# Truss Topology Optimization using an Improved Species Conserving Genetic Algorithm

Jian-Ping Li
School of Engineering & Informatics, University of Bradford, email: j.p.li@bradford.ac.uk

*Abstract*— The aim of this paper is to apply and improve the species conserving genetic algorithm (SCGA) to search in a single run multiple solutions of truss topology optimization problems. A species is defined as a group of individuals with similar characteristics and is dominated by its species seed. The solutions of an optimization problem will be selected from the found species. In order to improve the accuracy of solutions, a species mutation technique is introduced to improve the fitness of the found species seeds and the combination of a neighbor mutation and a uniform mutation is applied to balance exploitation and exploration. A real-vector is used to represent the corresponding cross-sectional areas and a member is thought to be existent if its area is bigger than a critical area. A finite element analysis model has been developed to deal with more practical considerations in modeling, such as existences of members, kinematic stability analysis and the computation of stresses and displacements. Cross-sectional areas and node connections are decision variables and optimized simultaneously to minimize the total weight of trusses. Numerical results demonstrate that some truss topology optimization examples have many global and local solutions and different topologies can be found by using the proposed algorithm on a single run and some trusses have smaller weight than the solutions in the literature.

*Keywords*— topology optimization, genetic algorithm, species conservation, truss design.

## 1. Introduction

Optimization of truss structures has always been one of the most active areas of researches for many years in the field of optimization algorithms and applications in engineering (Eschenauer 2001; Kicinger et al. 2005). Generally, truss optimizations can be classified into three categories: (1) Sizing optimization. Cross-sectional areas of members are considered as decision variables, while the structural geometry is fixed. (2) Configuration optimization. Node coordinates are decision variables, while the connections (elements) of nodes are fixed (Imai and Schmith 1981). (3) Topology optimization. The connections of nodes are decision variables (Deb and Gulati 2011; Ringertz 1985; Krisch 1989). Many of studies are the combination of the above three types of optimizations (Wang *et al.* 2002).

Various techniques have been developed to find optimal truss structures. Classical optimization methods, such as the branch-and-bound algorithm (Ringertz 1986), were firstly used. However, many recently developed algorithms use evolutionary computations to solve truss optimization problems, such as genetic algorithms (Hajela and Lee 1995; Tang *et al.* 2005), particle swarm optimizations (Fourie and Groenwold 2002), simulated annealing algorithms (Lamberti 2008), ant colony optimizations (Kaveh *et al.* 2008; Luh and Lin 2008) and artificial bee colony algorithms (Sonmez 2011). In this paper, only techniques for topology optimization will be discussed.

Assuming that nodes and their coordinates are given, there are two ways to present topological structures: (1) A combination of triangles (Kawamura *et al.* 2002). This method can be only used in plane trusses. (2) A collection of bars (elements) of two nodes. A ground structure, which is a complete truss with all possible connections among all the nodes, is commonly used in developing truss topology algorithms. Optimal results are parts of the ground structure.

A major difficulty in solving truss topology optimization is that some structures might represent a singular point in the design space of a given problem. Krisch (1989) presents some analytical conditions, such as loading conditions and structure stability, to obtain optimal geometries. In multi-level methods (Ringertz

1985; Hajela and Lee 1995), when a topology analysis is performed based on a ground structure, member cross-sectional areas and truss configurations are assumed to be fixed. Once a topology is found, the member areas and /or configuration of the obtained topology are optimized. It is obvious that such a method may not always provide a global solution. When genetic algorithms are used, the cross-sectional areas of members are represented by strings of some binary data. Grieson and Pak (1993) suggested the use of an extra bit to indicate the existence of a member. Ohsaki (1995) added a topological bit to the left of each string to indicate the existence of a member. Su *et al*. (1990) used two separate matrixes to present the cross-sectional areas and the existences of members. A random number is generated to decide the value of each topological bit in the individuals in the initial generation. Deb and Gulati (2001) introduced a new methodology to present the existences of members so that the cross-sectional areas and topologies can be optimized simultaneously. A real-vector is used to present the cross-sectional areas of a ground structure. When the cross-area of one member is bigger than a given critical number, which is the minimum cross area and a positive number, it is assumed that the corresponding element is existent; otherwise it is assumed to be absent. Generally, the lower limits of cross-sectional areas are less than the critical number. Since cross- sectional areas are real numbers, a real-coded genetic algorithm should be used. Deb and Gulati (2001) applied a simulated binary crossover (SBX) and a parameter-based mutation operator (Sue et al. 2009) to solve truss optimizations.

