

# Design Metrics and Visualization Techniques for Analyzing the Performance of MOEAs in DSE

Toktam Taghavi, Andy D. Pimentel

Computer Systems Architecture Group, Informatics Institute

University of Amsterdam

Amsterdam, the Netherlands

{T.TaghaviRazaviZadeh, A.D.Pimentel}@uva.nl

**Abstract**—VMODEX is an interactive visualization tool to support system-level Design Space Exploration (DSE) of MPSoC architectures. It was initially developed to help designers to get insight into the search process of Multi-Objective Evolutionary Algorithms (MOEAs) that are typically used in the DSE process, and facilitates the analysis of the DSE results. In this paper, we extend VMODEX to help algorithm developers as well. Since there are many different MOEAs to search a design space and there is no conclusive answer regarding which algorithm is the best for a specific problem, finding the best optimization algorithm is a big challenge. However, using VMODEX, algorithm developers can easily evaluate and compare the results of different searching algorithms, for a given problem, with respect to their efficiency and effectiveness, in order to find the best optimization algorithm. Then, the best optimization results are delivered to the designers for analyzing the design space exploration process.

**Keywords**—component; Design space exploration; multi-objective evolutionary algorithms, visualization, quality metrics.

## I. INTRODUCTION

Modern embedded systems come with contradictory design constraints. On one hand, these systems often target mass production and battery-based devices, and therefore should be cheap and power efficient. On the other hand, they need to achieve high (real-time) performance. This wide spectrum of design requirements leads to complex heterogeneous multi-processor system-on-chip (MPSoC) architectures. The complexity of these embedded systems forces designers to model and simulate systems and their components early during the design process to explore the wide range of design choices. Such design space exploration, during which multiple criteria should be considered simultaneously, is called multi-objective design space exploration. 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 (MOEA), 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. The algorithm usually starts with a randomly generated population and calculates the fitness value for each solution within the population. The fitness function is problem specific and determines how good the solution is. Solutions with high fitness in the current population are selected for reproduction. Such solutions are modified (recombined and randomly mutated) to form a new population. This is motivated by the hope that the new population will be better than the old one. The new population is then used in the next iteration of the algorithm. Generally, the algorithm terminates when either the predefined number of generations has been reached, or when there is no improvement in the fitness values of the solutions.

Although the goal of MOEAs is to find the Pareto optimal solutions with respect to the design criteria, there is no guarantee to reach real optimal solutions. This is because of the heuristic nature of the MOEAs. They try to find optimal solutions. However, typically they are only able to find good approximation of optimal solutions that are not far away from the true optimal solutions. Therefore, metrics are needed to evaluate the quality of the discovered solutions. Furthermore, many different MOEAs are proposed in literature, which may have different performance on different problems, and there is no conclusive answer regarding to which algorithm is the best for a specific problem. On top of that, MOEAs are highly sensitive to the parameters being used, such as mutation rate, crossover rate, repair strategy, individual encoding, etc. These parameters have major effects on the performance of the algorithm and have to be fine-tuned by hand. Therefore, coming up with the best searching algorithm, which efficiently and effectively explores the design space and finds high quality solutions is a big challenge.

In single-objective optimization, the quality of solutions can be defined by measuring the values of the objective function; the smaller (or larger) the value, the better the

solution. However, in the context of multi-objective optimization, it is difficult to define appropriate quality measures since it is not clear what “quality” means in the presence of multiple objectives. Therefore, several quality metrics have been proposed in the literature to assess different aspects of the goodness of the Pareto optimal solutions obtained by a MOEA. In this paper, we also propose three new metrics ( $W\text{SGR}$ ,  $\sigma_{mst}$  and  $DFPOS$ ), which are useful for comparing the performance of different algorithms or one algorithm with different parameter settings.

For evaluating and comparing different Pareto optimal sets, the results of a quality metric are typically shown in a table or displayed in a 2D graph in which the value of the quality metric is shown to compare one Pareto optimal set to another one. Although these kinds of representation are useful to find out which Pareto optimal set is better in respect to a certain quality aspect, they do not provide insight on why a Pareto optimal set is good (bad) according to the specific metric.

Utilizing visualization techniques can be helpful for further analysis of the Pareto optimal solutions. In general, a quality metric encapsulates the properties of a Pareto optimal set to one scalar value that somehow reflects certain quality aspect. Therefore, some useful information may be lost because of this compression. However, visualizations can provide better understanding of the quality of the results and enable us to perform more accurate analysis. But, there is a significant lack of studies on representing and visualizing the different quality aspects of Pareto optimal sets. The conventional way is plotting the optimal solutions in the objective space. Although this method is simple and displays the overall quality, it is limited to a maximum of three objectives (maximal three dimensions) and does not provide a detailed description of different quality aspects. Ang et al. [4] proposed Distance & Distribution (DD) charts, in the first of which the Pareto optimal solutions are plotted against their distance to the true Pareto front and in the second of them the Pareto optimal solutions are plotted against the distance between each consecutive solutions. In these plots, only the distance values of solutions are shown and the objective values are not considered. So, they do not reveal any information about the location of the Pareto optimal solutions in the objective space.

In this paper, we propose several visualization approaches, which enable researchers to do detailed analysis of the quality of Pareto optimal solutions. These visualizations provide insight on the reasons behind the strength/weakness of a Pareto optimal set with respect to a particular metric. To this end, we integrated various quality metrics (including both existing metrics and new ones) and their visualizations in our interactive visualization tool, VMODEX (Visualization of Multi-Objective Design space eXploration), which was proposed in [5]. VMODEX 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 help algorithm developers as well. Using VMODEX, they can easily evaluate and compare the results of different searching algorithms, for a given problem, with respect to their efficiency and effectiveness in order to find the best optimization

algorithm. Several quality metrics and their visualizations are provided in the extended version of VMODEX to enable algorithm developers to perform a comprehensive study on the properties of the discovered optimal solutions and evaluate the performance of the used searching algorithms from different perspectives. Therefore, if a Pareto optimal set is not good according to a specific metric, one can easily understand the reason behind this weakness. Our main contributions are:

- Proposing three new quality metrics:  $W\text{SGR}$ ,  $\sigma_{mst}$  and  $DFPOS$ .
- Visualizing both existing and new metrics for more detailed and accurate analysis.
- Presenting a case study, which demonstrates the usefulness of using VMODEX.

The rest of the paper is organized as follows. Section II describes the goals that should be considered in multi-objective optimizations to yield high quality results. In Section III we briefly explain how VMODEX visualizes the multi-objective DSE process. Section IV introduces quality metrics and their visualization methods we have provided in VMODEX. Section V presents a case study in which the qualities of the outputs of different MOEAs for a specific problem are compared using various metrics. This case study illustrates the benefits of our tool, which integrates and visualizes different quality metrics in a single environment. Finally, Section VI concludes the paper.

