

# Interactive Particle Swarm Optimization for the Architectural Design of Truss Structures

Juliana Felkner, Eleni Chatzi  
Chair of Structural Mechanics, IBK  
Dept. of Civil Engineering  
ETH Zurich, Zurich, Switzerland  
felkner@ibk.baug.ethz.ch, chatzi@ibk.baug.ethz.ch

Toni Kotnik  
Chair of Structural Design, ITA  
Dept. of Architecture  
ETH Zurich, Zurich, Switzerland  
kotnik@arch.ethz.ch

**Abstract**—This paper presents an interactive framework for the design of truss structures with aesthetic criteria. The truss chords are described using NURBS, a tool widely used in computer aided design (CAD) programs to describe free-form geometry. This allows for a convenient interface between the optimization scheme, a particle swarm optimizer, and the user. Driven from the fact that aesthetic design goals are not easily quantifiable, key elements are introduced and implemented herein towards an interactive framework for algorithmic design of truss structures. Within this framework, the user can visually assess interesting solutions, save them for later assessment, actively drive the optimization towards individual aims, re-initialize the optimization with a set of available solutions, or restart the design process. A criterion is introduced as a means of quantifying subjective goals, expressing the similarity of the shape of candidate solutions with respect to reference designs. The framework is tested on a benchmark case and then applied to the design of a truss tower. The effectiveness of the similarity criteria, as well as the ability of the user to drive the process towards specific design goals is demonstrated.

**Keywords**—PSO, NURBS, FEM, structural engineering, user interaction, architecture

## I. INTRODUCTION

Structural optimization typically aims at achieving efficient structural performance while at the same time minimizing structural weight under a set of constraints, such as maximum allowed displacements and stresses, buckling behavior, and natural frequencies. Typical design variables include geometric parameters (positions, cross-sectional shapes, etc.), material parameters (strength, density, etc.) or topological parameters, e.g. material distribution. Furthermore, the structure to be optimized, as well as the design space, may be continuous, discrete or both [1], [2].

Since the design space can be quite large, and the optimization problem may be non-convex (e.g. in topology optimization), population based heuristic search optimization methods have been proposed and used successfully in such problems, thereby outperforming gradient-based methods. Rajan used a genetic algorithm (GA) to perform shape, sizing and topology optimization of truss structures [3]. Xie introduced (bi-directional) evolutionary structural optimization ((B)ESO) to perform topology optimization of continuous structures [4].

Bel Hadj Ali et al. [28] have used a genetic algorithm based optimization scheme for the design of a tensegrity-based footbridge. In previous work of the second author [29], a GA optimization process was enforced for the optimal performance based design of frames based on a uniform damage criterion. Coello et al. [30] utilized a similar scheme for optimal design of truss structures, and several other researchers have most commonly relied on the use of GA's [31], [32] as the heuristic tool for the structural optimization problem. Other than GA's, which are fairly popular, several other heuristic approaches [33], [34], [35], [36] have been proposed for the solution of structural optimization problems while often incorporating response behavior criteria [37], [38].

In this paper, the particle swarm optimization (PSO) method is implemented as the optimizer tool. This was introduced by Kennedy and Eberhart in 1995, and essentially works on the basis of mimicking the behavior of flocks or "swarms" searching for food or escaping a predator [23]. The interested reader is referred to [25] for an extensive review on successful applications of the PSO method. The advantages of PSO compared to the GA include a simpler setup, an often faster convergence rate, and computational efficiency, while still providing the same quality solutions [5].

Fourie and Groenwold were the first to apply the particle swarm optimization (PSO) method to the design of truss structures, confirming its efficiency compared to GA's [24]. Since then, many variations of the algorithm, when applied to structural optimization, have been proposed and implemented [6], [39], [40], [42], [44].

Despite its wide applicability so far, within the framework of structural optimization, little attention has been given to incorporate aesthetic, or more specifically, architectural criteria, directly into the optimization process. In fact, mostly quantifiable, i.e. rather objective, criteria have been used. Shea assigned an aesthetic measure to truss structures based on the uniformity of their angles and the golden ratio [7]. Schein used volumetric constraints as architectural criteria in the optimization of a space frame by defining a volume in the structure that must not be crossed by members [8]. Pugnale

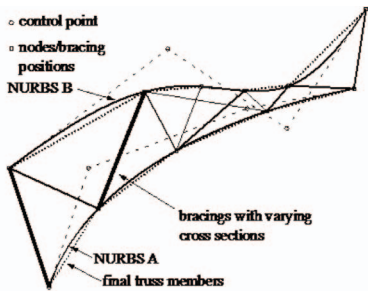


Figure 1. The general truss problem considered in this work: curved top and bottom chords connected with a variable number of bracings.

optimized a continuous roof structure using the position of the columns as architectural constraints [9]. The inherent difficulty met in these approaches is that, in general, aesthetic criteria are highly subjective to the architect, and, therefore, not quantifiable in a straightforward manner [10]. In recent work by Bailey and Raich [43] user influence is introduced through a period of training neural networks with several users and subsequent clustering of their preferences. Instead, the framework introduced herein aims at an online interaction customized to a given user. The versatility stems from the use of the similarity measure introduced later on.

On the other hand, capitalizing on the trial-and-evaluate characteristic of population type optimization methods, non-quantifiable objectives have been successfully achieved in interactive evolutionary computation for engineering and design applications [15], [16], [17]. Moreover, Hu and Eberhart recently analyzed the human-swarm interaction during the optimization process, by means of a computer game, and concluded that the combination of computational power and human intuitive knowledge is advantageous for complex tasks [22].

