

# Practical solutions for multi-objective optimization: An application to system reliability design problems

Heidi A. Taboada<sup>a</sup>, Fatema Baheranwala<sup>a</sup>, David W. Coit<sup>a,\*</sup>,  
Naruemon Wattanapongsakorn<sup>b</sup>

<sup>a</sup>Department of Industrial and Systems Engineering, Rutgers University, Piscataway, NJ 08854, USA

<sup>b</sup>King Mongkut's University of Technology Thonburi, Bangkok 10140, Thailand

Available online 5 June 2006

## Abstract

For multiple-objective optimization problems, a common solution methodology is to determine a Pareto optimal set. Unfortunately, these sets are often large and can become difficult to comprehend and consider. Two methods are presented as practical approaches to reduce the size of the Pareto optimal set for multiple-objective system reliability design problems. The first method is a pseudo-ranking scheme that helps the decision maker select solutions that reflect his/her objective function priorities. In the second approach, we used data mining clustering techniques to group the data by using the  $k$ -means algorithm to find clusters of similar solutions. This provides the decision maker with just  $k$  general solutions to choose from. With this second method, from the clustered Pareto optimal set, we attempted to find solutions which are likely to be more relevant to the decision maker. These are solutions where a small improvement in one objective would lead to a large deterioration in at least one other objective. To demonstrate how these methods work, the well-known redundancy allocation problem was solved as a multiple objective problem by using the NSGA genetic algorithm to initially find the Pareto optimal solutions, and then, the two proposed methods are applied to prune the Pareto set.

© 2006 Elsevier Ltd. All rights reserved.

**Keywords:** System reliability; Multi-objective optimization; Pareto optimal set; Clustering analysis

## 1. Introduction

Determination of an optimal system design configuration involves multiple considerations and trade-offs to maximize reliability, minimize cost, maximize system performance, etc. It is not uncommon for engineering design problems to have multiple competing objectives and multiple prospective solutions. However, one feasible solution must ultimately be selected to be implemented as the system design. This poses a problem because, when considering multiple objectives, one generally either determines a single solution or identifies a Pareto optimal set. Determination of a single solution can be done using methods such as utility theory, weighted sum

method, etc., but the difficulty lies in selecting the correct utility functions or quantifying the objective function trade-offs. In such methods, the value of the weights chosen or the utilities used dictates the final solution. This can be challenging for a decision maker who is new to such methods.

An alternate solution to this problem is to determine an entire Pareto optimal solution set. A Pareto optimal set is a set of solutions that are all nondominated with respect to each other. While moving from one Pareto point to another, there is always a certain amount of sacrifice in one objective to achieve a certain amount of gain in the other. Pareto optimal solution sets are often preferred to single solutions because they can be practical when considering real-life problems, since the final solution of the decision maker is almost always a trade-off between crucial parameters.

As the number of competing objectives increases, the problem of finding the best compromise solution becomes

\*Corresponding author. Department of Industrial and Systems Engineering, Rutgers University, 96 Frelinghuysen Road, Piscataway, NJ 08854, USA. Fax: +1 732 445 5467.

E-mail addresses: [heidita@eden.rutgers.edu](mailto:heidita@eden.rutgers.edu) (H.A. Taboada), [sb\\_fatema@hotmail.com](mailto:sb_fatema@hotmail.com) (F. Baheranwala), [coit@rutgers.edu](mailto:coit@rutgers.edu) (D.W. Coit), [naruemon@cpe.kmutt.ac.th](mailto:naruemon@cpe.kmutt.ac.th) (N. Wattanapongsakorn).

increasingly complex. Hence, it can become overwhelming to analyze the entire Pareto optimal solution set to pick one solution for implementation. In such cases, pruning becomes attractive to reduce the size of the solution set, and to assist the decision maker select a final solution [4,26].

In this paper, we describe and demonstrate two methods to intelligently reduce the size of the Pareto set. To show how these methods work, the redundancy allocation problem (RAP) was solved. In this problem, the reliability of the system can be improved by appropriately adding redundant components to improve the system reliability. In our formulation, the main goal is to find the best redundancy allocation for maximizing the overall system reliability while minimizing system cost and weight. Thus, RAP was formulated as a multiple objective problem. It was initially solved using the NSGA genetic algorithm (GA) to identify Pareto optimal solutions, and then, the two proposed pruning methods are applied to reduce the size of the Pareto set.

The first method uses an approach similar to the weighted sum method applied to the Pareto optimal set, except that explicit numerical weights are not required. Instead, the objective functions are ranked non-numerically. For numerous iterations, scaled objective functions for each solution are combined into a single composite objective function using randomly generated weight sets. These weight sets adhere to the decision maker's objective function rankings. The solution that gives the best result for a particular weight set is recorded. This process is repeated for a pre-selected number of iterations, and the final result obtained is a set of solutions that most often provided the best combined objective function.

In the second approach we made use of clustering techniques often associated with data mining. In this case, we grouped the data by using the  $k$ -means algorithm to find clusters of similar solutions. Since the members contained in each cluster are similar to one another, the decision maker has now just  $k$  general solutions to choose from. In the later approach, the decision maker does not have to specify any objective function preferences.

## 2. Multiple objective problems

Multiple objective problems are solved using a variety of different approaches. Often the multiple objectives are combined into a single objective so that traditional optimization and mathematical programming methods can be used. Alternatively, a Pareto optimal set is found. This is usually done by using an evolutionary algorithm such as GA.