## II. GOALS IN MULTI-OBJECTIVE OPTIMIZATION

Usually two distinct goals are considered in multi-objective optimizations:

- 1) Find solutions as close as possible to the true Pareto front
- 2) Discover solutions as diverse as possible.

The first goal is essential for any optimization task. Finding solutions, which are far away from the true Pareto optimal set, is not appropriate. In most multi-objective optimization problems, the true Pareto front is not known. Therefore, for measuring the closeness, a reference Pareto front that is the best-known approximation of the true Pareto front, is used. The reference Pareto front can be made by combining all the optimal solutions from numerous searches and then removing dominated solutions from the combined set. The second goal is completely specific to multi-objective optimization. Diversity means covering the entire Pareto optimal region uniformly. Only with a diverse set of solutions, we can have a good set of trade-offs among objectives. The diversity can be divided in two different measures: extent (the spread of extreme solutions) and distribution (the relative distance between solutions).

Since both goals are important, an efficient MOEA must satisfy both of them adequately. However, the multi-objective optimization goals are two distinct concepts. Thus, no single quality measure is able to indicate the performance of an MOEA in an absolute sense. Therefore, for measuring the performance of different MOEAs and comparing how well they achieved the multi-objective optimization goals, several metrics need to be used in order to assess the quality of the obtained Pareto fronts from different perspectives.

### III. A BRIEF INTRODUCTION TO VMODEX

In this section, we briefly explain how VMODEX [5] visualizes a multi-objective design space exploration. Fig. 1 represents an example Pareto optimal set, which is visualized by VMODEX. In this example, a parallel multi-media application is mapped to an MPSoC platform architecture. Here, three objectives are considered: processing time, energy consumption and cost of different MPSoC design instances. In Fig. 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. In this example, the platform architecture consists of an Application Specific Integrated Circuit (ASIC), an Application Specific Instruction Processor (ASIP), a general-purpose microprocessor (mP), a microcontroller (mC), two Dynamic RAMs (DRAM) and one Static RAM (SRAM). The fifth level shows the architecture cost. Since this is the first level that 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 Pareto optimal solutions found by the MOEA. Each solution represents a unique mapping of application tasks and communication channels onto the architecture components. Solutions are labeled by an index, in the order of increasing processing time. For each solution, its parents at the previous levels show its design parameters and cost. For instance, P7 has the following architectural components: two processors, of which one is an ASIP and the other one an mC, and one memory of the type DRAM. 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 the visual form of the quality metrics with the multi-objective DSE visualization, the same metaphors are used for showing the objective values and solutions. Thus, one will not be confused by the different

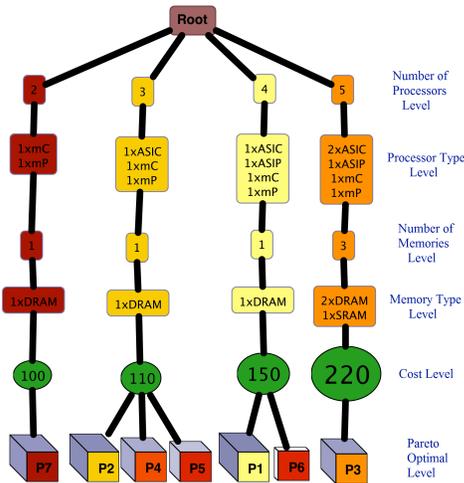


Figure 1. An example of Pareto optimal solutions visualized by VMODEX

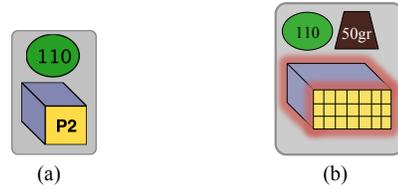


Figure 2. An example of representing a solution

representations of the same thing. For instance, in the visualization of quality metrics, the solution P2 in Fig. 1 is shown like in Fig. 2 (a). 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.

It should be mentioned that although here we only show three objectives, VMODEX is able to easily visualize more than three objectives. For those objectives, which are only dependent on the architectural components such as weight, physical size, etc., extra levels can be added after the parameter levels (like the cost level in our example). Showing objectives that are dependent on the mapping is also easy. Each node has some attributes like shape, orientation, size, color, transparency, texture, border, glow, etc. Each attribute can be assigned to one objective. Fig. 2(b) shows an example of representing a solution with six objectives. In this figure texture density indicates average utilization, the size of the glow shows the temperature and the size of the trapezoid indicates the weight. The other three objectives are the same as Fig. 2(a). For a detailed study of how VMODEX visualizes design space parameters and objectives as a tree, the interested reader is referred to [5].

### IV. QUALITY METRICS AND THEIR VISUALIZATION METHODS

Particularly, in proposing visualization methods for quality metrics, the following challenges have to be addressed:

- The visualization approach should not be limited to the number of objectives. It must be scalable to be used in more than three objectives problems.
- The visualization techniques should be simple and straightforward. Complex approaches do not reveal useful information clearly and some valuable issues may not be discovered. Furthermore, simple methods enable us to easily compare two or more non-dominated sets with each other.
- For showing a specific objective in different visualization methods for different metrics, the same metaphor should be used. Thus, there is no confusion because of representing one thing in different ways.

In this section, we first describe the different metrics provided in VMODEX for measuring the quality of non-dominated sets found by a MOEA. Then, we explain the visualization approach developed for each metric as well as the benefits of such visualization. All the visualizations proposed in this paper satisfy the above challenges. For all metrics, the normalized values of objectives are used. Therefore, all objective values are in the range [0, 1].

In the following subsections, we categorize the quality metrics in four groups: 1) closeness metrics to evaluate the distance of the obtained Pareto front to the true Pareto front, 2) diversity metrics to evaluate the spread or distribution of the solutions in the found Pareto front, 3) combined metrics to evaluate both the closeness and diversity of the discovered solutions in an implicit manner, and 4) dynamic metrics to show how the quality of solutions with respect to a specific metric is varying during MOEA execution.

#### A. Closeness Metrics

These metrics compute a measure of the closeness of a Pareto front found by a MOEA ( $PF_{known}$ ) from the true Pareto front ( $PF_{true}$ ).  $PF_{true}$  can either be a set of (theoretical) true Pareto optimal points (if known) or a reference set, which contains the best-known non-dominated points from a combination of numerous MOEA runs.

##### 1) Existing Metrics

Van Veldhuizen [2] proposed two metrics for measuring the distance of the found Pareto optimal set to the true Pareto front: Error Ratio (*ER*) and Generational Distance (*GD*). These metrics are widely used in literature because of their simplicity.

