Automation in Construction

Combining Structural Performance and Designer Preferences in Evolutionary Design Space Exploration

Caitlin T. Mueller[a,*], John A. Ochsendorf[a]

[a] Massachusetts Institute of Technology, Building Technology Program, Department of Architecture, Cambridge, MA 02139, USA
[*] Corresponding author. *Address:* 77 Massachusetts Avenue, Room 5-418, Cambridge, MA 02139, USA. *Tel.* +1 617 324 6236. *Email address:* caitlinm@mit.edu (C. Mueller)

Abstract

This paper addresses the need to consider both quantitative performance goals and qualitative requirements in conceptual design. A new computational approach for design space exploration is proposed that extends existing interactive evolutionary algorithms for increased inclusion of designer preferences, overcoming the weaknesses of traditional optimization that have limited its use in practice. This approach allows designers to set the evolutionary parameters of mutation rate and generation size, in addition to parent selection, in order to steer design space exploration. This paper demonstrates the potential of this approach through a numerical parametric study, a software implementation, and series of case studies.

Key Words: conceptual design, structural design, structural optimization, design space exploration, evolutionary algorithms

1. Introduction

In the conceptual design of buildings and bridges, designers must consider a wide range of goals related to performance. These include quantitative, measurable goals such as structural efficiency, cost, and embodied energy, as well as qualitative goals that cannot be expressed numerically, such as aesthetics, constructability, and contextual appropriateness. The designer's early-stage responsibilities include balancing these requirements to attain a satisfactory initial design concept.

Since the advent of computational tools, digital methods have emerged to help designers assess both quantitative and qualitative performance. For example, structural analysis software provides feedback on structural performance, and computer-aided drawing and modeling software can produce images and renderings that convey aesthetic performance. However, few computational tools exist that allow the designer to consider these two types of performance in one environment. Furthermore, while most widely used tools provide feedback to the designer, very few provide guidance, or suggested changes for design improvement. This leads to designers using software in a time-consuming trial-and-error mode that limits the number of design alternatives that can be considered, potentially reducing the quality of the chosen solution.

A potential remedy for design guidance is optimization, an approach that computes the best performing solution according to mathematically formulated objectives and subject to mathematically formulated constraints. There are a multitude of available optimization algorithms, but nearly all require the objective function to be quantitative. This means that while optimization can address some conceptual design needs, it is not an appropriate approach for meeting unquantifiable yet important goals.

The mismatch between optimization's capabilities and the needs of the practical design process is one likely reason for the limited adoption of optimization in the building industry [1]. A second, related reason deals with the concept of the optimum itself: since many qualitative goals are also subjective, it is impossible to say that one design solution is unequivocally the best in most design problems. Moreover, the combination of two or more competing goals leads to a subjective problem in deciding the relative tradeoff, even when both goals can be objectively evaluated on their own. Furthermore, goals in conceptual design are sometimes fuzzy or unknown, and designers adjust what they are looking for during their design explorations. To support adaptive goal adjustment, an ideal computational approach should expose designers to a diverse range of alternatives that may inspire new goals or spark new ideas. Because optimization produces a single solution to a design problem, it is unable to serve this role in its traditional form.

As an illustration of these issues, Fig. 1 shows a simple geometry optimization problem for a planar truss and the resulting design space for a weight minimization objective. This problem has two local optima, shown in the figure. However, it is also noteworthy that the design space is relatively flat; this means that there are many designs that vary significantly in their geometry while performing similarly to the optimal designs. Fig. 1 also shows three groups of designs that perform within ten, twenty, and thirty percent of the global optimum respectively. These designs exhibit significant aesthetic diversity despite their similar structural performance.

A second example is shown in Fig. 2, which depicts two design alternatives for the lateral and gravity load-resisting system of an airport terminal. The first design is a standard rigid frame, while the second is a shaped frame of similar volume that creates a richer architectural experience by opening the interior space up toward the windows, with more structural depth in the back of the space. Due to the static indeterminacy of a fixed-base rigid frame, many high-performing options for the shaped inner profile are possible, and again, there are opportunities to meet unformulated, qualitative goals by compromising slightly on quantitative performance, and to provide a much broader range of design possibilities.

The research presented in this paper aims to take advantage of these opportunities by providing a way for designers to explore structural design solutions in a directed, performance-conscious manner. In contrast with standard optimization, the approach used here addresses qualitative design goals and constraints by incorporating input from human designers, in addition to considering formulated quantitative objectives. This type of optimization algorithm is classified as interactive (sometimes called human-guided or human-computer) optimization, and is able to overcome the previously stated drawbacks of more conventional approaches.

1.1 Interactive optimization

Interactive optimization comprises a broad class of algorithms and approaches united by the inclusion of human input in the optimization process. An overview of the range of these approaches is given in [2]. Like standard optimization methods, interactive optimization methods can be divided into two groups: those that use gradient information, and those that use heuristics. Gradient-based methods have the advantage of better performance and guaranteed convergence rates, but are usually limited to design spaces that have mathematical properties not usually found in practical problems, such as continuity and differentiability. Heuristic methods are slower and cannot guarantee convergence, but often perform reasonably well over a broad range of problem types encountered in practice.

These two types of approaches can be extended to include interactivity in different ways. Interactive gradient-based methods generally incorporate human input in the form of the initial design considered, since in nonconvex design spaces the local optimum found depends on the starting position. This approach allows the designer to choose a broad neighborhood of a design space, indicating design intent, and the algorithm works autonomously using the continuous, differentiable objective function to find the local optimum of that neighborhood. An example in the structural design domain is given in [3]. This approach is limited by the standard drawbacks of gradient-based methods, and also still restricts the possible solutions to a small number. For example, for the design problem shown in Fig. 1, an interactive gradient-based method could only identify two of the 26 designs shown.

Interactive heuristic methods usually include interactivity by partially or completely using designer input in evaluating candidate designs. This approach is more flexible and is able to expose designers to a wider diversity of results than gradient-based methods, including all 26 of the designs shown in Fig. 1. The most popular interactive heuristic algorithms are interactive evolutionary algorithms, due to their intelligibility, ease of implementation, and demonstrated good performance on a wide range of problems. This paper therefore identifies interactive evolutionary algorithms as a leading method for combining quantitative structural performance with qualitative designer intent in conceptual design, and will propose modifications to standard interactive evolutionary algorithms to more broadly incorporate interaction.

1.2 Interactive evolutionary algorithms

Interactive evolutionary algorithms are an attractive option because they can integrate human input in a straightforward manner, and because they work with populations of design alternatives, inherently providing more than one possible solution. The most common type of interactive evolutionary algorithm is the interactive genetic algorithm (IGA), which is broadly represented in the literature [4].

IGAs and other interactive evolutionary algorithms are interactive in that they employ *subjective selection*, or selection based on human input, in evaluating the fitness of candidate designs. Early implementations of IGAs were developed for applications like graphic art and music, which are almost entirely subjective [5][6]. More recently,

researchers have combined subjective and objective selection in applications that require performance goals of both types. In the field of structural design, the most notable examples were developed by von Buelow, which illustrate this approach on a wide range of structural problems, including truss bridges, folded plate roofs, geodesic domes, and free-form surface structures [7][8].

These examples demonstrate that interactive evolutionary algorithms are a promising way to incorporate unformulated designer preferences into a guidance-based system. However, selection alone is a simplistic and limited way to incorporate user feedback, and only relates to some of the issues with traditional optimization. Existing work lacks the ability to incorporate broader user preferences, such as how quickly to converge upon a solution, the degree of diversity of generated designs, and how to balance formulated and unformulated objectives. Addressing these needs is an important step toward making performance-based exploration approaches relevant to practitioners.

1.3 Organization of paper

Based on the needs described above, this paper proposes an expansion of existing interactive evolutionary approaches for conceptual structural design that incorporates new types of designer input. Section 2 presents these new forms of interaction and details their implementation and expected impact on design space exploration. Section 3 introduces new numerical metrics for evaluating the quality of evolutionary design space exploration based on the goals described in Section 1.2. Section 4 presents results from a parametric study that shows how designers can achieve different values for these new metrics based on the interaction proposed in this paper. Section 5 introduces a software tool for designers that implements the proposed evolutionary exploration approach, and includes an overview of the user experience. Section 6 presents results from a series of case studies that show how the software tool can be used in conceptual structural design in several ways. Finally, Section 7 summarizes the contributions of the paper and suggests areas for further research.

2. Interaction through evolutionary parameters

Evolutionary algorithms offer a convenient, built-in way to further incorporate designer input through the evolutionary parameters of *generation size*, the number of designs in a population, and *mutation rate*, the degree of randomness introduced when a new population of designs is generated. In traditional evolutionary algorithms, the setting of these parameters is seen as a problem, and many researchers have studied the optimal settings of these parameters for maximal optimization performance [9]. However, in design space exploration, pure quantitative optimization is not usually the ultimate goal, so a wider range of settings for these parameters can be used to affect how the designer moves through the design space. This section details the proposed approach of designer-specified evolutionary parameter settings.