In general, for a problem with  $n$  objective functions, the multi-objective formulation can be as follows [20]:

$$\text{minimize/maximize } f_i(\mathbf{x}) \text{ for } i = 1, 2, \dots, n$$

subject to

$$\begin{aligned} g_j(\mathbf{x}) &\leq 0 & j = 1, 2, \dots, J, \\ h_k(\mathbf{x}) &= 0 & k = 1, 2, \dots, K. \end{aligned}$$

There are  $n$  objective functions and  $p$  variables so  $\mathbf{f}(\mathbf{x})$  is an  $n$  dimensional vector, and  $\mathbf{x}$  is a  $p$  dimensional vector corresponding to  $p$  decisions or variables. Solutions to a multi-objective optimization problem are often mathematically expressed in terms of nondominated or superior points. It is useful to express nondominance in terms of vector comparison; let  $\mathbf{x}$  and  $\mathbf{y}$  be two vectors of  $p$  decision variable values,  $\mathbf{x} = (x_1, x_2, \dots, x_p)$  and  $\mathbf{y} = (y_1, y_2, \dots, y_p)$ .

We say in a *maximization* problem that  $\mathbf{x}$  dominates  $\mathbf{y}$  iff

$$\begin{aligned} f_i(\mathbf{x}) &\geq f_i(\mathbf{y}) \quad \forall i \text{ and } f_i(\mathbf{x}) > f_i(\mathbf{y}) \\ &\text{for at least one } i \quad i \in \{1, 2, \dots, n\}; \end{aligned}$$

similarly, for a *minimization* problem,  $\mathbf{x}$  dominates  $\mathbf{y}$  iff

$$\begin{aligned} f_i(\mathbf{x}) &\leq f_i(\mathbf{y}) \quad \forall i \text{ and } f_i(\mathbf{x}) < f_i(\mathbf{y}) \\ &\text{for at least one } i \quad i \in \{1, 2, \dots, n\}. \end{aligned}$$

$X$  is defined as the set of feasible solutions or feasible decision alternatives. Thus, in a *maximization* problem  $\mathbf{x}$  is nondominated in  $X$ , if there exists no other  $\bar{\mathbf{x}}$  in  $X$  such that  $\mathbf{f}(\bar{\mathbf{x}}) \geq \mathbf{f}(\mathbf{x})$  and  $\bar{\mathbf{x}} \neq \mathbf{x}$ . We define  $N$  as the set of all nondominated solutions in  $X$ . Then, the optimal solutions to a multi-objective optimization problem are in the set of nondominated solutions  $N$ , and they are usually known as a Pareto optimal set [27].

Generally the two most common approaches to solve multiple objective problems are: (1) combine them into a single objective function and obtain a single solution such as in the cases of the weighted sum method or utility functions, or (2) obtain a set of non-dominated Pareto optimal solutions.

For multiple-objective problems, it can be problematic to combine the objectives into a single objective (e.g., weighted sum method, utility functions) to obtain a single solution. A slight perturbation in the parameters used to combine the objectives could result in very different optimal solutions. This can be a problem because decision makers are often unsure about the exact objective function weights or utility functions to use.

The Pareto set includes all rational choices, among which the decision maker has to select the final solution by trading the objectives against each other. The search is then not for one optimal solution but for a set of solutions that are optimal in a broader sense, i.e. they are Pareto-optimal. There are a number of techniques to search the solution space for a set of Pareto optimal solutions. Possibly, the main disadvantage of these methods is that the decision maker has too many solutions to choose from. Thus, there is a need to bridge the gap between single solutions and Pareto optimal sets as it is shown in Fig. 1.

The objective of this research is to achieve this balance, by introducing two practical methods that reduce the

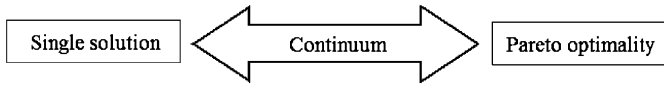


Fig. 1. Achieving a balance between single solutions and Pareto optimal solutions.

Pareto optimal set to achieve a smaller set called the “pruned Pareto set.”

**3. Multiple objective GAs**

GAs were developed by Holland [13]. GAs are a particular class of evolutionary algorithms that uses techniques inspired by some mechanisms of natural selection. They are essentially search techniques used to find approximate solutions to difficult combinatorial optimization problems.

The GA starts with a population of random individuals (called chromosomes) that are revised over successive generations. The crossover and mutation operators are used to introduce new prospective design solutions each generation. During each successive generation, each individual is evaluated and a value of fitness is returned by a fitness function. Individuals with high-fitness values rank at the top while individuals with low-fitness function values are likely to disappear from the population. The algorithm continues for a pre-determined maximum number of generations or until no additional improvement is observed.

Several versions of multi-objective GAs, most often referred as multiple-objective evolutionary algorithms (MOEAs), have been developed, such as

- vector evaluated genetic algorithm (VEGA) by Shaffer [24];
- multi-objective genetic algorithm (MOGA) by Fonseca and Fleming [9];
- niched-Pareto genetic algorithm (NPGA) by Horn, Nafpliotis and Goldberg [14];
- nondominated sorting genetic algorithm (NSGA) developed by Srinivas and Deb [25];
- strength Pareto evolutionary algorithm (SPEA) by Zitzler and Thiele [28];
- NSGA-II by Deb, Agrawal, Pratab and Meyerivan [6,7].

