} (c_{ru}^2 + c_{ru}^4) x r_{rut} \right) (1+I)^t. \quad (21)$$

3. Solution Approaches

As mentioned in Introduction and Literature Review, the solution methods were selected based on the Ruiz-Vanoye and Díaz-Parra [39] metaheuristics classification. In this section, the mechanisms of three algorithms, NSGA-II, MOPSO, and MOFA, are explained; then in Section 4, some numerical examples based on the proposed model are generated and are solved using these methods. Finally, obtained results by algorithms have been compared and analyzed.

The proposed model for the problem contains two conflicting objectives. We try to make a tradeoff between these objectives to achieve a desired level of optimality for each objective. One of the common approaches to solve multiobjective problems is the weighted-sum method that converts the problem into a single objective problem by making a weighted linear combination (WLC) of objectives. Although it is a very popular method due to its simplicity and ease of implementation, it has some major disadvantages such as determining weight of each objective and lack of information about it. Thus, another method called Pareto set has been developed. Pareto set method produces set of solutions within the feasible region of the problem that dominate other feasible solutions. The nondominated solution sets are called Pareto optimal solutions and other inferior solutions are called dominated solutions. The decision maker selects the final set of Pareto optimal solutions according to his/her preferences and considered criteria. In this paper, three metaheuristic algorithms have been employed to produce Pareto optimal solutions.

3.1. Nondominated Sorting Genetic Algorithm (NSGA-II). The NSGA-II is developed for solving multiobjective problems by adding two operators to the classic Genetic Algorithm (GA) to find Pareto optimal sets instead of finding the unique optimal solution [41]. The additional operators are (1) ranking operator which assigns a rank to each member of the generation based on nondominated sorting and (2) diversity operator which increases diversity of the produced solutions with equal ranks. In the following of this section, procedure of the proposed NSGA-II has been presented.

In our study, the proposed chromosome has two parts. The first part represents the electronic section and the second part represents the mechanical section. The electronic section

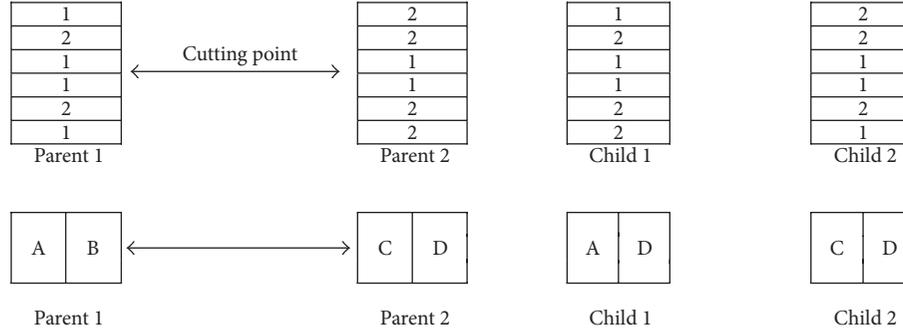


FIGURE 2: Crossover in Genetic Algorithm.

TABLE 1: Electronic section matrix.

i	1	2	\dots	S_i	Redundancy strategy
1	2	0	\dots	1	1
2	1	1	\dots	2	2
\vdots	\vdots	\vdots	\dots	\vdots	\vdots
E	0	2	\dots	1	1

contains E rows, which refer to the number of electronic subsystems, and S_i+1 columns where S_i columns are allocated to different types of components and the last column represents the selected redundancy strategy. If active redundancy is selected, the related element in the last column is equal to 1 while selecting cold standby redundancy makes this element equal to 2. Element (i, j) indicates the number of implemented redundant components for component type j of subsystem i . In other words, if element $(2, 3)$ is equal to 1 it means that one redundant component is implemented for component type 3 in subsystem 2 (Table 1).

The proposed chromosome for the mechanical section is indicated by two matrixes with M rows representing mechanical subsystems and mT columns representing inspection intervals. Elements of the first matrix take three values: 0, 1, or 2. Number 0 indicates that no repair or replacement action is required for that subsystem. Number 1 refers to replacement and number 2 refers to repair action. The second matrix determines type of the replacement and repair actions.

Crossover Operator. The proposed matrix for the electronic section is divided into two parts in order to perform crossover. The first part is related to selecting a redundancy strategy for each subsystem, that is, the last column, while the second part is related to determining the number of redundant components for components of each subsystem, that is, (i, j) elements. The chromosome is cut from a randomly selected point to diversify the selected redundancy strategies. Then, the resulting two slices from parent chromosomes are interchanged. This process changes the selected redundancy strategy for some subsystems. For the second part of the chromosome, a vertical cut is applied from a randomly selected

point. Then, the resulting two slices are interchanged. Figure 2 indicates the applied crossover operation.

Mutation Operator. Mutation process for the first chromosome, that is, electronic section, is performed as explained in the following. First, a subsystem is selected randomly. Next, one of the allowed components is selected for the selected subsystem. Then, a random number between 1 and N_{maxi} is assigned to the selected subsystem and component. Finally, redundancy strategy is selected randomly. For the mechanical section, mutation process is started by selecting a subsystem and a period randomly. Then, selected maintenance policy and type of repair/replacement actions are changed.

3.2. Particle Swarm Algorithm. The Multiobjective Particle Swarm Optimization (MOPSO) is a metaheuristic algorithm capable of producing high quality nondominated Pareto optimal solutions with high diversities for multiobjective problems. The MOPSO is widely used by researchers due to its simplicity and successful performance in continuous optimization problems. The idea of this algorithm is inspired by a swarm of birds looking for food [42]. In this algorithm, each feasible solution is indicated as a particle with known velocity and fitness value. Particles move in the search space and obtained results are classified based on a fitness criterion at the end of each time interval. Particles are changed into particles with higher fitness values gradually based on the following model:

$$\begin{aligned} \vec{V}_i(t+1) &= W \times \vec{V}_i(t) + C_1 r_1 (\vec{P}_i - \vec{X}_i(t)) \\ &\quad + C_2 r_2 (\vec{P}_g - \vec{X}_i(t)) \quad i = 1, 2, \dots, m, \quad (22) \\ \vec{X}_i(t+1) &= \vec{X}_i(t) + \vec{V}_i(t+1) \quad i = 1, 2, \dots, m, \end{aligned}$$

