UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

An interactive visualization tool for the analysis of multi-objective embedded systems design space exploration

Taghavi, T.; Pimentel, A.D.

Publication date
2011

Document Version
Final published version

Published in
Proceedings of the 3rd Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools (Rapido '11), in conjunction with HiPEAC'11

Link to publication

Citation for published version (APA):
Taghavi, T., & Pimentel, A. D. (2011). An interactive visualization tool for the analysis of multi-objective embedded systems design space exploration. In *Proceedings of the 3rd Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools (Rapido '11), in conjunction with HiPEAC'11* (pp. 1-10). HIPEAC. http://staff.science.uva.nl/~andy/artemis/RAPIDO4.pdf

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

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

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:09 Mar 2023

An Interactive Visualization Tool for the Analysis of Multi-Objective Embedded Systems Design Space Exploration

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—**The design of today's embedded systems involves a complex Design Space Exploration (DSE) process. Typically, multiple and conflicting criteria (objectives) should be optimized simultaneously such as performance, power, cost, etc. Usually, Multi-Objective Evolutionary Algorithms (MOEAs) are used to explore a large design space with a finite number of design point evaluations, providing the designer a set of tradable solutions with respect to the design criteria.**

Analyzing how such evolutionary algorithms searched the design space, understanding the characteristics of the optimum design points, the effect of design parameters on each objective and the relationships between different criteria is of invaluable importance to the designer. To this end, this paper proposes a novel interactive visualization tool, VMODEX (Visualization of Multi-Objective Design spacE eXploration), to realize the search dynamics of a MOEA and to visualize where the optimum design points are located in the design space and what objective values they have. In our tool, we provide several interactive capabilities, which enable designers to look at the exploration data from different perspectives and provide better analysis of the search results.

Keywords—Design space exploration, embedded systems, multi-objective evolutionary algorithms, visualization

I. 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 heavily 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 conflicting criteria need to be considered simultaneously such as performance, power, cost, etc. Therefore, there exist no 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 solutions provide the designer trade-offs between the design objectives.

In order to find a Pareto optimal set, the designer should ideally evaluate and compare every single point in the design space. However, such an exhaustive search quickly becomes infeasible, as the design space grows exponentially with the size of the application(s) and the number of possible architecture components.

In general, to trim down an exponential design space into a finite set of points, which are more interesting (or superior) with respect to design criteria, design space pruning can be used. In [1], e.g., the mapping decision problem 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. 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. Such analysis is, however, essential to the designer as it provides insight into the "landscape" of the design space (e.g., indicating which design parameters are more important than others).

To address these problems, we develop a novel interactive visualization tool, VMODEX, to understand how an evolutionary algorithm, such as presented in [1], searches the design space, where the optimum design points are

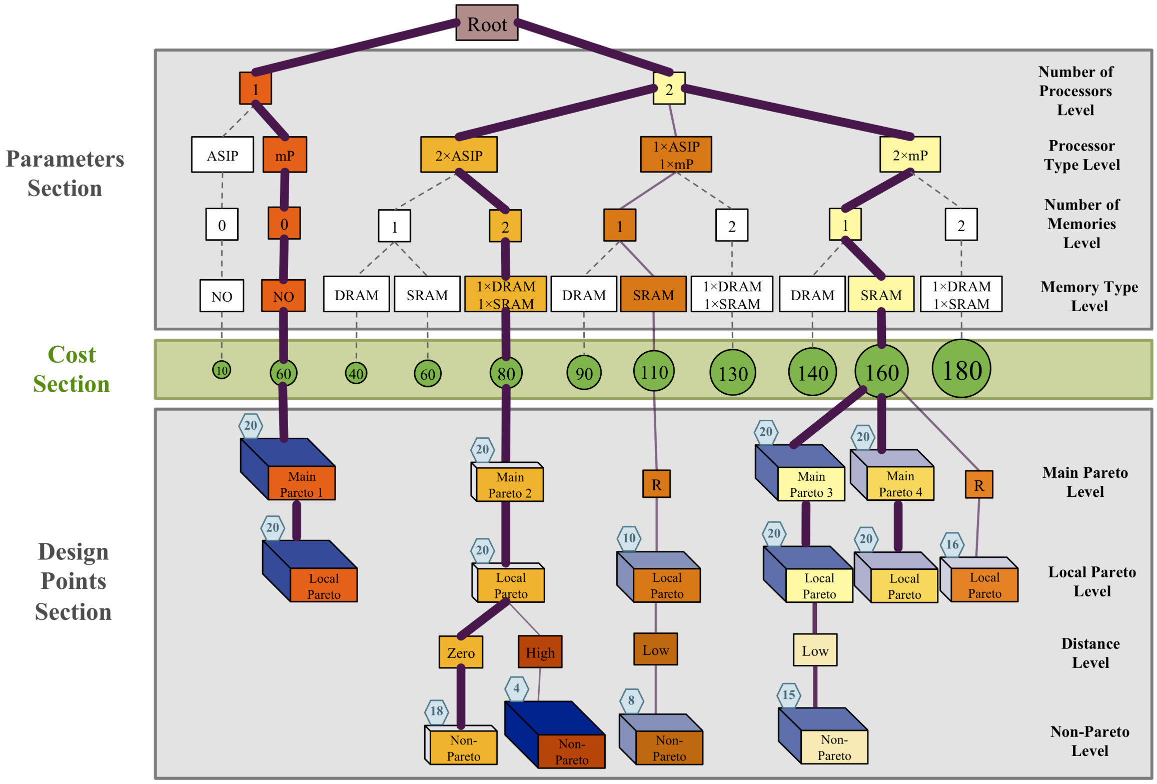

Fig. 1. Modeling the design space as a tree

located, how design parameters influence each objective and which clarifies the relationship among multiple objectives. In this respect, we visualize the design space as a tree in which both design parameters and objectives are shown in a single view.

The rest of the paper is organized as follows. Section II describes related work. Section III introduces techniques we have provided for visualizing multi-objective design space exploration. Section IV presents a case study with a Motion-JPEG encoder application to illustrate the benefits of using visualization in the design space exploration process. Finally, section V concludes the paper.

II. Related Work

In the field of computer architecture simulation, and especially in the area of system-level design space exploration, little research has been undertaken on visualization of simulation results in exploring alternative architectural solutions. Most of the visualization work in this area focuses on educational purposes (e.g., [2], [3]), or only provides some basic support for the visualization of simulation results in the form of 2D / 3D graphs.