For this research, the RAP was solved using the NSGA. NSGA uses a non-dominated sorting procedure proposed by Goldberg [11]. This algorithm uses a ranking method that emphasizes the good solution points and tries to maintain a population of such points. NSGA maintains diversity in its population by a sharing method. This eliminates focusing on certain regions of the solution space, and explores different regions in the Pareto front. This algorithm is highly efficient in obtaining good Pareto optimal fronts for any number of objectives and can handle any number of constraints. These features made NSGA attractive for this research effort.

**4. Reducing the size of the Pareto optimal set**

The next step after obtaining the set of Pareto optimal solutions is to prune that set if it is too large. One of the main goals of reducing the size of the Pareto optimal set is to make it easier for the decision maker to comparatively analyze a set of promising solutions. To achieve this, two methods are presented as practical approaches to reduce the size of the Pareto optimal set: (1) pruning by using non-numerical ranking preferences and (2) pruning by using data clustering.

Fig. 2 shows how to select the preferred Pareto optimal set pruning procedure. The decision maker should select the first method if he/she knows in advance the objective function preferences as shown in Fig. 2. Essentially, this method should be chosen by more experienced decision makers that are familiar with the importance of the objective functions. On the other hand, if the decision maker does not know a priori the objective function preferences, he/she may prefer to use the second method to cluster solutions in regions, and then, just analyze *k* solutions or focus on the most interesting regions to concentrate his/her efforts.

Many multi-objective optimization problems may have a large set of Pareto solutions. If the non-numerical ranking preferences method is applied to prune the size of this Pareto set, and if after its application, there are still a large number of solutions, data clustering can be applied to further reduce the set.

*4.1. Pruning by using non-numerical ranking preferences*

The first method is based on a non-numerical ranking of the objective functions based on their relative importance. This pruning method helps the decision maker select solutions that reflect his/her preferences. In a broader sense, this method is a pseudo-ranking scheme that accommodates preferences but it is different from assigning

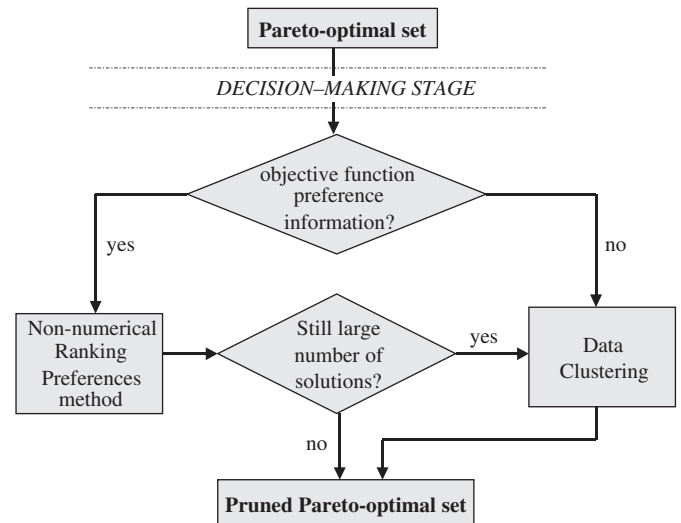


Fig. 2. Methods to prune the Pareto optimal set.

preselected weights or utility functions. This method allows objectives to have the same rank. One example of ranking objective functions is:

objective  $f_1(\mathbf{x})$  is more important than objective  $f_3(\mathbf{x})$ ;  
 objective  $f_3(\mathbf{x})$  is more important than objective  $f_2(\mathbf{x})$ ;  
 ranked objectives =  $\{f_1(\mathbf{x}), f_3(\mathbf{x}), f_2(\mathbf{x})\}$ ;  $f_1(\mathbf{x}) > f_3(\mathbf{x}) > f_2(\mathbf{x})$ .

Based on the objective function rankings, an weight function  $f_w(\mathbf{w})$  is developed, indicating the likelihood of different weight combinations. The weight function  $f_w(\mathbf{w})$  is derived from a region where all the weights in this set sum to one as shown in Fig. 3 for a case with three objective functions.

To illustrate, consider a case where the objective function preference is  $f_1 > f_2 > f_3$ , and the objectives have all been similarly scaled. The exact value of the weights is not known but we know that  $w_1 > w_2 > w_3$ . The resulting region where the weights are sampled, and then combined with the objective functions, is shown in Fig. 4.

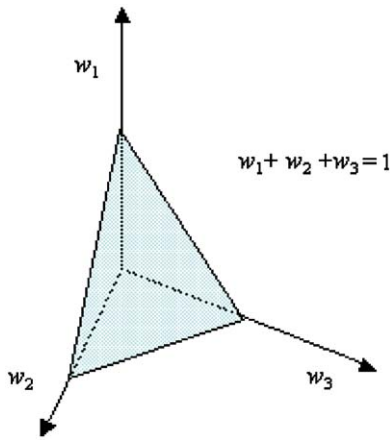


Fig. 3. Plane containing set of possible weights.