The *ER* metric indicates the percentage of solutions in  $PF_{known}$  that are not members of the  $PF_{true}$ . A smaller value of *ER* means a better approximation of the true Pareto front. This metric takes a value between zero and one.  $ER=0$  means that all solutions of  $PF_{known}$  are members of the  $PF_{true}$  and  $ER=1$  means that no solution is a member of the true Pareto front.

The *GD* metric calculates the average distance of the solutions in  $PF_{known}$  from  $PF_{true}$ . The distance measure is the Euclidean distance (in the objective space) between a solution in  $PF_{known}$  and the nearest member of  $PF_{true}$ . It is clear that a value of  $GD=0$  indicates that all the solutions of  $PF_{known}$  are in  $PF_{true}$ . A set having a smaller value of *GD* is better.

##### 2) Weighted Sum of True Pareto and relatively close points Generation Ratio (*WSGR*)

Both *ER* and *GD* metrics have some drawbacks. To overcome those shortcomings, we propose a new metric called *WSGR*. The drawback of the *ER* metric can be illustrated by the following example: if an algorithm finds only two Pareto optimal solutions, of which one is in  $PF_{true}$ , then its error ratio is 0.5. But if another algorithm finds 20 Pareto optimal solutions and 8 of them are in  $PF_{true}$ , then the error ratio is 0.60. Clearly, the second algorithm performs better since it finds more members of the true Pareto optimal set, but the error ratio indicates that it performs worse. The drawback of *GD* is that when all solutions in  $PF_{known}$  are in the true Pareto front ( $GD = 0$ ), it does not determine how good the algorithm is in finding true Pareto points. For example, if an algorithm finds 3 Pareto optimal solutions of which all of them are in  $PF_{true}$  and another one finds 10 Pareto optimal solutions of which all of them are in  $PF_{true}$ , the value of *GD* for both of them is zero. However, it is clear that the second algorithm performs better.

Both *ER* and *GD* metrics do not consider the proportion of the true Pareto points, which could be found by the MOEA. Thus, we propose a new metric that considers the percentage of the true Pareto optimal solutions found by the algorithm as well

as the percentage of solutions in  $PF_{known}$ , which are relatively close to the  $PF_{true}$  as follows:

$$WSGR = w_1 \frac{\sum_{i=1}^{|PF_{true}|} s_i}{|PF_{true}|} + w_2 \frac{\sum_{i=1}^{|PF_{known}|} c_i}{|PF_{known}|} \quad (1)$$

$$s_i = \begin{cases} 1 & i \in PF_{known} \\ 0 & i \notin PF_{known} \end{cases} \quad c_i = \begin{cases} 1 & d_i < \theta \\ 0 & d_i \geq \theta \end{cases}$$

For measuring the relative closeness, a threshold ( $\theta$ ) needs to be defined. If the minimum distance of solution  $i \in PF_{known}$  from  $PF_{true}$  is less than the threshold, then it is considered as a relatively close solution and the value of its  $c_i$  is one. The weights show the relative importance of two combined components of equation (1). Since finding true Pareto optimal points has a higher priority than discovering relatively close solutions,  $w_1$  should be higher than  $w_2$ . A bigger value of *WSGR* means that the MOEA has obtained more solutions of  $PF_{true}$  and also more solutions in  $PF_{known}$  are close enough to  $PF_{true}$  and therefore is better. Using each element of equation (1) separately, as a quality metric, has some drawbacks. By considering only the first part, the value of the metric for two algorithms that could not find any solutions in  $PF_{true}$  is zero. So the metric does not distinguish the relative closeness. However, the solutions found by one algorithm may be much closer to  $PF_{true}$  than the other ones. In the case of utilizing only the second term in equation (1), the performance of two algorithms, of which one finds solutions that all of them are in  $PF_{true}$ , and the other one finds solutions of which all of them are close enough to  $PF_{true}$ , but none of them is member of  $PF_{true}$  is the same. For both of them the calculated value is one. By combining these two components, the above shortcomings can be addressed.

$WSGR = w_1 + w_2$  indicates that all the true Pareto optimal solutions are found by the MOEA, and  $WSGR=0$  means that none of the solutions in  $PF_{known}$  is close enough to the  $PF_{true}$  and thus the performance of the MOEA, in terms of the closeness, is not desirable. Since we are interested in how well an algorithm performs, in the case of two algorithms with  $WSGR=0$  we can conclude that the performance of both of them is not acceptable with respect to the closeness aspect. It does not really matter which one performs worse. However, our visualization method for showing closeness enables us to do a more detailed analysis on relative closeness between  $PF_{known}$  and  $PF_{true}$ .

##### 3) Visualization of the Closeness

We propose a visualization method, which simply and clearly illustrates the closeness relations between  $PF_{known}$  and  $PF_{true}$ . Our visualization not only shows the values of different closeness metrics and how these values are achieved, but also represents some information about the outliers and fluctuation in the distance values (if they exist). One of the disadvantages of the *GD* metric is that if there is a large fluctuation in the distance values or there exist some outliers, the metric may not reveal the true distance. However, using our visualization, these properties can be recognized easily.

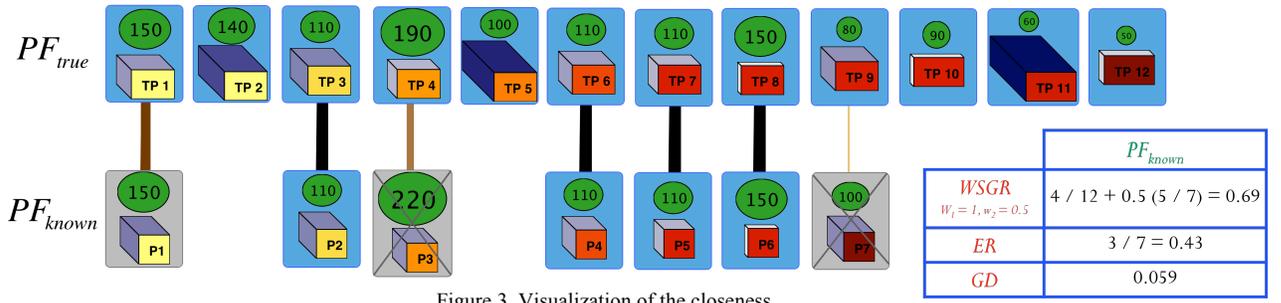


Figure 3. Visualization of the closeness

Fig. 3 represents the visualization of the closeness aspect for the Pareto optimal set shown in Fig. 1. The solutions in the true Pareto optimal set (or reference set if  $PF_{true}$  is not known) are shown in the first row and solutions of  $PF_{known}$  are shown in the second row. In the case of comparing the performance of different algorithms, for each of them, the solutions in its  $PF_{known}$  are drawn in a separate row (see Fig. 10). The background color of the true Pareto optimal solutions is blue. If a solution in  $PF_{known}$  is also a member of  $PF_{true}$  its background color is blue as well. Otherwise, a light grey background is used, and a cross is displayed at the dominated solution in  $PF_{known}$  to show that a solution in  $PF_{true}$  dominates this solution. However, if the distance of a dominated solution in  $PF_{known}$  is less than the defined threshold in  $WSGR$  metric, then the cross is not displayed to indicate that the solution is close enough.

