Multimodal Truss Structure Design Using Bilevel and Niching Based Evolutionary Algorithms

Md. Jakirul Islam
School of Science (CSIT)
RMIT University
Melbourne, Australia
md.jakirul.islam@rmit.edu.au

Xiaodong Li
School of Science (CSIT)
RMIT University
Melbourne, Australia
xiaodong.li@rmit.edu.au

Kalyanmoy Deb
Computer Science and Engineering,
Michigan State University
East Lansing, MI 48824, USA
kdeb@egr.msu.edu

ABSTRACT

Finding an optimal design for a truss structure involves optimizing its topology, size, and shape. A truss design problem is usually multimodal, meaning that the problem offers multiple optimal designs in terms of topology and/or size of the members, but they are evaluated to have similar or equally good objective function values. From a practical standpoint, it is desirable to find as many alternative designs as possible, rather than finding a single design, as often practiced. A few metaheuristics based methods with niching techniques have been used for finding multiple topologies for the truss design problem, but these studies have ignored any emphasis in finding multiple solutions in terms of size. To overcome this issue, this paper proposes to formulate the truss problem as a bilevel optimization problem, where stable topologies can be found in the upper level and the optimized sizes of the members of these topologies can be found in the lower level. As a result, a new bilevel niching method is proposed to find multiple optimal solutions for topology level as well as for the size level simultaneously. The proposed method is shown to be superior over the state-of-the-art methods on several benchmark truss-structure design problems.

CCS CONCEPTS

•Mathematics of computing → Combinatorial optimization;
•Applied computing → Industry and manufacturing;

KEYWORDS

Bilevel truss problem, PSO, Binary PSO, Speciation-based PSO

ACM Reference format:
Md. Jakirul Islam, Xiaodong Li, and Kalyanmoy Deb. 2017. Multimodal Truss Structure Design Using Bilevel and Niching Based Evolutionary Algorithms. In *Proceedings of GECCO '17, Berlin, Germany, July 15-19, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3071178.3071251

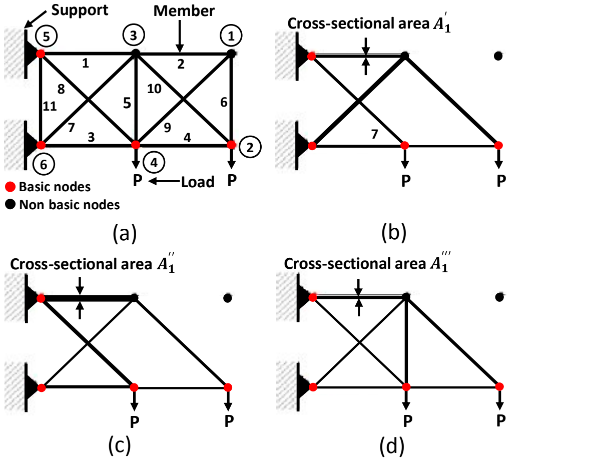
Support
Member
5
3
1
1
2
8
10
11
5
6
7
3
9
4
2
6
4
Basic nodes
Non basic nodes
P
Load
P
(a)
Cross-sectional area $A_1^{'}$
7
P
P
(b)
Cross-sectional area $A_1^{''}$
P
P
(c)
Cross-sectional area $A_1^{'''}$
P
P
(d)

Figure 1: Illustration of (a) 11-member, 6-node ground structure and (b), (c), and (d) its three different design solutions.

1 INTRODUCTION

The optimal design of truss structures is an important research topic in structural optimization. A truss structure comprises of the number of distinct members that are connected by means of pin joints called nodes. In truss, supports are used at both ends by using the hinged joints or rollers for transferring the structural loads to the ground, as shown in Fig. 1. Three types of optimization in truss design can be identified: *size, shape, and topology* [4]. In size optimization, the cross-sections (A) of truss members are optimized by considering the coordinates of nodes and connectivity among various members of the truss to be fixed. Shape optimization optimizes the nodal coordinates (ξ) of existing nodes, whereas the topology optimization deals with the selection of nodes and their connectivity. The goal of these optimization tasks is to find an optimal (i.e., light-weight) truss structure under specified conditions.

Basically, truss problems are governed by the user-defined objectives and constraints such as stress and displacement [14]. These constraints often conflict with the objective function and thus finding an optimal solution for such a problem is a challenging task. In addition, both the objective function and constraints (which are often nonlinear) cause the search space of a truss problem to become highly multimodal. There is a good chance that multiple optimal solutions not only may exist at the topology level (see Fig. 1(b) and Fig. 1(d)), but also may exist in terms of size for each fixed topology (see Fig. 1(b) and Fig. 1(c)). From a practical viewpoint it is often desirable for the decision maker to choose from among multiple equally good solutions in terms of both topologies and sizes.

GECCO '17, Berlin, Germany
© 2017 ACM. 978-1-4503-4920-8/17/07...$15.00
DOI: http://dx.doi.org/10.1145/3071178.3071251

In the past decade, several meta-heuristic algorithms have been proposed for the truss optimization [4, 9, 21, 23]. Generally speaking, two approaches can be found in literature for handling truss design problems, either a one-stage approach [4, 21] or a two-stage approach [8, 19, 20]. In the two-stage approach, topology, size, and shape variables are assumed to be linearly separable. Given this assumption, these methods find different topologies in the first stage by considering an equal cross-sectional area for each member of a truss structure. In the second stage, the size (cross-sectional areas) and shape variables (nodal coordinates) of the found topologies are optimized. It is obvious that with this approach, the optimal designs may not be attainable since these variables are often not linearly separable in practice [4]. On the other hand, the single-stage approach optimizes all these three types of variables together simultaneously to find out the stable topologies for a given truss problem, where each topology has the different size solutions [4, 6, 21?]. However, the way they formulated the truss problem is not suitable in terms of finding multiple optimal solutions at both the topology and size levels simultaneously in one optimization run.

In this paper, we propose a bilevel formulation for the truss problem. The proposed formulation treats the topology optimization as the upper level optimization task and the size and shape optimization as the lower level optimization task. The goal is to obtain multiple truss designs by considering both its topology and size simultaneously using a new bilevel niching method. The contributions are as follows:

- The truss optimization problem is formulated as a bilevel optimization problem.
- A bilevel niching method is developed to find multiple optimal solutions for the bilevel truss optimization problem.
- The numerical studies are carried out using three well-known benchmark truss problems. Our results demonstrate that the proposed bilevel niching method is able to find multiple optimal truss structures, some of which equal or outperform the best solutions found in literature.