The work presented in [4] provides advanced and generic visualization support, but tries to do so for a wide range of computer system related information which may not necessarily be applicable to computer architecture simulations and in particular to design space exploration, with its own domain-specific requirements.

In [5], an interactive visual tool is presented to visualize the results from system-level DSE experiments. The simulation results are visualized using a coordinated, multiple-view approach, which enables users to understand the information through different perspectives. But this tool does not provide any insight in the searching process as performed by e.g. a MOEA. For example, there is no way to find out which parts of the design space are not searched at all.

There are only a few research efforts addressing the visualization of MOEAs. Most visualization approaches simply use standard visual representations such as bar charts, line graphs, scatter plots, etc. [6] or they use 2D or 3D plots in which either variables or objectives are shown. Although such diagrams are useful to understand the overall properties of the explored solutions, they are limited to the three dimensions and do not provide detailed analysis of the search results. However, VMODEX enables designers to easily visualize more than three dimension problems as well as to see both the design parameters and objectives in a single view. Furthermore, it provides insight into the "landscape" of the multi-objective optimization process.

This paper proposes an extension to our previous work [7], in which we add some more features in VMODEX that enable designer to analyze the data and explore the search result from different perspectives in order to find the hidden important properties. Several filtering capabilities based on various criteria are provided, which help designers to define their preferences (from several point of views) and focus on only the design points that satisfy the preference conditions and find out the similarities between them. Furthermore, for each design point, an extra property (rather than the objective values and design parameters) is considered that is the generation numbers in which a solution is evaluated. Based on this property, different approaches are provided to explore the data from this aspect, such as visualizing the generation numbers, edge visualization based on the generation numbers, filtering, etc. Moreover, a more detailed case study is presented in which we analyze the exploration results from different perspectives and at multiple levels of abstraction. This case study shows how VMODEX provides designers a very powerful and rapid analysis of DSE data.

III. Multi-Objective Visualization

A. *Modeling the Design Space as a Tree*

As it is conceptually shown in Fig. 1, we model the design space as a tree. The tree has three sections: the parameters section, cost section and design points section.

Parameters section: In this section, each level shows one parameter of the design space, such as the number of processors in the MPSoC platform. So, the number of levels in this section is equal to the total number of parameters in the design space. For example, in the tree illustrated in Fig. 1 the design space has four parameters: number of processors, processor type, number of memories and memory type. In this example, the platform architecture consists of two Application Specific Instruction Processor (ASIPs), two microprocessors (mPs), one Static RAM (SRAM) and one Dynamic RAM (DRAM).

By modeling the design space as a tree, there is no limitation on the number of design variables as each parameter is located at one level of the tree. Therefore, we can easily visualize multivariate data.

Design points section: This section includes the design points searched by the MOEA. Here, a design point is defined as a specific instance of the architecture platform as well as a task and communication mapping. Each point is shown as a node, which is a child of its corresponding architecture. Design points are distributed in three levels: main Pareto, local Pareto and non-Pareto.

The main Pareto level shows the global Pareto points found by the MOEA. The solutions at this level are better than all other solutions in the entire design space but they are non-dominated by each other. On the other hand, each point that is not part of the main Pareto set is dominated by at least one main Pareto point.

At the local Pareto level, the local Pareto points are shown. 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. However, in the entire design space, a design point might exist which dominates the local Pareto point. It is clear that all the main Pareto points are local Pareto points as well. However, not all the local Pareto points are main Pareto points and therefore we use a relation node at the main Pareto level to make a connection between them and the previous level. These nodes are labeled with "R" in Fig. 1.

All the other design points are placed at the non-Pareto level. Each one becomes a child of a local Pareto point that dominates it. If a design point is dominated by more than one local Pareto point, we calculate the Euclidean distance (in the objective space) between the dominated point and each dominating local Pareto point and the design point becomes the child of the local Pareto point with the smallest distance. A smaller distance means that the points are more similar according to the objectives.

For easier interpretation and better analysis of the design points, the children of a local Pareto point are categorized into three groups according to their Euclidian distance from their parent. The solutions, which are equivalent to the local Pareto point with respect to all objectives, are put under the "zero" distance node. If the distance between a solution and its corresponding local Pareto point is more than a certain threshold (determined by the designer), it becomes a child of a "High" distance node, otherwise it becomes a child of a "Low" distance node.

B. *Showing Objectives in the Tree*

In this paper, we consider three objectives: processing time, energy consumption (i.e., power consumption times processing time) and architecture cost. The cost of each design point is dependent on the architectural components forming it. So, all solutions with the same architecture have the same cost. After the parameters section, the architecture cost can be computed since all components are known. Therefore, we add an extra section (see Fig. 1) between the parameters and design points sections, which is called the cost section and shows the costs of the different architectures. Since 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 other two objectives are dependent on the mapping and are therefore shown in a design point node. The size and color of the third dimension of a design point node 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 node 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. The color legends for processing time and energy consumption are shown in Fig. 2.

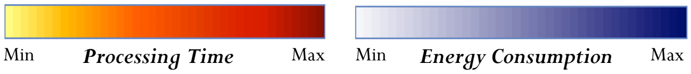

Fig. 2. Color legends

VMODEX can easily be extended to show more than three objectives. Each node has some attributes like shape, orientation, size, color, transparency, texture, border, etc. Each attribute can be assigned to one objective. In this paper, only color and size are used to show objectives.

Parameter nodes, however, do not represent single design

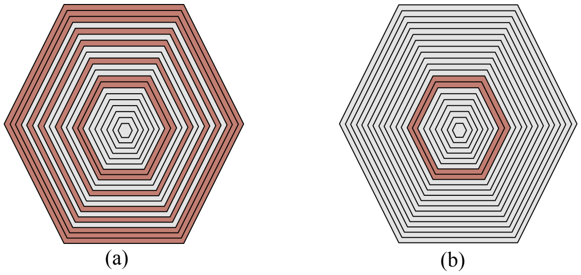

Fig. 3. Showing generation numbers

points and therefore do not have the direct notion of processing time or energy consumption. For this reason, there are some options to color the parameter nodes: based on the average, minimum, or maximum of either processing time or energy consumption of the design points in their sub trees. The color of parameter nodes that have no data node (i.e., do not have any DSE data) is white. In Fig. 1, the minimum processing time is chosen for coloring parameter nodes.

