

# Interactive Visual Analysis of Multiobjective Optimizations

Wolfgang Berger\*

VRVis Research Center, Vienna, Austria

Harald Piringer†

VRVis Research Center, Vienna, Austria

## ABSTRACT

Optimization problems are typically addressed by purely automatic approaches. For multi-objective problems, however, a single best solution often does not exist. In this case, it is necessary to analyze trade-offs between many conflicting goals within a given application context. This poster describes an approach that tightly integrates automatic algorithms for multi-objective optimization and interactive multivariate visualizations. Ad-hoc selections support a flexible definition of input data for subsequent algorithms. These algorithms in turn represent their result as derived data attributes that can be assigned to visualizations or be used as a basis for further selections (e.g., to constrain the result set). This enables a guided search that still involves the knowledge of domain experts. We describe our approach in the context of multi-run simulation data from the application domain of car engine design.

**Index Terms:** H.1.2 [Models and Principles]: User/Machine Systems—Human factors; H.5.2 [Information Interfaces and Presentation]: User Interfaces—Interaction Styles

## 1 INTRODUCTION

Optimization is a key task in many application domains (e.g., economics and engineering). It is also a field of research in its own right known as operations research [6] that focuses on mathematical approaches to obtain a (near) optimal solution for complex problems. However, very few approaches involve the user in the optimization process. This is surprising especially for problems comprising multiple potentially contradicting objectives. In this case, a unique and clearly optimal solution often does not exist but it is necessary to analyze trade-offs between objectives in the context of the application background. Tarkannen et al. [7] argue that interactive visualization has the potential to support the analysis of such trade-offs and interdependencies between objectives. Most existing interactive approaches focus on very specific applications like improving algorithms for evaluating microarray data [4] or optimizing p-median problems [3]. Miettinen and Mäkelä introduced a more general interactive method called NIMBUS [1], that asks the user to repeatedly examine the values of objective functions calculated at a current solution. However, NIMBUS requires an analytical representation of the objective functions that is not always available, e.g., for measured data like when comparing similar products with different strengths and weaknesses.

## 2 INTEGRATED VISUAL MULTIOBJECTIVE OPTIMIZATION

The contribution of this poster is an approach that tightly integrates the definition, algorithmic evaluation, and visual analysis of multi-objective optimizations within a single system.

Unlike most previous approaches, our method does not require an explicit representation of the analytical model of the objective functions. Instead, it operates on a set of known results as obtained from discrete measured data or by sampling the - typically constrained - parameter space of continuous objective functions. This

facilitates the use for new application domains and optimization tasks. In this poster, we demonstrate our approach by means of the discrete results of a multi-run simulation from car engine design where the parameters “engine speed” and “load signal” have been varied to mimic different driving cycles.

**Definition** The system supports the definition of optimization problems as sets of an arbitrary number of objectives. Each objective refers to a particular data dimension of the loaded dataset and can be set to be minimized, maximized, or to opt for a certain target value. For example, the goal of the demonstration dataset was to identify operating points of the simulated engine, that maximize and minimize emission of CO and CO<sub>2</sub> (see Fig. 1 (b)).

**Algorithmic Evaluation** Based on the problem definition, the optimization algorithm searches the available set of samples for pareto optimal entries. A particular solution is considered pareto optimal, if it is not possible to improve on one objective without being worse with respect to at least one of the others [5]. In our case of discrete data samples, the result of the algorithm is a non-empty subset of the dataset, also known as pareto frontier. For optimization problems involving more than one objective, the pareto frontier in general consists of multiple pareto optimal points, where each of them represents a certain trade-off. The 2D scatterplot in Fig. 1 (d) highlights the pareto frontier of the aforementioned optimization problem. Each point in this frontier represents a pareto optimal trade-off between torque and the emissions of CO and CO<sub>2</sub>. Data samples outside this frontier are not optimal as for each of them at least one other data sample exists that is better with respect to one or more objectives without being worse with respect to the other objectives. While also pareto optimal points are not necessarily appropriate solutions for a certain application context, they represent a reduction of the potential solution space that may significantly accelerate a subsequent search.