Truss optimization is also complex, in which there are many constraints, such as stresses, displacements, and buckling (Rozvany 1996). The number of stress constraints is dependent on the number of elements, and the number of displacement constraints is the function of nodes or dependent to requirements. Generally, when the cross sectional area of a bar member increases, the weight of the bar will increase and its stress will decrease. A solution should be located on boundaries of constraints. All intersectional points of constraints are potential solutions, therefore, a truss optimization may have multiple solutions and most of them are local optimal solutions. In order to search global solutions, the species conserving genetic algorithm (SCGA) (Li et al. 2002) is improved and applied in this paper to solve truss topology optimization problems.

This paper is organized as follow: Section 2 describes the general forms of truss topology optimization problems. In Section 3, the strategies of improving the species conserving genetic algorithm (SCGA) are presented. In Section 4, a number of truss topology optimization problems from the literature are solved with the proposed algorithms. Finally, some conclusions are included in Section 5.

## 2. Truss Topology Problems

A truss structure is a collection of bar members. The end points of members (bars) are called nodes and a member is a connection of two nodes. Mathematically, a truss topology optimization problem can be formulated as a nonlinear programming problem (Deb and Gulati 2001):

$$\underset{\mathbf{x},\mathbf{A},\mathbf{E}}{\text{minimize}}\, weight(\mathbf{x},\mathbf{A},\mathbf{E}) = \sum \rho_i l_i A_i^{/} \qquad (1)$$

Subject to

G1≡Truss is acceptable to the user,

G2≡Truss is kinetically stable,

G3≡ $\sigma_i^t \geq s_i \geq \sigma_i^c,\ \ \mathrm{i} = 1,2,\cdots,\mathrm{m}$

G4≡ $\delta_j^{\min} \leq \delta_j \leq \delta_j^{\max},\ \ \mathrm{j} = 1,2,\cdots,\mathrm{n}$

G5≡ $A_i^{\min} \leq A_i \leq A_i^{\max},\ \mathrm{i} = 1,2,\cdots,\mathrm{m}$

Where:

m Number of elements;

n Number of nodes;

$\mathbf{x}$ Coordinates of nodes, $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$;

$\varepsilon$ Critical area (a small positive number);

**A** A real vector of cross-sectional area of bars, $\mathbf{A} = (A_1, A_2, \cdots, A_n)$,

$$A_i^{/} = \begin{cases} A_i & \text{if } A_i \geq \varepsilon \\ 0 & \text{if } A_i < \varepsilon \end{cases};$$

$A_i^{\min}, A_i^{\max}$ The lower and upper limits of the cross area of Element *i*;

**E** Connections of nodes (Elements or Bars), $\mathbf{E} = \{e_{i,j}, i \neq j\}$;

$\rho_i$ Density of Element *i*;

$l_i$ Length of Element *i;*

$s_i$ Stress in Element *i;*

$\sigma_i^t, \sigma_i^c$ Material tensile and compressive strength of Element *i;*

$\delta_j$ Displacement of Node *j;*

$\delta_j^{\min}, \delta_j^{\max}$ The lower and upper allowance of the displacements of Node *j*.

The design *variables* may be the combination of:

- the coordinates of nodes ( $\mathbf{x}$ ). The coordinates of nodes are decision variables in configuration optimization;
- and/or the cross-sectional areas (**A)**. The cross-area of a member can be continuous (Deb and Gulati 2001) or discrete (Kaveh and Kalatjari 2003). For sizing optimization, $A_i$ will be non-negative number. For topology optimization, $A_i$ can be a real number (Deb and Gulati 2001);
- and/or elements (the connections of nodes). The connections of nodes will depend on how to construct the models of a given problem. Some algorithms introduce a series of variables or bits to present the connections of nodes. Deb and Gulati (2001) used the cross-areas to present the existences and no extra variables were introduced.

The relationships between optimization types and decision variables are listed in Table 1. In most studies, simultaneous optimization of sizing, configuration and topology has been used and all three types of variables will be decision variables (Wang et al. 2002; Ohsaki 1998).

Table 1: Decision variables of truss optimizations

| Optimization type | Decision variables |
|---|---|
| Sizing | Cross-sectional areas (A>0) |
| Configuration | Coordinates of nodes ( $\mathbf{x}$ ) |
| Topology | Connections of nodes |

Deb and Gulati's method (2001) of presenting cross-sectional areas is used in this paper for truss topology optimization. A solution is represented with a vector of *m*-real numbers (areas) while a phenotype of the truss may have *m* members. The presence and absence of a member is determined by comparing its cross-sectional area with a user-defined small and positive critical area, called the critical number.