where W , C_1 , C_2 , and t represent inertia weight, cognitive factor, social factor, and iteration number, respectively. Also r_1 and r_2 are random numbers in $[0, 1]$ and \vec{V}_i and \vec{X}_i are velocity and position vector of i th particles, respectively. Multiobjective approach of this algorithm (MOPSO) was proposed by Coello Coello and Lechuga in 2002 [43]. The improved version of this algorithm, in which constraint-handling mechanism and a mutation operator have been considered, was presented in 2004 [44].

```

Define objective functions:  $f_1(x), f_2(x), \dots, f_k(x)$  where  $x = (x_1, x_2, \dots, x_d)^T$ 
Generate initial population of fireflies  $x_i = (i = 1, \dots, n)$ 
Formulate light intensity  $I$ 
Define absorption coefficient  $\gamma$ ; randomization parameter  $\alpha_0, \epsilon_i$ ; vector of random numbers and maximum iteration
While  $t < \text{Maximum Iteration}$ 
  for  $i, j = 1 : n$  (all  $n$  fireflies)
    Evaluate approximations  $PF_i$  and  $PF_j$  to the Pareto Front
    if  $PF_j$  dominates  $PF_i$ 
      Move firefly  $i$  towards  $j$  using (24)
      if pervious position doesn't dominate new one
        New position replaced with old one
      end if
    end if
  end for
Update and pass non dominated solution to next generation
Update  $t = t + 1$ 
End while

```

ALGORITHM 1

In MOPSO, all objective functions are calculated and evaluated for each particle and the nondominated solutions (based on Pareto set concept) found by the particles are stored in a repository. The size of repository is limited and is set by decision maker. In addition, the search space is divided into hypercubes with a fitness value. Fitness value is inversely proportional to the number of particles it contains [45]. Then, the selection method is used to choose a hypercube and the best nondominated position (called leader) and finally, the velocities and positions of the particles are updated. This process is repeated for a certain number of iterations.

3.3. Firefly Algorithm. Firefly Algorithm (FA) is the last method applied in this study. It was introduced by Yang in 2010 [30] as a new approach for solving complex and continuous problems. The algorithm is inspired by the flashing behavior of fireflies to attract each other. Considering the following three rules, Firefly Algorithm introduced by Yang:

- (1) All fireflies are unisex and attracting a firefly by another one is regardless of its sex.
- (2) Attractiveness is proportional to the brightness and both of these features will decrease with increasing distance. Less bright fireflies are always attracted to the brighter one and if there is no brighter one, the move will be randomly.
- (3) The brightness of fireflies is defined according to the objective function (like fitness function in Genetic Algorithm).

Variation of light intensity and formulation of the attractiveness of the FA are two important issues. Brightness of each firefly at place x is defined as $I(x) \propto f(x)$ and attractiveness $\beta(r)$ is defined with respect to distance of the firefly i from firefly j that is calculated as following equation:

$$\beta(r) = \beta_0 e^{-\gamma r^m} \quad (m \geq 1), \quad (23)$$

where β_0 is attractiveness at $r = 0$ and γ is fixed light absorption coefficient. Based on the Cartesian distance, the distance between the firefly i and firefly j at x_i and x_j is obtained as $r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2}$ where $x_{i,k}$ is the k th component of x_i in spatial coordinate. Calculating distances is not limited only to the Euclidean coordinate, but given the type of the problem it can be defined differently, for example, as time interval. Moving of firefly i towards more attractive firefly j is calculated according to

$$x_{i+1} = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha_0 \epsilon_i. \quad (24)$$

The second term of (24) is movement because of attraction and the third term is the random movement where α_0 is a randomization parameter and ϵ_i is vector of random numbers with a Gaussian or uniform distribution. Although the basic design of this method was for continuous problems, recent studies have shown that this approach is also very efficient in discrete problems [46].

Single objective form of Firefly Algorithm (FA) was developed to Multiobjective Firefly Algorithm (MOFA) in 2013 by Yang [47]. In this approach, nondominated solutions are detected based on the objective function value. The main steps of the MOFA can be summarized as shown in Algorithm 1.

Since that, variables of our problem are binary and integer and MOPSO algorithm and MOFA find solutions in continuous space; round function is used to convert real number to the integer, and to change detected solutions to the binary style, sigmoid function is applied.

4. Numerical Examples

In this section, assigning proper values to the parameters of algorithms and using comparison metrics for evaluating solution methods are discussed first. Then, three different sets of test problems (small, medium, and large size) are tackled

TABLE 2: Range of the main parameters.

NSGA-II		MOPSO		MOFA	
Parameter	Range	Parameter	Range	Parameter	Range
Pop. size (N_{pop})	20–100	Pop. size (N_{pop})	20–100	Pop. size (N_{pop})	10–100
Max iteration number (Maxit)	50–200	Max iteration number (Maxit)	50–200	Max iteration number (Maxit)	50–200
Cross rate (Cr)	0.5–0.9	Inertia weight (W)	0.4–0.9	Randomization parameter (α_0)	0.1–0.9
Mutation rate (Mr)	0.01–0.3	Cognitive factor (C_1)	1–2	Fixed light absorption coefficient (γ)	1–3
		Social factor (C_2)	1–2		

TABLE 3: Optimum values of the algorithms parameters.

NSGA-II		MOPSO		MOFA	
Pop. size	100	Pop. size	100	Pop. size	25
Max iteration number	50	Max iteration number	50	Max iteration number	200
Cross rate	0.9	W	0.9	α_0	0.1
Mutation rate	0.3	C_1	1	γ	1
		C_2	2		

and solved using the chosen solution methods. Finally, obtained results by each algorithm have been compared and the obtained results for an example are explained.

4.1. Setting Parameters of the Algorithms. Setting proper values for the control parameters of metaheuristic algorithms has a significant effect on their desirable performance. Well-tuned parameters empower the algorithms in producing better solutions within shorter computation times. Thus, setting proper values for control parameters is a critical task [48]. A few additional techniques are applied to tune parameters. One of the common methods is response surface methodology (RSM). RSM is a mathematical and statistical technique that examines the relationship between one or more response variables and the set of parameters (input variables) influencing them. Using this method, the levels of parameters that optimize the response variables are identified. In the first step, main parameters and range of them are determined. In this study, we used related literature to identify these parameters and their range. Table 2 shows range of the main parameters of each algorithm.

