

# A FRAMEWORK FOR FLEXIBLE SEARCH AND OPTIMIZATION IN PARAMETRIC DESIGN

Robert Vierlinger<sup>1</sup>, Arne Hofmann<sup>1</sup>

<sup>1</sup> Researcher – University of Applied Arts, Vienna, Austria

Today architectural design processes are more and more influenced by parametric methods. As these allow for a multiplicity of alternatives, the design process can be enriched by computational optimization. Extensive research has shown the efficiency of optimization in engineering and design disciplines. Though, optimization is hereby rather a technical than a design task; it is limited to different autonomous specialist areas and does not enable a comprehensive approach.

Advanced optimization methods facilitate the generation of complex systems, but these procedures are directed and do not provide turnoffs, multiple solutions or altering circumstances. These however are things that are essential for architectural design processes, which mostly do not have clearly defined starting and end points. This practice subdivides the workflow into two independent and recurring tasks: the generation of a parametric model followed by optimization of its driving parameters. The result is then assessed with respect to its actual qualities. The design either is kept, or modifications on the parametric model, its auxiliary conditions and parameters are made and the optimization process starts again from scratch.

The aim of the research project, this paper is referring to, is the development of a flexible generation and optimization framework for practical use in the sense of a continuously accompanying design explorer, in which parameterization is adaptable and objective functions are changeable at any time during the design process. The user is supported in his/her understanding of correlations by identifying a multiplicity of optimal solutions utilizing state-of-the-art multi-objective search algorithms within the core of the framework. Considering the tool as an interactive design aid, an intuitive interface allowing for extensive manual guidance and verification of the search process is featured. Zooming, filtering and weighting within the genotypic, phenotypic and objective space comply with an extensive support of man-machine-dialogue and incorporation of non- or not-yet quantifiable measures. A reusable search history aids examination of design alternatives and the redefinition of constraints, maintaining the continuity of the search process and traceability of results in the sense of rational design verification. Within this work it is not planned to focus on specific optimization targets, but to build an open framework to allow for all kinds of objective functions and in particular the mediation between conflicting targets. In a broader context of general design research, the process of design development from early to final solution is examined, where not even optimization itself but the entire search for an adequate optimization setup is targeted.

Even the research process is at its very beginning, in this paper we already propose a tool that integrates key features of a continuous design-assistant. User guided, adaptive multi-objective search algorithms, re-entrant history records, parallelization of computation, and a user interface that allows control in a manifold and intuitive way.

## INTRODUCTION

### The problem of conflicting goals

A vast number of factors determine the final design of a building. Many of them as general conflicts between different targets. In all affairs, trade-offs have to be accepted in the context of contradictory interests. A complex network of dependencies and correlations between those interests exists, which restrains the designer in finding a good solution easily. Structural efficiency, daylight entry, constructive needs, or spatial concepts to name but a few can be major opponents in the design of a building that have to be analyzed holistically, especially in their relations to each other.

### Evolution of design techniques

The techniques and tools used to design a building have changed significantly over the last two centuries. For a long time, manual drawings and empirical design rules have been the only way of designing buildings and artifacts. With the rise of modern science, analytical approaches of quantification, and, later, also computer aided drawing technologies were applied in planning. Digital techniques allow the more flexible figuration of geometry and building information, as well as numerical methods enable quicker analysis of more complex systems. Finally, the last two decades show a shift

towards integrated, parametrically defined information- and analysis models. Via parametric associative modeling techniques, a representation is generated by a set of rules that are relating to each other (Schumacher, 2010). The term representation is used in an abstract way, meaning, e.g., a geometric model, BIM, solar data modeling, a structural model, etc. The data serving as the basis for these models can be altered easily, so a different alternative is simply produced by changing the input parameters. Ideally, all different models of a project are integrated into a single process of generation, so every change affects its related parts, resp. can be analyzed for compatibility.

### The modern design process and its potentials

Considering those two aspects, a fundamental exploitation of the concepts in design methods can be proposed. A design option can be set into relation to other alternatives in a goal-oriented way by comparing the individual trade-off states (see Fig. 2). To be able to make a justified decision, the designer needs to know about 1. the extreme trade-off configurations and 2. the relations that are causing the necessity for trade-offs. Traditionally, the process to gain this knowledge is relying on a designers' experience and, depending on the complexity and uncertainty of the task, additional try-and-error to unveil the relational nature of a design's sub-problems. We propose an assistant tool for parametric design that helps in this process, utilizing contemporary design tools, computational power, and findings in artificial intelligence that are novel to our time, thus firstly enabling this approach.

