# Web-Based GUI for Pareto Front Visualization and Analysis

Alexandr Murashkin
University of Waterloo
Wateroo, Ontario
amurashk@uwaterloo.ca

## ABSTRACT

In product line development, there is a problem of discovering product configurations that are optimal with respect to set of objectives within a product family. There are tools that solve the problem, but in most cases there are multiple optimal product configurations, and there is room for innovation here regarding visualization, choosing a subset of optimal products, trade-off analysis and comparison of generated configurations. We propose a software tool - a GUI that visualizes the set of optimal products and offers certain visualization and human interaction techniques to accomplish these tasks. Our pilot evaluation of this tool has shown feasibility of the approach and given some directions of further research in this area.

## Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Graphical user interfaces (GUI)*

## General Terms

Product lines, Multi-objective optimization, Pareto front, visualization

## Keywords

Visualization, GUI, bubble chart, user experiment

## 1. INTRODUCTION

### 1.1 Background

The problem occurs in product line development. Product line engineers, model architects, or developers design a product family - set of different products with shared architecture and common components [4]. Individual products are characterized by selection of features from the feature model of the product family [5]. Some features contain quality attributes that impact on the product total quality attributes - so called extra-functional features [2]. For example, presence of second CPU on mobile device makes the phone total performance rate two times higher, but increases the phone total cost. The challenge faced by engineers is discovering which products are optimal with respect to a set of goals (e.g. to achieve maximal performance) [7].

There is a tool named Clafer Multi-Objective Optimizer (ClaferMoo) [7]. The tool takes a feature model with quality attributes and set of objectives (written with a lightweight modeling language called Clafer [1]) as input), solves multi-objective optimization problem with respect of the objectives and generates all optimal model configurations. In case of product lines, the feature model defines a product family, and the optimal model configurations are simply optimal product configurations, or prospective products.

### 1.2 Motivation

The existing ClaferMoo tool was not designed to provide a good user interface. The tool outputs generated configurations one-by-one as a structured text. The problem is that product families can contain lots of features, and multiple optimal configurations can be generated. If manufacturer needs to choose 5 final products out of 50 possible configurations, then the task becomes difficult: manual comparison of generated product configurations is, obviously, time and effort consuming. Moreover, there are trade-offs between several features (e.g. performance versus cost) that cannot be processed in an automatic way, so trade-offs related tasks have to be completed by human in an effective and straightforward way.

### 1.3 Problem statement

Currently there is lack of commonly used techniques and tool support that help manufacturers effectively manage generated product configurations, compare them side-by-side, understand trade-offs, and pick up the most suitable products for manufacturing. Current visualization solutions do not allow interactive exploration and often are not connected to design decisions. The objectives are to build an an effective and usable software tool that can visualize the product models, trade-offs and help to identify an optimal solution. The tool also has to be light-weight and cross-platform and has to use ClaferMoo as a back-end.