The weights are uniformly sampled from the region of interest with a weight function that is defined as follows:

$$f_w(\mathbf{w}) = \begin{cases} c & \forall w_1 > w_2 > w_3, \\ 0 & \text{elsewhere.} \end{cases}$$

The marginal distribution of  $w_1$  (for  $w_1 > w_2 > w_3$ ) is derived, in which  $f_w(\mathbf{w})$  is integrated over  $w_2$ , where  $c$  is a constant. For three objective functions

$$f_{w_1}(w_1) = \int_{w_2} f_w(w_1, w_2) dw_2 = \begin{cases} 0 & 0 \leq w_1 < \frac{1}{3}, \\ 12(\frac{3}{2}w_1 - \frac{1}{2}) & \frac{1}{3} \leq w_1 < \frac{1}{2}, \\ 12(\frac{1}{2} - \frac{1}{2}w_1) & \frac{1}{2} \leq w_1 < 1. \end{cases}$$

The value of  $w_2$  is obtained based on

$$f_{w_2|w_1}(w_2|w_1) = \frac{f_w(w_1, w_2)}{f_{w_1}(w_1)}.$$

Finally, by knowing the values of  $w_1$  and  $w_2$ , the value of  $w_3$  is just  $1 - w_1 - w_2$ .

Then, random but ranked weights are generated using Monte Carlo simulation methods. These weights adhere to the ranking pattern used for the objective functions. A substantially large set of weights is generated, with each set containing one weight for each objective. As an example, Fig. 5 shows the distribution of 5000 ranked randomly generated weights; the  $y$ -axis represents the frequency or number of times a specific weight was generated and the  $x$ -axis represents the value of each individual weight. As can be seen, the possible values for the weights in the case  $f_1 > f_2 > f_3$  are:  $\frac{1}{3} < w_1 < 1$ ,  $0 < w_2 < \frac{1}{2}$  and  $0 < w_3 < \frac{1}{3}$ .

These weight sets are used to repeatedly combine the scaled objectives into a single objective function. The best solution for each combined objective is recorded from the set of the Pareto optimal solutions available. This is repeated with the next set of weights, and the best solution

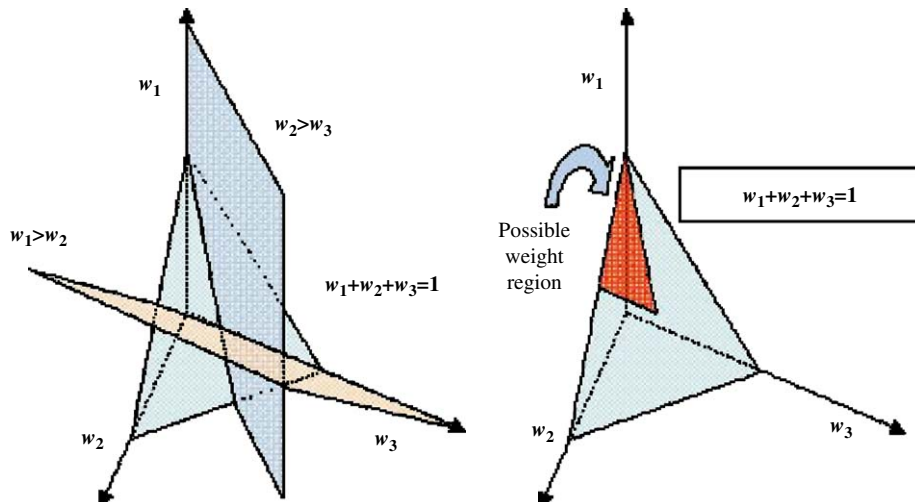


Fig. 4. Weight region for the  $f_1 > f_2 > f_3$  objective function preference.

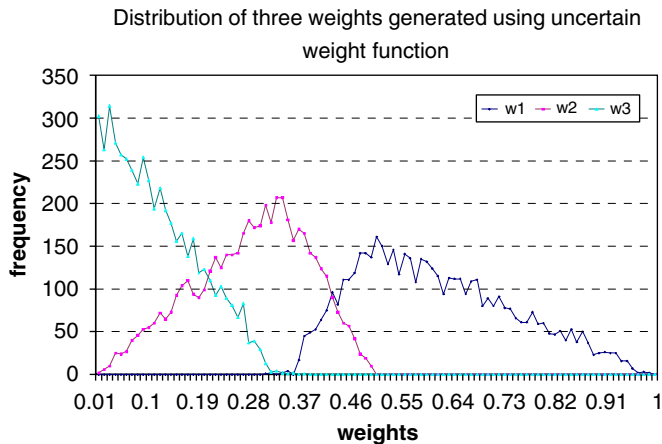


Fig. 5. Distribution of random weights used for a three objective problem.

for that combination is identified. This process is repeated many times (e.g., several thousand), and the end result is a “pruned” Pareto optimal set. This method has been observed to achieve a 90% reduction of the entire Pareto optimal set, as reported by Coit and Baheranwala [4].

This first approach is an extension of earlier research considering multi-criteria decision making with a finite set of alternatives. Lahdelma et al [16] considered uncertainty in weight selection similar to the uncertain weight function proposed here. Rietveld and Ouwersloot [21] and Hinloopen et al. [12] also describe solution methods where solutions must be selected based on ordinal data. The uncertain weight function combined with Tabu search was demonstrated by Kultural-Konak et al. [15].

#### 4.2. Pruning by using data clustering

The next approach to reduce the size of the Pareto set for RAP is to use data clustering. In multi-dimensional situations, clustering can be very insightful. Cluster analysis is a multivariate analysis technique that seeks to organize information about variables so that relatively homogeneous groups, or “clusters” can be formed. The clusters formed should be highly internally homogeneous (members are similar to one another) and highly externally heterogeneous (members are not like members of other clusters). The use of clustering for RAP is described in more detail by Taboada and Coit [26].