2.1 Basic interactive evolutionary algorithm

The standard evolutionary algorithm used in this paper is an iterative approach that can be implemented in five repeated steps, as shown in Fig 3: generation of designs, quantitative evaluation, satisfaction check, user selection, and breeding of designs. In each new generation, the entire population is replaced by designs generated from selected parents. Through the breeding of high-performing solutions over multiple generations, high quality and often optimal solutions can be found [10]. An interactive evolutionary algorithm modifies this approach by incorporating designer interaction. Existing methods include designer input at the selection step, in which the designer helps to decide which designs should become parents for the next generation, thereby incorporating qualitative goals. Fig. 4 illustrates the preferred version of this type of interaction for structural design problems, in which user input is combined with computed performance evaluations. This paper proposes incorporating additional designer input a second time, at the breeding step, as shown in Fig. 4. The two steps that incorporate designer input are shown in detail in Fig. 5.

Like all optimization algorithms, evolutionary algorithms operate on a numerical representation of a design called the design vector. Certain evolutionary algorithms, such as genetic algorithms, encode the design vector (sometimes referred to as the genome) using binary values. However, for generality and convenience, this paper will use real-valued design vectors, with values normalized to [0, 1], like those seen commonly in other optimization approaches [11]. Based on the format of the design vector, two key functions in the breeding step of evolutionary algorithms must be defined: crossover and mutation.

Conceptually, crossover creates offspring by blending source data from one or more parents. In biology, the evolutionary benefit of this process is the possibility to inherit good traits from both parents, and to potentially develop new advantageous traits from new genetic combinations. In evolutionary algorithms, crossover is used for similar purposes, although its implementation can be flexible and need not exactly mimic the biological analog. The crossover approach used in this paper is to generate each offspring design with a random weighted average of parent design vectors, as shown in Eq (1),

$$x_{i,crossed} = \frac{\sum_{j=1}^{p} w_j x_{i,j}}{p} \quad (1)$$

where $x_{i,crossed}$ = ith element of design vector $\boldsymbol{x_{crossed}}$, p = number of parents, w_j = uniformly distributed random variable within [0, 1], and $x_{i,j}$ = ith element of the jth parent's design vector. This approach is easy to implement, allows for any number of parents, and takes advantage of the continuity of a real-valued design space.

With this method, the resulting designs always fall within the bounds of parent variable settings. A second step, mutation (or random perturbation) is needed to generate variation that moves beyond parent designs. In biological evolution, mutation introduces beneficial or detrimental changes into DNA that increase biodiversity in a population of organisms, potentially leading to new advantageous traits or resilience in a changing fitness climate. Again, the algorithmic version of this operation has similar effects, but does not need to literally implement the biological operation. In this paper, mutation is carried out by adding a normally distributed random variable centered at zero to each element of the design vector, as shown in Eq (2)-(3),

$$x_{i,mutated} = \min(\max(x_{i,crossed} + \rho_i, 0), 1) \quad (2)$$

$$\rho_i \in \text{norm}(\mu = 0,\ \sigma^2 = 0.5r) \quad (3)$$

where $x_{i,mutated}$ = ith element of design vector $\boldsymbol{x_{mutated}}$. The standard deviation of the normal distribution is a function of the mutation rate, $r \in [0, 1]$. The mutation rate can be fixed, or as is proposed here, set by the user, which has the advantage of allowing the user to decide how much offspring should vary from parent designs. This is discussed further in Section 2.3.

2.2 Generation size

The generation size, n, sometimes called population size, determines the number of candidate designs evaluated in a single generation of evolutionary exploration. In traditional optimization applications, setting the generation size is a tradeoff between ensuring high performance and limiting computation time. A larger generation will likely include better performers to choose from among its population, increasing the likelihood that an optimum or near-optimal solution will be found quickly. However, because each member of the population needs to be evaluated by the fitness function, larger generations can also be very slow to analyze. This limitation is significant because most evolutionary algorithms require analysis of tens or hundreds of generations.

However, in interactive evolutionary algorithms, this behavior can be used to help control design space exploration. It is proposed to allow the user to vary this parameter significantly, which affects the degree to which the designer is pushed toward optimal design space regions. Additionally, the designer can adjust this parameter separately for each generation, allowing adaptable convergence toward structurally optimal designs.

Independent of the generation size, the designer is shown a fixed number – by default ten in this paper's implementation – of top-performing designs. Therefore, large generations are more likely to yield better results, since the algorithm displays a smaller percentage of the best quantitative performers. This is ideal behavior when the user is looking for optimal designs or optimal regions of the design space. However, the user may want to explore a suboptimal range of designs that are otherwise interesting for qualitative reasons. In this case, a small generation size will expose a higher percentage of the population to the user, potentially revealing qualitatively desirable options. A small generation is also less likely to contain many high performing designs that are far away from the selected parents, helping the user maintain the general area of exploration.

In general, design problems of higher dimension require larger generation sizes to adequately explore variation within the design space. For example, enumeration of all designs for an n-dimensional problem with m settings for each variable would result in m^n design options. Empirical research in genetic algorithms suggests that for optimal performance, generation sizes be set between n and $2n$ for binary variables [12]. The proposed approach allows the

user to vary the generation size from ten designs up to min(150, $8n_{equiv}$), where n_{equiv} is the length of the binary design vector equivalent to the n-dimensional real-valued version. This allows the maximum mutation rate to scale with the problem size up to a reasonable upper bound. Because the bounds for generation size are computed automatically by the framework, the user does not require expertise in evolutionary algorithms to get reasonable results.

2.3 Mutation rate

The mutation rate, r, determines how much randomness is introduced into a generation in evolutionary exploration. Like the generation size, the mutation rate is generally fixed in traditional optimization applications and existing interactive evolutionary algorithms. The fixed value is generally low. Alternately, the mutation rate is sometimes set high initially, and decreases as generations progress and an optimal design is converged upon. The advantage of a high mutation rate is the ability to move around the design space quickly, out of local minima or poorly-performing regions and potentially toward new and better-performing options. However, a mutation rate that is too high disrupts the evolutionary process, effectively eliminating the positive traits passed on by parents through large random perturbations. At the extreme, very high mutation rates reduce evolutionary exploration to random search.

As with generation size, the behavior of the mutation rate can be exploited to guide design space exploration. In the proposed approach, the designer may decide how much variation to introduce in each generation, based on qualitative preferences. A low mutation rate can be used to concentrate designs in high-performing regions, or near selected designs that do not necessarily perform well. This is preferable when the user has found a part of the design space of interest, and wishes to fine-tune the design by exploring small variations. Large mutation rates increase the likelihood of offspring to jump to regions of the design space far from their parents. This behavior is useful when the user is looking for a breadth of ideas. Because the mutation rate always varies between 0 and 1 regardless of problem specifics, it is simple for a non-expert user to manipulate this parameter with consistent results.

The proposed approach also uses an original technique, called a diversity booster, to ensure diversity in the top ten designs relative to the mutation rate. This ensures that the designs considered visually by the user are different enough to offer choice. The diversity booster filters out designs that are too similar to those already under consideration in the group of top performing designs. The similarity metric is a function of the mutation rate, so that generations with a high mutation rate meet a higher standard of design diversity than those with lower rates. This increases the number of significantly different designs that the user can consider, thereby also enhancing the design space exploration.

2.4 Interaction between mutation rate and generation size

When combined, the behavior of mutation rate and generation size can interact to lead to multiple different exploration opportunities beyond what is possible by varying only one. The interaction behavior is summarized qualitatively in Table 1.

	Low Mutation Rate	*High Mutation Rate*
Low Generation Size	Slight improvements in close neighborhood of parent designs	Large variation with reduction in performance
High Generation Size	High-performing designs in wide neighborhood of parent designs	Medium or low-performing designs outside of parent design neighborhood

Table 1. Properties of evolutionary explorations based on mutation rate and generation size.

Based on the settings of these evolutionary parameters, the designer can therefore steer the exploration of the design space. The proposed approach therefore offers significant additional control for the user beyond existing approaches, which use selection alone as the method of interaction. To illustrate the benefits of this new approach in a numerical way, the next section introduces new quantitative metrics for evaluating the algorithm's performance.

3. Metrics for measuring exploration performance

Traditional optimization algorithms are evaluated by how quickly they converge on a solution, and on the quality of the solution. However, interactive evolutionary exploration has broader goals than optimization, as detailed in Section 1.2, including the diversity of the displayed design options and the closeness of the generated designs to parent designs.

This section presents three new metrics to evaluate these qualities of evolutionary design space exploration at a generation level. Each of the metrics is a scalar aggregate value based on the top ten performers of a particular generation. The metrics are summarized in Fig. 6.

3.1 Cost

The first metric is a measure of quantitative performance for top designs in a particular generation, denoted as the *cost* of the generation. The metric is general and can represent a variety of minimization objectives in structural design, such as required material weight, strain energy, or total construction expenses. The metric is computed as the mean score (normalized by the score of the initial design) of the top ten designs in each generation, as shown in Eq (4),

$$C = \frac{\sum_{i=1}^{10} p_i}{10} \tag{4}$$

where p_i is the normalized score of an individual design. This metric relates to the designer's need to find high-performing designs, evaluated at the generation level. In design space exploration, the designer may require the cost to be as low as possible, but more often prefers to trade off this metric with qualitative design criteria.

