Techniques and Visualization Approaches for Analyzing Local and Global Pareto Optimal Sets in Multi-Objective Design Space Exploration

Toktam Taghavi, University of Amsterdam, the Netherlands, T.TaghaviRazaviZadeh@uva.nl
Andy D. Pimentel, University of Amsterdam, the Netherlands, A.D.Pimentel@uva.nl

Abstract

VMODEX is an interactive visualization tool to support system-level Design Space Exploration (DSE) of MPSoC architectures. It provides insight into the search process of Multi-Objective Evolutionary Algorithms (MOEAs) that are typically used in the DSE process, and it facilitates the analysis of the DSE results. In this paper we extend VMODEX to enable designers to evaluate and compare the properties of the discovered design points in different subspaces of the explored design space. Several techniques and visualization methods are provided to compare different parts of the design space from various aspects. Furthermore, some Multi-Objective Decision Making (MODM) methods are utilized to help designers with understanding the trade-offs between different criteria and guide them towards the most appropriate solutions among the Pareto optimal solutions. Moreover, new visualization approaches are proposed, which provide the designer with the visual interpretation and detailed analysis of the results of the MODM methods.

1 Introduction

The complexity of modern embedded systems has led to the emergence of system-level design. A key issue of system-level design is the notion of high-level modeling and simulation in which the models allow for capturing the behavior of system components and their interactions at a high level of abstraction. As these high-level models minimize the modeling effort and are optimized for execution speed, they can be applied at the very early design stages to perform, for example, architectural Design Space Exploration (DSE). Such early design space exploration is of eminent importance as early design choices influence the success or failure of the final product.

System-level simulation frameworks that are deployed for DSE of embedded systems that are based on heterogeneous Multi-Processor System-on-Chip (MPSoC) architectures, usually use independent application and architecture models. The application model describes the functional behavior of the system expressed as processes (computations) and channels (communications). The architecture model represents the hardware components in the system, such as processors, reconfigurable modules, memories, etc. Then, different mappings of processes and communication channels to the various architectural components are evaluated by simulation to find the optimum mapping solutions. Each mapping decision taken in this step corresponds to a single point in the design space.

Generally, for designing complex embedded systems, multiple criteria need to be considered simultaneously such as performance, power, cost, etc. Such design space exploration, during which multiple criteria should be considered simultaneously, is called Multi-Objective Design Space Exploration (MODSE). Since objectives are often in conflict, there cannot be a single optimum solution, which simultaneously optimizes all objectives. Instead, a set of optimal solutions denoted as the Pareto optimal set or non-dominated set has to be found. This is the set of those solutions for which one objective cannot be improved further without causing a simultaneous degradation in at least one other objective. These optimal decisions provide the designer trade-offs between the design objectives. The set of objective vectors corresponding to a set of Pareto optimal solutions is called “Pareto optimal front” or “non-dominated front”.

In order to find a Pareto optimal set with respect to the design criteria, the designer should ideally evaluate and compare every single point in the design space. However, such an exhaustive search is infeasible, as in real-scale problems the design space is too large to be explored in an exhaustive manner. Therefore, heuristic search techniques, such as Multi-Objective Evolutionary Algorithms (MOEAs), are often used to search the design space for optimum design points using only a finite number of design-point evaluations. MOEAs evaluate a population of design points (solutions) over several iterations, called generations. With the help of genetic operators, a MOEA progresses iteratively towards the best possible solutions.

As the searched design space still is vast, interpreting all evaluation data and understanding how the MOEA searches through or prunes the design space is cumbersome. Therefore, in [1] we proposed a novel interactive visualization tool, VMODEX[1], which is based on tree visualization. This tool shows both design parameters and objectives in a single view. It enables designers to easily understand how a MOEA explores the design space, where the optimum design points are located, how design parameters influence each objective, and find out the relationship between the different objectives. In this paper, we extend VMODEX to enable designers to evaluate and compare the properties of the discovered design points in different subspaces of the explored design space. By modeling the design space as a tree (like in VMODEX), it is divided in several subspaces. Each subspace represents a unique instance of

[1] Visualization of Multi-Objective Design Space eXploration

the architecture platform. On the other hand, solutions inside a subspace have exactly the same architecture components but the way in which the application is mapped onto those components is different. In each subspace, the Pareto optimal solutions found by a MOEA are called *local Pareto optimal solutions*. In the extended version of VMODEX several techniques and visualization methods are provided to compare various local Pareto optimal sets in different parts of the design space from various aspects. In most multi-objective optimization problems, the global Pareto optimal sets obtained by applying different MOEAs are compared to investigate which algorithm is more suitable for a specific problem. Comparing different local Pareto optimal sets (like we explain in this paper) is a new point of view in MODSE process and has not been considered before. Such comparison is, however, essential to the designer as it provides insight into the "landscape" of the design space and can help him to comprehensively understand the properties of the discovered design points in different subspaces of the explored design space (as it is illustrated later on in the paper).

Furthermore, in the extended version of VMODEX, some Multi-Objective Decision Making (MODM) methods are utilized to help designers to find out the trade-offs between different criteria and to aid them to distinguish those solutions that are better than the others in terms of non-dominance. Moreover, new visualization approaches are proposed, which provide the designer the visual interpretation and detailed analysis of the results of the MODM methods.

There are several MODM techniques, which help the decision maker to identify the most preferred solutions in multi objective optimization problems. One approach is based on cluster analysis. A clustering method such as Leader-Follower [2] or hierarchal clustering [3] is used for clustering the solutions in a Pareto optimal set. So, the Pareto optimal front is reduced to a set of k clusters. Each cluster consists of solutions with similar properties, and therefore the decision maker has to investigate only the representative solution of each cluster, which can be the closest solution to the cluster center. Another method is ranking such as TOPSIS [4] and PROMETHEE [5]. They rank the Pareto optimal solutions from the best to the worst with respect to the all objectives. Another approach is based on the selection of a subset of Pareto optimal points that contains the most satisfying solutions, such as filtering the solutions by the decision maker's preferences on each objective.