In the literature, there exists a variety of clustering algorithms. One of the most well known clustering methods is  $k$ -means, as introduced by MacQueen [17], which is known for its efficiency in clustering data sets. The  $k$ -means algorithm groups data into  $k$  sets. The grouping is done by calculating the centroid for each group, and assigning each observation to the group with the closest centroid. The centroids are calculated by minimizing the sum of all the squared-Euclidean distances to its closest centroid.

The general  $k$ -means clustering algorithm [17] is composed of the following steps:

1. Place  $k$  points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the  $k$  centroids.
4. Repeat steps 2 and 3 until the cluster assignments do not change.

The objective function that the  $k$ -means algorithm optimizes is

$$KM(N, C) = \sum_i \min_{j \in \{1, \dots, k\}} \|\mathbf{f}(\mathbf{x}_i) - \mathbf{c}_j\|^2,$$

where  $\mathbf{x}_i$  is the  $i$ th data vector,  $\mathbf{f}(\mathbf{x}_i)$  is  $i$ th objective function vector ( $f_1(\mathbf{x}_i), f_2(\mathbf{x}_i), \dots, f_n(\mathbf{x}_i)$ ),  $\mathbf{c}_j$  is  $j$ th cluster centroid,  $N$  is set of Pareto optimal solutions,  $C$  is set of centroids.

The algorithm seeks to minimize the within-cluster variance (the squared distance between each center and its assigned data points). The  $k$ -means algorithm often converges to one of the many local minima, which is not necessarily the global solution. To overcome this situation, several runs were performed to find the smallest sum over all elements of the squared Euclidean distance.

In order to determine the optimal number of clusters,  $k$ , Rousseeuw [22] suggested a cluster validity technique, the silhouette plot, to evaluate the quality of a clustering allocation, independently of the clustering technique that is used. Silhouettes are based on the evaluation of the silhouette width  $s(i)$  and it is calculated as follows:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}},$$

where  $a(i)$  is the average distance from the  $i$ th point to all the other points in its cluster.  $b(i)$  is average distance from the  $i$ th point to all the points in the nearest neighbor cluster.

From this formula it follows that  $s(i)$  has a value between  $-1$  and  $1$ . When  $s(i)$  is close to  $1$ , one may infer that the  $i$ th data point (Pareto solution) has been assigned to an appropriate cluster. When  $s(i)$  is close to zero, it suggests that the point could also be assigned to the nearest neighboring cluster, i.e., such a sample lies equally far away from both clusters. If  $s(i)$  is close to  $-1$ , one may argue that such a data point has been “misclassified.” Then, different partitions with several values of  $k$ ’s are performed using the  $k$ -means algorithm, the value of silhouette width is calculated for each different partition and the clustering with the highest average silhouette width is preferred.

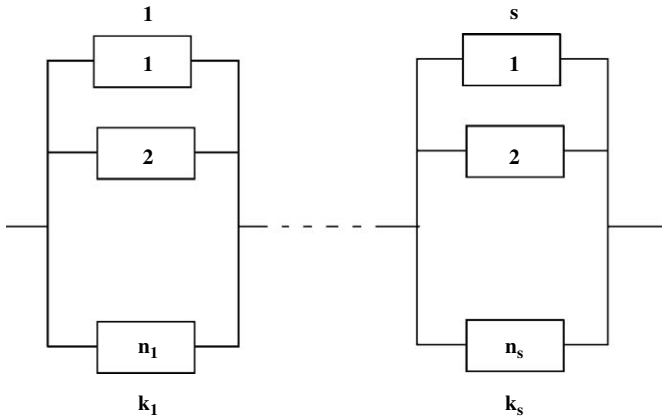


Fig. 6. General series parallel redundancy system.

### 5. Redundancy allocation problem

The RAP pertains to a system of  $s$  subsystems in series. For each subsystem, there are  $m_i$  functionally equivalent components, with different levels of cost, weight, reliability and other characteristics, which may be selected. A minimum of one component must be chosen for each subsystem, but it is often advantageous to add redundant components to improve reliability. A series-parallel system example is depicted in Fig. 6. The use of redundancy improves system reliability but adds to system cost, weight, etc. There are system-level constraints and the problem is to select the design configuration that maximizes some stated objective function.

Solving the RAP as a single-objective optimization problem involves either the maximization of the system design subject to limits on resource consumption or the minimization of the consumption of one resource subject to minimum requirements of system reliability and other resource constraints. Numerous researchers have addressed several methodologies and approaches to solve it. The redundancy allocation problem has been considered to be NP-hard [2].

In Fyffe et al. [10] the redundancy allocation problem was solved using dynamic programming by limiting the problem to only one type of component available for each subsystem. Nakagawa et al. [18] accommodated multiple constraints with dynamic programming using a surrogate constraints approach. Bulfin and Liu [1] used integer programming and they formulated the problem as a knapsack problem using surrogate constraints. Genetic algorithms have also been applied, e.g., Painton and Campbell [19], Coit and Smith [5].

In reality, most system design initiatives are multi-objective. When designing a reliable system, in addition to the maximization of system reliability, it is desirable to consider the minimization of other objectives such as cost, weight, etc. In such situations, the designer faces the problem of optimizing all objectives simultaneously.

For the multi-objective redundancy allocation problem the objective is to determine the optimal design configuration that will maximize system reliability, minimize the total cost and minimize the system weight, for a series-parallel system. The mathematical formulation of the problem is given below.