In the next step, the response variables should be determined. Three performance metrics are chosen as response variables which are CPU time, number of nondominated solutions (NNS), and Diversification Metric (DM). These metrics were selected based on the two features, the convergence speed and diversity of the detected solutions. More details on the performance metrics can be found in Section 4.2.

Central composite design (CCD) with 6 center points is applied for the experiments. Experiments are run by Design Expert 9. According to the number of input variables and type of the design, different number of experiments should be run. For instance, in case of the four parameters and 6 center points' design, 46 experiments are required. After performing the experiments, analysis of variance (ANOVA) is applied to fit an adequate model to the experimental data. Last step is setting goals for responses to generate optimal condition

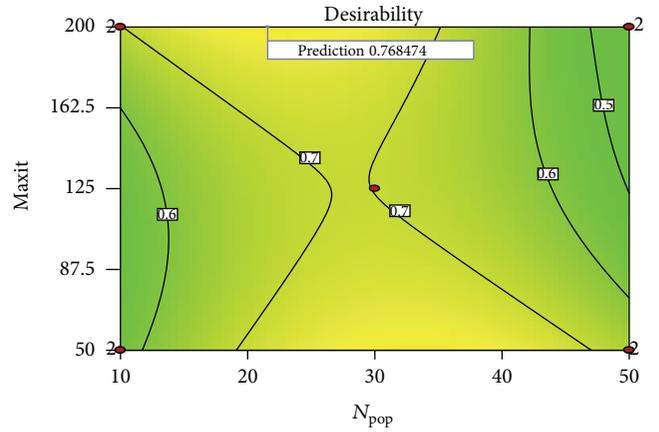


FIGURE 3: Counter plot for MOFA: desirability versus Maxit and N_{pop} .

(optimal level of the parameters). Here, we aim to minimize CPU time and maximize NNS and DM.

Figure 3 shows counter plot of RSM results for MOFA approach. Desirability displays the amount of goals that have been met. It can be concluded from Figure 3 that the highest desirability is obtained when N_{pop} level is medium and Maxit is high. RSM is applied for all three methods and the optimum values for the algorithms parameters are presented in Table 3.

Other parameters are set according to the literatures as follows:

$$\text{MOPSO: } r_1, r_2 = \text{rand}[0, 1], \text{ repository size} = 500 \text{ and} \\ \text{MOFA: } \epsilon_i = [\text{rand} - 1/2], \beta_0 = 1.$$

4.2. Performance Measures. According to Deb et al. [49], there are two main features that must be taken into account in order to evaluate performance of metaheuristic algorithms in solving multiobjective optimization problems: (1) convergence to the Pareto optimal set and (2) diversity of

TABLE 4: Test problems dimensions.

Case	Number of electronic components; i	Number of mechanical components; r	Number of component types in the electronic subsystem; j	Number of component types in the mechanical subsystem; u
Small size	*rand[3, 8]	rand[3, 7]	rand[2, 4]	rand[2, 4]
Medium size	rand[8, 13]	rand[7, 11]	rand[2, 4]	rand[2, 4]
Large size	rand[13, 18]	rand[11, 15]	rand[2, 4]	rand[2, 4]

*This function returns an integer number between lower and upper bounds randomly.

TABLE 5: Mean and standard deviation of the metrics for the different sizes of the test problems.

Problem size	Time (seconds)			NNS			DM			MS		
	NSGA-II	MOPSO	MOFA	NSGA-II	MOPSO	MOFA	NSGA-II	MOPSO	MOFA	NSGA-II	MOPSO	MOFA
Small	1.91 (0.26)	0.78 (0.23)	3.50 (1.16)	22.9 (13.54)	12.3 (3.45)	3.3 (1.26)	55.35 (30.22)	60.69 (20.54)	22.60 (11.72)	0.01493 (0.0257)	0.07999 (0.0542)	0.04230 (0.0376)
Medium	2.59 (0.21)	1.45 (0.46)	6.99 (2.05)	12.85 (5.60)	11.90 (3.80)	3.85 (1.14)	39.02 (17.32)	65.79 (23.50)	28.65 (12.71)	0.0238 (0.0367)	0.1562 (0.2004)	0.0482 (0.0303)
Large	31.49 (6.91)	22.35 (9.36)	91.25 (25.48)	15.60 (6.22)	12.84 (4.19)	4.11 (1.49)	51.35 (25.77)	97.44 (30.93)	34.35 (13.81)	0.00419 (0.0048)	0.14625 (0.0285)	0.07408 (0.0418)

*Numbers in the parentheses are standard deviation.

the produced Pareto optimal set. Numerous metrics have been proposed for quantifying these features in the literature. Yu and Gen [50] and Zitzler and Thiele [51] proposed some criteria such as number of nondominated solutions (NNS), Error Ratio (ER), and Generational Distance (GD) for measuring accuracy of algorithms. Higher values for NNS and lower values for GD and ER are preferred and show that the algorithm under analysis performed better in providing a set of Pareto optimal solutions. In addition, Diversification Metric (DM), Spacing Metric (SM), and Maximum Spread Metrics (MS) have been proposed to check ability of the algorithms in providing a diverse set of Pareto optimal solutions. In this study we utilized DM and MS to compare algorithms performance which are calculated using

$$DM = \left[\sum_{i=1}^N \max(\|x_i - y_i\|) \right]^{1/2}, \quad (25)$$

$$MS = \left[\sum_{m=1}^M \left(\max_{i=1}^{|N|} f_m^i - \min_{i=1}^{|N|} f_m^i \right)^2 \right]^{1/2},$$

where $\|x_i - y_i\|$ is the Euclidean distance between of the nondominated solution x_i and the nondominated solution y_i and N denote number of the Pareto solutions [52]. The MS presents the distance between the boundary solutions in the obtained nondominated solutions (NDS) where f_m^i is the m th objective function value for i th Pareto solution.