Usually the results of MODM methods are shown in a table or displayed in a 2D graph. Although these kinds of representations are useful to find out the most preferred solutions in the Pareto optimal set, they do not provide insight on how these results come out. In this paper, we propose several visualization techniques, which visualize the outcomes of MODM methods. They provide decision makers a detailed analysis of relations between Pareto optimal solutions and allow for finding out why one solution is considered a more preferred solution with respect to a specific MODM method.

The rest of the paper is organized as follows. In Section2 we briefly explain how VMODEX visualizes the multi-objective DSE process. Section 3 introduces several metrics and their visualization approaches for comparing design points discovered in different subspaces of the explored design space. Section 4 describes various MODM techniques and their visualization methods for comparing the global Pareto optimal solutions and choosing the most appropriate solutions among them. Finally, Section 6 concludes the paper.

2 Visualizing Multi-Objective DSE by VMODEX

For describing the visualization techniques provided in VMODEX, we use a case study of which the results are used as input data for all visualization methods explained in this paper. 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, a Multi-Objective Evolutionary Algorithm (MOEA) has been used to achieve a set of optimal alternative mapping decisions under the aforementioned criteria.

In this section, we briefly explain how VMODEX [1] visualizes a multi-objective design space exploration. Figure 1 represents the design space of our case study, which is visualized by VMODEX. It should be mentioned that this figure shows only those parts of the design space that are compared with each other in section 3 and not the entire design space. This is because the size of the tree representing the entire design space is large and it is not possible to include it in the paper. In Figure 1, the first four levels represent the design space parameters, which (from top to down) are: number of processors, processor type, number of memories and memory type. The fifth level shows the architecture cost. Since this is the first level for which all the components are known, the cost of the architecture can be computed. Because the cost is an objective and not a design parameter, we represent it with a different shape; a circle. For a better view, the size of the circle becomes bigger as the cost increases. The next level shows the global Pareto optimal solutions found by the MOEA. Each solution represents a unique mapping of the application onto the MPSoC and is a child of its corresponding platform architecture instance. That means, for each solution, its parents at the previous levels show its design parameters and cost. For instance, P4 has the following architectural components: three processors, of which one is an ASIP, one an mC and the other one is mP, and two memories, of which one type is DRAM and

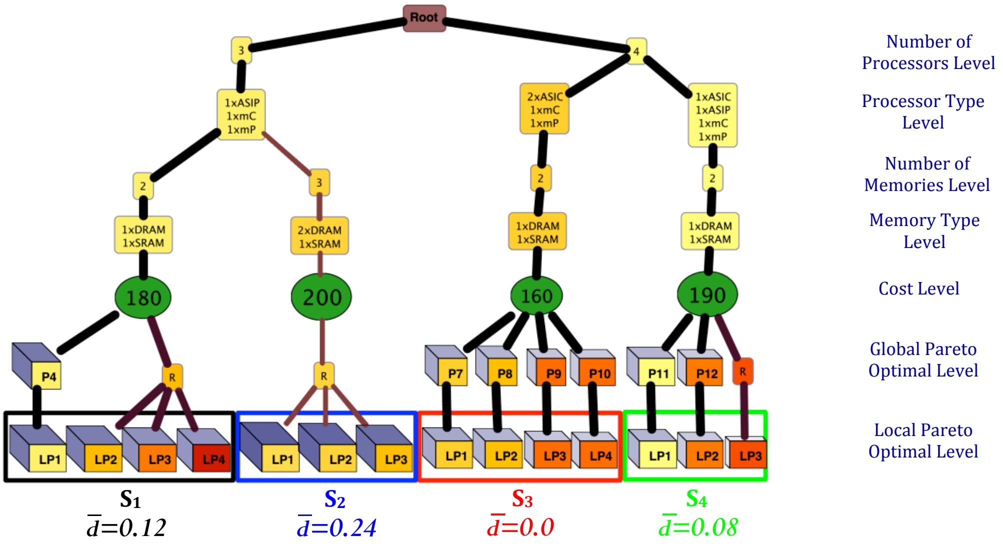


Figure 1. Visualization of multi-objective design space exploration by VMODEX

another type is SRAM. The last level shows the local Pareto optimal solutions. A design point is called a local Pareto point if within the design points with the same architecture (but with different mappings), there is no point dominating that one. So, it is an optimal solution with respect to a specific architectural instance. However, in the entire design space, a design point might exist which dominates the local Pareto point. Local Pareto points, which have the same parent at the cost level (and thus in the higher levels) are in the same subspace and therefore are members of the same local Pareto set. In Figure 1 four local Pareto optimal sets (S_1, S_2, S_3 and S_4) are shown of which their properties are compared from different perspectives in Section 3. It is clear that all the global Pareto points are local Pareto points as well. However, not all the local Pareto points are global Pareto points and therefore we use a relation node at the global Pareto level to make a connection between them and the previous level. These nodes are labeled with "R" in Figure 1. In this paper, we simply refer to the global Pareto optimal design points as Pareto optimal solutions.

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 color of the solution itself represents the processing time. 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.

In order to easily correlate different visualization methods provided in VMODEX with each other, the same metaphors are used for showing the objective values and solutions. Thus, the designer will not be confused by the different representations of the same thing. For instance, in all visualizations, the solution P4 in Figure 1 is shown like in Figure 2. Since in the DSE tree the costs of solutions are shown as separate nodes at the cost level, the corresponding cost node is drawn above the 3D rectangle.

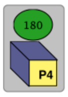


Figure 2. An example of representing a solution

3 Comparing Subspaces

As we described in the previous section, in VMODEX, the design space is modeled as a tree and this kind of modeling causes the design space to be divided in several subspaces. Each subspace represents a unique instance of the architecture platform. On the other hand, solutions inside a subspace have exactly the same architecture components but the way that the application is mapped onto those components is different. We have extended VMODEX to allow designers to evaluate and compare the interesting subspaces from various perspectives. In the following subsections, we explain the techniques VMODEX provides for analyzing and comparing the properties of discovered design points in different subspaces.

3.1 Distance from the Global Pareto Optimal Solutions

A subspace containing local Pareto optimal solutions, which are closer to the global Pareto optimal set, is more preferable. The distance can be measured in two ways: 1) the number of solutions in a local Pareto optimal set which are also in the global Pareto optimal set, and 2) the average of Euclidian distances (in the objective space) between the solutions in a local Pareto optimal set and the nearest member of the global Pareto optimal set.

