Visualization of heuristic-based multi-objective design space exploration of embedded systems
Taghavi Razavi Zadeh, T.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Multi-Objective Decision Making: Methods and Visualizations

One significant aspect of multi-objective optimization is the fact that finding the Pareto optimal solutions does not completely solve a Multi-Objective Optimization Problem (MOOP). Actually a MOOP involves two stages:

1. Searching the decision space and finding the Pareto optimal solutions
2. Selecting the most suitable solution from the Pareto optimal set

Finding only a set of Pareto optimal solutions is not sufficient. The decision maker still has to choose one solution from this set to be implemented. However, the process of choosing the most preferred solution among the several non-dominated solutions is not trivial. Therefore, many Multi-Objective Decision Making (MODM) methodologies have been suggested to guide decision makers towards the most appropriate solution among the Pareto optimal solutions.

In this chapter, we explain some MODM methods that are provided in VMODEX and can help decision makers to understand the trade-offs between different criteria and select the final solution for the implementation. Furthermore, new visualization approaches are proposed, which provide the visual interpretation and detailed analysis of the results of MODM methods.

5.1 Introduction

As we mentioned in Chapter 2 in multi-objective optimization with conflicting objectives, there is no single optimum solution that simultaneously optimizes all objectives. Instead, a set of Pareto optimal solutions has to be found in which no improvement can be obtained in any of the objectives without causing a simultaneous degradation in at least one other objective. However, after finding the Pareto optimal solutions of the multi-objective optimization problem, we are encountering some difficulties:
Although many efficient solutions have been found, only one or a reduced number of optimal solutions need to be considered for the final decision. Therefore, there is an issue regarding how a decision maker chooses the best solution from a set of Pareto optimal solutions.

In the mathematical sense, all Pareto optimal solutions are regarded as equally desirable. Thus, systematic approaches are needed to express preference information related to the multiple objectives and aid the decision maker to identify the most preferred solution. Without a systematic approach one cannot be sure that the proper decision has been made. Therefore, some decision making methods are provided in VMODEX to assist the decision makers to make better decisions.

Usually, the results of MODM methods are shown in a table or displayed in a 2D graph. Although these kinds of representations are useful and appropriate for gaining a general understanding of the results, they do not provide insight on how these results are achieved. In this chapter, we propose several visualization techniques that enable decision makers a deeper understanding and detailed analysis of the relations between Pareto optimal solutions and allow them to find out why one particular solution is considered as a more preferred solution with respect to a specific MODM method.

For describing the visualization techniques in VMODEX that are developed for visualizing the outcomes of MODM methods, we use a case study of which the results are used as input data for all the visualization approaches explained in this chapter. In our case study, we map a parallel multi-media application to an MPSoC platform architecture consisting of: two Application Specific Integrated Circuits (ASICs), an Application Specific Instruction Processor (ASIP), a general-purpose microprocessor (mP), a microcontroller (mC), two Dynamic RAMs (DRAMs) and one Static RAM (SRAM). The mapping decision problem (i.e. mapping application tasks and communication channels onto the architecture components) is formulated as a multi-objective optimization problem in which three criteria are considered: the processing time, energy consumption and cost of the architecture. To solve this problem, we utilized the SPEA2 [18] evolutionary algorithm to achieve a set of Pareto optimal solutions under the aforementioned criteria. The outcome of running the SPEA2 on our case study has resulted in 14 Pareto optimal points which are shown in Figure 5.1. Note that in this chapter, the considered objectives and their visual representations are the same as in Chapters 3 and 4. Processing time is shown by the color of the node representing a solution. Colors are varied from yellow to red with all color grades in between. Nodes with the lowest processing time are yellow and nodes with the highest processing time are red. The size and color of the third dimension of a solution shows the energy consumption. As the energy consumption increases, the size of the third dimension becomes bigger and its color becomes darker. The architecture cost is shown as separate nodes at the cost level. Cost nodes are represented with circle symbol. The size of the circle becomes bigger as the cost increases.

The rest of this chapter is organized as follows. In Section 5.2, we give some preliminary definitions that are used in multi-objective decision making methods. Section 5.3 describes the four basic problem formulations in MODM, which are: choice, clas-
Figure 5.1: Pareto optimal solutions found through our case study and visualized by VMODEX

sification/sorting, clustering and ranking problems. For each problem formulation, some methodologies (either existing ones or new ones) are explained. Furthermore, different visualization approaches are proposed for visualizing the results of MODM techniques. Finally, Section 5.4 concludes this chapter.

5.2 Preliminary Definitions

In this section, we explain some basic notions that are used in the Multi-Objective Decision Making (MODM) methods. Before introducing the preliminary definitions, the decision problem should be formally stated. In particular, we consider a set of $n$ solutions $S = \{s_1, \ldots, s_i, \ldots, s_n\}$. The solutions are evaluated in terms of $m$ objective functions $F = \{f_1, \ldots, f_j, \ldots, f_m\}$. The performance of solution $s_i$ on the objective function $f_j$ is denoted by $f_j(s_i)$. Note that in this chapter, index $i$ denotes the counter for the solutions and index $j$ indicates the counter for the objective functions.

5.2.1 Weights

Typically, multiple criteria have different importance. For instance, performance may be more important than cost to the decision-maker. Thus, weighting factors must be used to indicate the relative importance of multiple objectives. The decision maker expresses his preferences among different criteria by assigning a weight to each criterion. The weight of each objective represents its priority relative to the other objectives under consideration. The higher the assigned weight, the more important the objective is. Weights are usually normalized to sum up to 1, so that in a set of weights following condition should be satisfied:

$$\sum_{j=1}^{m} w_j = 1$$

Where $m$ is the number of objectives. There are several ways for deriving weights, however, in this chapter we only explain the two most common methods, which are: ranking and analytical hierarchy processes.

The ranking method is the simplest approach for calculating the weights. First, the objectives are ranked from the most important to the least important (the rank of the most important one is $m$, second most important is $m-1$, etc). Then the ranks are normalized into the range $[0, 1]$ by dividing by the sum of the ranks. Let $r_j$ be the
The weight of the $j^{th}$ objective ($w_j$) is calculated as:

$$w_j = \frac{r_j}{\sum_{k=1}^{m} r_k} \quad (5.1)$$

In the case that some objectives are equally important, an average rank is assigned to all of them. This average is calculated based on the ranks that they would have when they differ in importance. For example, consider a 4-objective case in which objective 1 is assessed to be the most important, objective 2 and 3 are equally important, and objective 4 is the least important. The ranks will be $r_1 = 4$, $r_2 = 2.5$, $r_3 = 2.5$, $r_4 = 1$.

The Analytic Hierarchy Process (AHP) has been developed by T. Saaty \cite{91, 92} and is based on pairwise comparison. The objectives are compared against one another. The result is a comparison matrix ($C$) in which the value of the entry $c_{jk}$ indicates the relative importance of the $j^{th}$ objective as compared with the $k^{th}$ objective. For each pair of objectives, the relative importance is determined by assigning a weight between 1 (equal importance) and 9 (extreme importance) to the more important objective, whereas the reciprocal of this value is assigned to the other objective in the pair. Thus, in the comparison matrix $C$, we have $c_{jk} = c_{kj}^{-1}$ and $c_{jj} = 1$. This means that only $\frac{1}{2}m(m - 1)$ comparisons are needed to establish the objective pairwise comparison matrix for $m$ objectives.

The next step is to calculate a set of weights $W = (w_1, w_2, ..., w_m)$ that are most consistent with the relativities determined in the comparison matrix. This means that for each element $c_{jk}$ of matrix $C$, the corresponding ratio of objective weights ($w_j / w_k$) should be as close as possible to the value of $c_{jk}$. There are several techniques for estimating the weighting factors from the comparison matrix. Here we describe the Eigenvector approach, which obtains the weighting vector in a short computational time. The procedure of calculating weights is as follows:

1. Compute the square of comparison matrix.
2. Sum the rows and then normalize the sum values by dividing by the total sum of the rows. The result is the Eigenvector that is considered as the weighting vector.
3. Repeat step 1 and 2 until the difference between two consecutive vectors is smaller than a predefined value.

For better understanding the AHP, we present a numerical example. In this example, we consider a 3-objective problem and we use the AHP approach to calculate the weights of these three criteria. Assume that the following matrix is our comparison matrix (Note that we used four decimal places in our calculations):
5.2. Preliminary Definitions

\[ C = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{3}{4} \\ \frac{2}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix} = \begin{bmatrix} 1.0000 & 0.5000 & 3.0000 \\ 2.0000 & 1.0000 & 4.0000 \\ 0.3333 & 0.2500 & 1.0000 \end{bmatrix} \]

\[ C^2 = \begin{bmatrix} 3.0000 & 1.7500 & 8.0000 \\ 5.3332 & 3.0000 & 14.0000 \\ 1.1666 & 0.6667 & 3.0000 \end{bmatrix} \]

\[ W = \begin{bmatrix} 3.0000 + 1.7500 + 8.0000 \\ 5.3332 + 3.0000 + 14.0000 \\ 1.1666 + 0.6667 + 3.0000 \end{bmatrix} = \begin{bmatrix} 12.7500 \\ 22.3332 \\ 4.8333 \end{bmatrix} \]

\[ \text{sum} = 39.9165 \]

\[ W = \begin{bmatrix} 0.3194 \\ 0.5595 \\ 0.1211 \end{bmatrix} \]

The difference between this vector and the previous one is not a lot with respect to four decimal places. Therefore, the iterative process can be terminated. As a result, we obtained the following weights for our three objectives: \( w_1 = 0.3196, w_2 = 0.5584, w_3 = 0.1220. \)

5.2.2 Utility Functions

Utility functions are widely used in MODM for describing the preferential modeling. A utility function \( U \) is a mathematical representation of the decision maker’s preferences such that \( U(x) > U(y) \) means that alternative \( x \) is preferred over alternative \( y \) and \( U(x) = U(y) \) indicates that both alternatives are equally preferred (\( x \) is indifferent from \( y \)).