Each solution in  $PF_{known}$  is connected to the nearest solution in  $PF_{true}$ , in which the Euclidian distance (in the objective space) between them is minimum. This minimum distance is the same as the distance measure used in calculating the  $GD$  metric. The color and thickness of each edge shows the distance between two connecting solutions. As the distance increases the edges become thinner and lighter. So the edges connecting true Pareto optimal solutions in  $PF_{known}$  to their corresponding solutions in  $PF_{true}$  are the thickest and darkest since the distance is zero. This edge visualization allows us to easily recognize the outliers and fluctuation in the distance values.

As can be seen in Fig. 3, four members of  $PF_{known}$  are also in  $PF_{true}$ . For these solutions, the background color is blue and their edges are the thickest and darkest (black color). The other three solutions in  $PF_{known}$  are dominated by solutions in  $PF_{true}$ . Each one is connected to the nearest solution in  $PF_{true}$  that dominates it. The solution P1 in  $PF_{known}$  is close enough to  $PF_{true}$  according to the distance threshold and therefore the cross is not drawn. The solution P7 in  $PF_{known}$  is an outlier since its distance value is too far from the other distance values. Its edge color and thickness differ a lot in comparison with other edges. The visualization of the closeness aspect has following advantages:

- It enables us to easily find out which solutions in  $PF_{known}$  are dominated by which solutions in  $PF_{true}$ .
- It allows us to understand which true Pareto optimal solutions have been found by the MOEA.
- Since for each solution its objective values are shown as well, it is easy to find out in which parts of the objective space the obtained solutions are near to the

true Pareto front and in which parts they are far away. For example, from Fig. 3 we can see that none of the low cost (less than 100) Pareto optimal solutions in  $PF_{true}$  have been found.

- It is possible to easily recognize outliers.
- It clearly shows the fluctuation in the distance values.

### B. Diversity Metrics

The Diversity metrics can be divided into two groups: 1) metrics evaluating the spread of solutions along the Pareto front and 2) metrics evaluating the distribution of solutions. In this section, we explain one metric for each group.

#### 1) $\nabla$ -metric (for measuring spread)

The  $\nabla$ -metric [3] calculates the volume of a hyperbox formed by the extreme objective values observed in the Pareto optimal set:

$$\nabla = \prod_{m=1}^M (f_m^{\max} - f_m^{\min}) \quad (2)$$

Where  $M$  is the number of objectives,  $f_m^{\max}$  and  $f_m^{\min}$  are the maximum and respectively minimum values of the  $m^{\text{th}}$  objective in the Pareto optimal set. A bigger value spans a larger portion and therefore is better. This metric does not reveal the exact distribution of intermediate solutions, so we have to use another metric for evaluating the distribution.

**Visualization of the spread:** For each objective, a horizontal axis from 0 to 1 is drawn and colored like the color-coding technique used for showing that particular objective in the multi-objective DSE visualization. For example, in our case, colors from yellow to red are used for representing the performance of design points (in terms of processing time); therefore, this color scheme is used for coloring the corresponding axis in the spread visualization. However, for representing the cost of design points, instead of the color, the size of the node in the visualization tree is used. Thus, the cost axis in our spread visualization is colored with a solid color, which is the same as the cost nodes in the DSE tree.

For each axis, the range between the minimum and maximum value of the corresponding objective in the Pareto optimal set is determined and shown above the (color-coded)

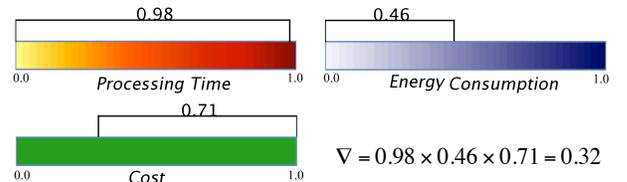


Figure 4. Visualization of the spread

objective axis. Fig. 4 represents the visualization of the spread for the Pareto optimal set shown in Fig. 1. As can be seen in Fig. 4, the Pareto optimal set has an almost perfect extent in processing time and its spread in terms of cost is fairly good. However, only a small portion of the energy consumption is covered by the solutions in this Pareto optimal set. As a result, the value of  $\nabla$ -metric is relatively small and indicates that this set does not have a good extent. The spread visualization has the following advantages:

- It is not limited to the number of objectives. Only one axis is added for each objective.
- For each objective, it can easily be seen that solutions are located in which part of the objective space. Both in terms of spread width (difference between the minimum and maximum) and objective values (the value of the minimum and maximum). For example, in Fig. 4 the spread width of the energy consumption is small. However, the covered portion is located near the optimum part of this objective space.

2)  $\sigma_{mst}$  - metric (for measuring distribution)

In [6], the well-known Schott's Spacing metric (SS) is proposed which tries to assess how evenly the non-dominated solutions are distributed. It is based on computing the shortest distance between solutions. The drawback of this metric is that in the case that solutions are clustered in small groups along the Pareto front, the distance between the groups are not considered since only the shortest distances are computed. More specifically, it considers some information more than once (distance between solutions inside a group) while ignoring some useful distribution information such as the distance between the groups. Therefore, for measuring the distribution of solutions in a Pareto optimal set, we propose a new metric,  $\sigma_{mst}$ , which is the standard deviation of the edges' weights in the Minimum Spanning Tree (MST) generated by Pareto optimal solutions. The weights of the tree edges are the Euclidian distances (in the objective space) between solutions.

A minimum spanning tree is a subgraph of a weighed graph, which is a tree and contains all of the graph's nodes and a subset of its edges, such that all nodes are connected and the total weight of the edges is minimal. We define the procedure of constructing an MST from a Pareto optimal set as follows:

Algorithm 1: constructing MST from a Pareto set
1. Compute the Euclidian distance (in the objective space) between any two solutions in the Pareto optimal set
2. Create a fully connected weighted graph (G) in such a way that each solution in the Pareto optimal set indicates a node in the graph G and the edge weight between two nodes is the Euclidian distance between the corresponding solutions
3. Generate a minimum spanning tree for the graph G (Prim's algorithm):
(a) Let MST be an empty tree
(b) Select a random node in G and add it to MST
(c) While MST has fewer nodes than G do:
i. Find the smallest edge connecting a node in MST to a node in G-MST
ii. Add the corresponding edge and node to the MST