3.2 Distance from selected parent designs

The second metric measures how far the top designs are in the design space from the selected parent designs. In general, distance in the design space can be measured using Euclidian distance, or mathematically, the ℓ_2 norm, of the difference between two n-dimensional design vectors. To capture distance from parents at a generation level, this metric computes the distance between the centroid of the selected parent designs and the centroid of the offspring designs, as shown in Eq (5).

$$S = \left\| \bar{\boldsymbol{x}}_{\boldsymbol{selected}} - \bar{\boldsymbol{x}}_{\boldsymbol{offspring}} \right\| \tag{5}$$

This metric quantifies how much offspring designs resemble their parents, or in other words, how much the designer's qualitative selection criteria were incorporated. Offspring designs that are very close to their parents reflect a strong consideration of qualitative goals, while those which are further from the parents indicate a generation that favored quantitative performance over qualitative preferences. This metric can therefore be considered as a numerical representation of designer intent. In design space exploration, the desired value for this distance varies depending on how flexible the user is on trading off qualitative goals with quantitative performance.

3.3 Diversity

The third metric measures the diversity within a particular set of top designs, computed as the maximum Euclidian distance between the centroid of the group and each of the individual designs, as shown in Eq (6).

$$D = \max \left\| \bar{\boldsymbol{x}}_{\boldsymbol{offpsring}} - \boldsymbol{x_i} \right\| \tag{6}$$

The purpose of this metric is to evaluate how different a group of top designs are from each other. A more diverse group gives designers more options to choose between, while a less diverse group represents consensus or convergence toward a particular solution. In design space exploration, a high diversity value is preferable when the designer is interested in considering a wide range of alternatives, while a lower value is desirable once the designer has honed in on a particular design or family of designs.

4. Parametric study

This section aims to more precisely quantify the interaction of mutation rate and generation size and their impact on exploration performance, as an expansion on the summary presented in Table 1. A parametric study investigates the effect of mutation rate and generation size on the new exploration metrics introduced in Section 3: cost (*C*), distance from selected designs (*S*), and diversity (*D*).

4.1 Data generation

This study considers the design problem shown in Fig. 7, a bilaterally symmetric planar truss with a gable geometry and four design variables. The truss is assumed to be made out of steel with round pipe members with a wall thickness of 5% of the outer radius. The objective function used in this study minimized volume, given allowable stress and Euler buckling requirements:

$$\text{minimize } \sum_{i=1}^{n} A_i L_i \tag{7}$$

$$A_i = \max(A_{i,stress}, A_{i,buckling}) \tag{8}$$

The stress- and buckling-based required areas were calculated using member forces computed using a direct stiffness method code written by the authors.

It should be noted that this objective does not represent full construction costs, since it does not include other expenses such as connections and labor, and it assumes a continuous range of possible section sizes instead of a more accurate discrete set. It also does not account for global buckling effects, such as lateral torsional buckling. Nevertheless, it is a reasonable approximation for structural performance in conceptual design, provided that these additional issues are considered and checked as the design is refined.

Using this design problem, a random initial population was created, and two parents from the top ten members were selected by a user. Using these parents, 2500 second generations were created based on a 50 by 50 combinatorial array of mutation rate and generation size values. Mutation rates varied between 0 and 1, and generation sizes varied between 10 and 150. For each generation, the cost (*C*), distance from selected designs (*S*), and diversity (*D*) metrics were computed. The results from this study are given in the next section.

Because these generations were all created from the same parents, the effects of problem specifics are controlled, distinguishing the impact of mutation rate and generation size from other influences. Further similar studies could be conducted for additional problems and parent designs to confirm the generality of the results presented here.

4.2 Generation metric results

The selected parents used for the study and a sampling of four of the 2500 subsequent generations are given in Fig. 8. The number beneath each design is its required volume, normalized by that of the initial design. Figs. 9-11 show the impact of mutation rate and generation size on the three aggregate generation metrics introduced in Section 3. In each figure, the surface plot shows the average behavior, while the stem plots show individual data points representing a single generation. Additionally, the four generations from Fig. 8 are called out for reference.

In general, these results show that there is a definitive relationship between mutation rate, generation size, and each of the three metrics. This means that the designer can control the characteristics of a generation of designs by varying the evolutionary parameters in particular ways. The designer is therefore able to balance quantitative performance, qualitative intent, and design diversity according to preferences in ways not previously possible in other interactive evolutionary approaches.

Specifically, the lowest costs are attained with large generation sizes and small mutation rates. The surface is rather flat in this region, meaning that costs are still relatively low for medium generation size and mutation rate values. However, costs increase significantly when generation sizes are low and generation rates are high. This is because small generation sizes do not offer the buffer of a high percentage of designs beyond the top ten, and high mutation rates can introduce large variations away from high-performing design space regions.

The closest generations to the selected parents are attained with small generation sizes and small mutation rates. Distance from parents quickly increases as both of these metrics grow, and then remains relatively flat for high values. High mutation rates increase distance from parents due to the variation they introduce, and large generation sizes increase the likelihood that the top ten designs will come from different, higher-performing regions of the design space.

Finally, the most diverse generations are attained with high mutation rates, independent of generation size. Again, this is due to the variation introduced by mutation. It should be noted that because the approach used here employs a diversity boosting algorithm, which ensures that the top ten designs are reasonably different from each other, larger generation sizes do not limit diversity through their buffer.

4.3 Tradeoff between quantitative and qualitative goals

The consideration of both quantitative and qualitative goals can be thought of as a multiobjective optimization problem, with tradeoffs inherent in balancing the two criteria. This understanding of the design approach can be studied quantitatively by using the distance from selected parents metric, *S*, as a proxy for designer intent. Designs that are closer to the parents meet qualitative goals better, and vice-versa.

Fig. 12 illustrates this concept by showing Pareto frontiers for the cost and distance from selected design metrics. The data is grouped by diversity level, and it is evident that for any desired diversity of a generation, the Pareto tradeoff between cost and designer intent exists. This result illustrates the new potential of the proposed approach: now the designer can interact with the evolutionary algorithm not only through selection, but also by deciding on the relative weighting of the quantitative and qualitative goals.

5. Implementation as digital design tool

To further investigate this new interactive evolutionary conceptual design approach, the algorithm has been implemented in a proof-of-concept design tool called *structureFIT* [13]. This tool allows users to explore geometric variations of planar trusses with a volume minimization quantitative goal in a graphical, web-based environment. The software is written in C# and uses Silverlight[14] for the graphical user interface. This section presents an overview of the user experience of structureFIT and explains how the tool could be expanded to work with a wide variety of structural types and quantitative design goals.

5.1 Problem types and generality

The organization of the software architecture for structureFIT is shown in Fig. 13. Although the tool works with trusses with variable nodal positions and a volume objective by default, the implementation was designed to be general in terms of structure type, variable type, and analysis method. This means that many different conceptual structural design problems, such as three-dimensional grid shells or frame systems, could be considered. An example of this extensibility is given in the case study in Section 6.

5.2 Design problem setup

The first mode the user encounters in structureFIT is the *Setup* mode, in which one can choose from a list of preset design problems, modify a preset problem, or define a new one. A screenshot of this mode is shown in Fig. 14. The Setup mode includes a graphical and table-based user interface for modifying initial geometry, loads, boundary conditions, material properties, section types, and design variables. It is also possible to specify symmetry conditions and other parametric relationships. Once the designer is satisfied with the problem setup, a button can be clicked to check the structure for stability, loads, and variables, and then move to the next mode, *Exploration*.

5.3 Interactive evolutionary exploration

The second and main mode of structureFIT is the Exploration mode, shown in Fig. 15. In this mode, the designer generates and reviews populations of designs in a graphical user interface. For each generation, the mutation rate and population size can be adjusted using sliders, and parents can be selected by clicking on designs.

The top ten designs of each generation are displayed in rows that also include the generation's number, mutation rate, and generation size. As in Fig. 8, a normalized score beneath each design communicates its performance to the user. The user may therefore choose to select designs that perform well, or ignore the scores and select based on visual characteristics alone. Designs selected by the user to be parents are denoted visually with a gray boxed outline. After selection, the user clicks a button to create a new generation based on the selected parents. If the user is dissatisfied with the exploration or wishes to change parents or parameter values, it is possible to return to a previous generation and restart the exploration process. Once a satisfactory design has been found, the designer may choose to make small, final adjustments to it in the *Refinement* mode.

5.4 Design refinement

The final mode allows the user to modify the design found in the Exploration mode, and view the impact on performance in real-time. A screenshot of this mode is shown in Fig. 16. The user modifies the design by clicking and dragging on the nodes to change their positions. The performance updates are communicated both graphically, through the rendered thickness of the members, and quantitatively, through a numerical score.

While real-time analysis alone is not an effective approach in conceptual design for problems with more than a few variables, it can be extremely effective in enabling small post-processing changes after optimization or exploration. The refinement mode also gives the user more agency and the opportunity to interact one last time, by editing designs found by the computer to meet aesthetic, constructability, or other qualitative requirements.

Once the user is satisfied with the final design, its geometry can be exported from structureFIT as a .dxf file to be imported into later-stage modeling and analysis tools. It is also possible to save structureFIT models as text files (called *.des* files) to be opened and edited at a later time.

6. Case studies

To illustrate the capabilities and flexibility of structureFIT, this section introduces a series of case studies that use the tool in several ways for evolutionary exploration of a given design problem, illustrated in Fig. 17. This problem, a shaped rigid frame with a variable inner profile, is more complex and realistic than the previous problems considered in this paper for several reasons.