For each objective function \( f_j \), a marginal utility function \( u_j \) needs to be defined. A marginal utility function is a monotone function (linear or nonlinear) defined on the criterion scale and provides a mechanism for transforming the scale of the criterion into a new scale ranging in the interval \([0, 1]\), by taking into account the following two conditions:

\[ u_j(f_j^{\text{best}}) = 1 \quad \text{and} \quad u_j(f_j^{\text{worst}}) = 0 \]
5.2. Preliminary Definitions

Figure 5.2: Examples of utility functions for typical preference models

Where \( f_{j}^{\text{best}} \) denotes the most preferred and \( f_{j}^{\text{worst}} \) worst indicates the least preferred value of criterion \( f_{j} \). These values are calculated according to all solutions in the set \( S \), as follows:

- For objectives in which higher values indicate higher preference
  \[
  f_{j}^{\text{best}} = \max_{i=1}^{n} f_{j}(s_{i}) \quad \text{and} \quad f_{j}^{\text{worst}} = \min_{i=1}^{n} f_{j}(s_{i})
  \]

- For objectives in which higher values indicate lower preference
  \[
  f_{j}^{\text{best}} = \min_{i=1}^{n} f_{j}(s_{i}) \quad \text{and} \quad f_{j}^{\text{worst}} = \max_{i=1}^{n} f_{j}(s_{i})
  \]

The form of the marginal utility functions depends upon the decision maker's preferential system. Figure 5.2 shows some examples of utility functions for three typical preference models. Figure 5.2(a) represents a concave form of the utility function. In this case, a small deviation from the worst value leads to a significant improvement in the preference score. This describes a situation in which the decision maker is satisfied with acceptable solutions and does not necessarily look for solutions with high performance. Figure 5.2(b) shows an inverse situation. In this case, the decision maker is mainly interested in solutions with high performance. The linear marginal utility function shown in Figure 5.2(c) indicates that the preference increases linearly by moving from the worst to the best criterion value.

The global utility provides an overall evaluation of the performance of the solution \( s_{i} \) when all criteria are taken into account. The simplest way for obtaining global utility is the weighted sum of the utility values of all criteria, as follows:

\[
U(s_{i}) = \sum_{j=1}^{m} w_{j} u_{j}(f_{j}(s_{i}))
\]

(5.2)

Where \( U(s_{i}) \) is the global utility of solution \( s_{i} \) and \( w_{j} \) is the weight of the \( j^{\text{th}} \) objective. The global utilities are in the range \([0, 1]\).
5.2.3 Preference Functions

Unlike the utility functions that assign an absolute utility to each solution (globally or on each criterion), the preference functions as introduced by Brans et al. [93, 94], are based on pairwise comparisons between solutions. The preference function $PF$ is used as a measure of intensity of preference of solution $x$ over solution $y$. This function should be defined separately for each criterion and its value is in the range [0,1]. In this case, the difference of performance between two solutions on a particular criterion $(f_j(x) - f_j(y))$ is considered as a degree of preference, $P_j(x, y)$, as follows:

$$P_j(x, y) = PF_j(d_j(x, y))$$

where

$$d_j(x, y) = f_j(x) - f_j(y)$$

for which

$$0 \leq P_j(x, y) \leq 1$$

A smaller deviation indicates the greater indifference and therefore a smaller preference value is allocated to the better solution and even possibly zero if the deviation is negligible. The larger deviation implies greater preference and the value is closer to one. Thus, the degree of preference $P_j(x, y)$ represents the decision maker’s preference intensity of $x$ over $y$ on the $j^{th}$ criterion as follows:

- $P_j(x, y) = 0$ represents indifference or no preference, $f_j(x) \simeq f_j(y)$
- $P_j(x, y) \simeq 0$ represents weak preference, $f_j(x) > f_j(y)$
- $P_j(x, y) \simeq 1$ represents strong preference, $f_j(x) \gg f_j(y)$
- $P_j(x, y) = 1$ represents strict preference, $f_j(x) \gg f_j(y)$

For each pair of solutions $x$ and $y$, the following condition is true:

$$P_j(x, y) > 0 \Rightarrow P_j(y, x) = 0$$

In [94], six types of particular preference functions are defined that can cover most of the practical applications. Figure 5.3 describes the three most commonly used of these functions. The parameters $p$ and $q$ have to be identified by the decision maker. The meaning of these parameters is as follows:

- $q$ is an indifference threshold. When the difference is smaller than $q$ it is considered as negligible by the decision maker. Therefore, the corresponding preference degree is zero.
- $p$ is the strict preference threshold. If the difference is greater than $p$ it is considered to be significant enough to generate a strict preference. Therefore the preference degree is one.
5.2. Preliminary Definitions

### Preference Function

<table>
<thead>
<tr>
<th>Preference Function</th>
<th>Definition</th>
</tr>
</thead>
</table>
| ![U-shape criterion](image) | \( P_F(d) = \begin{cases} 
0 & d \leq q \\
1 & d > q 
\end{cases} \) |
| ![Level criterion](image) | \( P_F(d) = \begin{cases} 
0 & d \leq q \\
0.5 & q < d \leq p \\
1 & d > p 
\end{cases} \) |
| ![V-shape criterion with indifference criterion](image) | \( P_F(d) = \begin{cases} 
0 & d \leq q \\
\frac{d-q}{p-q} & q < d \leq p \\
1 & d > p 
\end{cases} \) |

**Figure 5.3**: Types of Preference functions

The preference functions shown in Figure 5.3 are applicable for the criteria that need to be maximized. In the case of criteria that need to be minimized, the preference function should be reversed or alternatively given by:

\[
P_j(x, y) = P_F(-d_j(x, y))
\]

The global preference index, \( \pi(x, y) \), is defined as the weighted sum of the preference functions \( P_j(x, y) \) for all the criteria:

\[
\pi(x, y) = \sum_{j=1}^{m} w_j P_j(x, y)
\]  \hspace{1cm} (5.3)

for which \( 0 \leq \pi(x, y) \leq 1 \)

Where \( w_j \) is the weight given to the criterion \( j \). \( \pi(x, y) \) indicates the degree in which solution \( x \) is preferred over \( y \) for all the criteria. It is clear that \( \pi(x, y) \approx 0 \) implies a weak global preference of \( x \) over \( y \) and \( \pi(x, y) \approx 1 \) implies a strong global preference of \( x \) over \( y \).
5.3 Problem Formulations in Multiple-Objective Decision Making

The aim of Multi-Objective Decision Making (MODM) is to help decision makers to comprehensively understand the trade-offs between different criteria and guide them towards the most preferred decision. The decision making methods allow decision makers to apply their preferences to a decision problem in a logical and mathematical approach. The final recommendation in MODM may take different forms, according to the manner in which a problem is formulated. Four basic problem formulations are as follows:

- **The choice** problem that aims to select a subset of Pareto optimal set, as small as possible, which contains the most satisfying solutions.

- **The classification/sorting** problem that intends to assign each solution to one of the predefined groups.

- **The clustering** problem that aims to assign Pareto optimal solutions to different clusters according to some similarity measure.

- **The ranking** problem that aids to determine the relative goodness of each solution in the Pareto optimal set as compared with the other solutions. The result is usually presented as a ranking of the Pareto optimal points from the best to the worst, with respect to the decision maker preferences.

While the aim of both classification and sorting is the same, they differ depending on whether the groups are preferentially ordered or not. Classification refers to the case where the classes are defined in a nominal way (i.e. there is no order on the classes) whereas in sorting the classes are defined in an ordinal way starting from those containing the most preferred solutions to those containing the least preferred solutions.

It is necessary to emphasize the difference between classification and clustering. They have two distinctive differences: 1) in the classification problems, the aim is to assign objects to the groups that are defined a priori and these groups are so-called "classes". Whereas in the clustering, there is no knowledge about the groups in advance and the groups are so-called "clusters". Thus, the purpose is to elicit clusters of objects sharing similar characteristics. 2) Classification is based on absolute judgments, while clustering is based on relative judgments. In classification, the assignment of an object to an appropriate class relies on the intrinsic evaluation of that object, on all criteria, with respect to the profiles defining the classes (not on the comparison of that object to the other objects in the set). On the contrary, clustering is based on comparing the objects inside a set with each other, according to some similarity measure, and organizing them into clusters including objects with similar characteristics.

In the following subsections, we explain some decision making methods for each problem formulation.
5.3. Choice Problem

In VMODEX, we provide two approaches for selecting a set of best Pareto optimal solutions: filtering and a fuzzy scheme.

Filtering

By applying filtering operations, any solution that does not satisfy the decision maker’s preference constraints will be left out for further consideration, and the decision making process proceeds with a set of most preferred design points.

In VMODEX, there are several ways in which the preferences of decision makers can be defined. The decision maker can determine his preference objective values by setting the upper and lower limits on each objective. Then, those solutions with objective values inside the selected ranges are chosen as most preferred solutions and all the other solutions that exceed the (objective values) limits are eliminated from the Pareto optimal set.

The decision maker is also able to determine his preferences on the design parameters and consider only the solutions with preferred parameters. For instance, in our case study, the designer may prefer the points that have an ASIC in their underlying architecture. Therefore, by applying the filtering option, the points without an ASIC will be eliminated from the Pareto optimal set.

Another possibility for filtering the solutions, which is specific to our domain, is based on the mapping decision (mapping of the application tasks and channels onto the architecture components). The decision maker can indicate his preferences on the mapping (e.g. a specific task should be mapped to a specific processor or some specific tasks should be mapped on the same processor) and then filter the design points to see only those solutions that do not violate the determined mapping constraints. A more detailed description about the mapping filtering is given in Chapter 3, Section 3.2.5. Establishing the preferences on the combination of objective values, design parameters and mapping decisions is provided as well.

Fuzzy Scheme

Fuzzy logic is a superset of conventional (Boolean) logic that has been extended to handle the concept of partial truth, being values that lie between "absolutely true" and "absolutely false". Boolean logic deals with situations that can be true or false. However, fuzzy logic allows degrees of truth (expressed as a membership function) in the range of zero to one. A degree of zero means absolutely false and a degree of one means absolutely true.