C. *Showing Generation Numbers*

In some cases, the designer wants to know what interesting design points are evaluated in which search generations. Therefore, we have developed a method in VMODEX that allows the designer to easily find out this kind of information. During the process of design space exploration using an MOEA, some design points that are near to the optimal solutions may be regenerated in different generations. In the DSE tree, for each design point, the number of the last generation in which it was generated is written inside a hexagon that is drawn at the upper left corner of the node (see Fig. 1). However, the designer is able to select a specific design point and see all the generation numbers in which the design point was evaluated. Fig. 3 shows the visualization approach for showing the generation numbers. For each generation, a hexagon is drawn. The size of the hexagons increases from the first to the last generations. To save space, these pentagons are nested together. If within a generation the selected design point is found, then the color of the pentagon representing that generation is red. Otherwise, a gray pentagon is drawn. Fig. 3(a) shows the situation that the corresponding design point is close to the optimum. Since it is regenerated in many search generations during the entire search. But in Fig. 3(b) the corresponding design point is generated in only two generations. This indicates that the design point is far from the optimal solutions and after a few generations it is not regenerated any more.

D. *Edge Visualization*

Edge visualization helps designers to navigate through the DSE tree and easily find more important parts. One feature of the design points is chosen as an importance factor and then the tree edges are visualized according to that factor, as follows:

- A minimum and maximum edge width is defined, and this range is linearly mapped against the range of importance factor values. Wider edges lead toward more important subtrees. The effect is a bit like a network of roads, in which the more important roads are wider.
- A specific color with various saturations is chosen. Similar to the line width, a linear mapping is done between the maximum and minimum saturation and the importance factor values range. Darker edges represent more important parts and lighter edges show less important subtrees.

In VMODEX, two importance factors are defined: minimum Euclidian distance and last generation number. In the following subsections we explain these factors.

1) *Minimum Euclidian Distance*

For each design point, the Euclidian distance (in the objective space) between that solution and the nearest main Pareto optimal point is calculated. A smaller distance indicates that the solution is closer to the optimal solutions and therefore is better. Thus, in the DSE tree, the edges in the path from the root to the main Pareto points are the thickest and darkest since the distance is zero (see Fig. 1). In this manner, just by looking at the DSE tree, the designer can easily determines which parts of the design space, contain optimal solutions and which parts contain poor solutions.

2) *Last Generation Number*

The number of the last generation in which a design point is evaluated can be considered as an importance factor. As the MOEA gradually converges to a set of Pareto optimal points, we expect better design points in the later generations. The edge visualization can show this progress. The edges with a higher generation number in their subtrees are thicker and darker. As a result, the paths from the root to the last generated data nodes are the darkest and thickest paths. As the importance factors are not applicable for edges that have no data node in their sub tree, these edges are shown by gray dashed lines.

E. *Filtering*

The filtering option in VMODEX allows designers to easily view only preferred design points. Therefore, they can focus on the more interesting design points. Four kinds of filtering are available based on the: 1) objective values, 2) design parameters, 3) distance from the main Pareto set and 4) generation numbers. The combinations of different filtering approaches are also provided.

1) *Filtering Based on the Objective Values*

In some cases, the designer wants to consider only design points with some specific objective values. The value of each objective is controlled by a range slider bar, in which the designer can set upper and lower limits on that objective. Design points with objective values inside the selected ranges are visible and the others become invisible. Therefore, the designer has the ability to easily view only design points with preferred objective values. There is an option to view all design points that fall within the filtering conditions or to only show local Pareto points or only main Pareto points. In Section IV, we explain two examples of this filtering, which are applied on our case study results.

2) Filtering Based on the Design Parameters

VMODEX allows designers to hide the parts of the design space, which are not being considered for further analysis, in order to make the tree smaller and pay more attention to the more interesting parts. For instance, the designer may wants to consider only the parts of the design space, which contain evaluated data. By using this filtering option, those subtrees in the DSE tree which are not visited by the searching algorithm, and therefore do not contain evaluated design points become invisible. This way, the designer can focus on the sub trees which are more important and can also easily see which parts of the tree are searched by the MOEA. Another example is the case in which the designer is not interested in design points with certain parameter values, like e.g. design points with more than four processors. Then it is possible to hide the subtrees in the DSE tree, which contain more than four processors. By hiding a subtree, a blue triangle is drawn at the bottom of its root node specifying that the children of the node are invisible. The size of the triangle represents the size of the sub tree. The bigger the triangle means the more nodes in the sub tree.

3) Filtering Based on the Distance from the Main Pareto Set

In the Multi-objective context, the goodness of a design point can be evaluated by the minimum Euclidian distance (in the objective space) from the Pareto optimal solutions. This distance measure can be useful for filtering design points to see only those solutions which have a distance from the Pareto optimal set that is less than a certain threshold. Thus, the designer can focus on only the design points, which are not far away from the Pareto optimal set. Hence, in VMODEX, we provide a filtering option based on this distance measure. The designer is able to define a threshold and then filters the design points to see only solutions, which are good enough according to their distance. Furthermore, as in VMODEX both the design parameters and objectives are shown in a single view, the designer can easily understand which parts of the design space contains solutions that are close to the main Pareto set.

4) Filtering Based on the Generation Numbers

Sometimes, the designer needs to consider only the design points generated in some specific generation(s). For example, showing only design points generated in the three last generations or comparing design points in the three first generations with the three last generations, and so on. Therefore, we provide a filter option based on the generation numbers. The user can simply add (or remove) generation numbers to the list of generation numbers. Only design points with their generation numbers in the list are visible and the others become invisible. The parameter nodes with at least one child in the generation list are still visible.

IV. Case Study

In this section, we present a case study with a real application to demonstrate the benefits of using visualization in the design space exploration process. In this case study, we map a Motion-JPEG (M-JPEG) encoder to an MP-SoC platform architecture consisting of a general-purpose microprocessor (mP), two Application Specific Instruction Processor (ASIPs), two Application Specific Integrated Circuits (ASICs), one SRAM and two DRAMs.

Using a multi-objective evolutionary optimizer [1], we intend to find a set of optimal design points (in terms of alternative architectural solutions and mappings) under three criteria: processing time, energy consumption and architecture cost. For this study, we run the MOEA for 100 generations with 50 individuals per population. Therefore, 5000 design points are searched by MOEA.

VMODEX allows designers to look at the evaluated data from different perspectives and analyze the search results at multiple levels of abstraction. In the following subsections, we analyze the M-JPEG case study with respect to the following issues: 1) General information about the searched design space, 2) The characteristics of the main Pareto optimal points, 3) Comparing local Pareto sets of different architectures, 4) Investigating the effect of different mappings for a certain architecture on the objectives, and 5) Filtering design points by their objective values.

