DOI: 10.14621/tna.20160203

Exploration of Optimal Solutions in Architecture

Martin Kaftan

Institute for Architecture and Media, Technical University Graz
Graz, Austria; martin.kaftan@tugraz.at

Abstract

Architects must deal with increasing amount of design constraints, which is the consequence of increasing demands on building's performance in terms of sustainability and construction cost. On the other hand, complex geometries has become common part in architectural projects. Therefore, it is nowadays more true than before that the building's qualities depend on architect's ability to find the optimal solution for all, often contradicting constraints. This is a task for which due to the complexity necessitates the use of sophisticated solving algorithms integrated into the design workflow. The research proposes an integration of optimization apparatus called "Cognitive Control System" (CCS) into a parametric design framework. The CCS contains a set of global and local solvers. Its part is also an interface, the Interactive Graph Control by which the user can steer and control the optimization process in a transparent fashion. This interactive platform presents the user not only the best optimal solution, but also the whole range of other possible solution, even if less optimal.

Keywords:	Multi-objective optimization; Parametric design; Solution space

Article history:	Received: 02 May 2016 Revised: Accepted: 29 July 2016

1. Introduction

Buildings are responsible for approximately 20-40 % of the total energy consumption [1]. However, for the smallest cost have the buildings the highest potential to reduce the emissions [2]. Society tries to respond to these problems, for example by implementing the directive 2010/31/EU, which was adopted in order to strengthen the energy performance requirements on buildings. This directive requires that all new buildings need to be "nearly zero energy buildings" by 2020. As a consequence, the CAD industry is driven by task of convergence of BIM and simulation programs. However, the reviews [3], [4] indicate that the simulation driven optimization methods are nowadays used only for a fine tuning of building design.

This "green awakening" has brought with it a significant amount of constraints for architectural form finding. This gaining complexity makes the design task very difficult without using computational tools. In addition, in order to optimize building's performance effectively, the decisions must be made in the early stages of the design process. Decisions such as positioning, orientation and shape of the building, layout and distribution of windows on façades, consideration of solar potential and shading options have a significant impact on the overall energy performance and cost of a building. For example, optimizing orientation of a building can save up to 40% of the energy consumption without an additional cost [5].

And if the basics are non-optimal, more sophisticated technological solutions need to be deployed in order to meet requirements on the building performance, making such designs more expensive, complex and thus requiring more costly maintenance during operation. Since energy performance criteria are not the only criteria on buildings and other criteria are often contradictory, the design decision support system must support not only single-objective, but also a multi-objective approach [6]. However, if the building's optimization is performed, typically 60% of the cases use only single-objective approach [4]. And if the multi-objective optimization is performed, only solution space is reviewed without taking in account the design range of the input variables [7].

2. Optimisation

Architectural design process is typically non-linear and non-quantifiable task of creating spaces and forms [8]. It is an iterative process of trial and error where must be addressed multiple objectives, such as aesthetics, building performance, structure, cost, comfort, etc. But the more energy-efficient is the building, the more expensive tends to be its construction. One might thus want to find a compromise between energy efficiency and construction costs. Another well-known trade-off in this context is the exploration of natural daylighting versus thermal insulation. The bigger the windows, the more daylight will possibly come in; but windows are usually much less efficient in terms of thermal insulation than walls. Then, some savings in the use of electricity for lighting might be compensated by a higher need of active HVAC systems, which are usually the most energy-demanding equipment in a building.

This multiple design constraints construct a design space of possible solutions that the architect must find. However, the exploration of vast design space is limited without the use of computers and mathematical optimizing algorithms [9]. Therefore, the final building is most often not optimal to all criteria, but only satisfactory equal to the designer's level of ambition.

$$
\begin{aligned}
&\min f\ \mathbf{x,p} \\
&\text{subject to } \mathbf{g(x,p)} \leq 0 \\
&\mathbf{h(x,p)} = 0 \\
&x_{i,LB} \leq x_i \leq x_{i,UB}\ (\mathrm{i} = 1,\ldots,\mathrm{n}) \\
&(\text{g and h are real-valued functions})
\end{aligned}
\tag{1}
$$

Mathematical optimization, also called mathematical programming (MP) is in generally a mathematical method with goal of minimizing or maximizing some quantity (Eq. 1). The solver works with design variables. A design variable is a range of values for the given variable. If $x_1, x_2, \ldots, x_n$ is considered to be the design variables, $f(x)$ is considered to be the objective function, the optimization problem can be defined as *minimize* $f(x)$*; subject to* $x \varepsilon \Omega$. During the optimization process the solver computes number of configurations to find out the most feasible configuration which gives the best objective function.