In a simple example, consider the set of "tall" men. The conventional method requires establishing a height threshold (e.g. 170 cm) and any person with the height more than the threshold is declared as a "tall" man and with the height less than the threshold as "not tall". So, a man with 171 cm would consider as a "tall" man (is a member of tall
5.3. Problem Formulations

set) and a man with 169 cm would not consider as a "tall" man (is not a member of tall set). Therefore, in Boolean logic, the degree of membership can be zero or one. Figure 5.4(a) shows the "tall" function in Boolean logic. However, using the fuzzy logic approach, membership in the set of "tall" men is a number between zero and one. An example of a fuzzy membership function for "tall" set is shown in Figure 5.4(b). The degree of membership for men with 171 cm height is one and for men with 169 cm is 0.95. Therefore, if person \( x \) has a higher degree of membership in "tall" set than person \( y \) this means that \( x \) is taller than \( y \).

We can use fuzzy logic to extend the notion of the dominance relation between two solutions. Therefore, instead of saying "\( x \) dominates \( y \)" or "\( x \) does not dominate \( y \)" we can say that "\( x \) dominates \( y \) by degree \( \mu \)." This fuzzy approach enables us to compare non-dominated solutions with each other and quantify the goodness of each solution within a Pareto optimal set.

**Fuzzy Dominance** For computing the dominance degree of solution \( x \) over solution \( y \), we should consider for each objective, in which degree solution \( x \) is better than or equal to solution \( y \). An example of definition of fuzzy sets for "better" and "equal" functions is shown in Figure 5.5. Here we consider a minimization problem, which means that all objectives are to be minimized and therefore smaller objective values are better. In Figure 5.5, two parameters \( \varepsilon \) and \( \gamma \) should be determined by the decision maker who is familiar with the problem. \( \varepsilon \) indicates the interval within which an improvement or degradation on the objective is meaningless and \( \gamma \) defines the relevant but not a significant improvement. The value of these parameters may be different for each objective. Although these parameters should be identified carefully, their intended meaning is evident and a decision maker with prior knowledge on the problem can easily specify them.

A straightforward definition for the fuzzy dominance relation can be expressed as
Equation 5.4 It is said that solution $x$ dominates solution $y$ by degree $\mu_d(x, y)$ with:

$$
\mu_d(x, y) = \frac{1}{m} \mu_b(x, y) + \frac{1}{2m} \mu_e(x, y)
$$

Where $\mu_b(x, y) = \sum_{j=1}^{m} \mu^j_b(f_j(x) - f_j(y))$

$$
\mu_e(x, y) = \sum_{j=1}^{m} \mu^j_e(f_j(x) - f_j(y))
$$

Where $m$ is the number of objectives and $f_j$ is the $j^{th}$ objective function. $\mu^j_b$ indicates the degree of "betterness" and $\mu^j_e$ denotes the degree of "equality" on the $j^{th}$ objective. $\mu_d(y, x)$ shows the dominance degree of $y$ over $x$ or, on the other hand, indicating in which degree solution $x$ is dominated by solution $y$. Note that the fuzzy dominance relation is not symmetric and therefore "dominating by degree $\mu$" and "being dominated by degree $\mu$" have different fuzzy values.

For clarifying the fuzzy dominance relation described above, we give a numerical example. In this example, we calculate the dominance degrees for two non-dominated solutions $x = (0.1, 0.5, 0.3)$ and $y = (0.6, 0.4, 0.57)$. The fuzzy membership functions for "better" and "equal" are the same as ones shown in Figure 5.5. For all objectives $\varepsilon = 0.1$ and $\gamma = 0.3$. Figure 5.6 shows the values of $\mu_b$ and $\mu_e$ for all
5.3. Problem Formulations

The dominance degree of $x$ over $y$ is computed as follows:

$$
\mu_b(x, y) = \mu_1^b(0.1 - 0.6) + \mu_2^b(0.5 - 0.4) + \mu_3^b(0.3 - 0.57) \\
= \mu_1^b(-0.5) + \mu_2^b(0.1) + \mu_3^b(-0.27) = 1 + 0.85 = 1.85
$$

$$
\mu_e(x, y) = \mu_1^e(-0.5) + \mu_2^e(0.1) + \mu_3^e(-0.27) = 0 + 0.15 = 1.15
$$

$$
\mu_d(x, y) = \frac{1}{3}\mu_b(x, y) + \frac{1}{6}\mu_e(x, y) = \frac{1}{3}(1.85) + \frac{1}{6}(1.15) = 0.81
$$

In the same way we can calculate the dominance degree of $y$ over $x$, $\mu_d(y, x)$, which is 0.19. Therefore $x$ dominates $y$ by degree 0.81 and is dominated by $y$ by degree 0.19. Thus solution $x$ is more preferred than $y$.

In this thesis, we propose a choice approach based on the fuzzy dominance relations between the solutions in the Pareto optimal set $S$. The decision maker should determine a dominance threshold ($\theta$). Then, each solution $s_i \in S$ can be defined as a 2-tuple $<D^+, D^->$ where:

$$
D^+(s_i) = \{ s_k \in S \mid \mu_d(s_i, s_k) > \theta \} \\
D^-(s_i) = \{ s_k \in S \mid \mu_d(s_k, s_i) > \theta \}
$$

$D^+(s_i)$ is a set of solutions such that $s_i$ dominates them with degree higher than the threshold and $D^-(s_i)$ contains those solution for which $s_i$ is dominated by them with degree higher than the threshold. Considering the 2-tuple representations of solutions, we define two ways for selecting the most satisfying solutions:

1. Select those solutions for which $|D^+(s_i)| > |D^-(s_i)|$. Where $||$ indicates the number of solutions in the set. So, each solution that more dominates rather than is dominated by a degree higher than $\theta$ is chosen as a preferred solution.
2. Specify a difference threshold \( \theta' \), and then select all solutions with \( |D^+(s_i)| - |D^-(s_i)| > \theta' \). The first method is a specific instance of this approach for which \( \theta' \) is equal to zero.

For better analysis of the fuzzy dominance relations between non-dominated solutions in a Pareto optimal set, we visualize their relations in a directed graph called Fuzzy Dominance Graph (FDG). This graph shows more detailed information about the dominance degrees and can be used to compare the solutions. Figure 5.7 represents the fuzzy dominance graph for our case study. Each solution is shown as a node in the FDG. The dominance threshold \( \theta \) is used for connecting the nodes. For each two solutions \( s_i \) and \( s_k \), if the dominance degree of \( s_i \) over \( s_k \) is greater than the threshold \( \mu_{d}(s_i, s_k) > \theta \) then a directed edge is established from \( s_i \) to \( s_k \). Thus, for each node (solution) in the FDG, the outgoing edges indicate \( D^+ \) and the incoming edges denote \( D^- \). A node with more outgoing edges means that it dominates more solutions with the degree higher than \( \theta \) and a node with more incoming edges means it is dominated by more solutions with the degree higher than \( \theta \). The size of each solution \( s_i \) indicates the value of \( |D^+(s_i)| - |D^-(s_i)| \). The bigger the node means the greater difference between the number of solutions in \( D^+ \) and \( D^- \) and therefore the solution is more preferred. In Figure 5.7 the threshold is 0.6 and, as can be seen in this figure, solution \( P5 \) is the biggest node in the graph. It has 10 outgoing edges (\( |D^+(P5)| = 10 \)) which shows that it dominates 10 (out of 14) solutions with a degree higher than \( \theta \) while there is no incoming edge (\( |D^-(P5)| = 0 \)). There is no solution that dominates \( P5 \) with a degree higher than \( \theta \). The solution \( P3 \) is the smallest node in the graph. It has no outgoing edge, whereas it is dominated by more than half of the solutions (8 out of 14) with a degree higher than \( \theta \) (\( |D^+(P3)| - |D^-(P3)| = 0 - 8 = -8 \)). In Figure 5.7 solutions with \( |D^+| - |D^-| > 0 \) are highlighted by a blue background color, which are \( P5, P2, P11, P7, \) and \( P13 \) in the order of decreasing the value of \( |D^+| - |D^-| \).

If the decision maker wants to know more about the dominance degrees of one specific solution with respect to the other solutions, it is possible to select that particular solution and see more detailed information. In Figure 5.8 the dominance relations between \( P5 \) and all the other solutions are shown. For each solution, the length of the blue bar represents the dominating degree of \( P5 \) over that solution and the length of the red bar shows the degree in which \( P5 \) is dominated by that specific solution. As can be seen in this figure, except \( P2 \), the dominance degree of \( P5 \) over each solution (blue bar) is higher than the degree in which \( P5 \) is dominated by that solution (red bar).

### 5.3.2 Classification/Sorting Problem

The assignment of objects into predefined groups is referred to as the classification/sorting problem, depending on whether the groups are specified in nominal or ordinal terms. In this type of problem, groups are usually defined independently of the objects under consideration and assigning an object to the appropriate group requires the absolute comparison of that object to some reference profiles that distinguish the
5.3. Problem Formulations

Figure 5.7: Visualization of fuzzy dominance relations

Figure 5.8: Visualization of fuzzy dominance relations between P5 and all the other solutions
In VMODEX, for sorting a set of Pareto optimal solutions in the predefined classes, we have utilized the UTADIS method (UTilits Additives DIScriminantes) \cite{95}. The UTADIS method is one of the simplest and most popular methodologies for addressing the sorting problems. UTADIS employs utility functions to classify a set of alternatives (in our case Pareto optimal solutions) into predefined ordinal groups.

Before proceeding with describing the UTADIS method, the problem should be formally stated. In particular, the problem under consideration is to assign a set of $n$ Pareto optimal solutions $s_i (i = 1, 2, \ldots, n)$ into $q$ predefined ordinal classes $C_k (k = 1, 2, \ldots, q)$. Where $C_1$ indicates the group including the most preferred solutions and $C_q$ denotes the group of the least preferred solutions ($C_1 > C_2 > \ldots > C_q$).

