

# State on the Art of Computational Tools for Conceptual Structural Design

Anke ROLVINK<sup>\*</sup>, Caitlin MUELLER<sup>a</sup>, Jeroen COENDERS<sup>b</sup>

<sup>\*</sup>Delft University of Technology, White Lioness technologies  
van Diemenstraat 118, 1013CN Amsterdam, the Netherlands  
ankerolvink@wlncs.com

<sup>a</sup> Massachusetts Institute of Technology

<sup>b</sup> Delft University of Technology, White Lioness technologies

## Abstract

This paper presents a review of existing research, projects, developments and applications in the domain of design tools for conceptual structural engineering. The availability of these tools and research into software for conceptual structural design stages has shown a number of interesting developments over the last past few years. The purpose of this investigation is to understand the requirements for software for the early stages of structural design. It investigates the current conceptual design practice, discusses a number of novel trends, and characterizes the relative effectiveness of the available technologies in relation to the nature of the early design stages.

**Keywords:** Structural engineering, computational tools, conceptual design, software development, collaboration.

## 1. Introduction

The conceptual design phase is one of the earliest phases of the design process. The act of conceptual design deals with the establishing of a number of base principles, reasoning, argumentations and justifications that will act as a starting point and guidance for the next phases in the design. Generally, this process starts with the generation of different design alternatives based on a set of initial requirements. These processes are often complex, creative and iterative in nature. Using software tools during these design stages can potentially have significant influence on initial design decisions and key parameters in the design process as these tools can give a structural designer more control over the design and the ability to quickly generate and compare different alternatives.

Despite the fact that the composition of design models, communication and visualisation is very difficult during conceptual design stages, the impact of decisions made during these early stages is often very large and influential during the rest of the design process. Decisions are made while there is still uncertainty about the potentially large consequences of these decisions for the rest of the design process. It is important to acknowledge the consequences of decisions or to get a sense for their magnitude on the key drivers, such as cost, material quantities, energy consumption, etc. During later stages it becomes very costly and sometimes even impossible to compensate for poor design choices made at the beginning of the design process. This stands in contrast to the influence of decisions in later design phases. Figure 1 shows the general trend of the influence of decisions made throughout the design process, compared to the number to tools available for the structural engineer.

## 2. Design tools

Most current computational tools for structural design and engineering are not developed for early design stages as they require more information and input than is available during these design stages. Finite element analysis applications, for example, are more focussed on the act of analysis rather than the generation of design alternatives. Another example is BIM software (Kensek [12]), which offers the possibility to assemble highly complex 3D models and aims to support the entire process from early design to construction. However, in

practice the application of BIM software is limited to the later phases of engineering and to the early phases of construction. This is due to the fact that the object-oriented nature of BIM software, which requires 'objects' to be known and modelled even though in the early design stages these objects (beams, columns, etc) are not decided upon.