The rest of the paper is organized as follows: Section 2 presents the related work. Section 3 describes the bilevel formulation of the truss optimization problem. Section 4 describes the proposed niching method. This is followed by experimental studies in Section 5. Our concluding remarks are provided in Section 6.

2 RELATED WORK

In the last decades, several techniques based on classical optimization methods have been developed for the optimal design of truss structures [13]. Recently, meta-heuristic algorithms have also been adopted because of their appealing properties, e.g., they do not make certain mathematical assumptions hence possess better global search abilities than the traditional exact methods. For example, Rajeev and Krishnamoorthy [26] proposed a discrete genetic algorithm for the truss optimization problems, where a penalty-based method is used to transform a constrained truss problem into an unconstrained problem. Deb and Gulati [4] proposed a real-coded genetic algorithm to efficiently handle the truss optimization problems, where GA operators are directly applied to real-value coded variables instead of binary strings. Furthermore, mixed encoding schemes including both binary, floats, and integers are also employed to code the variables of the truss problems in a GA [27]. This encoding scheme, so called surrogated reproduction is adopted for the creation of offspring from the parent solutions. Like GA, many other meta-heuristic algorithms including harmony search (HS) [15] and particle swarm optimization (PSO) [23] have been proposed for the truss optimization problems. The optimization results provided by HS and PSO showed that they can provide better optimized trusses as compared to those using the GA based methods. It can be observed that despite having multiple optimal solutions, all the above methods were designed for obtaining a single optimal or near optimal solution for the truss optimization problems in one optimization run.

Niching methods are well-suited optimization methods for finding multiple optimal solutions in a single optimization run [7]. Classic niching methods include fitness sharing [7], clearing [25], speciation [16], and SPSO [24]. Niching methods were first adopted in [19] for truss optimization, where an ant colony algorithm is combined with fitness sharing with an aim to obtain multiple topologies for the truss design problems. They also incorporate the fitness sharing concept into a modified binary PSO (MBPSO) for multimodal optimization [20]. In both these methods [19, 20], the truss problem is formulated as a multi-stage optimization problem where the topology, size, and shape variables are assumed to be linearly separable. In such a multi-stage approach, different topologies are obtained in the first stage by assuming an equal cross-section area for each member of a ground structure. In the later stage, the area of each member of these obtained topologies is optimized to realize the optimal design of a truss problem. It is apparent that the true optimal/near-optimal designs of a truss structure may not be achievable by such a method, since these three types of variables are actually not linearly separable [4]. Furthermore, the truss problem formulated in a single-stage [19, 20] or bilevel [6?] are not capable of locating multiple solutions for all variables at different levels, i.e., the topology level, size level, and/or shape level simultaneously. This motivates us to propose a new bilevel formulation for the truss problem, which is described in the subsequent section.

3 PROBLEM FORMULATION

3.1 A general bilevel problem

In bilevel optimization [2], two different levels of optimization take place, with one level (i.e., lower level) of optimization being nested within the other (i.e., upper level). For a bilevel problem, if the upper level optimizer wants to optimize its objective, then it needs to obtain the optimal response of the lower level optimizer. For the upper level objective function F and lower level objective function f, the bilevel optimization problem can be expressed as:

$$\begin{aligned} \min_{\vec{x_u}\in X_u, \vec{x_l}\in X_l} \quad & F(\vec{x_u},\vec{x_l}) \\ s.t. \quad & \vec{x_l} \in \operatorname*{argmin}_{\vec{x_l}\in X_l}\{f(\vec{x_u},\vec{x_l}) : g_j(\vec{x_u},\vec{x_l}) \leq 0, j = 1 \ldots J\} \\ & G_k(\vec{x_u},\vec{x_l}) \leq 0, k = 1 \cdots K, \end{aligned} \tag{1}$$

where G_k represents the upper level constraints and g_j represents the lower level constraints, respectively. In this formulation, the upper level objective function evaluates the performance of the lower level objective function through $f(\vec{x_u},\vec{x_l})$, which is obtained by solving the lower level variable $\vec{x_l}$ for fixed $\vec{x_u}$. Based on this idea, the truss problem can be formulated as a bilevel optimization problem, as described in the following section.

3.2 Truss as a bilevel problem

We begin the bilevel formulation for a truss problem by considering its ground structure, which is a complete truss with all possible member (m) connections among all nodes (n) in the structure. Fig. 1(a) shows the ground structure of 11-member and 6-node truss, where the nodes with load and support are called *basic nodes* and remaining nodes are called *non-basic nodes*. For the bilevel formulation, the topology variables of such a ground structure are treated as the variables of the upper level problem, and at the same time, size and shape variables are treated as the variables of the lower level problem. Considering this, the objective functions of the upper level problem and lower level problem are provided below.

Upper level: For a given set of members M and nodes N of a ground structure, the upper level optimization task is to select a subset of members to find a stable topology subject to the given constraints. In this case, we consider $\vec{x} \in M \times N$ as a found topology of the ground structure consisting of m' members and n' nodes. Let $\vec{y} \in A' \times \xi'$ be the variable of the lower level problem, where A' represents the set of member cross-sectional areas and ξ' represents the set of nodal coordinates of topology $\vec{x}$. Considering the upper level variable $\vec{x}$ and lower level variable $\vec{y}$, the objective function of the upper level problem is formulated as:

$$\begin{aligned} &\text{Find} && \vec{x} \\ &\text{to minimize} && W(\vec{x},\vec{y}) \\ &\text{subject to} && G_1 : \textit{Truss is acceptable to the user}, \\ & && G_2 : \textit{Truss is internally and externally stable}, \\ & && G_3 : \vec{y} \in \underset{\vec{y}\in A'\times\xi'}{\operatorname{argmin}}\{w(\vec{x},\vec{y}) : g_j(\vec{x},\vec{y}) \le 0, j = 1 \ldots J\}. \end{aligned} \tag{2}$$

In Eq. (2), the function W represents the weight of the topology $\vec{x}$, constraint G_1 ensures that the truss consists of all the basic nodes, and constraint G_2 ensures the internal and external stability of the truss. The internal and external stability of a truss can be checked by the following equations:

$$m' \ge 2n' - 3, \tag{3}$$

and

$$m' + r \ge 2n', \tag{4}$$

where m' is the number of members, r denotes the number of reaction components [14], and n' denotes the number of nodes in truss. Finally, constraint G_3 associates with the lower level problem which is described below.