### 1.4 Related terms and conventions

The paper does not deal with multi-objective optimization problem directly, so there is no problem description in this paper. The only related term used - *Pareto front* - is a set of non-dominant solutions of multi-objective optimization problem. In this paper, Pareto front is just a set of optimal
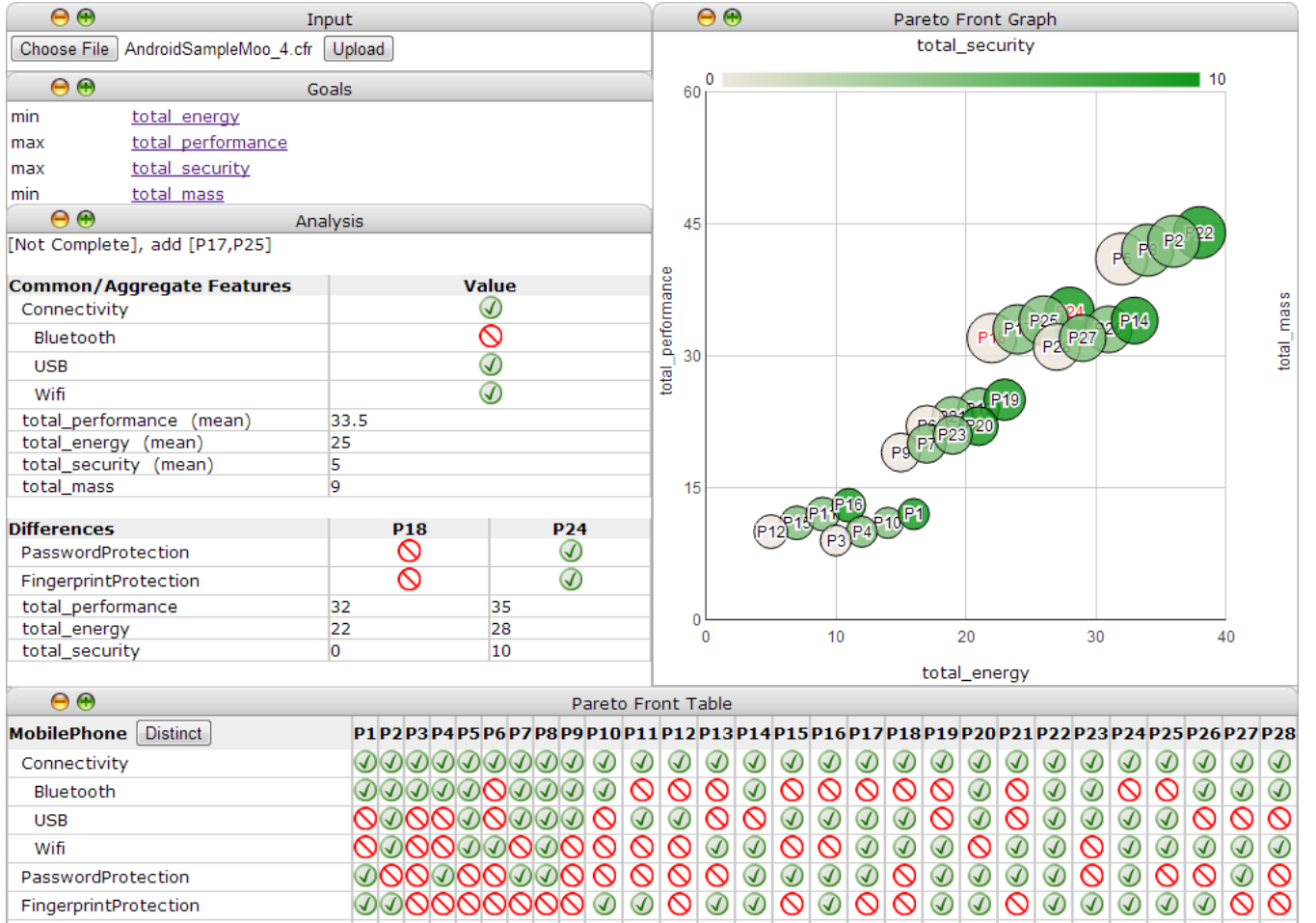
**Figure 1: ClaferMoo Visualizer: user interface**

**Input**

Choose File | AndroidSampleMoo_4.cfr | Upload

**Goals**

| | |
|---|---|
| min | total_energy |
| max | total_performance |
| max | total_security |
| min | total_mass |

**Analysis**

[Not Complete], add [P17,P25]

| Common/Aggregate Features | Value |
|---|---|
| Connectivity | ✓ |
| Bluetooth | 🚫 |
| USB | ✓ |
| Wifi | ✓ |
| total_performance (mean) | 33.5 |
| total_energy (mean) | 25 |
| total_security (mean) | 5 |
| total_mass | 9 |

| Differences | P18 | P24 |
|---|---|---|
| PasswordProtection | 🚫 | ✓ |
| FingerprintProtection | 🚫 | ✓ |
| total_performance | 32 | 35 |
| total_energy | 22 | 28 |
| total_security | 0 | 10 |

**Pareto Front Table**

MobilePhone [Distinct]

| | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | P11 | P12 | P13 | P14 | P15 | P16 | P17 | P18 | P19 | P20 | P21 | P22 | P23 | P24 | P25 | P26 | P27 | P28 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Connectivity | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Bluetooth | ✓ | ✓ | ✓ | ✓ | ✓ | 🚫 | ✓ | ✓ | ✓ | ✓ | 🚫 | ✓ | 🚫 | ✓ | 🚫 | 🚫 | 🚫 | 🚫 | ✓ | 🚫 | ✓ | ✓ | 🚫 | 🚫 | 🚫 | ✓ | ✓ | ✓ |
| USB | 🚫 | ✓ | 🚫 | 🚫 | ✓ | ✓ | ✓ | ✓ | ✓ | 🚫 | ✓ | ✓ | 🚫 | ✓ | ✓ | ✓ | ✓ | ✓ | 🚫 | ✓ | 🚫 | ✓ | ✓ | ✓ | 🚫 | 🚫 | 🚫 | 🚫 |
| Wifi | 🚫 | ✓ | 🚫 | ✓ | ✓ | ✓ | 🚫 | ✓ | 🚫 | ✓ | 🚫 | 🚫 | ✓ | ✓ | ✓ | 🚫 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| PasswordProtection | ✓ | 🚫 | ✓ | ✓ | 🚫 | ✓ | ✓ | ✓ | ✓ | 🚫 | ✓ | 🚫 | ✓ | ✓ | ✓ | ✓ | 🚫 | ✓ | ✓ | ✓ | 🚫 | ✓ | ✓ | ✓ | 🚫 | ✓ | 🚫 | 🚫 |
| FingerprintProtection | ✓ | ✓ | 🚫 | 🚫 | 🚫 | 🚫 | 🚫 | 🚫 | ✓ | ✓ | 🚫 | ✓ | ✓ | 🚫 | ✓ | 🚫 | ✓ | 🚫 | ✓ | ✓ | 🚫 | ✓ | ✓ | ✓ | ✓ | ✓ | 🚫 | 🚫 |

product configurations generated by ClaferMoo tool. For instance, for some product family model of mobile phones, ClaferMoo generates several product configurations: mobile phone $P1$, mobile phone $P2$, mobile phone $P3$, and mobile phone $P4$ that have the highest total performance and the lowest total mass. Each phone can have distinct features (e.g. $P1$ and $P4$ have second CPU, $P1$ and $P3$ have physical keyboard) and total quality attributes (e.g. total performance of $P1$ is 500 MHz and total mass of $P4$ is 300 grammes). Pareto front is simply the set $P = \{P1, P2, P3, P4\}$. In this paper, the terms "set of optimal product configurations", "set of optimal products", "Pareto front" are used as synonyms and often denoted as just "products".

## 1.5 Paper structure

Implementation section gives brief overview of our solution - a software tool, shows two techniques of visualizing Pareto Front - the graph and the table, and opportunities of the tool related to the Pareto Front analysis. User evaluation section contains results of our pilot study - a controlled experiment aimed to do the initial evaluation of the implemented tool and visualization techniques. Related work section compares the work to related approaches regarding Pareto front visualization and user interfaces. Conclusion section summarizes completed work and lists work implications.

## 2. IMPLEMENTATION

## 2.1 Overview

We implemented a web-based tool called ClaferMoo Visualizer. The tool is a GUI for ClaferMoo tool and can be considered as a presentation layer, since its input data is actually the output data of ClaferMoo. The user interface is mainly composed of several collapsible windows (Fig. 1). Each window has certain purposes as listed below.

- Input - allows input file uploading.

- Goals - shows the multi-objective optimization goals.

- Pareto Front Table - a tabular representation of Pareto Front.

- Pareto Front Graph - a graphical representation of Pareto Front.

- Analysis - shows commonalities and differences among Pareto Front, and class equivalences.

A typical workflow is the following. First, user navigates to the tool page, uploads a Clafer file using the input window, and ClaferMoo runs successfully. The tool processes

the data transferred from ClaferMoo and fills in the windows Goals, Pareto front table and Graph. Then user can perform various actions depending on use cases. User can inspect the table view and see what products are generated, what features and quality attributes they have. User can inspect the graph and see all the products there displayed as labeled bubbles, view the popup with concrete values for each product, change the dimensions that should be visualized on X and Y axes, and using opacity and bubble size. User can select products on the graph for analysis, the corresponding window is filled in with selected products, and user can see commonalities and differences. User can upload another file or just close the browser page window when finished working.

## 2.2 Pareto Front Table



**Figure 2: Pareto Front Table. Distinct features are highlighted**

Pareto Front Table (Fig.2) is a straight-forward way to represent the whole set of product configurations. Row headers represent features and total quality attributes, the hierarchical structure is preserved. Column headers represent product configurations and denoted as $P1, P2, ...PN$, $N$ - the number of generated configurations. Cells at the intersection of column $C$ and row $R$ are defined in the following way:

- If $R$ is a feature, and product $C$ has the feature, then the cell contains "yes" (a green tick), otherwise "no" (a red crossed circle).

- If $R$ is a quality attribute, then the cell contains the numeric value of the quality attribute.