The UTADIS method uses the global utilities for assigning solutions to the predefined classes. The global utility of solution $s_i$ is computed using Equation 5.2, in which the weighted sum of the utility values of all criteria is considered as a global utility. The sorting process is performed through comparing the global utilities of solutions with some utility thresholds that specify the lower bound of each class, as follows:

$$U(s_i) \geq u_1 \Rightarrow s_i \in C_1$$
$$u_2 \leq U(s_i) < u_1 \Rightarrow s_i \in C_2$$
$$\ldots \ldots$$
$$u_k \leq U(s_i) < u_{k-1} \Rightarrow s_i \in C_k$$
$$\ldots \ldots$$
$$U(s_i) < u_{q-1} \Rightarrow s_i \in C_q$$

Where $u_1 > u_2 > \ldots > u_{q-1}$ are the utility thresholds that are considered as cut-off points to discriminate the classes. The utility threshold $u_k$ separates two consecutive classes $C_k$ and $C_{k+1}$. The parameters of the UTADIS method (utility functions and thresholds) are estimated through a linear programming formulation, so that the minimum classification error is achieved. For further details on the model development process, one is referred to the work of \cite{96}.

### 5.3.3 Clustering Problem

The goal of clustering is to organize a collection of objects into several groups (clusters), such that objects within the same cluster are similar (in some way), while objects belonging to different clusters are dissimilar. Cluster analysis enables us to look at the properties of clusters instead of individual objects and provides simplification of data with minimal loss of information. The clustering methods use a similarity/distance measure as fundamental to evaluate the closeness/separation of the objects under consideration. In general, a similarity measure maps the similarity between the parameterized descriptions of two objects into a single numeric value. The definition of the similarity measure is a key issue for accurate clustering and is closely related to
the problem context. Note that the term similarity is often conceived in terms of dissimilarity or distance as well. Dissimilarity corresponds to the notion of distance. Thus, smaller distance means smaller dissimilarity, and thus larger similarity. A variety of similarity/distance measures have been proposed and widely applied, such as Euclidean distance, cosine similarity, Jaccard coefficient, etc.

In the conventional clustering, the similarity measure is based on the geometric distances or related metrics. However, such metrics are not well suited for the clustering problems in the context of MODM. They do not incorporate the decision maker’s preferences in the cluster analysis. In the MODM field, a clustering problem is considered as a preference similarity oriented problem in which clusters should be conceived in terms of preference closeness. Thus, the similarity measure should be defined based on the preference relations between the objects such that all objects inside the same cluster are similar in the sense that they have the same preference structure. The preference similarity measures enable us to partition a set of objects into clusters that are meaningful from the MODM perspective.

To clarify the importance of a preference similarity measure, in the MODM context, consider the example shown in Figure 5.9. In this example, both objectives \( f_1 \) and \( f_2 \) are decreasing preference criteria (lower value indicates higher preference) and they have similar importance (same weight). Suppose that we intend to group the four solutions \( A, B, C \) and \( D \) into two clusters. By applying the Euclidean distance as a similarity measure, the two solutions \( A \) and \( B \) would be in the same cluster and the two solutions \( C \) and \( D \) would be in the other cluster. However, note that, \( A \) dominates \( B \) (is better in both objectives) and \( C \) dominates \( D \). Therefore, the decision maker prefers \( A \) over \( B \) and \( C \) over \( D \). So, in terms of preference, \( A \) and \( B \) are not considered the same and thus they should not be grouped in the same cluster. This is true for solutions \( C \) and \( D \) as well. So, the oriented preference clustering should separate \( A \) from \( B \) and \( C \) from \( D \). By involving the decision maker’s preferences in the clustering, the two solutions \( A \) and \( C \) would be in the same cluster, since they are not dominated by any other solutions, and the two solutions \( B \) and \( D \) would be in the other cluster, since they are dominated by the other solutions.

**Figure 5.9:** A trivial example for illustrating the importance of preference similarity measure.
The importance of integrating decision maker’s preferences in the multi-objective cluster analysis was firstly pointed out by [97]. They introduced a distance based on the preference structure defined by the decision maker. Up to now, however, relatively little research has been done on addressing the multi-objective preference clustering [98–100]. In this thesis, we propose a new preference similarity measure, which is based on the preference relations between solutions. In our approach, the preference functions (see Section 5.2.3) are utilized to identify the preference structure. In the following section, we will give the detailed description of our proposed preference similarity measure. In the rest of this chapter, we simply refer to preference similarity measure as the similarity measure.

Proposed Preference Similarity Measure

For better understanding the procedure of calculating the similarity between two solutions, we explain our approach via a simple numerical example. This example is shown in Figure 5.10. Suppose that both objectives are decreasing preference and the objective \( f_1 \) is more important. Their relative importance is such that \( w_1 = 0.7 \) and \( w_2 = 0.3 \). For both objectives, the “Level Criterion” preference function is used for constructing the preference structure. This preference function is displayed in Figure 5.3.

As we described in Section 5.2.3, for each two solutions \( s_i \) and \( s_k \), the global preference degree of \( s_i \) over \( s_k \), \( \pi(s_i, s_k) \), is defined as the weighted sum of the preference functions \( P_j(s_i, s_k) \) for all the criteria:

\[
\pi(s_i, s_k) = \sum_{j=1}^{m} w_j P_j(s_i, s_k)
\]

for which \( 0 \leq \pi(s_i, s_k) \leq 1 \)
5.3. Problem Formulations

Table 5.1: \( \pi \) values (global preference) of pairwise comparisons between the solutions in our numerical example

<table>
<thead>
<tr>
<th>( \pi )</th>
<th>( s_1 )</th>
<th>( s_2 )</th>
<th>( s_3 )</th>
<th>( s_4 )</th>
<th>( s_5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s_1 )</td>
<td>0</td>
<td>0.35</td>
<td>0.5</td>
<td>0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>( s_2 )</td>
<td>0.3</td>
<td>0</td>
<td>0.3</td>
<td>0.3</td>
<td>0.5</td>
</tr>
<tr>
<td>( s_3 )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.35</td>
</tr>
<tr>
<td>( s_4 )</td>
<td>0</td>
<td>0</td>
<td>0.15</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( s_5 )</td>
<td>0</td>
<td>0</td>
<td>0.3</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.2: \( \Pi \) values (net preference) of pairwise comparisons between the solutions in our numerical example

<table>
<thead>
<tr>
<th>( \Pi )</th>
<th>( s_1 )</th>
<th>( s_2 )</th>
<th>( s_3 )</th>
<th>( s_4 )</th>
<th>( s_5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s_1 )</td>
<td>0</td>
<td>0.05</td>
<td>0.5</td>
<td>0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>( s_2 )</td>
<td>-0.05</td>
<td>0</td>
<td>0.3</td>
<td>0.3</td>
<td>0.5</td>
</tr>
<tr>
<td>( s_3 )</td>
<td>-0.5</td>
<td>-0.3</td>
<td>0</td>
<td>-0.15</td>
<td>0.05</td>
</tr>
<tr>
<td>( s_4 )</td>
<td>-0.7</td>
<td>-0.3</td>
<td>0.15</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( s_5 )</td>
<td>-0.7</td>
<td>-0.5</td>
<td>-0.05</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Definition 1 (Net Preference): We define \( \Pi(s_i, s_k) \) which represents the net preference of \( s_i \) over \( s_k \) and is computed using Equation 5.5. The net preference \( \Pi(s_i, s_k) \) is the balance between the degree in which \( s_i \) is preferred to \( s_k \) and the preference degree of \( s_k \) over \( s_i \).

\[
\Pi(s_i, s_k) = \pi(s_i, s_k) - \pi(s_k, s_i)
\] (5.5)

for which \(-1 \leq \Pi(s_i, s_k) \leq 1\)

From the definition it is clear that \( \Pi(s_i, s_k) = -\Pi(s_k, s_i) \)

The results of pairwise comparisons between the solutions in our numerical example (shown in Figure 5.10) are pointed out in Tables 5.1 and 5.2. Table 5.1 shows the \( \pi \) values and Table 5.2 represents the \( \Pi \) values.

Definition 2 (Preference Relations): Based on the value of \( \Pi(s_i, s_k) \), we define three different relations between \( s_i \) and \( s_k \): positive Preference \( (P^+) \), negative Preference \( (P^-) \) and Indifference \( (I) \):

\[
Preference \ Structure = \begin{cases} 
  i f \ & \Pi(s_i, s_k) > 0 \Rightarrow s_i P^+ s_k \\
  i f \ & \Pi(s_i, s_k) < 0 \Rightarrow s_i P^- s_k \\
  i f \ & \Pi(s_i, s_k) = 0 \Rightarrow s_i Is_k 
\end{cases}
\]
Table 5.3: 3-tuple representations of the solutions in our numerical example

<table>
<thead>
<tr>
<th></th>
<th>S_1</th>
<th>S_2</th>
<th>S_3</th>
<th>S_4</th>
<th>S_5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$P^*(s_i) = {x, x_1, x_4, x_3}$</td>
<td>$P^*(s_i) = {x, x_1, x_3}$</td>
<td>$P^*(s_i) = {x_1}$</td>
<td>$P^*(s_i) = {x_1}$</td>
<td>$P^*(s_i) = {}$</td>
</tr>
<tr>
<td></td>
<td>$P^*(s_i) = {}$</td>
<td>$P^*(s_i) = {x_1}$</td>
<td>$P^*(s_i) = {x_1, x_3}$</td>
<td>$P^*(s_i) = {x_1, x_3}$</td>
<td>$P^*(s_i) = {x_1, x_3}$</td>
</tr>
<tr>
<td></td>
<td>$I(s_i) = {x_1}$</td>
<td>$I(s_i) = {x_1}$</td>
<td>$I(s_i) = {x_1}$</td>
<td>$I(s_i) = {x_1}$</td>
<td>$I(s_i) = {x_1}$</td>
</tr>
</tbody>
</table>

**Definition 3 (3-tuple representation):** Once the preference relations are elicited, any solution can be characterized as a 3-tuple $<P^+, P^-, I>$, where:

$$
\lambda^1(s_i) = P^+(s_i) = \{s_k \in S \mid s_i P^+ s_k\}
$$

$$
\lambda^2(s_i) = P^-(s_i) = \{s_k \in S \mid s_i P^- s_k\}
$$

$$
\lambda^3(s_i) = I(s_i) = \{s_k \in S \mid s_i Is_k\}
$$

The 3-tuple representations of the solutions in our example are shown in Table 5.3.

