



Research Article

Hybrid NSGA-II based decision-making in fuzzy multi-objective reliability optimization problem

Hemant Kumar¹  · Shiv Prasad Yadav¹

Received: 16 March 2019 / Accepted: 17 October 2019 / Published online: 26 October 2019
© Springer Nature Switzerland AG 2019

Abstract

This paper suggests a hybrid NSGA-II based decision-making method in a fuzzy multi-objective reliability optimization problem. Multi-objective evolutionary algorithms (MOEAs) are popular techniques to be solved various kinds of multi-objective optimization problems efficiently. NSGA-II is one of the elitist MOEAs, which is largely used in engineering design problems. The reliability-based system design problem comprises various kinds of uncertainties such as expert's information character, qualitative statements, vagueness, incompleteness, unclear system boundaries, etc. Fuzzy optimization techniques can be useful during the initial stage of the conceptual design of a system. In many complex problems, it is not possible to produce the entire Pareto-optimal set in one simulation run. Apart from this, getting a well diverse solution set is another important phenomenon in this field. The proposed approach finds the optimal system design by resolving these issues in a fuzzy multi-objective reliability optimization problem. A numerical example of the over-speed protection system consisting of three mutually conflicting objectives such as system reliability, system cost, and system weight is considered with several design constraints to illustrate the method. Finally, the results obtained by the proposed approach are discussed with the existing approach.

Keywords System reliability · NSGA-II · Pareto-optimal front (POF) · Fuzzy set theory · Local search · Clustering

1 Introduction

Practically, reliability enhancement may be involved with other mutually conflicting objectives such as system's cost, weight, volume, etc. Multi-objective optimization problems (MOOPs) are an evitable part of the reliability-based system design. Conventional optimization methods assume that all the design data involved in the system design are precisely known, the constraints delimit a well-defined set of feasible decisions, and the objectives are well defined and easy to formulate. However, incompleteness and unreliability of input information are typical for many practical problems of multi-objective optimization decision-making of reliability. Park [1] suggested that a fuzzy non-linear optimization technique is a superior way to analyze the system reliability. The fuzzy approach [2] deals with the kind

of uncertainty that arises due to imprecision associated with the complexity of the system as well as the vagueness of human judgment. Dhingra and Moskowitz [3] used fuzzy set theory effectively in engineering design problems where uncertainty or ambiguity arises about the preciseness of permissible parameters, degree of credibility, and correctness of statements and judgments. Bellman and Zadeh [4] initially proposed a fuzzy optimization technique, which combines fuzzy goals and fuzzy decision space by using aggregate operators. Dhingra [5], Rao and Dhingra [6] solved fuzzy multi-objective optimization problems in four and five stages series system subject to several design constraints. Huang [7] developed a fuzzy multi-objective decision-making method in the series system. Ravi et al. [8] proposed various kinds of aggregate operators to look into the impacts on their optimal system design in complex

✉ Hemant Kumar, hkumar1@ma.iitr.ac.in; Shiv Prasad Yadav, yadavfma@iitr.ac.in | ¹Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand 247667, India.



systems. Huang et al. [9] proposed a coordination method in solving fuzzy multi-objective optimization of system reliability. Mahapatra and Roy [10] proposed a fuzzy multi-objective optimization method in the decision-making of multi-objective reliability optimization. Garg and Sharma [11] also report that the probabilistic approach is unsuitable to reliability analysis due to its random nature and model the multi-objective reliability-redundancy allocation problem in a fuzzy environment.

In order to solve the MOOPs, generally weighted sum approach is used [12]. This method finds one Pareto solution at a time by assigning a weight to each objective function. Such a method is not fit for producing a well distributive solution set which is considered an ideal approach to an MOOP. Apart from this, such methods require tedious work to find multiple solutions, adequately showing limitations in the search space like plateaus, or ridges in the fitness (search) landscape, or discontinuous. A set of well-distributive trade-off optimal solutions is an important demand for ideal multi-objective optimization. This procedure is more practical, methodical and less subjective [12]. Keeping these views in mind, a number of MOEAs such as MOGA-Multi-Objective Genetic Algorithm [13], NPGA-Niched Pareto Genetic Algorithm [14], NSGA-Non-dominated Sorting Genetic Algorithm [15], SPEA-Strength Pareto Evolutionary Algorithm [16], PAES-Pareto Archived Evolution Strategy [17], PESA-Pareto Envelop-based Selection Algorithm [18], MOMGA-Multiobjective Optimization with Messy Genetic Algorithm [19], PESA-II [20], SPEA2 [21], NSGA-II [22], MOEA/D-Multi-objective Evolutionary Algorithm based on Decomposition [23], AGE-Approximation Guided Evolutionary-II [24], NSGA-III [25] etc., are suggested to tackle the different types of the MOOPs. MOEAs and their solution approaches can be viewed in Deb [12], Coello et al. [26], and Das and Panigrahi [27]. A more updated survey on MOEAs can be viewed in Zhang and Xing [28].

NSGA-II is a second-generation MOEA, effectively used by Salazar et al. [29] in constrained multi-objective reliability and redundancy problems to identify a Pareto-optimal set. Later, Wang et al. [30] solved the multi-objective reliability redundancy allocation problem (MORRAP) using NSGA-II and compared their results with single-objective approaches. Kishore et al. [31, 32] suggested fuzzy multi-objective reliability optimization using NSGA-II. Safari [33] proposed a variant of NSGA-II in solving an MORRAP. Vitorino et al. [34] tackled the reliability problem in multi-objective power distribution system using NSGA-II. Sharifi et al. [35] solved MORRAP for a series-parallel problem and k-out-of-n subsystems with 3 objectives using NSGA-II. Kumar et al. [36–39] analyzed the reliability optimization problem using NSGA-II. In recent work, Muhuri et al. [40] addressed the problem of MORRAP using NSGA-II with an interval type-2 fuzzy set.

In this paper, an approach is developed to find the optimal system design from the POF. The optimization goals are set in the following fashion:

- The distance of the resulting POF to the true POF should be minimized.
- The obtained Pareto-optimal solutions should be uniformly distributed.
- The spread of the obtained POF should be maximized, i.e., each objective function should have a wide range of values.

To achieve the above goals, hybrid NSGA-II is proposed for fuzzy multi-objective reliability optimization. The optimal system design is found in a purely multi-objective manner where no aggregate operator is used and a well distributive solution set is obtained. Finally, a fuzzy ranking method [41] is used to obtain the best compromise solution from the solution set. A comparative experiment has been performed to show its efficacy with the existing approach.

The rest of the paper is organized as follows. Section 2 gives some preliminaries. Section 3 gives a brief description of NSGA-II. Section 4 gives the proposed approach in step by step. Section 5 gives the problem statement of the over-speed protection system. Section 6 gives the results and discussion and Sect. 7 gives conclusions.

2 Preliminaries

In this section, some basic facts of this paper are given.

2.1 Formulation of an MOOP

In general, an MOOP is defined as follows:

$$\begin{aligned} & \text{Minimize } F(X) = [f_1(X), f_2(X), \dots, f_k(X)]^T \\ & \text{subject to } g_i(X) = 0, \quad i = 1, 2, \dots, m_e; \\ & g_i(X) \geq 0, \quad i = m_e + 1, \quad m_e + 2, \dots, m; \\ & x_j^l \leq x_j \leq x_j^u, \quad j = 1, 2, \dots, n \end{aligned} \quad (1)$$

where $k \geq 2$ is the number of objectives; m is the total number of constraints; m_e is the number of equality constraints; $X = [x_1, x_2, \dots, x_n]^T$ is n -dimensional decision vector from the feasible region or decision space Ω defined by

$$\Omega = \left\{ X \in \mathbb{R}^n \mid \begin{array}{l} g_i(X) \geq 0, \quad g_i(X) = 0, \quad x_j^l \leq x_j \leq x_j^u, \\ j = 1, 2, \dots, n \end{array} \right\}$$

The image of the feasible region is denoted by $Z \subset \mathbb{R}^k$ and it is called a feasible objective region or objective space which is defined by $Z = \{F(X) \in \mathbb{R}^k \mid X \in \Omega\}$. The elements of Z are called objective vectors or criterion vector denoted by $z = F(X) = [f_1(X), f_2(X), \dots, f_k(X)]^T$; x_j^l and

x_j^u are the lower and upper bounds of the decision variable x_j respectively. For every point X in the decision space Ω , there exists a point $F(X)$ in the objective space Z . It is a mapping between the n -dimensional solution vector and the k -dimensional objective vector.

2.2 Basic definitions

The concept of optimality in an MOOP depends on *Pareto optimality*. Therefore, the following definitions can be defined in terms of Pareto terminology [26].