#### 5.1. Problem MORAP

$$\left\{ \begin{array}{l} \max \left[ R = \prod_{i=1}^s R_i(\mathbf{x}_i) \right], \min \left[ C = \sum_{i=1}^s \sum_{j=1}^{m_i} c_{ij} x_{ij} \right], \\ \min \left[ W = \sum_{i=1}^s \sum_{j=1}^{m_i} w_{ij} x_{ij} \right] \end{array} \right\}$$

s.t.

$$1 \leq \sum_{j=1}^{m_i} x_{ij} \leq n_{\max,i} \quad \forall i = 1, 2, \dots, s$$

$$x_{ij} \in \{0, 1, 2, \dots\},$$

where  $R$ ,  $C$ ,  $W$  is the reliability, cost and weight of the system respectively,  $s$  is number of subsystems,  $x_{ij}$  is quantity of  $j$ th component in subsystem  $i$ ,  $n_{\max,i}$  is user-defined maximum number of components in parallel used in subsystem  $i$ ,  $m_i$  is total number of available components for subsystem  $i$ ,  $R_i(\mathbf{x}_i)$  is reliability of subsystem  $i$ ,  $c_{ij}$ ,  $w_{ij}$ , is cost and weight for the  $j$ th available component for subsystem  $i$ .

Other researchers have considered multiple objective versions of system reliability optimization problems. Sakawa [23] considered a large series system with four objectives: maximization of system reliability, minimization of cost, weight and volume. Sakawa proposed a method to derive Pareto optimal solutions by optimizing the composite objective functions that were obtained as linear combinations of the four objective functions. The Lagrangian function for each composite problem was decomposed into parts and optimized by applying both, the dual decomposition method and the surrogate worth trade-off method.

Dhingra [8] used a combination of goal programming and the goal attainment method to generate Pareto optimal solutions in a four-stage series system to maximize system reliability and minimize cost, weight and volume. Kulturel-Konak et al. [15] solved the problem using Tabu search method with three objective functions: maximize reliability, minimize the cost of the system, and minimize the weight of the system. Coit et al. [3] proposed a multicriteria formulation to maximize the system reliability estimate and minimize the associated variance.

### 6. Example

An example problem was solved for a system consisting of three subsystems, with an option of five, four and five types of components in each subsystem, respectively. The maximum number of components is eight in each

Table 1  
Component choices for each subsystem

| Design alternative <i>j</i> | Subsystem <i>i</i>    |                       |                       |                       |                       |                       |                       |                       |                       |
|-----------------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
|                             | 1                     |                       |                       | 2                     |                       |                       | 3                     |                       |                       |
|                             | <i>r<sub>ij</sub></i> | <i>c<sub>ij</sub></i> | <i>w<sub>ij</sub></i> | <i>r<sub>ij</sub></i> | <i>c<sub>ij</sub></i> | <i>w<sub>ij</sub></i> | <i>r<sub>ij</sub></i> | <i>c<sub>ij</sub></i> | <i>w<sub>ij</sub></i> |
| 1                           | 0.95                  | 2                     | 5                     | 0.99                  | 4                     | 4                     | 0.90                  | 6                     | 5                     |
| 2                           | 0.93                  | 1                     | 4                     | 0.98                  | 3                     | 6                     | 0.85                  | 5                     | 4                     |
| 3                           | 0.91                  | 2                     | 2                     | 0.97                  | 1                     | 5                     | 0.82                  | 3                     | 3                     |
| 4                           | 0.90                  | 1                     | 3                     | 0.96                  | 2                     | 7                     | 0.79                  | 3                     | 5                     |
| 5                           | 0.95                  | 2                     | 8                     |                       |                       |                       | 0.99                  | 2                     | 4                     |

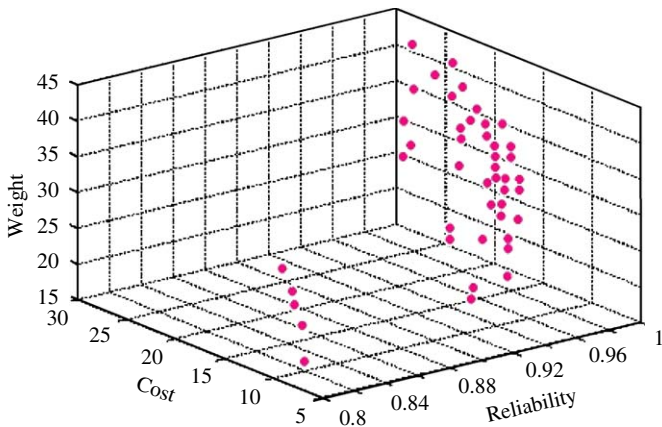


Fig. 7. Pareto optimal set.

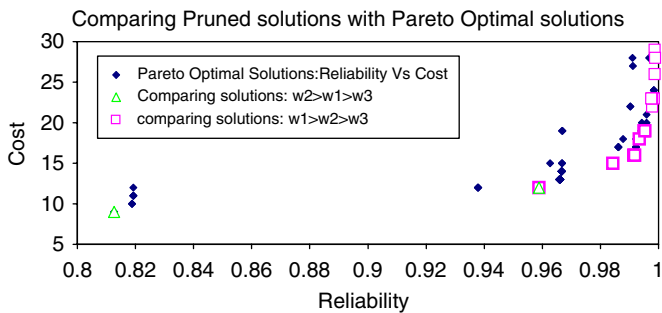


Fig. 8. Comparing pruned Pareto solution with the Pareto optimal solution set for reliability versus cost.

