Utilisation of pruned Pareto-optimal solutions in the multi objective optimisation: an application to system redundancy allocation problems

Asghar Moeini
School of Computer Science, Engineering and Mathematics, Flinders University, Bedford Park 34, Adelaide, Australia
E-mail: moei0001@flinders.edu.au

Mehdi Foumani
School of Applied Sciences and Engineering, Monash University, Gippsland Campus, Churchill, VIC 3842, Australia
E-mail: mehdi.foumani@monash.edu

Kouroush Jenab*
Society of Reliability Engineering-Ottawa, 812-761 Bay Street, Toronto, Ontario, Canada
E-mail: jenab@ieee.org
*Corresponding author

Abstract: Multi-objective optimisation problems normally have not one but a set of solutions, which are called Pareto-optimal solutions or non-dominated solutions. Once a Pareto-optimal set has been obtained, the decision-maker faces the challenge of analysing a potentially large set of solutions. Selecting one solution over others can be quite a challenging task because the Pareto set can contain an unmanageable number of solutions. This process is called post-Pareto optimality analysis. To deal with this difficulty, this study proposes the approach that promisingly prunes the Pareto optimal set. In this study, the newly developed approach uses Monte-Carlo simulation taking into account the decision maker’s prioritisation to prune the Pareto optimal set. Then, the central weight vector, the optimal frequently appearance index and upper and lower bands of weights are enclosed to each solution to facilitate selecting a final solution. The well-known redundancy allocation problem is used to show the performance of the proposed method.

Keywords: multi-objective optimisation; post-Pareto optimality analysis; reliability optimisation.


Copyright © 2013 Inderscience Enterprises Ltd.
1 Introduction

Many complex and real-world decision-making problems such as portfolio selection, sourcing selection and supplier evaluation are concerned with multiple conflicting objectives (Xidonas and Psarras, 2008; Parthiban et al., 2009; Jajimoggala et al., 2011). There are two general approaches to solve multi-objective problems. The first approach applies a relative importance to each objective function in order to consolidate them into an overall single objective function:

1 utility theory (Fishburn, 1982; Barbera et al., 1998; Chandra et al., 2012)
2 value function (Goebel, 2002)
3 weighted sum method (Geoffrion, 1968).

Then, mathematical techniques are used to solve the single objective problem. However, this approach is so sensitive to the changes in trade-off parameters, which means, that even small changes in the parameters could thoroughly change the final solution. This is
A. Moeini et al.

compounded by the fact that in most decision-making problems there is insufficient information to assign a specific weight to each objective function. Therefore, authors will discuss how to mitigate this difficulty in this study.

The second general approach searches the multi-objective problems and extracts an entire Pareto optimal solution set. It is noteworthy that a Pareto optimal set is a set of solutions, which are all non-dominated with respect to each other. In order to extract the Pareto-optimal set, Haimes et al. (1971) proposed an approach which is based on keeping one of the objectives and restricting the others to within a set of user-specified values. The most common method is the weighting method mentioned above. Chankong and Haimes (1983) and Li et al. (1999) proposed the weighted Minimax method as an extended version of the weighted method (Chankong and Haimes, 1983).

The Pareto set can contain a large number of solutions, especially when the number of objectives is large. This approach renders considering all of the Pareto-optimal solutions inefficient. To deal with this inefficiency, it is recommended that the size of the solution set be decreased (Li et al., 2009). Our proposed approach aims at reducing the size of the solution set as an extension of latest research in multi-objective problems (Taboada et al., 2007; Kulturel-Konak et al., 2008). Authors also use Monte-Carlo of simulation to intelligently prune the Pareto set; however, there are two important extensions in this article. First, decision-maker’s preferences are not limited to ranks of importance of objective functions. Second, we investigated characters of the pruned set of solutions. That is, an acceptability index and central weight vector are computed for each solution of the pruned set. These details can assist the decision-maker to perform a comparative analysis among the solutions of pruned sets.