A. General Information about the Searched Design Space

Fig. 4 shows a snapshot of the visualization of the M-JPEG case study. In this figure the parameter nodes that have no evaluated data are omitted and the minimum processing time is used for coloring the parameter nodes. Just by looking at the depicted tree, the designer can

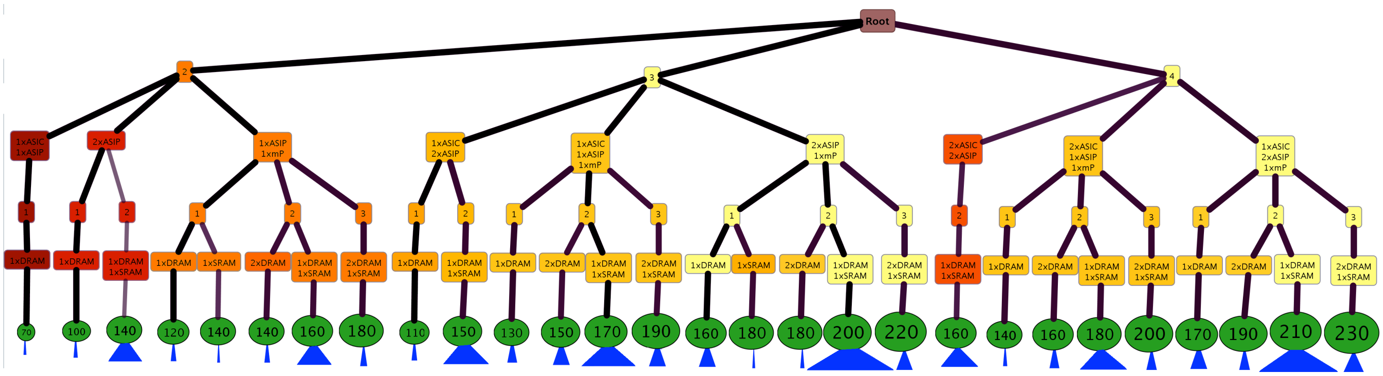

Fig. 4. Screenshot of the M-JPEG case study (parts of the design space having DSE data)

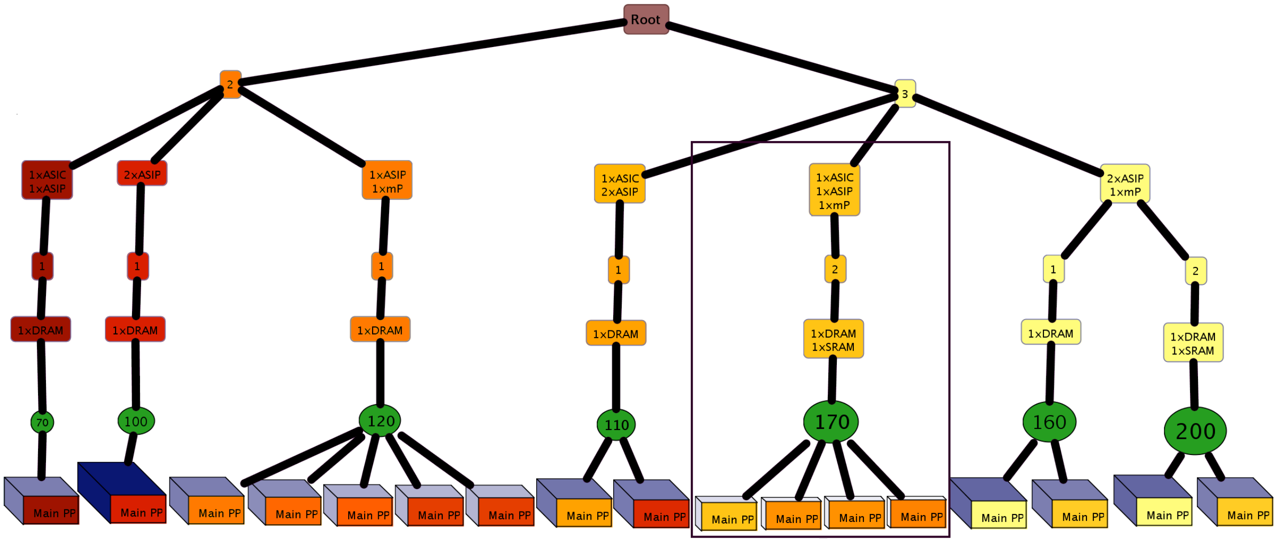

Fig. 5. Main Pareto points

immediately understand some general information about the design space searched by the MOEA. For example, it is obvious that there is no design point evaluated for single processor and five-processor architecture platforms since these parameter nodes are omitted from the DSE tree. Moreover, we can see that with two processors platforms, the MOEA cannot find a design point with a good processing time. This because the color of these parameter nodes are red which indicates that the minimum processing time of the design points in their subtrees is quite high. From the picture, it is also clear that most of the design points being searched by the MOEA contain two memories; one DRAM and one SRAM because all the biggest blue triangles have this memory configuration. Moreover, all design points with the minimum processing time include at least two ASIPs and one mP (node color is yellow)

B. *The Characteristics of the Main Pareto Points*

Fig. 5 shows the main Pareto points found by the MOEA. By looking at the picture, the designer can immediately recognize the characteristics of the main Pareto points, which are the best design points with respect to the design criteria. For example, in our case study, there is no main Pareto point with four processors. That means that with less processors (which is cheaper) the designer can get the same or better processing time and energy consumption. Therefore, using four processors is not appropriate for this application. Another interesting feature is that all the main Pareto points have at least one DRAM memory. A few of them have one SRAM besides the DRAM. Thus, using three memories or two DRAMs is not an appropriate solution in this case study. It can also be seen that all the main Pareto points with the lowest energy consumption have one ASIC, one ASIP and one mP in their underlying architectures (subtree indicated by "A"). So, by using VMODEX, the designer can easily find out which combinations of architectural components yield optimum design points.

C. *Comparing Local Pareto sets*

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. In each subspace, the Pareto optimal solutions found by a MOEA are called *local Pareto optimal solutions*. These solutions may or may not be main Pareto optimal. In the DSE tree, the local Pareto optimal solutions are located at a particular level, which is called the local Pareto level. Solutions at this level, which have the same parent at the cost level (and thus at the higher levels) are in the same subspace and therefore are members of the same local Pareto set. Fig. 6 shows an example of three local Pareto optimal sets (Q_1, Q_2, Q_3).