In the DSE tree, both distance measures can simply be evaluated. Just by looking at the tree, one can easily recognize which solutions of a local Pareto optimal set are in the global Pareto optimal set as well. For those solutions, their parents (in the tree) are Pareto optimal nodes; otherwise they become children of a relation node. For example, in Figure 1, all solutions in set S_3 are globally Pareto optimal. However, in set S_1, only one solution (out of four) is in the global Pareto optimal set. For the set S_2 the situation is the worst, none of the local Pareto points in this set is member of the global Pareto optimal set.

The second distance measure can be seen directly in the DSE tree as well. The color and thickness of edges show the distance from the nearest main Pareto optimal solutions. The edges in the path from the root to the main Pareto optimal solutions are the thickest and darkest since the distance is zero. As the distance increases the edges become thinner and lighter. The value of the average distance between solutions in a local Pareto optimal set and global Pareto optimal set is shown at the bottom of the local Pareto set (denoted by $\bar{d}$) If the designer is interested to know the exact value of the distance measure for a particular solution, then the distance value is shown by clicking the corresponding edge. For example, in Figure 1, the average distance for S_3 is zero, since all solutions in this set are globally Pareto optimal. However, the solutions in set S_2 are relatively far from the global Pareto optimal set since their edges are thin and light and the average distance is 0.24.

3.2 Coverage of Local Pareto Sets

Ziztler and Thiele [6] introduced the Coverage (*C*) metric, which directly compares two Pareto optimal sets with each other. The metric $C\ (S_1, S_2)$ calculates the proportion of solutions in S_2 which are dominated at least by one solution in S_1:

$$C(S_1,S_2) = \frac{\left|\left\{ s' \in S_2 \mid \exists s \in S_1 : s \leq s' \right\}\right|}{\left|S_2\right|}$$

Where ≤ is the dominance relationship. For two solutions *s* and *s′* it is said that "*s* dominates *s′*", if *s* is not worse than *s′* in all objectives and is strictly better in at least one objective. In fact, the function *C* maps the ordered pair (S_1, S_2) to the interval [0, 1]. The value $C\ (S_1, S_2)=1$ means that all members of S_2 are dominated by S_1 and $C\ (S_1, S_2)=0$ represents the situation where none of the solutions in S_2 are dominated by S_1. Since the domination operator is not symmetric, $C\ (S_1, S_2)$ is not necessarily equal to 1-$C\ (S_2, S_1)$ and thus both directions $C\ (S_1, S_2)$ and $C\ (S_2, S_1)$ have to be considered to understand how many solutions of S_1 are covered by S_2 and vice versa.

The *C* metric compares only two sets with each other. Thus, for comparing more than two sets, we propose a new metric called Total Coverage (*TC*), as follows:

$$TC(S_i) = \sum_{j=1, j \neq i}^{n} C(S_i,S_j) - C(S_j,S_i)$$

Where *n* is the number of comparing sets. *TC*>0 means that the dominating rate is higher than the dominated rate and the *TC*<0 implies that the solutions are more dominated by the other sets than they dominate solutions in the other sets. Therefore, a set with a bigger *TC* value is better.

In VMODEX, the designer is able to select the interesting local Pareto sets and compare them using the *TC* metric. For better understanding the dominance relations between sets, we visualize this metric. A directed weighted graph is used for visualizing the *TC* metric. Each comparing set is shown as a node in the *TC* graph. For each two sets S_1 and S_2, if $C\ (S_1, S_2) \neq 0$ then an edge is drawn from S_1 to S_2, of which the weight is equal to the *C* value. For each node, the *TC* value is calculated by the sum of the weights of outgoing edges minus the sum of the weights of incoming edges. The size of the nodes in the graph indicates the *TC* value. Therefore, nodes with higher *TC* values are bigger. Nodes with *TC*>0 (more dominating) are shown in blue while nodes with *TC*<0 (more dominated) are shown in red. Figure 3 shows the visualization of the *TC* metric for sets S_1 to S_4 shown in Figure 1. As can be seen in this figure, solutions in S_2 are dominated by all the other comparing sets (there are three incoming edge from S_1, S_3 and S_4), while there is no solution in S_2 that dominates a solution in the other sets (S_2 does not have any outgoing edge). Furthermore, we can understand that solutions in S_3 dominate more than half of the solutions in both sets S_1 and S_2. As a result, S_3 has the best *TC* value and therefore is the biggest node in the *TC* graph.

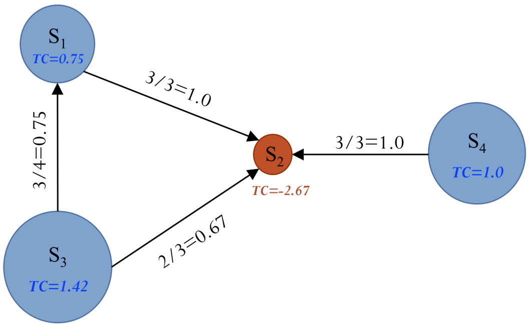


Figure 3. Visualization of *TC* metric

If the designer is interested to know more about the dominance relation between each two sets, such as which solutions in one set dominate which solutions in the other set, it is possible to select those sets to see more details. To this end, we visualize the dominance relation between two sets as follows. Solutions in both sets A and B are shown in two different rows. If a solution in A dominates a solution in B, an arrow is drawn between them coming out from the solution in A to the solution in B. Furthermore, a cross is displayed at the dominated solution in B to show that this solution is dominated by another one. Figure 4 shows the visualization of the dominance relation between two local Pareto sets S_1 and S_2. From this figure, we can understand that solution "LP1" in set S_1 dominates all the solutions in S_2 and therefore S_1 is absolutely better than S_2. The architectural components of each local Pareto set are shown on the left side of each set. As can be seen in Figure 4, both local Pareto sets S_1 and S_2 have exactly the same components except that S_2 has one more DRAM memory. Adding one DRAM is not beneficial since it increases the cost but does not yield a better processing time or a better energy consumption.