First, the structure type is no longer a truss, but a solid frame. Because bending action can be approximated with equivalent truss behavior [15], the frame is still modeled as a truss for analysis, but uses a new objective function. The frame is to be cast in reinforced concrete, so the minimal-volume solution will have a minimum enclosed area in the planar model. The objective function therefore minimizes the area of the enclosed polygon, plus a penalty term that is very large (10^8) when a stress constraint is violated in the outer "profile" members in the truss model, and zero otherwise. Constructability considerations are not included in the quantitative objective function in this case. Because of the conceptual nature of this example, dimensional requirements such as minimum rebar cover are also neglected, but could be incorporated in later design stages. Additionally, global buckling effects such as lateral torsional buckling are not considered, and should be checked after this initial design exploration.

Second, this problem is conceived as the gravity and lateral system of a long-span airport terminal, and therefore resists lateral loads from wind and earthquakes in both directions. Therefore, two load cases are considered, as shown in Fig. 17. The objective function checks for stress constraint violation for a given geometry in either load case.

Finally, this problem uses parametric relationships beyond symmetry. The nodes between the variable coordinates are constrained along a line between them, resulting in relatively smooth variations. This also allows the problem to reduce its dimensionality: although there are 20 degrees of freedom along the frame's inner profile, there are only 10 design variables.

Using structureFIT, three modes of exploration are exemplified: an optimization-like mode that attempts to find the best quantitative performer, a free exploration mode that favors qualitative characteristics of designs alone, and a hybrid that considers both types of goals.

6.1 Optimization-like exploration

Four of eight generations from an optimization-like exploration approach are shown in Fig. 18. In order to optimize, the highest scoring designs were selected in each generation, and low mutation rates and high generation sizes were used.

The design selected after exploration has a score of 0.68, meaning that it uses 32% less material than the initial orthogonal design in 8 generations, an exploration that can be completed in a few minutes by the designer. Because both directions of lateral load were considered, the design is nearly symmetrical, although symmetry is not required. Slight modifications to the design made in the Refinement mode are shown in Fig. 19: the tapering of the columns is exaggerated. This case study illustrates that it is possible to use an interactive evolutionary approach for pure optimization problems (although convergence to an optimal design is not theoretically guaranteed). The interaction does not hinder the algorithm's performance, and can improve it if the user selects parent designs astutely.

6.2 Free exploration

Generations from the second case study, which uses a free exploration approach, are shown in Fig. 20. This approach initially used high mutation rates to generate a diversity of results, and then low mutation rates and relatively low generation sizes to concentrate results near designs selected for aesthetic reasons.

The selected design performs only slightly better than the initial design. However, it varies significantly in appearance. It also displays a considerable savings over the design selected in Generation 0, which captures the designer's intent, while remaining visually close to it. The final design developed in the Refinement mode is shown in Fig. 21, and reflects adjustments that smooth out the design's curves and increase the score to 1.00, the same score as the initial design.

This case study shows how the proposed interactive evolutionary algorithm can allow for free design exploration with only slight pressure from the evolutionary objective function. In this case, the user interacts with the algorithm to prioritize aesthetic goals. The algorithm is able to produce designs that fit with the designer's creative intentions while still providing performance as good as the standard initial design.

6.3 Hybrid exploration

The final case study combines the optimization-like approach with some degree of free exploration, with generations shown in Fig. 22. The approach uses medium mutation rates and high generation sizes to find new designs that resemble their parents, but also perform considerably better. Furthermore, parent designs are selected based both on qualitative characteristics and numerical scores. This exploration likely represents most closely the design approach that would be useful to designers in practice.

Although initial generations do not perform particularly well, after eight generations, a design is selected with a score of 0.75, meaning it uses 25% less material than the initial design. The user then modifies the design slightly in the Refinement mode, resulting in a final design with a score of 0.77, as shown in Fig. 23.

This last case study shows how this new approach can be used to quickly and intuitively discover design alternatives that save material without significantly sacrificing architectural goals. Through selection and setting the mutation rate and generation size, the designer can focus the design space search as desired, unlike in traditional optimization, while still improving performance in a way not possible with geometry-based tools alone.

6.4 Case study summary

Fig. 24 presents the generation-based cost, *C*, over eight generations for each of the three case studies. Each mode shows cost improvement as the generations progress, with steepest improvement in the early generations. As expected, the optimization-like exploration has the best cost over the course of the evolution, and the free exploration mode performs worst. However, the downward trend of all three modes proves that the interactive evolutionary approach enables better performance, regardless of whether the user has reduced cost as a primary goal. This also means that a wide variety of mutation rates and generation sizes can be used, as proposed here, without substantially disrupting evolutionary improvement.

The three design options for the airport terminal structural system found in the case studies are rendered in Fig. 25, confirming the point made in Fig. 2: there are many high-performing solutions for most conceptual structural design problems. This point is underscored by the design space plot in Fig. 26, which shows the projected design space of the airport terminal problem across the ten variables introduced in Fig. 17. Three isoperforming designs are illustrated at six performance levels, ranging from 1.00 down to 0.60. As with the truss designs in Fig. 1, these designs illustrate the considerable variation possible with small compromises away from the single optimal design. The plot also shows that some variables, such as x_2 and x_9, have limited values in high performing regions, while others can vary more significantly.

7. Conclusion

7.1 Summary of contributions

This paper has proposed a new way to incorporate user feedback in interactive evolutionary algorithms for conceptual structural design, by allowing the designer to set the evolutionary parameters and generation size. This new approach improves upon existing methods by giving the designer control over the diversity of designs considered, the rate of convergence, and the multiobjective tradeoff between formulated quantitative goals, like structural volume, and unformulated qualitative goals, such as architectural value.

The paper has also introduced new numerical metrics for evaluating characteristics of generations in evolutionary exploration, as a way to numerically illustrate the potential of the proposed interactive approach. Based on these metrics, this paper has presented results from a large parametric study that demonstrated the relationship between mutation rate, generation size, and metrics related to cost, designer intent, and design diversity. The parametric study showed that varying the mutation rate and generation size gives the user unprecedented control over the nature of the design space exploration, allowing one to prioritize performance, qualitative preferences, diversity, or a combination of these goals.

Finally, this paper demonstrated the use of the proposed design space exploration method through a software tool implementation and a series of design problem case studies. The design tool and case studies showed that multiple modes of exploration are possible with the new approach, including a prioritization of performance, but also compromises that consider qualitative goals. Each case study showed cost reduction through the evolutionary process, while illustrating the diversity of possible design options.

7.2 Future work

The work presented here focuses on combining a single structural performance objective with qualitative architectural design goals. In many design problems, there are additional quantitative goals that should also be considered in the conceptual design of buildings, such as energy and daylighting performance, constructability, and embodied energy. Future work in this area should aim to expand the proposed approach to consider multiple quantitative objectives, resulting in a multiobjective exploration method that can handle multiple objectives, both quantitative and qualitative in nature. Additional incorporation of user preferences is possible in deciding the tradeoff between quantitative objectives, and this should be integrated into the expanded approach.

Further future improvements include developing case studies with more complex structure types and implementations that can connect more directly to complex architectural geometry. This could be achieved by extending the existing tool, structureFIT, to work with generative architectural modeling software like Rhinoceros [16] and Revit [17].

7.3 Concluding remarks

Despite the strong need for highly efficient structures in buildings and bridges, motivated by rapidly decreasing resources and high construction costs, structural optimization methods that prioritize efficiency have not taken hold in practice. The limitations of optimization, including its lack of flexibility and its single result, have prevented its widespread use in the building industry, where qualitative preferences and architectural goals are of great importance. This paper supports efforts to broaden optimization so that it can be used by designers in practice, and has presented a new approach to give designers unprecedented control and interaction in performance-focused design space exploration. There are an infinite number of solutions to a structural design problem, and this approach allows designers to focus on a broad subset of them that perform well and offer architectural variation and expression.

References

[1] M. Z. Cohn, “Theory and practice of structural optimization,” *Structural Optimization*, vol. 7, pp. 20–31, 1994.

[2] S. D. Scott, N. Lesh, and G. W. Klau, “Investigating Human-Computer Optimization,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2002, pp. 155–162.

[3] R. Clune, J. J. Connor, J. A. Ochsendorf, and D. Kelliher, “An object-oriented architecture for extensible structural design software,” *Computers & Structures*, vol. 100–101, pp. 1–17, Jun. 2012.

[4] H. Takagi, “Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation,” *Proceedings of the IEEE*, vol. 89, no. 9, pp. 1275–1296, 2001.

[5] K. Sims, “Interactive Evolution of Dynamical Systems,” in *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, 1992, pp. 171–178.

[6] M. Herdy, “Evolution Strategies with Subjective Selection,” in *Parallel Problem Solving from Nature—PPSN IV*, 1996, pp. 22–31.

[7] P. von Buelow, “Suitability of genetic based exploration in the creative design process,” *Digital Creativity*, vol. 19, no. 1, pp. 51–61, Mar. 2008.

[8] P. von Buelow, “Parametric exploration of discrete structures using evolutionary computation,” in *Symposium of the International Association for Shell and Spatial Structures (50th. 2009. Valencia). Evolution and Trends in Design, Analysis and Construction of Shell and Spatial Structures: Proceedings*, 2010.

[9] R. L. Haupt, “Optimum population size and mutation rate for a simple real genetic algorithm that optimizes array factors,” in *Antennas and Propagation Society International Symposium, 2000*, 2000, pp. 1034–1037.

[10] P. Bentley, Ed., *Evolutionary design by computers*. Morgan Kaufmann, 1999.

[11] P. Venkataraman, *Applied optimization with MATLAB programming*. Wiley, 2009.

[12] J. T. Alander, “On optimal population size of genetic algorithms,” in *CompEuro’92.'Computer Systems and Software Engineering', Proceedings*, 1992, pp. 65–70.