Lower level: In a bilevel truss problem, the lower level task is associated with the size and shape optimization of a topology provided by the upper level, subject to the given constraints. The goal of the lower level optimization is to minimize the weight of a topology $\vec{x}$ (the upper level problem) by choosing the optimal value for its member cross-sectional areas and nodal coordinates. To achieve this goal, the lower level problem is expressed as a nonlinear programming problem (NLP) in the following way:

$$\begin{aligned} \min_{\vec{y}\in A'\times\xi'} \quad & w(\vec{y}) = \sum_{i=1}^{m'} \rho'_i \ell'_i(\xi^s_i, \xi^e_i) A'_i \\ \text{subject to} \quad & g_1 : \textit{Truss is kinematically stable} \\ & g_2 : S_i \ge \sigma_i(A',\xi'),\ i = 1,2,\ldots,m', \\ & g_3 : \delta^{max}_j \ge \delta_j(A',\xi'),\ j = 1,2,\ldots,n', \\ & g_4 : A'_{min} \le A'_i \le A'_{max},\ i = 1,2,\ldots,m', \\ & g_5 : \xi'_{min} \le \xi'_j \le \xi'_{max},\ j = 1,2,\ldots,n', \end{aligned} \tag{5}$$

where m' and n' represent the number of members and nodes of the topology $\vec{x}$, respectively. The parameters ρ'_i and ℓ'_i are the material density and length of the i-th member, respectively. Here, the length ℓ'_i depends on the start coordinate ξ^s_i and end coordinate ξ^e_i of the i-th member of topology $\vec{x}$. The parameter A'_i represents the cross-sectional area of the i-th member. In this equation, constraint g_1 ensures that the kinematical stability of the truss is checked by determining the positive definiteness of the stiffness matrix. Constraint g_2 ensures that the stress σ_i of a member is less than or equal to the allowable stress S_i. Constraint g_3 ensures that the displacement δ_j of j-th node is less than or equal to the allowable displacement δ^{max}_j. Constraint g_4 ensures that the value of A'_i is within the limits $[A'_{min}, A'_{max}]$. Finally, constraint g_5 is used to ensure that the coordinates of the j-th node ξ'_j is within the limits $[\xi'_{min},\xi'_{max}]$.

4 NICHING FOR BILEVEL TRUSS PROBLEMS

The bilevel truss problem proposed in the above section can be used to obtain multiple designs in terms of topology as well as the size of the truss problem. Although it is possible to apply niching at both the upper and lower levels, applying niching to the upper level alone seems to be sufficient in providing the necessary diversity in topology and size spaces. Hence, there is no need to apply niching again to the lower level. Furthermore, by applying niching only to the upper level, for each given topology defined from the upper level, its corresponding search space is only optimized by a standard optimizer, which will reduce the search space substantially. Fig. 2 shows an example where niching is applied only to the upper level, but only a standard optimizer is used at the lower level.

Following the above, the proposed method uses a binary niching method as the upper level optimizer based on a well-known niching method, speciation-based PSO (SPSO) [24]. Here, the idea of SPSO is implemented using a modified BPSO (MBPSO) which is referred to as the binary SPSO (B-SPSO). For the lower level problem, the proposed method uses a standard PSO. For clarity, a brief description of PSO and MBPSO is presented first in the following sections, before introducing the working steps of the proposed niching method.

4.1 Particle swarm optimization (PSO)

Particle swarm optimization (PSO) is originally proposed in [11] for optimizing the continuous problems. In PSO, particles (i.e., individuals of a swarm) are first initialized by placing them randomly in the d-dimensional search space. In the subsequent iterative process, each particle (or the i-th particle in the swarm) knows about two pieces of information: its best position visited so far, i.e., its personal best position $\mathbf{p}_i$=[$p_{i1}, p_{i2}, \ldots, p_{id}$] and the global best position of all particles in the swarm $\mathbf{p}_g$=[$p_{g1}, p_{g2}, \ldots, p_{gd}$]. At each iteration, the velocity $\mathbf{v}_i$=[$v_{i1}, v_{i2}, \ldots, v_{id}$] of i-th particle of the swarm is updated according to the following equation:

$$\mathbf{v}_i^{k+1} = w\mathbf{v}_i^k + c_1 r_1^k(\mathbf{p}_i^k - \mathbf{x}_i^k) + c_2 r_2^k(\mathbf{p}_g^k - \mathbf{x}_i^k), \tag{6}$$

where k represents the current iteration, w the inertia weight parameter for maintaining a good balance between exploration and exploitation, and d denoting d-th dimension. r_1 and r_2 are two random numbers drawn from $\mathcal{U}(0,1)$ for each dimension, and $c_1 > 0$ and $c_2 > 0$ are the cognitive and social coefficients, respectively.

In practice, once $\mathbf{v}_i^{k+1}$ updated, a limit v_{max} is placed on it to ensure that the velocities do not exceed this limit. After that, the position $\mathbf{x}_i$ of i-th particle is updated according to the following equation:

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \mathbf{v}_i^{k+1} \tag{7}$$

4.2 Modified binary PSO (MBPSO)

To tackle discrete/combinatorial optimization problems, Kennedy and Eberhart developed the first binary version of PSO (BPSO) [12]. Like the continuous PSO, the original binary PSO (BPSO) defines the flight of particles through their velocity and position updates in the binary search space, in order to find the best solution. BPSO uses Eq. (6) for updating the velocity $\mathbf{v}_i^{k+1}$ of the i-th particle, and the following equation for updating the position $\mathbf{x}_i^{k+1}$:

$$\mathbf{x}_i^{k+1} = \begin{cases} 0 & \text{if } rand() > S_T(\mathbf{v}_i^{k+1}), \\ 1 & \text{otherwise.} \end{cases} \tag{8}$$

where $rand()$ is a random number drawn from $\mathcal{U}(0,1)$ and $S_T(\mathbf{v}_i^{k+1})$ denotes a sigmoid transfer function, which is determined by the following:

$$S_T(\mathbf{v}_i^{k+1}) = \frac{1}{1 + e^{-\mathbf{v}_i^{k+1}}}. \tag{9}$$

Basically, the sigmoid transfer function $S_T(\mathbf{v}_i^{k+1})$ (see Fig. 3) provides the probabilities (in the range of [0,1]) of the bits in $\mathbf{x}_i^{k+1}$ taking the value of 0 or 1.