Element *i* is assumed to be absence in a truss, if

$$A_i < \varepsilon \qquad (2)$$

or Element *i* exists if

$$A_i \geq \varepsilon \qquad (3)$$

In order to find a truss topology, the lower limit ( $A_i^{\min}$ ) of cross-sectional area should be less than the given positive critical number (ε). Generally, $A_i^{\min} = -A_i^{\max}$, therefore, one member will have a similar probability being present or absent in a truss. When a truss is generated, all the elements will be checked if they are present or not. When an element has been identified to be absent in the given truss, the element will be removed from the element list of the truss.

The above method introduces another problem. If many elements do not exist in the given truss, some nodes may not be used by any existing element. In order to avoid singular stiffness in calculating stresses and displacements, these nodes should be removed.

A feasible solution/truss must satisfy the following constraints:

**Constraint G1**: The user specifies the locations and the numbers of basic nodes for supports and loads. A topological truss must have all the basic nodes. Since some unused nodes will be removed from the model, some basic nodes may be absent. If any of the basic nodes is absent, the truss is marked as 'fail' and the penalty value is set as one (P=1) and no future calculation is done.

**Constraint G2**: A truss must be kinematically stable so that it is not a mechanism. If a truss is instable, the corresponding FEA model cannot be solved. In order to speed up the FEA process, Grubler's criterion (Tuma and Walsh 1997) is used to check the degrees of freedom (DOF), which is defined as:

$$\text{DOF} = D \times n - m - n_i \tag{4}$$

Where $D$ is the dimension of a truss system (D=2 for 2-D trusses; while D=3 for 3 D trusses); $n$ is the number of nodes; m is the number of existing elements and $n_i$ is the number of degrees-of-freedom lost at the support nodes.

If the degree-of-freedom (DOF) is positive, the corresponding truss is a mechanism and the corresponding truss is marked as 'fail' and the penalty value is set as one (P=1) and no further calculation is needed. However, for some complex trusses, this method can not determine if a truss is a machine. For example, if there are some redundant bars in one region and local machines in other, its DOF may be not positive. In this case, A FEA will be run to determine if the structure satisfies the required displacement constraints. When a truss is composed of local machines , the FEA stiffness matrix becomes singular.

**Constraint G3**: A material has two allowable strengths: a compressive strength and a tensile strength. A bar is compressed if its stress is negative, or it is stretched if its stress is positive. The stresses of any member in a truss must be between its compressive strength and tensile strength, or the truss will fail.

The normalized penalty value is calculated as follow:

$$P_i = \begin{cases} \dfrac{|s_i|}{|\sigma_i^c|} & s_i < \sigma_i^c \\ 0 & \sigma_i^c \le s_i \le \sigma_i^t \\ \dfrac{|s_i|}{|\sigma_i^t|} & s_i > \sigma_i^t \end{cases} \tag{5}$$

where $\sigma_i^c$ is the compressive strength of Bar *i* and $\sigma_i^t$ is the tensile strength of Bar *i*.

The penalty value of stresses in the given truss is the maximum value of the penalty of each member:

$$P = \{\max P_i\} \tag{6}$$

If a truss is subjected to a number of loading conditions applied separately, the FEA process will be run for each case. When the penalty value in any loading case is greater than one (marked as 'fail'), the FEA process will be stopped.

**Constraint G4**: Some nodes may have allowable displacements. They must not defect more than the allowable limits. The similar method to the one used in G3 is applied to calculate its penalty value.

**Constraint G5:** The cross-sectional areas must be bounded with specified limits. Those conditions will be automatically satisfied, because all the trusses are generated according to its lower and upper limits.

The feasible region of a truss optimization problem is the set of all possible feasible solutions that satisfied all the above constraints.

The objectives of truss optimizations may be the minimization of (1) the total weight, (2) the total cost (Ohsaki 1995, 1998). In this paper, the *objective* of a truss optimization problem is to minimize the total weight of structures. The cross-sectional area of an element will be taken into account in calculating the weight of the truss only when its area is bigger than the critical area ($\varepsilon$). In real truss designs, some other properties, such as cost, may need to be taken into account.