The reminder of the paper is organised as follows. The next section is devoted to defining the general multi-objective problem and briefly describing usual solving procedures. In Section 3, the proposed approach concerning the pruned Pareto-optimal set is described. In Section 4, a numerical example of redundancy allocation problem (RAP) is solved as a multi-objective problem, and consequently the proposed approach is used in order to prune the Pareto set. Finally, the results and conclusion are discussed in Section 5.

2 Multi-objective optimisation

The general multi-objective problem with $n$ objective functions is formulated as follows (Rao, 1991):

$$\text{maximise } z_i = f_i(X) \text{ for } i = 1, 2, \ldots, n$$

$$\text{s.t. } X \in D$$

where

- solution $X = (x_1, \ldots, x_j)$ is a vector of decision variables
- $D$ is the set of feasible solutions.

The image of a solution $X$ in the objective space is a point $z^X = (z_1^X, \ldots, z_n^X)$ such that $z_i^X = f_i(X)$, $i = 1, 2, \ldots, n$. 
Utilisation of pruned Pareto-optimal solutions

- Solution $X'$ dominates $X$ if $\forall i \; z_{i}^{X'} \geq z_{i}^{X}$ and $z_{i}^{X'} > z_{i}^{X}$ for at least one $i$.

- A solution $X \in D$ is a Pareto-optimal if there is no $X' \in D$ that dominates $X$. The set of all Pareto-optimal solutions is called the Pareto-optimal set (Zeleny, 1982).

Later, literature pertaining to the multi-objective optimisation had focused on multi-objective linear programming (Steuer, 1986; White, 1990). However, at the same time, studies on complex cases such as combinatorial optimisation, and non-linear programming considered evolutionary algorithms. The aim of these methods was to find an approximate set of Pareto-optimal solutions.

Masin and Bukchin (2008) proposed another method called the diversity maximisation approach (DMA) in order to find Pareto-optimal solutions in mixed-integer problems and combinatorial optimisation problems. With this method, a Pareto-optimal set of solutions can be obtained by using any multi-objective evolutionary algorithm (MOEA) available, such as MOGA (Fonseca and Fleming, 1993), MOTS (Hertz et al., 1994), NSGA (Srinivas and Deb, 1994), MOSA (Ulungu et al., 1998), SPEA (Zitzler and Thiele, 1999), non-dominated sorting genetic algorithm II (NSGA-II) (Deb et al., 2000).

It is not our intention here to compare these MOEAs to each other. In this study, the fast elitist NSGA-II is applied to generate an approximate Pareto-optimal set. This procedure is efficient in finding good Pareto sets for any number of constraints and objectives.

3 Determination and comprehension of pruned Pareto-optimal set

In most decision making problems the decision makers prefer methods which do not need to explain their preferences distinctively, but rather display the potential actions and their consequences in a suitable form, so that the final decision can be made by the decision maker (Lahdelma et al., 1998). The proposed approach only needs the decision maker’s preference about the objective function without having to provide specific utility functions or exact weight values. Thus, the result of this method is a pruned Pareto set that clearly reflects the decision maker’s objective function preferences which are performed in the first part of the approach. In the second part, complimentary computations give some additional details that make it easier for the decision maker to comparatively analyse a set of these pruned solutions. The following notation is used to describe the proposed approach:

- $a_i$ acceptability index of the $i^{th}$ Pareto-optimal solution
- $w$ weight vector $[w_1,\ldots,w_n]$ ($w$ is a set includes all generated weight vectors)
- $w_j$ $j^{th}$ element of each weight vector
- $w_i$ central weight vector for $i^{th}$ Pareto-optimal solution
- $W_i$ set of favourable weight vector for $i^{th}$ Pareto-optimal solution
- $N$ number of weight sampling
- $n_i$ number of times that $i^{th}$ Pareto-optimal solution has been the best choice
\( L_{Bi} \) lower bound for \( j^{th} \) element of the favourable set of \( i^{th} \) Pareto-optimal solution

\( U_{Bi} \) upper bound for \( j^{th} \) element of the favourable set of \( i^{th} \) Pareto-optimal solution

\( LB_i \) lower bounds vector of favourable set for \( i^{th} \) Pareto-optimal solution

\( UB_i \) upper bounds vector of favourable set for \( i^{th} \) Pareto-optimal solution.