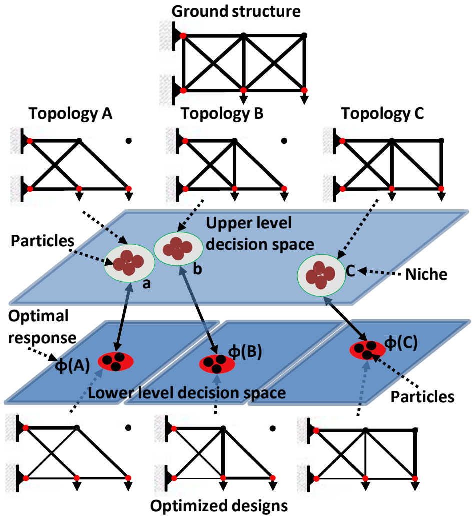

Figure 2: Illustration of an example where niching is applied in the upper level and a standard optimizer is used at the lower level to optimize a bilevel truss problem.

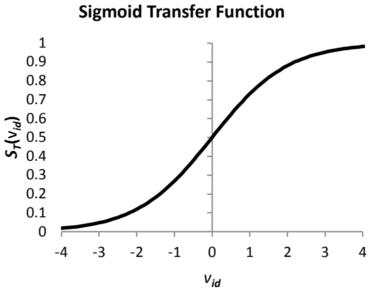

Figure 3: Illustration of the sigmoid transfer function (S_T).

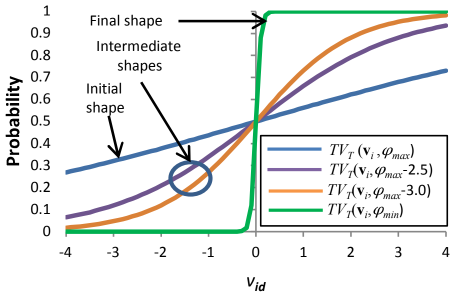

Figure 4: Illustration of the time-varying sigmoid transfer function (TV_T).

It is reported that BPSO has difficulties in providing a good balance between exploration and exploitation [1, 22]. It is shown in [18] that for $\mathbf{p}_i^k = \mathbf{p}_g^k$, BPSO provides a smaller $\mathbf{v}_i^{k+1} \simeq 0$ in the first few iterations, after that $\mathbf{v}_i^{k+1}$ reaches to the maximum limit ($\pm v_{max}$) and remain constant for the rest of the iterations. With the higher velocity, the sigmoid transfer function provides very low bit flipping probability (see Fig. 3). In this case, the sigmoid transfer function promotes a higher level of exploitation than exploration. Thus BPSO cannot maintain a balance between exploration and exploitation. This study suggests a remedy to this by introducing a time-varying parameter φ into the sigmoid transfer function, creating a new time-varying transfer function (TV_T), as illustrated in Fig. 4. The curves in Fig. 4 are produced by this time-varying sigmoid transfer function TV_T as defined below:

$$TV_T(\mathbf{v}_i^{k+1}, \varphi) = \frac{1}{1 + e^{-\frac{\mathbf{v}_i^{k+1}}{\varphi}}}, \tag{10}$$

where $\mathbf{v}_i^{k+1}$ is the velocity of the i-th particle at (k+1)-th iteration; φ is a time starting with a large value (φ=4.0) and gradually decreased to (φ=0.5) as the run progresses. With TV_T, the new modified BPSO (MBPSO) updates $\mathbf{v}_i$ and $\mathbf{x}_i$ according to the following equations:

$$\mathbf{v}_i^{k+1} = \mathbf{v}_i^k + c_1 r_1^k(\mathbf{p}_i^k - \mathbf{x}_i^k) + c_2 r_2^k(\mathbf{p}_{g,i}^k - \mathbf{x}_i^k), \tag{11}$$

and

$$\mathbf{x}_i^{k+1} = \begin{cases} 0 & \text{if } rand() > TV_T(\mathbf{v}_i^{k+1}, \varphi), \\ 1 & \text{otherwise,} \end{cases} \tag{12}$$

respectively, where $\mathbf{p}_{g,i}$ denotes the local best of the i-th particle, and other symbols have their usual meaning as in Eq. (6).

MBPSO provides a better balance between the exploration and exploitation in the following way: for each given velocity value, the curve $TV_T(\mathbf{v}_i, \varphi_{max})$ provides the highest amount of bit flipping probability (i.e., exploration), because the curve is the closest to the probability value of 0.5 than any other curves. On the other hand, $TV_T(\mathbf{v}_i, \varphi_{min})$ provides the lowest amount of bit flipping probability for changing the position $\mathbf{x}_i$ of i-th particle (i.e., exploitation). Based on this observation, we propose MBPSO to adopt curves $TV_T(\mathbf{v}_i, \varphi_{max})$ at the start of a run in order to provide a stronger exploration; $TV_T(\mathbf{v}_i, \varphi_{max}\text{-}2.5)$ to $TV_T(\mathbf{v}_i, \varphi_{max}\text{-}3.0)$ in the intermediate stage of the run to provide a moderate level of exploration; and towards the final stage of the run $TV_T(\mathbf{v}_i, \varphi_{min}+0.3)$ to $TV_T(\mathbf{v}_i, \varphi_{min})$ to have a stronger exploitation.

4.3 The proposed niching method

The proposed niching algorithm is described in detail here. In this algorithm, *U/u* and *L/l* denote the parameters of the upper level optimizer (B-SPSO) and the lower level optimizer (PSO), respectively. The working steps of the proposed niching method are described as follows.

Step U1*-Generate an initial swarm*: For a truss ground structure with m members and n nodes, a population of N_u particles is initialized randomly in the m-dimensional search space. The velocity of i-th particle is denoted by $\mathbf{v}_i^u$ which is drawn from $\mathcal{U}(\text{-}v_{max},v_{max})$. The position of i-th particle is denoted by $\mathbf{x}_i^u$. Each element of $\mathbf{x}_i^u$ holds a binary number 1 or 0 that represents the presence (or absence) of a member of a ground structure. For example, consider a 6-node and 11-member ground structure (see Fig. 1), the position $\mathbf{x}_i^u$=[10110011010] represents a topological solution of this structure where the elements 1, 3, 4, 7, 8, and 10 are present and the element 2, 5, 6, 9, and 11 are absent from the structure. Each initialized particle represents an initial truss structure based on the random bit information contained in this position vector $\mathbf{x}_i^u$.