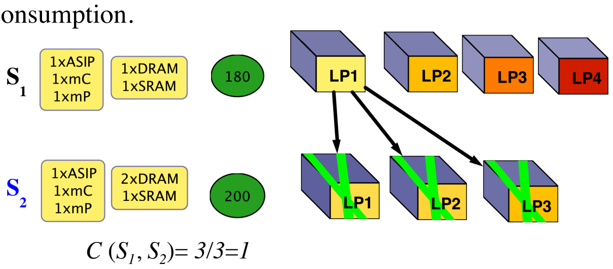


Figure 4. Visualization of *C* metric

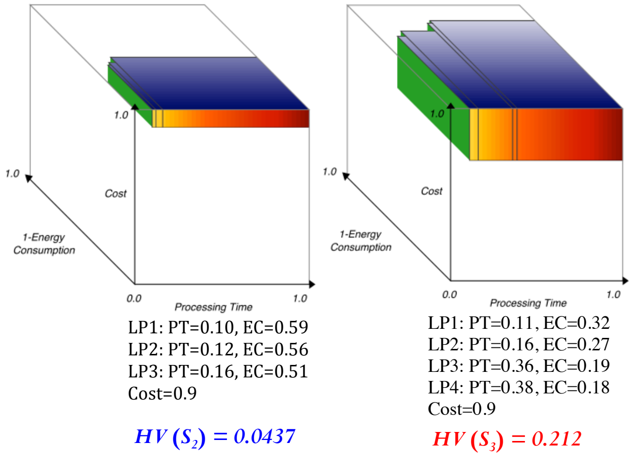


Figure 5. Visualization of the Hypervolume metric

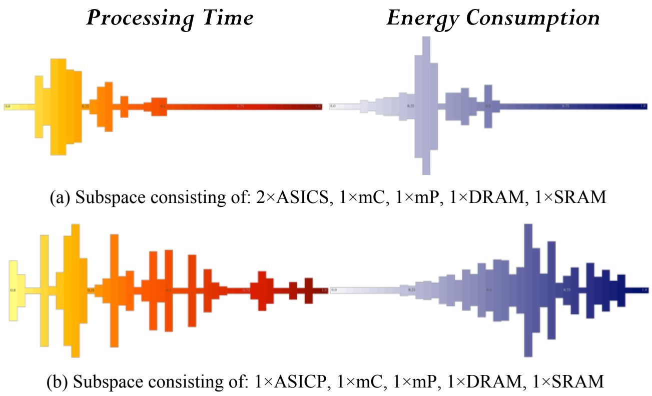
3.3 Size of the Dominated Region

In [6], the *Hypervolume* metric is proposed, which measures how much of the objective space is dominated by a given non-dominated set. A set with a larger hypervolume is desirable. We use this metric to compare the local Pareto optimal sets in different subspaces of the design space. Moreover, we visualize this metric to clearly show which area of the objective space is dominated by solutions in a local Pareto set. Figure 5 represents the visual from of the hypervolume metric for two local Pareto sets S_2 and S_3 shown in Figure 1. For better vision, each side of the cube (which shows one objective) is colored with the corresponding color scheme in the DSE tree. As can be seen in Figure 5, in both sets, only a small portion of the objective space is covered by the solutions in the local Pareto set. However, the size of the dominated space in S_3 is larger than S_2 (nearly five times).

3.4 Frequency Distribution of Objective Values

By using the color-coding technique in the DSE tree, the designer can generally see the objective values of design points in each subspace. However, for detailed analysis of the distribution of objective values in each subspace and also for easier comparison between different subspaces we propose a new visualization approach to show the frequency of objective values in a subspace. Since all solutions in a subspace have the same architecture cost, the frequency distribution is not applicable for this objective and should only be considered for the other two objectives: processing time and energy consumption.

For each objective, a horizontal axis from 0 to 1 is drawn and colored like the color-coding technique used for showing that objective in the DSE tree. For example, in our case, colors from yellow to red are used for representing the processing time and therefore, this color scheme is used for coloring the corresponding axis in the frequency distribution visualization. The height of each color bar in the (color-coded) objective axis indicates the number of design points with the objective value inside that range. Figure 6 shows the visualization of the frequency distribution for two subspaces. In Figure 6(a), the frequency distribution of all solutions in the subspace that contains two ASICs, one mC, one mP, one DRAM and one SRAM is shown. This subspace is indicated as S_3 in Figure 1. As can be seen in this figure, there is no design point with the best neither processing time nor energy consumption. However, for all design points, both the processing time and energy consumption is relatively good (less than 0.5). Therefore, for this particular architecture, with different mappings we can get approximately good design points. So can flexibly deal with different mappings, this architecture is a good solution. The subspace in Figure 6(b) consists of one ASIP, one mC, one mP, one DRAM and one SRAM, which is denoted as S_1 in Figure 1. In this subspace, the processing time is varying from the best to almost the worst. However, most of the design points have approximately good processing time. But, the energy consumption of most design points is quite high. Therefore, this architecture is not appropriate for obtaining low energy but if the designer is interested in performance, he should take care about the mapping because a wrong mapping decision can make the difference between the best or the worst processing time.



(a) Subspace consisting of: 2×ASICS, 1×mC, 1×mP, 1×DRAM, 1×SRAM

(b) Subspace consisting of: 1×ASICP, 1×mC, 1×mP, 1×DRAM, 1×SRAM

Figure 6. Visualization of frequency distribution of objective values

4 Multi-Objective Decision Making

As we mentioned before, 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. Usually, there can be a lot of Pareto optimal solutions, which need to be considered for the final decision. It is an issue to know how a decision maker chooses the 'best' solution from the set of Pareto optimal solutions. Therefore, some decision-making methods are provided in VMODEX to assist the designer in making better decisions.

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. Three basic problem formulations are as follows:

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

The *clustering* problem that intends to assign Pareto optimal solutions to different categories.

The *ranking* problem that aids to order the Pareto optimal points from the best to the worst, with respect to the designer's preferences.