Definition 1 (Pareto dominance) A vector $U = [u_1, u_2, \dots, u_k]^T$ is said to dominate another vector $V = [v_1, v_2, \dots, v_k]^T$ (denoted by $U \leq V$) if and only if U is partially less than V , i.e., $\forall r \in \{1, \dots, k\}, u_r \leq v_r$ and \exists at least one $\hat{r} \in \{1, \dots, k\}$ s.t. $u_{\hat{r}} < v_{\hat{r}}$.

Definition 2 (Pareto optimality) A solution vector $X \in \Omega$ is said to be Pareto-optimal with respect to Ω if and only if there is no $X' \in \Omega$ for which $V = [f_1(X'), f_2(X'), \dots, f_k(X')]^T$ dominates $U = [f_1(X), f_2(X), \dots, f_k(X)]^T$. The phrase "Pareto-optimal" is taken to mean with respect to the entire decision variable space unless otherwise specified.

Definition 3 (Pareto-optimal set) For a given MOOP, $F(X)$, the Pareto-optimal set, \wp^* , is defined as:

$$\wp^* := \{X \in \Omega \mid \neg \exists X' \in \Omega \text{ s.t. } F(X') \leq F(X)\}$$

Definition 4 (Pareto-optimal front) For a given MOOP, $F(X)$, and the Pareto-optimal set, \wp^* , the Pareto-optimal front $\wp^{\mathcal{F}}$ is defined as:

$$\wp^{\mathcal{F}} := \left\{ [f_1(X), f_2(X), \dots, f_k(X)]^T \mid X \in \wp^* \right\}$$

Definition 5 (Ideal objective vector) [26] If a decision vector $X^* = [x_1^*, x_2^*, \dots, x_n^*]^T \in \Omega$ is such that $f_i(X^*) = \min_{X \in \Omega} f_i(X)$; $i \in \{1, 2, \dots, k\}$ then the vector $F(X^*) = [f_1(X^*), f_2(X^*), \dots, f_k(X^*)]^T \in Z$ is called an ideal objective vector for an MOOP given by (1) and $X^* = [x_1^*, x_2^*, \dots, x_n^*]^T \in \Omega$ is called an ideal vector.

Definition 6 (Fuzzy set) [2] Let X be a collection of objects generically denoted by x . A fuzzy set \tilde{A} in X is a set of ordered pair defined in the form as

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \mid x \in X\}$$

where $\mu_{\tilde{A}}: X \rightarrow [0, 1]$ is called the membership function and its value is called the grade of membership of x in \tilde{A} .

Definition 7 [Linguistic hedge (or modifier)] [37] A linguistic hedge (or a linguistic modifier) is an operation that modifies the meaning of the term. Suppose \tilde{H} is a fuzzy set

in X , then the modifier m generates the composite term $\tilde{M} = m(\tilde{H})$. Modifiers are frequently used in mathematical models as follows.

Concentration It decreases the membership grades of all the members of \tilde{H} by spreading in the curve. It is defined as:

$$\mu_{con(\tilde{H})}(x) = \left(\mu_{(\tilde{H})}(x)\right)^2 \quad \text{for all } x \in X$$

Dilation Similarly, it increases the membership grades of all members by spreading out the curve. It is defined as:

$$\mu_{dil(\tilde{H})}(x) = \left(\mu_{(\tilde{H})}(x)\right)^{1/2} \quad \text{for all } x \in X$$

Therefore, in general, strong and weak modifiers are given as: $m_{\delta}(\mu_{(\tilde{H})}(x)) = \left(\mu_{(\tilde{H})}(x)\right)^{\delta}$ = a strong modifier or concentrator, if $\delta > 1$ and a weak modifier or dilator if $\delta < 1$.

The following linguistic hedges are associated with above mathematical operators: *very* $\tilde{H} = con(\tilde{H})$; *more or less* $\tilde{H} = dil(\tilde{H})$; *Indeed* $\tilde{H} = Int(\tilde{H})$; *plus* $\tilde{H} = \tilde{H}^{1.25}$; *slightly* $\tilde{H} = int[\text{plus } \tilde{H} \text{ and not (very } \tilde{H})]$.

3 Elitist non-dominated sorting genetic algorithm (NSGA-II)

Srinivas and Deb [15] proposed the NSGA which is an early dominance-based MOEA. The purpose of developing NSGA is to find better solutions according to their non-domination levels. NSGA uses the naive and slow [12] sorting approach to distribute a population into different non-domination levels, and a sharing function method to maintain the diversity of the population. However, it has high computational complexity $O(kN^3)$, where k is the number of objectives and N is the population size in the non-dominated sorting. So, NSGA is computationally expensive for large population sizes. Moreover, NSGA is a non-elitist approach that affects its convergence rates compared to other MOEAs and it also requires a sharing parameter to calculate the sharing fitness, which ensures the diversity of the population.

Deb et al. [22] proposed NSGA-II to overcome the drawbacks of NSGA. Specifically, NSGA-II presents a fast-non-dominated sorting approach with the worst-case computational complexity $O(k(2N)^2)$. This approach searches iteratively non-dominated solutions into different fronts.

First, for each solution i in the population, the algorithm calculates two entities:

1. n_i , the number of solutions dominating i ,
2. S_i , a set of solutions dominated by i .

The solutions for which $n_i = 0$ belong to the first front. Second, for each member j in the set S_i , the value of n_j is reduced by one. If any n_j is reduced to zero during this stage, the corresponding member j is put in the second front. The above process is continued with each member in the second front to identify the third front and so on. Furthermore, NSGA-II applies the concept of crowding-distance with the worst-case computational complexity $O(k(2N) \log(2N))$. The introduction of crowding-distance replaces the fitness sharing approach that requires a sharing parameter to be set by the user.

The crowding-distance value (CD_i) (See Fig. 1) of the i th solution is calculated as follows:

$$d_{ip} = \frac{f_p^{i+1} - f_p^{i-1}}{f_p^{max} - f_p^{min}} \tag{2}$$

$$CD_i = \sum_{p=1}^k d_{ip} \tag{3}$$

where f_p^{i+1} and f_p^{i-1} denote the p th objective function of the $(i + 1)$ th and $(i - 1)$ th individual (solution) respectively, and f_p^{max} and f_p^{min} represent the maximum and minimum values of the p th objective function.

According to Deb et al. [22] “A higher value of crowding-distance gives the lesser crowded region and vice versa”. After applying the fast-non-dominated sorting procedure, the crowding-distance picks those solutions which locate in the less-crowded region or possesses a higher value of crowding distance. This procedure is extended up to the entire POF for maintaining the diversity in the solutions set. Finally, NSGA-II applies an elitist strategy with the worst-case computational complexity of $O(2N \log(2N))$. The elitist strategy [42, 43] improves the convergence of an MOEA and avoids the loss of optimal solutions after getting it.

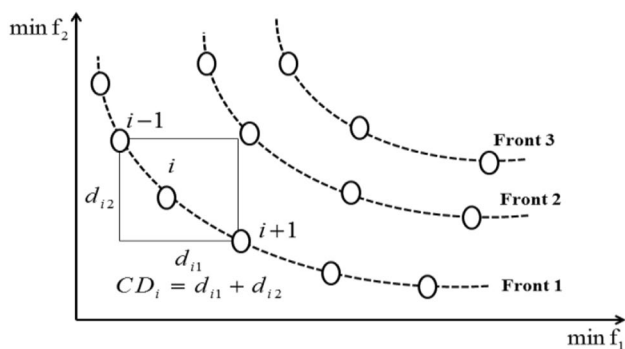


Fig. 1 Fitness evaluation and individual crowding distance estimation

For handling the constraints, the binary tournament selection method is applied. A search space (decision space) is categorized into two regions—feasible and infeasible. According to Deb et al. [22], “a solution X is called a constrained-dominate to a solution Y if

- X is feasible and Y is infeasible.
- X and Y are infeasible, but X contains a smaller overall constraint violation.
- X and Y are feasible, but X dominates Y ”.

In Fig. 2, the evaluation cycle of the NSGA-II is shown. First, an offspring Q_t of size N is obtained by using genetic operators such as selection, recombination, and mutation. A combined population R_t of size $2N$ is then formed which consists of the current population P_t and the offspring population Q_t . By using fast non-dominated sorting, R_t is divided into different fronts PF_1, PF_2, \dots, PF_n . Let the number of solutions in each front PF_i be N_i . Next, we choose members for the new population P_{t+1} from the front PF_1 to PF_{t-1} , noting that $N_1 + N_2 + \dots + N_t > N$ and $N_1 + N_2 + \dots + N_{t-1} \leq N$. Afterwards, to get the exactly N population members in P_{t+1} , we sort the solutions in front PF_t using the crowding distance sorting procedure and choose the best solutions to fill an empty slot in the new population P_{t+1} . This process is continued until the termination condition is satisfied. The flow diagram of NSGA-II is shown in Fig. 3.