[13] C. Mueller, “structureFIT.” www.caitlinmueller.com/structurefit, 2013.

[14] “Silverlight 5.” Microsoft, http://www.microsoft.com/silverlight/, 2013.

[15] J. Schlaich, K. Schäfer, and M. Jennewein, “Toward a consistent design of structural concrete,” *PCI Journal*, vol. 32, no. 3, pp. 74–150, 1987.

[16] “Rhinoceros.” Robert McNeel & Associates, http://www.rhino3d.com/, 2013.

[17] “Revit.” Autodesk, http://www.autodesk.com/products/autodesk-revit-family/overview, 2013.

Figures

Note: The images below are low-resolution previews only. Use separate image files for publication.

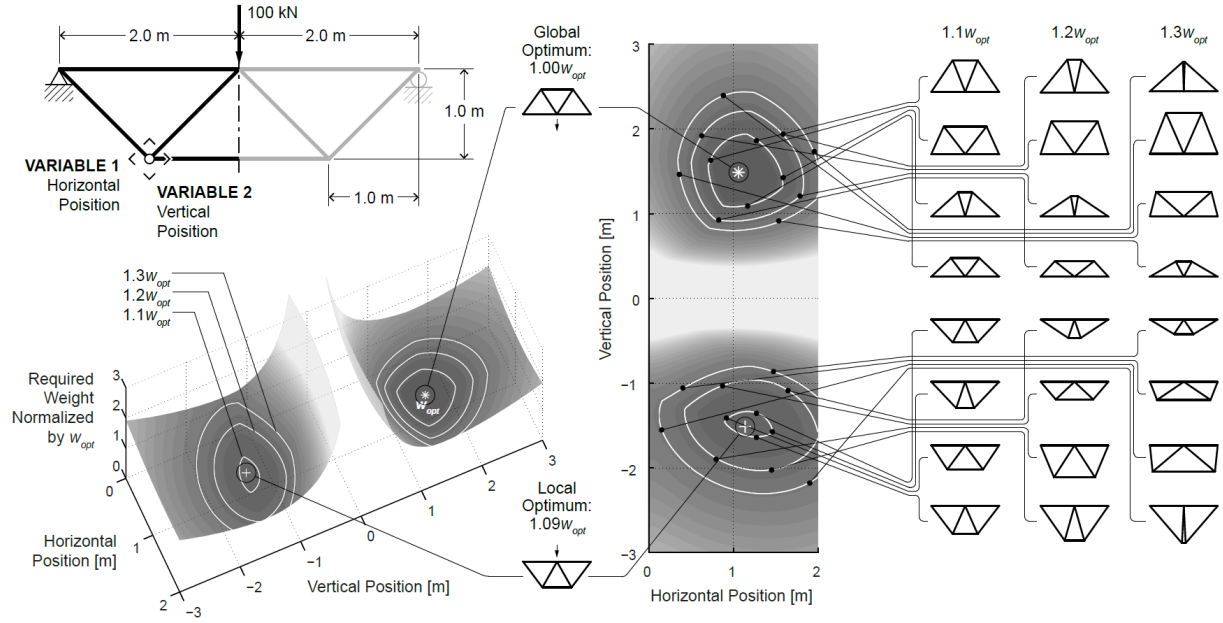

Fig. 1. A weight minimization problem for a seven-bar planar truss with two design variables, and the resulting design space. The global and local optimal solutions are shown, along with isoperformance contours for 1.1, 1.2, and 1.3 times w_{opt}, the optimal design's weight. For each contour, eight designs are highlighted, illustrating the aesthetic diversity of designs that perform almost as well as the best solution.

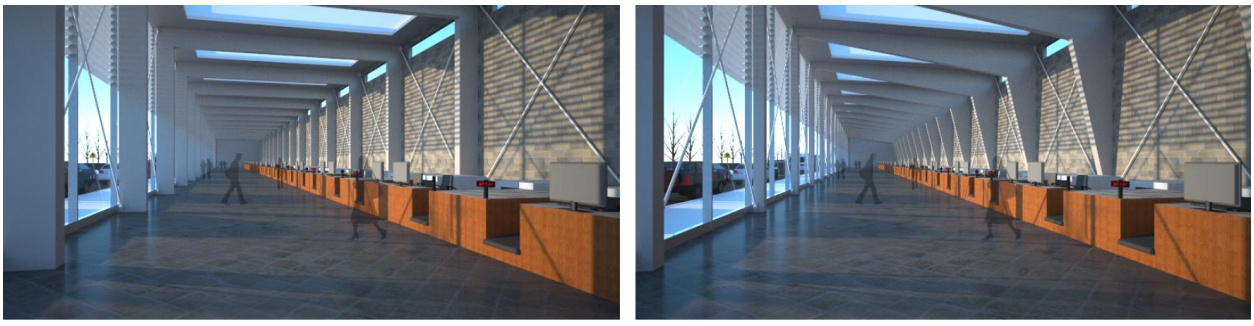

Fig. 2. Two options for the design of the lateral and gravity structural system for an airport terminal: (a) a standard rigid frame, and (b) a shaped rigid frame, which uses a similar amount of material and creates a more architecturally expressive interior space.

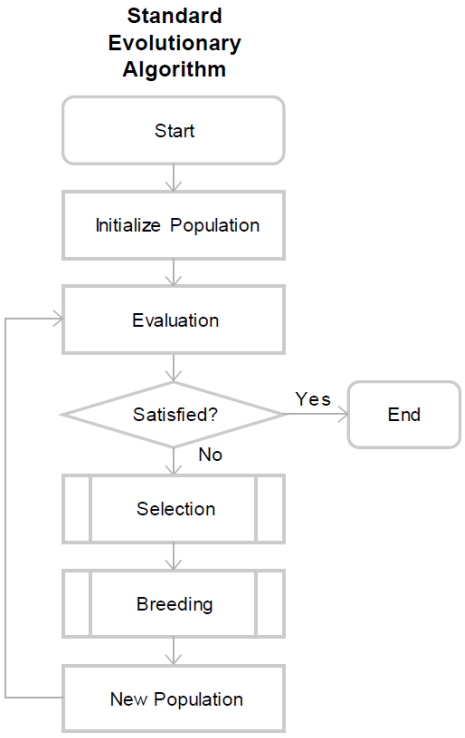

Fig. 3. A generalized standard evolutionary algorithm used in design optimization problems.

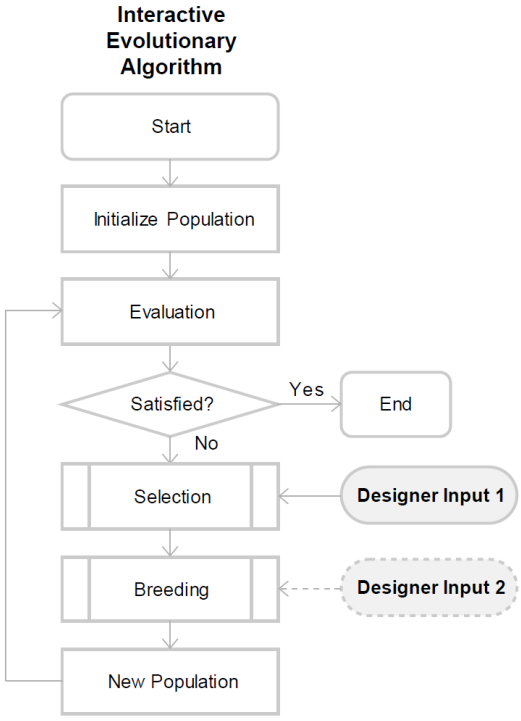

Fig. 4. An interactive evolutionary algorithm, which typically introduces input from a human designer at the *Selection* step. This paper proposes incorporating additional designer input at the *Breeding* step.

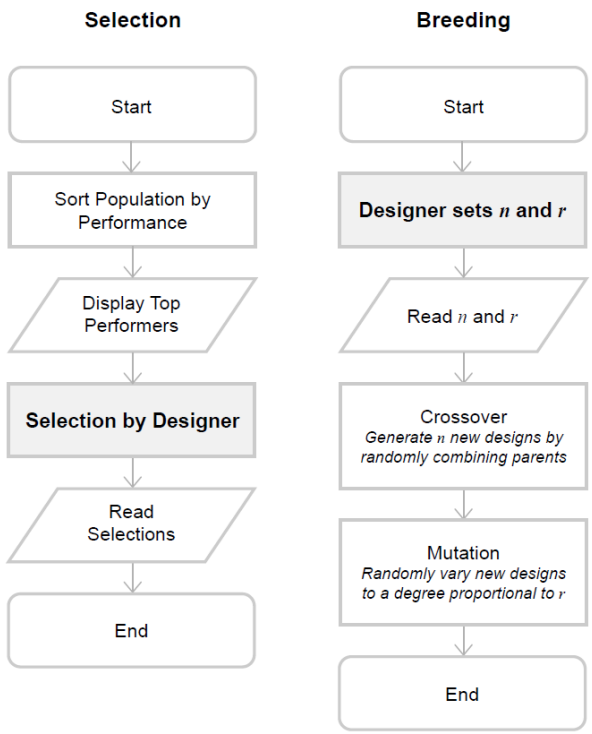

Fig. 5. The *Selection* and *Breeding* subroutines in the proposed interactive evolutionary algorithm, with incorporation of human input highlighted.