The fitness function is defined as:

$$f(\mathbf{x},\mathbf{A},\mathbf{E}) = -weight(\mathbf{x},\mathbf{A},\mathbf{E}) - c_f P \qquad (7)$$

Where P is the penalty value and $c_f$ is the weight factor of penalty values. According the above descriptions, the penalty value is positive if the truss is infeasible or it is zero. The value of $c_f$ should depend on problems and should be bigger enough. In this paper, $c_f = 10^5$.

With Equation (7), the original truss optimization problem is changed to maximize the fitness subject to all the above constraints, because in genetic algorithms, the bigger the value of the fitness for an individual, the better the individual higher a fitness of a solution and the more chances to survive in the next generation.

$$\underset{\mathbf{x},\mathbf{A},\mathbf{E}}{\text{maximize}}\, f(\mathbf{x},\mathbf{A},\mathbf{E}) \qquad (8)$$

A finite element analysis (FEA) model shown in Figure 1 has been developed in this paper to check the existence of basic nodes, to compute the “degree-of-freedom” and to calculate the stresses and displacements of a truss under the given loadings and support constraints. The developed FEA is capable of dealing with multiple loading cases.

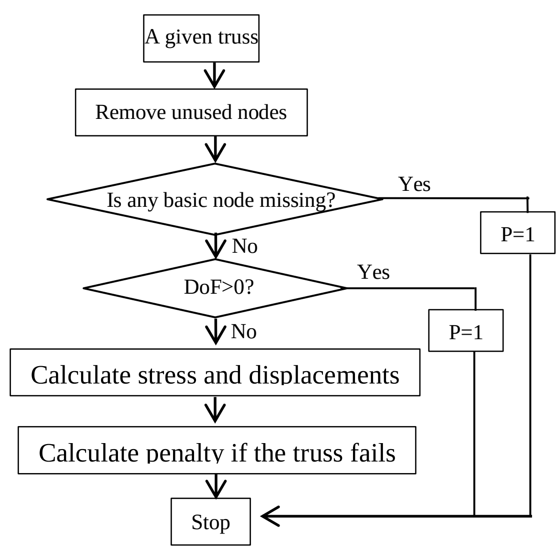



Figure 1. The FEA process.

## 3. Proposed Optimization Algorithm

### *3.1 Species Conserving Genetic Algorithms*

Many real-engineering optimization problems are very complex and have many global and local solutions. The species conserving genetic algorithm (SCGA) (Li et al. 2002) was developed to search multiple solutions of multimodal optimization problems on a single run.

The SCGA is based upon the definition of species. A species is defined as a group of individuals in a population with similar characteristics and dominated by the best individual called the species seed. Briefly, a species $S_i$ is centered upon its dominating individual, called the species seed $\mathbf{x}^*$, if for every individual $\mathbf{y} \in S_i$

$$d(\mathbf{x}^*, \mathbf{y}) < r_s \tag{9}$$

and

$$f(\mathbf{y}) \le f(\mathbf{x}) \tag{10}$$

Where $r_s$ is the species radius, $d(\mathbf{x}^*, \mathbf{y})$ is the distance between $\mathbf{x}^*$ and $\mathbf{y}$. Figure 2 illustrates a sample distribution of species in a two-dimensional domain. A species is formed of actual individuals and occupies a region of the feasibility domain. Some of the individuals of a species are located in the intersection. This is because a fixed radius is used to identify species.

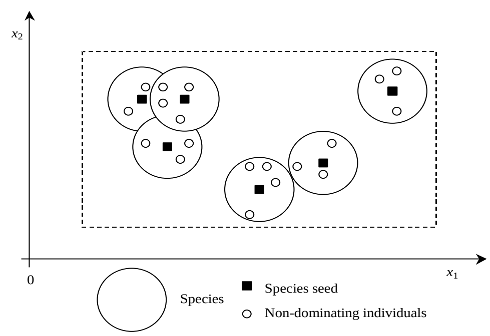



Figure 2. A sample distribution of species in a two-dimensional domain.

The procedure of the SCGA (Li et al. 2002) is shown in Figure 3.

**Begin**
- $t = 0$;
- **Initialize** $G(t)$ **;**
- **Evaluate** $G(t)$ **;**
- **while (not termination condition) do**
  - **Determine species seeds** $X_s$ **;**
  - **Select** $G(t+1)$ **;**
  - **Crossover** $G(t+1)$ **;**
  - **Mutate** $G(t+1)$ **;**
  - **Evaluate** $G(t+1)$ **;**
  - **Conserve species from** $X_s$ **in** $G(t+1)$ **;**
  - $t = t+1$;
- **end (while)**
- **Identify global optima;**

**end**

Figure 3: The structure of the SCGA

In the SCGA, the population is divided into several species according to their similarities and each of these species is built around a dominating individual, called species seed. Species seeds that are found in the current generation will be conserved by moving them into the next generation.