Although having a single pareto frontier that contains the best available trade-offs is the most important information in most cases, some tasks require a more fine-grained ranking from best to worst. Some second- or even third-best results contained in the respective second or third frontiers may also be acceptable and could be superior with respect to other application specific properties. Another possible reason for considering more than one frontier is the loss of precision in case of a discretization of a continuous parameter space. Therefore, our implementation calculates multiple frontiers of decreasing optimality by performing a repeated search on entries that have not yet been assigned to any frontier, i.e., it incrementally peels the dataset like an onion.

**Integrated Visual Analysis** Besides the definition of multiobjective optimization problems and their evaluation in the context of sampled data, our system provides various multivariate views (e.g., scatter plots and parallel coordinates) that are linked by ad-hoc selections on the data. These interactive visualizations are the key technique to involve the user at multiple stages of the optimization. Before the algorithmic evaluation, the user may brush visualizations to restrict the optimization to feasible parts of the data (e.g., parts not violating known constraints), or to local features (e.g., certain load states of the simulated engine) instead of computing pareto frontiers for the entire dataset. This step could also involve the exclusion of data samples that are implausible to application experts, e.g., non-converged simulation runs (see Fig. 1 (a)).

\*e-mail:berger@vrvis.at

†e-mail:piringer@vrvis.at

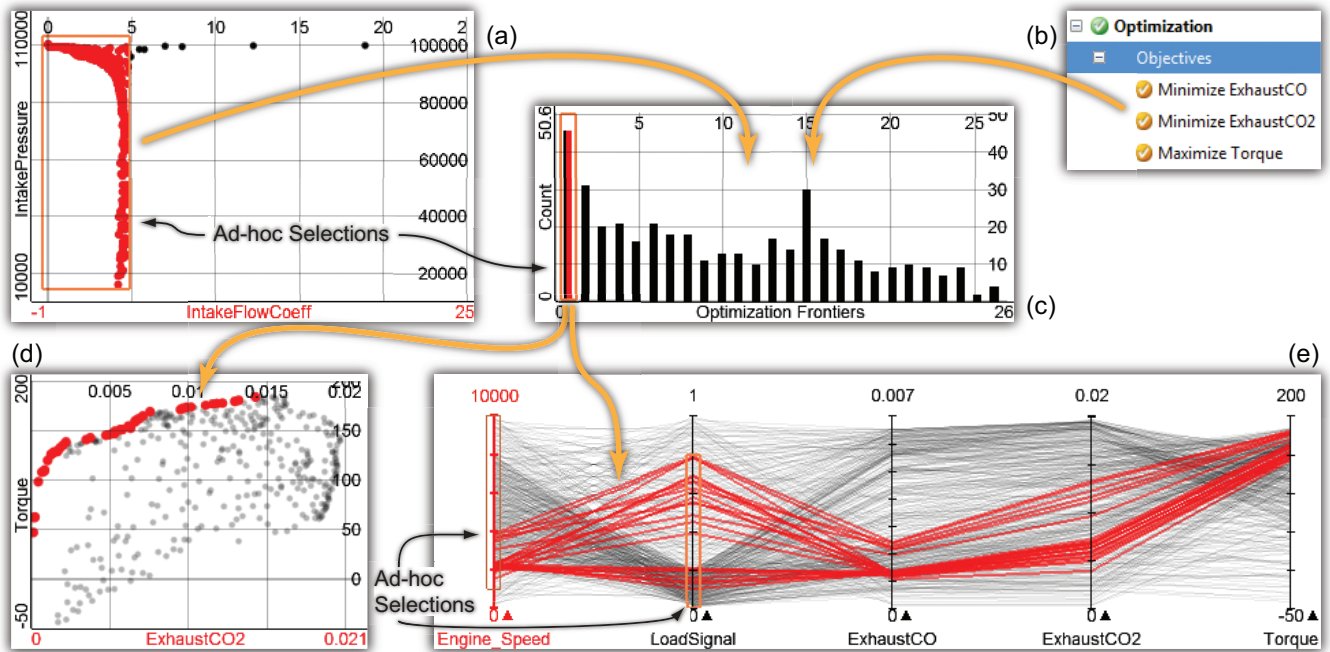


Figure 1: The integrated optimization process. Selections may be used to restrict the input data for the optimization as shown in (a). Based on the problem definition (b), multiple frontiers of pareto optimal points are calculated and written to a dynamic dimension, which can be assigned to visualizations like the bar-chart displayed in (c). In this view, the first pareto frontier has been brushed and is highlighted in linked views ((d), (e)). Subsequent refinement brushes in the parallel coordinates restrict the result set and allow for interactive exploration of additional constraints.