Step U2*-Evaluate* $\mathbf{x}_i^u$: If the truss structure corresponding to $\mathbf{x}_i^u$ is not feasible to the constraints G_1 and G_2, then a large penalty is assigned to $\mathbf{x}_i^u$ to indicate that this solution is not a good solution and B-SPSO will follow the next step ***Step U3***. Otherwise, the truss structure corresponding to $\mathbf{x}_i^u$ will be sent to the lower level optimizer for the size optimization. In this case, the standard PSO is used (see section 4.1) whose working steps in terms of the truss size optimization are described below.

Step L1*-Generate an initial swarm*: For a given truss topology with n' nodes and A' cross-sectional areas of m' members. A population of N_l particles is initialized randomly in m'-dimensional search space. For the j-th particle, the velocity $\mathbf{v}_j^l$ is drawn from $\mathcal{U}(A'_{min},A'_{max})$ and the position $\mathbf{x}_j^l$ is drawn from $\mathcal{U}(0,A'_{max})$, where A'_{min} and A'_{max} are the lower and upper bounds of the cross-sectional areas of the members of the topology obtained by the upper level optimizer.

Step L2*-Update rule*: To evaluate the topology of the j-th particle, the fitness value considered constraint violation from [4] are adopted in this study:

$$f^j(A', \xi') = \begin{cases} 10^7, & \text{if } g_1 \text{ is violated,} \\ C, & \text{otherwise,} \end{cases} \tag{13}$$

where C=$w^j(A', \xi') + 10^5 \sum_{p=1}^{m'} | < g_2^p > | + 10^5 \sum_{q=1}^{n'} | < g_3^q > |$. Here, $f^j(A', \xi')$ and $w^j(A', \xi')$ denote the fitness value and weight of the truss structure corresponding to the j-th particle. The operator $< . >$ is the bracket-operator penalty term [3]. Note that since the standard PSO allows the size variables and shape variables to be bounded within specified limits $[A'_{min},A'_{max}]$ and $[\xi'_{min},\xi'_{max}]$, the constraints g_4 and g_5 are automatically satisfied.

After calculating the fitness value of j-th particle, PSO determines the personal best position $\mathbf{p}_j^l$ of this particle and the best global position $\mathbf{p}_g^l$ for the whole swarm. Subsequently, PSO updates $\mathbf{v}_j^l$ and $\mathbf{x}_j^l$ of j-th particle according to Eq. (6) and Eq. (7), respectively.

Step L3*-Stopping criteria*: The PSO run is terminated once a predefined number of iterations is reached, otherwise it goes back to *Step L2*.

Step L4*-Returning the optimized truss*: The optimized truss that is held by $\mathbf{g}^l$ is returned to the i-th particle of the upper level optimizer for upper level optimization, as follows.

Step U3*-Sort all the particles*: In this case, all the particles of B-SPSO are sorted in an ascending order according to the fitness values of their personal best $\mathbf{p}_i^u$. The B-SPSO stores these sorted particles in a list called P_{sorted}.

Step U4*-Determine the species and their seeds*: In this step, B-SPSO uses P_{sorted} to determine the species based on the niche radius r_s, as described in [24]. Since the upper level optimization is associated with the binary-value, the species can be determined by comparing the Hamming distance between the particles in P_{sorted}. After that we use the same procedure as in [24] for determining the species seeds which are stored in an another list called L_{seed}.

Step U5*:* Assign $\mathbf{p}_{g,i}^u = \mathbf{x}_i^{seed}$, where $\mathbf{x}_i^{seed}$ is the i-th seed of the list L_{seed}.

Step U6*:* Update $\mathbf{v}_i^u$ and $\mathbf{x}_i^u$ according to Eq. (11) and Eq. (12), respectively.

Step U7*-Stopping criteria*: B-SPSO is terminated once a predefined number of iterations is reached, otherwise go to ***Step U2***.

5 NUMERICAL EXAMPLES

Three well-known truss design benchmark problems are considered to demonstrate the effectiveness of the niching algorithm proposed, including the 11-member ground structure [4, 5, 21], 25-member ground structure [4, 10, 17], and two-tier 39-member ground structure [4, 20, 23]. The experimental setup for the proposed niching algorithm is provided in Table 1. In this study, the value for r_s=1, φ_{max}=5, φ_{min}=1, N_u=100, and N_l=10 have been set based on some preliminary study, making sure that these values are robust. Note that in all our experiments we consider only topology and size optimization, leaving out the shape optimization for future studies.

Table 1: Parameters used in B-SPSO (upper level) and PSO (lower level) of the proposed niching method.

Parameters	B-SPSO	PSO
No. of particles (N_u and N_l)	100	10
No. of function evaluations (FEs)	6000	10000
Niche radius (r_s)	1	-
Acceleration constant c_1 and c_2	2.00	2.00
Initial inertia weight (w_{max})	1	0.9
Final inertia weight (w_{min})	1	0.2
Velocity limit [v_{min},v_{max}]	[-6,6]	[A'_{min}, A'_{max}]
Position limit [x_{min},x_{max}]	[0,1]	[0, A'_{max}]
Control parameter limit [φ_{min},φ_{max}]	[1,5]	-

5.1 11-member, 6-node truss

The 11-member, 6-node ground structure is presented in Fig. 1(a). To optimize this truss, the following parameters are used: Length of each member is 360 inches (in), young's modulus is 10^4 ksi, density is 0.1 lb/in^3, allowable stress is 25 ksi, and allowable displacement is 2 inches. The lower (A'_{min}) and upper (A'_{max}) bounds on the member areas are considered to be 0.0 in^2 to 35.0 in^2. The critical member area is set to 0.09 in^2. A vertical load 100,000 lb is considered at node 2 and 4, respectively.

Fig. 5 shows the four optimal designs obtained by the proposed method. It can be seen that all four optimal trusses are the same in terms of the topology, but they are different in terms of the cross-sectional area (size) of the members, as presented in Table 2. According to Table 2, these optimal trusses have the similar weight values, but the cross-sectional area of their members are slightly different from each other. It is evident that the weight (on average 4875.38 lb) of the designs in Fig. 5(a)-5(d) are much lower than the weights (4899.15 lb, 4899.15 lb, and 5109 lb, respectively) provided by Deb and Gulati[4], Miguel et al. [21], and Flager et al. [5], respectively. To the best of our knowledge, the weight of the design solution presented in Fig. 5(a) is the best weight ever found for 11-member, 6-node ground structure.