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

4.1 Choice Problem

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

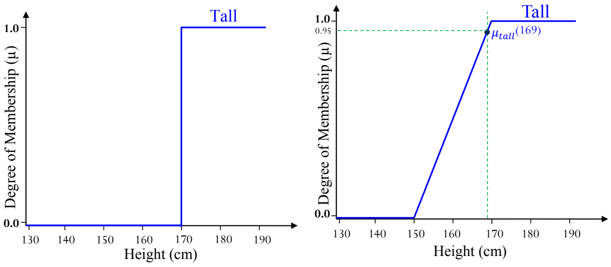
4.1.1 Filtering

VMODEX enables designers to filter solutions in the Pareto optimal set based on either objective values or design parameters or both of them. The designer can determine his preference objective values by setting the upper and lower limits on each objective. Then those Pareto optimal points with objective values inside the selected ranges are chosen as most preferred solutions. Therefore, the designer has the ability to eliminate the non-preferred points from the Pareto optimal set and consider only solutions, which satisfy the preferences with respect to the objective values.

The designer is also able to determine his preferences on design parameters and consider only the Pareto optimal points with preferred parameters. For instance, in our case study, the designer may preferred the points, which 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. Establishing the preferences on combination of objective values and design parameters is provided as well.

4.1.2 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. 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 is "tall" (degree of membership is one) and a man with 169 cm is "not tall" (degree of membership is zero). Therefore, in Boolean logic, a person is either a member of tall set or is not a member. Figure 7(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 fuzzy membership function for "tall" set is shown in Figure 7(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 "a" has higher degree of membership in "tall" set than person "b"; this means that "a" is taller than "b".



(a) Boolean logic

(b) Fuzzy logic

Figure 7. Tall membership function

We can use fuzzy logic to extend the notion of the dominance relation between two solutions. Therefore, instead of saying "a dominates b" or "a does not dominate b", we can say that "a dominates b by degree μ". 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

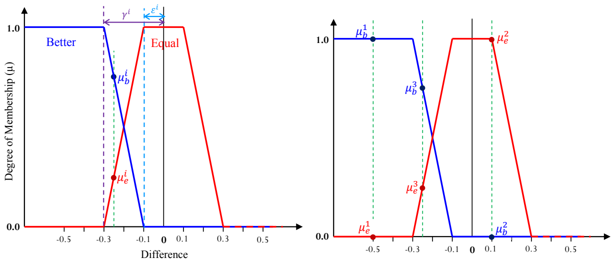
For computing the dominance degree of solution *a* over solution *b*, we should consider for each objective, in which degree solution *a* is better than or equal to solution *b*. An example of definition of fuzzy sets for "better" and "equal" functions is shown in Figure 8(a). Here we consider a minimization problem that means all objectives are to be minimized and therefore smaller objective values are better. In Figure 8(a), two parameters ε and γ should be determined by the decision maker who is familiar with the problem. The value of these parameters may be different for each objective. ε^i indicates the interval within which an improvement or degradation on objective *i* is meaningless. γ^i defines the relevant but not a significant improvement for objective *i*. In Figure 8(a), μ_b^i indicates the degree of "betterness" and μ_e^i denotes the degree of "equality" on the ith objective. The fuzzy dominance relation can be expressed as follows:

It is said that solution *a* dominates solution *b* by degree μ_d (*a*, *b*) with:

$$\mu_d(a,b) = \frac{1}{M}\sum_{i=1}^{M}\mu_b^i(f_i(a)-f_i(b)) + \frac{1}{2M}\sum_{i=1}^{M}\mu_e^i(f_i(a)-f_i(b))$$

Where *M* is the number of objectives and f_i is the ith objective function. μ_d (*b*, *a*) shows the dominance degree of *b* over *a* or, on the other hand, it indicates in which degree solution *a* is dominated by solution *b*. Note that the fuzzy dominance relation is not symmetric and therefore "dominating by degree μ" and "being dominated by degree μ′ " have different fuzzy values.

For clarifying the fuzzy dominance relation described above, we are going to give a numerical example. In this example, we calculate the dominance degrees for two solutions *a*=(0.1,0.5,0.3) and *b*=(0.6,0.4,0.57). The fuzzy membership functions for "better" and "equal" are the



(a) Fuzzy sets for the ith objective

(b) Fuzzy sets and membership values for the numerical example

Figure 8. An example of fuzzy membership functions for better and equal

same as ones shown in Figure 8(a). For all objectives ε=0.1 and γ=0.3. Figure 8(b) shows the values of μ_b and μ_e for all objectives. The dominance degree of *a* over *b* is computed as follows:

$$\begin{aligned}\mu_d(a,b) &= \frac{1}{3}(\mu_b^1(0.1-0.6)+\mu_b^2(0.5-0.4)+\mu_b^3(0.3-0.57)) \\ &+\frac{1}{6}(\mu_e^1(-0.5)+\mu_e^2(0.1)+\mu_e^3(-0.27)) \\ &=\frac{1}{3}(1+0+0.85)+\frac{1}{6}(0+1+0.15)=0.67+0.19=0.86\end{aligned}$$

In the same way we can calculate the dominance degree of *b* over *a*, μ_d (*b*, *a*), which is 0.19. Therefore *a* dominates *b* by degree 0.79 and is dominated by *b* by degree 0.19. Thus solution *a* is more preferred than *b*.

We define the goodness of solution s_i in a Pareto optimal set *S* as the average of dominance degrees of that solution over other solutions in the set, as follows:

$$G(s_i)=\frac{1}{N-1}\sum_{j=1,j\neq i}^{N}\mu_d(s_i,s_j)$$

Where *N* is the number of solutions in the Pareto optimal set *S*. The solutions with the highest *G* values are considered as the most appropriate solutions.

In VMODEX there are two ways for selecting most preferred solutions using the fuzzy scheme:

1. The decision maker can set a threshold for the *G* value and then all the solutions with a *G* value higher than the threshold are selected.
2. The decision maker can determine the desirable number of selected solutions (*n*). Then the solutions with the *n* highest *G* values are selected.

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 9 represents the fuzzy dominance graph for our case study. Each solution is shown as a node in the FDG. The size of the node indicates its *G* value. The bigger the node means the greater *G* value and therefore a more preferred solution. The decision maker should define a threshold (θ) for connecting the nodes. If the dominance degree of *a* over *b* is greater than the threshold (μ_d (*a*, *b*)> θ) then a directed edge is established from *a* to *b*. Therefore, a node with more outgoing edges means that it dominates more solutions with the degree higher than θ and a node with more incoming edges means it is dominated by more solutions with the degree higher than θ. In Figure 9, the threshold is 0.6 and as can be seen in this figure solution P5 has the greatest *G* value since the size of its corresponding node in the graph is the biggest and it dominates 10 (out of 14) solutions with degree higher than θ as it has 10 outgoing edges. Furthermore, there is no solution that dominates P5 with a degree higher than θ as there is no incoming edge. However, P3 is the most inappropriate solution since its node size is the smallest and half of the solutions (7 out of 14) dominate it with degree higher than θ. In Figure 9, the five solutions with the highest goodness values are highlighted by a blue background color, which are P5, P2, P11, P8, and P7 respectively.