4 Proposed methodology

This section describes the proposed methodology in step by step as follows.

Step 1 Determine the best and worst values for each objective function.

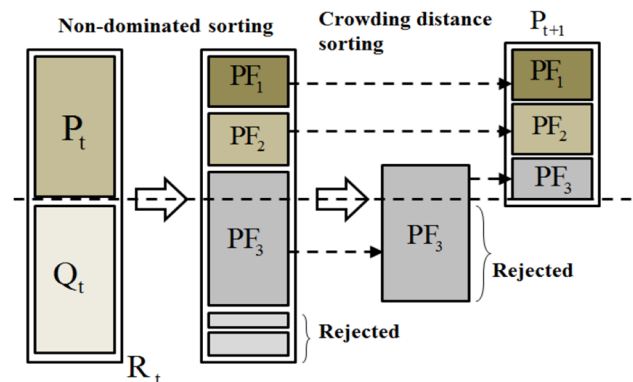


Fig. 2 An evaluation cycle of the NSGA-II algorithm

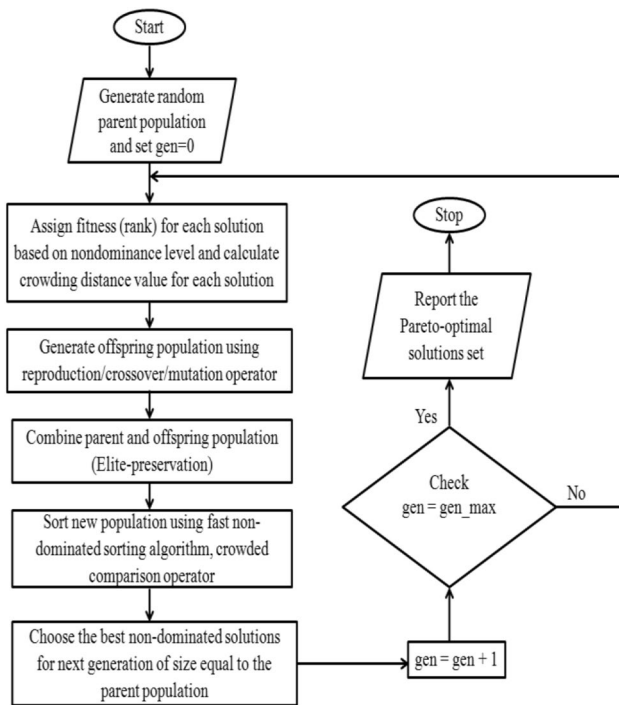


Fig. 3 Flow diagram of the NSGA-II

$$M = \begin{bmatrix} f_1(X_1^*) & f_2(X_1^*) & \dots & f_k(X_1^*) \\ f_1(X_2^*) & f_2(X_2^*) & \dots & f_k(X_2^*) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(X_k^*) & f_2(X_k^*) & \dots & f_k(X_k^*) \end{bmatrix} \tag{5}$$

where the diagonal entries of matrix M are found to be the minima in their respective columns. The best m_p and worst M_p values of the p th objective function f_p are identified as:

$$m_p = f_p(X^*) = \min_p f_p(X_p^*) \tag{6}$$

$$M_p = f_p(X^*) = \max_p f_p(X_p^*) \tag{7}$$

Step 2 Construct the membership function.

Here, the sigmoidal shape is chosen for the membership function in the given problem because “When the decision-maker (DM) is worse off with respect to a goal for the objective function, the DM tends to have a higher marginal rate of satisfaction with respect to that objective function. A convex shape captures that behavior in the membership function. On the other hand, when the DM is better off with respect to a goal, the DM tends to have a smaller marginal rate of satisfaction. Such behavior is modeled using the concave portion of the membership function” [5]. It is defined as:

$$\mu_H(f_p) = \begin{cases} 1, & \text{if } f_p \leq m_p \\ \frac{1}{2} \tanh\left(\left(\frac{M_p + m_p}{2} - f_p\right)\alpha_p\right) + \frac{1}{2}, & \text{if } m_p \leq f_p \leq M_p; \quad \alpha_p = 6/(M_p - m_p) \\ 0, & \text{if } f_p \geq M_p \end{cases} \tag{8}$$

Each objective function is solved as a single-objective optimization problem (SOOP) one by one to get its boundaries as follows:

$$\begin{aligned} &\text{Minimize } f_p(X); \quad p = 1, 2, \dots, k \\ &\text{subject to } g_i(X) = 0, \quad i = 1, 2, \dots, m_e; \\ &g_i(X) \geq 0, \quad i = m_e + 1, m_e + 2, \dots, m; \\ &x_j^l \leq x_j \leq x_j^u, \quad j = 1, 2, \dots, n \end{aligned} \tag{4}$$

Let us consider $X_i^*, i = 1, 2, \dots, k$ be the optimal design vector for each objective function f_i , then construct a matrix as

Step 3 Reformulate the MOOP as a fuzzy MOOP.

Each membership function is expected to achieve the maximum satisfaction level. Therefore, a fuzzy MOOP is suggested to maximize all the membership functions so that it could get the maximum satisfaction level simultaneously [39]. From the extreme values of f_p obtained by (6) and (7), a fuzzy MOOP can be given as:

$$\text{Maximize } \mu(F) = [(\mu_1(f_1), \mu_2(f_2), \dots, \mu_k(f_k))]^T$$

or, Minimize

$$\begin{aligned} &\mu(F) = [(-\mu_1(f_1), -\mu_2(f_2), \dots, -\mu_k(f_k))]^T \\ &\text{subject to } g_i(X) = 0, \quad i = 1, 2, \dots, m_e; \\ &g_i(X) \geq 0, \quad i = m_e + 1, m_e + 2, \dots, m; \\ &x_j^l \leq x_j \leq x_j^u, \quad j = 1, 2, \dots, n \end{aligned} \tag{9}$$

Theorem 1 *The Pareto-optimal solution of the fuzzy MOOP (9) satisfies the MOOP (1).*

Proof Let X^* be a Pareto-optimal solution of the fuzzy MOOP (9). Then, it is defined by definition as follows:

- $\nexists X \in \Omega$ s.t. $-\mu_i(f_i)(X) \leq -\mu_i(f_i)(X^*) \forall i = 1, 2, \dots, k$, and $-\mu_j(f_j)(X) < -\mu_j(f_j)(X^*)$ for at least one $j \in \{1, 2, \dots, k\}, j \neq i$.
- $\nexists X \in \Omega$ s.t. $-h_i[f_i(X)] \leq -h_i[f_i(X^*)] \forall i = 1, 2, \dots, k$, and $-h_j[f_j(X)] < -h_j[f_j(X^*)]$ for at least one $j \in \{1, 2, \dots, k\}, j \neq i$.
- $\nexists X \in \Omega$ s.t. $f_i(X) \leq f_i(X^*) \forall i = 1, 2, \dots, k$, and $f_j(X) < f_j(X^*)$ for at least one $j \in \{1, 2, \dots, k\}, j \neq i$; since h is monotonically decreasing function.
- $\nexists X \in \Omega$ s.t. $-f_i(X) \geq -f_i(X^*) \forall i = 1, 2, \dots, k$, and $-f_j(X) > -f_j(X^*)$ for at least one $j \in \{1, 2, \dots, k\}, j \neq i$; since h is monotonically increasing function.
- $\nexists X \in \Omega$ s.t. $f_i(X) \leq f_i(X^*) \forall i = 1, 2, \dots, k$, and $f_j(X) < f_j(X^*)$ for at least one $j \in \{1, 2, \dots, k\}, j \neq i$; since h is either monotonically decreasing or increasing function.
- X^* is a Pareto-optimal solution of the MOOP given by (1). \square

Step 4 Biased Pareto-optimal solutions.

The proposed approach does not find a single compromise solution; instead, it finds a biased distribution of solutions. If a user wants to have a bias towards a particular objective, the modified membership function produces more solutions towards the preferred region in the search space. Finding more dense solutions in the region of preference is a much better approach than predefining a weighted sum of objective and then finding only one the optimum solution. Here, the linguistic modifier is proposed to show biasedness towards the objective.

Step 5 Set the parameters of NSGA-II and apply to the fuzzy MOOP (9) with modification of linguistic modifier.