4.3. Numerical Examples. In order to examine performance of the algorithms, three sets of test problems with different sizes (small, medium, and large) are simulated. The dimensions of these sets are shown in Table 4. The properties of the test problems were generated using random functions. All three algorithms are programmed in MATLAB and the test problems have been solved on a PC with 4 GB RAM/1.80 GHz

CPU. Using the algorithm's parameters in Table 3, twenty different problems in each set are solved by NSGA-II, MOPSO, and MOFA. All problems are run for five times and the mean of the obtained values for the metrics is considered for algorithm's performance comparison. In each run, 500 solutions are generated (according to the number of population and maximum iterations) that included infeasible, dominated, and similar solutions. Detected solutions by the algorithms are checked for feasibility firstly and if all the constraints are satisfied, the generated solution is accepted; otherwise another solution would be found. This process has been coded in MATLAB. Mean and standard deviation values of CPU time, NNS, DM, and MS for each case are presented in Table 5.

Four metrics are considered to evaluate algorithms performance (Table 5). These are CPU time, NNS, DM, and MS. The lower values for time and the higher values for the rest of the metrics are desirable. The first rank of the CPU time metric in all cases (small, medium, and large sized instances) belongs to MOPSO approach. It means that MOPSO is more time-efficient. Second rank belongs to NSGA-II and MOFA represents a very long CPU time in comparison with the other competing methods.

It also can be concluded from Table 5 that the average of NNS in the NSGA-II method is higher than other methods. This means that the NSGA-II can generate more nondominated solutions on average. However, MOPSO has provided close values of this metric to NSGA-II in medium and large sized cases.

Obtained values for the DM and MS indicate higher diversity solutions of the produced nondominated solutions by MOPSO in all three cases. This means that the MOPSO method has a wider spread.

Our model has two conflicting objectives, reliability and cost. As the reliability of the system goes higher its cost increases too. We try to find an optimal tradeoff between

TABLE 6: Boundary values for reliability.

Method	Small size problems			Medium size problems			Large size problems		
	NSGA-II	MOPSO	MOFA	NSGA-II	MOPSO	MOFA	NSGA-II	MOPSO	MOFA
Mean	0.991548	0.945431	0.96181	0.985369	0.914826	0.938127	0.99101	0.86809	0.90353
Lower bound	0.981583	0.908162	0.939269	0.970327	0.860268	0.909683	0.98771	0.80074	0.86255
Upper bound	0.996518	0.988152	0.980975	0.994157	0.974125	0.957834	0.99190	0.94698	0.93663

TABLE 7: Boundary values for cost.

Method	Small size problems			Medium size problems			Large size problems		
	NSGA-II	MOPSO	MOFA	NSGA-II	MOPSO	MOFA	NSGA-II	MOPSO	MOFA
Mean	1713.687	1734.326	1870.389	2944.94	3035.20	3189.41	4725.122	4900.882	5166.778
Lower bound	1652.382	1632.584	1790.569	2891.92	2883.30	3070.47	4662.533	4586.47	5015.414
Upper bound	1848.888	2014.234	1987.349	3065.41	3356.64	3346.01	4889.001	5529.174	5369.501

TABLE 8: Mean and standard deviation values for performance metrics.

Algorithm	NSGA-II		MOPSO		MOFA	
	Mean	Std Dev	Mean	Std Dev	Mean	Std Dev
Time	2.74	0.12	1.49	0.06	6.84	0.49
NNS	20.40	5.19	13.20	3.65	3.70	1.06
DM	44.96	10.79	91.46	15.11	30.17	12.21
MS	0.0068	0.0165	0.1436	0.0132	0.0508	0.0218

TABLE 9: Objectives boundary values for the sample.

Method	Reliability			Cost		
	NSGA-II	MOPSO	MOFA	NSGA-II	MOPSO	MOFA
Mean	0.999421	0.87153	0.90719	3848.62	3900.48	4285.62
Lower bound	0.999825	0.802065	0.841807	3760.333	3485.712	4044.209
Upper bound	0.999833	0.977786	0.951062	4027.349	4488.861	4585.62

these objectives. To compare quality of solutions generated by three approaches, mean and lower and upper bounds of objectives are reported in Tables 6 and 7. It should be noted that the average values of the parameters had been calculated in each case.

As it can be concluded from Tables 6 and 7, MOPSO method provides wider ranges of the objective functions but the quality of the solutions which are detected by NSGA-II is better; in other words, solutions with higher system reliability and lower values of cost were generated by NSGA-II.

In remainder of this section, an example is considered based on the proposed model and the results are explained. Tables 11 and 12 provide data of failure rate, Weibull distribution parameters, repair, replacement, and downtime costs, and required human resources (number of operators) for each maintenance action in the mechanical section. The electronic section includes 14 subsystems connected in series in which each subsystem can have 3 or 4 types of components. Also, the mechanical section includes 11 repairable and replaceable subsystems. Table 13 presents upper bounds for some problem parameters.

This example is solved using three approaches for ten times and the mean and standard deviation values for

the performance metrics and objective functions boundary are reported in Tables 8 and 9.

As concluded before, MOPSO method provides better values for the metrics (time, NNS, DM, and MS) and wider boundary values for the objectives but as Table 9 shows, NSGA-II approach detects Pareto solutions lead to higher system reliability and lower cost.

The proposed model in this paper aims at finding proper maintenance policies and effective redundancy strategies. Table 10 represents one of the nondominated solutions that are found by NSGA-II for the mentioned example. The selected maintenance policies and redundancy strategies including number and type of the redundant components and the implemented redundancy strategy in the electronic section along with the selected repair and replacement actions in the mechanical section are shown in Table 10. The first four columns in the electronic section represent the number of redundant components and the last column indicates the implemented redundancy strategy in each subsystem, where letter A means active strategy and letter C indicates cold strategy. In the mechanical subsystem, Xr and Xm indicate replacement and repair action in each period, respectively. Xr and Xm are followed by a number which determines

TABLE 10: Selected maintenance actions and redundancy strategies by the NSGA-II.