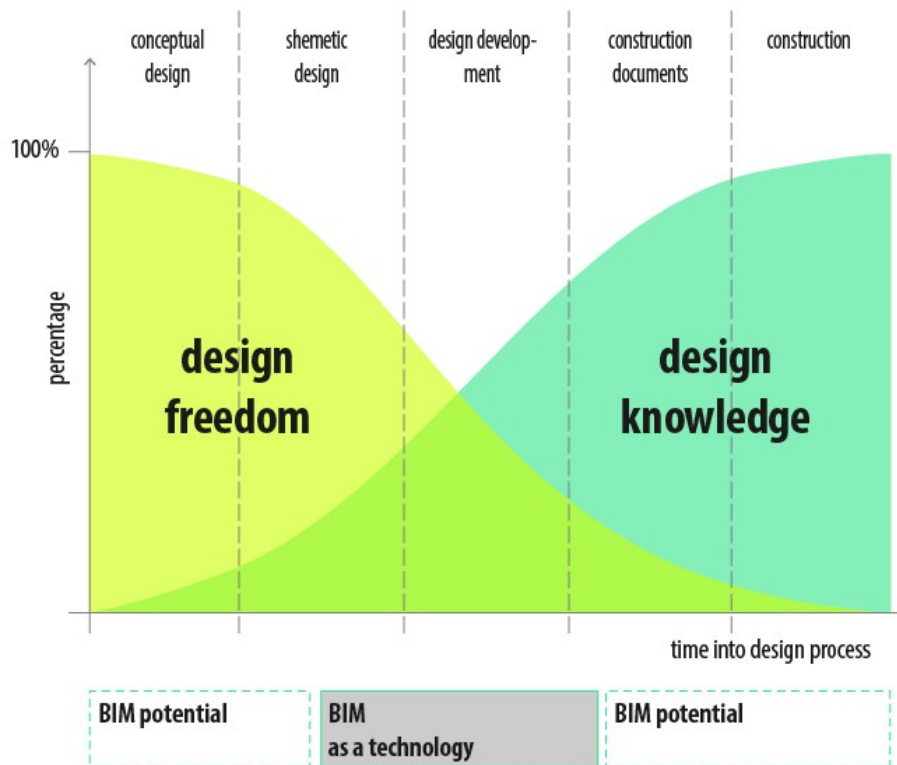


Figure 1: Relationship between design freedom and design knowledge in building design projects. The most opportunity for design impact and creativity occurs during conceptual design, but structural considerations usually enter the process far later. This limits the ability of structural engineers to contribute impactful ideas in the design process (after Fabrycky [7] and Paulson [17]). It also shows the availability of BIM software during these design phases and the possible area's for new developments (from: Coenders [4]).

Moreover, a BIM model is set up to approach the as-built situation as closely as possible, as it documents what the contractor needs to build. A structural engineering model, on the other hand, needs to represent more than the idealised truth and also needs to take alternate scenarios into account, since structural engineers are responsible for the safety of a building and accidents usually happen in situations which are not foreseen in the idealised model. As such, the purpose of a BIM model and an analysis model are different. On one hand, the BIM model needs to be as close as possible to the idealised desired situation and therefore represents only one outcome, namely the end situation. On the other hand, the main purpose of the analysis model is to give an engineer insight in the structural behaviour of a design and prove that the design is structurally sound. In order to prove this, several models are required which do not represent the expected, idealised reality, but that incorporate imperfections, such as differential settlement of the foundation, lack of reinforcement, material imperfections and tolerances.

Recent developments in the field of structural design and engineering attempt to implement new approaches for early design stages, which offer more flexibility and the ability to compose, generate, explore, or communicate design alternatives. They generally do not produce highly accurate solutions, but aim to enable the quick generation and assessment of different alternatives, and to give insight(s) into the impact of decisions. The next subsections give an overview of a range of novel developments in this area.

## 2.1. Graphic static tools

The first development is graphic statics, which is a graphical, as opposed to numerical, method of calculating internal forces in axially-loaded structures such as arches, cables, and trusses. Developed from fundamentals established in the early 1800s, the method was formalized in 1866 by Culmann in his book *Die graphische Statik* [6]. A recent book by Allen and Zalewski [2] gives an overview of the method and applies the method to conceptual design problems. Engineers made widespread use of this technique for both design and analysis until the 1960s, when numerical methods gained prominence due to the increasing calculation power of computers.

Recently, there has been renewed interest in graphic statics because of its simplicity and power. Several researchers have developed computational implementations that allow designers to manipulate structures in real time and observe how internal forces change through the force polygon. One pioneering example is Active Statics, an online tool that contains seven interactive design examples (Greenwold and Allen [9]).

Further advancements are evident in eQULIBRIUM, an online interactive tool that illustrates graphic statics techniques on a wider and more complex range of example problems (Van Mele *et al.* [22]). Additionally RhinoStatics is a plug-in for the 3D modeling software Rhinoceros which performs graphics statics analysis of structures drawn by users (Shearer [20]).