There are three special procedures in the SCGA:

- Determining species seeds: This procedure is to identify species seeds from a current population. A best unmarked individual is selected as a species seed and all individuals within this species in the population are marked. This process is continued until all the individuals in the population have been marked.
- Conserving species seeds: The new generation is constructed by applying the usual genetic operations: selection, crossover and mutation and by "copying" the found species into the population to keep its diversity according to their similarity.
- Identifying global solutions: The global solutions are the fittest individual in $X_S$ (the species seed set) and all the individuals in $X_S$ that are "close to" the minimal objective. For this purpose a *solution acceptance threshold* $r_f$ $(0 < r_f \leq 1)$ is used to determine if a species is a solution. An individual ($\mathbf{x}$) in $X_S$ will be identified as a solution, if it satisfies the following inequality:

$$f(\mathbf{x}) \leq f_{\max} - (f_{\max} - f_{\min}) \times r_f \qquad (11)$$

Where $f_{\max}$ and $f_{\min}$ are the best and worst fitness (objective value).

### *3.2 Modifications of the SCGA*

The SCGA is based on the traditional genetic algorithms, and the accuracies of solutions are dependent on the selection of crossover and mutation methods. In order to improve the quality of solutions, the following modifications of the SCGA have been implemented:

(1) All the trusses must be within the feasible region. The reason is that there are many constraints in a truss optimization. Only a small part of the region bounded by the lower and upper limits of the cross- sectional areas is within the feasible region. If all the trusses generated by the SCGA are calculated by a FEA model, lots of calculations are wasteful. This means that all infeasible solutions will be dead immediately and removed from the current population. Therefore, the initializing process must be continued for some extra objective evaluations until at least one feasible solution has been found. If no feasible solution is found in the given number of tries, the optimization process will be terminated. All the infeasible solutions will be replaced with a feasible solution. If offspring generated by crossover and mutation operators are infeasible, they will be replaced by one of their parents. These strategies will guarantee that all individuals in a population are within the feasible region.

(2) The distance between two individuals $\mathbf{x}_i = [x_{i1}, x_{i2}, \cdots, x_{im}]$ and $\mathbf{x}_j = [x_{j1}, x_{j2}, \cdots, x_{jm}]$ is defined as:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{k=m} d_k^2} \qquad (12)$$

Where:

$$d_k = \begin{cases} |x_{ik} - x_{jk}| & \text{if } x_{ik} > 0 \text{ or } x_{jk} > 0 \\ 0 & \text{others} \end{cases}$$

Distances are important parameters in species determination and species conservation procedures. In the above distance definition, only positive components are taken into account, because if a cross area is negative number or less than the critical number (ε), the corresponding element is absent in the truss.

(3) Mutation operators: The combination of a uniform mutation and a uniform neighborhood mutation are used in this paper. A random number ( $R$ ) is used to control which mutate operator will be used in the current generation.

$$\begin{cases} R \le t_m & \text{Uniform Mutation} \\ R > t_m & \text{Neighbour-hood Mutation} \end{cases} \quad (13)$$

If the random number ( $R$ ) is less or equal to the mutation threshold ( $t_m$ ), a uniform mutation is used,

$$x_j^{/} = x_j^l + R_0 \times (x_j^u - x_j^l) \quad (14)$$

or the following neighborhood mutation is applied:

$$x_j^{/} = x_j + R_1 \times r_m \quad (15)$$

Where $R_0$ is a random number over [0,1], $R_1$ is a random number over [-1,1] and $r_m$ is the neighborhood mutation radius that is used to determine the mutation neighborhood boundary.

The advantage of the combination is to combine global searches and local searches. Global searches can explore the entire space and improve the probability to find global solutions, while local searches can help the improvement of the quality of the solutions.

(4) A species mutation technique is added into the optimization process. A species mutation technique is a neighborhood mutation. A found species seed will search its local area to improve its fitness. Li *et al.* (2010) showed that a species mutation technique can greatly decrease objective evaluations for the SCGA in searching solutions of the test problems.