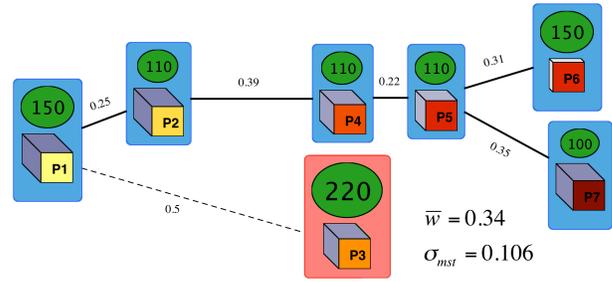


Figure 5. Visualization of the distribution

When the MST is made,  $\sigma_{mst}$  can be computed as follows:

$$\sigma_{mst} = \sqrt{\frac{1}{|E|-1} \sum_{i=1}^{|E|} (\bar{w} - w_i)^2} \quad (3)$$

Where  $|E|$  is the number of edges in the MST,  $w_i$  is the weight of the  $i^{th}$  edge and  $\bar{w}$  is the average weight of the edges in the MST. The  $\sigma_{mst}$  metric measures the standard deviation of the edges' weights in the MST. The edges' weights denote the minimum distances between connecting solutions. Therefore, a smaller value indicates that the distribution of solutions is closer to the uniform distribution and thus is better.

**Visualization of the distribution:** We propose a visualization method, which clearly shows how discovered Pareto optimal solutions are distributed in the objective space. For visualizing the distribution, the constructed MST is drawn in such a way that the length of the edge between two nodes (solutions) represents the edge weight (the Euclidian distance between two solutions). A longer edge implies a larger distance. Therefore, if in a Pareto optimal set all the edges have almost the same length, then this means that the solutions are distributed (nearly) uniformly. Each node in the MST, which indicates a solution in the Pareto optimal set, is drawn in such a way that its objective values can be seen as it is shown in Fig 2.

For better viewing and analyzing the distribution of solutions, it is possible to cluster the nodes (solutions) in the MST according to their edge weights (distance). If the distance between a solution and its parent is less than a certain threshold (determined by the user) then it is in the same cluster as its parent. Otherwise, it becomes a member of a new cluster. The solutions in the same cluster have the same background color. A Pareto optimal set with a smaller number of clusters has a better distribution. For a better view, the edges connecting two different clusters are drawn by dashed lines.

Fig. 5 shows the visual representation of the distribution for the Pareto optimal set shown in Fig. 1. In this figure,  $\bar{w} + \sigma_{mst}$  is chosen as a threshold for clustering the solutions. Here, the solutions are distributed into two clusters, since there are two different background colors. Excluding the solution labeled by "P3", all the other solutions are in the same cluster, which means that the distance between them is less than the threshold and therefore their distribution is fairly uniform. However, the large distance between P3 and its parent causes the value of the  $\sigma_{mst}$  metric to become slightly bigger and thus indicates that the distribution of solutions is suboptimal. Using distribution visualization has the following advantages:

- For each two consecutive solutions, besides the distance value, the amount of difference between their objective values can be seen. Therefore, it is easy to find out which objective value(s) have a high impact on the distance value. For example, in Fig. 5, solutions P2 and P4 have exactly the same cost and their energy consumption is almost the same too. But their difference in processing time is significant. Thus this objective has the highest impact on the distance value.
- It can be easily used for problems with more than three objectives because objective values are shown by node attributes (see Fig. 2).
- By clustering the solutions, it is easy to understand which parts of the objective space are properly covered by well-distributed Pareto optimal solutions and in which parts the coverage and distribution is poor.

### C. Combined Metrics

The combined metrics evaluate both multi-objective optimization goals in a combined sense. They provide a measure of closeness as well as diversity in an implicit manner.

#### 1) Hypervolume

This metric [1] measures the hypervolume of the objective space covered by members of a Pareto optimal set and a reference point. The hypervolume represents the size of the region dominated by the solutions in the Pareto optimal set. The reference point can simply be found by constructing a vector of worst objective values. The hypervolume metric is interesting because it is sensitive to the closeness of solutions to the true Pareto optimal set as well as the distribution of solutions across the objective space. The hypervolume value is calculated by summing the volume of hyper-rectangles constructing the hypervolume. A Pareto optimal set with a large value for the hypervolume is desirable.

**Visualization of the Hypervolume:** As we mentioned before, one of the main challenges of proposing a visualization method for a performance metric is that it can be used for problems with more than three objectives. The visualization of the hypervolume metric, which is typically used in literature, is limited to three objectives. However, we propose a new visualization method, which is not restricted to the number of objectives. To do that, we divide the  $m$ -dimensional objective space to  $m-1$  two-dimensional spaces. One certain objective (determined by the user) is chosen as a base and is a member of all 2D spaces, while all the other objectives are in separate 2D spaces. For each two-dimensional space, an x-y graph is drawn in which the x-axis shows the base objective and therefore is the same in all graphs and the y-axis represents the other objective. In each x-y graph the colored region shows the dominated part. As an example, consider a problem with three objectives:  $f_1, f_2$  and  $f_3$ . We divide this 3-dimensional objective space to two 2D spaces and represent them with two x-y graphs:  $f_2$  versus  $f_1$  and  $f_3$  versus  $f_1$ . Here,  $f_1$  is chosen as the base objective.

By dividing the objective space to two-dimensional spaces, each  $m$ -dimensional hyper-rectangle in the hypervolume region, is divided into  $m-1$  rectangles (two dimensional hyper-

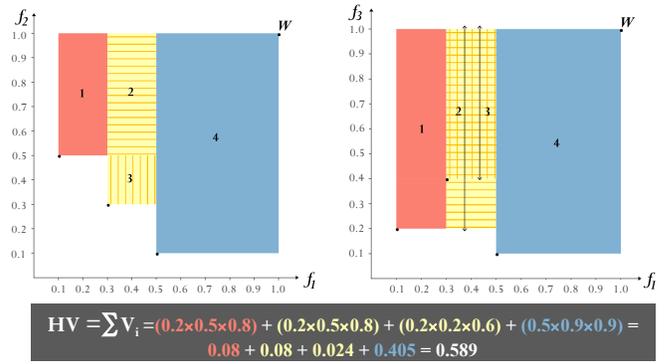


Figure 6. Visualization of the Hypervolume metric

rectangle). Each rectangle is drawn in its corresponding x-y graph. The base objective (x-axis) constructs the width of the all  $m-1$  rectangles and each of the other objectives constructs the height of its corresponding rectangle (y-axis). To distinguish the divided rectangles of a particular hyper-rectangle in different x-y graphs, they are colored with the same color in all graphs. If two or more hyper-rectangles have overlap in some spaces, textures are used. Each hyper-rectangle has a specific texture. Therefore, the overlapping areas contain several textures.