### METHODOLOGY

Design methods in architectural engineering are subject to constant change and evolution, thus the techniques and tools developed should be applicable to different [future] parametric modeling platforms. As a basis for our research, Grasshopper for Rhinoceros was chosen as a platform for flexible, intuitive and integrated modeling. The tasks to be tackled are:

1. Development of multi-objective search strategies in parametric design
2. Interface design
3. Performance of implementation

### TERMINOLOGY

To clarify the terms used in this paper, the important distinction of three instances of a parametric model is given here:

- **Parameter space** (genotypic space) is the range of parameters to a model and how they can be set, basically the input-side of the model.
- **Algorithmic definition** which contains the generative rules that translate the parameters.
- **Result**
  - **Phenotype** (e.g. visual output)      - **Objectives** (numeric fitness values)

Parameter space and objective space both contain a set of numeric values per alternative. In both of them, a solution can be seen as a point in n-dimensional space, where n is either the number of parameters or the number of objectives respectively. Since the parameter- and objective-space can correlate in highly complex ways, a main effort of algorithms for search and optimization is to overcome non-linear relations imposed by the algorithmic definition, which connects the parameter- to the objective-space (Bader, 2010; Zitzler, 2001).

### EXAMPLE 1

One of the main characteristics of architecture is that a designer always has to handle conflictive objectives, since architecture is always a trade-off of different targets. A very simple example can be seen in Fig. 1. A shell-like structure is clad with solar panels, so the geometry of the shell influences both the solar impact and the structural behaviour of the shell. To evaluate both in the design process, parametric solar analysis and calculation software is used in the generation process. Accordingly two objectives are defined: first goal is to maximize the annual solar impact, second is to minimize the structures' deflection under self-weight. It is evident that improving one goal downgrades the other. There is no best solution – it is up to the designers to choose one depending on the own perception. Fig. 1 shows three alternative trade-off designs and the graph of the examples' 2-dimensional objective space. A point in the graph corresponds to one solution while it clearly shows the weighting of the solutions' goal values *sun impact* and *deflection*. The set of points, and especially the extreme solutions 1 and 2 as well as an average solution 3 give a good impression of what the design space looks like, and what the options for the designer are within the current concept (respectively the current parametric setup). This example shows in a very basic way the principles of our work.