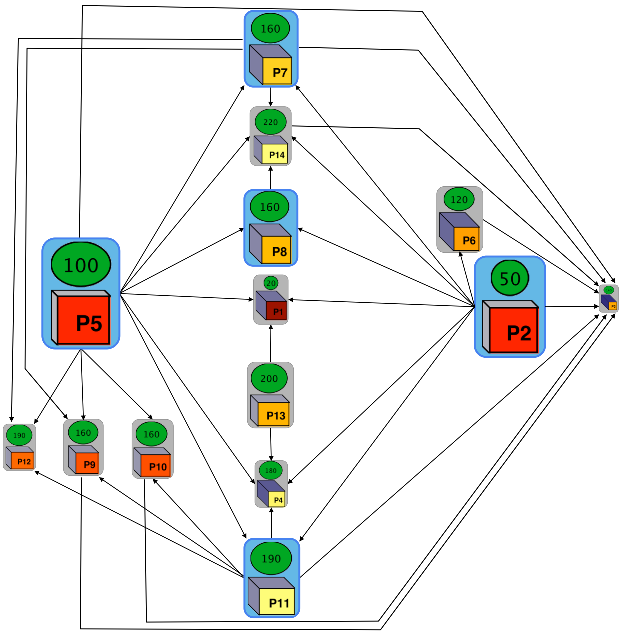


Figure 9. Visualization of fuzzy dominance relations

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 10 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).

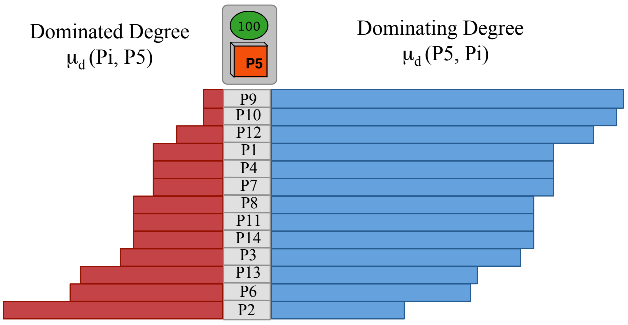


Figure 10. Visualization of fuzzy dominance relations between P5 and all the other solutions

4.2 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 can be broadly classified into two types: exclusive and hierarchical. In the 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.

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 the Euclidian distance in the objective space. The smaller distance indicates higher similarity.

4.2.1 Exclusive Clustering

The Leader-Follower algorithm [2] is one of the most popular exclusive methods. It is simple and efficient for clustering data sets. The decision maker must specify a threshold for the distance. 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 no current cluster is sufficiently close to the solution, a new cluster is created. We define the cluster center to be the average of all points in the cluster. For instance, if the ith cluster contains one solution s_1=(4,3,5) and during the clustering procedure another solution such as s_2=(4,1,7) needs to be added to the ith cluster, then its new center is calculated as:

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

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.

For each cluster, the closest solution to the cluster center is chosen as a representative solution. These representative solutions can be considered as the most promising solutions for the final decision. Therefore, by applying the proposed clustering, the Pareto optimal set can be reduced to a smaller subset of representative solutions, which are the most preferred solutions.

To better understand the result of the clustering, we propose a visualization method showing how the solutions are grouped in different clusters. Figure 11 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 11 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 C1 in Figure 11). The cluster C4 in Figure 11 is the biggest cluster containing 5 solutions. In each cluster, the representative solution is highlighted by a yellow border such as solution P12 in cluster C6.

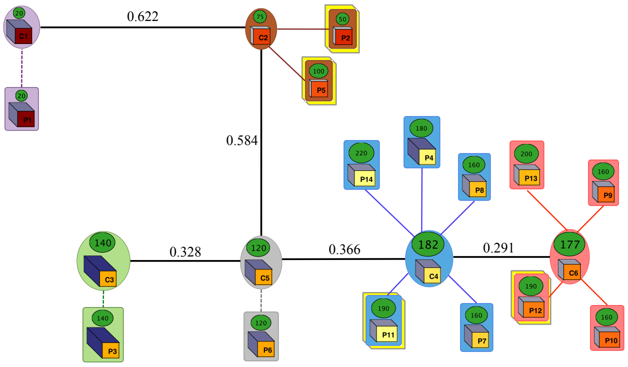


Figure 11. Visualization of Leader-Follower clustering

To understand the relative position of clusters in the objective space, i.e., which clusters are closer to each other and which ones are far away and to know in which parts of the objective space more Pareto optimal solutions are found, we visualize the relative position of clusters. 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 Euclidian distance between the corresponding cluster centers. Then a Minimum Spanning Tree (MST) is created for the graph *G*. The constructed MST is used to show the distances between clusters. In Figure 11 the way that the cluster centers are connected together is the same as the connections in its 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 11, clusters C3, C4, C5 and C6 are relatively close to each other and contain more than 75% (11 out of 14) of discovered Pareto optimal points. Therefore we can see that in a subspace of objective space, which has relatively good processing time but a higher cost, there are more Pareto optimal solutions.

4.2.2 Hierarchical clustering

The Leader-Follower clustering algorithm is efficient and conceptually simple, but it has some drawbacks. It requires a specific distance threshold as input, which has a major effect on the clustering results. Furthermore, the output is nondeterministic and strongly depends on the order of the processed solutions. Hierarchical clustering does not require any predetermined parameter and is deterministic. However, the complexity of hierarchical clustering increases at least quadratic with the number of data points. The process of hierarchical clustering is as follows:

1. Assign each solution to a separate cluster. Therefore the number of clusters in this step is equal to the number of solutions in the Pareto optimal set.
2. Find the two closest (most similar) clusters and merge them into a single cluster. This new cluster replaces the two old clusters.
3. Repeat step 2 until only one cluster is left, which contains all the solutions.