## 2.3 Pareto Front Graph

Pareto Front Graph is aimed to graphically represent Pareto Front in such a way that user can clearly see the whole set of solutions and pick one or more solutions for the analysis. The graph is actually a bubble chart (Fig.3).

Let $m$ denote the cardinality of the metric space, and $s$ the cardinality of the solution space. Let's denote bubble chart representations as $X$ (horizontal axis), $Y$ (vertical axis), $Z$ (bubble opacity) and $T$ (bubble size). Bubble chart appears to be good for representing a Pareto Front with $m$ is less than or equal to 4, and $s$ is not very big (the upper bound depends on diversity of configurations and available space on the screen). Moreover, there are some special cases:

- If $m = 2$, then $X$ and $Y$ representations are used.

- If $m = 3$, then $X$, $Y$ and $Z$ representations are used.

- If $m = 4$, then $X$, $Y$, $Z$ and $T$ representations are used.

- If $m > 4$, then the user may choose which metrics are to be represented using $X$, $Y$, $Z$ and $T$ representations by drag-and-drop from the "Goals" window onto the four sides of the graph. So projections are used to deal with higher dimensions.

In Fig.3 showing the graph of the phone example, $X$ axis represents total energy consumption. Bubbles to the right represent products with higher energy consumption, to the left - lower consumption. $Y$ coordinate - total performance. The higher the items, the more performance they can provide. $Z$ representation - total security. The more opaque bubble, the more security the corresponding product has. $T$ (bubble size) represents total mass. Wider bubbles correspond to heavier phones. So, all bubbles are presented according to their total energy and total security ($X$ and $Y$ values), their opacity ($Z$ representation) is proportional to total security, and size ($T$ representation) is proportional to total mass (Fig.3).

Some representations can be assigned in a way that analogies can help users with the future analysis. Representation $T$ (bubble size) can denote something like mass, volume or size: "heavier" items will be bigger, "lighter" - smaller. Representation $Z$ (bubble opacity) can denote something that has the notion of "good", "bad" and "intermediate" : security, pressure, etc.: fully transparent means completely insecure, fully opaque means "totally secure".

Users can mouse over on some bubble, and the popup appears that shows the exact quality attribute values. Users can also pick one or more products for analysis by clicking on the corresponding bubbles on the graph.

## 2.4 Pareto front analysis
The Analysis window is designed to support the following tasks:

- Show features of one selected product.

- Show common features of multiple selected products.

- Show differences among multiple selected products.

- Denote whether common features of the selected products define a "complete equivalence class".
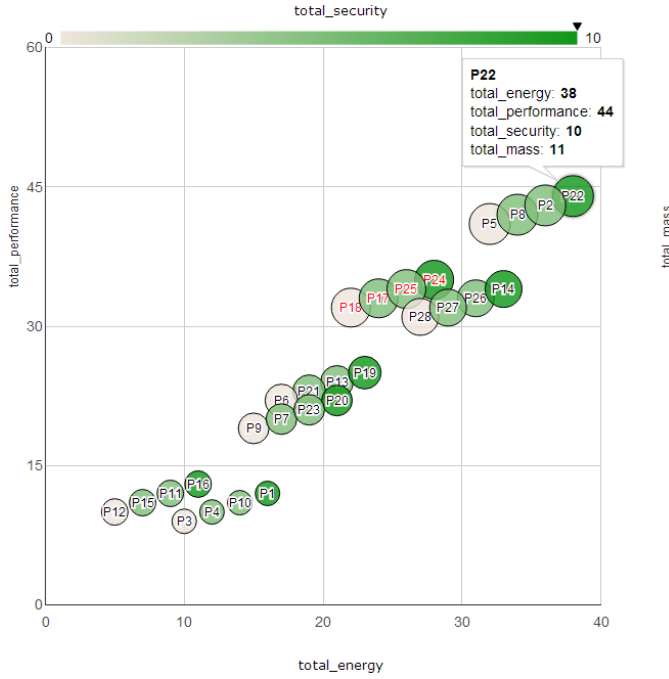
**Figure 3: Pareto Front Graph for the mobile phone example. Legends: horizontal axis - total energy, vertical axis - total performance, bubble opacity - total security (label and legend are shown at the top), bubble size - total mass (label shown at the right). Popup window shows exact values of the product in all 4 dimensions.**

- Denote what products can be added to the selection set to define a "complete equivalence class".