While constituting an important step forward, this class of tools is limited in several ways. First, graphic statics techniques are restricted to relatively simple problems, generally two-dimensional and statically determinate. Second, with the exception of RhinoStatics, most currently available graphic statics computational tools work only on pre-set examples, and are generally not flexible enough to provide feedback on a design problem presented by the user. Finally, while graphic statics can be used to generate structural design concepts, as opposed to simply analysing them, the process is generally manual. To move beyond this, tools should offer a way to synthesize new geometries using structural principles implicitly.

## 2.2. Form-finding tools

One compelling way for designers to explore this synthesis is with a set of tools that employ form-finding techniques. These tools use various algorithms to discover equilibrium configurations for spatial structures that contain little or no bending, and move beyond mere analysis in important ways. Key examples of such tools include CADenary, a particle-spring tool for exploring pure-compression and pure-tension structures (Kilian and Ochsendorf [13]), RhinoVAULT, a tool for designing compression-only structures using thrust network analysis (Rippmann *et al.* [18]), and a web-based numerical form-finding tool from Princeton's Form Finding Lab that uses dynamic relaxation for shell design (Adriaenssens [1]). These tools move beyond analysis to generate high-performing design options. However, they only work for a narrow range of structural typologies, and are not generally applicable to problems beyond membrane and shell structures. It is therefore often necessary to consider a broader approach that can be used systematically on a range of problem types.

## 2.3. Design optimisation

Optimisation, the mathematical process that systematically computes the best-performing solution according to one or more quantitatively defined objectives, can serve this broader role in theory. Structural optimisation seeks structures that are maximally efficient in terms of material volume, stiffness, dynamic behaviour, or other related criteria. However, despite its rich academic history, structural optimisation has had relatively little impact on structural engineering in practice. One important counterexample is the work of SOM's William Baker and his collaborators, who have worked to apply structural optimisation to real design projects in new ways (Stromberg *et al.* [21], Baker *et al.* [3]). However, their efforts remain an exception in the broader building engineering and design industries.

Fundamentally, the limited use of optimisation can be attributed to an inherent difference in goals between optimisation and the conceptual design of buildings. While optimisation is necessarily a convergent process, or one in which an iterative and systematic algorithm converges upon a single solution, design is decidedly divergent. In design, it is recognized that a variety of significantly different yet suitable solutions can be found from a single starting point. Moreover, the exercise of mathematically formulating objectives and constraints is difficult or impossible in the design of buildings. Many important goals and requirements are qualitative, or even subjective, such as visual impact, spatial experience, contextual fit, and overall architectural value. Since most structural design cannot occur in the absence of architectural goals, this presents a significant challenge.

In addition, the conceptual design process for buildings is often one of discovery: designers do not know all of their objectives and constraints at the beginning of the process, but develop them as they explore design

possibilities. The designer's interaction with the process of evaluation and iteration is critical. In contrast, standard optimisation is a relatively rigid and automated process in which goals and requirements must be enumerated completely at the start. Unlike the human design process, optimisation on its own cannot handle unformulated objectives and constraints.

Finally, most structural designers lack intensive training in optimisation, and there are few tools or approaches available that make optimisation accessible to non-experts. Furthermore, optimisation tools that do exist are often text-based or severely limited in their graphical interfaces and interoperability, and often rely on piecing several pieces of software together. Human designers are necessarily highly visual, and can process and evaluate information much more effectively and fully when it is presented graphically. Therefore, in order to be of practical use for designers, tools that use optimisation should be easy to use, integrated, and strongly graphical.