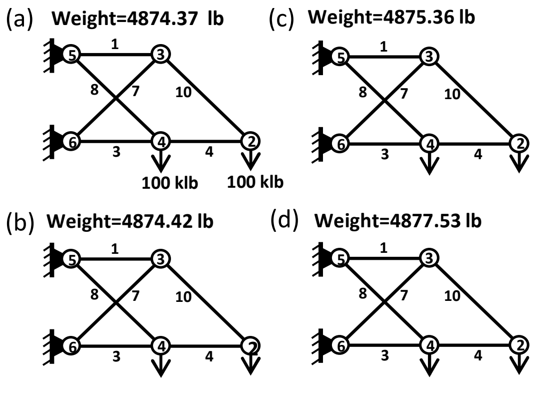

Figure 5: Four optimal solutions obtained by the proposed method, from 11-member, 6-node ground structure.

Table 2: Members areas (in^2) of the optimal trusses for 11-member, 6-node ground structure.

Member	Fig. 5(a)	Fig. 5(b)	Fig. 5(c)	Fig. 5(d)	Deb [4]	Miguel [21]	Flager [5]
A_1	29.5041	29.5387	30.0065	30.3603	29.6800	29.6800	33.7000
A_2	-	-	0.0000	0.0001	-	-	0.1000
A_3	22.303	22.163	21.753	22.1216	22.0700	22.0700	23.6800
A_4	15.0666	14.9324	15.4647	14.5404	15.3000	15.3000	13.3300
A_5	0.0000	0.0000	-	0.0000	-	-	0.1000
A_6	0.0001	-	0.0001	0.0002	-	-	0.1000
A_7	21.1486	21.1028	21.4445	21.958	21.4400	21.4400	19.1800
A_8	6.0352	6.0425	6.0646	6.0447	6.0900	6.0900	7.1920
A_9	0.0001	-	0.0005	-	-	-	0.1000
A_{10}	21.2708	21.4799	20.7167	20.4089	21.2900	21.2900	23.680
Weight (lb)	**4874.37**	**4874.42**	**4875.36**	**4877.53**	4899.15	4899.15	5109

5.2 25-member, 10-node ground structure

For this experiment, we choose a widely-used 3-D truss structure which has an 25-member and 10-node ground structure [4, 10, 17], as presented in Fig. 6. Here, members are grouped considering the symmetry on opposite sides, as done by Deb and Gulati in [4]. Material properties and design parameters are set as follows: Young's modulus and density of the materials are the same as before. The allowable stress and displacement are set to 40 ksi and 0.35 inches, respectively. The lower (A'_{min}) and upper (A'_{max}) bounds on the member areas are set to 0.0 to 3.0 in^2, respectively. The critical member area is set to 0.005 in^2. According to [4, 10, 17], this truss is optimized by applying four forces: (1000; 10000; -5000) on node 1, (0; 10000; -5000) on node 2, and (500; 0; 0) on node 3 and 6, respectively.

Fig. 7 shows four design solutions obtained for the 25-member, 10-node truss problem, which has the same overall weight. Like the previous example, all these four design solutions of 25-member truss structure are the same in terms of the topology, but they are different in terms of the members cross-sectional area. The cross-sectional areas of the active members of these optimal trusses are provided in Table 3. According to Fig. 7, the proposed method is able to provide the same design as Deb and Gulati [4]. However, the weight (524.99 lb) obtained by the proposed method is much lower than the weights (544.85 lb, 545.19 lb, 545.09 lb, respectively) provided by Deb and Gulati [4], Li [17], and Kaveh et al. [10], respectively. Note that,

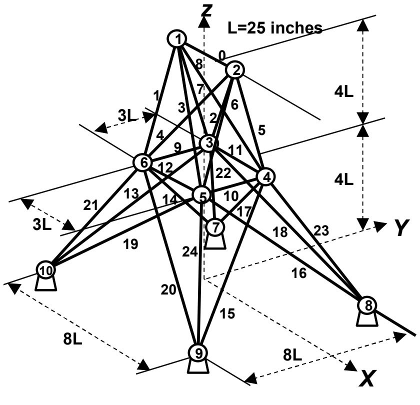

Figure 6: Illustration of 25-member, 10-node ground structure.

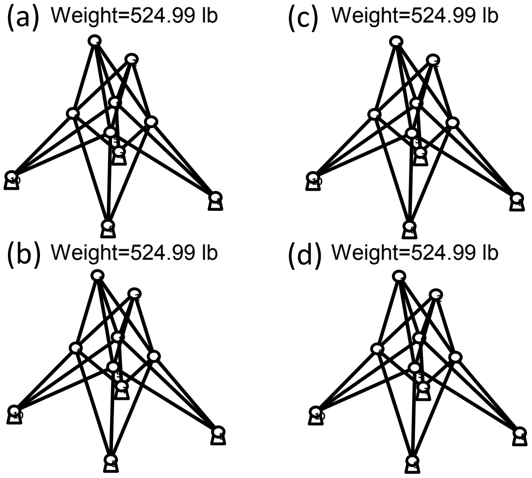

Figure 7: Four optimal solutions obtained by the proposed method, from 25-member, 10-node ground structure.

Table 3: Member areas (in^2) of the optimal trusses of 25-member, 10-node ground structure.

Mem. No.	Fig. 7(a)	Fig. 7(b)	Fig. 7(c)	Fig. 7(d)	Deb [4]	Li [17]	Kaveh [10]
A_0	-	0.0000	-	0.0000	-	0.010	0.010
A_1-A_4	2.0840	2.0787	2.1109	2.1113	2.0370	1.970	1.910
A_5-A_8	2.4451	2.4537	2.4346	2.4382	2.9690	3.016	2.798
A_9-A_{12}	-	-	0.0000	0.0000	-	0.010	0.010
A_{13}-A_{16}	0.8880	0.8917	0.8893	0.8878	0.6990	0.694	0.708
A_{17}-A_{20}	1.1924	1.1870	1.1791	1.1781	1.6440	1.681	1.836
A_{21}-A_{24}	3.0000	3.0000	3.0000	3.0000	2.6580	2.643	2.645
Weight (lb)	**524.99**	**524.99**	**524.99**	**524.99**	544.85	545.19	545.09

the weight obtained by the proposed niching method is the lowest weight ever found for the 25-member and 10-node ground structure.