The NSGA-II parameters such as number of objectives, number of design constraints, number of design variables, population size, the maximum number of generations, crossover probability, mutation probability, distribution indices [12] for SBX crossover and Polynomial mutation are set accordingly to the given problem. Rigorous experimentation and tuning of the parameters are required to achieve a well-distributive Pareto-optimal solution set.

Step 6 Find the Pseudo-weight vector.

The purpose of local search can be fulfilled by finding a pseudo weight to each Pareto point. It is calculated as follows:

$$w_i = \frac{\mu_i(f_i)}{\sum_{j=1}^k \mu_j(f_j)}, \quad i = 1, 2, \dots, k \tag{10}$$

$w = [w_1, w_2, \dots, w_k]^T$ represents the Pseudo-weight vector dictating roughly the priority of different objective functions for the solution $X, \sum_{i=1}^k w_i = 1, w_i \geq 0$.

Step 7 Start the local search.

After getting the pseudo-weight to each Pareto-point, the local search strategy is used to update the Pareto-optimal solution. It helps in enhancing the convergence rate to each Pareto point. The local search strategy is given as follows.

The crisp formulation of fuzzy MOOP (9) is given by

$$\begin{aligned} & \text{Maximize } (1 \wedge \mu_1(f_1)/w_1) \wedge (1 \wedge \mu_2(f_2)/w_2) \wedge \dots \wedge (1 \wedge \mu_k(f_k)/w_k) \\ & \text{subject to } g_i(X) = 0, i = 1, 2, \dots, m_e; \\ & g_i(X) \geq 0, i = m_e + 1, m_e + 2, \dots, m; \\ & x_j^l \leq x_j \leq x_j^u, \quad j = 1, 2, \dots, n \end{aligned} \tag{11}$$

where $\mu(F) = [(\mu_1(f_1), \mu_2(f_2), \dots, \mu_k(f_k))]^T$ is a vector of k -fuzzy regions of satisfaction corresponding to k -objective vector, \wedge denotes the aggregate operator known as intersection and it is defined as the minimum operator i.e., $(1 \wedge \mu_i(f_i)/w_i)$ gives the minimum of two values 1 and $\mu_i(f_i)/w_i$ that value lies in $[0, 1]$.

Step 8 Apply the clustering technique.

In order to maintain a good spread in the solutions set, a clustering technique is used. It also reduces the size of the solutions set that is useful for a practical point of view [12]. Let P_{t+1} be the set of non-dominated solutions of size N .

The clustering algorithm (see Fig. 4) is given as follows:

Step 1 Initially, each solution belongs to a distinct cluster or $C_i = \{i\}$ so that $C = \{C_1, C_2, \dots, C_N\}$.

Step 2 If $N \leq N'$ (required no. of solutions), go to **Step 5**. Otherwise, go to **Step 3**.

Step 3 For each pair of clusters (there are $\binom{|C|}{2}$ of them), calculate the cluster-distance by using the following formula:

$$D_{12} = (1/|C_1||C_2|) \sum_{i \in C_1, j \in C_2} D(i, j) \tag{12}$$

where $|C_1|$ and $|C_2|$ number of solutions in the cluster set C_1 and C_2 respectively, and $D(i, j)$ is the Euclidean distance between two solutions i and j . Find the pair (i_1, i_2) which corresponds to the minimum cluster-distance.

Step 4 Merge the clusters C_{i_1} and C_{i_2} together. This deducts the size of C by one. Go to **Step 2**.

Step 5 Select only one solution from each cluster and remove others. The solution having the minimum average distance from other solutions in the cluster represents a solution to that cluster.

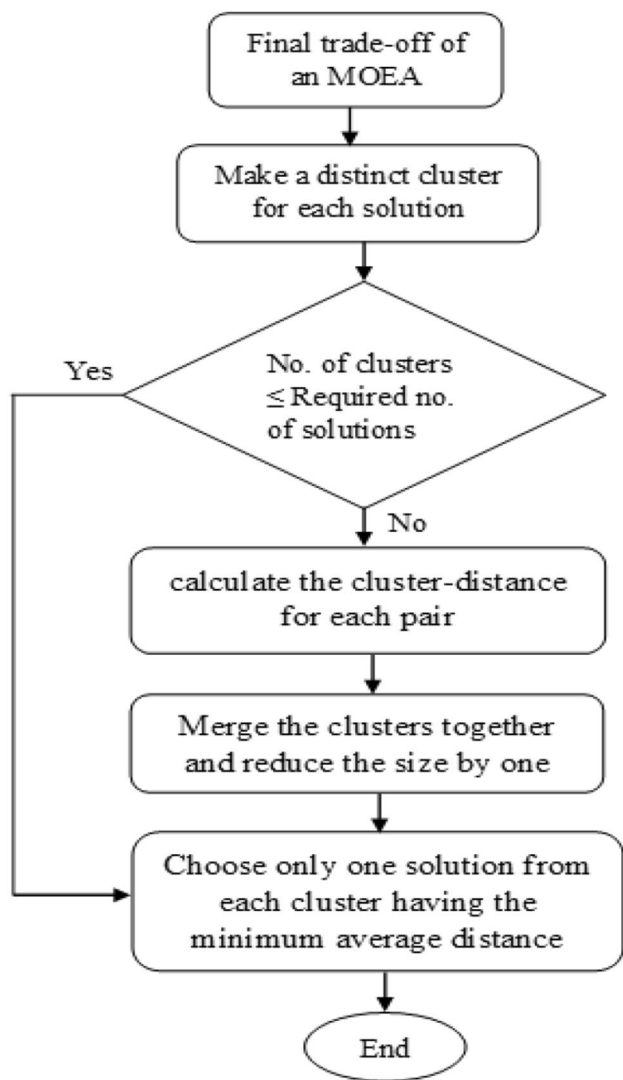


Fig. 4 Flow diagram of the clustering technique

Step 9 Choose the best compromise solution.

The fuzzy ranking method [41] is used to rank the solutions as per their satisfaction levels. In this way, a solution achieves the highest degree of satisfaction is considered as the best compromise solution. It is given as follows.

$$\mu_{best} = \max_p [\min\{\mu_1(f_1), \mu_2(f_2), \dots, \mu_k(f_k)\}] \quad (13)$$

where P is the number of Pareto-optimal solutions.

5 Problem statement

Let us consider a four-stage over-speed protection system for a gas turbine [5]. Here, the electrical and mechanical systems continuously provide detection. If an over-speed occurs, then 4-control valves (V1–V4) get closed and the

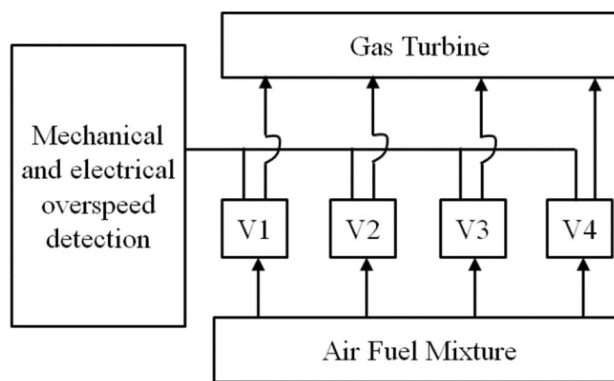


Fig. 5 A schematic diagram of a four-stage over-speed protection system

fuel supply gets interrupted. Each component has a constant failure rate in the system. A schematic representation of the over-speed protection system is shown in Fig. 5. This system is a series–parallel system configuration having several parallels and identical components arrayed at each stage. It is shown in Fig. 6. The objective is to determine the optimal design variables i.e., r_i (component reliability) and n_i (number of the redundant components) at stage i such that:

1. the system reliability (R_S) is maximized,
2. the system cost (C_S) is minimized,
3. the system weight (W_S) is minimized.

In addition, several design constraints such as the minimum requirement for the system reliability R_{lim} , overall system cost of the system C_{lim} , the total permissible volume of the system V_{lim} , and maximum allowable weight of the system W_{lim} are considered.