#### **2.4. Interactive evolutionary exploration**

One compelling approach that combines the spirit of traditional optimisation with features that address its challenges is interactive evolutionary exploration. This approach makes use of an interactive evolutionary algorithm that considers both formulated, quantitative structural design goals as well as important but qualitative input from the human designer. The algorithm functions like a traditional evolutionary algorithm for optimisation, such as a genetic algorithm, with the important distinction of an interactive step in the iterative cycle. In each iteration, new design alternatives are randomly generated, structurally evaluated via the objective function, and ranked to identify candidate parent designs for the next generation. At the interactive step, the user contributes to the final selection of parents, based on preferences that are difficult or impossible to include directly in the optimisation objective function, such as aesthetics and constructability.

Tools that use this method are powerful because they can generate a broad range of high-performing design options that more closely fit the full range of design needs of the user. An important early example of such tools is the IGDT (Intelligent Genetic Design Tool) developed by von Buelow, originally implemented for exploring the geometry and topology of planar trusses (von Buelow [24]). ParaGen, a more recent version, incorporates a general parametric formulation to reduce design variables and enhance the richness and variation of explored designs (von Buelow [25]). Another example of interactive evolutionary exploration is structureFIT, a web-based design tool that includes flexible problem setup, increased means for designer interaction, and real-time analysis for post-processing refinement (Mueller and Ochsendorf [15], Mueller [16]).

While these tools overcome many of the problems of traditional optimisation, they still present a few issues. First, they can be slow for large and complex problem types, a challenge that can be addressed to some degree by approximation of the objective function. Second, their generated design concepts are often similar, or at least in the same family of structural ideas, due to the limitations of the parametric formulation. A broader range of results is often desirable in conceptual design. Finally, while the interactive evolutionary algorithm can process which designs are selected by the user, it has a more difficult time interpreting the reasons for selection. This means that the appeal of designs due to emergent properties not directly formulated as design variables, such as global geometry, is not directly registered. Subsequent generations based on selected designs can therefore lack the desired characteristics, despite the interactive process.

#### **2.5 Parametric and associative design**

Parametric and associative design regards objects, such as structural elements of buildings, as a series of user-defined changeable parameters and derives other objects through a set of user-defined changeable associations. This forms a logic in which the designer can easily change the parameters, such as dimensions, of an object, resulting in automatic updating and regeneration of the object following that same logic. A design process set up in this way benefits from the ability to explore design spaces while maintaining inherent logic.

Several papers from the past years propose the use of a parametric and associative approach during the early design stages. In 2008, Van de Straat et al. [23] published a paper about computational design strategies for the early phases of structural design. They state that: "A lack of design strategies for application of advanced computational methods and techniques in the design process exists. And as a result, the full potential of software for structural design has not been reached and therefore large opportunities exist for computation in structural design."

Holzer et al. [10] performed research on this topic. They state that "By linking parametric design to structural analysis and optimisation, architects and structural engineers can explore design in the conceptual design phase through informed geometry alterations. [...] From an architect's perspective, the immediate visualisation of

structural feedback, provided by the structural engineers proved valuable to understanding the effects of changes which might otherwise only be driven by aesthetic considerations.”

The parametric and associative nature is an effective and efficient way to generate a lot of alternatives and adapt them in (near) real time. Therefore, parametric and associative design approaches provide a convincing alternative for existing conceptual structural design approaches. Besides the above mentioned advantages using a parametric and associative design approach also offers new opportunities such as sensitivity studies, optimisation, etc.

Over the past years several tools have been developed which integrate structural modelling in a parametric environment. An example of such a tool is Karamba. This is a parametric engineering tool (Karamba [11]), developed as a plugin for Grasshopper and provides the capability to set up structural models in a parametric environment (Grasshopper [8]). Thereby enabling its user to combine parametric modelling and finite element analysis. Another example of a development which aims to combine structural analysis and parametric modelling is Geometry Gym (Mirtschin [14]). This toolbox comprises of several plugins which link parametric models in Grasshopper to structural analysis or BIM software.