Electronic section						Mechanical section															
<i>i</i>	<i>j</i>				Strategy	<i>r</i>	<i>t</i>														
	1	2	3	4			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	1	0	A	1	00	00	00	Xr1	00	Xr2	00	Xr2	00	00	Xr2	Xm1	00	Xr1	00
2	1	1	0	0	C	2	00	00	00	Xr2	00	00	Xr2	00	00	Xr2	00	Xr2	Xr1	Xm2	00
3	1	0	0	0	A	3	00	Xr1	00	Xr1	00	00	Xr1	Xr2	00	00	Xr2	Xr2	Xm2	Xr2	00
4	0	1	0	0	C	4	00	00	00	Xr2	00	00	Xm2	00	00	Xm2	00	Xr2	Xr2	Xm2	00
5	0	1	0	0	A	5	00	Xr1	00	00	Xr2	00	00	Xr1	00	00	Xr1	00	Xr2	Xr1	00
6	0	1	0	1	A	6	00	Xr2	00	Xr2	Xr2	00	00	Xr2	00	00	Xr1	00	Xm2	Xm2	00
7	0	0	1	0	A	7	00	Xr2	00	00	Xm2	00	00	Xr2	00	00	Xr1	00	Xr2	Xm2	Xr2
8	0	1	0	0	A	8	00	Xr2	00	00	Xr2	Xr2	00	00	Xm2	00	00	Xr2	00	Xr2	00
9	1	0	1	0	A	9	00	Xr2	Xm2	Xr2	00	00	Xr2	00	00	Xr1	00	Xr1	00	Xr2	00
10	1	0	0	0	C	10	00	Xr2	00	Xr2	00	00	Xr1	00	00	Xr1	00	Xr2	Xr1	Xr2	00
11	0	0	1	0	C	11	00	Xr2	00	00	Xr2	00	00	Xr2	00	00	Xr2	00	Xr1	Xm2	00
12	2	0	0	1	A																
13	0	1	0	0	C																
14	1	0	0	0	A																

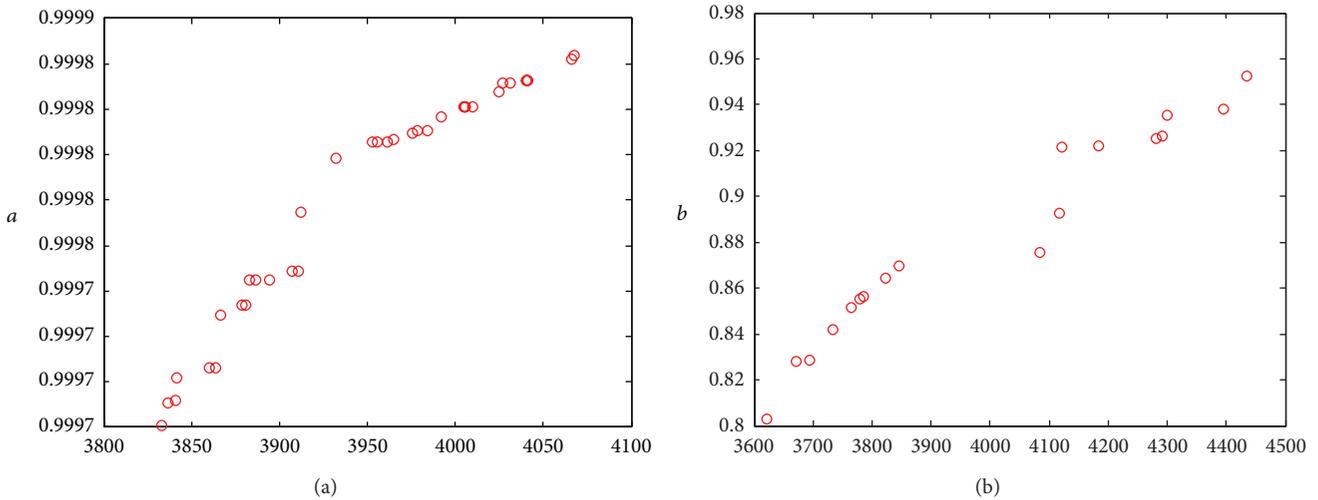


FIGURE 4: Reliability-cost Pareto solutions obtained by (a) NSGA-II and (b) MOPSO.

the type of replacement/repair action. For instance, *Xr2* means replacement type two is required and element 00 indicates that no repair or replacement action has been taken in that period. Figure 4 shows Pareto solutions obtained by NSGA-II and MOPSO methods.

5. Conclusion

In this paper, a biobjective reliability model by integrating redundancy allocation problem (RAP) and reliability-centered maintenance (RCM) problem for a system of non-repairable electronic components and repairable mechanical components has been proposed. Objectives of the problem are maximizing the system reliability and minimizing the system operational and maintenance costs. In order to improve system reliability, active and cold standby redundancy strategies and periodic maintenance actions are considered for

electronic section and mechanical section, respectively. Total system cost includes initial costs for purchasing the required equipment and secondary costs such as operational costs of the electronic section and maintenance costs of the mechanical section. Initial costs are taken into account by setting a budget constraint and operational costs are considered as the second objective of the problem. Three metaheuristic algorithms, NSGA-II, MOPSO, and MOFA, are used to solve the proposed model. Different sets of test problems (small, medium, and large size) have been generated for evaluating solution methods. Obtained results indicate that MOPSO algorithm requires less time to produce Pareto optimal sets with high diversities according to the DM and MS. NSGA-II outperforms MOPSO and MOFA in terms of generating more nondominated solutions (NNS) with better values for system reliability and cost. Finally an example was solved and the obtained results were explained.

TABLE 11: Component data for electronic subsystem.

(a)

i	Choice 1 ($j = 1$)			Choice 2 ($j = 2$)			Choice 3 ($j = 3$)			Choice 4 ($j = 4$)		
	λ_{ij1}	λ_{ij2}	k_{ij}									
1	0.00532	0.004	2	0.000726	0.0012	1	0.00499	0.004	2	0.00818	0.007	3
2	0.00818	0.007	3	0.000619	0.0001	1	0.00431	0.003	2	—	—	—
3	0.0133	0.012	3	0.0110	0.011	3	0.0124	0.011	3	0.00466	0.004	2
4	0.00741	0.006	2	0.0124	0.012	3	0.00683	0.006	2	—	—	—
5	0.00619	0.005	1	0.00413	0.004	2	0.00818	0.007	3	—	—	—
6	0.00436	0.003	3	0.00567	0.005	3	0.00268	0.002	2	0.000408	0.0004	1
7	0.0105	0.01	3	0.00466	0.004	2	0.00394	0.003	2	—	—	—
8	0.0105	0.01	3	0.00105	0.0006	1	0.0105	0.01	3	—	—	—
9	0.00268	0.002	2	0.000101	0.00005	1	0.00408	0.003	1	0.000943	0.00005	1
10	0.0141	0.013	3	0.00683	0.006	2	0.00105	0.0005	1	—	—	—
11	0.00394	0.003	2	0.00355	0.003	2	0.00314	0.002	2	—	—	—
12	0.00236	0.001	1	0.00769	0.007	2	0.0133	0.012	3	0.0110	0.01	3
13	0.00215	0.001	2	0.00536	0.005	3	0.00665	0.006	3	—	—	—
14	0.0110	0.001	3	0.00834	0.003	1	0.00355	0.003	2	0.00436	0.004	3