Fig. 6 shows an example of hypervolume visualization for a Pareto optimal set containing three solutions:  $S_1 = \{0.1, 0.5, 0.2\}$ ,  $S_2 = \{0.3, 0.3, 0.4\}$ ,  $S_3 = \{0.5, 0.1, 0.1\}$ . Small black circles in the graphs display the Pareto optimal solutions. The reference point ( $W$ ) is  $\{1, 1, 1\}$ . As can be seen in this figure the hypervolume region consists of four hyper-rectangles, which are denoted by numbers from 1 to 4. For each hyper-rectangle, the values of its axis in each dimension and its volume are written at the bottom of the visualization with the same color as its corresponding rectangles in x-y graphs. For instance, in Fig. 6, the volume of the hyper-rectangle denoted as 1 is  $0.2 \times 0.5 \times 0.8 = 0.08$

The second and third hyper-rectangles have overlap in the  $f_3$  versus  $f_1$  graph. Therefore, textures are used. The texture of the second hyper-rectangle is horizontal lines and the texture of the third hyper-rectangle is vertical lines. Thus, the overlapping area has both textures. The hypervolume value is calculated by summing the volume of hyper-rectangles, as shown in Fig. 6. This type of visualization enables us to clearly see the dominating area of each objective surface separately. Since the x-axis is the same in all surfaces, the comparison between them can be made easily. For instance, in the example shown in Fig. 6, we can see that the size of the dominating region in the  $f_3$ - $f_1$  surface is bigger than the dominating area in the  $f_2$ - $f_1$  surface.

### D. Dynamic Metrics

Previous metrics described in this section are used to assess the quality of the obtained Pareto optimal set at the end of the MOEA execution. However, dynamic performance metrics show how an MOEA achieves the final quality during its execution. They also illustrate how the quality of solutions is varying during MOEA execution. Such information provides insight in the working of the algorithm and allows detailed evaluation of the strengths and weaknesses of the algorithm. Moreover, these metrics are especially useful when the quality

of the end results of compared algorithms do not differ significantly; however, the way that these results are achieved may provide valuable information such as how fast a certain search algorithm converges to the Pareto optimal set. The output of dynamic metrics is not a single value. Instead, a set of values is computed that shows the dynamic behavior of the algorithm. In VMODEX, three dynamic metrics are available: 1) dynamic hypervolume 2) dynamic closeness 3) the dynamic process of finding final Pareto optimal solutions.

It should be mentioned that the examples depicted here to illustrate the dynamic metrics, are captured from the case study results explained in the next section. In our case study, the performance of two MOEAs using different repair mechanisms (intensive-repair and moderate-repair) is compared with various quality metrics described in this paper. To save space, the results of the comparison with respect to the dynamic metrics are shown in this section.

### 1) Dynamic Hypervolume (DH)

This metric shows how the value of the hypervolume metric evolves over the evaluations. If an algorithm can reach to a desirable hypervolume with fewer evaluations, this means its optimization speed is faster and therefore its performance is better. As we mentioned before, the hypervolume metric measures both the closeness and diversity of obtained solutions in an implicit manner. Therefore, this metric provides an overall evaluation of how the MOEA finds better approximations of Pareto optimal solutions during its execution. For showing this metric, the hypervolume value is drawn versus the number of evaluations. Fig. 7 represents this metric for our case study. The hypervolume value for zero evaluations indicates the hypervolume for the initial population, which has been generated randomly. As can be seen in Fig. 7, both algorithms reach to almost the same hypervolume at the end of the execution. However, the intensive-repair approach attains this value within 5500 evaluations whereas moderate-repair reaches this hypervolume after 9000 evaluations. Furthermore, we can see that intensive-repair has a significant improvement after approximately 1000 evaluations, after which the hypervolume only slightly increases in subsequent evaluations. However, the behavior of moderate-repair is quite different as it has a much smoother improvement.

### 2) Dynamic Closeness (DC)

This metric is similar to the dynamic hypervolume, except that one of the closeness metrics is used. Therefore, it shows how fast the found Pareto front converges towards the true Pareto front. In Fig. 8, the generational distance is used for evaluating the dynamic closeness of the two aforementioned MOEAs. Since, the true Pareto front is not known in our case study, a reference set is used. It can be seen that at the end of

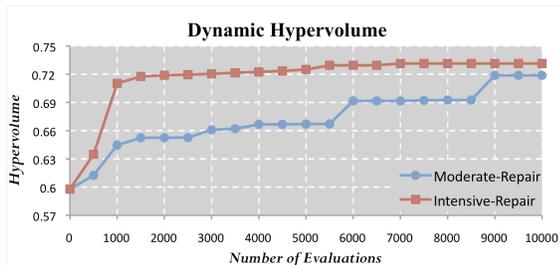


Figure 7. Comparing performance with DH metric

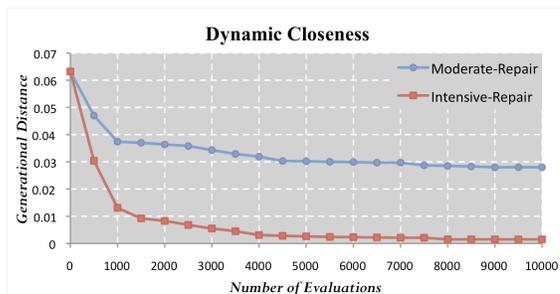


Figure 8. Comparing performance with DC metric

the execution as well as during the evaluations, the solutions found by intensive-repair are much closer to the reference set. Moreover, the most significant improvement for both algorithms is achieved after approximately 500 and 1000 evaluations. However, the amount of this improvement for intensive-repair is more significant than for moderate-repair. After 1000 evaluations, both algorithms only slightly improve.

### 3) Dynamic process of Finding final Pareto Optimal Solutions (DFPOS)

We propose a new dynamic metric, called DFPOS, which shows the progress of the algorithm in finding the final Pareto optimal solutions. A final Pareto optimal solution is a solution that will not be dominated by any other solution during the MOEA execution. This metric represents a set of generation numbers in which a *final* Pareto optimal solution is found for the *first* time. Thus, it is easy to understand in which generations the algorithm has found a new final Pareto optimal solution. Note that this metric can be calculated, when the MOEA execution is terminated. The benefit of this metric is illustrated by our case study results later in this section.

We also propose a visualization method for showing the results of the DFPOS metric, which is shown in Fig. 9. For each generation a line is drawn. If within a generation a new final Pareto optimal solution is found, then the color of the line representing that generation is blue. Otherwise, a gray line is used. For better viewing, these lines are shown as radiuses of a circle in which the angle between any two lines is the same. In Fig. 9, moderate-repair has found some new Pareto optimal solutions in the last generations. So, it is more likely to find other Pareto optimal solutions by running this algorithm for more generations and improve the quality of the obtained solutions. However, in intensive-repair there is no new Pareto optimal solution after 54<sup>th</sup> generation. Therefore, by evaluating more solutions, it will probably not find a new Pareto optimal solution and the quality of the results will not improve. Thus, we can reduce the number of generations in order to decrease the computational cost.