Interactive PSO was introduced in 2005 [20], and only very few applications have been presented so far, including chemical engineering [20] and facial composite generations for criminal suspect profiling [21]. In this paper, we explore the potential of interactive PSO for architectural design with structural optimization.

A second aspect of the novel contribution of this work lies in the use of non-uniform rational b-spline (NURBS) curves to describe parts of the structure. NURBS curves are a generalization of Bézier (b-) splines, and are, together with NURBS surfaces, widely used in computer-aided design (CAD) programs, due to the computationally efficient description of free-form geometry. In addition, employing the NURBS description allows us to use the initial architectural design as an initial configuration and perform the optimization on these natural variables. Therefore, NURBS are ideal for describing problems of structural optimization with architectural freedom of design.

To summarize, the contributions of this paper are the application of an interactive (user influenced) PSO-based

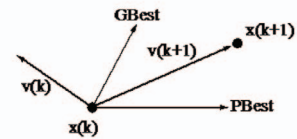


Figure 2. Particle swarm optimization scheme

framework to the algorithmic design of truss structures, as well as the use of NURBS in the optimization process, with the purpose of reducing computational complexity and increasing the architectural freedom of design by allowing for free-form shapes that meet structural requirements.

The paper is organized as follows. The next section details the background for the tools used in this work. Section III describes the proposed algorithm. Section IV shows the interactive design of a truss tower. Finally, Section V concludes the paper and discusses future research directions.

## II. BACKGROUND

### A. NURBS Curves as Truss Chords

The development of NURBS curves and surfaces was pioneered in the 1950s by the French engineers, Bézier and Casteljau, in their research for a sound mathematical description of free-form structures, in their case, for car bodies. A complete description of NURBS curves is beyond the scope of this paper, and can be found in standard textbooks [11]. In brief, a NURBS curve is described by its degree, a set of weighted control points, and its knot vector. A single variable  $u \in (0,1)$  is used to parameterize the position of a point on the curve. In addition, at each point the derivative of the NURBS can be computed, which is again a NURBS curve tangent at each point to the original curve.

From an interaction perspective, using NURBS curves (which can also represent straight lines) is appropriate, since they allow the maximum freedom of design and are standard in CAD software. Thus, in order to conveniently fuse early stages of the design process with structural optimization, NURBS are a versatile tool to employ.

In this research, the general problem of shape, topology and size optimization is considered by looking into the problem of a truss structure whose geometry is defined by a curved chord, in order to minimize its weight under a given set of constraints (see Fig. 1). The shape of that chord is given by a NURBS curve, as described above, and a given maximum number of divisions is assumed. Shape optimization considers finding the optimal position of the control points. Topology optimization deals with finding the optimal number, position and connectivity of the nodes between the cords. Finally, the objective of sizing is to find the optimal cross sectional area.

### B. Particle Swarm Optimization (PSO)

At each iteration  $k$ , a candidate solution (represented as a particle) in the swarm is described by a position  $x(k)$  encoding a candidate solution, and a velocity  $v(k)$  encoding the direction and magnitude of motion in the search space. The position of the particle for iteration  $k+1$  is updated as [23]:

$$x(k+1) = x(k) + v(k+1) \quad (1)$$

and the velocity of the next iteration is found from (see Fig. 2)

$$v(k+1) = w*v(k) + r_1*c_1(PBest-x(k)) + r_2*c_2(GBest-x(k)) \quad (2)$$

where PBest is the best position that the particle has encountered so far, and GBest is the overall best position that the swarm has encountered. The inertia weight  $w$ , and the cognitive and social factor,  $c_1$ ,  $c_2$ , are settings of the PSO algorithm, scaling the influence of the respective terms. Selecting PSO parameters that yield good performance has, therefore, been the subject of much research [12], [13], [14]. Based on common practice met in the literature  $c_1$ ,  $c_2$ , are set as  $c_1=c_2=2$  and the weight is reduced linearly at every iteration, as

$$w(k) = (w_{\max}-w_{\min})(k_{\max} - k)/k + w_{\min} \quad (3)$$

with  $w_{\max} = 0.9$  and  $w_{\min} = 0.4$  being the initial and final inertia weight, and  $k_{\max}$  being the iteration number when the final inertia weight is reached. Typically,  $k_{\max}$  is set to 75% of the maximum iterations.

The factors  $r_1$  and  $r_2$  in (2) are random numbers uniformly distributed in the range [0 1]. Additionally, a maximum velocity for the particle is set in order to ensure a smooth coverage of the search space. If, during the optimization, a particle hits its design boundary, the direction of the velocity is reversed, i.e. the particle bounces off the boundary. Finally, in order to keep track of the global best solution, the best particle is always kept in the swarm, and its position is only updated if a new global best position is found.

The goal of the optimization process is to minimize the objective value, typically the weight or volume of the structure for a given set of constraints. These constraints are added as a penalty to the objective function and are defined as normalized nonnegative values. Thus, the optimization problem is formulated as follows.

$$\text{Minimize } f_{\text{obj}}(x) + f_{\text{penalty}} \quad (4)$$

with the objective function being the structural weight

$$f_{\text{obj}}(x) = \sum_i x_i L_i \rho_i \quad (5)$$

where  $x_i$  is a design variable reflecting the cross sectional area of element  $i$ ,  $L_i$  is the length of that element and  $\rho_i$  the material density.