The *complete equivalence class* notion is a largest set of products which have the same commonality - set of pairs $(f, v)$, where $f$ - a feature from product family, $v$ - a boolean that denotes whether this feature included in certain product. Adding another product to the set will decrease the commonality. Formally, we can define a relation $R$ over Pareto front $P$, $aRb$ if and only if the products $a$ and $b$ have the same commonality $C$.

Considering the phone example, if user picks P18 and P24, then Analysis window displays that the only different features are $PasswordProtection$ and $FingerprintProtection$: P18 has neither, P24 has both. The features $Connectivity$ (a composite feature), $USB$ and $WiFi$ are present in both products, as well as both have the same total mass. $Bluetooth$ is not included in either. So P18 and P24 have the same commonality set { (Connectivity, $true$), (USB, $true$), (WiFi, $true$), (Bluetooth, $false$) } (Fig.4). Mean value (if quality attribute value is not the same among all selected products) and exact value (if quality attribute value is the same among all selected products) of quality attributes are shown in the Common/Aggregate features region.

Now, to make the equivalence class complete, the tool recommends to add P17 and P25: if the user picks them, then a complete equivalence class is defined: the set of all products

[Not Complete], add [P17,P25]

| Common/Aggregate Features | Value |
|---|---|
| Connectivity | ✅ |
|   Bluetooth | 🚫 |
|   USB | ✅ |
|   Wifi | ✅ |
| total_performance (mean) | 33.5 |
| total_energy (mean) | 25 |
| total_security (mean) | 5 |
| total_mass | 9 |

| Differences | P18 | P24 |
|---|---|---|
| PasswordProtection | 🚫 | ✅ |
| FingerprintProtection | 🚫 | ✅ |
| total_performance | 32 | 35 |
| total_energy | 22 | 28 |
| total_security | 0 | 10 |

**Figure 4: Analysis window: P18 and P24 are picked up**

with $USB$ and $WiFi$ and *not Bluetooth* (Fig.5).

[Complete] - The set is a complete concept

| Common/Aggregate Features | Value |
|---|---|
| Connectivity | ✅ |
|   Bluetooth | 🚫 |
|   USB | ✅ |
|   Wifi | ✅ |
| total_performance (mean) | 33.5 |
| total_energy (mean) | 25 |
| total_security (mean) | 5 |
| total_mass | 9 |

| Differences | P17 | P18 | P24 | P25 |
|---|---|---|---|---|
| PasswordProtection | ✅ | 🚫 | ✅ | 🚫 |
| FingerprintProtection | 🚫 | 🚫 | ✅ | ✅ |
| total_performance | 33 | 32 | 35 | 34 |
| total_energy | 24 | 22 | 28 | 26 |
| total_security | 5 | 0 | 10 | 5 |

**Figure 5: Analysis window: a complete equivalence class is shown**

Actually complete equivalence classes might be visualized as separate clusters (See evaluation and future work section for ideas of clustering).

## 3. EVALUATION
### 3.1 Form and purpose
The aim of the user evaluation was to determine whether visualization and interaction techniques implemented in the tool are good enough for now, and what is the further direction of development. This evaluation is a pilot study, not the final one: we are planning to conduct further evaluations with our industry partners.

To evaluate our tool, we performed a controlled experiment in which the participants were asked to perform a predefined set of tasks using the tool and answer provided questions. Occasionally, we also asked additional questions as appropriate in the given context.

The experiment included 11 tasks split into 3 groups based

on the difficulty level followed by a final questionnaire (Appendix A). Difficulty levels made the main part of the experiment well-structured and reduced learning effect. The final questionnaire was used to get an informal feedback upon the tasks completion. Three subjects - two full-time graduate students and one postdoctoral fellow of Generative Software Development Lab, University of Waterloo. All three subjects have background in product lines development and two subjects have expertise in multi-objective optimization, so no background questionnaire was used to determine their background. Prior to the experiment, we gave a short presentation of tool features to the participants, intended ways of using the tool, and completion of some sample tasks.

## 3.2 Results

### 3.2.1 Task completion

All participants were able to complete almost all tasks: some tasks were not understood properly or brought confusion (for example, the question in T10 did not clarify that common features are asked, not common quality attributes as one subject understood), but after clarification (the experiment form allowed that), the subjects were able to complete them. Two subjects completed question 1 in T11 in the way they were not supposed to do: they did not notice the suggestion of the tool and manually looked at the Pareto front table and searched for the products that have USB and Bluetooth. However, the result was still correct.