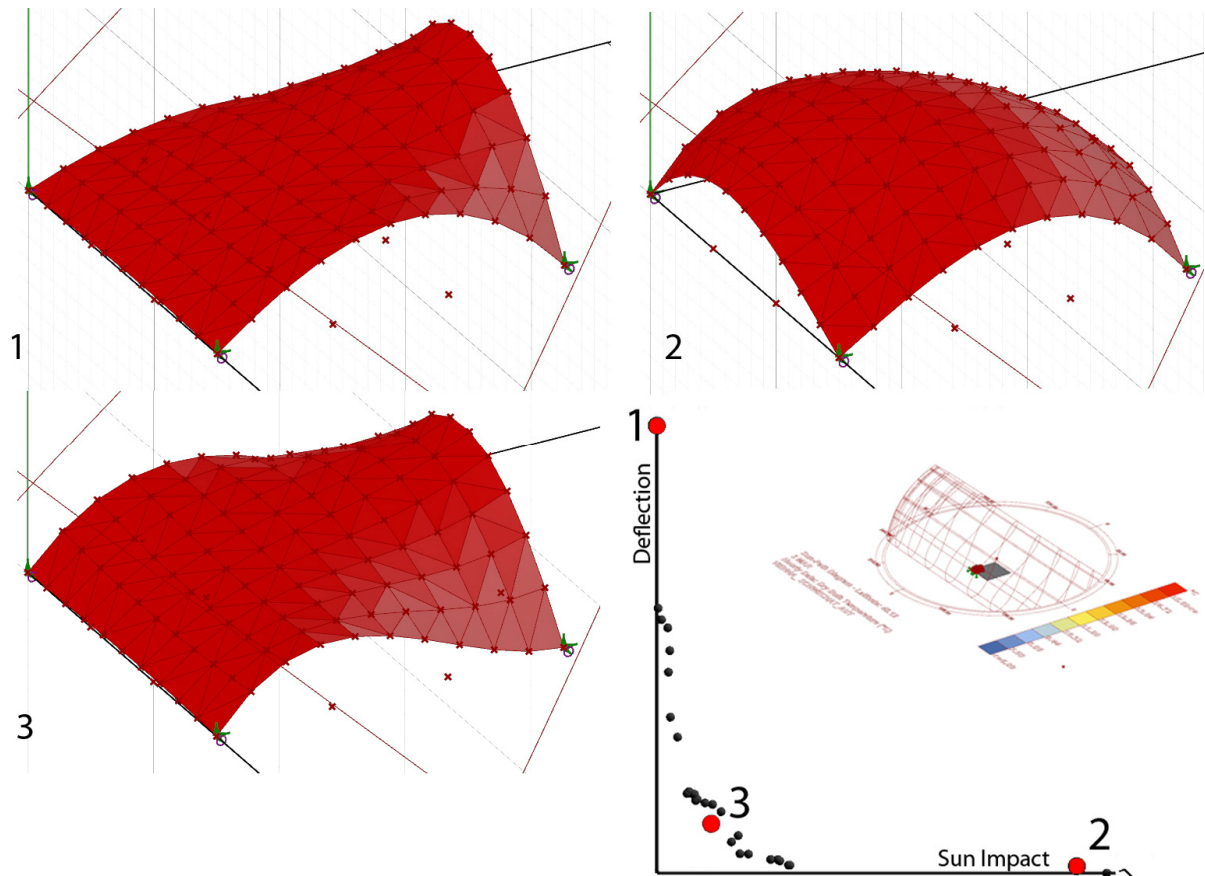


Figure 1: Conflicting objectives of sun exposure and bearing behaviour

#### SEARCH DIRECTION FOR SEARCH / FIND A SEARCH DIRECTION

From the introductory example it can be seen that a multi-objective optimization shifts from directed search to spanning a solution space. A numerousness of differing solutions is derived from the very beginning of the design process, such that firstly a possibility space without evaluation is spanned. Assessment and evaluation becomes feasible when objectives are defined. As in most cases comprehensive and accurate objectives are not defined at start, finding of these is an integral part of every design task. This finding of relevant objectives out of a number of numeric measures can be obvious, but especially for hard problems our tool can help in the identification of the important relations and goals.

#### THE SEARCH ALGORITHMS

To tackle generic multi-objective problems, most state of the art research proposes evolutionary algorithms that lever the computational power provided by recent technology, but simultaneously are highly applicable to hard search problems (Bader, 2010; De Landa, 2011; Zitzler, 2001). At their basis, a traditional genetic algorithm (GA) tries to improve a set of initial solutions over generations of trial and error, whilst informing the creation of new solutions with properties of the best tries so far. All techniques described in this paper essentially configure the environment of this GA, such as the translation of multiple goal-values, input data of user guidance, filtering rules, or which parameters and objectives to consider. Analogous to an open-ended design process, the evolution process of a GA never reaches a final state, but also at each point yields something that can be considered a result. Further, GAs are highly suitable to incorporate user interaction. There is little research on GAs operating with changing problem definitions, which is important to our concept of usage but difficult to implement in the GA itself. However, simply migrating certain parts of found solutions into new runs of the GA promises good continuance of the search information. These parts, e.g., can be parameter values that also are part of the new definition, or goal values that serve as attractor points in objective space.

The same technique of attractor points in objective space is used for the articulation of user preferences to direct the search. When a search algorithm has reached certain coverage of the design space, the user might want to incorporate his preferences to the process in order to get more refined solutions around the choice. Such methods have been developed and tested successfully for interactive evolutionary computation in, e.g., Thiele (2009).

## COMPARING MULTI-OBJECTIVE SOLUTIONS

As mentioned, different alternatives can be compared looking at their trade-off character. This essentially is comparing points in n-dimensional objective space, where n is the number of chosen goals and one point corresponds to an individual solution. Pareto dominance (after Vilfredo Pareto 1848-1923) is a basic way of determining the quality of a multi-objective solution against another. Fig. 2 shows a set of solutions to a two-objective problem, with the axes showing the respective fitness values. The solution A is said to Pareto-dominate a solution B if A is at least as good as B in all objectives, and superior to B in at least one objective. Pareto-optimality of a solution means an optimal trade-off between two or more contradicting objectives, where one goal cannot be improved without degrading the others. The theoretical sum of all Pareto-optimal solutions is called the Pareto-front, which can be thought of as a hypersurface in n-dimensional objective space. Fig. 3 illustrates what is called a non-dominated set of solutions or approximation of the Pareto non-dominated front to a two-objective problem.

A multi-objective optimization algorithm in the best case yields a set of Pareto-optimal solutions that have the following properties:

1. They are near to the real Pareto front
2. They are diverse and evenly distributed
3. Many extreme solutions are included

While those criterions are relatively easy to understand regarding the practical advantages, they are not trivial to implement. The Pareto principle serves as the basis for a sophisticated strategy to filter the solutions best suitable for further processing in each iterative step. The Hypervolume Indicator (HI) up to now is the only measure to combine all three of the abovementioned characteristics (or goals) of an evolutionary multi-objective optimization in one mathematical formulation (Bader, 2010).

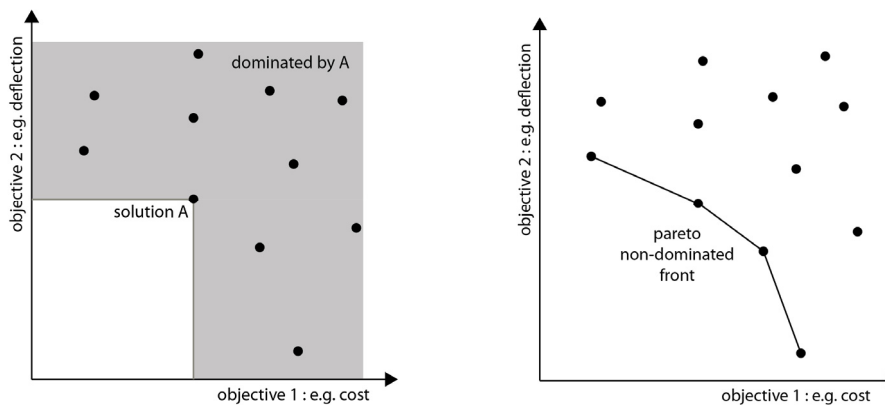


Figure 2: Pareto-dominated space of solution A      Figure 3: Pareto-non-dominated front of a set

## ALTERNATIVE COMPARISON STRATEGIES

The objective values are the most obvious criterion for the application of a filtering algorithm (Fig. 4 right). However, there are other logic levels that can be targeted with filtering strategies.