The mathematical formulation of the over-speed protection system is given as:

$$\text{Maximize } f_1 \equiv R_S(r, n) = \prod_{i=1}^4 [1 - (1 - r_i)^{n_i}] \quad (14)$$

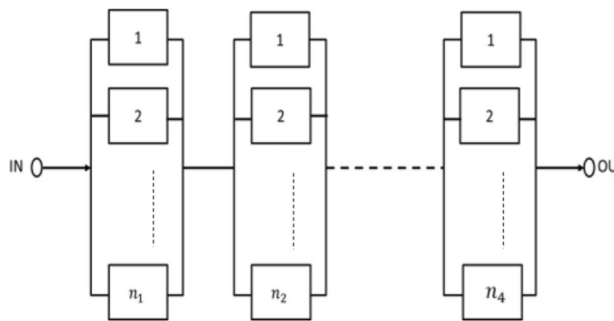


Fig. 6 Series-parallel system configuration

$$\text{Minimize } f_2 \equiv C_s(r, n) = \sum_{i=1}^4 c_i(r_i) [n_i + \exp(n_i/4)] \quad (15)$$

$$\text{Minimize } f_3 \equiv W_s(r, n) = \sum_{i=1}^4 w_i n_i \exp(n_i/4) \quad (16)$$

$$\text{subject to } g_1 \equiv \sum_{i=1}^4 v_i n_i^2 \leq V_{lim} \quad (17)$$

$$g_2 \equiv \sum_{i=1}^4 w_i n_i \exp(n_i/4) \leq W_{lim} \quad (18)$$

$$g_3 \equiv \prod_{i=1}^4 [1 - (1 - r_i)^{n_i}] \geq R_{lim} \quad (19)$$

$$g_4 \equiv \sum_{i=1}^4 \gamma_i [-T / \ln(r_i)]^{\delta_i} [n_i + \exp(n_i/4)] \leq C_{lim} \quad (20)$$

$$0.5 \leq r_i \leq 1 - 10^{-6}, 1 \leq n_i \leq 10, i = 1, 2, 3, 4; \quad (21)$$

“Each component of the system has a constant failure rate λ_i that follows an exponential distribution” [5]. The reliability of each component is given as:

$$r_i(T) = \int_T^\infty \lambda_i e^{-\lambda_i T} dT = e^{-\lambda_i T} \quad (22)$$

It is also mentioned that “ $c_i(r_i)$ is assumed to be an increasing function of r_i (conversely, a decreasing function of the component failure rate)” [5] in the form:

$$c_i(r_i) = \gamma_i / \lambda_i^{\delta_i} \quad (23)$$

From (22) and (23), we have

$$c_i(r_i) = \gamma_i [-T / \ln(r_i)]^{\delta_i} \quad (24)$$

where the parameters δ_i and γ_i are the physical feature (shaping and scaling factor) of the cost-reliability curve

of each component in the i th subsystem; T is the active operational time; v_i is the volume of a component at stage i ; w_i is the weight of a component at stage i ; the factor $\exp(n_i/4)$ is responsible for the additional cost due to the interconnection between the parallel components. The design data of the given problem is listed in Table 1. The given problem is an MORRAP which contains a series-parallel system configuration. This system contains several parallels and identical components arrayed at each stage. Generally, redundancy is applied to increase the system reliability but this technique gives more complexity in terms of cost, weight, and volume to the system design. So, it is suitable to adopt the multi-objective programming model-based problem. Here, two-fold design variables are required to determine the optimal design of the system. One is the reliability of each component and the other is to select a number of redundant components at each stage. Moreover, several design constraints are considered in this model. Such type of problem is a mixed-integer, non-linear programming problem as well as NP-hard [44] for which only approximate solutions have been proposed [45, 46]. Heuristic approaches to such types of problems can be viewed in [47–50].

6 Results and discussion

After applying the proposed approach, this section describes the numerical results. Initially, the lower and upper limits on each objective are evaluated with the given constraints using MATLAB optimization tool-box function namely “fmincon” [51]. Here, integer variables n_i are initially treated as real variables but during the evaluation of the objective functions, the real values are transformed into the nearest integer values. The results of the SOOPs are given in Table 2 and compared with heuristic methods such as Genetic Algorithm (GA) [52], Particle Swarm Optimization (PSO) [53], and Dhingra’s approach [5]. It gives the best and worst solutions to each of the objective functions. These values are used as boundaries in the membership function of the objective.

All experiments are performed in MATLAB (R2017a) on Intel(R) Core (TM) i3-2370 M CPU @ 2.40 GHz with 4 GB RAM. The parameter settings for the MOEAs presented in this study are as follows.

Table 1 Design data for the given problem

Stage	$10^5 \gamma_i$	δ_i	v_i	w_i	R_{lim}	C_{lim}	W_{lim}	V_{lim}	T
1	1.0	1.5	1	6	0.75	400	500	250	1000 h
2	2.3	1.5	2	6					
3	0.3	1.5	3	8					
4	2.3	1.5	2	7					

Table 2 The optimal solutions for each objective function

	MATLAB optimization function "fmincon"		GA		PSO		Dhingra's approach	
	Design variables	Attributes	Design variables	Attributes	Design variables	Attributes	Design variables	Attributes
Max R_S	(0.91403, 5)	$R_S=0.99994$	(0.92472, 5)	$R_S=0.99810$	(0.91513, 6)	$R_S=0.9996$	(0.81604, 6)	$R_S=0.99961$
	(0.86723, 6)	$C_S=399.52$	(0.87732, 3)	$C_S=396.95$	(0.87883, 4)	$C_S=393.92$	(0.80309, 6)	$C_S=399.94$
	(0.94282, 3)	$W_S=439.02$	(0.83430, 6)	$W_S=434.05$	(0.94312, 5)	$W_S=442.30$	(0.98364, 3)	$W_S=495.65$
	(0.87349, 5)	$V_S=174$	(0.91449, 4)	$V_S=183$	(0.87959, 4)	$V_S=175$	(0.80373, 5)	$V_S=185$
Min C_S	(0.5, 4)	$R_S=0.7541$	(0.5, 4)	$R_S=0.7608$	(0.53447, 4)	$R_S=0.7615$	(0.5, 4)	$R_S=0.7604$
	(0.5, 4)	$C_S=20.30$	(0.5, 4)	$C_S=20.62$	(0.5, 4)	$C_S=20.71$	(0.5, 4)	$C_S=20.72$
	(0.54536, 5)	$W_S=314.55$	(0.53150, 5)	$W_S=314.55$	(0.51812, 5)	$W_S=314.55$	(0.59251, 5)	$W_S=314.55$
	(0.5, 3)	$V_S=141$	(0.51457, 3)	$V_S=141$	(0.5, 3)	$V_S=141$	(0.5, 3)	$V_S=141$
Min W_S	(0.95638, 1)	$R_S=0.83681$	(0.91623, 1)	$R_S=0.7972$	(0.94566, 1)	$R_S=0.8318$	(0.96221, 1)	$R_S=0.80786$
	(0.94286, 1)	$C_S=396.63$	(0.96254, 1)	$C_S=397.89$	(0.95262, 1)	$C_S=397.73$	(0.92315, 1)	$C_S=399.51$
	(0.97705, 1)	$W_S=34.67$	(0.95627, 1)	$W_S=34.67$	(0.97205, 1)	$W_S=34.67$	(0.98787, 1)	$W_S=34.67$
	(0.94984, 1)	$V_S=8$	(0.94526, 1)	$V_S=8$	(0.94984, 1)	$V_S=8$	(0.92065, 1)	$V_S=8$

1. NSGA-II: Population size $N=80$; crossover probability $p_c=0.9$; mutation probability $p_m=1/n$, where n is the number of design variables; the distribution indices [22] for crossover and mutation operators as $\eta_c=10$ and $\eta_m=100$; maximum number of generation $t_{max}=100$.
2. SPEA2 [21]: Population size $N=80$; $p_c=0.9$, $p_m=1/n$; and $\eta_c=10$, $\eta_m=100$, archive size = 80; $t_{max}=100$.
3. PESA-II [20]: Hyper-grid size = 10×10 ; $p_c=0.9$, $p_m=1/n$; and $\eta_c=10$, $\eta_m=100$, archive size = 80; $t_{max}=100$.
4. MOPSO [54]: Inertia weight $w=0.8$; acceleration constants $c_1=c_2=2$; archive size = 80; $t_{max}=100$.