(b)

i	Choice 1 ($j = 1$)				Choice 2 ($j = 2$)				Choice 3 ($j = 3$)				Choice 4 ($j = 4$)			
	c_{ij}	c'_{ij}	v_{ij}	w_{ij}												
1	10	1	3	3	10	1	4	4	20	2	2	2	20	2	5	5
2	20	2	8	8	10	1	10	10	10	1	9	9	—	—	—	—
3	20	2	7	7	30	3	5	5	10	1	6	6	40	4	4	4
4	30	3	5	5	40	4	6	6	50	5	4	4	—	—	—	—
5	20	2	4	4	20	2	3	3	30	3	5	5	—	—	—	—
6	30	3	5	5	30	3	4	4	20	2	5	5	20	2	4	4
7	40	4	7	7	40	4	8	8	50	5	9	9	—	—	—	—
8	30	3	4	4	50	5	7	7	60	6	6	6	—	—	—	—
9	20	2	8	8	30	3	9	9	40	4	7	7	30	3	8	8
10	40	4	6	6	40	4	5	5	50	5	6	6	—	—	—	—
11	30	3	5	5	40	4	6	6	50	5	6	6	—	—	—	—
12	20	2	4	4	30	3	5	5	40	4	6	6	50	5	7	7
13	20	2	5	5	30	3	5	5	20	2	6	6	—	—	—	—
14	40	4	4	6	40	4	6	7	50	5	6	6	60	6	5	9

Nomenclature

Indices

- i : Electronic components, $i \in \{1, 2, \dots, E\}$
- r : Mechanical components, $r \in \{1, 2, \dots, M\}$
- k : Redundancy strategy; 1 for active and 2 for cold standby, $k \in \{1, 2\}$
- j : Component type in the electronic subsystem, $j \in \{1, 2, \dots, S_i\}$
- u : Component type in the mechanical subsystem, $u \in \{1, 2, \dots, S_r\}$
- S_i : Number of available types for component i in electronic subsystem
- S_r : Number of available types for component r in the mechanical subsystem

- l : Type of maintenance activities performed on component r , $l \in \{1, 2, \dots, L\}$
- t : Time, $t \in \{1, 2, \dots, mT\}$.

Parameters

- V, W : Maximum allowed weight and volume for system
- N_{maxi} : Maximum allowed number of components in subsystem i
- B_0 : Initial available budget to purchase electronic components
- H_0 : Available operators to perform maintenance activities in each period

TABLE 12: Component data for mechanical subsystem.

Parameters	<i>r</i>											
	1	2	3	4	5	6	7	8	9	10	11	
λ_{rt}	—	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004167	0.0004167	0.0004167
$\bar{\lambda}_{rl}$	<i>l</i> = 1	0.00056	0.00056	0.00056	0.00056	0.00056	0.00056	0.00056	0.00056	0.0005834	0.0005834	0.0005834
	<i>l</i> = 2	0.00085	0.00085	0.00085	0.00085	0.00085	0.00085	0.00085	0.00085	0.0008784	0.0008784	0.0008784
$\bar{\lambda}_{ru}$	<i>u</i> = 1	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004167	0.0004167	0.0004167
	<i>u</i> = 2	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005167	0.0005167	0.0005167
α_r	—	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.6	2.6	2.6	2.4
c_{rl}^1	<i>l</i> = 1	2	1.5	2.5	3	1.8	3.5	3	2	2.5	1.5	2
	<i>l</i> = 2	1	1	1.5	1.5	1	2	1.5	1	1.5	0.5	1
c_{ru}^2	<i>u</i> = 1	4	3	5	6	4	7	6	4	3.5	3	4.5
	<i>u</i> = 2	3	1.5	3	2	2	4	3	1.5	2	1.5	2.5
h_{rl}	<i>l</i> = 1	1	1	2	1	2	1	1	2	1	1	2
	<i>l</i> = 2	2	1	1	1	1	2	1	1	2	1	1
h_{ru}	<i>u</i> = 1	1	2	1	2	2	1	2	2	1	2	1
	<i>u</i> = 2	1	2	2	1	2	2	1	2	2	1	2
c_{rl}^3	<i>l</i> = 1	2	2	2	2	2	2	2	2	2	2	2
	<i>l</i> = 2	1.5	1.5	1.5	1	1	1.5	1.5	1	1	1.5	1.5
c_{ru}^4	<i>u</i> = 1	8	7	6	8	9	7	6	5	7	6	9
	<i>u</i> = 2	3	2	5	4	5	5	2	3	3	2	3

TABLE 13: Upper bound of parameters.

Parameter	<i>T</i>	<i>M</i>	$N_{\max i}$	$\lambda_{\max, r}$	B_0	H_o	<i>W</i>	<i>V</i>	η_r	<i>I</i>	$\rho_i(t)$
Value	5	3	4	0.0009	1000	20	200	200	0.001	0.03	0.99

- w_{ij} : Weight of type *j* of component *i* in the electronic subsystem
- v_{ij} : Volume of type *j* of component *i* in the electronic subsystem
- c_{ij} : Purchasing cost for type *j* of component *i*
- c'_{ij} : Operational cost for type *j* of component *i* in active redundancy
- c_{rl}^1 : Cost of repair type *l* for component *r*
- c_{ru}^2 : Replacement cost for type *u* of component *r*
- c_{rl}^3 : Cost of system downtime due to performing repair type *l* on component *r*
- c_{ru}^4 : Cost of system downtime due to replacing for type *u* of component *r*
- h_{rl} : Number of operators required to perform repair type *l* on component *r*
- h_{ru} : Number of operators required to replace type *u* of component *r*
- λ_{ijk} : Failure rate of type *j* of component *i* when redundancy strategy *k* is implemented.
- $\bar{\lambda}_{rl}$: Failure rate of component *r* after repair type *l*
- $\bar{\lambda}_{ru}$: Failure rate of type *u* of component *r* at time zero and after each replacement
- $\lambda_{\max, r}$: Maximum allowed failure rate for each mechanical component in each period
- η_r : Rate of increase in failure rate for mechanical component *r*

- I*: Compound interest rate based on time periods
- T*: System mission time
- m*: Number of inspections during each time unit.