VMODEX enables designers to evaluate and compare different local Pareto optimal sets in different subspaces of the design space, and therefore, understand which parts of the design space contain solutions with a higher quality. For measuring the quality of a local Pareto optimal set, two distinct aspects should be considered:

1. Closeness to the main Pareto optimal set
2. Diversity in the objective values

The first aspect is essential in any optimization. Local Pareto optimal solutions that are not close to the main Pareto optimal set are not desirable. Closeness can be measured by two ways: 1) the number of solutions in a local Pareto optimal set which are also in the main Pareto optimal set and 2) the Euclidian distances (in the objective space) between the solutions in a local Pareto optimal set and the nearest member of the main Pareto set.

The second aspect (i.e. diversity) is also important. Since only a diverse set of solutions provides the designer flexibility in choosing the level of trade-offs between objectives. In the DSE tree, all the solutions in a local Pareto optimal set have the same architectural components and therefore they have the same architecture cost. Thus, the diversity can only be considered for the other two objectives: processing time and energy consumption.

Using VMODEX, both aspects of the quality of the local Pareto optimal sets can easily be evaluated. In terms of the

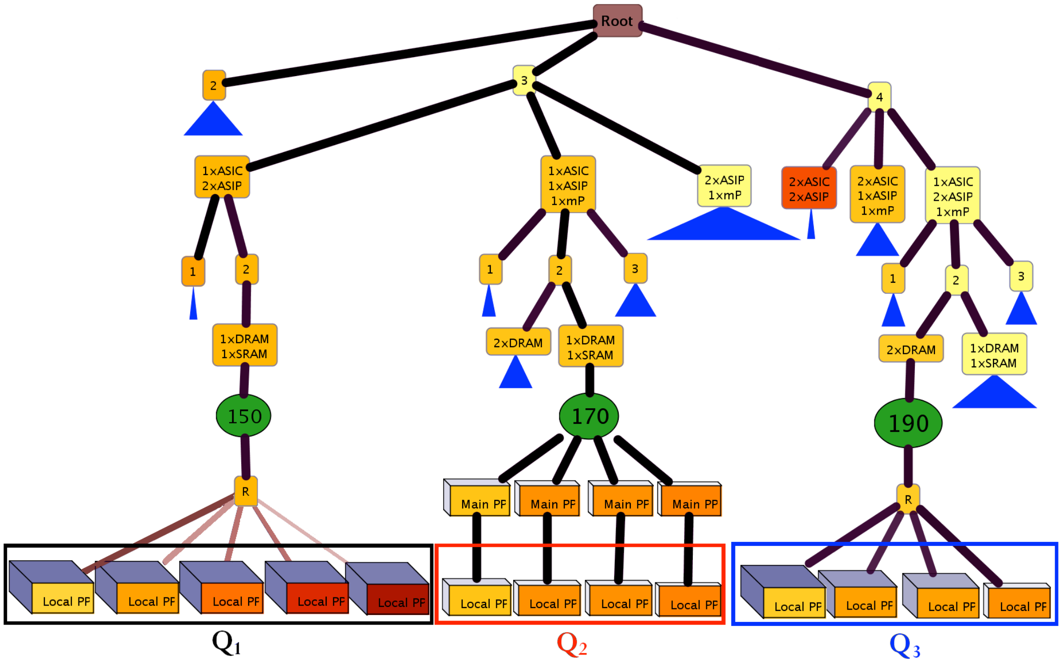

Fig. 6. Comparing local Pareto optimal sets

closeness, 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. Furthermore, it is easy to recognize those solutions of a local Pareto optimal set that are in the main Pareto set as well. In that case, their parents (in the tree) are main Pareto points. Otherwise they become children of a relation node.

In terms of the diversity, the variety of the nodes' colors represents the diversity in the processing time and the variety in the size and color of the third dimensions of nodes shows the diversity in the energy consumption.

In Fig. 6, the solutions in set Q_1 have a fairly high diversity in the processing time. Since the nodes' colors are varied from yellow to red. However, the diversity in energy consumption is quite low. All the design points have almost the same energy consumption. That means that for this specific architecture, depending on the mapping, we may get a good or a poor processing time. However, the energy consumption of the design points in this part of the design space is quite high. Therefore, this architecture is not appropriate for obtaining a low energy consumption. Regarding the closeness, none of the solutions in this set is member of the main Pareto set (their parent is a relation node) and also their distances from the main Pareto set is large (their edges are thin and light).

In the local Pareto optimal set Q_3, the diversity in energy consumption is fairly good but the diversity in the processing time is limited. With this specific architecture no main Pareto point can be found but its local Pareto optimal solutions are close to the main Pareto optimal set. The color and thickness of edges represent this feature.

In the set Q_2, the diversity in both processing time and energy consumption is poor. However, the solutions have relatively good processing time and energy consumption.