## V. CASE STUDY

There are various multi-objective evolutionary optimizers known from literature such as SPEA2, NSGA-II, etc. Thus, for solving a multi-objective problem, it is important to choose an

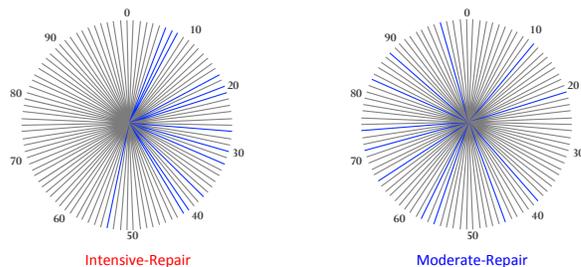


Figure 9. Comparing performance with DFPOS metric

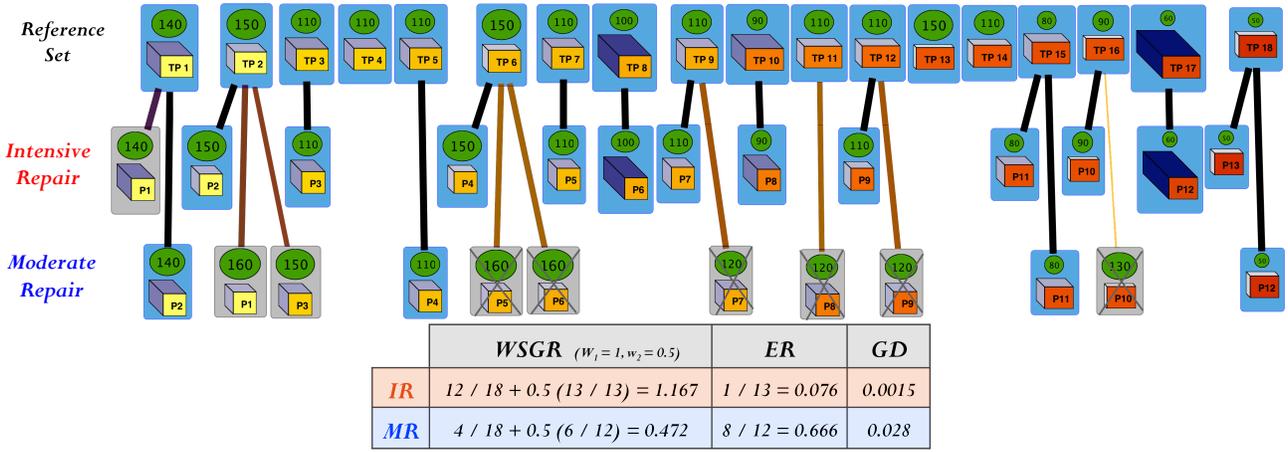


Figure 10. Comparing performance with *closeness* metrics

appropriate optimization technique that provides the best Pareto optimal set. Quality metrics can be used to compare the outcomes of different multi-objective optimizers in a quantitative manner. The purpose of these quality metrics is to reveal the strengths and weaknesses of each optimization approach and identify the most promising techniques. As we mentioned in the introduction, the performance of MOEAs is highly dependent on their parameter settings. So, the quality metrics can be used for investigating the effect of different parameter settings for a single algorithm as well.

In our case study, we map a motion-JPEG (M-JPEG) encoder to an MPSoC platform architecture consisting of a general-purpose microprocessor (mP), a microcontroller (mC), an application specific instruction processor (ASIP), two Application Specific Integrated Circuits (ASICs), one SRAM and two DRAMs. For solving the mapping decision problem, the SPEA2 [3] multi-objective evolutionary optimizer is used to achieve a set of the best alternative mapping decisions under three criteria: the processing time, energy consumption and cost of the architecture. Due to the randomness in MOEAs (in initialization, recombination and mutation steps), they are prone to violating the problem constraints and producing infeasible mappings. Therefore, we use a repair mechanism to handle constraint violations. In order to examine the effect of repair usage on the MOEA performance, we utilize two different repair strategies. The quality of the Pareto optimal set found by each strategy is measured and visualized by VMODEX with a variety of metrics described in Section IV.

With respect to repair, we apply two strategies [3]. (1) A *moderate-repair* strategy (*MR*) in which at the end of each generation all invalid individuals are repaired. This allows infeasible individuals to enter the mutation step, and therefore, may help to explore new feasible areas over unfeasible solutions. This is especially important for problems in which the feasible regions may not be connected. (2) An *intensive-repair* strategy (*IR*) in which all invalid individuals are repaired immediately after creation. Hence, all individuals entering

mutation are feasible. For each repair strategy, we run SPEA2 for 100 generations with 100 individuals per population. In the following, we compare the results of these repair strategies with different quality metrics.

**Closeness metrics:** Since in our case study the true Pareto front is not known, we use a reference set to evaluate the closeness of discovered Pareto optimal solutions. For estimating the reference Pareto front, we first combined the results of running SPEA2 for 1000 generations for both repair strategies. The results of our case study are also combined. Then we removed the dominated solutions from the combined set. As a result, we found 18 Pareto optimal solutions that are used as an estimation of the true Pareto optimal solutions. The visualization of closeness metrics for both repair strategies is shown in Fig. 10. As can be seen in this figure, the *IR* strategy has found 66% (12 out of 18) of solutions in the reference set. The *MR* has only found 22% (4 out of 18) of the reference Pareto optimal solutions. From the visualization we can see that all the cheap solutions (less than 100) in the reference set are found by the *IR* strategy and none of the solutions in the reference set with high-energy consumption are found by the *MR*. With respect to the *ER* metric, except one solution (P1), all other solutions of *IR* are in the reference set. However, P1 is close enough to the reference set with respect to the distance threshold. But, more than half of the solutions found by *MR* are dominated by the solutions in the reference set. Furthermore, solutions in *IR* are more than 18 times closer to the reference set than those in *MR* (according to the *GD* metric). Moreover, we can see that in the Pareto optimal set found by *MR* there is one outlier (P10). As a result, the *IR* performs much better than moderate-repair with respect to the all closeness metrics.

**V-Metric:** The visualization of the V-metric for each repair strategy is shown in Fig. 11. As can be seen in this figure, both repair strategies have almost the same extent in processing time and cost. However, the *IR* (red line) has a much better extent in energy consumption, and thus, in general performs better in terms of V-metric.