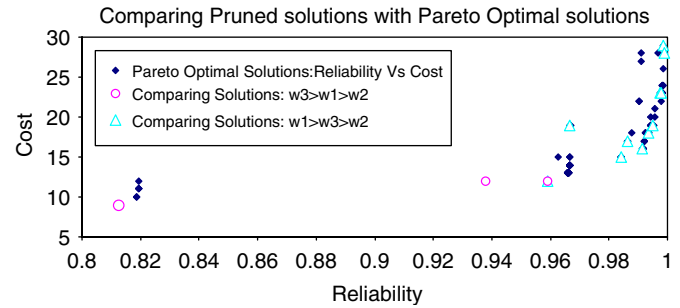


Fig. 9. Comparing pruned Pareto solution with the Pareto optimal solution set for reliability versus cost.

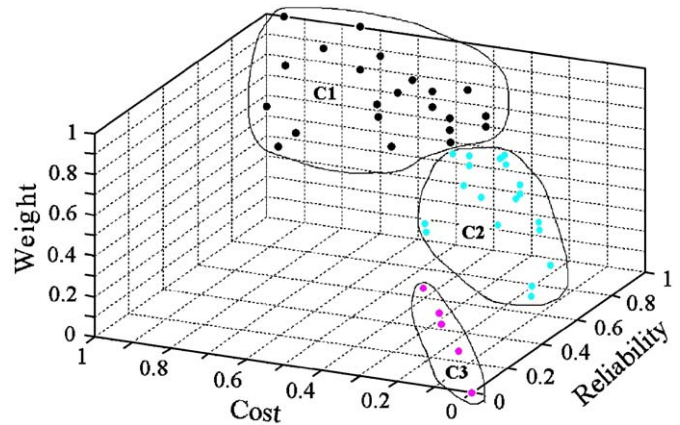


Fig. 10. Clustered Pareto optimal set.

subsystem. Table 1 defines the component choices for each subsystem.

For this case, NSGA was solved with a population size of 100. There were 46 solutions in the Pareto optimal set. This set is shown in Fig. 7. The Pareto set was then pruned using both methods previously described.

6.1. Pruned results by using non-numerical ranking preferences method

Pareto optimal solutions were obtained using NSGA and the pruned solutions identified by using the proposed

method. The objective function priorities used on these solutions were: ( $R > C > W$ ), ( $C > R > W$ ), ( $R > W > C$ ) and ( $W > R > C$ ). Fig. 8 shows the pruned solution set for  $w_1 > w_2 > w_3$  and  $w_2 > w_1 > w_3$ , compared to the original Pareto optimal set (obtained by using NSGA), and Fig. 9 shows the pruned solution set for  $w_1 > w_3 > w_2$  and  $w_3 > w_1 > w_2$ , compared to the original Pareto optimal set. Considering two objective functions at a time, the charts map reliability versus cost. The pruned solution sets for all four possibilities are shown in the figures. Pruning the solutions caused almost a 90% reduction in the size of the Pareto optimal set.

Table 2  
Summary of results obtained with the clustering analysis

|           | # Of solutions | Representative solutions | Reliability | Cost | Weight |
|-----------|----------------|--------------------------|-------------|------|--------|
| Cluster 1 | 22             | #39                      | 0.9978541   | 22   | 34     |
| Cluster 2 | 19             | #91                      | 0.984265    | 15   | 25     |
| Cluster 3 | 5              | #87                      | 0.819216    | 11   | 24     |

## 6.2. Pruned results by using data clustering

The  $k$ -means algorithm was then used to cluster the original 46 solutions in the Pareto optimal set. The three objective functions were scaled from 0 to 1 using the following equation:

$$\frac{f_i(\mathbf{x}) - f_i^{\min}(\mathbf{x})}{f_i^{\max}(\mathbf{x}) - f_i^{\min}(\mathbf{x})} \quad \forall i = 1, 2, \dots, n,$$

where  $f_i^{\min}(\mathbf{x})$  is the minimum value for  $f_i(\mathbf{x})$  in the Pareto optimal set,  $f_i^{\max}(\mathbf{x})$  is maximum value for  $f_i(\mathbf{x})$  in the Pareto optimal set.

We made use of silhouette plots using the cluster indices output from the  $k$ -means algorithm to determine the optimal number of clusters for this particular data set. We found three to be the optimum number of clusters for this problem. The three clusters are shown in Fig. 10 from standardized data. Cluster 1 contained 22 solutions; there were 19 solutions in cluster 2 and five in cluster 3.

The clusters formed are highly internally homogeneous. That is, members within a cluster are similar to one another. One way for the decision maker to pick one solution, among the solutions contained in a cluster, is to identify what solution is the closest to its centroid. Table 2 shows the summary of results obtained by the cluster analysis. The representative solutions are those that are closest to their corresponding centroid; each solution is shown with its corresponding reliability, cost and weight. With the information from Table 2, the decision maker now has a small set of solutions, and it is thus, easier to make his/her choice regarding the importance of the different objectives.

Another way to take advantage of this method is that, once having the optimal number of clusters selected (three in our case), then we looked for the cluster that contained the most interesting solutions of the Pareto optimal set. These are the solutions where a small improvement in one objective would lead to a large deterioration in at least one other objective. These solutions are sometimes called “knees.” In this case, as we can see from Fig. 10, solutions in cluster 2 are likely to be more relevant to the decision maker. Thus, solution #91 can be chosen as a good representative solution of this mentioned knee region.