The form of the penalty function in (4) is similar to [3]

$$f_{\text{penalty}} = C_1 \cdot f_{\text{Min}} \cdot \sum_i \max(g_i, 0) \quad (6)$$

where  $C_1$  is a user-specified, problem dependent constant set to  $C_1=1$ ,  $f_{\text{Min}}$  is the minimum objective value over all feasible solutions in the previous iteration, and  $g_i$  indicates a constraint function. We consider displacement constraints as

$$g_i = u_{\max}/u_a - 1 \leq 0 \quad (7)$$

where  $u_{\max}$  is the maximum nodal displacement, and  $u_a$  is the admissible displacement. Stress constraints are expressed as

$$g_i = |\sigma_{\max}|/\sigma_{\text{all}} - 1 \leq 0 \quad (8)$$

where  $\sigma_{\max}$  is the maximum axial stress in the members, and  $\sigma_{\text{all}}$  is the allowable stress. Additionally, the Euler buckling constraint is imposed as

$$g_i = A\sigma_x/(\pi^2 EI/L^2) - 1 \leq 0 \quad (9)$$

where  $A$  is the cross sectional area,  $E$  the Young's modulus and  $L$  the length and  $I$  the area moment of inertia of the member, and  $\sigma_x$  is the axial stress in the member.

Finally, heuristic constraints such as length of members and truss stability, i.e. no mechanism, are included. To determine whether the structure is a mechanism, the degrees of freedom (DOF) are computed using Gruebler's formula, which for a planar truss, simplifies to

$$\text{DOF} = 2*(\# \text{ nodes}) - (\# \text{ members}) - 2*(\# \text{ fixed nodes}) \quad (10)$$

For  $\text{DOF}>0$ , the truss is a mechanism. Another heuristic constraint is used to reflect the similarity of the truss to reference designs. The approach is to compare the position and the orientation of the NURBS at the bracing positions. For the former, the NURBS itself is evaluated, resulting in a pair of  $(x,y)$  coordinates of the bracing position. For the orientation, the derivative of the NURBS is evaluated, resulting in the coordinates  $(x',y')$  of the tangent vector at this position. The coordinates are gathered in vectors

$$\mathbf{x}\mathbf{B} = (x_1, x_2, \dots, x_N), \quad \mathbf{y}\mathbf{B} = (y_1, y_2, \dots, y_N) \quad (11)$$

$$\mathbf{x}'\mathbf{B} = (x'_1, x'_2, \dots, x'_N), \quad \mathbf{y}'\mathbf{B} = (y'_1, y'_2, \dots, y'_N) \quad (12)$$

for the candidate solution and

$$\mathbf{x}\mathbf{B}_{\text{ref}} = (x_{1,\text{ref}}, x_{2,\text{ref}}, \dots, x_{N,\text{ref}}), \quad \mathbf{y}\mathbf{B}_{\text{ref}} = (y_{1,\text{ref}}, y_{2,\text{ref}}, \dots, y_{N,\text{ref}}) \quad (13)$$

$$\mathbf{x}'\mathbf{B}_{\text{ref}} = (x'_{1,\text{ref}}, x'_{2,\text{ref}}, \dots, x'_{N,\text{ref}}), \quad \mathbf{y}'\mathbf{B}_{\text{ref}} = (y'_{1,\text{ref}}, y'_{2,\text{ref}}, \dots, y'_{N,\text{ref}}) \quad (14)$$

for the reference shape, and the similarity constraint is computed as

$$g_i = 0.5 ( |\mathbf{x}\mathbf{B} - \mathbf{x}\mathbf{B}_{\text{ref}}| / |\mathbf{x}\mathbf{B}_{\text{ref}}| + |\mathbf{y}\mathbf{B} - \mathbf{y}\mathbf{B}_{\text{ref}}| / |\mathbf{y}\mathbf{B}_{\text{ref}}| ) + 0.5 ( |\mathbf{x}'\mathbf{B} - \mathbf{x}'\mathbf{B}_{\text{ref}}| / |\mathbf{x}'\mathbf{B}_{\text{ref}}| + |\mathbf{y}'\mathbf{B} - \mathbf{y}'\mathbf{B}_{\text{ref}}| / |\mathbf{y}'\mathbf{B}_{\text{ref}}| ) - \text{LIM} \leq 0 \quad (15)$$

where LIM is a user-defined constant, representing how much room for exploration the algorithm has on the similarity. The use of this constraint is elaborated upon further in the next

section. A heuristic constraint for the topology is detailed in Section III.C.

If any of the heuristic checks fails, a large penalty value is assigned, and the structure is not analyzed further. Otherwise, structural analysis is performed using ANSYS, a well-known engineering simulation software. The idea behind selecting this package, which comes with a strong finite element (FE) solver, is to later expand this framework of design to surface and 3D structures. Finally, the results, i.e. the weight of the structure, the new geometry, the maximum displacement, and maximum stress values are returned to the PSO procedure, and the objective and penalty functions are computed. If the FE analysis fails, due to, e.g. an unstable structure, a penalty value of  $5*f_{Min}$  is assigned. The implementation of the algorithm is done in MATLAB using the PSO and NURBS toolboxes [26], [27].