**Definition 4 (Intersection Set):** We define the set $\Lambda^\theta(s_i, s_k)$ such that $\Lambda^\theta(s_i, s_k) = \lambda^\theta(s_i) \cap \lambda^\theta(s_k)$. So, $\Lambda^\theta(s_i, s_k)$ contains the common solutions in the $g^{th}$ element of the 3-tuple of $s_i$ and $s_k$. For instance, in our example, the intersection sets for two solutions $s_2$ and $s_4$ are as follows:

$$
\Lambda^1(s_2, s_4) = \lambda^1(s_2) \cap \lambda^1(s_4) = P^+(s_2) \cap P^+(s_4) = \{s_3\}
$$

$$
\Lambda^2(s_2, s_4) = \lambda^2(s_2) \cap \lambda^2(s_4) = P^-(s_2) \cap P^-(s_4) = \{s_1\}
$$

$$
\Lambda^3(s_2, s_4) = \lambda^3(s_2) \cap \lambda^3(s_4) = I(s_2) \cap I(s_4) = \{\}
$$

**Definition 5 (Similarity Measure):** The basic idea for defining the similarity of two solutions is that two solutions are as similar as their 3-tuple and net preferences are alike. This means that, all solutions inside the same cluster are similar in the sense that they are preferred ($P^+$), not preferred ($P^-$) and indifferent ($I$) to more or less the same solutions and with the same degree ($\Pi$). The measure of similarity between two solutions $s_i$ and $s_k$ is given by:

$$
Sim(s_i, s_k) = \frac{\sum_{g=1}^{3} \sum_{t=1}^{\|\Lambda^\theta(s_i, s_k)\|} (1 - abs(\Pi(s_i, s_t) - \Pi(s_k, s_t)))}{\forall s_t \in \Lambda^\theta(s_i, s_k)}
$$

(5.6)
Where \( n \) is the number of solutions in the set \( S = \{s_1, s_2, ..., s_n\} \), \( \text{abs} \) denotes the absolute value and \(||\) indicates the number of elements in a set. By expanding the outer sum in Equation 5.6, the similarity measure can be expressed as:

\[
\text{Sim}(s_i, s_k) = \frac{1}{n} \left[ \sum_{t=1}^{\left| P^+(s_i) \cap P^+(s_k) \right|} (1 - \text{abs}(\Pi(s_i, s_t) - \Pi(s_k, s_t))) \\
+ \sum_{t=1}^{\left| P^-(s_i) \cap P^-(s_k) \right|} (1 - \text{abs}(\Pi(s_i, s_t) - \Pi(s_k, s_t))) \\
+ |I(s_i) \cap I(s_k)| \right]
\]

As we explained before, the value of \( \Pi(s_i, s_t) \) when \( s_i \) is indifferent to \( s_t \) is zero. Thus, in the third sum, the values of both \( \Pi(s_i, s_t) \) and \( \Pi(s_k, s_t) \) is zero. So, the third sum simply counts the number of common solutions in the sets \( I(s_i) \) and \( I(s_k) \).

Our proposed similarity measure has the following properties:

\[
0 \leq \text{Sim}(s_i, s_k) \leq 1 \\
\text{Sim}(s_i, s_i) = 1 \\
\text{Sim}(s_i, s_k) = \text{Sim}(s_k, s_i)
\]

\( \text{Sim}(s_i, s_k) \) ranges between 0 and 1, where 1 means that two solutions are the same and 0 means they are completely different. The corresponding distance measure is \( \text{Dis}(s_i, s_k) = 1 - \text{Sim}(s_i, s_k) \) and we will use \( \text{Dis}(s_i, s_k) \) instead in the clustering procedure.

To clarify the process of measuring the similarity between two solutions, in the following we calculate the similarity between the two solutions \( s_2 \) and \( s_4 \) of our example:

\[
\text{Sim}(s_2, s_4) = \frac{(1 - \text{abs}(\Pi(s_2, s_3) - \Pi(s_4, s_3))) + (1 - \text{abs}(\Pi(s_2, s_1) - \Pi(s_4, s_1)))}{5}
\]

\( \Lambda^3(s_2, s_4) = \{\} \)

\[
= \frac{(1 - \text{abs}(0.3 - 0.15)) + (1 - \text{abs}(-0.05 - (-0.7)))}{5}
\]

\[
= \frac{0.85 + 0.35}{5} = 0.24
\]
5.3. Problem Formulations

Table 5.4: The similarity measures for each pair of solutions in our numerical example

<table>
<thead>
<tr>
<th>Sim</th>
<th>s₁</th>
<th>s₂</th>
<th>s₃</th>
<th>s₄</th>
<th>s₅</th>
</tr>
</thead>
<tbody>
<tr>
<td>s₁</td>
<td>1</td>
<td>0.44</td>
<td>0.16</td>
<td>0.13</td>
<td>0</td>
</tr>
<tr>
<td>s₂</td>
<td>0.44</td>
<td>1</td>
<td>0.22</td>
<td>0.24</td>
<td>0.07</td>
</tr>
<tr>
<td>s₃</td>
<td>0.16</td>
<td>0.22</td>
<td>1</td>
<td>0.36</td>
<td>0.12</td>
</tr>
<tr>
<td>s₄</td>
<td>0.13</td>
<td>0.24</td>
<td>0.36</td>
<td>1</td>
<td>0.76</td>
</tr>
<tr>
<td>s₅</td>
<td>0.07</td>
<td>0.32</td>
<td>0.76</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5.4 shows the measure of similarity between each pair of solutions in our example.

**Definition 6 (Cluster Center):** Some clustering approaches, such as leader-follower [101], k-means [102] and Partitioning Around Medoids (PAM) [103] algorithms, utilize the concept of cluster center for assigning the objects to different clusters. In these approaches, the similarity of a solution to a cluster is measured based on its similarity to the cluster center. Taking into account the characteristics of all objects inside a cluster identifies the center of that cluster. In the following, we explain the way that we define a cluster center. Suppose that $C_r$ is the $r^{th}$ cluster and $α_r$ is its corresponding center. The 3-tuple representation of the cluster center $α_r$ is given by the union of the 3-tuple of solutions belonging to the cluster $C_r$, as follows:

\[
λ^1(α_r) = P^+(α_r) = \bigcup_{s_i ∈ C_r} P^+(s_i)
\]

\[
λ^2(α_r) = P^-(α_r) = \bigcup_{s_i ∈ C_r} P^-(s_i)
\]

\[
λ^3(α_r) = I(α_r) = \bigcup_{s_i ∈ C_r} I(s_i)
\]

Suppose that in our example, the two solutions $s_2$ and $s_3$ are grouped in the same cluster ($C_1$). The 3-tuple of their cluster center ($α_1$) is:

\[
λ^1(α_1) = P^+(α_1) = \{s_3, s_4, s_5\}
\]

\[
λ^2(α_1) = P^-(α_1) = \{s_1, s_2, s_4\}
\]

\[
λ^3(α_1) = I(α_1) = \{s_2, s_3\}
\]

Note that in the 3-tuple of a solution, there is no common solution between different elements of its 3-tuple (e.g. there is no solution that belongs to both $P^+$ and $P^-$). On the other hand, for each solution, the intersection between the sets $P^+$, $P^-$ and $I$ is empty. However, there might be some common solutions between different elements of the 3-tuple of a cluster center, such as $s_4$ in the above example. $s_4$ is a member of
5.3. Problem Formulations

both $P^+(\alpha_1)$ and $P^-(\alpha_1)$. Thus, for each solution $s_i$ inside a cluster, two different net preferences of cluster center $\alpha_r$ over solution $s_i$ is defined: the positive net preference $\Pi^+(\alpha_r, s_i)$ for those solutions belong to $P^+(\alpha_r)$, and the negative net preference $\Pi^-(\alpha_r, s_i)$, for those solutions belong to $P^-(\alpha_r)$.

The positive net preference of cluster center $\alpha_r$ over solution $s_i$ is the average of the net preferences of all solutions ($s_k$) in the cluster $C_r$ such that $s_k P^+ s_i$, formally:

$$\forall s_i \in \lambda^1(\alpha_r) \quad \Pi^1(\alpha_r, s_i) = \frac{\sum s_k \in C_r \mid s_k P^+ s_i \Pi(s_k, s_i)}{|s_k \in C_r \mid s_k P^+ s_i|}$$

for which $0 < \Pi^+(\alpha_r, s_i) \leq 1$

In the same way, the negative net preference is computed, formally:

$$\forall s_i \in \lambda^2(\alpha_r) \quad \Pi^2(\alpha_r, s_i) = \frac{\sum s_k \in C_r \mid s_k P^- s_i \Pi(s_k, s_i)}{|s_k \in C_r \mid s_k P^- s_i|}$$

for which $-1 \leq \Pi^-(\alpha_r, s_i) < 0$

For the solutions that belong to $I(\alpha_r)$, the net preference is zero:

$$\forall s_i \in \lambda^3(\alpha_r) \quad \Pi^3(\alpha_r, s_i) = 0$$

In our example, the net preferences of cluster center $\alpha_1$ over different solutions are:

$$\Pi^+(\alpha_1, s_5) = \frac{\Pi(s_2, s_5) + \Pi(s_3, s_5)}{2} = \frac{0.5 + 0.05}{2} = 0.275$$

$$\Pi^+(\alpha_1, s_3) = \Pi(s_2, s_3) = 0.3$$

$$\Pi^+(\alpha_1, s_4) = \Pi(s_2, s_4) = 0.3$$

$$\Pi^- (\alpha_1, s_1) = \frac{\Pi(s_2, s_1) + \Pi(s_3, s_1)}{2} = \frac{-0.05 - 0.5}{2} = -0.275$$

$$\Pi^- (\alpha_1, s_2) = \Pi(s_3, s_2) = -0.3$$

$$\Pi^+(\alpha_1, s_4) = \Pi(s_3, s_4) = -0.15$$

The measure of similarity between a solution and a cluster center is the same as measuring similarity between two solutions except that the notation of $\Pi^+$ and $\Pi^-$ should be considered. Equation [5.7] shows the calculation of the similarity between the solu-
5.3. Problem Formulations