5.3 Two-tier, 39-member, 12-node ground structure

A two-tier, 39-member, 12-node ground structure is employed here for topology and size optimization, as shown in Fig. 8. For this experiment, the following parameter settings are considered: the material properties and maximum allowable displacement are the same as in 11-member ground structure problem, except the allowable stress is 20 ksi. In addition, according to [4, 20], the lower (A'_{min}) and upper (A'_{max}) bounds of the member areas are set to 0.0 and 2.5 in^2, respectively and a critical member area of 0.05 in^2 is chosen.

Fig. 9 shows the four optimal/near-optimal designs obtained by the proposed niching method. The member areas of these trusses are listed in Table 4. It can be observed that the obtained trusses have almost the similar weight values (191.16 lb, 191.18 lb, 192.80 lb, and 193.42 lb), but either their members' connectivity i.e., topology, or members' cross-sectional areas i.e., sizes, are very different from each other (see Table 4). This shows that for the 39-member truss, the proposed method can provide multiple topology and size solutions simultaneously in a single run. Fig. 10(a-b) illustrates two optimal trusses employed by Luh and Li [20] with overall weights

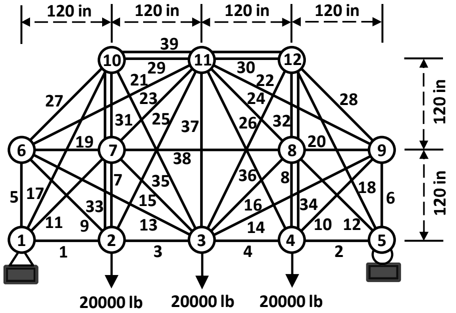

Figure 8: Illustration of a two-tire, 39-member, 12-node ground structure.

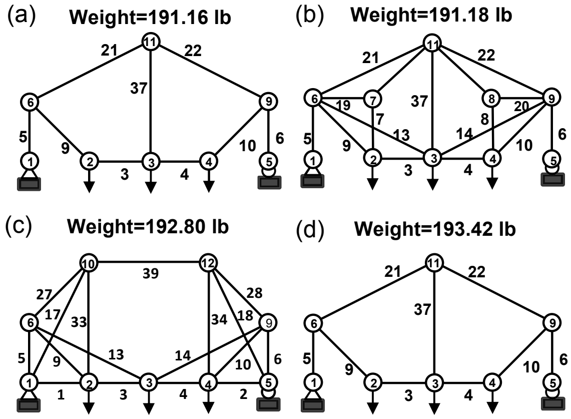

Figure 9: Optimal trusses for 39-member, 12-node ground structure derived employing the proposed niching method.

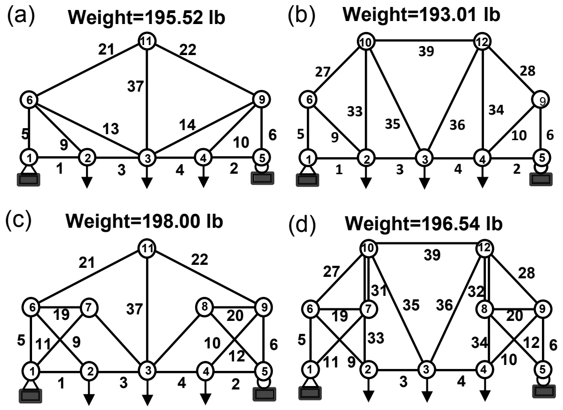

Figure 10: Optimal trusses derived employing (a-b) PSO algorithms by Luh and Lin [20], and (c-d) GA by Deb and Gulati [4].

Table 4: Weights and member areas (i.e., solutions) found by the proposed method and those from literature for 39-member, 12-node truss problem. Note that the solutions provided by the proposed method (Fig. 9(a-c)) are better than those from literature (Fig. 10).

		Member cross-sectional area (in.^{2})																		
Design	Weight	A_1-A_2	A_3-A_4	A_5-A_6	A_7-A_8	A_9-A_{10}	A_{11}-A_{12}	A_{13}-A_{14}	A_{15}-A_{16}	A_{17}-A_{18}	A_{19}-A_{20}	A_{21}-A_{22}	A_{23}-A_{24}	A_{25}-A_{26}	A_{27}-A_{28}	A_{31}-A_{32}	A_{33}-A_{34}	A_{35}-A_{36}	A_{37}	A_{39}
Fig. 9(a)	**191.16 lb**		0.9951	1.4952		1.4072						1.1125	0.9951						0.9951	
Fig. 9(b)	**191.18 lb**		0.9949	1.4926	0.0006	1.407		0.0006			0.0001	1.1127	0.0009	0.0001					0.9949	
Fig. 9(c)	**192.80 lb**	0.017	0.7463	1.4812		1.0504		0.0001		0.0177					1.0474		0.2561	0.5563		0.9951
Fig. 9(d)	193.42 lb		1.0791	1.4928		1.4072						1.1125							0.9952	
Fig. 10(a)	195.52 lb	0.043	1.000	1.500		1.415		0.045				1.119			1.061				1.000	
Fig. 10(b)	193.01 lb	0.039	0.750	1.500		1.061						1.061					0.250	0.560		1.000
Fig. 10(c)	198.00 lb	0.050	1.001	1.501		1.416	0.050		0.050		0.050	1.118							1.002	
Fig. 10(d)	196.54 lb		0.751	1.502		0.051	1.061				0.051				1.063	0.052	0.251	0.559		1.004

of 195.52 lb and 193.01 lb, respectively. In addition, Fig. 10(c-d) illustrates two other optimal trusses obtained by Deb and Gulati [4] with overall weights of 198 lb and 196.54 lb, respectively. It can be observed that the trusses in Fig. 9(a), 9(b), and 9(d) are similar to the trusses in Fig. 10(a) and 10(c). However, the weights (191.16 lb, 191.18 lb, and 193.42 lb, respectively) of the obtained trusses in Fig. 9 are less than the weights (195.52 lb and 198 lb, respectively) of the trusses in Fig. 10. Likewise, the weight (192.80 lb) of the obtained truss in Fig. 9(c) is also less than the weight (193.01 lb and 196.54 lb, respectively) of the trusses in Fig. 10(b) and 10(d).