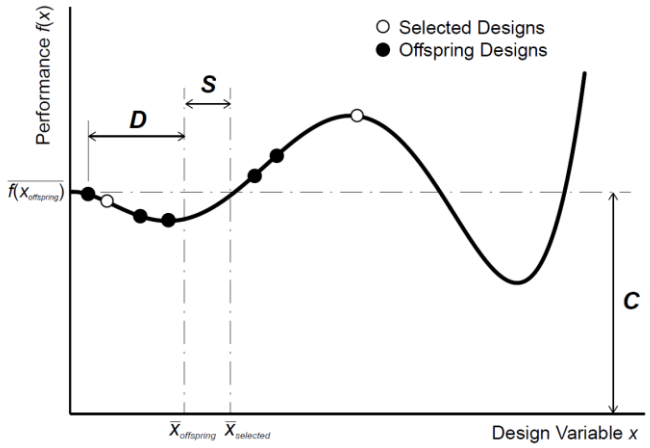

Fig. 6. One-dimensional sample design problem illustrating three new metrics for measuring exploration characteristics for a generation: cost (*C*), distance from selected designs (*S*), and diversity (*D*).

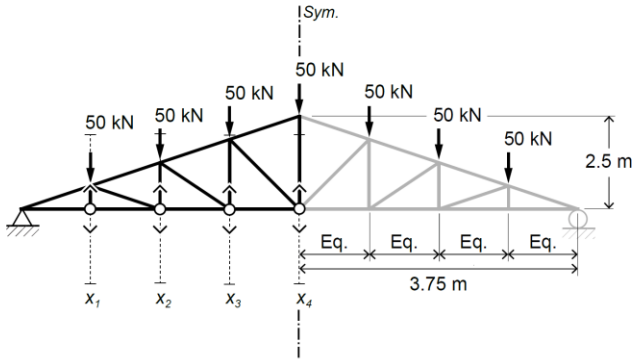

Fig. 7. Problem setup for numerical study: symmetrical planar truss with four nodal coordinate design variables.

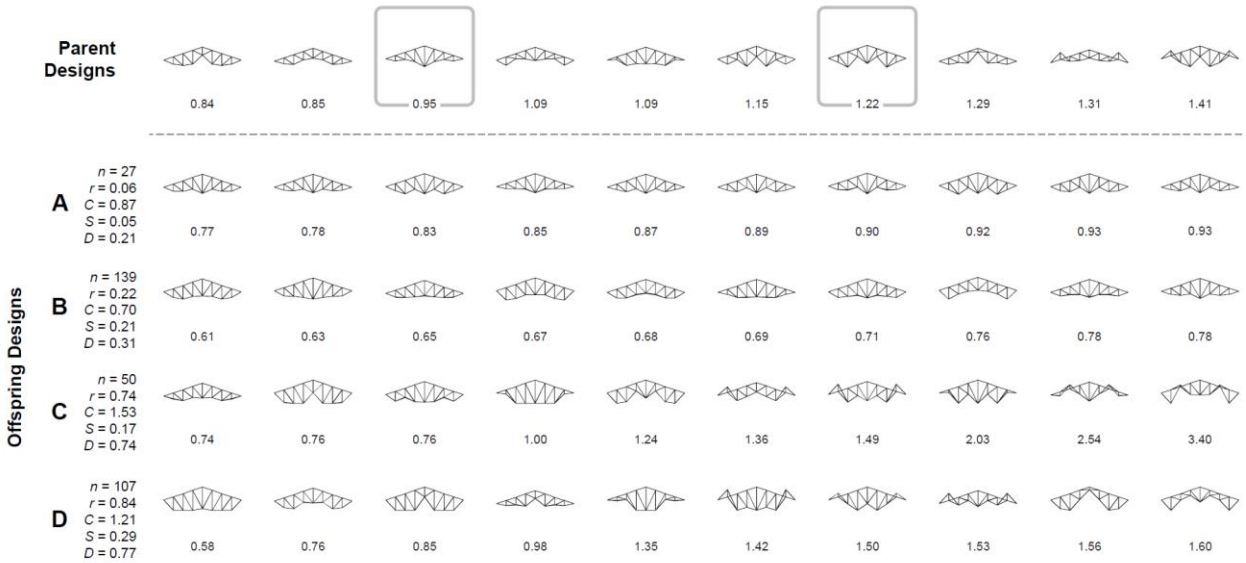

Fig. 8. Top ten designs of four populations generated from the same user-selected parent designs with varying mutation rates and generation sizes, resulting in a wide range of *C*, *S*, and *D* values as noted.

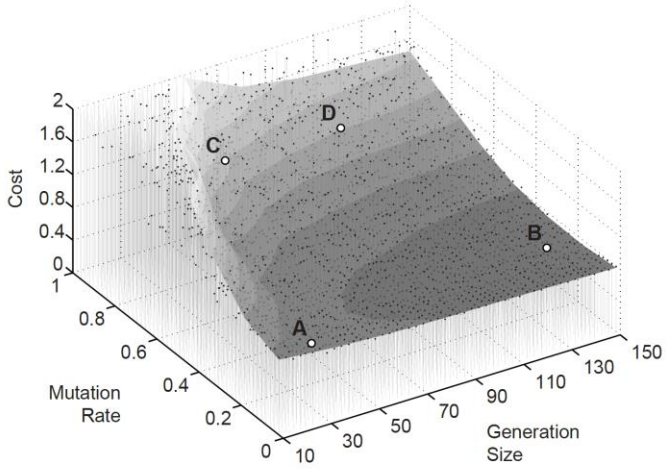

Fig. 9. Summary of *C*, or cost, values resulting from a variation in mutation rate and generation size, with called out points referring to the generations shown in Fig. 8.

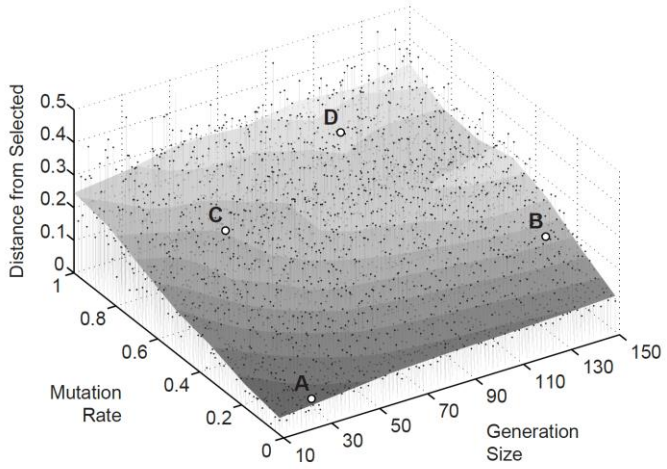

Fig. 10. Summary of *S*, or distance from selected designs, values resulting from a variation in mutation rate and generation size.

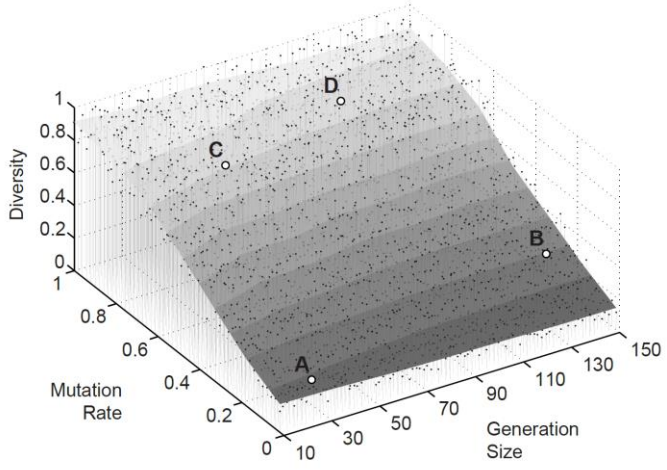

Fig. 11. Summary of *D*, or diversity, values resulting from a variation in mutation rate and generation size.

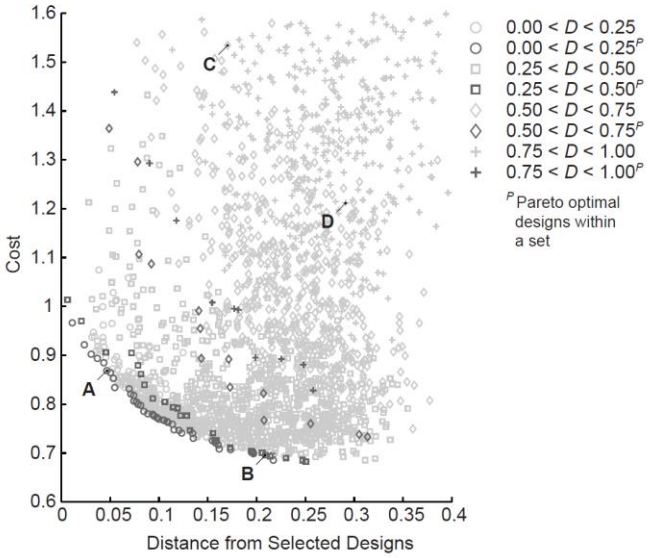

Fig. 12. Pareto frontiers of distance from selected designs and cost, showing tradeoff between the qualitative and quantitative objectives at different *D* values. The called out points refer to the four sample generations shown in Fig. 8.