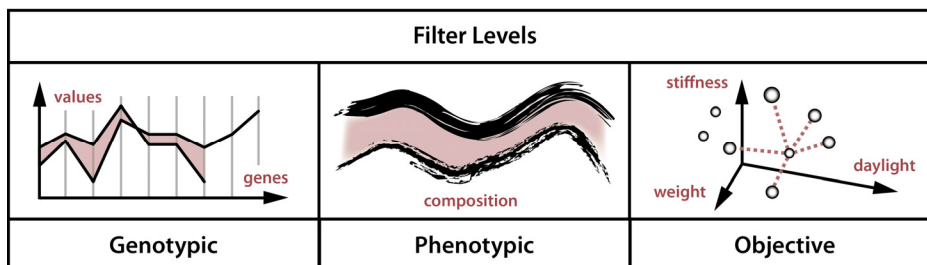


Figure 4: Levels of interaction with / filter levels of the design assistant

The genotypic space (Fig. 4 left) by now mainly is targeted by the Genetic Algorithm. It modifies the parameters and compares different configurations by a single accumulated fitness value. As mentioned, parameters can be interdependent highly non-linear, which is one of the main reasons to apply algorithmic optimization at all. Nevertheless, secondary systems to identify sensitivity-relations between parameters and objective- or phenotypic characteristics can be very helpful in practical work with large problems. Hereby, instead of explicit sensitivity analysis, a strategy to analyze the parameter-goal relations with the solutions produced during the search seems most suitable.

A general concept to interact with the phenotypic level (Fig. 4 middle) is not possible to define, since it is individual to every problem, and how the phenotype is defined. Basically incomputable measures such as aesthetics or composition are one of the main reasons to include the designer as the final decision maker in our work. Nevertheless it should be targeted of how categories and techniques to effectively handle phenotypic relations could be implemented for recurring architectural tasks.

## EXAMPLE 2

A case study for a viewing platform shows the functionality and usability of a sophisticated multi-objective search engine. A supporting structure for a viewing platform, which is 22 m above ground, is designed (see Fig. 5). There are only a few areas where supports for the platform's columns can be placed (Fig. 6). At level +6.00 m a smaller platform is scheduled, which must not intersect with a column at all. All generated profiles are checked against each other, whether their smallest distances fall below a certain value. If so, their nearest points are merged and the members are connected to each other. Additionally the number of members is not fixed a priori.