The similarity of two clusters can be measured in three ways: single-linkage, complete-linkage and average-linkage. In 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 two clusters. In 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. 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 12 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 12, the two solutions P9 and P10 are merged first, and at the last merge the solution P1 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).

Once the complete hierarchical dendrogram is constructed, depending on the type of decision, different cutting policy 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 12 cutting the dendrogram at 0.35 yields 3 clusters: one large cluster containing 11 solutions (black lines), one cluster consisting of P2 and P5 (blue lines), and one cluster including only P1 (red line).
- Cutting in such a way that a certain number of clusters remain. For organizing solutions in k cluster, the k-1 links from the top should be removed.

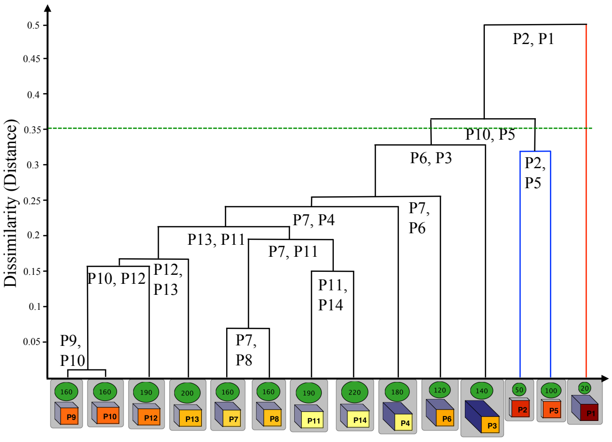


Figure 12. Visualization of hierarchical clustering

4.3 Ranking Problem

In the ranking approach, Pareto optimal solutions are systematically compared to one another, taking into account all objectives. The comparisons between the solutions lead to numerical results that somehow reflect the preference values of solutions. Then solutions are sorted by their preference values. The decision maker expresses his preferences among different criteria by assigning a weight to each criterion. These weights represent the relative importance of objectives. Before applying a ranking method, we normalize objective values to make them scale independent. At the end of normalization, all solutions get a value in the range [0,1] for their objective values. In the literature, several methods have been proposed for ranking the Pareto optimal solutions. Here we explain three ranking methods, which are provided in VMODEX. The main difference between these ranking methods is the way that they compute the aggregation of the normalized objective values and their weights for obtaining an overall preference value for each solution.

For describing the ranking methods, we consider a Pareto optimal set of n solutions S_i ($i=1,2,\ldots,n$) in which their objective values are normalized. These solutions are to be evaluated with respect to a set of m criteria C_j ($j=1,2,\ldots,m$), which should be minimized. V_{ij} denotes the value of the i^{th} solution in terms of the j^{th} objective. A weighting vector $W=(W_1,W_2,\ldots,W_m)$ represents weights of n objectives. PV_i indicates the overall preference value for solution i.

4.3.1 TOPSIS

TOPSIS (Technique for Order Preference by Similarity to Ideal solution) [4] 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. IS^+ and IS^- are obtained from the weighted objective values in the Pareto optimal set. IS^+ is found by constructing a vector of best values and IS^- is calculated by constructing a vector of worst values as follows:

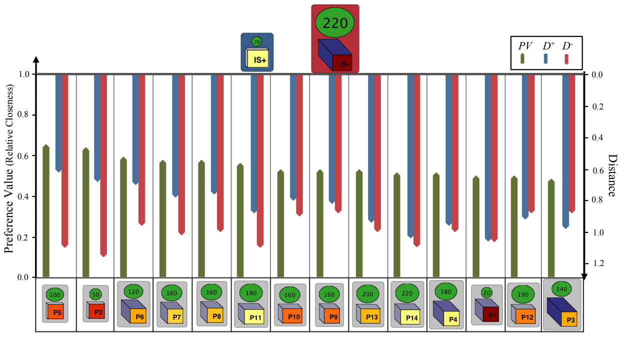


Figure 13. Visualization of TOPSIS ranking method

$$R_{ij} = W_j V_{ij}$$

$$IS^+ = (R_1^+, \ldots, R_j^+, \ldots, R_m^+) \qquad R_j^+ = \{\min_{i=1}^{n}(R_{ij})\}$$

$$IS^- = (R_1^-, \ldots, R_j^-, \ldots, R_m^-) \qquad R_j^- = \{\max_{i=1}^{n}(R_{ij})\}$$

For each solution, the Euclidian distance from the positive ideal solution (D^+) and negative ideal solution (D^-) is computed and then the overall preference value for each solution (PV_i) is calculated as follows:

$$PV_i = \frac{D_i^-}{D_i^+ + D_i^-} \qquad 0 \le PV_i \le 1$$

PV_i indicates the relative closeness of the ith solution to the positive ideal solution. Therefore, a greater value means that the solution is closer to IS^+ and thus is more preferred. 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 the IS^+ and IS^- and their *PV* values, we visualize this information such as shown in Figure 13. The length of the blue bar represents the distance between the corresponding solution and IS^+ (D^+). 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^- (D^-). 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 13, the solutions 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 (D^+ and D^-), 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 13, except P12 and P3, all the other solutions are closer to IS^+ than IS^- ($D^+<D^-$) since their blue bars are shorter than their red bars.

4.3.2 PROMETHEE

Instead of comparing the distances from ideal solutions, the PROMETHEE (Preference Ranking Organization METHod for Enrichment Evaluations) [5] is based on the pairwise comparisons of solutions along 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 in a Pareto optimal set. The procedure of PROMETHEE methodology can be summarized as follows:

1. Solutions are compared pairwise for each objective. For example, the difference between two solutions S_1 and S_2 on objective *j* is indicated as d_j (S_1, S_2). Afterwards, the amount of difference is mapped to a value in the preference range [0, 1] and denoted as P_j (S_1, S_2). This preference value indicates the degree of preference associated to the better solution on the jth objective. For small deviations, a small preference value is allocated to the better solution and even possibly 0 if the deviation is negligible. The larger the deviation, the larger the preference value.
2. The overall preference index of solution S_1 over S_2 for all objectives is computed as follows:
$$\pi(S_1, S_2) = \sum_{j=1}^{m} W_j P_j(S_1, S_2)$$
3. For each solution S_i, the positive flow (ϕ^+) and the negative flow (ϕ^-) are defined as:
$$\phi^+(S_i) = \frac{1}{n-1} \sum_{k=1, k\neq i}^{n} \pi(S_i, S_k) \qquad \phi^-(S_i) = \frac{1}{n-1} \sum_{k=1, k\neq i}^{n} \pi(S_k, S_i)$$
4. For each solution S_i, the net flow ϕ (S_i) is considered as:
$$\phi(S_i) = \phi^+(S_i) - \phi^-(S_i) \qquad -1 \le \phi(S_i) \le 1$$

The net flow is the balance between the positive and negative flows. The higher the net flow means the better the solution. ϕ (S_i) >0 means that the solution is more dominating all the other solutions on all objectives and ϕ (S_i) <0 indicates it is more dominated. The complete ranking of all solutions from the best to the worst is obtained by sorting their ϕ values in descending order.