In order to solve SOOP, the parameter settings for GA are taken as selection type = binary tournament, pop_size = 30, crossover rate = 0.9, mutation rate = 0.01, max_gen = 100; in PSO swarm size = 30, cognition learning factors as $c_1=c_2=2$, maximum and minimum inertia weights 0.9 and 0.4 respectively. To avoid the stochastic discrepancy, 25

independent runs are made to these algorithms. The given problem is solved by Dhingra [5] using a fuzzy optimization technique (Zimmerman's approach) [2], goal programming [12], goal attainment [25] methods. The intention was to show how the fuzzy optimization technique can be effective to solve such type of a problem. The sigmoidal shape is chosen for each membership function. At the same time, the preference is given to reliability more than the system cost and the system weight of the system. It is mentioned by Dhingra [5] that the reliability is twice as important as cost and weight. Linguistic hedge "very" is used instead of assigning the weight. The shape of the membership function of the objectives is as shown in Fig. 7a and b. Apart from this, crisp multi-objective reliability optimization is shown by goal programming, goal attainment methods by assigning the weights to the objectives as $w_1=0.5$, $w_2=0.25$, $w_3=0.25$. Huang's approach [7] is also applied, where the fuzzy optimization technique combines the

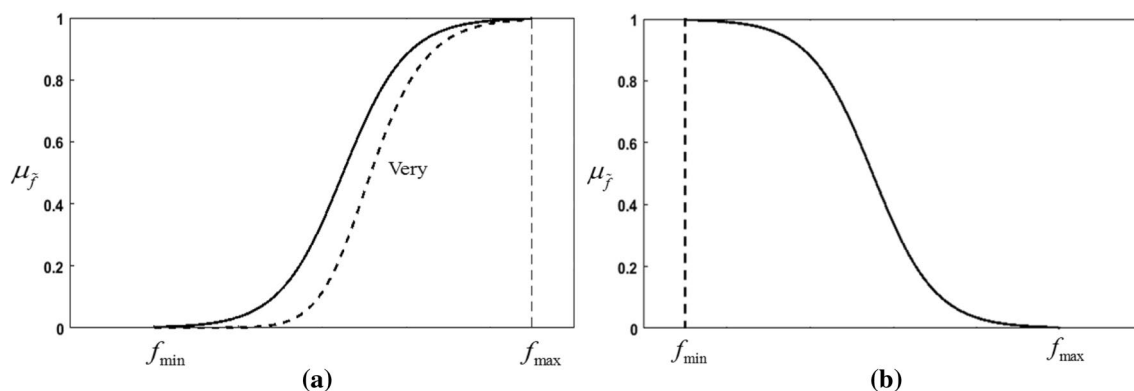


Fig. 7 Membership function for **a** monotonically increasing function i.e. system reliability; **b** monotonically decreasing functions i.e. system cost and weight

degree of satisfaction to each of the objectives using the “min” operator. In this approach, the weights are given as $w_1 = 1, w_2 = 0.5, w_3 = 0.5$ following the non-convexity. In fuzzy optimization, the assigned weight to the objective function may affect the result especially the case of convexity and non-convexity search space. It is advantageous to adopt the ideal approach by taking the whole POF instead of one solution. That is why the proposed approach does not use any kind of aggregation and finds the solution set in one simulation run. The biasedness of the POF is shown by the linguistic hedge “very” to the reliability while system cost and its weight are taken with no hedge. This behavior is used in the proposed methodology to make the results effective in terms of better POF. This preference is the same as Dhingra [5]. The proposed approach gives the solution set in the desired region. This is an ideal multi-objective optimization procedure to search for the optimal system design [12]. In Fig. 8a, firstly, multiple fronts are shown in the membership values (degree of satisfactions) by applying NSGA-II to fuzzy MOOP. In Fig. 8b, all the fronts are compared with respect to the highest degree of satisfaction. In Fig. 9a, the degree of satisfactions lead to find the solutions in the objective space. From here, the search for

the optimal front starts using local search which is later maintained the diversity in the solution set with the help of the clustering technique. In Fig. 9b, the optimal front of the hybrid NSGA-II is compared with first front of the NSGA-II. The obtained optimal front is comparatively shown with other MOEAs in Fig. 10a and b. In Fig. 11a and b, the optimal front of the proposed approach is compared with MOPSO and the preference-based approach (goal programming, goal attainment, Huang’s approach, Zimmerman’s approach) [12] respectively. In Table 3, a well diverse solution set is shown by the proposed approach. In Table 4, the list of the optimal system design obtained by the various approaches is given. Table 5 gives a comparative analysis of the hypervolume metric [55] of the POFs obtained by the various approaches. This performance is based on 10 independent runs of each multi-objective algorithm.

7 Conclusions

In this piece of work, a hybrid NSGA-II based decision-making is proposed for fuzzy multi-objective optimization of system reliability. The results are shown by taking

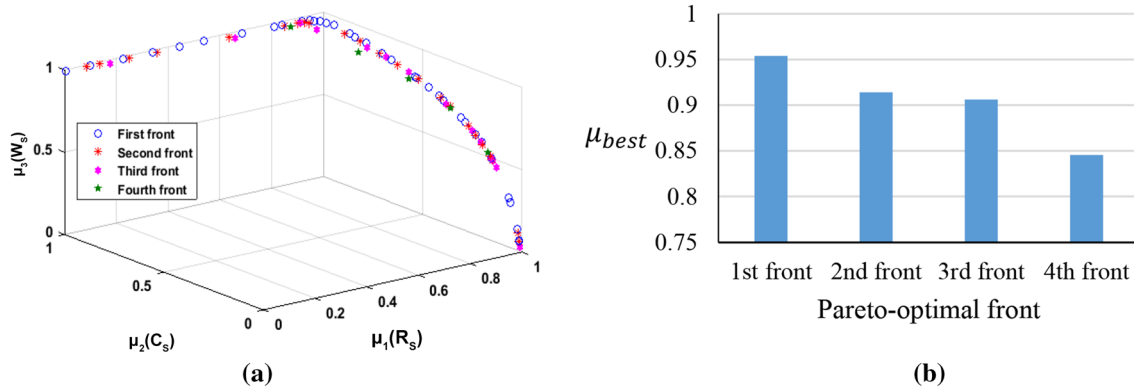


Fig. 8 **a** Multiple fronts are shown in the membership values; **b** satisfaction level achieved by multiple fronts

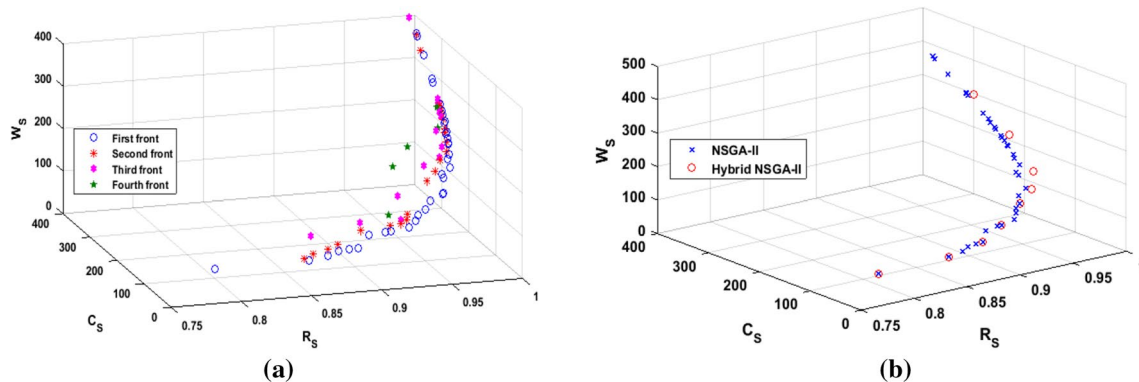


Fig. 9 **a** Multiple fronts are shown in the objective space; **b** The optimal front obtained by the proposed approach is compared with the first front of NSGA-II

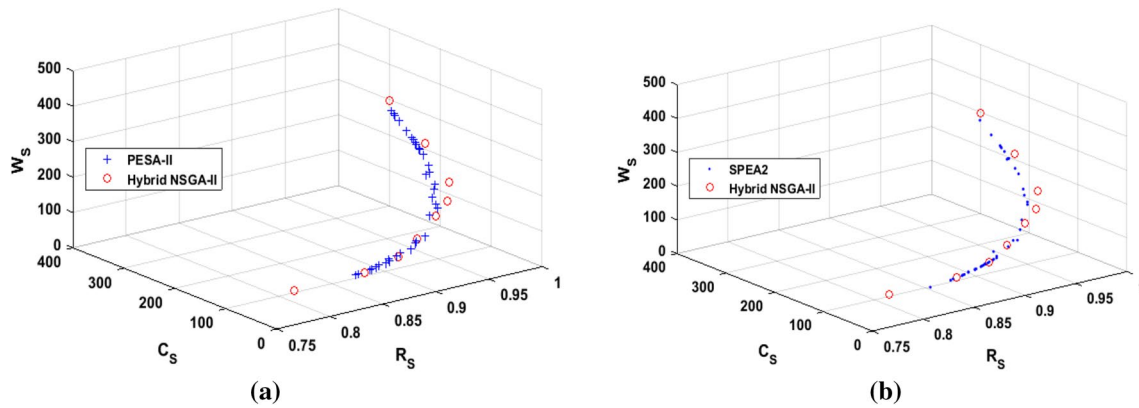


Fig. 10 The optimal front by the proposed approach is compared with other MOEAs **a** PESA-II; **b** SPEA2