Figure 5: Rendering of a design alternative

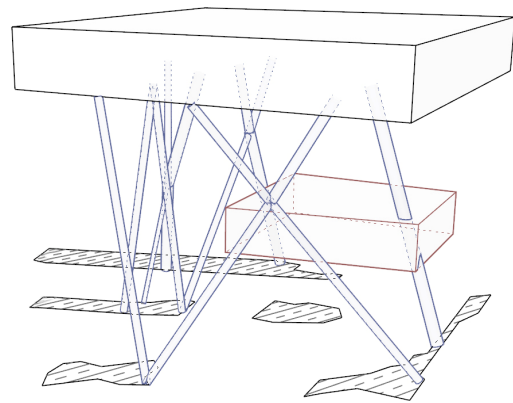


Figure 6: Limited support areas for platform

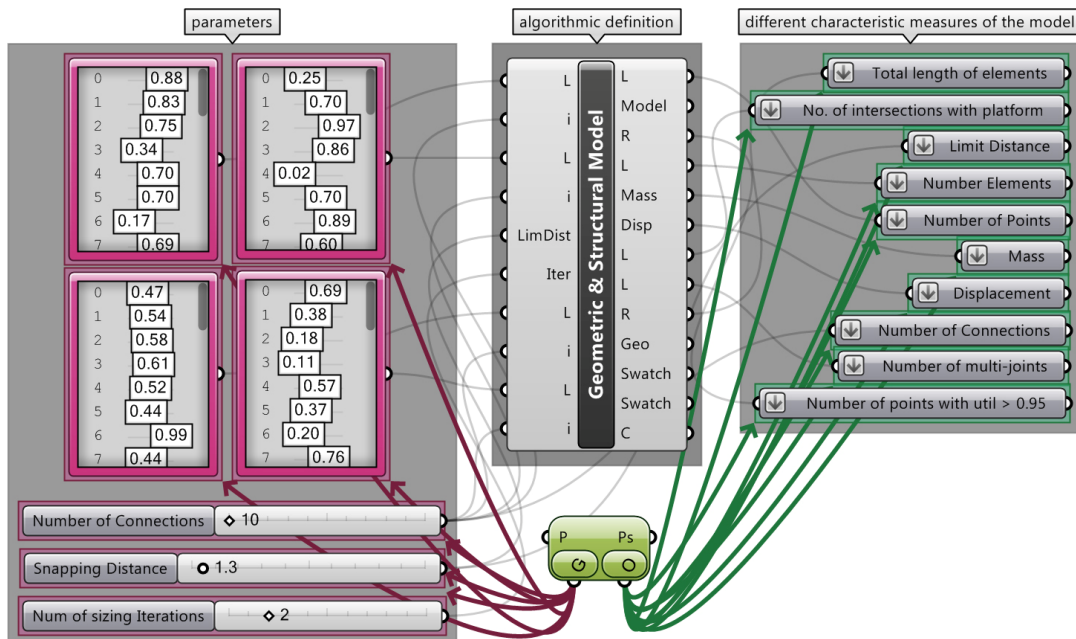


Figure 7: Parametric setup for example 2, search assistant connecting to all numeric in- and outputs

In Fig. 7 the principle parts of the example's parametric setup are shown. The setup is grouped into parameters, algorithmic definition and potential objectives according to the differentiation mentioned above. The model includes the generation of line-geometry which serves as the basis for integrated parametric finite element analysis to determine realistic structural behaviour. Each structural member is sized for its maximum allowable utilization under the influence of multiple load cases. Fig. 8 shows a collection of numeric measures that can be obtained easily from the model. These measures all have different significance for exploring the final design. Deflections and self-weight after the size-optimization are values that reflect the structures' load bearing capacity and efficiency. To assure the functionality, collisions with the platform have to be zero. The number of joints and elements are indicators for the feasibility. At the beginning of the design process there is little knowledge about the significance of the individual measures and their interrelation. For this reason, the design assistant firstly considers every numeric measure in the model as a potential objective. It tries to find solutions that by default minimize all of them, but at the same time maintains extreme solutions of each measure. This way, an overview of possible configurations is obtained. Fig. 10 shows the interface to choose which objectives are displayed in the solution explorer, so different relations of objective dimensions can be assessed (see section below). Also, a logarithmic graph of each goal value's development during the search is shown.

The following categorisation of obtained insights could be made:

- Variations of the phenotype as visually different alternatives
- Dependency relations, determining or/and influencing
- Identification of design drivers and design goals
- Identification of hard constraints
- Near-linear correlations between certain objective-values

With these findings, the raw listing of Fig. 8 then can be transformed to a visual expression of the design's character, as exemplified in Fig. 9 (Kilian, 2006). By looking at this process not as a singular event of analysis, but as a continuous development of both the measures listed in Fig. 8 and the relations illustrated in Fig. 9, our concept of enhanced computational design assistance is apparent.

Fig. 11 shows three different alternatives of the final setup in generation 149. Parameters and objectives are configured as illustrated in Fig. 9 and Fig. 10. Axe 1 (red) shows the total number of joints, axe 2 (green) the total structural mass, axe 3 (blue) the maximum structural displacement, the colour-range from green to red shows the number of multi-joints, and the size from small to big shows the number of over-utilized sample points within the structure. The solutions shown have been marked as preferred for the subsequent process, they are marked blue in the solution explorer.