### C. Interactive Optimization

Typically, interactive evolutionary computation is performed by asking the user to assess the fitness of candidate solutions, for example, by ranking them [15]. As an example in structural optimization, Machwe and Parmee perform cantilever bridge design using an interactive GA, by allowing the user to rank the 10 best solutions in each generation. A numerical value is then assigned according to this ranking, and then used in a weighted sum with the structural criteria to compute the overall fitness of a solution [18]. However, this approach requires that the problem and its variables be well defined at the onset of the optimization, which is not the case for the architectural design process. Architectural design is highly complex and does not follow an analysis-synthesis approach, mainly because of the large number of possible design solutions. Rather, to reduce the complexity of the problem, an architect may identify a small group of highly subjective key aims, the “primary generators”, which then give rise to possible solutions. These can then be analyzed, allowing for the requirements to be clarified and tested [10].

In addition, structural design, especially when dealing with exposed structural members, is as highly interdisciplinary design process, requiring a close collaboration between the architect and the structural engineer. This is especially the case for large-scale, iconic structures such as bridges or tall buildings, for example, the John Hancock Center in Chicago. Here, a human, deliberate choice and arrangement of material is required, and could not be simulated by an algorithm.

Therefore, the aim of interactive optimization for architectural design should not be to simply generate solutions, which are then assessed by the architect. Rather, its objective is to use the available computational power in order to support the exploration of feasible design variations. For this, Candy and Edmonds identified three key features [41]. First, the user must be able to update the design rules, i.e. the constraints, easily during the process. Second, the user must be given support for the evaluation of the results, e.g. allow the user to ask “why or why not about the results?”. Finally, the user

should have the ability to compare results stemming from different constraint sets.

Therefore, the typical interactive optimization approach of solely assessing the candidate solutions during the optimization process is not sufficient for successful design generations. Rather, in view of the three features above, it is proposed that an interactive optimization process for structural design has, but is not limited to, the following components:

1. use the initial design(s) of the architect
2. set constraints and ranges for design variables
3. visually assess solutions
4. drive optimization towards architect’s design goals
5. keep track of all solutions for later assessment
6. support the iterative design process by easily allowing to update constraints and design objectives

In the next Section, a possible implementation for truss design is discussed.

## III. ALGORITHM

### A. User Interaction Framework

To incorporate the components 1 - 6 detailed in Section II-C, the following user interaction framework is proposed.

1. At the beginning of the process, the designer can draw the chords of the truss by means of a NURBS curve. The user could also import a design from another program, which is then automatically parsed for the NURBS structure. This initial design is used as a reference for the similarity measure.
2. The designer can specify a) the amount of shape similarity (15) below which candidate solutions are penalized, b) the degree of influence of the visual assessment (“high”, “medium”, “low”), and c) the number of segments into which the NURBS is to be divided to create the truss. Based on this, the bracing is assigned with uniform spacing for the initial design.
3. - 5. As new candidate solutions become available at subsequent iterations of the optimization, they are shown to the designer together with their weight, so that a comparison can be made. In addition, the current optimum, as well as the initial design, is shown. The user can then select preferred designs. A selected design is marked and saved for possible later assessment. In addition, in subsequent iterations, the similarity measure is evaluated for every saved design, and the best result is kept for the objective term. This allows the designer to update the design constraint dynamically during the optimization process, should the algorithm reveal an interesting solution. Finally, depending on the user’s setting of the influence of the visual assessment, the weight of the structure is artificially reduced, i.e. a “high” influence results in a weight reduction of, e.g. 35%. This effectively drives the simulation towards the ideas of the user.

6. The user is given three choices to continue after the design selections. First, the optimization can proceed with the generation of the next candidate solutions. Second, the optimization can be restarted using a selection of candidate solutions as initial designs (reseeding). Or third, the designer can go back to the initial design of the NURBS and start over (restarting). This ensures an iterative design process.

The encoding and decoding procedures are described herein by a general formulation; simplified variants apply to suit a specific problem description.

### B. Particle Encoding

A particle  $x(x_1, \dots, x_n)$  encodes the shape, topology and sizing variables as

$$x = [\text{shape (control points, bracing pos.)} \mid \text{topology} \mid \text{sizing}]$$

The shape variables are the coordinates of the control points, followed by the linear coordinates for the nodes, i.e. the bracing positions. The topology of the truss is represented as a connectivity matrix  $C$ , whose rows represent the nodes on one chord, the columns represent the nodes on the second chord, and its entries  $c_{ij}$  are

$$c_{ij} = 0 \text{ if nodes } i \text{ and } j \text{ are not connected, and} \\ c_{ij} = 1 \text{ if nodes } i \text{ and } j \text{ are connected}$$

The  $c_{ij}$  values are therefore binary, which effectively is treated by rounding towards 0 or 1. As an example, the truss shown in Fig. 3 has the connectivity matrix (with rounded values)

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (16)$$

Problems involving more NURBS envelopes would be treated via the use of several connectivity matrices. Finally, the sizing variables, i.e. the cross sectional areas, are used. Depending on the problem, these can be grouped to represent a set of members, or used for each member individually.

### C. Particle Decoding

For decoding, the NURBS curves are recreated based on the control points. The coordinates of the bracing positions are determined by evaluating the NURBS curves at the linear position values requested by the PSO algorithm. The NURBS curves are discretized into a polygonal chain at the bracing positions. For standard truss design, where members are usually chosen as linear, the NURBS tool is used to provide an overall envelope and linear segments are then chosen between division points (see Figure 1).

Next, the connection matrix for the topology is reassembled and heuristic checks on the structure are performed. It is

straightforward to carry out checks for member lengths, since the curves are discretized. The polygonal chains can also be used to determine intersections between the NURBS, if necessary.

The definition of the connectivity matrix  $C$  also allows for a heuristic check against a reference  $C_{\text{ref}}$ . This reference matrix is set up with entries of 1's for possible connections, and 0's for connections that are not allowed. Upon decoding, an entry-wise logical AND operation is performed between  $C$  and  $C_{\text{ref}}$ , which effectively removes any undesired connections.