If the parametric model has multiple variable inputs and only single evaluated output, the function can be described in multidimensional space. For example if the variable inputs are x and y then the function $f(x,y)= 9 - x^2 - y^2$ can be described with a 3D surface (Figure 1). The optimal solution lays on the horizontal tangent plane to the surfaces, where the gradient value is zero.

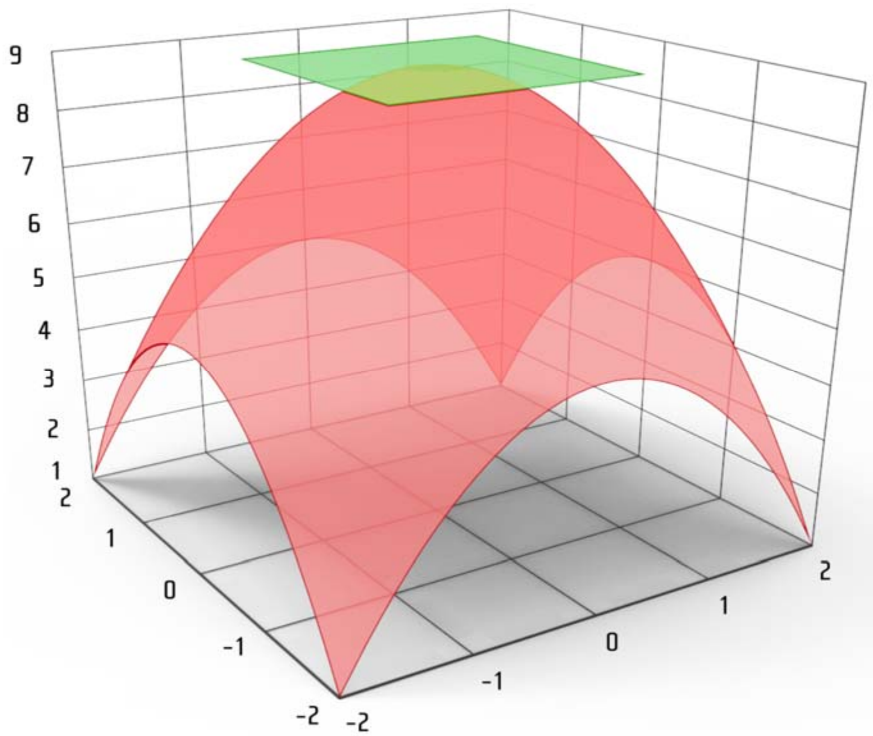

Figure 1. 3D surface from function $f(x,y)= 9 - x^2 - y^2$ with horizontal tangent plane (gradient = 0)

a)

$$J = \begin{bmatrix} J_1 \\ J_2 \\ J_3 \\ J_i \\ \cdot \\ \cdot \\ \cdot \\ J_n \end{bmatrix} = \begin{bmatrix} \text{Energy consumption } [KWh\ /m^2] \\ \text{Daylight factor} \\ \text{Stress displacement } [cm] \\ \text{Cost } [e] \\ \cdot \\ \cdot \\ \cdot \\ \text{Return ofinvestment } [\%] \end{bmatrix}$$

$$J_{\text{weighted sum}} = w_1 J_1 + w_2 J_2 + w_3 J_3 + \ldots + w_n J_n$$

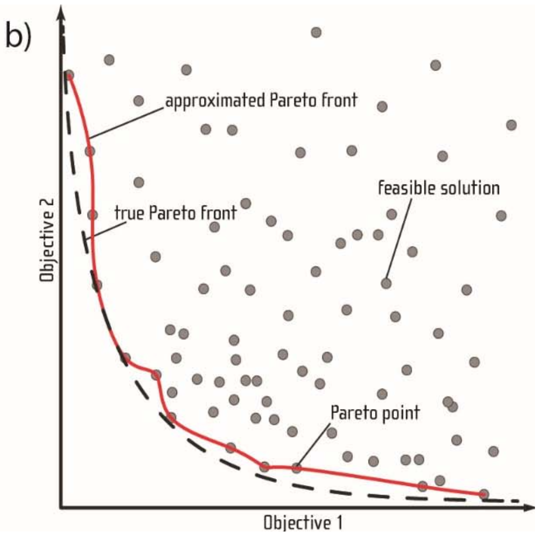

Figure 2. a) weighted sum, b) two-dimensional Pareto front

2.1. Single and multi-objective optimization

When a single-objective optimization is set, the used algorithm will eventually find the optimal solution with respect of its fitness function. On the other hand, in cases of multi-objective optimization (MOO) problems it is not almost always possible to find one optimal design solution that satisfies all design objectives, therefore, referring to this solution as "utopia point" [10].In practice, there are two common approaches to multi-objective optimization problems:

1. Weighted Sum
2. Pareto Front

In weighted sum approach, a composite objective function is defined by combining all of the individual objective functions (Figure 2a). This composite objective function can be determined with various methods, such as the use of weighting factors. It enables the use of only a single-objective algorithm, but determining the composite objective function needs knowledge of the relationships among individual objectives and their weighting factors which is not always possible. In addition, is method has other issues, for example diverse units for different objectives.

Compare to the single-objective optimization where only one optimal solution is found, the MOO results in several equally efficient solutions; this group of solutions is called the Pareto optimal solutions or Pareto front (Figure 2b) [11], [12]. Pareto Optimality supports decision making by finding the equally optimal solutions such that it is not possible to improve a single individual objective without causing at least one other individual objective to become worse.

3. Solvers

Contemporary architects and engineers typically generate computer building models that get modified by the design process until final drawings are produced for construction. Designing a building is a one-time operation compared to other industries, such as software development, where the final output can be thoroughly tested and updated in the future. In building industry any corrections of the design mistakes during or after construction are usually expensive and can cause the project to go over budget. Since it is seldom possible to build a full-scale prototype in order to test the building's performance, the architects most of the time rely on analysis from computer simulations. The use of computation to solve building performance problems in practice typically falls into two categories: Problem specific computation and problem generic computation.

3.1. Problem specific computation

Computation is in the architectural design most commonly used for solving a singular specific problem. As such it can be in the context of optimization process also called a heuristic approach. In computer science, heuristic methods are techniques for solving problems that require some expert knowledge embedded in the algorithm. Their task is to typically optimize a single criteria. In building design these computational methods are most commonly employed in several design tasks.

The first main category includes form finding algorithms. These have an embedded structural logic and are used for developing a geometric form computationally where the geometry faces structural problems that cannot or are difficult to be solved analytically [13], [14]. In the second main category are algorithms that generate building components; these have an embedded construction logic and boundary conditions. They are typically needed in design cases where large amount of the same components is used, but the local conditions for each component fluctuate.

An example of such optimal component placement is the point-fixed structural glazing system for double-glazing façades. The supporting cable structure can be tensioned against each other to form a double curved surface, an example is the Seattle Tacoma International Airport [15]. However, using this construction system for complex geometry is limited by the boundary conditions of the point-fixed connectors - the spiders (Figure 3a). But because these conditions are well defined, the problem can be algorithmically described within a solver, which then places the component in a balanced configuration on each node of the glass panels (Figure 3b). If the connector cannot be placed, it indicates on the model how and where it exceeds its boundary conditions.

The heuristic methods are suitable to solve architectural and engineering problems that can be clearly declared and the method for solving the problem is well known. In practice heuristics dominate the design process, for example, one algorithm is used for surface generation, another one for re-meshing that surface into planar panels, and another for replacing the panels with façade components, etc. Therefore, these methods are suitable for the architect/engineer that wants to solve a well-defined problem by using computation.

3.2. Problem generic computation

Generic algorithms make no assumptions about the problem, and therefore require searching through a very large space of candidate solutions. Because they do not rely on a specific problem, they can be generalized and interchanged, based on the suitability to a particular problem. This independence classifies them as metaheuristic solvers. They are more appropriate than heuristic methods when it is not clear which direction will make improvements, i.e. a poor knowledge of the possible space of solutions [16].

Unlike heuristic methods that embed engineering performance within the process itself, metaheuristics must by their nature conduct some sort of analysis at each iteration to understand the current state of the proposed solution. The quality of the found solution is then influenced by how good it is compared to a measured performance.

Metaheuristics cannot guarantee that an optimal solution has been found without assessing the entire solution space. However, these methods have embedded search logic to find solutions, which task is to narrow the possible number of designs to be evaluated while still ensuring a good solution is achieved.