6 CONCLUSIONS

This paper has presented a bilevel formulation for the truss optimization problem, making it possible to consider multiple optimal solutions in terms of both topology and size. This is usually difficult to achieve with existing formulations of the truss problems. For this bilevel formulation, a binary SPSO niching method has been applied at the upper level while a standard PSO is applied at the lower level. In the binary SPSO, a time-varying transfer function is employed to enhance its search ability. The performance of the proposed method is evaluated over three well-known truss problems. The numerical studies show that the proposed method has the ability to locate multiple optimal solutions (topologies) for the upper level of the bilevel truss problem. In addition, it can provide different optimal combinations of the member cross-sectional areas for the same topology, i.e., multiple solutions for the lower level of the bilevel truss problem as well. The results show the superiority of the proposed niching method over the traditional single and two staged methods for the truss problems that can be found in literature.

REFERENCES

[1] Jagdish Chand Bansal and Kusum Deep. 2012. A modified binary particle swarm optimization for knapsack problems. *Appl. Math. Comput.* 218, 22 (2012), 11042–11061.

[2] Jonathan F. Bard. 1998. *Practical Bilevel Optimization: Algorithms and Applications.* Springer US.

[3] Kalyanmoy Deb. 2001. *Multi-Objective Optimization Using Evolutionary Algorithms.* Wiley.

[4] Kalyanmoy Deb and Surendra Gulati. 2001. Design of truss-structures for minimum weight using genetic algorithms. *Finite Elements in Analysis and Design* 37, 5 (2001), 447 – 465.

[5] Forest Flager, Grant Soremekun, Akshay Adya, Kristina Shea, John Haymaker, and Martin Fischer. 2014. Fully Constrained Design: A general and scalable method for discrete member sizing optimization of steel truss structures. *Comput. & Structures* 140 (2014), 55 – 65.

[6] Ana Friedlander and Francisco A. M. Gomes. 2011. Solution of a truss topology bilevel programming problem by means of an inexact restoration method. *Computational and Applied Mathematics* 30 (2011), 109 – 125.

[7] David E. Goldberg and Jon Richardson. 1987. Genetic algorithms with sharing for multimodal function optimization. In *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms.* L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 41–49.

[8] Prabhat Hajela and E Lee. 1995. Genetic algorithms in truss topological optimization. *International journal of solids and structures* 32, 22 (1995), 3341–3357.

[9] V. Ho-Huu, T. Nguyen-Thoi, T. Vo-Duy, and T. Nguyen-Trang. 2016. An adaptive elitist differential evolution for optimization of truss structures with discrete design variables. *Computers & Structures* 165 (2016), 59 – 75.

[10] A. Kaveh, R. Sheikholeslami, S. Talatahari, and M. Keshvari-Ilkhichi. 2014. Chaotic swarming of particles: A new method for size optimization of truss structures. *Advances in Engineering Software* 67 (2014), 136 – 147.

[11] J. Kennedy and R. Eberhart. 1995. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks, 1995*, Vol. 4. 1942–1948.

[12] J. Kennedy and R. C. Eberhart. 1997. A discrete binary version of the particle swarm algorithm. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 5. 4104–4108.

[13] Uri Kirsch. 1989. Optimal topologies of truss structures. *Computer Methods in Applied Mechanics and Engineering* 72, 1 (1989), 15 – 28.

[14] Steen Krenk and Jan Høgsberg. 2013. *Truss Structures.* Springer Netherlands, Dordrecht, 39–89.

[15] Kang Seok Lee, Zong Woo Geem, Sang ho Lee, and Kyu woong Bae. 2005. The harmony search heuristic algorithm for discrete structural optimization. *Engineering Optimization* 37, 7 (2005), 663–684.

[16] Jian-Ping Li, Marton E. Balazs, Geoffrey T. Parks, and P. John Clarkson. 2002. A species conserving genetic algorithm for multimodal function optimization. *Evolutionary Computation* 10, 3 (Sept. 2002), 207–234.

[17] L.J. Li, Z.B. Huang, F. Liu, and Q.H. Wu. 2007. A heuristic particle swarm optimizer for optimization of pin connected structures. *Comput. & Structures* 85 (2007), 340–349.

[18] Jianhua Liu, Yi Mei, and Xiaodong Li. 2015. An analysis of the inertia weight parameter for binary particle swarm optimization. *IEEE Transactions on Evolutionary Computation* PP, 99 (2015), 1–1.

[19] Guan-Chun Luh and Chun-Yi Lin. 2008. Optimal design of truss structures using ant algorithm. *Structural and Multidisciplinary Optimization* 36, 4 (2008), 365–379.

[20] Guan-Chun Luh and Chun-Yi Lin. 2011. Optimal design of truss-structures using particle swarm optimization. *Comput. & Structures* 89, 23–24 (2011), 2221 – 2232.

[21] Leandro Fleck Fadel Miguel, Rafael Holdorf Lopez, and Letcia Fleck Fadel Miguel. 2013. Multimodal size, shape, and topology optimisation of truss structures using the Firefly algorithm. *Advances in Engineering Software* 56 (2013), 23 – 37.

[22] Seyedali Mirjalili and Andrew Lewis. 2013. S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization. *Swarm and Evolutionary Computation* 9 (2013), 1–14.

[23] Ali Mortazavi and Vedat Toğan. 2016. Simultaneous size, shape, and topology optimization of truss structures using integrated particle swarm optimizer. *Structural and Multidisciplinary Optimization* 54, 4 (2016), 715–736.

[24] D. Parrott and Xiaodong Li. 2006. Locating and tracking multiple dynamic optima by a particle swarm model using speciation. *IEEE Transactions on Evolutionary Computation* 10, 4 (2006), 440–458.

[25] A. Petrowski. 1996. A clearing procedure as a niching method for genetic algorithms. In *Proceedings of IEEE International Conference on Evolutionary Computation, 1996.* 798–803.

[26] S Rajeev and CS Krishnamoorthy. 1992. Discrete optimization of structures using genetic algorithms. *Journal of structural engineering* 118, 5 (1992), 1233–1250.

[27] Wenyan Tang, Liyong Tong, and Yuanxian Gu. 2005. Improved genetic algorithm for design optimization of truss structures with sizing, shape and topology variables. *Internat. J. Numer. Methods Engrg.* 62, 13 (2005), 1737–1762.