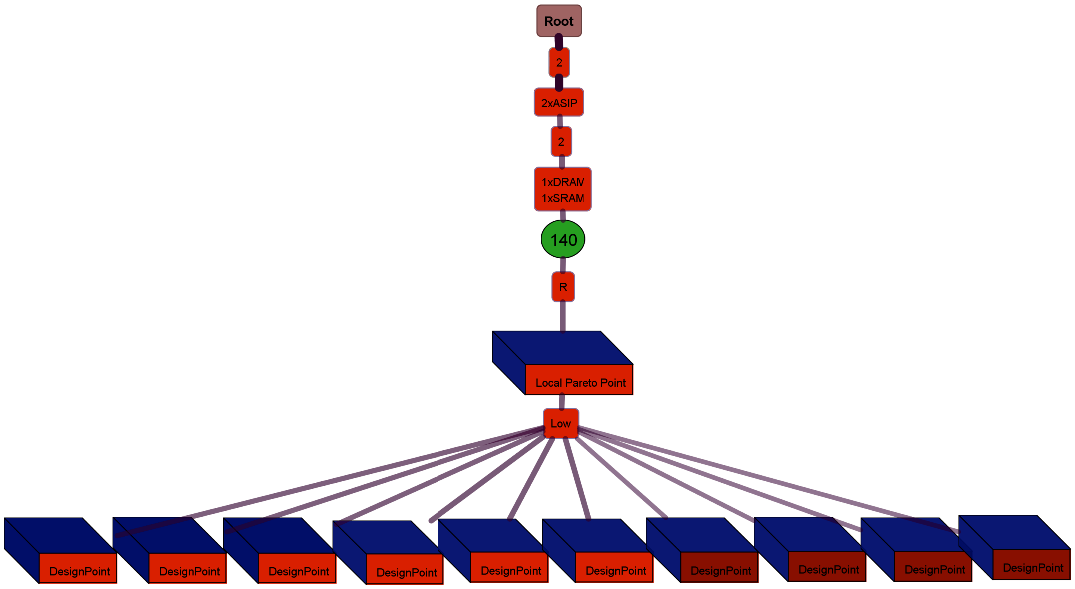

Fig. 7. All design points are extremely poor

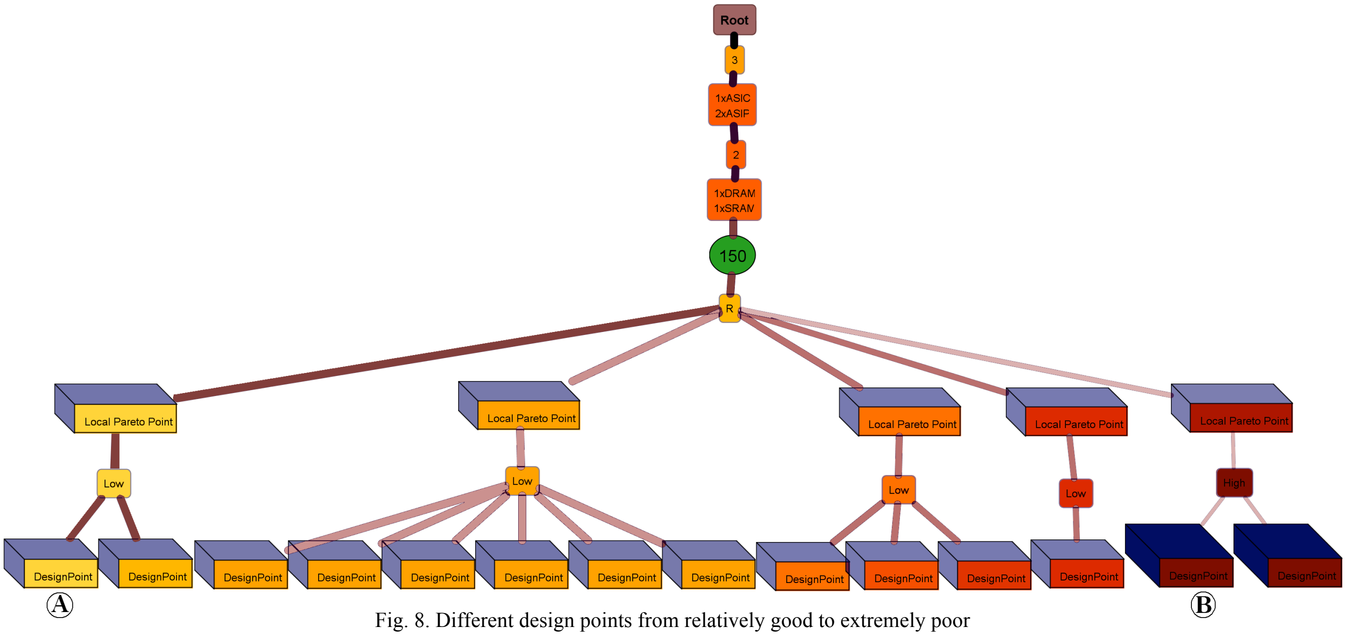

Fig. 8. Different design points from relatively good to extremely poor

Furthermore, all of the solutions in this set are in the main Pareto set as well.

D. *Investigating the effect of different mappings for a certain architecture on the objectives*

Using our visualization tool, the designer can easily explore the influence of different mappings on the design criteria. Here we consider three architecture instances and study how the objective values change with modifying the mapping decisions. Since all the design points with the same architecture have the same cost, we examine the effect of mappings on the processing time and energy consumption.

First consider the architecture platform consisting of two ASIPs, one DRAM and one SRAM. The evaluated design points for this architecture are shown in Fig. 7. As can be seen in this figure, for all design points, both the processing time and the energy consumption are extremely poor. Even with different mappings, we cannot get a good design point with this specific architecture. Therefore, this architecture is not a suitable solution for our case study.

Fig. 8 shows the discovered solutions for the architecture that contains one ASIC, two ASIPS, one DRAM and one SRAM. Depending on the mapping, you may get a relatively good (e.g. the design point indicated by A) or a very poor (e.g. the design point indicated by B) processing time. However, the energy consumption of these 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.

In Fig. 9, all design points have two ASICs, one ASIP, one mP, two DRAMs and one SRAM in their underlying architecture. From this figure, we can see that all solutions have relatively good processing time and energy