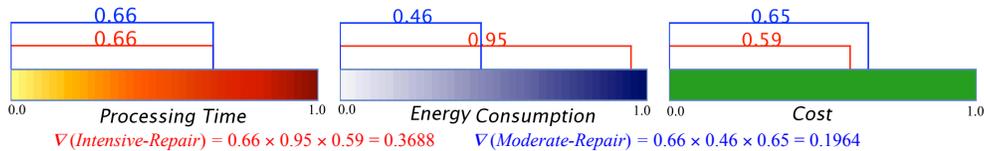


Figure 11. Comparing performance with V metrics

**$\sigma_{mst}$ -Metric:** Fig. 12 shows the visualization of the  $\sigma_{mst}$ -metric for both repair strategies. In this figure,  $\bar{w} + \sigma_{mst}$  of the reference set is chosen as a threshold for clustering the solutions. All the Pareto optimal solutions found by *MR* are in the same cluster which indicates that the distances between connecting solutions is less than the threshold and therefore the distribution is nearly uniform. However, the solutions of the *IR* are distributed into four clusters. Solutions with high-energy consumption (P8, P6, P12) are grouped in different clusters and all the other solutions are in the same cluster. Thus, we can conclude that in the part of the objective space containing high-energy consumption the distribution of solutions is poor and in the other parts the distribution is relatively uniform.

**Hypervolume Metric:** Fig. 13 represents the visual form of the hypervolume metric for both repair strategies. As can be seen in this figure, in both intensive and moderate repair, a large portion of objective space is dominated by Pareto optimal solutions. However, the size of the dominating area in the processing time-energy consumption surface is bigger than the processing time-cost surface.

The comparison of the performance between repair strategies in terms of dynamic metrics is discussed in the previous section. As a result, the moderate-repair performs better in only finding well-distributed solutions ( $\sigma_{mst}$ -metric) while for all the other quality aspects, the intensive-repair obtains more preferable Pareto optimal solutions.

## VI. CONCLUSION

In this paper, we have extended VMODEX to allow algorithm developers to evaluate and compare the performance of different MOEAs that are used for searching the design space. Several metrics (from literature and new ones) and their visualizations are provided to enable algorithm developers to assess the quality of the discovered Pareto optimal solutions from different perspectives. The visualization techniques can reveal some useful and interesting information, which has been hidden in the quantitative representation of a quality aspect. Therefore, by using VMODEX, algorithm developers can easily examine different optimization algorithms for exploring the design space and find the best one for their specific problems. Then, the results of the best algorithm are delivered to designers for further analysis. VMODEX is helpful for designers as well. They can use VMODEX [5] to gain insight into the landscape of the design space and understand how the design space is searched by a heuristic search algorithm.

## REFERENCES

- [1] Zitzler, E., Thiele, L.: Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study, in PPSN-V, 1998.
- [2] Van Veldhuizen, D.A.: Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations, PhD thesis, Graduate School of Engineering of the Air Force Institute of Technology, Air University, June 1999.
- [3] Erbas, C., Cerav-Erbas, S., Pimentel, A.D.: Multiobjective Optimization and Evolutionary Algorithms for the Application Mapping Problem in Multiprocessor System-on-Chip Design, IEEE Transactions on Evolutionary Computation, pp. 358-374, Vol. 10 (No. 3), June 2006.
- [4] Ang, K.H., Chong, G., Li, Y.: Visualization Technique for Analyzing Non-Dominated Set Comparison, In SEAL'02, Singapore, 2002.

- [5] Taghavi, T., Pimentel, A.D.: Visualization of multi-objective design space exploration for embedded systems, DSD 2010, Lille, France.
- [6] Schot, J.R., Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization, Master's thesis, 1995.

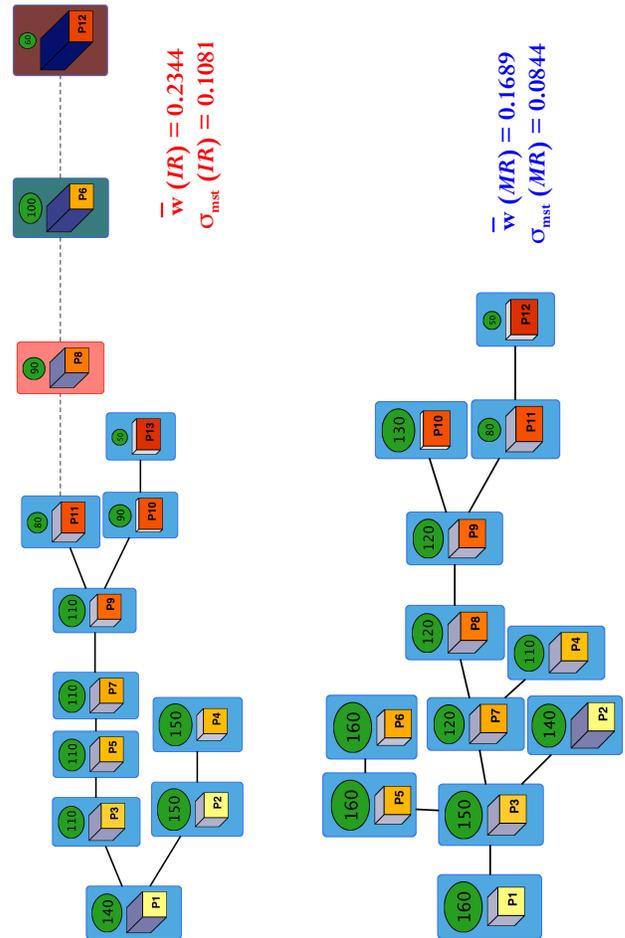


Figure 12. Comparing performance with  $\sigma_{mst}$  metrics

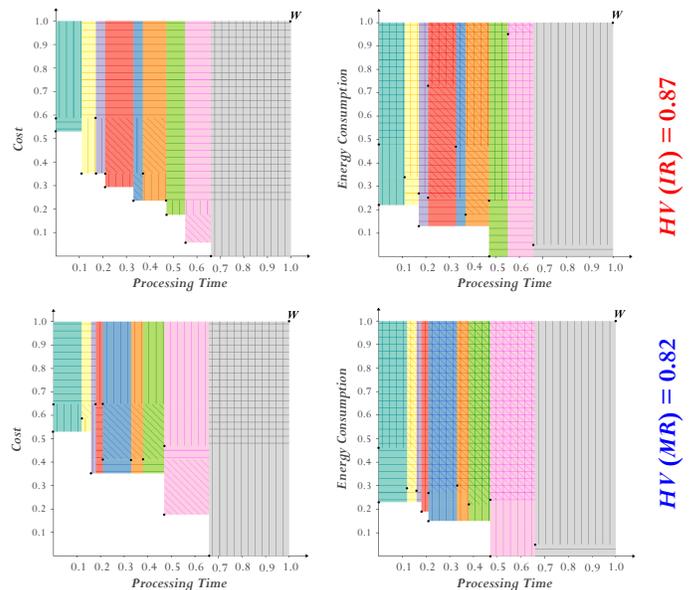


Figure 13. Comparing performance with *HV* metrics