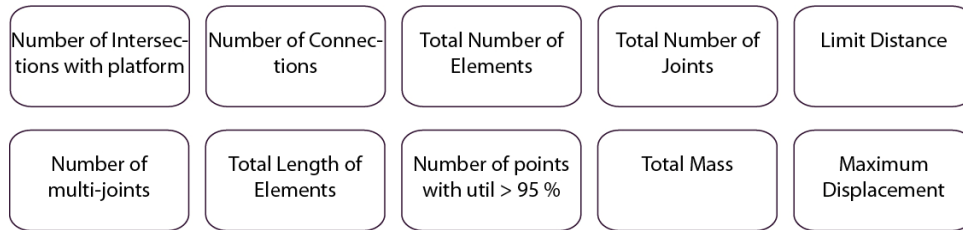


Figure 8: Numeric measures at the beginning of the process

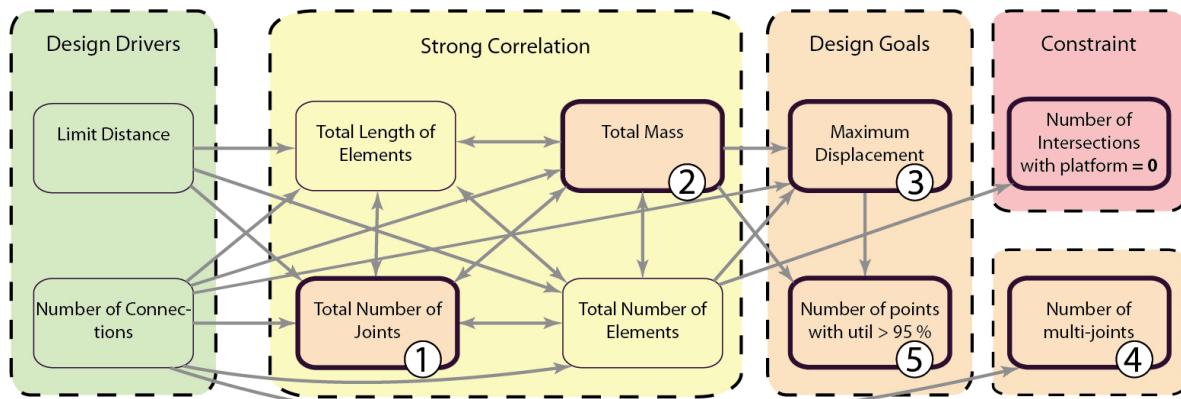


Figure 9: Dependency graph and final categorisation of numeric measures in example 2; final objectives 1-5 and hard constraint



Figure 10: Interface for objective-configuration and logarithmic graph of objective values over time

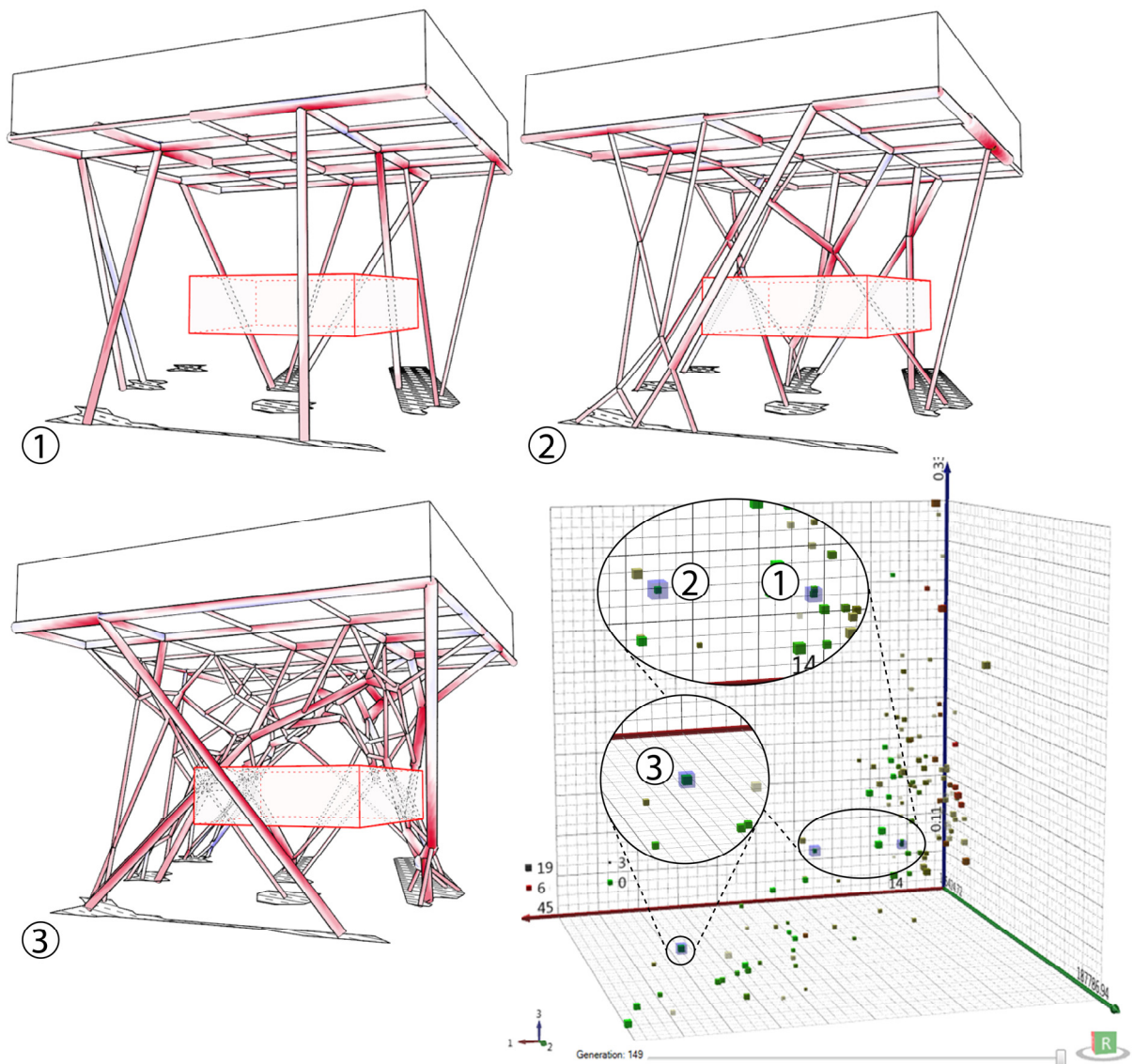


Figure 11: Different alternatives of example 2 in its final setup of parameters and objectives.