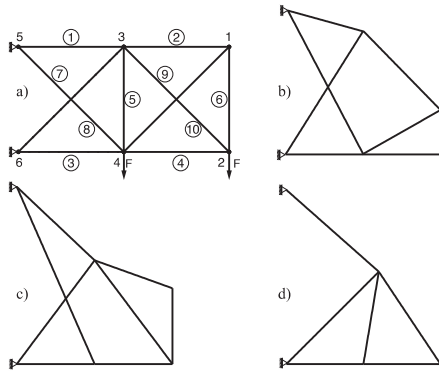
If the end-points of the two NURBS curves are allowed to coincide, and if this situation occurs, then the two nodes are merged by removing the obsolete connections in the connection matrix.

## IV. RESULTS

### A. 10-bar-truss validation and NURBS

First, without any user interaction, the PSO method is validated with previous results for shape, sizing and topology optimization of the 10-bar-truss shown in Fig. 3a). The problem is a well-established truss optimization benchmark. The properties and ultimate requirements are set up as described in [3], considering stress, displacement and buckling constraints, as well as discrete variation (I-beam sections) of the cross sectional areas. Using the suggested approach, the top chord is designed as a NURBS curve, and, in a first step, in order to closely comply with the benchmark, the curve's control points are only moved vertically during the optimization in an interval of 4.57 to 25.4 m (as in the reference paper [3], whereas the middle node is fixed to the center of the curve.

Figure 3c) shows the final design after 50 iterations with 40 particles. Compared to the benchmark solution in [3] (Fig. 3b), a slightly different topology, and a weight reduction of about 11% have been achieved, with the final weight being 1298kg versus 1454kg. As noted in [3], the shape of two designs may be very different, whereas their weights can only differ slightly by a few percent, which suggests the presence of e.g. multiple local minima for structurally optimal solutions. This, in turn, suggests that architectural criteria can be considered without necessarily increasing the weight, and thus the cost of the structure. Note that this solution includes two members that carry no force (Nr.2 and Nr.6). Their inclusion in the final solution is an artifact of using a NURBS for the top chord, as the algorithm never removes member 2. To improve this situation, in Fig. 3d), the range of motion of node 1 has been modified to allow it to be merged with node 2. In addition, the control points of the NURBS are allowed to move freely, and the middle top node is move freely along the length of the chord. As seen in Fig. 3d), the no-force carrying members have been removed, and the weight has been further reduced to 1210kg (or 17% compared to the solution in Fig.3b). Table I compares the final solutions.



**Figure 3.** 10-bar truss optimization problem. a) ground structure, b) benchmark solution [3], c) optimal solution (this work) with constrained NURBS, d) optimal solution (this work) with freely moving NURBS.

Note that the use of NURBS for this rather small scale example is merely instructive. The advantage of NURBS in using few parameters to describe the shape of the chord becomes more beneficial for larger truss structures.

**Table I:** Comparison of the solutions for the 10-bar truss.

Variable	Benchmark[3] (Fig. 3b)	NURBS-PSO	
		Constrained (Fig. 3c)	Free (Fig. 3d)
Weight (kg)	1454	1298	1210
c/s area 1 (m <sup>2</sup> )	0.00535	0.00774	0.0104
c/s area 2 (m <sup>2</sup> )	0.000967	0.00327	0
c/s area 3 (m <sup>2</sup> )	0.0115	0.00603	0.00437
c/s area 4 (m <sup>2</sup> )	0.00774	0.00379	0.00481
c/s area 5 (m <sup>2</sup> )	0	0	0.00439
c/s area 6 (m <sup>2</sup> )	0.00232	0.01039	0
c/s area 7 (m <sup>2</sup> )	0.00367	0.00237	0.00481
c/s area 8 (m <sup>2</sup> )	0.00264	0.00379	0
c/s area 9 (m <sup>2</sup> )	0	0.00108	0.00665
c/s area 10 (m <sup>2</sup> )	0.0162	0	0
y-coord. node 5 (m)	18.01	20.85	20.71
y-coord. node 3 (m)	11.30	12.20	10.99
y-coord. node 1 (m)	5.72	8.92	0

### B. Design of a Truss Tower

The user interaction framework detailed in Section III.A is applied in the abstract design of a planar truss tower. Formal, i.e. non-optimized, requirements are a bottom width of 100m, top width of 20m, a height of 300m, a horizontal bracing at 2/3 of the height to mimic a platform, and the use of steel as the material. The tower is pinned at the bottom, and subject to horizontal wind forces, which are applied at the bracing positions proportionally to their height. The user defines an initial “proposed” geometry using the interface (Fig. 4a, Fig. 5a). In that interface the options and active criteria for optimization are also selected. Regarding structural specifications, the S4 x 7.7 profile (I-beam) is used for each member, i.e. sizing optimization is not performed, as the focus is on affecting the shape. Buckling constraints, as well as length constraints on the members are considered. The initial number of segments on the profile is set to six. For this, three bracing positions are considered between the ground and the platform and one between the platform and the top. Thus, there are seven possible initial bracing positions. The reference connectivity matrix is defined as

$$C_{\text{ref}} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \quad (17)$$

where the top left entry corresponds to the connection between the bottom two nodes. If during the optimization, the bracing positions are sufficiently close, i.e. the difference of their respective u coordinates on the NURBS is less than 0.05, which translates to 15m on the tower design, they are merged by updating the connectivity matrix accordingly. Table II summarizes the settings of the problem. A total of 40 particles are used, five of which are initialized to the design of the user with  $C_{\text{ref}}$  as bracing topology. The remaining particles are random. The maximum number of iterations is set to 300.