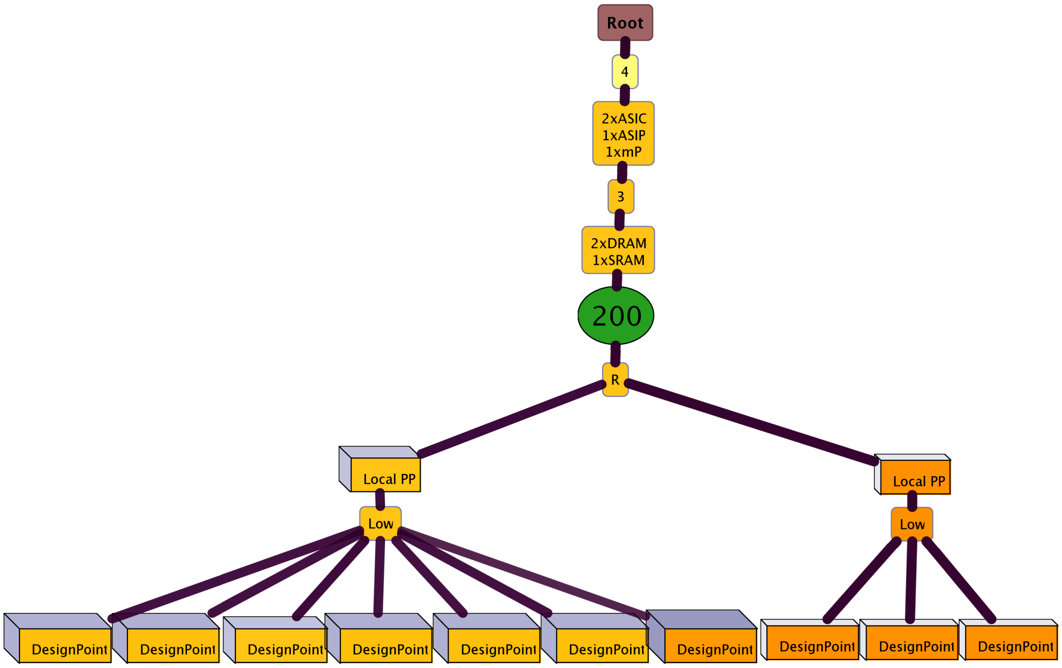

Fig. 9. All design points are relatively good

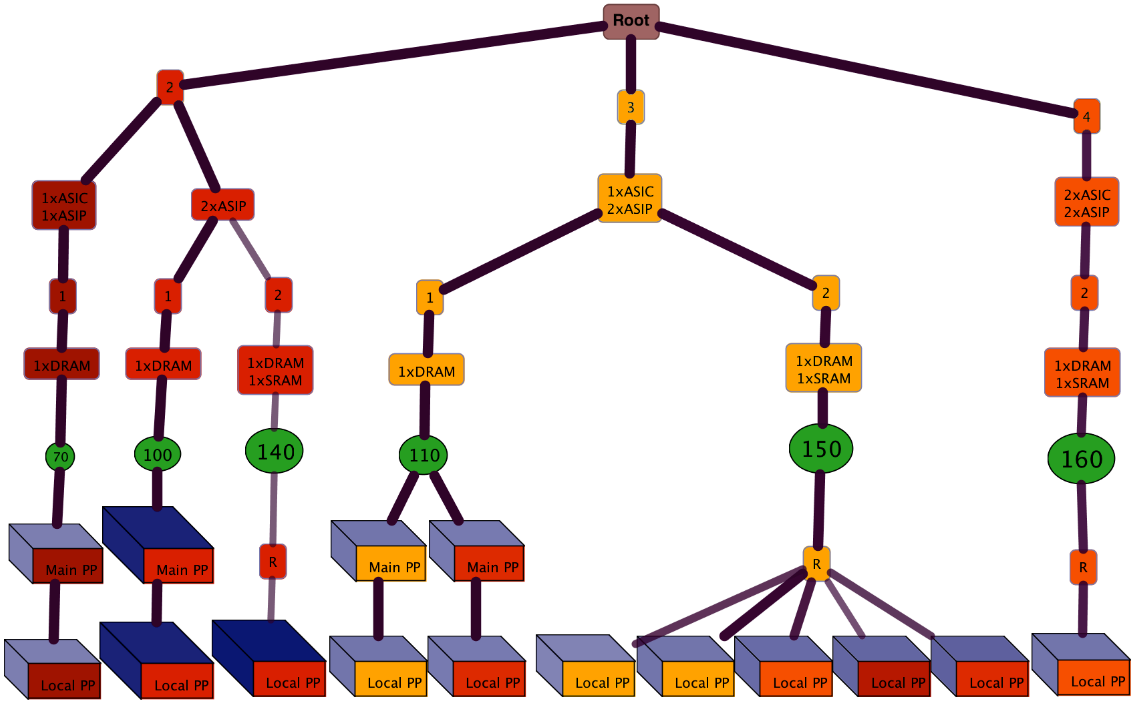

Fig. 10. Local Pareto points without mP in their underlying architecture

consumption. Therefore, for this particular architecture, with different mappings we can get approximately good design points. So, if the designer is looking for a system that can flexibly deal with different mappings, this architecture may be a good solution.

E. Filtering Design Points by their Objective Values

The filtering option in VMODEX allows designers to easily view only preferred design points. In this case study, we apply three filtering scenarios to see the characteristics of design points with some specific objective values and to consider the properties of design points in some particular parts of the design space.

In the first scenario, we are investigating the properties of local Pareto points, which do not have a microProcessor (mP) in their underlying architecture. Fig. 10 shows the parts of the design space that do not contain a microprocessor. As can be seen in this figure, for all points, both the processing time and energy consumption is poor. Therefore, the fast but expensive microprocessor has essential effect on getting a good processing time and energy consumption.

In the second scenario, we are interested in those design points, which have the best processing time. Fig. 11 shows these solutions. From this figure, we can see that all of them are fairly expensive. Their cost is higher than 200, while the average cost is 160. Furthermore, none of them have good energy consumption. This is because for all design points, the size of the third dimension is big and its color is dark.