ϕ_j (S_i) is the single objective net flow for solution S_i when only the jth objective is considered and can be defined as follows:

$$\phi_j(S_i) = \frac{1}{n-1} \sum_{k=1, k\neq i}^{n} [P_j(S_i - S_k) - P_j(S_k - S_i)]$$

It expresses how a solution S_i is dominating (ϕ_j (S_i)>0) or dominated (ϕ_j (S_i)<0) by all the other solutions on the jth objective. ϕ_j (S_i) is not dependent on the weights of the objectives but the degrees of preference are taken into account. It is particularly useful for analyzing the “quality” of solutions on each specific objective separately. For comparing the various Pareto optimal solutions based on their amount of domination on each objective, we visualize the ϕ_j (S_i) values, such as shown in Figure 14 (the data are captured from our case study results explained in section 3). 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 ϕ_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

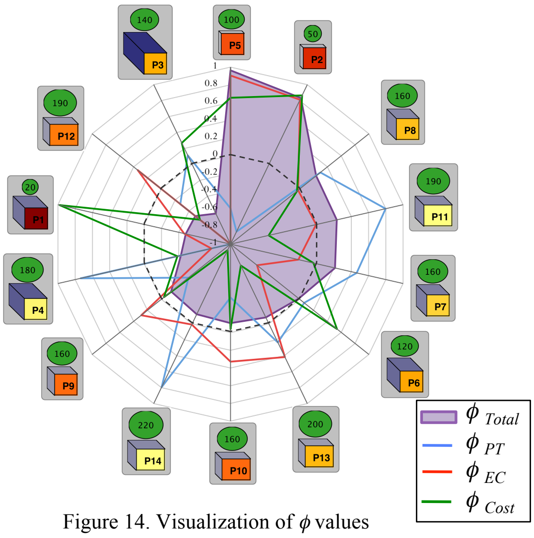


Figure 14. Visualization of ϕ values

each objective, a radar plot is drawn with a specific color. The points constructing a radar plot are the corresponding ϕ_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 is based on the sorted ϕ values in descending order. In Figure 14, the filled radar plot shows the total ϕ values (considering all objectives). As can be seen in Figure 14, the cost net flow for solution P1, ϕ_{Cost} (P1), is 1, which means it dominates all the other solutions in terms of the cost. While the ϕ_{PT} (P1) is -1 which indicates that P1 is dominated by all the other solutions with respect to the processing time. Its net flow for energy consumption is less than zero (ϕ_{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 ϕ_{Total} (P1), which considers all the objectives is small and it is on the three worst solutions according to the PROMATHEE ranking method.

4.3.3 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 shown in Figure 15. This figure represents the results of ranking the Pareto optimal solutions obtained from our case study by applying three ranking methods: SWO (Sum of Weighted Objectives), TOPSIS and PROMETHEE. In the SWO method, the weighted sum of objective values over all objectives is considered as overall preference value. Since we consider the minimization problem, a smaller value indicates the more preferred solution. 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.

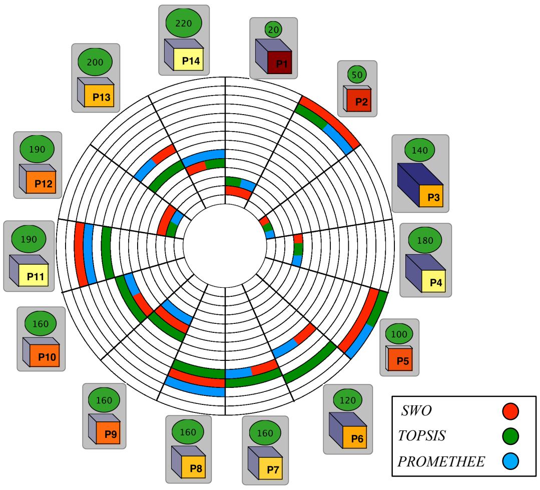


Figure 15. Visualization of ranking results

The innermost one indicates the worst rank and the outermost one shows the best rank. For example, in Figure 15, 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 Conclusion

In this paper we extend our visualization tool, VMODEX, which is developed for the analysis of multi-objective design space exploration for embedded systems. In the extended version of VMODEX, we have provided several methods and visualization approaches for evaluating and comparing the properties of discovered design points in different subspaces of the explored design space. Furthermore, we have included various multi-objective decision making techniques to help the designer analyze the trade-offs between Pareto optimal design points and select the most appropriate one for system implementation. Moreover, several visualization schemes are proposed for better understanding and detailed analysis of the results of multi-objective decision making techniques.

References

[1] Taghavi, T., Pimentel, A.D.: Visualization of multi-objective design space exploration for embedded systems, DSD 2010, Lille, France.

[2] Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd Edition, John Wiley & Sons, 2001.

[3] Johnson, S. C., Hierarchical Clustering Schemes, *Psychometrika*, 2:241-254, 1967.

[4] Hwang, C. L., Yoon, K., Multiple attributes decision making methods and applications, Springer: Berlin Heidelberg, 1981

[5] Brans, J.P., Mareschal, B., and Vincke, P., PROMETHEE: A new family of outranking methods in MCDM", in: IFORS 84, Holland.

[6] Zitzler, E., Thiele, L.: Multiobjective Optimization Using Evolutionary Algorithms, in PPSN-V, 1998, pp. 292-301.