**Table II:** Data for truss tower design

Item	Value
Young’s Modulus	210 GPa
Density	8000 kg/m <sup>3</sup>
Maximal allowed stress $\sigma_{\text{all}}$	250 MPa
Wind loading at Top, $F_{\text{Wind}}$	10kN
Height, H	300m
Width at bottom, $W_{\text{bottom}}$	100m
Width at top, $W_{\text{top}}$	20m
Maximal allowed displacement	$H/500 = 0.6\text{m}$
Maximal bracing length	120m

Figure 4b) shows the interface after a few iterations. On the left, the current candidate solutions are shown, from which the user can choose by clicking on the image. Only solutions that have passed the structural constraints of stress and displacement are shown. In the middle, the current optimum and the initial design are shown as comparison. Finally, on the right, all the designs selected so far are visible. In addition, the weight of each design is displayed above it. The buttons for proceeding, reseeding, and restarting are shown at the bottom.

Three design case studies are performed based on an initial design (Fig. 5a), mimicking different types of architect-engineer collaboration. First (Fig. 5b), the design is discarded, initial seeds are randomly generated within the confines of the site requirements, no designer interaction/similarity measure is employed, and the tower is optimized for weight without architectural input. In the next case (Fig. 5c), the initial profile is the designer defined one, and the similarity constraint is set to LIM=100%. This corresponds to the situation of an architect presenting a design to the engineer, and giving him the possibility to alter it only within a limited range, e.g. only the cross-bracings. Finally, in a case with (online) user interaction (Figure 5d), the user can actively modify the design of preference during the process, as in a 50/50 collaboration, balancing mass and design. For this, the initial design is again utilized to drive shape similarity, but in addition, alternative designs approved by the user during the optimization receive a weight reduction of 35% (online feature).

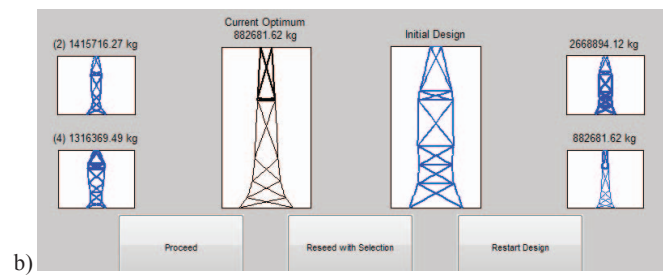
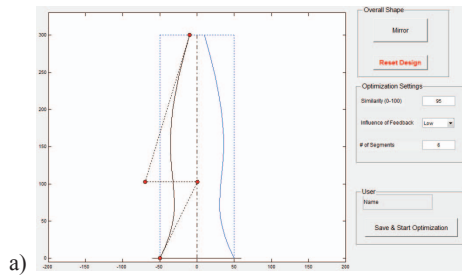


Figure 4. a) Screenshot of the initial interface where the user can design the profile of the tower, and define optimization settings . b) The user interface after a few iterations, showing candidate solutions on the left that the user can select, and previously selected solutions on the right.

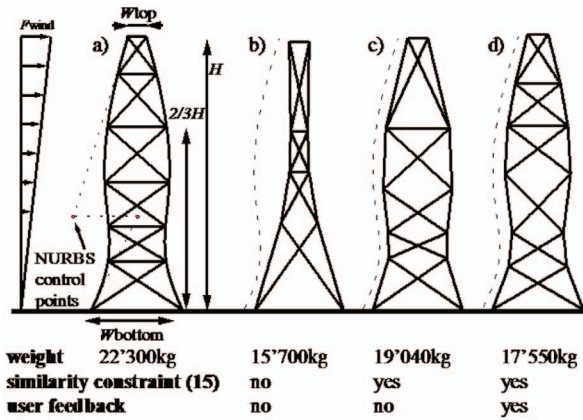


Figure 5. a) initial user design b) optimal solution with only structural constraints. c) optimal solution with similarity constraint (LIM=100%). d) optimal design with similarity constraint (LIM=100%) and strong user feedback (35% weight reduction).

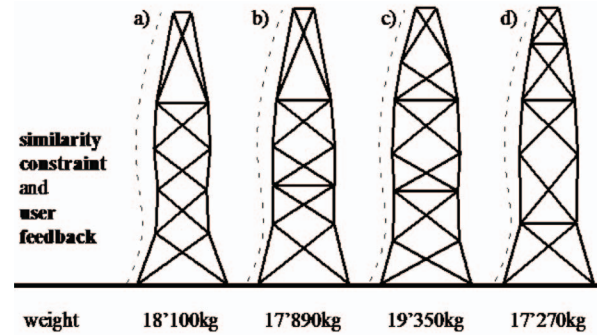


Figure 6. Variations obtained with user interaction for the same settings as in Fig 5d) (similarity constraint + user feedback)

Typically, solutions converged within 50-100 iterations. The tower in Figure 5b) has only structural objectives, as its shape optimally withstands the wind loading for the formal requirements. Obviously this corresponds to the lightest solution but is far from the user preference. In Figure 5c), the optimal solution using the initial design is shown. It can be seen that the algorithm uses the bracing positions to create a shape similar to the initial one. This effectively demonstrates the use of the similarity constraint (15). We can see that this solution already yields an improved weight with respect to the reference. Finally, Figure 5d) shows the optimal solution with user interaction, which led the algorithm to converge onto a solution which is slightly different (less curvature) than the initial shape. This demonstrates that the user was able to drive the optimization towards his goals, changing the constraint dynamically during the process. In addition, the weights for each tower are quite close to each other, confirming the possibility for effective structural optimization that approaches a closer reconciliation with the architect's preference than does pure optimization without interaction.