Not all tasks required subject to provide some concrete correct end result. For instance, for question 3 in the task T11, all subjects made conclusions that total mass differs not significantly, but one of them made a notice that the domain knowledge is important to determine whether the difference in total mass significant or not.

### 3.2.2 Pareto front visualization

Regarding the visualization, all three participants found the bubble chart visualization useful. Participants were able to make decisions in 4-dimensional space, compare bubbles and pick up the ones they needed.

Regarding limitations, all participants noticed that is is hard or even impossible to compare products that differ in third and forth dimensions by a small value. If products differ in positions ($X$ and $Y$ axises), it is easy to notice, because one pixel difference on the plane is noticeable. In contrast, regarding $Z$ and $T$ dimensions this exact comparison seems to be not possible in bubble charts. The experiment has shown that the bubbles of products with $total\_mass = 2$ and $total\_mass = 3$ appeared to have almost the same size, and it was hard to notice a small change in diameter. In the example, total security had three discrete values: 0, 5 and 10. Two participants quickly understood the difference among all the three opacities, but one was confused with zero value: he said that from the graph it is not clear whether the product represented by the bubble has exact zero total security or slightly bigger. Probably, the reason was that bubbles representing products with 0 total security actually were not 100% transparent: they were slightly grayed (internal bubble chart feature for bubble readability improvement). Fortunately, all three participants could rely on the popup feature of the graph to know the exact values of quality attributes. The idea to deal with the limitation in the

future is the following: the user chooses all the bubbles that look to be the same, there should be an option to zoom in, or to take them all out and display on a separate graph. The scale is recalculated in order to make the differences more noticeable. Another limitation noticed by one participant is that bubble diameter matters, and this can limit the opportunities of such a visualization (lack of the space on the screen). Noticed limitations is a good result.

There were some small problems that included the lack of conventions how the labels of $Z$ and $T$ representations should be specified and where should they be located. Current position of labels was misleading for one participant: he said that the mass label ($T$ representation) should be put in another place, otherwise it is confused with meaning of $Y$ axis. Moreover, as two participants mentioned, the mass label should be more informative. Probably, the label should include few empty bubbles with distinct sizes to show that total mass is proportional to the bubble size. Two participants noticed the correlation between mass, security and performance, but they were not asked to do so. It is a good indicator that users can reason among multiple dimensions with this visualization. Overall, Pareto front visualization using bubble charts seems to be very promising and natural to use.

### 3.2.3 Clustering

It is interesting that during our experiment all three participants started to talk about clustering before they met the question on that. They used various metaphors: "line to separate products", "scope", "square" or "range" to represent the same notion - cluster. It appears that grouping products together by some algorithm (by common features, common values of quality attributes, or proximity) would be helpful for them to complete various tasks or just to compare together two groups. Two participants found the idea of clustering by feature useful, another one said that this would be probably useful, but for some concrete scenarios. One participant noted that he is not sure that equivalence classes is the best way to do clustering. So, clustering appeared to be an important prospective feature.

### 3.2.4 Other problems and suggestions

There were other suggestions and problems related to the tool in general. All three participants mentioned that filtering by features is important, and there is a need in top-down decision making: given a set of features, how to choose suitable products that have these features. Participants wanted to select products in the Pareto front table by clicking on the columns and also by clicking on the features in the table. The tool did not allow to accomplish this, but users mentioned that they would add necessary constraints to the model, so the tool will recalculate Pareto front and give the right result. In any case, this is a good point for the future work.

Two participants brought some interesting point: in task T8 they could make the conclusion that certain features (PasswordProtection and FingerprintProtection) contribute to total security even without knowledge about the exact values of feature attributes. However, they wanted to look at the original model to give a precise answer.

There were other minor usability problems. Selection and

unselection of bubbles was not clear, since users were not asked to pay attention on dealing with selections properly assuming this is easy. One user experienced problems while finding bubbles with certain label on the graph. Anyway, the problems mentioned in this section are specific to use cases and we can easily resolve them in the future.