5.3.1 Problem Formulations

The similarity of solution \( s_i \) and cluster center \( \alpha_r \).

\[
Sim(\alpha_r, s_i) = \sum_{g=1}^{3} \sum_{t=1}^{\left| \Lambda^g(\alpha_r, s_i) \right|} (1 - \text{abs}(\Pi^g(\alpha_r, s_t) - \Pi(s_i, s_t)))
\]

\[
Sim(\alpha_r, s_i) = \frac{\sum_{g=1}^{3} \sum_{t=1}^{\left| \Lambda^g(\alpha_r, s_i) \right|} (1 - \text{abs}(\Pi^g(\alpha_r, s_t) - \Pi(s_i, s_t)))}{n} \quad (5.7)
\]

Note that for the solutions in the set \( \Lambda^3(\alpha_r, s_i) = I(\alpha_r) \cap I(s_i) \) the value of both \( \Pi^3(\alpha_r, s_i) \) and \( \Pi(s_i, s_i) \) is zero. Thus, the third outer sum (i.e. \( g = 3 \)) in Equation 5.7 simply counts the number of common solutions in the sets \( I(\alpha_r) \) and \( I(s_i) \).

For instance, in our example, the similarity of \( s_4 \) to the cluster center \( \alpha_1 \) is:

\[
Sim(\alpha_1, s_4) = \frac{(1 - \text{abs}(\Pi^+(\alpha_1, s_3) - \Pi(s_4, s_3)))}{5} \\
+ \frac{(1 - \text{abs}(\Pi^-(\alpha_1, s_1) - \Pi(s_4, s_1)) + (1 - \text{abs}(\Pi^-(\alpha_1, s_2) - \Pi(s_4, s_2)))}{5} \\
+ \frac{\left| \Lambda^3(\alpha_1, s_4) \right| = \{\}}{5} \\
= \frac{0.85 + 0.575 + 1}{5} = 0.485
\]

The originality of our proposed similarity measure comes from the definition of the similarity that takes into account the preference structure defined by the decision maker. So, two solutions are considered as similar solutions in the sense that they are preferred (\( P^+ \)), not preferred (\( P^- \)) and indifferent (\( I \)) to more or less the same solutions and with the same degree (\( \Pi \)).

For better understanding the similarity relations between solutions, we propose a visual representation of the preference structure. Figure 5.11 illustrates the visual form of the preference structures for 5 solutions in our example. For each solution, a circular bar chart is drawn that shows the preference relations between the corresponding solution and all the other solutions. The name written in the center of the bar chart indicates its corresponding solution. In each bar plot, the number of bars is equal to the total number of solutions (\( n \)) (one bar per solution). Solutions from \( s_1 \) to \( s_n \) are placed clockwise in the circular bar chart. For each solution \( s_i \), the length of the colored portion of the bar corresponding to \( s_k \) is equal to value of \( \Pi(s_i, s_k) \). If \( \Pi(s_i, s_k) \) is greater than zero (means that \( s_i P^+ s_k \)) a blue color is used. When \( \Pi(s_i, s_k) \) is less than zero (means that \( s_i P^- s_k \)) a red color is used. In the case that \( s_i I s_k \), the length of the colored part is zero, since \( \Pi(s_i, s_k) = 0 \). By using the visual representation of
5.3. Problem Formulations

Preference structures, it is easy to find out the similarity relations between solutions without doing the respective mathematical calculations. Two solutions are as similar as their visual form is alike. For instance, we can see in Figure 5.11 that the bar charts of $s_4$ and $s_5$ are relatively similar to each other. Thus, we can conclude that their similarity is relatively high. This conclusion can be verified by looking at Table 5.4, which shows the computed similarity values for each pair of solutions. As can be seen in this table, the similarity measure between $s_4$ and $s_5$ is 0.76, which is fairly high (Note that the similarity measure is in the range [0,1]). From Figure 5.11 we can also see that the two solutions $s_1$ and $s_5$ have completely different visual forms and thus their similarity should be too small. As can be seen in Table 5.4, the similarity of $s_1$ and $s_5$ is zero.

Clustering Methods

The clustering methods can be broadly classified into two types: exclusive and hierarchical. In exclusive clustering, data are grouped into a set of disjoint clusters. Therefore, if an object is a member of a certain cluster then it cannot be included in another cluster. Instead, the hierarchical clustering is based on the union between the two nearest clusters. At the beginning, each object is in a separate cluster and then pairs of clusters are successively merged until all clusters have been merged into a single cluster that contains all objects.
5.3. Problem Formulations

Algorithm 5.1 Basic Leader-Follower Clustering

Notations: \( C_r \) = the \( r^{th} \) cluster, \( \alpha_r \) = the center of the \( r^{th} \) cluster, \( s_i \) = the \( i^{th} \) solution, and \( N \) = number of clusters

Input: Distance threshold (\( \theta \)) and a set of solutions (\( S \))

Output: A set of \( N \) clusters (Cluster assignment for each solution)

Begin
\[
C_1 = \{ s_1 \}, \quad N = 1 \quad // \text{Assign the first solution to the first cluster}
\]
\[
i = 2
\]
Do accept new solution \( s_i \)
\[
r = \arg\min_{r' \in \{1,2,...,N\}} (\text{Dis}(\alpha_{r'}, s_i)) \quad // \text{Find the nearest cluster center}
\]
If \( \text{Dis}(\alpha_r, s_i) < \theta \)
Then
\[
C_r = C_r + \{ s_i \} \quad // \text{Assign} \ s_i \text{ to the closet cluster}
\]
Update \( \alpha_r \)
Else
\[
C_{N+1} = \{ s_i \} \quad // \text{Create new cluster that contains} \ s_i \text{ as its first member}
\]
\[
N = N + 1
\]
\[
i = i + 1
\]
Until no more solutions
End

VMODEX allows a decision maker to cluster Pareto optimal solutions with both exclusive and hierarchical methods. The similarity measure between the Pareto optimal points is our proposed preference similarity measure described in the previous section. However, in the clustering procedure, instead of the similarity measure, the corresponding distance measure \( \text{Dis}(s_i, s_k) = 1 - \text{Sim}(s_i, s_k) \) is used.

Exclusive Clustering In this thesis, we use Leader-Follower approach for grouping the solutions in an exclusive way. The Leader-Follower algorithm [101] is one of the most popular exclusive methods and is simple and efficient for clustering data sets. The basic Leader-Follower clustering procedure is described in Algorithm 5.1. The decision maker must specify a threshold (\( \theta \)) for the distance. The first solution is assigned to the first cluster. From then, for each solution, the nearest cluster center is found. If the distance is less than the threshold, the solution becomes a member of that cluster and the cluster center should be updated. Otherwise, if the distance exceeds the threshold \( \theta \), it means that no current cluster is sufficiently close to the solution and a new cluster is created having the current solution as its first member. Note that in
5.3. Problem Formulations

Algorithm 5.1 $\arg\min_x f(x)$ denotes the set of values of $x$ for which $f(x)$ attains its smallest value.

The Distance threshold should be chosen carefully, since it implicitly specifies the number of clusters obtained at the end. A large threshold leads to a small number of large clusters, which contain solutions with less similarity. While with a small threshold the probability of creation of new clusters is higher and therefore it will produce a large number of small clusters, which is somewhat contradictory with the aim of clustering.

The drawback of the Leader-Follower clustering algorithm is that it is not deterministic. The initial order of the solutions may strongly influence the structure of clusters. To overcome this weakness, we sort the solutions by their total net preferences before starting the clustering process. The total net preference of solution $s_i$ is calculated as:

$$\Pi_{total}(s_i) = \sum_{k=1}^{n} \Pi(s_i, s_k) \quad (5.8)$$

Where $n$ is the number of solutions in the set. The sorting is performed in descending order. This means that the solution with the highest total net preference value is used as the first solution in the clustering procedure and the solution with the smallest $\Pi_{total}$ value is taken as the last solution. For assigning a solution to a cluster, we use a deterministic way of choosing between clusters with an equal distance from the solution in question. If the solution that is in process of assigning a cluster has the same nearest distance to two or more cluster centers, it becomes a member of the cluster that contains the closest solution to the current solution in the ordered sequence of the solutions. For instance, consider the set $\{A, B, C, D, E, F\}$ as an ordered set of solutions. Suppose that $A$ and $D$ are in the same cluster ($C_1$) and $B$, $C$ and $E$ are in the other cluster ($C_2$). Assume that $F$ has the same distance from the centers of both clusters. By applying our deterministic approach, it will be assigned to cluster $C_2$ since this cluster contains $E$ which is closest solution to $F$ in the ordered set.

In each cluster, the solution with the highest $\Pi_{total}$ value is chosen as a representative solution. A representative solution can be considered as the most promising solution among the solutions in its cluster. Therefore, by applying the clustering, the Pareto optimal set can be reduced to a smaller subset of representative solutions.