In Figure 6, further solutions incorporating user interaction are shown (the same settings are kept as in Fig.5d). As can be seen, a multitude of feasible solutions can be generated within similar ranges for shape and weight reflecting alternate

bracing/constructability configurations. This indicates the practical existence of many feasible near optimal solutions, which can be visualized with the interface.

## V. CONCLUSIONS & OUTLOOK

In this paper, an interactive optimization framework for the structural design of truss structures has been described. In addition, the potential for using NURBS curves in truss optimization has been demonstrated. The use of NURBS provides a flexible design tool, easily extended to more challenging, free-form geometry problems. The optimization procedure is carried out using the particle swarm method. The key elements for a successful interactive design have been identified and implemented in a prototype system. The proposed framework has been applied in the design of a truss tower, successfully demonstrating the potential of interactive optimization using NURBS curves.

Possible future directions of research include the parallelization of the process to reduce computation times, machine learning techniques to reduce fatigue of the user by detecting patterns in his choices [19],[43], and dynamic loading considerations. The extension of the user design to topology and sizing, as well as to other than truss-type structures, such as frames or continuous designs, is part of the current research and will be facilitated by the choice of ANSYS as the structural analysis tool. Also, the integration with standard architectural design software is necessary in order to successfully apply the proposed interactive framework to real designs.