Some examples of generic solvers are:

1. Brute-Force Search
2. Hill Climbing
3. Evolutionary Algorithms (including Genetic algorithms)
4. Neural Networks
5. Simulated Annealing
6. Particle Swarm Optimization
7. Nonlinear programming algorithms

Considering the task of optimization of building's performance, it is apparent that generic solvers are those suitable (and used) for the task. But building's parametric models are fairly complex and

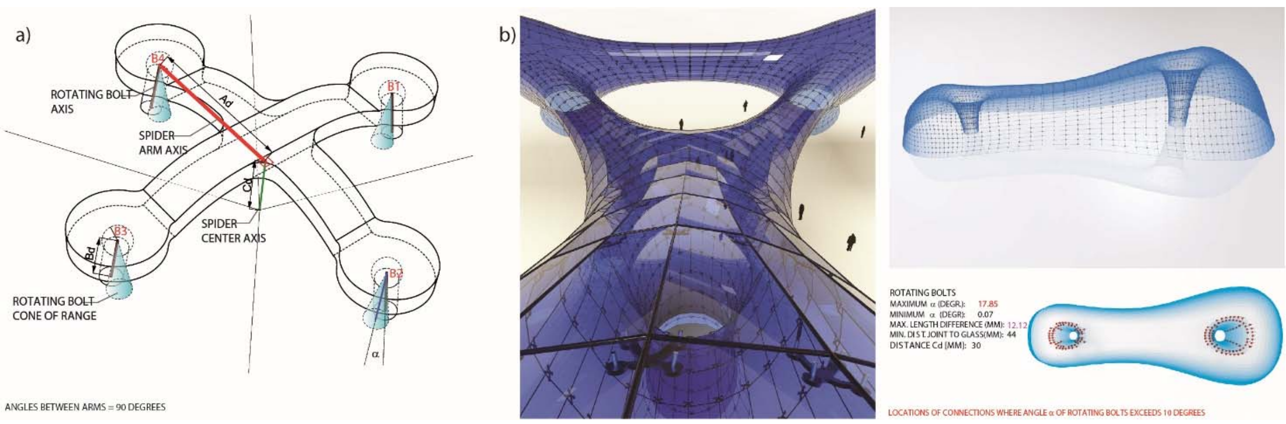

Figure 3. a) point-fixed connector (spider),
b) c) optimal placement of point-fixed connectors on double curved surfaces with embedded error analysis