## THE INTERFACE

Target of our interface design is to make the process of algorithmic search and optimization be available as a background process, which can be considered optionally. Long-term goal hereby is the implementation of a robust parallelisation method to free the user from computational loads during his normal work. This is partly accomplished for the current platform of Grasshopper3D, but is still in testing phase. Parallelisation like in our use-case is expected to be technically less demanding on future platforms such as Autodesk's DesignScript.

To be able to assess solutions in an intuitive way, a solution explorer in the form of a navigable 3D viewport is implemented. Three objective dimensions are shown on the x-, y-, and z-axis, while a fourth and fifth dimension optionally can be displayed as different colouring and sizing of the solution points. Tendencies of the search process are shown as dissolving yellow solution-points of the past generations, so an impression of the convergence-speed and direction of the search progress is given.

Thinking of solutions as points in a multi-dimensional space, where the coordinates are their objective values, a display-axe can be assigned to each objective according to the user's needs for analysis. The user is able to decide during the process which objective is

- shown on which axe in the 5D-viewport,
- and if its diversity should be enforced.
- if it is considered for optimization,

Regarding the integration of our tool, the connection to the platform's usual workflow is important. As Grasshopper demands an explicit modelling of the data-flow, possibilities for even more interactivity are limited as opposed to platforms that allow the immediate recording of a history-graph while modelling. Nevertheless, as a first step the tool recognizes possible parameter- and objective values within the model (Fig. 7) and lists them as shown in Fig. 10.