### 3.1 Part 1: pruning of Pareto-optimal set

Having a Pareto-optimal set, the objective functions are ranked non-numerically by the decision maker to prune the set. To achieve this, decision-makers express their preferences. Generally, these preferences can be categorised in four groups:

- **Random objectives**: A range for the objectives of the system may be sampled uniformly when it is impossible to reflect in advance decision maker’s preferences in that there is not any prior information about his preferences (Butler et al., 1997; Wang and Zionts, 2006).

- **Rank order objectives**: For this case, as we can see from Jimenez et al. (2003), the policy maker arranges non-numerically the weight preferences. In this regard, the simulation produces random objective while keeping the rank arrangement of weights.

- **Response distribution objectives**: For this group of decision maker’s preferences, it is recognised that the objective estimation procedure is under the influence of variation (Barron and Barrett, 1996; Jimenez et al., 2003). It is assumed that the decision maker knows weights, which equal the true objectives plus a response error. Using simulation methods, objectives are produced from these distributions, and then these generated weights are applied to decision making stage.

- **Random generation of objectives under the influence of constraint sets**: In some particular conditions, further detail of information may be provided by the decision makers. Note that this detailed information is predominantly extracted from constraint sets.

In this study, the last alternative of preferences is considered due to the fact that this alternative is flexible enough to reflect dominances. Furthermore, this alternative includes first and second alternative properties. Regarding the forth alternative, a number of examples of dominances which can be extracted via linear constraint sets are shown:

- **Weak decision maker’s preferences relation between alternatives**: \( w_i \geq w_j, i \neq j \)

- **Strict decision maker’s preferences relation between alternatives**: \( w_i - w_j \geq a, i \neq j \)

- **Decision maker’s preference about weights with ratio comparison**: \( w_i \geq b w_j, i \neq j \)

- **Interval decision maker’s preference on weights with numerical ranges**: \( c \leq w_i \leq c + \varepsilon \)

- **Decision maker’s preference about weights differences**: \( w_i - w_j \geq w_k - w_l \) for \( i \neq j \neq k \neq l \)

\( i, j, k, l = 1, 2,...n \), and \( a, b, c, \varepsilon \) are real numbers in aforementioned examples. The interested reader is referred for details of these constraints to Park and Kim (1997). In
general form, linear constraint sets $Aw \leq b$ are considered in this study. Regarding these linear constraints, the weight vector is shown by $w = (w_1, \ldots, w_n)$, $A$ is a matrix with $m$ rows and $n$ columns, $m$ shows the number of the constraints, and finally $b$ is a real $m$-vector. In this regard, a literature review is also provided by Weber (1987) and Park and Kim (1997).

After determination of preferences, an n-dimensional weight generator is developed to generate the weight vectors in the desired region determined by constraints. The algorithm used to prune Pareto-optimal solutions is described as follows:

- express the preferences
- normalise all objectives to be for maximisation and Scale objectives (from 0 to 1)
- randomly generate weights based on ranks using the weights generator
- sum weighted objectives to form a overall function
- from the Pareto-optimal set, find the solution ($s$) that yields the maximum value for $f'$
- increment the counter corresponding to that solution ($s$) by a value of one and also record this weight vector (the weight vector is used in this iteration) in set of favourable weight vectors of that solution ($s$)
- repeat steps 3 to 6 for $N$ (several thousand) times
- determine the pruned Pareto-optimal set, i.e., the solutions that have non-zero counter values and then compute indices (which will be defined consequently in second part) for each solution of pruned set
- calculate the mean of the weights for each solution in Pareto-optimal set.