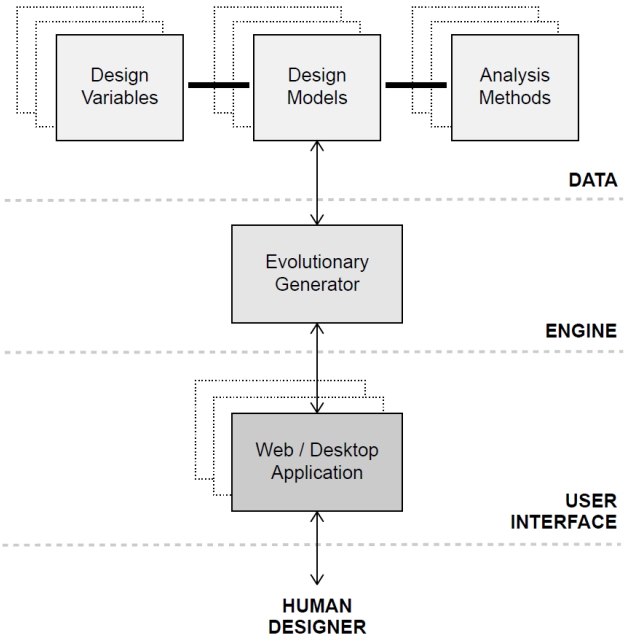

Fig. 13. Software architecture diagram, showing data, engine, and user interface layers. The types of design models design variable, analysis method, and user interface are flexible and can be expanded to apply to a wide range of design problems.

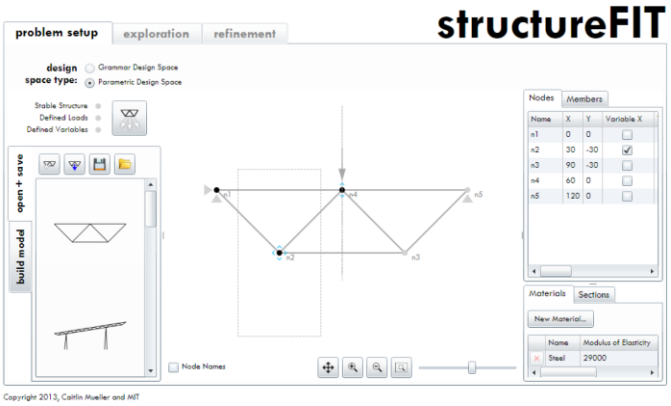

Fig. 14. Setup mode, in which the user can select a predefined structural design problem or define their own.

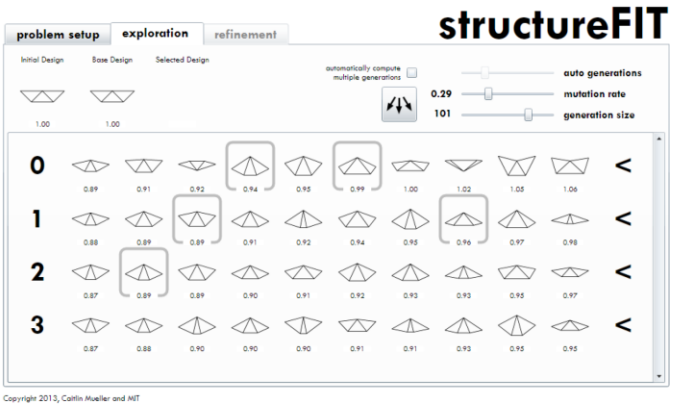

Fig. 15. Exploration mode, in which the user explores the design space using the interactive evolutionary algorithm described in Fig. 4. The sliders on the upper right allow the user to adjust the mutation rate and generation size.

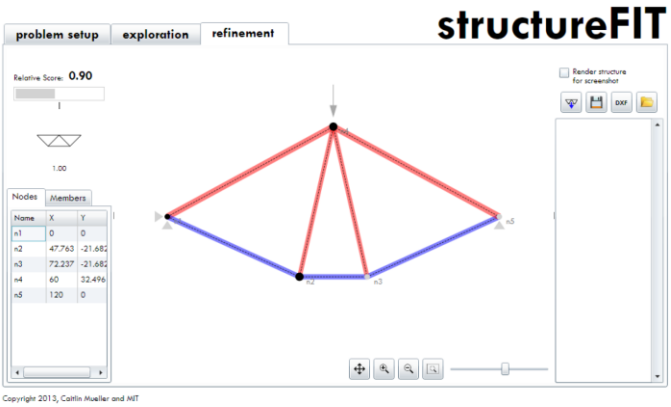

Fig. 16. Design refinement mode, in which the user may make changes to a selected designs by dragging nodes. The performance bar in the upper left updates in real-time to provide instant feedback on design changes.

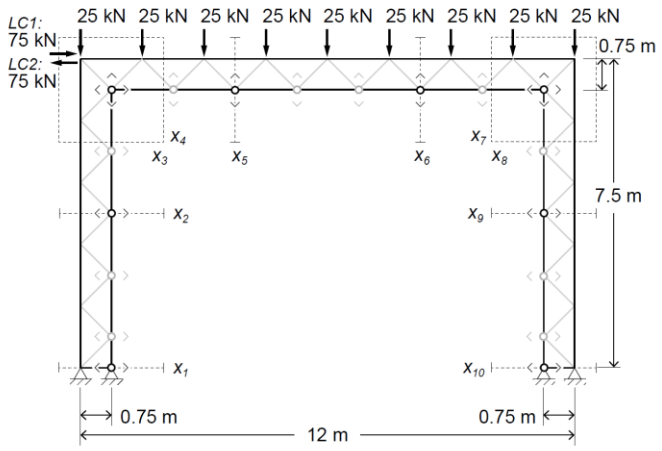

Fig. 17. Problem setup for case study: rigid frame modeled as planar truss, with gravity and lateral loads.

Fig. 18. Excerpt of eight successive generations for an optimization-like exploration approach, in which designs are selected by the user primarily based on quantitative performance.

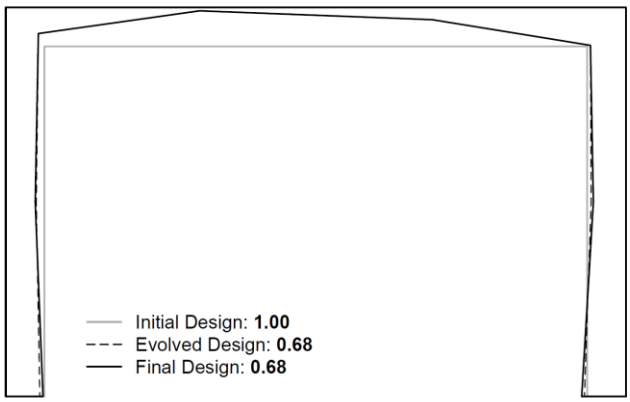

Fig. 19. Initial design, evolved design from the exploration in Fig. 18, and the final design found by the user in the refinement mode.

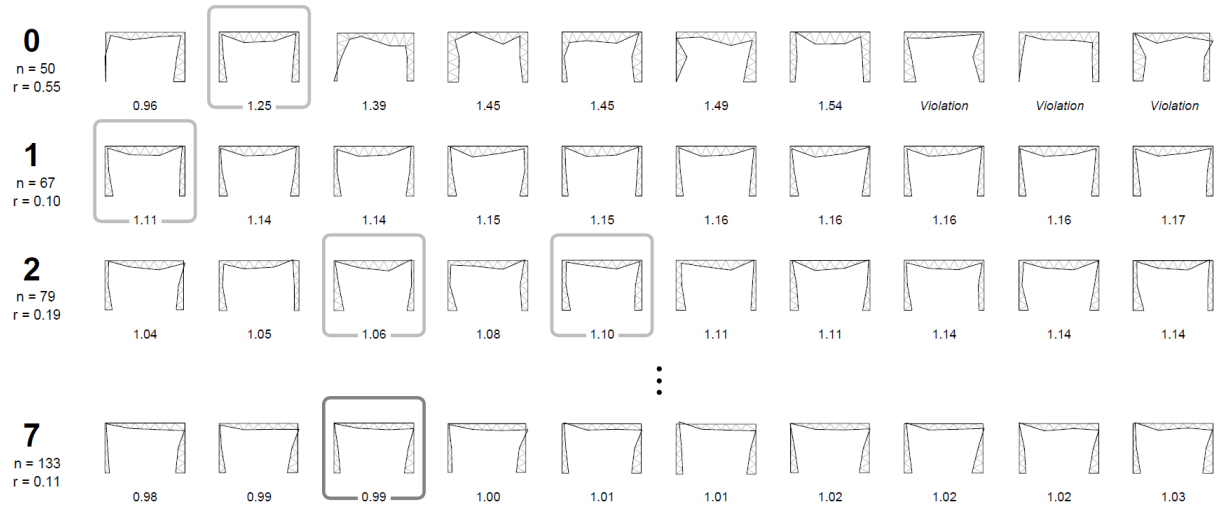

Fig. 20. Excerpt of eight successive generations for a free exploration approach, in which designs are selected by the user primarily based on qualitative aesthetic characteristics.