## REFERENCES

- [1] Christensen, P.W., and Klarbring, A., "An Introduction to Structural Optimization." Springer, 2009.
- [2] Bendsoe, M.P. and Sigmund, O., "Topology Optimization: Theory, Methods and Applications." Springer, 2003.
- [3] Rajan, S., "Sizing, Shape, and Topology Design Optimization of Trusses Using Genetic Algorithm," *J. of Structural Engineering*, vol.121, no. 10, pp. 1480-1487, 1995.
- [4] Xie, Y.M. and Steven, G.P., *Evolutionary Structural Optimization*. Springer-Verlag, London, June, 1997.
- [5] Hassan, R., Cohanin, B. and de Weck O., "A Comparison of Particle Swarm Optimization and the Genetic Algorithm," in *Proc. 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Austin, TX, USA, pp. 1-13, 2005.
- [6] Perez, R.E. and Behdinan, K., "Particle swarm approach for structural design optimization," *Computers & Structures*, vol. 85, no. 19–20, pp. 1579-1588, 2007.
- [7] Shea, K. and Cagan, J., "Languages and semantics of grammatical discrete structures," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 13, pp. 241-251.
- [8] Schein, M. and Tessmann, O., "Structural Analysis as Driver in Surface-Based Design Approaches," *Int'l. J. of Architectural Computing*, vol. 6, no. 1, pp. 19-39, 2008.
- [9] Pugnale, A. and Sassone, M., "Morphogenesis and Structural Optimization of Shell Structures with the Aid of a Genetic Algorithm," *J. of Int'l. Association for Shell and Spatial Structures*, vol. 48, no. 155, pp. 161-166, 2007.
- [10] Darke, J., "The primary generator and the design process," *Design Studies*, vol. 1, no. 1, pp 36-44, 1979.
- [11] Piegl, L. and Tiller, W., *The NURBS Book* (2nd Ed.). Springer-Verlag New York, Inc., New York, NY, USA, 1997.
- [12] Shi, Y. and Eberhart, R.C., "Parameter selection in particle swarm optimization," in *Proceedings of Evolutionary Programming VII (EP98)*. pp. 591–600, 1998.
- [13] Eberhart, R.C. and Shi, Y., "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proceedings of the Congress on Evolutionary Computation*, pp. 84–88, 2000.
- [14] Carlisle, A. and Dozier, G., "An Off-the-Shelf PSO," in *Proceedings of the Particle Swarm Optimization Workshop*, pp. 1–6, 2001.
- [15] Takagi, H., "Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation," in *Proceedings of the IEEE*, vol.89, no.9, pp.1275-1296, 2001.
- [16] Takagi, H. and Ohsaki, M., "Interactive Evolutionary Computation-Based Hearing Aid Fitting," *IEEE Trans. on Evolutionary Computation*, vol.11, no.3, pp.414-427, 2007.
- [17] Kim, H.-S. and Cho, S.-B., "Application of interactive genetic algorithm to fashion design," *Engineering Applications of Artificial Intelligence*, Vol. 13, No. 6, pp. 635-644, 2000.
- [18] Machwe, A. and Parmee, I.C., "Integrating aesthetic criteria with evolutionary processes in complex, free-form design - an initial investigation," in *IEEE Congress on Evolutionary Computation*, Vancouver, Canada, pp.165-172, 2006.
- [19] Machwe, A. and Parmee, I.C., "Towards an Interactive, Generative Design System: Integrating a 'Build and Evolve' Approach with Machine Learning for Complex Freeform Design," *Lecture Notes in Computer Science*, vol. 4448, pp.449-458, 2007.
- [20] Madar, J., Abonyi, J. and Szeifert, F., "Interactive particle swarm optimization," in *Proc. Int'l. Conf. on Intelligent Systems Design and Applications (ISDA)*, pp. 314- 319, 2005.
- [21] Kurt, B., Etaner-Uyar, A.S., Akbal, T., Demir, N. Kanlikilicir, A.E., Kus, M.C. and Ulu, F.H., "Active Appearance Model-Based Facial Composite Generation with Interactive Nature-Inspired Heuristics," *Lecture Notes in Computer Science*, vol. 4105, pp.183-190, 2006.
- [22] Xiaohui H. and Eberhart, R., "Human vs. swarm: An NK landscape game," in *IEEE Swarm Intelligence Symposium (SIS)*, IEEE, pp.1-5, 2008.
- [23] Kennedy, J. and Eberhart, R., "Particle swarm optimization," in *Proc. IEEE Int'l Conf on Neural Networks*, vol.4, pp.1942-1948, 1995.
- [24] Fourie, P.C. and Groenwold, A.A., "The particle swarm optimization algorithm in size and shape optimization," *Structural and Multidisciplinary Optimization*, vol. 23, no. 4, pp. 259-267, 2002.
- [25] Poli, R., "Analysis of the publications on the applications of particle swarm optimisation," *J. Artificial Evolution and Applications*, vol. 2008, Article ID 685175, pp.1-10.
- [26] [online] <http://www.mathworks.com/matlabcentral/fileexchange/7506>, retrieved July, 2012.
- [27] [online] <http://www.mathworks.com/matlabcentral/fileexchange/26390-nurbs-toolbox-by-d-m-spink>, retrieved July 2012.
- [28] Bel Hadj Ali, N., Rhode-Barbarigos, L, Pascual Albi, A.A. and Smith, I.F.C., "Design optimization and dynamic analysis of a tensegrity-based footbridge," *Engineering Structures*, vol. 32, no. 11, pp. 3650-59, 2010.
- [29] E. N. Chatzi, V. K. Koumoussis, "Optimal Inelastic Design of Multistorey Reinforced Concrete Buildings Towards Uniform Distribution of Minimal Damage," *International Conference on Structural Engineering Dynamics*, Ericeira, Portugal, 22-24 June 2009.
- [30] Coello Coello, C.A., Rudnick, M. and Christiansen, A.D., "Using genetic algorithms for optimal design of trusses," in *Proc. of the Sixth Int'l Conf. on Tools with Artificial Intelligence*, pp.88-94, 6-9 Nov. 1994.
- [31] Adeli, H. and Cheng, N., "Augmented Lagrangian Genetic Algorithm for Structural Optimization," *J. Aerosp. Eng.*, vol. 7, no. 1, 104–118, 1994.
- [32] Sarma, K. and Adeli, H., "Fuzzy Genetic Algorithm for Optimization of Steel Structures." *J. Struct. Eng.*, vol. 126, no. 5, pp. 596–604, 2000.
- [33] Paya, I. Yepes, V., González-Vidosa, F. and Hospitaler, A., "Multiobjective Optimization of Concrete Frames by Simulated Annealing," *Computer-Aided Civil and Infrastructure Engineering*, vol. 23, no. 8, pp. 596-610, 2008.
- [34] Kaveh, A. and Shojaei, S., "Optimal Design of Scissor-Link Foldable Structures Using Ant Colony Optimization Algorithm," *Computer-Aided Civil and Infrastructure Engineering*, vol. 22, no. 1, pp. 56-64, 2007.
- [35] Adeli, Hojjat and Park, Hyo Seon, "Optimization of space structures by neural dynamics," *Neural Networks*, vol. 8, no. 5, pp- 769-781, 1995.
- [36] P.Y. Shim, and S. Manoocheer, "Generating optimal configurations in structural design using simulated annealing," *Int. J. Numer. Methods Engng.*, vol.40, pp. 1053-1069, 1997.
- [37] Manolis Papadrakakis and Nikos D Lagaros, "Reliability-based structural optimization using neural networks and Monte Carlo simulation," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 32, pp. 3491-3507, 7 June 2002.
- [38] Frangopol, D.M., Moses, F., "Reliability-Based Structural Optimization," Chapter 13 in *Advances in Design Optimization* (Edited by H. Adeli), Chapman & Hall, London, 1994, 492-570.
- [39] Luh, Guan-Chun and Lin, Chun-Yi, "A Binary Particle Swarm Optimization for Structural Topology Optimization," in *Third International Joint Conference on Computational Science and Optimization (CSO)*, pp.395-399, 28-31 May 2010.
- [40] L.J. Li, Z.B. Huang, and F. Liu, "A heuristic particle swarm optimization method for truss structures with discrete variables," *Comput. Struct.*, vol.87, pp. 435-443, 2009.
- [41] Candy, L. and Edmonds, E.A., "Supporting the creative user: a criteria-based approach to interaction design," *Design Studies*, vol. 18, No. 2, pp. 185-194, 1997.
- [42] Camp, C.V., Meyer, B.J., and Palazolo, P.J., "Particle Swarm Optimization for The Design of Trusses." *Proc. of the 2004 Structures Congress. Building on the Past: Securing the Future*. 2004.
- [43] Bailey, B. and Raich, A., "Modeling of User Design Preferences in Multiobjective Optimization of Roof Trusses." *J. Comput. Civ. Eng.*, Vol 26, No. 5, pp. 584–596, 2012
- [44] Richardson J., Adriaenssens S., Bouillard P., Coelho R "Multi-objective topology optimisation of truss structures with kinematic stability repair." In: *Structural and Multidisciplinary Optimisation*, Vol.46, No.4, pp.513-532, 2012