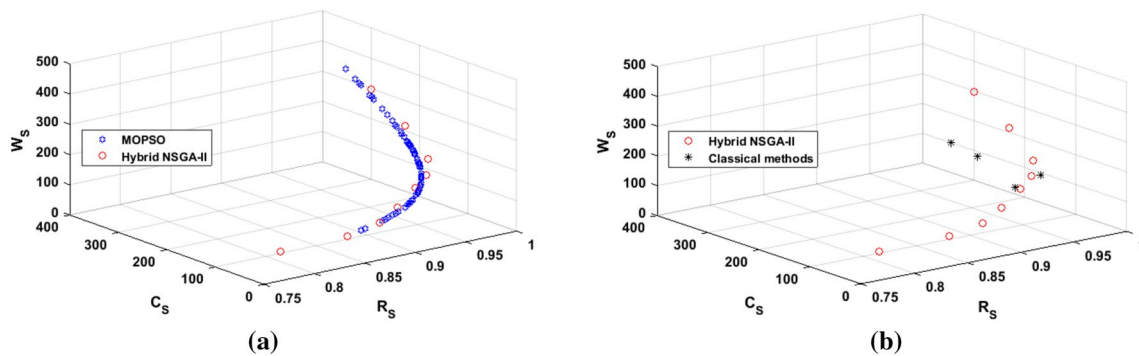


Fig. 11 The optimal front by the proposed approach is compared with **a** MOPSO; **b** classical (preference-based) approach

Table 3 The trade-off optimal solutions or Pareto-optimal solutions obtained by the proposed approach

	$\mu_1(R_S)$	$\mu_2(C_S)$	$\mu_3(W_S)$	R_S	C_S	W_S	V_S	r_1	n_1	r_2	n_2	r_2	n_3	r_4	n_4
1	0.9950	0.0572	0.3082	0.9997	298.82	298.69	128	0.8898	4	0.8802	4	0.9123	4	0.8812	4
2	0.9944	0.3870	0.7637	0.9974	224.70	221.85	72	0.8806	3	0.8707	3	0.8959	3	0.8716	3
3	0.9914	0.8335	0.9463	0.9884	159.19	156.08	72	0.8882	3	0.8399	3	0.9201	3	0.8441	3
4	0.9830	0.9210	0.9730	0.9741	132.44	128.34	52	0.8610	3	0.8425	3	0.8937	2	0.8514	2
5	0.9547	0.9570	0.9826	0.9532	111.99	110.92	42	0.8522	2	0.8393	2	0.8772	2	0.8403	2
6	0.8256	0.9812	0.9910	0.9228	85.01	85.02	42	0.7290	3	0.6822	3	0.7191	3	0.6932	3
7	0.5297	0.9900	0.9947	0.8955	64.75	64.34	32	0.7280	3	0.7183	3	0.7200	3	0.6917	3
8	0.0953	0.9931	0.9961	0.8582	52.91	52.40	32	0.6936	3	0.6958	2	0.7290	3	0.6842	3
9	0.0003	0.9936	0.9963	0.7909	50.52	50.35	32	0.7561	2	0.5353	3	0.8252	4	0.7282	2

three mutually conflicting objectives such as system reliability, system cost and system weight subject to several design constraints. The conclusions of the proposed work can be drawn as follows.

- From Table 2, a design with high system reliability has a high total cost and a large weight. While attempting to minimize the system cost, it results in heavy design with low reliability. When the system weight is mini-

mized, the resulting system has a high cost and low reliability.

- Single-objective optimization is unable to give compromise solution among these mutually conflicting objectives.
- From Table 3, a reasonable number of “compromise” or “trade-off” optimal solutions are obtained which can be useful in practical practices.
- The proposed technique is seen as an ideal multi-objective optimization procedure, where a set of well-distributive solutions is obtained.

Table 4 The Pareto-optimal results by the classical (preference-based) and MOEAs approaches

Methods	R_s	C_s	W_s	V_s	r_1	n_1	r_2	n_2	r_2	n_3	r_4	n_4	μ_{best}
Goal programming	0.9849	312.83	128.73	47	0.9433	3	0.8928	2	0.9535	2	0.8913	3	–
Goal attainment	0.9974	287.19	89.03	32	0.9413	2	0.9092	2	0.9408	2	0.9129	2	–
Huang’s approach	0.9708	107.14	147.05	57	0.8421	3	0.8049	3	0.8939	2	0.8098	3	0.9570
Zimmerman’s approach	0.9448	104.47	128.73	52	0.8650	2	0.8181	3	0.8425	2	0.8064	3	0.9357
NSGA-II	0.9528	112.51	112.47	100	0.7148	4	0.6823	3	0.8187	4	0.6845	3	0.9540
PESA-II	0.9492	114.85	114.81	72	0.7869	3	0.7289	3	0.7473	3	0.8186	3	0.9456
SPEA2	0.9534	118.16	118.16	107	0.6803	4	0.6539	4	0.8653	3	0.6860	4	0.9482
MOPSO	0.9566	109.26	109.30	114	0.7634	4	0.6955	4	0.7829	4	0.7356	3	0.9604
Hybrid NSGA-II	0.9532	111.98	110.92	42	0.8522	2	0.8393	2	0.8772	2	0.8403	2	0.9547

Table 5 Comparison of performance calculated by the hypervolume metric

	Best	Worst	Mean	Median	SD
NSGA-II	32,914.83	32,884.89	32,900.77	32,903.05	10.2906
PESA-II	32,716.84	32,716.84	32,700.04	32,696.10	11.0171
SPEA2	33,060.14	33,041.78	33,052.08	33,052.44	6.5572
MOPSO	32,969.86	32,943.78	32,953.38	32,950.39	8.2685
Hybrid NSGA-II	33,061.43	33,042.34	33,052.58	33,052.56	6.4312

- From Table 4, the hybrid technique finds the optimal system design which satisfies a comparatively maximum degree of satisfaction compared to other MOEAs.
- From Table 5, a statistical analysis of the hypervolume metric performance is shown which concludes that the proposed approach gives a better distribution of solution set with minimum deviation compared to the others.
- The preference-based approach does not follow the ideal multi-objective optimization procedure. In fact, it behaves like a random walk in the search space.
- However, in our experiment, MOPSO achieves the maximum degree of satisfaction (see Table 4), but slightly worse than the proposed approach with respect to the hypervolume metric (see Table 5).
- The proposed approach can be effective in the decision-making of a fuzzy multi-objective reliability-based system design problem.

Acknowledgements The authors acknowledge the financial grant provided by the Ministry of Human Resource Development (MHRD), Government of India, New Delhi for this study.

Funding This study was funded by the Ministry of Human Resource Development (MHRD) (Grant Number MHR 01-23-200-428).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Park KS (1987) Fuzzy apportionment of system reliability. *IEEE Trans Reliab* 36:129–132
2. Zimmermann HJ (1996) Fuzzy set theory and its applications. Kluwer Academic Publishers, Boston. ISBN 0-7923-9624-3
3. Dhingra AK, Moskowitz H (1991) Application of fuzzy theories to multiple objective decision making in system design. *Eur J Oper Res* 53:348–361. [https://doi.org/10.1016/0377-2217\(91\)90068-7](https://doi.org/10.1016/0377-2217(91)90068-7)
4. Bellman R, Zadeh LA (1970) Decision making in a fuzzy environment. *Manag Sci* 17:141–164
5. Dhingra AK (1992) Optimal apportionment of reliability and redundancy in series systems under multiple objectives. *IEEE Trans Reliab* 41(4):576–582
6. Rao SS, Dhingra AK (1992) Reliability and redundancy apportionment using crisp and fuzzy multiobjective optimization approaches. *Reliab Eng Syst Saf* 37:253–261
7. Huang HZ (1997) Fuzzy multi-objective optimization decision-making reliability of series system. *Microelectron Reliab* 37(3):447–449
8. Ravi V, Reddy PJ, Zimmermann HJ (2000) Fuzzy global optimization of complex system reliability. *IEEE Trans Fuzzy Syst* 8(3):241–248
9. Huang HZ, Wu WD, Liu CS (2005) A coordination method for fuzzy multi-objective optimization of system reliability. *J Intell Fuzzy Syst* 16:213–220
10. Mahapatra GS, Roy TK (2006) Fuzzy multi-objective mathematical programming on reliability optimization model. *Appl Math Comput* 174:643–659
11. Garg H, Sharma SP (2013) Multi-objective reliability-redundancy allocation problem using particle swarm optimization. *Comput Ind Eng* 64:247–255. <https://doi.org/10.1016/j.cie.2012.09.015>
12. Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, New York