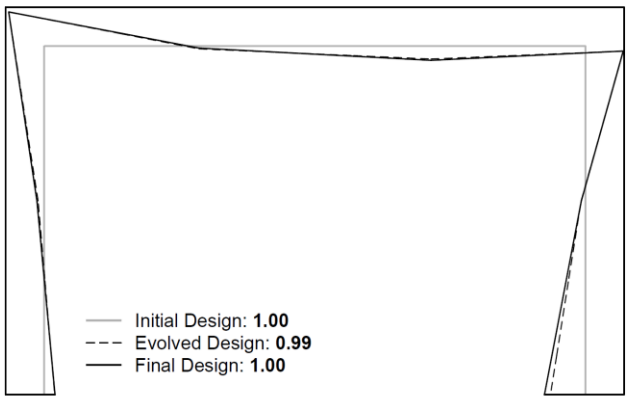

Fig. 21. Initial design, evolved design from the exploration in Fig. 20, and the final design found by the user in the refinement mode.

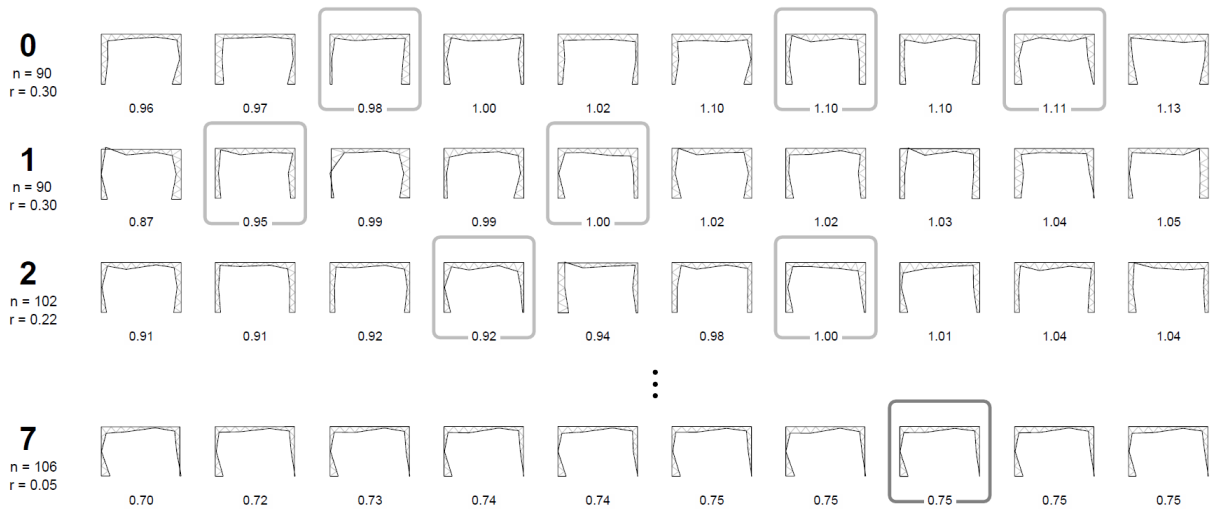

Fig. 22. Excerpt of eight successive generations for a hybrid exploration approach, in which designs are selected by the user based on both quantitative and qualitative performance.

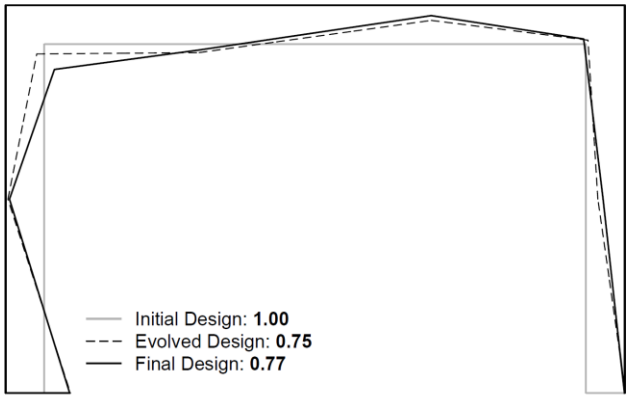

Fig. 23. Initial design, evolved design from the exploration in Fig. 22, and the final design found by the user in the refinement mode.

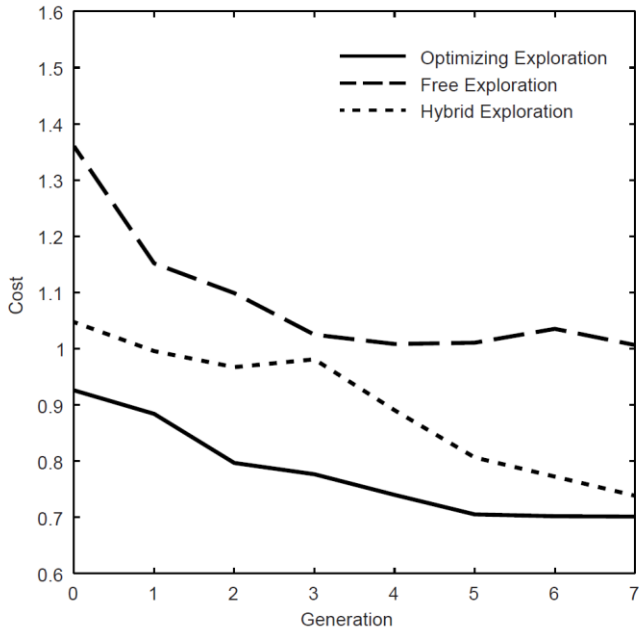

Fig. 24. Cost values for the generations shown in Figs. 18, 20, and 22.

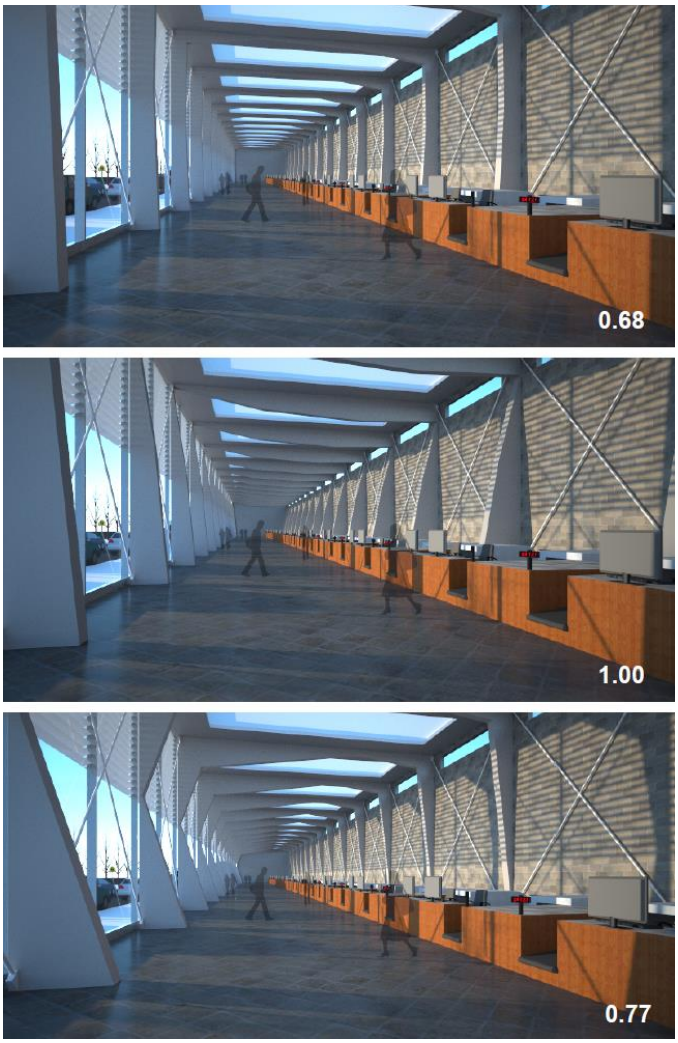

Fig. 25. Three conceptual designs for an airport terminal structural system based on the explorations shown in Fig. 18-23.

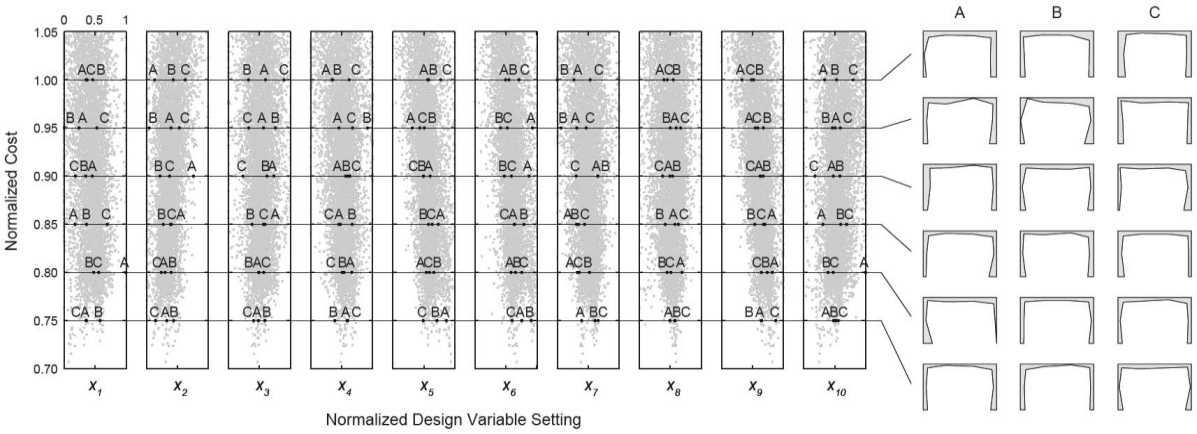

Fig. 26. Design space plot with called out designs (A, B, and C) for the ten-dimensional problem introduced in Fig. 17.