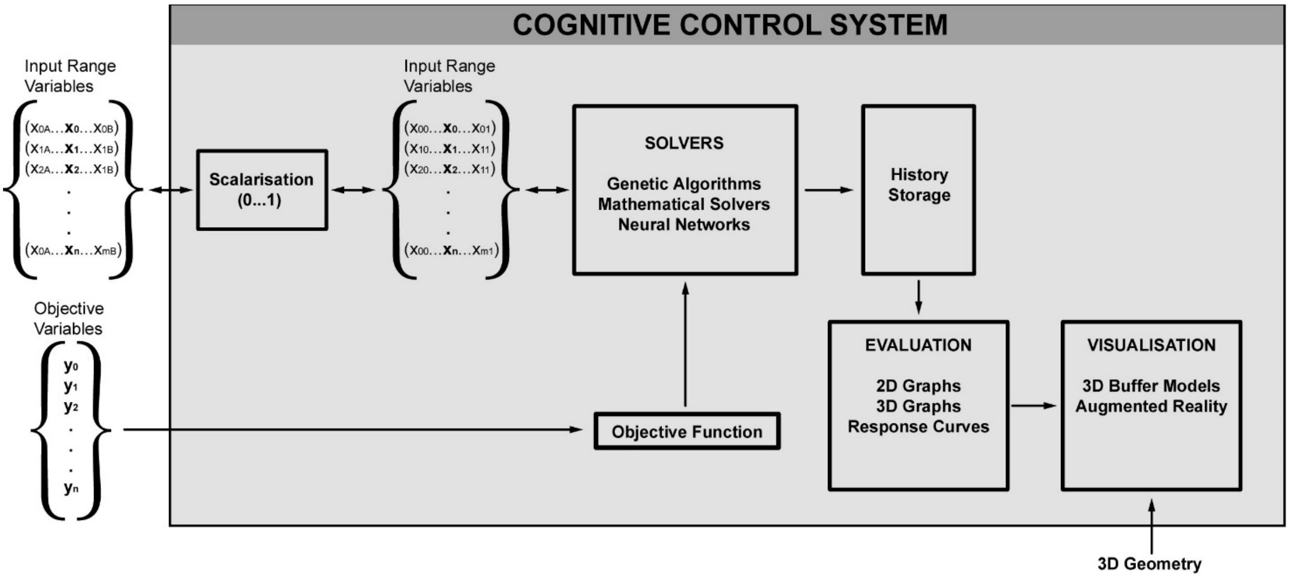

Figure 4. Diagram of Cognitive Control System

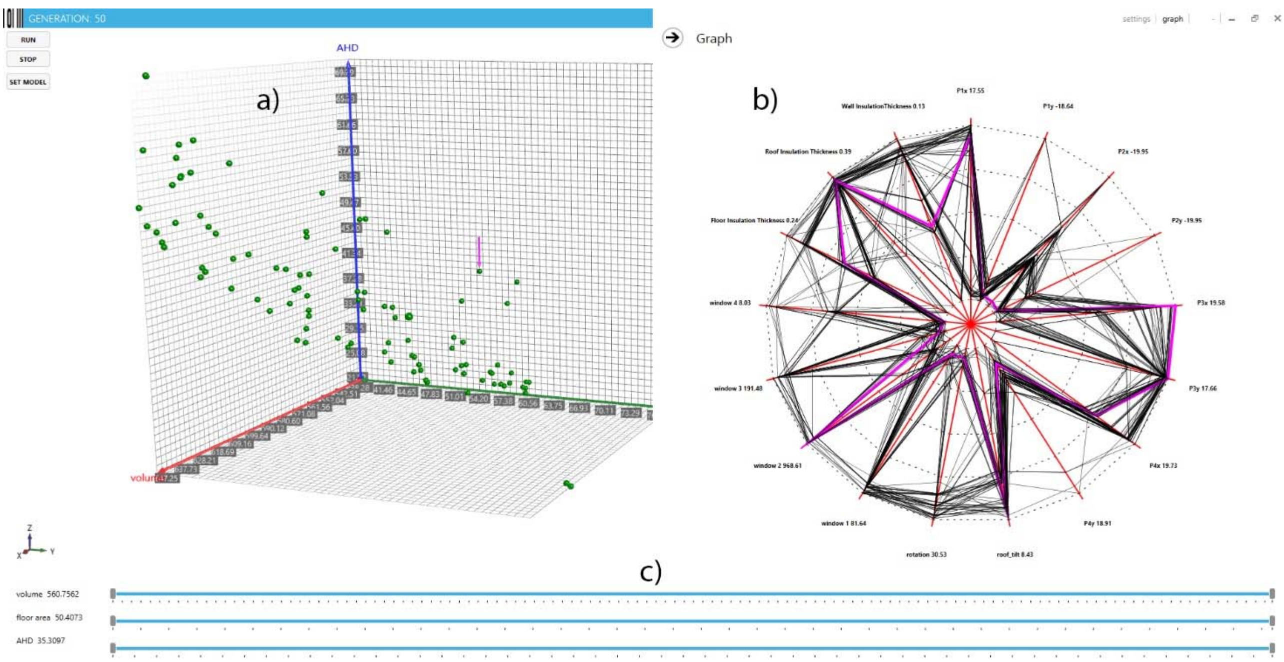

Figure 5. Example figure

computational time needed for analysis can take seconds or minutes (for example for daylight analysis). For an architect this means that to be able to use optimization framework on daily bases, the amount of iterations that the solver must compute needs to be as lower as possible.

4. Control system

In 1981 G. Stiny and L. March in their article "Design Machines" [17] presented the principles of development of an autonomous system for creating designs. In accordance with their theory, the missing part in contemporary CAD software is a cognitive link that bridges the gap between what is possible and what is actual. Cognition or "knowing" can be according to Stiny defined in terms of the ability to respond to environmental events, and the 'stimulus' is the part of the environment that is absorbed by the structure of the model. It is the selector of the best design solutions, describing how and when the language and the context correspond to each other.

In this notion has been developed the "Cognitive Control System" (CCS) as part of the parametric framework for buildings optimization called Augmented Parametrics (AugP) [6]. The CCS is a package including set of optimization algorithms connected to visualisation tools. Its goal is not only to find optimal solutions to given criteria by using an automatic solver, but also to analyse and present relations among design variables and objectives (Figure 4).