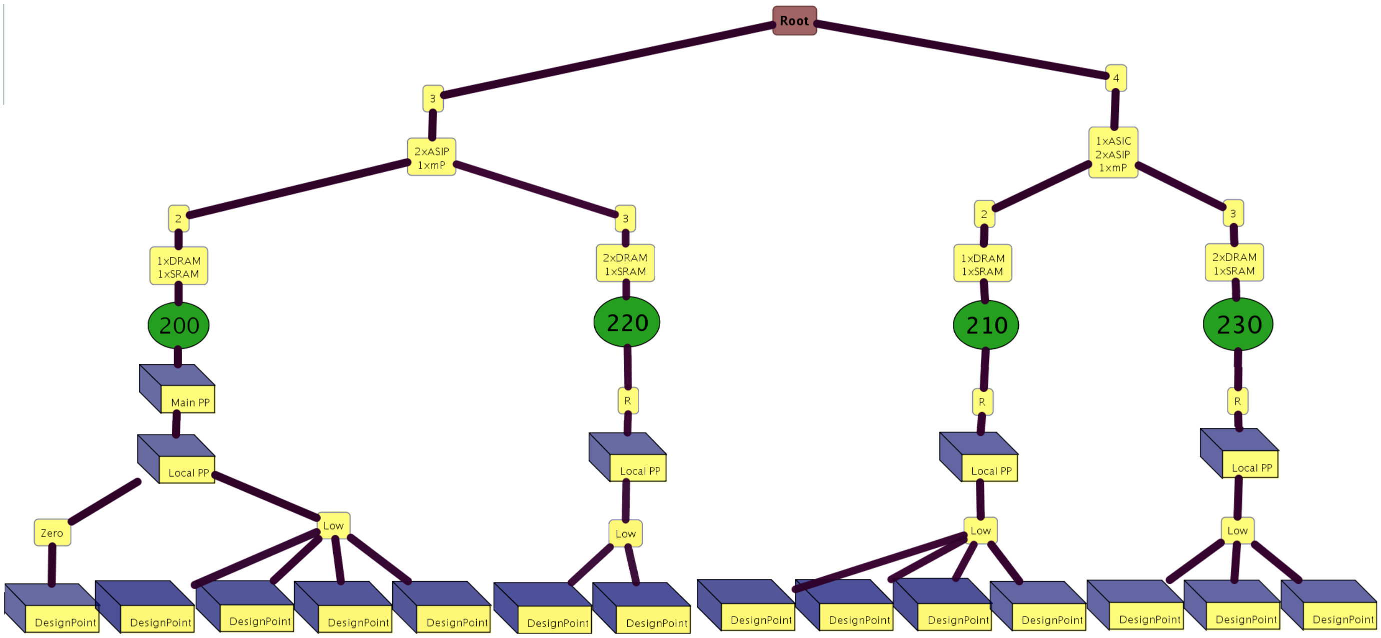

Fig. 11. Filtering by the best processing time

Moreover, as in VMODEX both the design parameters and objectives are shown in a single view, we can easily find out which combinations of architectural components yield design points with objective values inside the interested ranges. As can be seen in Fig. 11, only with four architecture instances we can get design points with the best processing time.

In the third scenario, we are looking for design points with the best energy consumption. The result of this filtering is shown in Fig. 12. As can be seen in this figure, the costs of the solutions are varied from 130 to 230. That means that with a cheap architecture, the designer can get the same energy consumption as with an expensive one. In addition, we can see that they are scattered almost in the entire design space. Moreover, all of them have relatively good processing time.

V. Conclusion

In this paper, we presented a visualization tool, VMODEX, which helps designers to understand how an MOEA explores the design space during its iterations and converges towards the optimal design points. It also enables very powerful and rapid analysis of DSE data such as: where the optimum design points are located in the design space and what objective values they have, how design parameters influence each objective and the correlations among multiple objectives.

VMODEX models the design space as a tree in which both the design parameters and objectives can be seen in a single view. Furthermore, there is no limitation on the number of neither parameters nor objectives. In our tool, we provide several interactive capabilities, which allow designers to play with data and find out some interesting and important features that may not be found just by looking at the raw data or by using the tradition 2D/3D graphs.

References

[1] C. Erbas, S. Cerav-Erbas and A.D. Pimentel. “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.

[2] P. Marwedel, B. Sirocic. “Multimedia components for the visualization of dynamic behavior in computer architectures”, in the Proc. of the Workshop of Computer Architecture Education, 2003.

[3] C. Yehezkel, W. Yurcik, M. Pearson, D. Armstrong. “Three simulator tools for teaching computer architecture: Easycpu, little man computer, and rtlsim”, Journal on Educational Resources in Computing (JERIC), vol. 1, no. 4, pp. 60-80, 2001.

[4] R. Bosch, et al, “Rivet: A flexible environment for computer systems visualization”, SIGGRAPH Computer Graphics, vol. 34, no. 1, pp. 68-73, 2000.

[5] T. Taghavi, A. D. Pimentel, and M. Thompson. “Visualization of Computer Architecture Simulation Data for System-level Design Space Exploration”, in the Proc. of Int. Symposium on Systems, Architectures, MOdeling and Simulation (SAMOS '09), July 2009.

[6] E. Hart and P. Ross. “GAVEL - A New Tool for Genetic Algorithm Visualization”, IEEE Trans on Evolutionary Computation, Vol. 5, No. 4, pp. 335-348, August 2001.

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

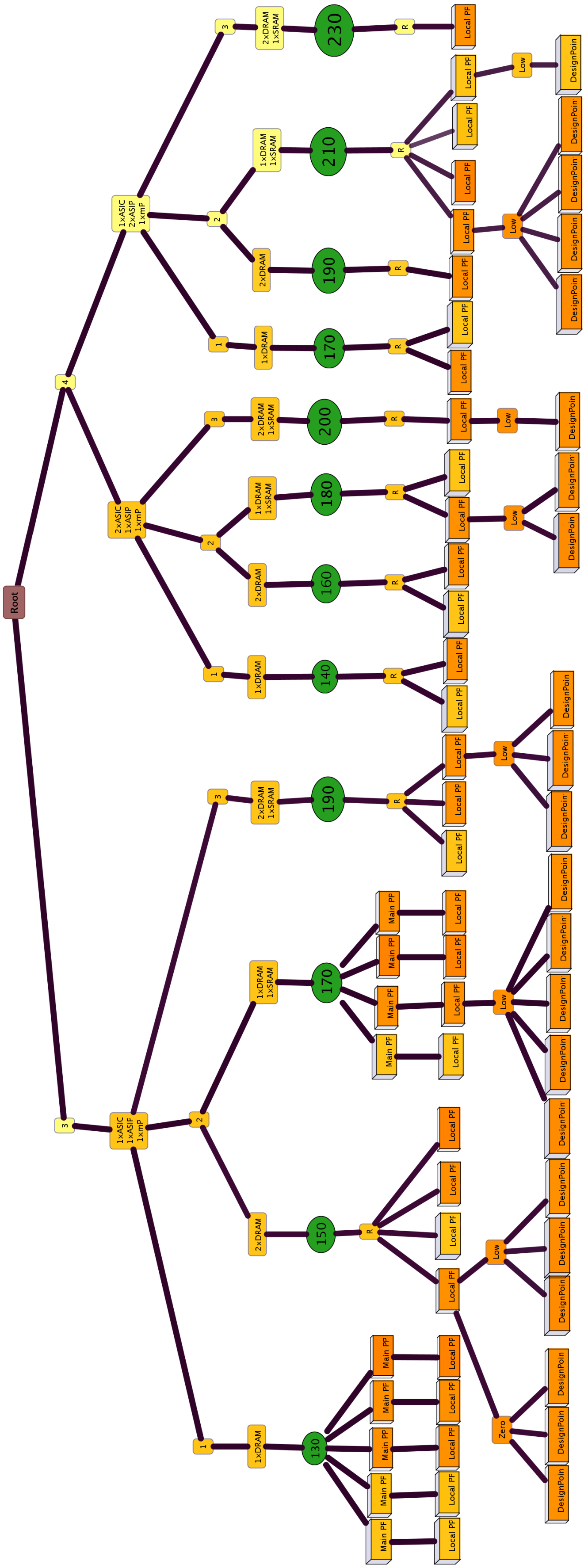

Fig. 12. Filtering by the best energy consumption