## 4. RELATED WORK

Regarding Pareto Front Visualization, there are several various approaches. [11] shows an approach of visualization of 4-dimensional solution space by projection into 3-dimensional space and usage of 3D scatter plots. Our approach is to visualize 4-dimensional solution space on the plane and use bubble charts. Bubble charts are more informative than simple scatter charts (since each bubble has a label), and various in perception ways (bubbles differ by position, size and opacity). However, bubble charts are less suitable for visualization of big data (see limitations in the evaluation section) compared to scatter plots. The reason of our usage of bubble charts and avoidance of classical three-dimensional graphs is that it is straight-forward to select multiple bubbles on the plane (no rotations and additional projections are needed if the number of metrics is 4 or less) for further analysis. Visualization of big data with our approach can be done if clustering is implemented, and this is our future work direction.

Bubble charts were also used in [8], but for other purposes: the first and second metrics (bubble position) represent the two quality attributes, third and forth (bubble color and size) represent standard deviations of the two quality attributes for robustness analysis results.

Other papers presented Pareto front visualization techniques were considered, for instance, Level Diagrams [3], heatmaps [9], self-organized maps [6]. In contrast to our approach, the papers above did not make an effort to connect the metric visualization with the design decisions and they did not support interactive exploration of the Pareto front.

Regarding user interfaces dealing with multi-objective optimization problems, there is a related work [10]. The tool presented in the paper is general-purpose and designed for the entire process: from problem specification to result visualization, focusing more on inputs. In contrast, the our tool is designed to deal with solution results (the outputs) and focus on product-line related use cases. It is worth to notice that our tool fulfills a future work statement (incorporation of appropriate visualizations) and shares a goal (to make the tool accessible to business people) of the latter paper. There is another work in progress where multiple users are driven by particular decisions and understanding their trade-offs, whereas our approach assumes one user who cares more about total values and bottom-up exploration.

## 5. FUTURE WORK

Important future work includes identifying use cases for the tool from engineers working for our industrial collaborators, implementing support for theses use cases, and conducting evaluation with the engineers. Next, a work in Pareto Front clustering is encouraged, since all the user evaluation subjects mentioned clustering. Instead of representing the whole Pareto Front, users would be able to focus on clusters. Clus-

tering can be based not only on bubble positions, but on the features shared by corresponding products, so each cluster can be defined using a notion of complete equivalence class. It appeared that clustering will be useful for visualization and top-down analysis purposes. And finally, there is an idea to implement top-down analysis. It turned out that current implementation is focused more on bottom-up exploration, given that user sees individual products. However, during the experiment, users wanted to make decisions from top to bottom, filtering out the products automatically by presence or absence of certain features.

## 6. CONCLUSION

### 6.1 Summary

We implemented a working and usable tool, which can be presented as an effective graphical user interface for Clafer-Moo tool. User experiment showed that the implemented tool helps to successfully complete basic tasks and to make conclusions on Pareto fronts in 4-dimensional space as well as revealed future work directions.

### 6.2 Implications

- Four-dimensional Pareto front visualizations can be effectively implemented using bubble charts.

- Pareto front analysis can be simplified with usage of effective visualizations.

- Top-down Pareto front analysis is necessary in some scenarios.

- Product configuration sets can be characterized by common features of its elements.

## 7. REFERENCES

[1] Kacper Bąk, Krzysztof Czarnecki, and Andrzej Wasowski. Feature and meta-models in clafer: Mixed, specialized and coupled. In *3rd International Conference on Software Language Engineering*, pages 291–301, October 2010.

[2] David Benavides, Pablo Trinidad, and Antonio Ruiz-Cortés. Automated reasoning on feature models. In *Advanced Information Systems Engineering: 17th International Conference*, LNCS. Springer, 2005.

[3] X. Blasco, J. M. Herrero, J. Sanchis, and M. Martínez. A new graphical visualization of n-dimensional pareto front for decision-making in multiobjective optimization. *Inf. Sci.*, 178(20):3908–3924, October 2008.

[4] Paul C. Clements and Linda Northrop. *Software Product Lines: Practices and Patterns*. SEI Series in Software Engineering. Addison-Wesley, 2001.

[5] Kyo C Kang, Sholom G. Cohen, James A. Hess, William E. Novak, and A. Spencer Peterson. Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical report, Software Engineering Institute - CMU, 1990.

[6] Shigeru Obayashi and Daisuke Sasaki. Visualization and data mining of pareto solutions using self-organizing map. In Carlos Fonseca, Peter Fleming, Eckart Zitzler, Lothar Thiele, and Kalyanmoy Deb, editors, *Evolutionary Multi-Criterion Optimization*,

volume 2632 of *Lecture Notes in Computer Science*, pages 71–71. Springer Berlin / Heidelberg, 2003. 10.1007/3-540-36970-8_56.