Consider a problem with three objective functions. The weight generator is active in an area where all the weights in this set are non-negative and add up to one as shown in Figure 1.

**Figure 1** Plane containing set of possible weights

Assume the objective function preference is $f_1 > f_2 > f_3$, which means objective $f_1(x)$ is more important than objective $f_2(x)$, and objective $f_2(x)$ is more important than objective $f_3(x)$. Also, the objectives are similarly scaled. Although the exact value of the weights is unknown, it is known that $w_1 > w_2 > w_3$. This results in the sampling region shown in Figure 2.
Figure 2 Weight region for the $f_1 > f_2 > f_3$ objective function preference (see online version for colours)

The weights are sampled from the area of interest with the following distribution:

$$f_w(w) = \begin{cases} 
  c, & \text{desired region} \\ 
  0, & \text{otherwise} 
\end{cases}$$

where $c$ is a constant.

Then, random weights are generated by Monte-Carlo simulation methods (Wang and Zionts, 2006; Moeini et al., 2011). These weights are consistent with the desired region. A set of weight vectors is generated and each vector contains one weight for each objective. These weight vectors are applied frequently to aggregate the scaled objectives into an overall objective function. The best solution for each overall objective (from the set of the Pareto-optimal solutions available) and other needed information (according to step 6 of pruned algorithm) are recorded. This process is repeated many times (e.g., several thousand), and the end result is a pruned Pareto-optimal set. By using this method almost 90% of solutions will be outranked (Coit and Baberanwala, 2005). Similar approaches have been explored in multiple criteria decision analysis. For example (Rietveld and Ouwersloot, 1992; Hinloopen et al., 2004) explain methods where solutions must be selected based on ranked preference. It should be noted that, although the example illustrated here was for the ordinal case, the method can handle for all kind of linear constraints.

3.2 Part 2: complementary information about solutions of pruned set

Up to now, for the solution $i$ of Pareto-optimal set, it is determined that the set of weight vectors $W_i$ which makes the $(f')$ of solution $i$ greater than or equal to the $(f')$ of any other solutions. $W_i$ is called the set of favourable weight vectors for solution $i$, because solution $i$ becomes the best solution (not necessarily unique) with any $w \in W_i$.

It is obvious that the set of favourable weight vectors are non-empty only for the solutions of pruned sets, so the following analyses are required for these solutions.
3.2.1 Central weight vector

The central weight vector for the $i^{th}$ Pareto-optimal solution is an unbiased estimate for $w$ under the condition that the solution $i$ is preferred based on uniform weight distributions. It is obtained by computing the mean on the set of favourable weight vectors for the $i^{th}$ Pareto-optimal solution.

$$W_i^c = \frac{1}{n_i} \sum_{w \in W_i} w$$  \hspace{1cm} (1)

The central weight vector is the best single vector representation of the valuations of a typical decision maker who claims to prefer the solution $i$.

3.2.2 Bounds for the favourable weight vectors

The ranges for the favourable weights can also be useful for the decision maker. They can be easily obtained from the favourable weights for each solution of the pruned set.

$$LB_i = \left( LB_{i1}, LB_{i2}, LB_{i3}, \ldots, LB_{in} \right)$$  \hspace{1cm} (2)

$$UB_i = \left( UB_{i1}, UB_{i2}, UB_{i3}, \ldots, UB_{in} \right)$$  \hspace{1cm} (3)

where

$$LB_{ij} = \min \left( w_j \right) \left\{ w \in W_i \right\}, \quad UB_{ij} = \max \left( w_j \right) \left\{ w \in W_i \right\}, \quad w = [w_1, w_2, \ldots, w_n]$$

It is noteworthy that the interpretation of $w_i^c$ is very simple; that is minimum of $i^{th}$ elements of the set of generated weight vectors which cause $i^{th}$ pruned solution becomes the best choice.