One of the disadvantages of parametric and associative modelling is the difficulty of changing the design logic at the end of the design process. If something has to change in a relationship between two parameters, it will usually have consequences for the whole model and therefore be hard to accomplish. Another problem in the current use of these tools is the direct link between the geometry modelled in a parametric environment and the structural model. This makes it hard to set up different structural scenarios for a design concept as these models only represent the perfect model. In structural design, it is important to set up different models and scenarios which also include imperfections.

## 2.6. Dashboard-based design tools

Dashboard-based design tools are tools that attempt to support the design process by providing a dashboard or set of tools which the user can use to assemble and analyse models. This approach does not take the technology as a starting point but studies the process of design and tries to adapt existing strategies combined with new strategies to fit design and technology closely together. The goal is not to have an all-comprising design model, but assumes a collection of tools that can be chosen by the engineer to support the design process (Coenders [5]).

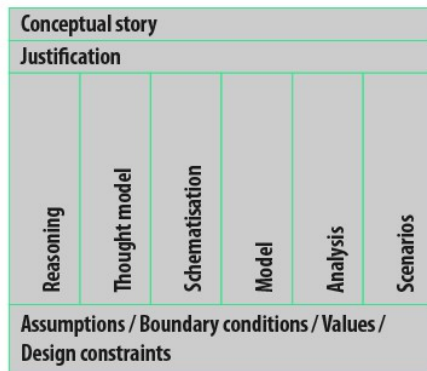


Figure 2: Conceptual figure showing how elements of design are based on boundary conditions and form the foundation of the justification and conceptual story.

The structural engineer will get an understanding about the design by setting up various alternatives and step-by-step justification and reasoning through the concepts by understanding the relationships between the scenarios the design will encounter and its corresponding responses. This is based on the conceptual design thought model as shown in Figure 2. It starts with a definition of the boundary conditions, requirements and design constraints. These form the base for the generation of design alternatives which involves reasoning, thought, schematisation, modelling, analysis and defining scenarios. These elements of the structural design process subsequently form the foundation for the evaluation of the different design alternatives. Justification and forming

the conceptual story conclude the early stage design process. The aim of setting up the design justification is to build confidence in the solution, used methodologies, tools, etc. The design story forms the basis for later design stages and the engineer will always be able to refer back to earlier decisions made in the design process to control or verify the developments in the design against the earlier documented benchmark.

An example of a development with such an approach is StructuralComponents (Rolvink [19]). The goal of StructuralComponents is to document the design process. In current practice, this happens on paper and it is difficult to capture all decisions made by the engineer for use later in the design process. The goal is to replace the paper but not of the value of the process. Therefore, StructuralComponents tries to combine the advantages and intrinsic value of the engineer's process on paper with the advantages of parametric and associative design. This requires (amongst others) the development of different scenarios, various levels of detail and simplification and modifications of the 'design' to include imperfect situations. StructuralComponents aims to provide a dashboard-based approach which offers the user various tools and methods to accomplish this.

One of the limitations of StructuralComponents is that it is based on existing parametric and associative application, such as Grasshopper and therefore only single-directional. In reality the design process requires a bi-directional interaction between the 'design' – a representation of the object which we desire as an outcome – and conceptualisations that justify the design. A solution for this is to combine the toolbox with NetworkedDesign, a next-generation modelling infrastructure that supports more advanced definitions of logic and application of multiple solvers of logic. This will solve some of the limitations that currently exist (Coenders [4]).