Decision Variables

- x_{ijk} : Number of components *i* with type *j* used in redundancy strategy *k*
- y_{ik} : 1 if redundancy strategy *k* is selected for subsystem *i*; otherwise 0
- xm_{rlt} : If repair type *l* is performed on component *r* in period *t* equals 1; otherwise 0
- xr_{rut} : If type *u* of component *r* is replaced in period *t* equals 1; otherwise 0
- λ_{rt} : Failure rate of component *r* at period *t*.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors are grateful to the anonymous reviewer for the precious and constructive comments which led to significant improvement in the paper.

References

- [1] M. Gen and Y. Yun, "Soft computing approach for reliability optimization: state-of-the-art survey," *Reliability Engineering & System Safety*, vol. 91, no. 9, pp. 1008–1026, 2006.
- [2] L. R. Goel and R. Gupta, "Multi-standby system with repair and replacement policy," *Microelectronics Reliability*, vol. 23, no. 5, pp. 805–808, 1983.
- [3] H. D. Goel, J. Grievink, and M. P. C. Weijnen, "Integrated optimal reliable design, production, and maintenance planning for multipurpose process plants," *Computers and Chemical Engineering*, vol. 27, no. 11, pp. 1543–1555, 2003.
- [4] Y.-T. Tsai, K.-S. Wang, and L.-C. Tsai, "A study of availability-centered preventive maintenance for multi-component systems," *Reliability Engineering and System Safety*, vol. 84, no. 3, pp. 261–270, 2004.
- [5] D. K. Mohanta, P. K. Sadhu, and R. Chakrabarti, "Deterministic and stochastic approach for safety and reliability optimization of captive power plant maintenance scheduling using GA/SA-based hybrid techniques: a comparison of results," *Reliability Engineering & System Safety*, vol. 92, no. 2, pp. 187–199, 2007.
- [6] S. Martorell, M. Villamizar, S. Carlos, and A. Sánchez, "Maintenance modeling and optimization integrating human and material resources," *Reliability Engineering & System Safety*, vol. 95, no. 12, pp. 1293–1299, 2010.
- [7] A. Certa, G. Galante, T. Lupo, and G. Passannanti, "Determination of Pareto frontier in multi-objective maintenance optimization," *Reliability Engineering & System Safety*, vol. 96, no. 7, pp. 861–867, 2011.
- [8] R. Moghaddass, M. J. Zuo, and M. Pandey, "Optimal design and maintenance of a repairable multi-state system with standby components," *Journal of Statistical Planning and Inference*, vol. 142, no. 8, pp. 2409–2420, 2012.
- [9] M. Doostparast, F. Kolahan, and M. Doostparast, "A reliability-based approach to optimize preventive maintenance scheduling for coherent systems," *Reliability Engineering & System Safety*, vol. 126, pp. 98–106, 2014.
- [10] F. J. Samaniego, *System Signatures and their Applications in Engineering Reliability*, vol. 110 of *International Series In Operations Research & Management Science*, Springer, New York, NY, USA, 2007.
- [11] M. H. Moradi and A. Khandani, "Evaluation economic and reliability issues for an autonomous independent network of distributed energy resources," *International Journal of Electrical Power and Energy Systems*, vol. 56, pp. 75–82, 2014.
- [12] A. Oyarbide-Zubillaga, A. Goti, and A. Sanchez, "Preventive maintenance optimisation of multi-equipment manufacturing systems by combining discrete event simulation and multi-objective evolutionary algorithms," *Production Planning & Control*, vol. 19, no. 4, pp. 342–355, 2008.
- [13] P. Hilber, V. Miranda, M. A. Matos, and L. Bertling, "Multiobjective optimization applied to maintenance policy for electrical networks," *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 1675–1682, 2007.
- [14] M. S. Chern, "On the computational complexity of reliability redundancy allocation in a series system," *Operations Research Letters*, vol. 11, no. 5, pp. 309–315, 1992.
- [15] D. W. Coit, "Cold-standby redundancy optimization for nonrepairable systems," *IIE Transactions*, vol. 33, no. 6, pp. 471–478, 2001.
- [16] R. Zhao and B. Liu, "Stochastic programming models for general redundancy-optimization problems," *IEEE Transactions on Reliability*, vol. 52, no. 2, pp. 181–191, 2003.
- [17] Y.-C. Liang and A. E. Smith, "An ant colony optimization algorithm for the redundancy allocation problem (RAP)," *IEEE Transactions on Reliability*, vol. 53, no. 3, pp. 417–423, 2004.
- [18] R. Tavakkoli-Moghaddam, J. Safari, and F. Sassani, "Reliability optimization of series-parallel systems with a choice of redundancy strategies using a genetic algorithm," *Reliability Engineering & System Safety*, vol. 93, no. 4, pp. 550–556, 2008.
- [19] S. J. Sadjadi and R. Soltani, "An efficient heuristic versus a robust hybrid meta-heuristic for general framework of serial—parallel redundancy problem," *Reliability Engineering and System Safety*, vol. 94, no. 11, pp. 1703–1710, 2009.
- [20] R. Kumar, K. Izui, M. Yoshimura, and S. Nishiwaki, "Multi-objective hierarchical genetic algorithms for multilevel redundancy allocation optimization," *Reliability Engineering & System Safety*, vol. 94, no. 4, pp. 891–904, 2009.
- [21] N. Beji, B. Jarboui, M. Eddaly, and H. Chabchoub, "A hybrid particle swarm optimization algorithm for the redundancy allocation problem," *Journal of Computational Science*, vol. 1, no. 3, pp. 159–167, 2010.
- [22] E. Zio and R. Bazzo, "Level diagrams analysis of pareto front for multiobjective system redundancy allocation," *Reliability Engineering & System Safety*, vol. 96, no. 5, pp. 569–580, 2011.
- [23] B. Soylu and S. K. Ulusoy, "A preference ordered classification for a multi-objective max–min redundancy allocation problem," *Computers & Operations Research*, vol. 38, no. 12, pp. 1855–1866, 2011.
- [24] J. Safari, "Multi-objective reliability optimization of series-parallel systems with a choice of redundancy strategies," *Reliability Engineering and System Safety*, vol. 108, pp. 10–20, 2012.
- [25] K. Khalili-Damghani and M. Amiri, "Solving binary-state multi-objective reliability redundancy allocation series-parallel problem using efficient epsilon-constraint, multi-start partial bound enumeration algorithm, and DEA," *Reliability Engineering and System Safety*, vol. 103, pp. 35–44, 2012.
- [26] A. Chambari, S. H. A. Rahmati, A. A. Najafi, and A. Karimi, "A bi-objective model to optimize reliability and cost of system with a choice of redundancy strategies," *Computers & Industrial Engineering*, vol. 63, no. 1, pp. 109–119, 2012.
- [27] H. Zoulfaghari, A. Z. Hamadani, and M. A. Ardakan, "Bi-objective redundancy allocation problem for a system with mixed repairable and non-repairable components," *ISA Transactions*, vol. 53, no. 1, pp. 17–24, 2014.
- [28] D. Cao, A. Murat, and R. B. Chinnam, "Efficient exact optimization of multi-objective redundancy allocation problems in series-parallel systems," *Reliability Engineering & System Safety*, vol. 111, pp. 154–163, 2013.
- [29] H. Garg and S. P. Sharma, "Multi-objective reliability-redundancy allocation problem using particle swarm optimization," *Computers & Industrial Engineering*, vol. 64, no. 1, pp. 247–255, 2013.
- [30] X. S. Yang, *Nature-Inspired Metaheuristic Algorithm*, Luniver Press, 2010.
- [31] L. dos Santos Coelho, D. L. de Andrade Bernert, and V. C. Mariani, "A chaotic firefly algorithm applied to reliability-redundancy optimization," in *Proceedings of the IEEE Congress of Evolutionary Computation (CEC '11)*, pp. 517–521, New Orleans, La, USA, June 2011.