[7] Rafael Olaechea, Steven Stewart, Krzysztof Czarnecki, and Derek Rayside. Modeling and multi-objective optimization of quality attributes in variability-rich software. In *International Workshop on Non-functional System Properties in Domain Specific Modeling Languages (NFPinDSML'12)*, Innsbruck, Austria, 10/2012 2012.

[8] Silvia Poles, Paolo Geremia, F. Campos, S. Weston, and M. Islam. Moga-ii for an automotive cooling duct optimization on distributed resources. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization*, volume 4403 of *Lecture Notes in Computer Science*, pages 633–644. Springer Berlin / Heidelberg, 2007. 10.1007/978-3-540-70928-2_48.

[9] Andy Pryke, Sanaz Mostaghim, and Alireza Nazemi. Heatmap visualization of population based multi objective algorithms. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization*, volume 4403 of *Lecture Notes in Computer Science*, pages 361–375. Springer Berlin / Heidelberg, 2007. 10.1007/978-3-540-70928-2_29.

[10] Derek Rayside and H.-Christian Estler. A spreadsheet-like user interface for combinatorial multi-objective optimization. In *Proceedings of the 2009 Conference of the Center for Advanced Studies on Collaborative Research*, CASCON '09, pages 58–69, Riverton, NJ, USA, 2009. IBM Corp.

[11] Tea Tušar and Bogdan Filipič. Visualizing 4d approximation sets of multiobjective optimizers with prosections. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, GECCO '11, pages 737–744, New York, NY, USA, 2011. ACM.

# APPENDIX
# A. USER EXPERIMENT TASKS AND QUESTIONNAIRE

### I. Introductory presentation

### II. Tasks

Tasks are split into 3 categories: **Overview**, **Graph** and **Case Studies**. All tasks are to be performed using the think-aloud protocol, that is, the participants should verbally describe what and why they are doing. All tasks are based on the file *AndroidSampleMoo_4.cfr*.

#### 1. Overview:

**T1.** Upload a model file *AndroidSampleMoo_4.cfr*.
**T2**. Tell what features and quality attributes are specified in the Android phone model.
**T3**. Tell what objectives are presented in the Android phone model.
**T4**. Tell how many Android phone configurations are generated.

#### 2. Graph:

**T5**. Identify 4 phones with the highest total performance (Just say: "P29, P30, P12, P4", for instance). What is their total mass?
**T6**. Identify a phone with the lowest energy consumption. What is its total performance?
**T7**. Identify the phones with very low total mass.
**T8**. How many phones have perfect (highest) total security? What features contribute to total security?

#### 3. Cases studies:

**T9.** Your boss says that among all the sets of phones you need to issue only one. He says that he needs to choose one among the phones with the best total performance. He is OK with any high energy consumption, since new Android battery is very good. He says he does not need a perfect total security, but it should be more than 0. So he is OK to sacrifice total security to get more total performance.
*Your actions:*
1) Which product(s) will you choose and why?

**T10.** Your boss says that it is interesting that bubbles representing P5, P8, P2, and P22 are located close to each other. And he wants to know why this happened. Because, maybe, the phones are equivalent in some sense and we can consider them separately as an equivalence class.
*Your actions:*
1) What do the products have in common and what are the differences?
2) Why the bubbles are located close to each other?

**T11.** Your boss needs you to analyze the total performance and total mass of all prospective phones that have USB and WiFi features. He knows that P5, P8, P2, P22 and P18 have USB and WiFi. But he wants to consider all the phones that have USB and WiFi features.
*Your actions:*
1) Make the set to be a complete equivalence class by adding other relevant products to it.
2) Among the selected products within the equivalence class, what is the:
Minimal,
Mean,
Maximal
total performance? Are the phones differ in total mass significantly or not?
3) Based on the question above, is it reasonable for the boss sacrifice total mass to gain more total performance?

### Questionnaire

**Q1.** Do you find the idea of representing the products in a four-dimensional space using bubble chart useful? Any comments on that? Are there limitations? Does the repre-

sentation of the four dimensions make it easy to find "better" products?

**Q2.** What do you think about the notion of the complete equivalence class? What is it and how it can be used? Describe it in your own words.

**Q3.** I noticed that I can group products together based on their features, so each cluster will be denoted by an equivalence class. Let's call it "clustering by features". Do you think it is a good idea or not?