3.2.3 Acceptability index

The acceptability index $a_i$ for solution $i$ is defined as the ratio between the $n_i$ and $N$:

$$a_i = \frac{n_i}{N}$$  \hspace{1cm} (4)

The acceptability index is a measure of the variety of different valuations (represented by feasible weight vectors) which allow for the solution that yields the maximum value. The acceptability index can also be interpreted as the probability for a certain solution to be the best solution (based on uniform weight distributions).

Ultimately, the information can be used to comprehend and compare solutions for doing final selection. For example, if a decision maker can realize his preference is similar to which central weight vector, then the central weight vector will be a good instrument for the final selection. When the decision-maker is faced with high uncertainty, the acceptability index offers a more rational approach. The decision-maker simply selects the solution with the highest acceptability index as a final selection.

Similar analyses were applied by Lahdelma et al. (1998) that computed these indices analytically, but this approach required time-consuming and expensive computations.
In our proposed approach, the information was estimated with enough accuracy by Monte-Carlo simulation.

4 Redundancy allocation problem

The RAP is subject to a system consisting of $s$ subsystems arranged in series. Within each subsystem, functionally equivalent components exist which vary in cost, weight, reliability, etc., each of which can be independently applied in the subsystems. While only one copy of each component is essentially needed in each subsystem, for improved reliability purposes, application of redundant components is recommended. A sample series-parallel system is shown in Figure 3.

Although application of redundant components improves the reliability, it also increases system cost and weight, which are negative impacts on the system. With respect to system-level constraints, the aim of a multi-objective RAP is to find an optimal design configuration which maximises reliability and minimises cost and weight at the same time.

Figure 3 Series-parallel system configuration

The mathematical formulation of the multi-objective RAP can be shown as:

\[
\begin{align*}
\max & \left[ R = \prod_{j=1}^{j_{\max}} \left(1 - \prod_{i=1}^{i_{\max}} (1-r_{ij})^{x_{ij}} \right) \right], \quad \min \left[ C = \sum_{j=1}^{N} \sum_{i=1}^{n_{ij}} w_{ij} x_{ij} \right] \\
\text{s.t.} & \quad 1 \leq \sum_{j=1}^{j_{\max}} x_{ij} \leq n_{\max,i} \quad \forall i = 1, 2, \ldots, s \\
& \quad x_{ij} \in \{0, 1, 2, \ldots\}
\end{align*}
\]

where
- reliability, total cost and weight of the system are also shown as $R$, $C$, and $W$, respectively
- $s$ indicates the number of subsystems
- $x_{ij}$ represents the quantity of the $j^{th}$ component in subsystem $i$
- $n_{\max,i}$ shows the maximum number of components in parallel which is determined by the user in subsystem $i$
• $m_i$ refers to overall quantity of component in reach for subsystem $i$

• cost, weight and reliability values related to the $j^{th}$ component for subsystem $i$, are shown as $c_{ij}$, $w_{ij}$, $r_{ij}$.

4.1 Numerical example

A numerical example of RAP was solved for a system consisting of three subsystems, with an option of five, four and five types of components in each subsystem. The maximum number of components per subsystem is four. Table 1 defines the component choices for each subsystem.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Component choices for each subsystem</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Subsystem $i$</strong></td>
<td><strong>Design alternative $j$</strong></td>
</tr>
<tr>
<td></td>
<td>$r_{ij}$</td>
</tr>
<tr>
<td>1</td>
<td>0.90</td>
</tr>
<tr>
<td>2</td>
<td>0.85</td>
</tr>
<tr>
<td>3</td>
<td>0.82</td>
</tr>
<tr>
<td>4</td>
<td>0.79</td>
</tr>
<tr>
<td>5</td>
<td>0.99</td>
</tr>
</tbody>
</table>

Pareto-optimal solutions were obtained using (NSGA-II). As shown in Figure 4, this set includes 339 non-dominated solutions.