- [32] S. J. Sadjadi and R. Soltani, "Alternative design redundancy allocation using an efficient heuristic and a honey bee mating algorithm," *Expert Systems with Applications*, vol. 39, no. 1, pp. 990–999, 2012.
- [33] T.-J. Hsieh and W.-C. Yeh, "Penalty guided bees search for redundancy allocation problems with a mix of components in series—parallel systems," *Computers and Operations Research*, vol. 39, no. 11, pp. 2688–2704, 2012.
- [34] L. D. Afonso, V. C. Mariani, and L. D. S. Coelho, "Modified imperialist competitive algorithm based on attraction and repulsion concepts for reliability-redundancy optimization," *Expert Systems with Applications*, vol. 40, no. 9, pp. 3794–3802, 2013.
- [35] L. Wang and L.-P. Li, "A coevolutionary differential evolution with harmony search for reliability-redundancy optimization," *Expert Systems with Applications*, vol. 39, no. 5, pp. 5271–5278, 2012.
- [36] D. E. Fyffe, W. W. Hines, and N. K. Lee, "System reliability allocation and a computational algorithm," *IEEE Transactions on Reliability*, vol. 17, no. 2, pp. 64–69, 1968.
- [37] G. Kanagaraj, S. G. Ponnambalam, and N. Jawahar, "A hybrid cuckoo search and genetic algorithm for reliability-redundancy allocation problems," *Computers & Industrial Engineering*, vol. 66, no. 4, pp. 1115–1124, 2013.
- [38] M. A. Ardakan and A. Z. Hamadani, "Reliability-redundancy allocation problem with cold-standby redundancy strategy," *Simulation Modelling Practice and Theory*, vol. 42, pp. 107–118, 2014.
- [39] J. A. Ruiz-Vanoye and O. Díaz-Parra, "Similarities between meta-heuristics algorithms and the science of life," *Central European Journal of Operations Research*, vol. 19, no. 4, pp. 445–466, 2011.
- [40] G. Levitin, L. Xing, and Y. Dai, "Cold vs. hot standby mission operation cost minimization for 1-out-of- N systems," *European Journal of Operational Research*, vol. 234, no. 1, pp. 155–162, 2014.
- [41] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.
- [42] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE Conference on Neural Networks*, Piscataway, NJ, USA, 1998.
- [43] C. A. Coello Coello and M. S. Lechuga, "MOPSO: a proposal for multiple objective particle swarm optimization," in *Proceedings of the Congress on Evolutionary Computation (CEC '02)*, pp. 1051–1056, Piscataway, NJ, USA, May 2002.
- [44] C. A. Coello Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, 2004.
- [45] K. E. Parsopoulos and M. N. Vrahatis, "Multiobjective particle swarm optimization approaches," in *Multi-Objective Optimization in Computational Intelligence: Theory and Practice*, chapter 2, pp. 20–42, University of New South Wales, Sydney, Australia, 2008.
- [46] M. K. Sayadi, A. Hafezalkotob, and S. G. Naini, "Firefly-inspired algorithm for discrete optimization problems: an application to manufacturing cell formation," *Journal of Manufacturing Systems*, vol. 32, no. 1, pp. 78–84, 2013.
- [47] X.-S. Yang, "Multiobjective firefly algorithm for continuous optimization," *Engineering with Computers*, vol. 29, no. 2, pp. 175–184, 2013.
- [48] A. A. Najafi, S. T. A. Niaki, and M. Shahsavari, "A parameter-tuned genetic algorithm for the resource investment problem with discounted cash flows and generalized precedence relations," *Computers & Operations Research*, vol. 36, no. 11, pp. 2994–3001, 2009.
- [49] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [50] X. Yu and M. Gen, in *Introduction to Evolutionary Algorithms*, chapter 6, Springer, London, UK, 2010.
- [51] E. Zitzler and L. Thiele, "Multi-objective optimization using evolutionary algorithms—a comparative case study," in *Parallel Problem Solving from Nature—PPSN V*, vol. 1498 of *Lecture Notes in Computer Science*, pp. 292–301, Springer, Berlin, Germany, 1998.
- [52] K. Khalili-Damghani, A.-R. Abtahi, and M. Tavana, "A new multi-objective particle swarm optimization method for solving reliability redundancy allocation problems," *Reliability Engineering & System Safety*, vol. 111, pp. 58–75, 2013.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