For each cluster center, we define its objective values to be the average of all points in the cluster. For instance, if the $r^{th}$ cluster contains two solutions $s_1 = (4, 3, 5)$ and $s_2 = (4, 1, 7)$, then the objective values of cluster center $\alpha_r$ is calculated as follows:

$$\alpha_r = \left(\frac{4 + 4}{2}, \frac{3 + 1}{2}, \frac{5 + 7}{2}\right) = (4, 2, 6)$$

To understand the similarity relation between clusters, we define a procedure to connect the clusters based on their similarity. First, a fully connected weighted graph $G$ is created in such a way that each cluster is considered as a node in the graph and the edge weight between two nodes is the distance between the corresponding cluster
5.3. Problem Formulations

centers. Then a Minimum Spanning Tree (MST) is created for the graph $G$ (A detailed description of MST is given in Chapter 4, Section 4.3.1). The clusters are connected together based on their connections in the corresponding MST. Thus, we get a tree of clusters with minimum weight (most similar connections).

We propose a visualization method that shows how the solutions are grouped in different clusters. This visualization allows decision makers to quickly and effectively analyze the result of clustering. Figure 5.12 shows the visualization of Leader-Follower clustering for the Pareto optimal solutions in our case study. Each cluster is shown by a different color. In Figure 5.12 the distance threshold is 0.3, which leads to the six clusters. Each solution is connected to the center of the cluster to which it belongs. The length of the connected link indicates the distance from the center. A shorter edge means closer to the center. In a cluster with only one solution, a dashed line is used for connecting the solution to the cluster center since the distance is zero (like $C_1$ in Figure 5.12). The cluster $C_4$ is the biggest cluster containing 5 solutions. In each cluster, the representative solution is highlighted by a yellow border such as solution $P_{11}$ in cluster $C_4$. In Figure 5.12, the way that the cluster centers are connected together is the same as their connections in the corresponding MST. The length of the edge between two cluster centers represents the distance between them. The longer the edge implies a larger distance. The distance value between two connecting clusters is written above the corresponding edge. In Figure 5.12, clusters $C_3$, $C_4$, $C_5$ and $C_6$ are relatively close (similar) to each other and contain more than 75% (11 out of 14) of discovered Pareto optimal points.

Hierarchical Clustering  The Leader-Follower clustering algorithm is efficient and conceptually simple, but it requires a specific distance threshold as input, which has a major effect on the clustering results. While the hierarchical clustering does not require any predetermined parameter and is deterministic. However, the complexity
5.3. Problem Formulations

Algorithm 5.2 Basic Agglomerative Hierarchical Clustering

**Input:** a set of \( n \) solutions  
**Output:** A hierarchical cluster tree

**Begin**

Start with \( n \) clusters, each containing a single solution  
Calculate the \( n \times n \) similarity matrix between all pairs of clusters  
**Do** accept new solution \( s_i \)

- Find the two clusters \( C_i \) and \( C_j \) such that their similarity is maximum (two closest clusters)  
- Merge these clusters into a single cluster \( C_r = C_i \cup C_j \)  
- Remove the two old clusters \( C_i \) and \( C_j \)  
- Determine the similarities between the new cluster \( C_r \) and the remaining clusters

**Until** only one cluster is left, which contains all the solutions

**End**

of hierarchical clustering increases at least quadratic with the number of data points. Several hierarchical clustering algorithms have been proposed such as SLINK [104], CURE [105] and CHAMELEON [106]. In this thesis, we use Agglomerative hierarchical clustering. The basic steps of Agglomerative hierarchical clustering are described in Algorithm 5.2.

The similarity of two clusters can be measured in three ways: single-linkage, complete-linkage and average-linkage. In the single-linkage clustering, the shortest distance from any member of one cluster to any member of another cluster is considered as the similarity between two clusters. In the complete-linkage method, the greatest distance between any two solutions in the different clusters determines the distance between the two clusters. In the average-linkage clustering, the distance between two clusters is computed as the average of distances between all pairs of solutions, in which one solution is in the first cluster and another solution is in the second cluster. Table 5.5 summarizes the definition of different linkage methods between two clusters \( A \) and \( B \).

A hierarchical clustering is typically visualized as a binary tree (called dendrogram), which represents the hierarchical, pair-wise clustering of the items in the data set. Figure 5.13 shows the dendrogram obtained from the hierarchical clustering (single-linkage) of the Pareto optimal solutions in our case study. The individual solutions are placed at the bottom of the dendrogram and construct the leaf nodes. Each merge is represented by an internal node that connects the two joining clusters. The \( y \)-coordinate of the internal node shows the distance between two merging clusters. Each
internal node has exactly two subtrees of clustered solutions (a right and a left subtree). Moving from the bottom to the top of the dendrogram shows the history of merges during the clustering process. For example, we can see that in Figure 5.13 the two solutions $P_9$ and $P_{10}$ are merged first, and at the last merge the solution $P_1$ is added to a cluster consisting of all the other solutions. To understand in each merging step, which solutions are considered for the similarity among the two joining clusters, the name of the two closest/furthest solutions are written at the bottom of the corresponding internal node (this is applicable for only the single and complete linkage strategies).

Table 5.5: Definition of different linkage methods

<table>
<thead>
<tr>
<th>Linkage Method</th>
<th>Definition</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Linkage</td>
<td><img src="image" alt="Diagram" /> $\min{\text{Dis}(a,b) : a \in A, b \in B}$</td>
<td></td>
</tr>
<tr>
<td>Complete Linkage</td>
<td><img src="image" alt="Diagram" /> $\max{\text{Dis}(a,b) : a \in A, b \in B}$</td>
<td></td>
</tr>
<tr>
<td>Average Linkage</td>
<td><img src="image" alt="Diagram" /> $\frac{1}{</td>
<td>A</td>
</tr>
</tbody>
</table>
Once the complete hierarchical dendrogram is constructed, depending on the type of decision, different cutting policies can be applied on it, as follows:

- Cutting at a specific level of dissimilarity. In this case we get clusters of which the distance between the solutions inside the same cluster is less than the determined dissimilarity threshold. For example, in Figure 5.13, cutting the dendrogram at 0.35 yields 3 clusters: one large cluster containing 11 solutions (black lines), one cluster consisting of $P_2$ and $P_5$ (blue lines), and one cluster including only $P_1$ (red line). Each cluster contains solutions of which the dissimilarity (distance) between them is less than 0.35.

- Cutting in such a way that a certain number of clusters remain. If the decision maker prefers to organize solutions in $k$ clusters, he just should cut the $k-1$ links from the top. Therefore, the hierarchical clustering is flexible in the sense that the decision maker can easily change the number of clusters he is interested in, without repeating the clustering process. In Figure 5.13, cutting the dendrogram for getting 3 clusters produces the same clusters as cutting at level 0.35.

- Cutting the dendrogram at the point where the gap between two levels of dissimilarity, before and after merging a cluster to another one, is largest (the highest vertical line in the dendrogram). Such a large gap indicates that merging one more cluster significantly decreases the quality of the clustering. Therefore, cutting before this deterioration is desirable. In Figure 5.13, the greatest gap occurs at the last merging (red line), when the solution $P_1$ is added to a cluster consisting of all the other solutions.

5.3.4 Ranking Problem

In the ranking approach, the decision problem is to order all the solutions in terms of their overall preference value, which is derived by taking into account all objectives. Several methods have been proposed for addressing the ranking problem. Here, we explain the three most commonly used ranking methods that are provided in VMODEX. The main differences between these ranking methods are 1) the normalization process for comparing all criteria on a common scale, and 2) the way that they aggregate the normalized objective values and their weights for obtaining an overall preference value for each solution.

For describing the ranking methods, we consider a MODM problem with a set of $n$ solutions $s_i (i = 1, 2, ..., n)$ and $m$ objective functions $f_j (j = 1, 2, ..., m)$. The performance of solution $s_i$ on the objective function $f_j$ is denoted by $f_j(s_i)$. We assume that all objective functions need to be minimized, that is, the smaller the value, the better it is. The relative importance of objective functions is indicated by the weight assigned to each objective. Suppose that $w_j$ denotes the weight of the objective function $f_j$. 
5.3. Problem Formulations

Weighted Sum Model (WSM)

The WSM method is the best known and simplest ranking method. However, it is applicable only when the decision criteria are expressed in identical units of measure. If criteria are not in the same unit, then normalization is required to provide a common scale for comparisons of the objectives. The normalized value of \( f_j(s_i) \) can be obtained by:

\[
\bar{f}_j(s_i) = \frac{f_j(s_i) - \min_{k=1}^{n} f_j(s_k)}{\max_{k=1}^{n} f_j(s_k) - \min_{k=1}^{n} f_j(s_k)}
\]

Where \( \bar{f}_j(s_i) \) indicates the normalized performance of solution \( s_i \) on objective function \( f_j \). This normalization approach linearly transforms all objective values in the range \([0,1]\). Thus, it preserves the relative order of magnitude of the original objective values.

The overall preference value of each solution \( s_i \) is calculated using the weighted sum of its normalized values over all objectives, as follows:

\[
PV(s_i) = \sum_{j=1}^{m} w_j \bar{f}_j(s_i)
\]  

(5.9)

Since we consider the minimization problem, a smaller value of \( PV \) indicates the more preferred solution. For ranking the Pareto optimal solutions, they are sorted by their preference values \( (PV) \) in ascending order.

TOPSIS

TOPSIS (Technique for Order Preference by Similarity to Ideal solution) \[107\] is based on the concept that the most preferred solution should have the shortest distance from the positive ideal solution \( IS^+ \) and the longest distance from the negative ideal solution \( IS^- \). TOPSIS ranks the solutions according to these two distances. The procedure of TOPSIS method can be summarized as follows:

1. The normalized objective values are computed by:

\[
\bar{f}_j(s_i) = \frac{f_j(s_i)}{\sqrt{\sum_{k=1}^{n} f_k(s_k)^2}}
\]

2. \( IS^+ \) and \( IS^- \) are obtained from the weighted normalized objective values. \( IS^+ \) is found by constructing a vector of best values and \( IS^- \) is calculated by con-
5.3. Problem Formulations

Structuring a vector of worst values as follows:

\[ R_j(s_i) = w_j \bar{f}_j(s_i) \]

\[ IS^+ = (R_1^+, ..., R_j^+, ..., R_m^+) \quad R_j^+ = \min_{k=1}^{n} R_j(s_k) \]

\[ IS^- = (R_1^-, ..., R_j^-, ..., R_m^-) \quad R_j^- = \max_{k=1}^{n} R_j(s_k) \]

3. For each solution \( s_i \), the Euclidian distance from the positive ideal solution (\( ED^+ \)) and negative ideal solution (\( ED^- \)) is computed as follows:

\[ ED^+(s_i) = \sqrt{\sum_{j=1}^{m} (R_j(s_i) - R_j^+)^2} \]

\[ ED^-(s_i) = \sqrt{\sum_{j=1}^{m} (R_j(s_i) - R_j^-)^2} \]

4. The overall preference value for each solution is calculated as follows:

\[ PV(s_i) = \frac{ED^-(s_i)}{ED^-(s_i) + ED^+(s_i)} \quad 0 \leq PV(s_i) \leq 1 \quad (5.10) \]

\( PV(s_i) \) indicates the relative closeness of the \( i^{th} \) solution to the positive ideal solution. Therefore, a greater value means that the solution is closer to \( IS^+ \) and thus is more preferred.