### 3. Summary of novel developments

The previous section gave an overview of novel developments in the field of computational tools for conceptual structural engineering. Table 1 shows a framework of criteria on which the different developments can be judged.

|  | <b>GST</b> | <b>FFT</b> | <b>DO</b> | <b>IEE</b> | <b>PAD</b> | <b>DDT</b> |
|--|------------|------------|-----------|------------|------------|------------|
| <b>Speed</b>                           | 5          | 3          | 3         | 3          | 4          | 3          |
| <b>Interactivity</b>                   | 3          | 2          | 1         | 4          | 5          | 5          |
| <b>Informative</b>                     | 4          | 3          | 2         | 3          | 3          | 5          |
| <b>Easy to use</b>                     | 3          | 3          | 2         | 4          | 4          | 3          |
| <b>Overview/ Feedback</b>              | 4          | 3          | 3         | 4          | 3          | 5          |
| <b>Insight</b>                         | 3          | 4          | 4         | 5          | 2          | 5          |
| <b>Open to many typologies</b>         | 2          | 1          | 4         | 4          | 5          | 5          |
| <b>Generation of many alternatives</b> | 2          | 2          | 1         | 5          | 4          | 4          |
| <b>Alternative comparison</b>          | 2          | 1          | 1         | 4          | 4          | 4          |
| <b>Adaptability</b>                    | 1          | 1          | 2         | 3          | 3          | 5          |

Table 1: Table summarizing the novel developments reviewed in this paper, with numerical scores in each category out of 5; GST = Graphic Static Tools, FFT = Form Finding Tools, DO = Design Optimisation, IEE = Interactive Evolutionary Optimisation, PAD = Parametric and Associative Design, DBD = Dashboard-based Design Tools.

### 4. Discussion

With the growing development of computational design tools for conceptual structural design phases, new possibilities arise. Generating and assessing an extensive variety of design options in a short time supports the creative process that is often bounded by time limitations and greatly increases the design flexibility throughout

the entire process. By integrating structural design knowledge into these tools, it will be possible to better support early design decisions for conceptual structural design. However, one of the difficulties that arises is the implementation of creativity in design tools. It seems almost impossible to develop software that supports the way structural engineers design. On one hand, the form-finding and optimisation tools as discussed in the previous section do not support the creativity of the structural engineer, since they generate alternatives based on a set of given boundary conditions. On the other hand, the tools based on a parametric and associative approach or the dashboard based design tools do support the creativity of an engineer but are often very limited to specific design problems and not applicable to general conceptual design problems. Design challenges in the building industry are often unique and therefore it is difficult to develop generic tools which can be used by a structural designer to support these challenges.

Currently, it is also very hard to maintain data persistence during the different design stages. It is impossible to capture all the data, logic and knowledge in such a way that during a design stage, this knowledge can be used subsequently without loss of information. During every transition information gets lost, which makes it harder in later stages to form the design justification.

It is also important to note that it is not enough just to develop new and advanced design tools. Their rate of success depends on the adaptation and recognition within the structural engineering practice. The use of these technologies requires a change in current working methodologies and the structural engineer may need to innovate in order to use and understand the new technologies. The use of new technologies also requires confidence in them and therefore an engineer will need to know what these tools do and how they work.

As mentioned in previous sections, design projects are often unique and therefore design tools need to be adapted and extended to fit a specific project. Therefore it is inevitable that more engineers require programming skills.

## 5. Conclusion

This paper has presented novel developments of computational design tools for structural engineering. These tools aim to support the structural engineer during early design stages and generate design alternatives while little information is available, but decisions have a large impact on the rest of the design process (see Figure 1).

It can be concluded that the research and development of computational tools for conceptual structural design remains quite open for further research. Most challenging is supporting the creativity of an engineer in the conceptual design stage, which requires tools that are more generally applicable.

## Afterword

The authors would like to start a collaborative research group on the topic of "computational tools for early design stages". Please contact us if you are interested in participating in this collaboration.