Figure 4 Pareto-optimal set of solutions (see online version for colours)

The pruned solutions are identified by using our proposed Monte-Carlo simulation method (based on 5,000 iterations). The objective function priorities used on these solutions were reliability > cost > weight ($R > C > W$). Figures 5 and 6 show the pruned solution sets (it includes nine solutions) for $w_1 > w_2 > w_3$, compared to the original Pareto-optimal set.
Figure 5  Comparing pruned Pareto solution with the Pareto-optimal solutions (see online version for colours)

Figure 6  Comparing pruned Pareto solution with the Pareto-optimal solution set for cost versus weight (see online version for colours)

Favourable weight sets for each solution of the pruned sets are shown in Figure 7. According to these favourable sets, complementary computations were done subsequently.

The summary of results obtained by these computations is shown in Tables 2 and 3, respectively. Table 2 shows the design configuration of pruned solutions, based on a sorting acceptability index.
Utilisation of pruned Pareto-optimal solutions

**Figure 7** Favourable weight vectors for each member of a pruned set (see online version for colours)

**Table 2** Design configuration and acceptability index for solutions of pruned set

<table>
<thead>
<tr>
<th>Solution</th>
<th>( a_i )</th>
<th>Design of solutions</th>
<th>Cost</th>
<th>Weight</th>
<th>Reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>332</td>
<td>0.7296</td>
<td>5 2 2</td>
<td>4</td>
<td>14</td>
<td>0.9023</td>
</tr>
<tr>
<td>278</td>
<td>0.0878</td>
<td>5 2 2</td>
<td>5</td>
<td>18</td>
<td>0.9654</td>
</tr>
<tr>
<td>285</td>
<td>0.0564</td>
<td>5 2 4</td>
<td>5</td>
<td>16</td>
<td>0.9605</td>
</tr>
<tr>
<td>339</td>
<td>0.0536</td>
<td>5 2 4</td>
<td>4</td>
<td>13</td>
<td>0.8732</td>
</tr>
<tr>
<td>279</td>
<td>0.035</td>
<td>5 2 2</td>
<td>6</td>
<td>24</td>
<td>0.9848</td>
</tr>
<tr>
<td>317</td>
<td>0.023</td>
<td>5 2 2</td>
<td>5</td>
<td>17</td>
<td>0.9634</td>
</tr>
<tr>
<td>109</td>
<td>0.0078</td>
<td>5 2 2</td>
<td>9</td>
<td>32</td>
<td>0.9992</td>
</tr>
<tr>
<td>55</td>
<td>0.003</td>
<td>5 2 2</td>
<td>9</td>
<td>31</td>
<td>0.999</td>
</tr>
</tbody>
</table>
Table 2  Design configuration and acceptability index for solutions of pruned set (continued)

<table>
<thead>
<tr>
<th>Solution</th>
<th>$a_i$</th>
<th>Design of solutions</th>
<th>Cost</th>
<th>Weight</th>
<th>Reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>76</td>
<td>0.0022</td>
<td><img src="image1" alt="Diagram" /></td>
<td>9</td>
<td>30</td>
<td>0.9988</td>
</tr>
<tr>
<td>120</td>
<td>0.0016</td>
<td><img src="image2" alt="Diagram" /></td>
<td>9</td>
<td>29</td>
<td>0.9985</td>
</tr>
</tbody>
</table>

Table 3 shows central weight vectors $w_i^c$ and bounds (lower bounds, upper bounds) for each solution in Pareto set.