13. Fonseca CM, Fleming PJ (1993) Genetic algorithms for multi-objective optimization: formulation, discussion, and generalization, genetic algorithms. In: Proceedings of the fifth international conference, pp 416–423
14. Horn J, Nafpliotis N, Goldberg DE (1994) A niched Pareto genetic algorithm for multiobjective optimization. In: Proceedings of the first IEEE conference on evolutionary computation. IEEE world congress on computational intelligence, vol 1, pp 82–87. <https://doi.org/10.1109/icc.1994.350037>
15. Srinivas N, Deb K (1994) Multiobjective optimization using nondominated sorting in genetic algorithms. *Evol Comput* 2(3):221–248
16. Zitzler E, Thiele L (1998) An evolutionary algorithm for multi-objective optimization: the strength Pareto approach. Technical report 43, Zurich, Switzerland: Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH)
17. Knowles J, Corne D (1999) The Pareto archived evolution strategy : a new baseline algorithm for Pareto multiobjective optimisation. In: Proceedings of the 1999 congress on evolutionary computation. IEEE Press, Piscataway, pp 98–105. <https://doi.org/10.1109/cec.1999.781913>
18. Corne DW, Knowles JD, Oates MJ (2000) The Pareto envelope-based selection algorithm for multiobjective optimization. In: Schoenauer M, Deb K, Rudolph G, Yao X, Lutten E, Merelo JJ, Schwefel H-P (eds) Proceedings of the parallel problem solving from nature VI conference. Springer. LNCS no. 1917, pp 839–848
19. Veldhuizen DV, Lamont GB (2000) Multiobjective optimization with messy genetic algorithms. In: Proceedings of the 2000 ACM symposium on applied computing, pp 470–476
20. Corne DW, Jerram NR, Knowles JD, Martin J (2001) PESA-II: region-based selection in evolutionary multiobjective optimization. In: Proceeding GECCO'01 proceedings of the 3rd annual conference on genetic and evolutionary computation, pp 283–290
21. Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength Pareto evolutionary algorithm. Technical report no. 35, Comput Eng Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Gloriastrasse 35, CH-8092, Zurich, Switzerland. <https://doi.org/10.3929/ethz-a-004284029>
22. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6:182–197. <https://doi.org/10.1109/4235.996017>
23. Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 11:712–731
24. Wagner M, Neumann F (2013) A fast approximation-guided evolutionary multi-objective algorithm. In: Proceedings of the 15th annual conference on genetic and evolutionary computation 2013, pp 687–694
25. Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE Trans Evol Comput* 18(4):577–601
26. Coello CAC, Lamont GB, Van Veldhuizen DA (2007) Evolutionary algorithms for solving multi-objective problems. Springer, New York
27. Das S, Panigrahi BK (2009) Multi-objective evolutionary algorithms. *Encycl Artif Intell* 3:1145–1151
28. Zhang J, Xing L (2017) A survey of multiobjective evolutionary algorithms. In: Proceedings of the 2017 IEEE international conference on computational science and engineering IEEE/IFIP international conference on embedded and ubiquitous computing CSE EUC 2017, vol 1, pp 93–100
29. Salazar D, Rocco CM, Galván BJ (2006) Optimization of constrained multiple-objective reliability problems using evolutionary algorithms. *Reliab Eng Syst Saf* 91:1057–1070. <https://doi.org/10.1016/j.res.2005.11.040>
30. Wang Z, Chen T, Tang K, Yao X (2009) A multi-objective approach to redundancy allocation problem in parallel-series systems. *IEEE*, pp 582–589
31. Kishor A, Yadav SP, Kumar S (2009) A multi-objective genetic algorithm for reliability optimization problem. *Int J Perform Eng* 5(3):227–234
32. Kishor A, Yadav SP, Kumar S (2009) Interactive fuzzy multi-objective reliability optimization using NSGA-II. *OPSEARCH* 46:214–224. <https://doi.org/10.1007/s12597-009-0013-2>
33. Safari J (2012) Multi-objective reliability optimization of series-parallel systems with a choice of redundancy strategies. *Reliab Eng Syst Saf* 108:10–20. <https://doi.org/10.1016/j.res.2012.06.001>
34. Vitorino RM, Jorge HM, Neves LP (2015) Multi-objective optimization using NSGAII for power distribution system reconfiguration. *Int Trans Electr Energy Syst* 25:38–53
35. Sharifi M, Guilani PP, Shahriari M (2016) Using NSGA-II algorithm for a three objective redundancy allocation problem with k-out-of-n sub-systems. *J Optim Ind Eng* 19:87–95
36. Kumar H, Singh AP, Yadav SP (2019) NSGA-II based analysis of fuzzy multi-objective reliability-redundancy allocation problem using various membership functions. *INAE Lett*. <https://doi.org/10.1007/s41403-019-00076-8>
37. Kumar H, Yadav SP (2019) Fuzzy rule-based reliability analysis using NSGA-II. *Int J Syst Assur Eng Manag*. <https://doi.org/10.1007/s13198-019-00826-5>
38. Kumar H, Yadav SP (2019) NSGA-II based decision-making in fuzzy multi-objective optimization of system reliability. In: Deep K, Jain M, Salhi S (eds) Decision science in action. Asset analytics (performance and safety management). Springer, Singapore, pp 105–117
39. Kumar H, Yadav SP (2017) NSGA-II based fuzzy multi-objective reliability analysis. *Int J Syst Assur Eng Manag* 8:817–825. <https://doi.org/10.1007/s13198-017-0672-y>
40. Muhuri PK, Ashraf Z, Lohani QMD (2018) Multi-objective reliability-redundancy allocation problem with interval type-2 fuzzy uncertainty. *IEEE Trans Fuzzy Syst* 26(3):1339–1355. <https://doi.org/10.1109/TFUZZ.2017.272242>
41. Pandiarajan K, Babulal CK (2014) Fuzzy ranking based nondominated sorting genetic algorithm-II for network overload alleviation. *Arch Electr Eng* 63:367–384. <https://doi.org/10.2478/aee-2014-0027>
42. Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. *Evol Comput* 8(2):173–195
43. Laumanns M, Thiele L, Deb K, Zitzler E (2002) Combining convergence and diversity in evolutionary multiobjective optimization. *Evol Comput* 10(3):263–282
44. Chern MS (1992) On the computational complexity of reliability redundancy allocation in a series system. *Oper Res Lett* 11(5):309–315
45. Tillman FA, Hwang CL, Kou W (1977) Determining component reliability and redundancy for optimum system reliability. *IEEE Trans Reliab R-26:162–165*
46. Sakawa M (1978) Multiobjective optimization by the surrogate worth trade-off method. *IEEE Trans Reliab R-27:311–314*
47. Konak A, Coit DW, Smith AE (2006) Multi-objective optimization using genetic algorithms: a tutorial. *Reliab Eng Syst Saf* 91:992–1007. <https://doi.org/10.1016/j.res.2005.11.018>
48. Tavakkoli MR, Safari J, Sassani F (2008) Reliability optimization of series-parallel systems with a choice of redundancy

- strategies using a genetic algorithm. *Reliab Eng Syst Saf* 93(4):550–556
49. ZArinchAng A, ZArinchAng J (2012) An application of genetic algorithm toward solving the reliability problem of multiobjective series-parallel systems. *Maint Reliab* 14(3):243–248
 50. Chambari A, Sadeghi J, Bakhtiari R (2016) A note on a reliability redundancy allocation problem using a tuned parameter genetic algorithm. *OPSEARCH* 53:426–442. <https://doi.org/10.1007/s12597-015-0230-9>
 51. Coleman T, Branch MA, Grace A (1999) Optimization toolbox for use with MATLAB: User's guide. The Math Works, Inc. <https://instruct.uwo.ca/engin-sc/391b/downloads/optimtb.pdf>
 52. Goldberg DE, Holland JH (1988) Genetic algorithms and machine learning. *Mach Learn* 3:95–99
 53. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: International conference on neural networks proceedings of the fourth IEEE. Perth, Australia
 54. Carlos CAC, Lechuga MS (2002) MOPSO: a proposal for multiple objective particle swarm optimization. In: Proceedings of the evolutionary computation; 0-7803-7282-4/02
 55. Michael E, Beume N, Boris N (2005) An EMO algorithm using the hypervolume measure as selection criterion. In: LNCS, vol 3410, pp 62–76

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.