After the computation of pareto frontiers, the key element of our integrated approach is a derived dimension that holds the index of the assigned pareto frontier for each considered entry. This derived dimension provides the link from performing the optimization back to visual analysis: it can be assigned to visualizations and it can be brushed by the user just as any other dimension of the dataset. Thus the analysis of the optimization results does not require the introduction and adoption of new interaction techniques and can easily be integrated into an existing visual analysis process. For example, in Fig. 1 (c) the derived dimension has been assigned to a bar-chart which displays the number of entries for each pareto frontier. The analyst is able to investigate different frontiers by simply brushing them in the bar-chart, which highlights the corresponding data samples in all linked views (see Fig. 1 (d) and (e)). In this example, the parallel coordinates contain axes for “engine speed” and “load signal” to identify the inputs which correspond to the pareto optimal points. This knowledge could serve as a starting point for further investigations. For example, the analyst can add refinement brushes to interactively explore the impact of additional constraints on the result set from within the visualization (see Fig. 1 (e)). Here the engine speed has been restricted to a minimum of 1000 rpm, while entries with a load signal higher than 80 % are excluded. These constraints describe the average operating range of the engine.

### 3 DISCUSSION

The computation of pareto frontiers transforms an n-dimensional problem definition to a single result dimension. This provides good scalability with respect to the number of data dimensions. Concerning the number of data samples, our implementation takes one second on a 2.4 GHz Quad-Core (4 GB RAM) machine to compute all pareto frontiers for 10000 data samples and a problem involving 5 objectives. We employ a straightforward implementation with  $O(n^2)$  complexity for computing the pareto frontiers. This is sufficient in many cases but could easily be replaced by more sophisticated algorithms; however, our implementation utilizes multi-threading to ensure the responsiveness of the application and it updates all visualizations and ad-hoc selections each time a frontier has been computed to provide feedback early [2].

As the main benefit of our approach, domain experts get guidance in their search for an optimal solution while retaining the flexibility to explore trade-offs and to analyze the effect of constraints instantaneously. The tight integration supports the analysis of many scenarios within a short time; for example, users can temporarily deactivate certain objectives or adjust thresholds of constraints and get feedback immediately. The fact that our approach operates on sampled data has the advantage of being applicable also to cases where analytical representations of the objective functions are not available (or do not exist at all). We also consider it a conceptual advantage of our approach that it relies on standard multivariate visualization and interaction techniques. This facilitates an integration in other systems, where users may also be interested in formally specifying their intention and goals as optimization problem.

### REFERENCES

- [1] K. Miettinen and M. M. Mäkelä. Synchronous approach in interactive multiobjective optimization. *European Journal of Operational Research*, 170(3):909–922, 2006.
- [2] H. Piringer, C. Tominski, P. Muigg, and W. Berger. A Multi-Threading Architecture to Support Interactive Visual Exploration. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1113–1120, 2009.
- [3] H. Pirkul, R. Gupta, and E. Rolland. VisOpt: A Visual Interactive Optimization Tool for P-Median Problems. *Decision Support Systems*, 26(3):209–223, Sept. 1999.
- [4] J. Seo, M. Bakay, Y.-W. Chen, S. Hilmer, B. Shneiderman, and E. P. Hoffman. Interactively optimizing signal-to-noise ratios in expression profiling: project-specific algorithm selection and detection p-value weighting in Affymetrix microarrays. *Bioinformatics*, 20(16):2534–2544, Nov. 2004.
- [5] R. E. Steuer. *Multiple Criteria Optimization: Theory, Computation and Application*. Wiley, New York, USA, 1986.
- [6] H. A. Taha. *Operations Research: An Introduction*. Prentice Hall, Upper Saddle River, NJ, USA, 8th edition, 2006.
- [7] S. Tarkkanen, K. Miettinen, and J. Hakanen. Interactive Poster: Interactive Multiobjective Optimization - A New Application Area for Visual Analytics. *IEEE Symposium on Visual Analytics Science and Technology*, pages 237–238, 2009.