Table 3  Central weight vector and the bounds for the favourable weight vectors

<table>
<thead>
<tr>
<th>Label</th>
<th>Lower bound</th>
<th>Upper bound</th>
<th>$w_i^c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>332</td>
<td>(0.3877, 0.1494, 0)</td>
<td>(0.7596, 0.4971, 0.2665)</td>
<td>(0.5699, 0.3142, 0.1159)</td>
</tr>
<tr>
<td>278</td>
<td>(0.7496, 0.0797, 0.0001)</td>
<td>(0.8724, 0.2372, 0.0581)</td>
<td>(0.8136, 0.1582, 0.0282)</td>
</tr>
<tr>
<td>285</td>
<td>(0.7072, 0.0776, 0.0665)</td>
<td>(0.8458, 0.1889, 0.1463)</td>
<td>(0.7668, 0.1378, 0.0954)</td>
</tr>
<tr>
<td>279</td>
<td>(0.8706, 0.0455, 0)</td>
<td>(0.9311, 0.1182, 0.0613)</td>
<td>(0.898, 0.078, 0.024)</td>
</tr>
<tr>
<td>339</td>
<td>(0.3368, 0.2732, 0.2313)</td>
<td>(0.4568, 0.3824, 0.3282)</td>
<td>(0.3923, 0.329, 0.2787)</td>
</tr>
<tr>
<td>317</td>
<td>(0.7485, 0.0688, 0.0547)</td>
<td>(0.868, 0.196, 0.0746)</td>
<td>(0.8036, 0.1336, 0.0628)</td>
</tr>
<tr>
<td>109</td>
<td>(0.9345, 0.0068, 0.0005)</td>
<td>(0.9878, 0.0585, 0.0145)</td>
<td>(0.9592, 0.0334, 0.0074)</td>
</tr>
<tr>
<td>55</td>
<td>(0.9301, 0.0273, 0.0158)</td>
<td>(0.9548, 0.0526, 0.0195)</td>
<td>(0.944, 0.0388, 0.0173)</td>
</tr>
<tr>
<td>76</td>
<td>(0.9286, 0.0249, 0.0204)</td>
<td>(0.9507, 0.0479, 0.0266)</td>
<td>(0.9407, 0.0356, 0.0237)</td>
</tr>
<tr>
<td>120</td>
<td>(0.9223, 0.0364, 0.0274)</td>
<td>(0.9319, 0.046, 0.0378)</td>
<td>(0.9275, 0.0399, 0.0326)</td>
</tr>
</tbody>
</table>

In this numerical example, a decision-maker can select a solution with a central weight vector similar to his/her preferences vector. Thus, when a decision maker is faced with high uncertainty; he/she can see that a promising solution for implementation is the solution with label 332, because it has a high acceptability index ($a_{332} = 0.7296$) which means that in most preference situations (more than 72%), this solution has dominated others.

5 Conclusions

The proposed method is a practical approach which makes a significant contribution to solving multi-objective problems and promisingly achieves a manageable set of solutions. There are two important extensions in this paper. Firstly, decision-maker’s preferences are not limited to ranks of importance of objective functions. The method gets the information of decision makers’ preferences via a set of linear constraints which can be so flexible including the case without any information to the cases with
Utilisation of pruned Pareto-optimal solutions

completely determined information. Indeed, this set of linear constraints can also include random weights and rank order weights, which shows that previous research is a subdivision of this study. Secondly, on the contrary to related studies, the proposed method determines an acceptability index for each solution of a pruned set, describing the variety of different valuations (weight combinations) that support the preference for the solution. The method also computes the central weight vector for each solution of a pruned set, corresponding to the typical valuations which a decision maker prefers in that solution. More information is provided on pruned Pareto-optimal sets and how they can assist a decision maker in conveniently comprehending and comparing them. Hence, the proposed method provides a very useful managerial insight into selecting final solution. Further studies will focus on extending the proposed method for handling non-linear constraints where the decision makers’ preferences are given via a set of non-linear constraints.

Acknowledgements

The authors would like to thank the anonymous reviewers and the editor for their insightful comments and suggestions.

References