5. For ranking, solutions are sorted in descending order of their \( PV \) values.

To compare different solutions in a Pareto optimal set in terms of their distances from \( IS^+ \) and \( IS^- \) and their \( PV \) values, we visualize this information such as shown in Figure 5.14. The length of the blue bar represents the distance between the corresponding solution and \( IS^+ \) (\( ED^+ \)). A longer bar indicates that the solution is further away from \( IS^+ \). Therefore, a smaller blue bar is desirable. Similarly, the length of the red bar shows the distance from \( IS^- \) (\( ED^- \)). Thus, a longer red bar is more favorable. The length of the green bar indicates the relative closeness to \( IS^+ \) (\( PV \) value). The longer the bar denotes the solution is closer to the \( IS^+ \) and thus is more preferred. In Figure 5.14, the solutions (from left to right) are sorted by their \( PV \) values in descending order. Therefore, the position of each solution is the same as its rank in the TOPSIS method. For each solution, besides the distance values (\( ED^+ \) and \( ED^- \)), the amount of difference between their objective values can be seen (color coding). Therefore, it is easy to find out which objective value(s) have a high impact on the distance values. For example, \( P2 \) has the same energy consumption as the \( IS^+ \) (the color and size of the third dimension of their representing nodes are the same) and the difference between their costs is small. However, they are significantly different in the processing time (node color) and thus this objective has the highest impact on the distance value. In Figure 5.14, except \( P12 \) and \( P3 \), all the other solutions are closer to \( IS^+ \) than \( IS^- \) (\( ED^+ < ED^- \)) since their blue bars are shorter than their red bars.
PROMETHEE

PROMETHEE (Preference Ranking Organization METHod for Enrichment Evaluations) \([94, 108]\) is based on the pairwise comparisons of solutions for each objective. For each solution, it calculates positive and negative preference flows. The positive flow expresses how much a solution is dominating the other solutions and the negative flow indicates how much it is dominated by the other ones. Based on the balance of these two preference flows, the PROMETHEE ranks the solutions. PROMETHEE uses preference functions (see Section 5.2.3) to compute the degree of preference associated to the best solution in the case of pairwise comparisons. For each objective, a specific preference function must be defined. The procedure of PROMETHEE methodology can be summarized as follows:

1. Solutions are compared pairwise for each objective. Afterwards, preference functions are used to map their deviations on the objectives to the range \([0,1]\). For the objective function \(f_j\), its corresponding preference function \(PF_j\) is used. The degree of preference of solution \(s_i\) over \(s_k\) on the \(j^{th}\) objective is given by:

\[
P_j(s_i, s_k) = PF_j(f_j(s_i) - f_j(s_k)) \quad 0 \leq P_j(s_i, s_k) \leq 1
\]

\(P_j(s_i, s_k) = 0\) means no preference and \(P_j(s_i, s_k) = 1\) indicates the strict preference of \(s_i\) over \(s_k\). For small deviations, a small preference is allocated to the better solution and even possibly no preference if the deviation is negligible. The larger the deviation, the larger the preference.

2. The overall preference index of solution \(s_i\) over \(s_k\) for all objectives is computed
as follows:

\[ \pi(s_i, s_k) = \sum_{j=1}^{m} w_j P_j(s_i, s_k) \]

3. For each solution \( s_i \), the positive flow (\( \phi^+ \)) and the negative flow (\( \phi^- \)) are defined as:

\[ \phi^+(s_i) = \frac{1}{n-1} \sum_{k=1, k \neq i}^{n} \pi(s_i, s_k) \quad \phi^-(s_i) = \frac{1}{n-1} \sum_{k=1, k \neq i}^{n} \pi(s_k, s_i) \]

4. For each solution \( s_i \), the net flow \( \phi(s_i) \) is considered as:

\[ \phi(s_i) = \phi^+(s_i) - \phi^-(s_i) \quad -1 \leq \phi(s_i) \leq 1 \quad (5.11) \]

The net flow is the balance between the positive and negative flows. The higher the net flow means the better the solution. \( \phi(s_i) > 0 \) means that the solution is more dominating all the other solutions on all objectives and \( \phi(s_i) < 0 \) indicates it is more dominated.

5. The complete ranking of all solutions from the best to the worst is obtained by sorting their \( \phi \) values in descending order.

\( \phi_j(s_i) \) is the single objective net flow for solution \( s_i \) when only the \( j^{th} \) objective is considered (100% of the total weight is assigned to objective \( j \)). It expresses how a solution \( s_i \) is dominating \( \phi_j(s_i) > 0 \) or dominated \( \phi_j(s_i) < 0 \) by all the other solutions on the \( j^{th} \) objective. \( \phi_j(s_i) \) is defined as follows:

\[ \phi_j(s_i) = \frac{1}{n-1} \sum_{k=1, k \neq i}^{n} [P_j(s_i, s_k) - P_j(s_k, s_i)] \quad (5.12) \]

\( \phi_j(s_i) \) is particularly useful for analyzing the "quality" of solutions for each specific objective separately.

For comparing the Pareto optimal solutions based on their amount of domination on each objective, we visualize the \( \phi_j(s_i) \) values such as shown in Figure 5.15 (the data are captured from our case study results explained in Section 5.1). In this figure, each solution is shown as an axis. The axes start from the same point, have the same length and are equi angular. Since the \( \phi_j(s_i) \) values are in the range \([-1, 1]\), the center of the chart indicates \(-1\) and the endpoints of axes denote \(1\). For each objective, a radar plot (also known as kiviat diagram) is drawn with a specific color. The points constructing a radar plot are the corresponding \( \phi_j(s_i) \) values. Because the values greater than zero (means more dominating) or less than zero (means more dominated) have completely opposite meaning, the zero axis is highlighted. In this figure, the order of solutions (clockwise) is the same as their position in the PROMETHEE ranking method, which
5.3. Problem Formulations

is based on the sorted $\phi_{Total}$ values in descending order. In Figure 5.15, the filled radar plot shows the total $\phi$ values (considering all objectives). As can be seen in Figure 5.15, the net flow of the "cost" objective for the solution $P1$, $\phi_{Cost}(P1)$, is 1, which means it dominates all the other solutions in terms of the cost. On the other hand, the $\phi_{PT}(P1)$ is $-1$ which indicates that $P1$ is dominated by all the other solutions with respect to the processing time ($PT$). Its net flow for energy consumption ($EC$) is less than zero ($\phi_{EC}(P1) < 0$), which denotes that $P1$ is more dominated by all the other solutions in terms of energy consumption. As a result, its total net flow $\phi_{Total}(P1)$, which considers all the objectives is small and it is on the three worst solutions according to the PROMATHEE ranking method.

Showing ranking results

The outcomes of different ranking methods (described above) may not be the same since they use different procedures for ordering the Pareto optimal solutions. For comparing the results of different ranking methods and investigating the rank of a specific solution with respect to the different ranking methods, we propose a visualization approach as shown in Figure 5.16. This figure represents the results of ranking the Pareto optimal solutions obtained from our case study by applying three ranking methods: WSM, TOPSIS and PROMETHEE. Each ranking method is shown with a specific color. Assume there are $n$ solutions in the Pareto optimal set. The circle is divided to $n$ equal parts, where each part represents one solution. The number of nested circles is also $n$; each one denotes a rank position. The innermost one indicates the worst rank and the outermost one shows the best rank. For example, in Figure 5.16, the solutions $P2$ and $P5$ are in the top two for all ranking methods. Therefore, these
solutions can be a good choice for the final decision. However, the solution $P3$ is ranked as the worst solution by all the ranking methods. So, this solution is probably not appropriate as a final decision. Furthermore, from this figure we can see that all the solutions have almost the same rank in different ranking methods.

5.4 Conclusion

In this chapter, we discussed the final step in solving the multi-objective optimization problems, which is selecting the most preferred solution from the set of Pareto optimal solutions. After finding the Pareto optimal set by an optimization algorithm, we face another challenge: which solution from this set should be implemented. To solve this latter problem, Multi-Objective Decision Making (MODM) methods are used. These methods systematically apply the decision maker’s preference information and provide some guidelines for choosing the most satisfying solution.

We described the four basic problem formulations in MODM, which are: choice, classification/sorting, clustering and ranking problems. For each problem formulation, some decision making methodologies are explained. We proposed a new method addressing the choice problem. Our method is based on the fuzzy dominance relations between the Pareto optimal solutions. Furthermore, we defined a new preference similarity measure for clustering the solutions. In our proposed similarity measure, unlike the conventional measures, the decision maker’s preferences are integrated in the multi-objective cluster analysis. That is, two solutions are as similar as they are preferred, not preferred and indifferent to more or less the same solutions. Moreover,
we introduced a scheme for constructing the cluster centers, considering the properties of all solutions inside the same cluster. Some clustering approaches utilize the concept of cluster center for assigning the objects to different clusters.

For each MODM method, we proposed a new visualization approach that provides a detailed analysis of the results of MODM techniques. These visual representations allow decision maker to find out how and why a particular solution is considered as a most preferred solution with respect to a specific MODM method.