4.1. Interaction with design and solution spaces

Most architects lack intensive training in optimization, and optimization tools are usually severely limited in their graphical interface or the data must be exported to other software for graphical evaluation. On the other hand, designers are highly visual and are able to process

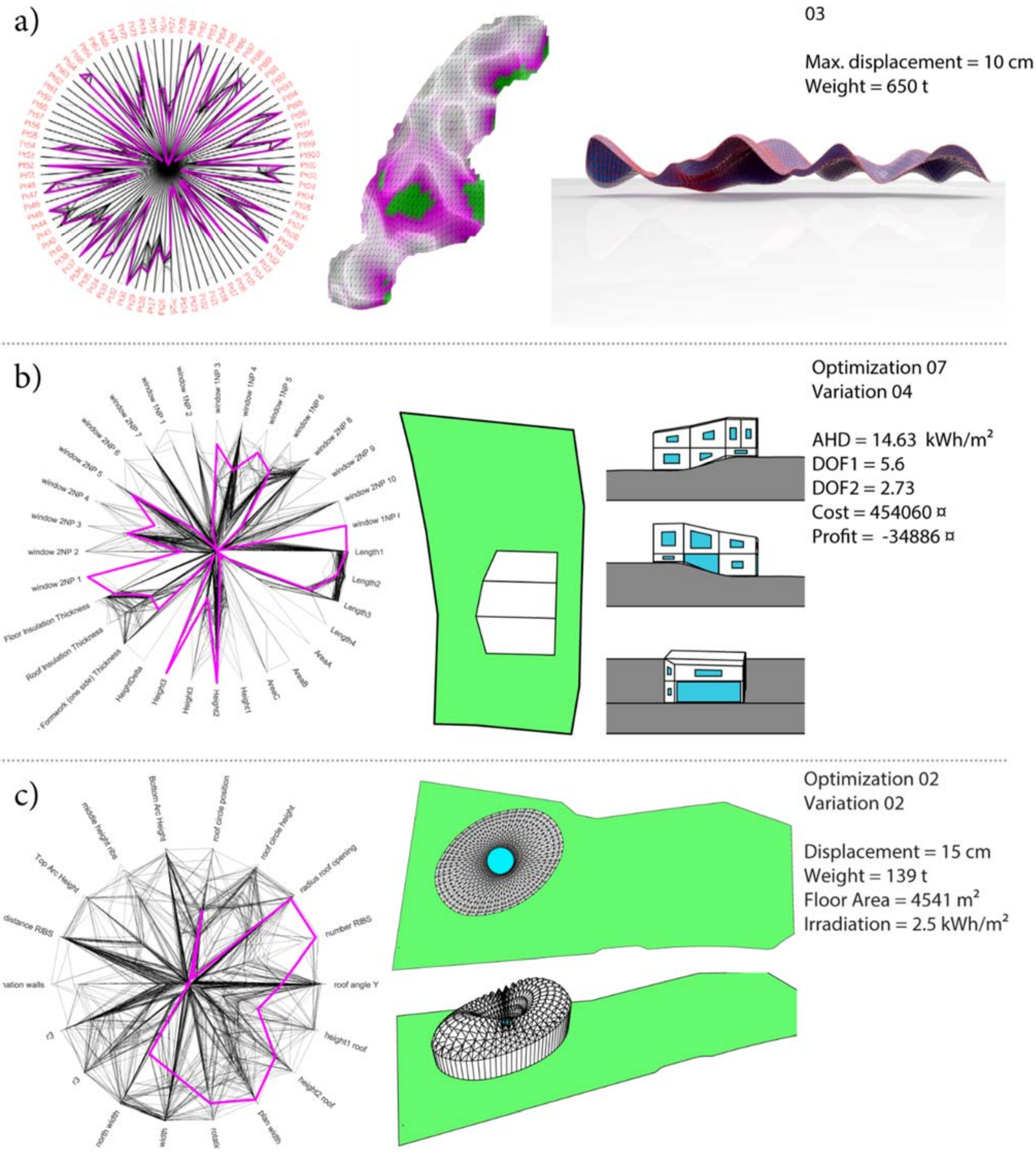

Figure 6. a) Roof structurally optimized form finding, b) Optimization of family house in terms of energy consumption, daylight and cost, c) Optimization of museum of aeronautics for structure, sun exposure and floor area

and evaluate information more quickly graphically. Therefore, beside optimization solvers it is crucial to also integrate a strongly graphical interface (Figure 5).