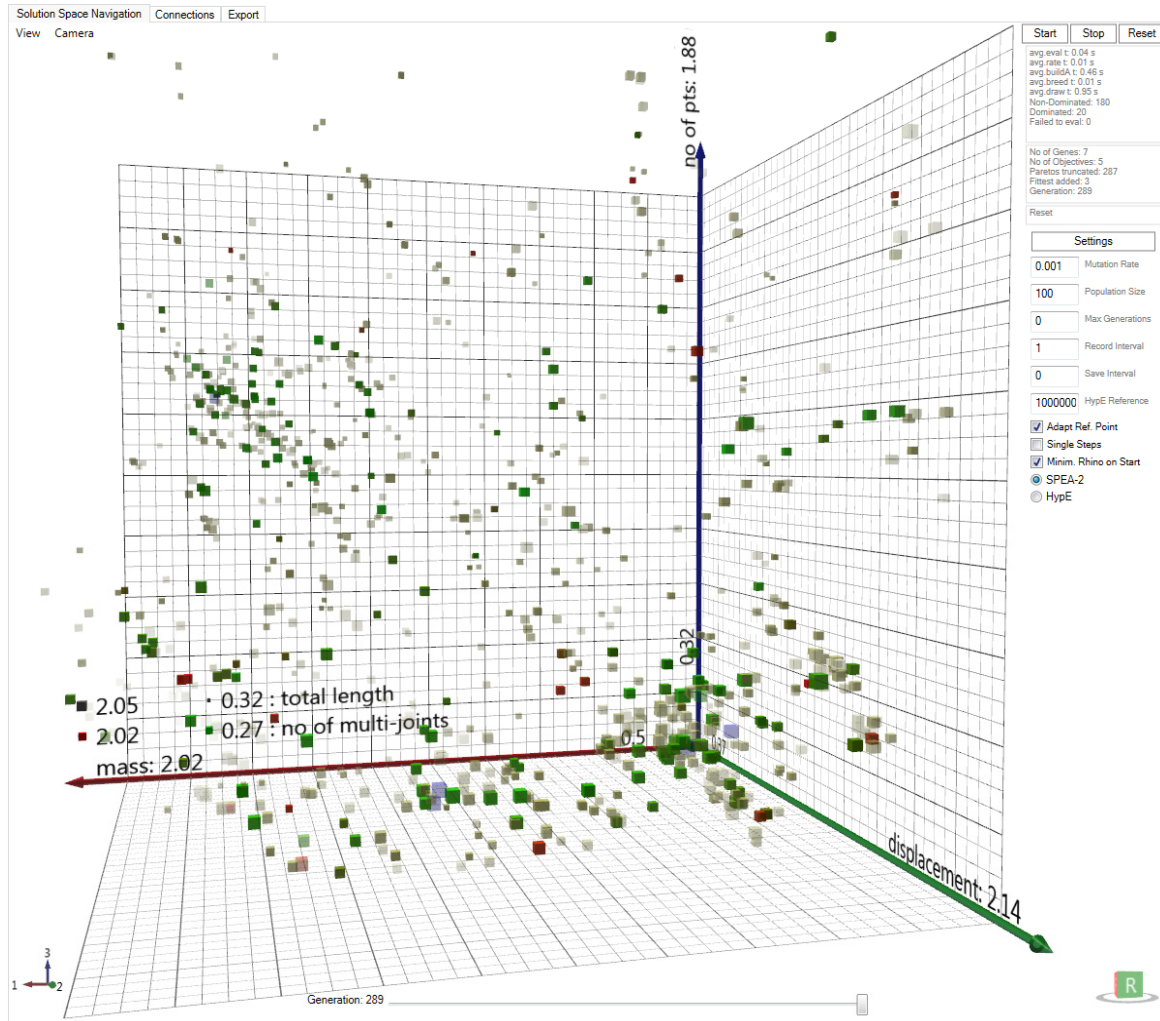


Figure 12: Densification of search around the user's preferred solutions, marked blue

Besides the basic selection, parameters and objectives can be modified only for the algorithm's consideration. An interface of this kind is also serving as the basis for future modules of sensitivity analysis. The parametric setup as well as all settings of our tool can be altered in real-time, which convey an additional experience of interactivity. Changes to the parametric setup are handled as stated in the section above, so as much information as possible is transferred from one design setup to the next one.

User-guidance of the evolutionary process itself is accomplished by letting the designer mark single solutions as preferred or un-preferred in the viewport. All subsequent evaluations will take objective-similarity to marked solutions into account, so densities will evolve according to the user's choices as shown in Fig. 12.

Every step performed by and with the design assistant is automatically recorded in a re-entrant history tree, so turn-offs in the design process are traced also for later analysis. A visual illustration of the history of a design (Fig. 13) should in the future support the overview of a design process even more.

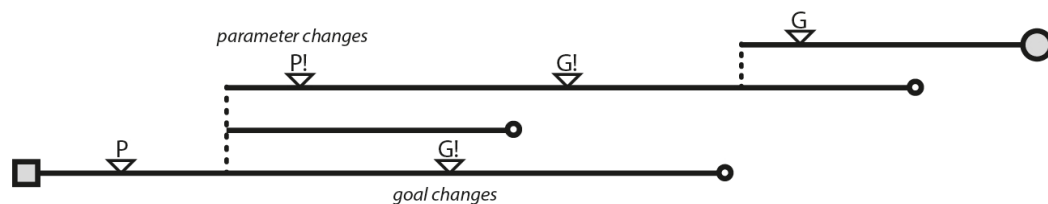


Figure 13: Proposed scheme of history records in the form of branching timelines

## CONCLUSION

*'The first industrial revolution showed us how to do most of the world's heavy work with the energy of machines instead of human muscle. The new industrial revolution is showing us how much of the work of human thinking can be done by and in cooperation with intelligent machines. Human minds with computers to aid them are our principal productive resource. Understanding how that resource operates is the main road open to us for becoming a more productive society and a society able to deal with the many complex problems in the world today. The tools now being forged for aiding architectural design will provide a basis for building tools that can aid in formulating, assessing, and monitoring public energy or environmental policies, or in guiding corporate product and investment strategies.'* (Simon, 1986). Having received a Nobel Prize for fundamental research in decision making and rationality, Herbert Simon coined the term of artificial intelligence in 1955 with the development of the 'Logic Theorist', after that attempting research on a 'General Problem Solver'.

As one of the most generic contemporary approaches to computationally tackle problems, the usage of Genetic Algorithms is chosen. Some precedents facilitating evolutionary optimization within Grasshopper do exist, however none of them consider the problems of multi-objective settings, incorporate user-interaction or try to overcome the computational limitations for professional application. A multitude of recent research projects in computer science, philosophy, and engineering indicates the potential still lying in the development of concepts, algorithms and platforms regarding multi-objective evolutionary optimization and search.

## REFERENCES

- Bader J.: *Hypervolume-Based Search for Multiobjective Optimization: Theory and Methods*, PhD thesis at ETH Zurich, published by CreateSpace Independent Publishing Platform, 2010
- De Landa M.: *Philosophy and Simulation: The Emergence of Synthetic Reason*, Continuum International Publishing Group, 2011
- Kilian A.: *Design innovation through constraint modeling*, PhD thesis, Cambridge, MA: MIT, 2006
- Schumacher P.: *The Autopoesis of Architecture*, Volume I and II, John Wiley & Sons, 1st Edition, 2010
- Simon, H.A.: *Decision making and problem solving*, Research Briefings 1986, National Academy of Sciences, Washington DC, 1986
- Thiele L., Miettinen K., Korhonen P. J., Molina J.: *A Preference-based Evolutionary Algorithm for Multiobjective Optimization*, Evolutionary Computation Vol. 17, No. 3, p. 411-436, MIT Press, 2009
- Zitzler E., Laumanns M., Thiele L.: *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*, Swiss Federal Institute of Technology (ETH) Zürich, TIK-Report 103, 2001