## 7. Conclusions

The proposed methods are practical approaches to the solution of multi-objective system reliability design pro-

blems that help achieve a balance between Pareto optimal solution sets and single solutions. These pruned Pareto sets give the decision maker a workable sized set of solutions.

With the first method, pruning by using the non-numerical ranking preferences, the decision maker obtains a pruned set of solutions that match his/her preferences. In the second approach, the solutions in the Pareto optimal set are filtered and clustered so that the Pareto optimal front is reduced to a set of  $k$  clusters. Each cluster consists of solutions with similar properties, and therefore the decision maker only has to investigate one solution per cluster.

Moreover, with this method, once the optimal number of clusters is identified, the decision maker can focus his/her search to the cluster(s) that contain(s) the knee region, which are the solutions to be likely to be more interesting to the decision maker.

## References

- [1] Bulfin RL, Liu CY. Optimal allocation of redundant components for large systems. *IEEE Trans Reliab* 1985;34:241–7.
- [2] Chern M. On the computational complexity of reliability redundancy allocation in a series system. *Oper Res Lett* 1992;11:309–15.
- [3] Coit D, Jin T, Wattanapongsakorn N. System optimization considering component reliability estimation uncertainty: a multi-criteria approach. *IEEE Trans Reliab* 2004;53(3):369–80.
- [4] Coit DW, Baهرانwala F. Solution of stochastic multi-objective system reliability design problems using genetic algorithms. In: *Proceedings of the European safety and reliability conference (ESREL)*, June 2005, Gdansk, Poland. p. 391–8.
- [5] Coit D, Smith A. Solving the redundancy allocation problem using a combined neural network/GA approach. *Comput Oper Res* 1996; 23:515–26.
- [6] Deb K, Agarwal S, Pratap A, Meyarivan T. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. KanGAL report number 200001. Kanpur, India: Indian Institute of Technology; 2000a.
- [7] Deb K, Agarwal S, Pratap A, Meyarivan T. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: *Proceedings of the parallel problem solving from nature VI conference*, Paris, France. 2000b. p. 849–58.
- [8] Dhingra AK. Optimal apportionment of reliability & redundancy in series systems under multiple objectives. *IEEE Trans Reliab* 1992; 41:576–82.
- [9] Fonseca CM, Fleming PJ. Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In: *Proceedings of the fifth international conference on genetic algorithms*, San Mateo California. 1993. p. 416–23.
- [10] Fyffe DE, Hines WW, Lee NK. System reliability allocation and a computational algorithm. *IEEE Trans Reliab* 1968;17:64–9.
- [11] Goldberg D. *Genetic algorithms in search, optimization and machine learning*. Reading, MA: Addison Wesley; 1989.

- [12] Hinloopen H, Nijkamp P, Rietveld P. Integration of ordinal and cardinal information in multi-criteria ranking with imperfect compensation. *Eur J Oper Res* 2004;158(2):317–38.
- [13] Holland J. *Adaptation in natural and artificial systems*. U. Michigan Press; 1975.
- [14] Horn J, Nafpliotis N, Goldberg DE. A niched pareto genetic algorithm for multiobjective optimization. In: *Proceedings of the first IEEE conference on evolutionary computation*. 1, Piscataway, NJ:IEEE World Congress on Computational Intelligence, IEEE. 1994. p. 82–7.
- [15] Kulturel-Konak S, Coit DW, Baheranwala F. Pruned pareto-optimal sets for the system redundancy allocation problem based on multiple prioritized objectives. *Rutgers University ISE working paper* 05-009, 2005.
- [16] Lahdelma R, Hokkanen J, Salminen P. Stochastic multiobjective acceptability analysis. *Eur J Oper Res* 1998;106(1):137–43.
- [17] MacQueen J. Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1. 1967. p. 281–97.
- [18] Nakagawa Y, Miyazaki S. Surrogate constraints algorithm for reliability optimization problems with two constraints. *IEEE Trans Reliab* 1981;30:175–80.
- [19] Painton L, Campbell J. Genetic algorithms in optimization of system reliability. *IEEE Trans Reliab* 1995;44(2):172–8.
- [20] Rao SS. *Optimization theory and application*. New Delhi: Wiley Eastern Limited; 1991.
- [21] Rietveld P, Ouwersloot H. Ordinal data in multicriteria decision making: a stochastic dominance approach to siting nuclear power plants. *Eur J Oper Res* 1992;56:249–62.
- [22] Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math* 1987; 20:53–65.
- [23] Sakawa M. Optimal reliability-design of a series-parallel system by a large-scale multiobjective optimization method. *IEEE Trans Reliab* 1981;30:173–4.
- [24] Schaffer JD. Multiple objective optimization with vector evaluated genetic algorithms. *Genetic algorithms and their applications: proceedings of the first international conference on genetic algorithms*. Hillsdale, NJ. 1985. p. 93–100.
- [25] Srinivas NK, Deb A. Multiobjective optimization using nondominated sorting in genetic algorithms. *J Evol Comput* 1994;2(3): 221–48.
- [26] Taboada H, Coit D. Data clustering of solutions for multiple objective system reliability problems. *Quality Technology and Quantitative Management* 2006, in print.
- [27] Zeleny, M. *Multiple criteria decision making*. McGraw-Hill series in quantitative methods for management. 1982.
- [28] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Trans Evol Comput* 1999;3(4):257–71.