The design space is presented by an interactive graph, for example the radar chart, where design variables are arranged on equiangular spokes (figure 5b). With the graph can be analysed the solver's search progression as it converges to optimal solutions, which differs according the used solving method. The interactive interface provides visual representations of the design space which helps the architect to understand a problem in an organized, systematic way that can help inform the critical conceptual decisions. Beyond global behaviour, the CCS interface can also convey information about the relative importance and behaviour of individual design variables. For example, which variables contribute significantly to changes in performance and which matter less? Which variables must be set to particular values for reasonable performance?

While the first benefit helps architects prepare for design space exploration and optimization approaches, it can also be used during exploration and optimization to better understand these processes (Figure 6). In both cases, visualisation shows the designers how considered designs connect to each other, and how direct or meandering the path toward a selected solution is. This information can feed back into the exploration and optimization processes to improve their performance.

More broadly, the CCS can be also seen as "Design by Shopping", which is a design method that was introduced by Balling [18]. This idea is motivated by the need of designers to consider many alternative options, prior to formalizing their design goals in the strict, numerical manner required by traditional optimization. In contrast with optimization, the shopping approach aims to present designers with a catalogue of options and affiliated prices (i.e. performance).

References

[1] Pérez-Lombard, L., Ortiz, J. and Pout, C., A review on buildings energy consumption information, *Energy and Buildings*, 40, 2008, pp. 394-398.

[2] UNEP, *Buildings and Climate Change,* United Nations Environment Programme, Paris, 2009.

[3] Attia, S., Hamdy, M., et al, Assessing the Gaps and Needs for Integrating Building Performance Optimization Tools in Net Zero Energy Buildings Design, *Energy and Buildings*, 60, 2013, pp. 110–124.

[4] Evins, R., A review of computational optimisation methods applied to sustainable building design, *Renewable and Sustainable Energy Reviews,* 22, 2013, pp. 230-245.

[5] Kaftan, M., and Pavlicek, J., Form finding of passive houses, *Proceedings of workshop SUBA (Sustainable Construction of Buildings and Sustainable Development of Urban Space)*, Prague, CVUT, 2010, pp. 14-21.

[6] Kaftan, M., Manahl, et al, Augmented Parametrics - A Novel Framework for Numerical Optimizations in a Parametric Design Workflow, *Rethinking Comprehensive Design: Speculative Counterculture,* CAADRIA, 2014, pp. 615-624.

[7] Kaftan, M., *COGNITIVE BUILDING SYSTEMS – Optimization Algorithms in Architecture from Design to Production*, Prague, Ph.D. thesis, CTU, 2015.

[8] Andersen, M., Kleindienst, et al, An intuitive daylight performance analysis and optimization approach, *Building Research adn Information*, 36, 2008, pp. 593-607.

[9] Woodbury, R., Burrow, A. , Whither Design Space, Artificial Intelligence for Engineering Design, *Analysis and Manufacturing*, 20, 2006, pp. 63-82.

[10] Deb, K., *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, New York, 2001.

[11] Ryu, Jong-hyun, Pareto front approximation with adaptive weighted sum method in multiobjective simulation optimization, *Proceedings of the 2009 Winter Simulation Conference*, Austin, 2009, pp. 623-633.

[12] Kim, I.Y., de Weck, O.L., Adaptive weighted sum method for multiobjective optimization: a new method for Pareto front generation, *Structural and Multidisciplinary Optimization*, 31, 2006, pp. 105-116.

[13] Chilton, J. C., *Heinz Isler: The Engineer's Contribution to Contemporary Architecture*, Thomas Telford, 2000.

[14] Otto, F, Bayern, W., et al, *Finding Form: Towards an Architecture of the Minimal*, Axel Menges, 1995.

[15] Post, N.M., *Cable-Supported Glass Wall Made Feasible by Extra Curve*, http://enr.construction.com/nexs/buildings/archives/050411.asp., 2005.

[16] Blum, Ch., Roli, A., Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison, *ACM Computing Surveys*, 35, 2003, pp. 268-308.

[17] Stiny, G., March, L., Design machines, *Environment and Planning B*, 8, 1981, pp. 245-255.

[18] Balling, R., Design by shopping: A new paradigm*?, Proceedings of the Third World Congress of structural and multidisciplinary optimization (SCSMO-3)*, 1983, pp. 295-297.