## References

- [1] Adriaenssens, S., *Numerical Form Finding*, from: <http://formfindinglab.princeton.edu/tools/>, access date: June 2014.
- [2] Allen, E. and Zalewski, W., *Form and Forces: Designing Efficient, Expressive Structures*. John Wiley & Sons, 2009.
- [3] Baker, W. F., Beghini, A., & Mazurek, A., Applications of Structural Optimization in Architectural Design, in *Structures Congress 2012: 20th Analysis and Computation Specialty Conference*, 2012, 257-266.
- [4] Coenders, J.L., *NetworkedDesign – next generation infrastructure for computational design*. PhD thesis, Delft University of Technology, 2011.
- [5] Coenders J.L. and Wagemans L.A.G., *Structural design tools: the next step in modelling for structural design*, Proceedings of the International Association of Bridge and Structural Engineering: Responding to Tomorrows Challenges in Structural Engineering, volume 92, pp. 306-307, 2006.
- [6] Culmann K., *Die graphische Statik*. Meyer & Zeller, 1866.
- [7] Fabrycky W.J. and Blanchard B.S., *Life-Cycle cost and Economic Analysis*, Prentice-Hall, 1991.
- [8] Grashopper, from: <http://www.grashopper3d.com>, access date: June 2014.

- [9] Greenwold, S. and Allen, E., *Active Statics*, from: <http://acg.media.mit.edu/people/simong/statics/Start.html>, access date: June 2014.
- [10] Holzer D., Hough R., Burry M., *Parametric Design and Structural Optimisation for Early Design Optimisation*, in *Journal of Architectural Design Computing*, issue 04, volume 05, pp. 625-643.
- [11] Karamba, from <http://www.karamba3d.com>, access date: June 2014.
- [12] Kensek K.M. and Noble D.E., *Building Information Modelling: BIM in Current and Future Practice*. Wiley, 2014.
- [13] Kilian, A., and Ochsendorf, J., Particle Spring Systems for Structural Form-Finding. *Journal of the International Association for Shell and Spatial Structures*, 2005; **46**(2); 77-84.
- [14] Mirtschin, J., *Geometry Gym*, from: <http://ssi.wikidot.com>, access date: June 2014.
- [15] Mueller, C. and Ochsendorf, J., From analysis to design: A new computational strategy for structural creativity, in *Proceedings of the 2nd International Workshop on Design in Civil and Environmental Engineering*, 2013.
- [16] Mueller, C., *structureFIT*, from: <http://www.caitlinmueller.com/structurefit/>, access date: June 2014.
- [17] Paulson B., *Designing to reduce construction costs*, *Journal of the Construction Division*, volume 102, number 4, pp. 587-592, 1976.
- [18] Rippmann, M., Lachauer, L., and Block, P., Interactive vault design. *International Journal of Space Structures*, 2012; **27**(4); 219-230.
- [19] Rolvink A., Breider J. and Coenders J., *StructuralComponents – a toolbox for conceptual structural design*, *Proceedings of the International Association of Bridge and Structural Engineering: Large structures and infrastructures for environmentally constrained and urbanised areas*, pp. 768-769, 2010.
- [20] Shearer, M., *Analyzing and Creating Forms: Rapid Generation of Graphic Statics Solutions through RhinoScript*. MIT S.M. Thesis, 2009.
- [21] Stromberg, L. L., Beghini, A., Baker, W. F., and Paulino, G., Application of layout and topology optimization using pattern gradation for the conceptual design of buildings. *Structural and Multidisciplinary Optimization*, 2011; **43**; 165-180.
- [22] Van Mele, T., Block, P., Ernst, C., and Ballo, L., *eQUILIBRIUM*, from: <http://block.arch.ethz.ch/equilibrium/>, access date: June 2014.
- [23] Van de Straat R., Coenders J. and Vermeij P., *Computational Design Strategies for the Early Phases of Structural Design*, *Proceedings of the International Association of Shell and Spatial Structures*, 2008.
- [24] von Buelow, P., Suitability of genetic based exploration in the creative design process. *Digital Creativity*, 2008; **19**(1); 51-61.
- [25] von Buelow, P., Genetically Enhanced Parametric Design for Performance Optimization. *7th International Seminar of the IASS Structural Morphology Group*, 2011.