The following modified species mutation technique is applied in this paper after all the operators:

Let **x** be the selected species for mutation, $r_{ms}$ be the species mutation neighborhood [0,1], the new individual ( $\mathbf{x}^{/}$ ) will be generated with the following formula:

$$\mathbf{x}^{/} = \mathbf{x} + R_1 \cdot r_s \cdot r_{ms} \quad (16)$$

If $f(\mathbf{x}^{/}) > f(\mathbf{x})$ and $\mathbf{x}^{/}$ is within the feasible region, the species $\mathbf{x}$ is replaced with $\mathbf{x}^{/}$ .

In order to further improve the quality of solutions, the species mutation radius ( $r_{ms}$ ) is calculated by using the following equation:

$$r_{ms} = 0.5 \times e^{-\frac{g}{N_{ms}}} \quad (17)$$

Where g is the current generation number. The species mutation radius ( $r_{ms}$ ) will decrease exponentially with the increase of the number of generations. $N_{ms}$ is a species mutation control parameter and is used to control the decreasing speed of $r_{ms}$ .

The species mutation technique is added into the SCGA and the updated process is shown in Figure 4.

**Begin**

$t = 0$;
**Initialize** $G(t)$ **;**
**Evaluate** $G(t)$ **;**
**while (not termination condition) do**
  **Determine species seeds** $X_s$ **;**
  **Select** $G(t+1)$ **;**
  **Crossover** $G(t+1)$ **;**
  **Mutate** $G(t+1)$ **;**
  **Evaluate** $G(t+1)$ **;**
  **Conserve species from** $X_s$ **in** $G(t+1)$ **;**
  ***Species Mutation;***
  $t = t+1$;
**end (while)**
**Identify global optima;**

**end**

Figure 4. The structure of the improved SCGA.

## 4. Numerical Examples

There are many references on truss topology optimization in the literature. Only the references, in which the cross-sectional areas are real numbers and the optimal objective is to minimize the total weight, are compared in this section.

Table 2. Default algorithm parameters.

| Name | Value |
|---|---|
| Population Size ( $N_p$ ) | 50 |
| Crossover Probability ( $p_c$ ) | 0.5 |
| Mutation Probability( $p_m$ ) | 0.1 |
| Mutation threshold ( $t_m$ ) | 0.3 |
| Solution acceptance threshold ( $r_f$ ) | 0.9 |
| Species Mutation Control Parameter ( $N_{ms}$ ) | 500 |

In the proposed algorithm, the genetic operators are a roulette selection, arithmetic crossover and the combination of uniform mutation and neighborhood mutations. The other default parameters are listed in Table 2.The species radius ( $r_s$ ) and neighborhood mutation radius ( $r_m$ ) are dependent on the complexity of a problem. In all figures showing trusses, dimensions are in inches and forces are in lb. The neighbor mutation radius ( $r_m$ ) is set as

$$r_m = 0.1 \times r_s \tag{18}$$

In order to obtain more solutions on a single run, the solution acceptance threshold ( $r_f$ ) is set as 0.9.

### *4.1 Example 1: 15-members, Six-node Truss*

#### *4.1.1 Problem Description*

The 15-member and six node truss (the ground structure) and the loadings are shown in Figure 5. There are 15 possible members. For clarity, the overlapping members are shown with a gap in the figure and the design parameters are listed in Table 3. All 15 cross-sectional areas are used as design variables. In this example, since some bars are very long, buckling constraints should be taken into account in real truss optimal designs. In order to compare with existing algorithms, buckling constraints are not included in this example.

Table 3. Parameters for Example 1.

| | |
|---|---|
| Young’s modulus | 104 ksi |
| Density | 0.1 lb/in3 |
| Allowable compressive strength | 25 ksi |
| Allowable tensile strength | 25 ksi |
| Allowable displacement | 2 in |
| Limits of cross-area | [-3,35]in |

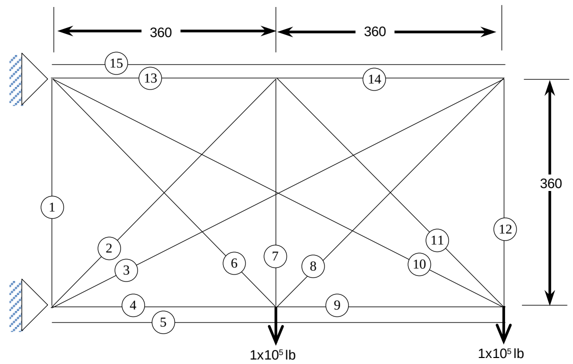


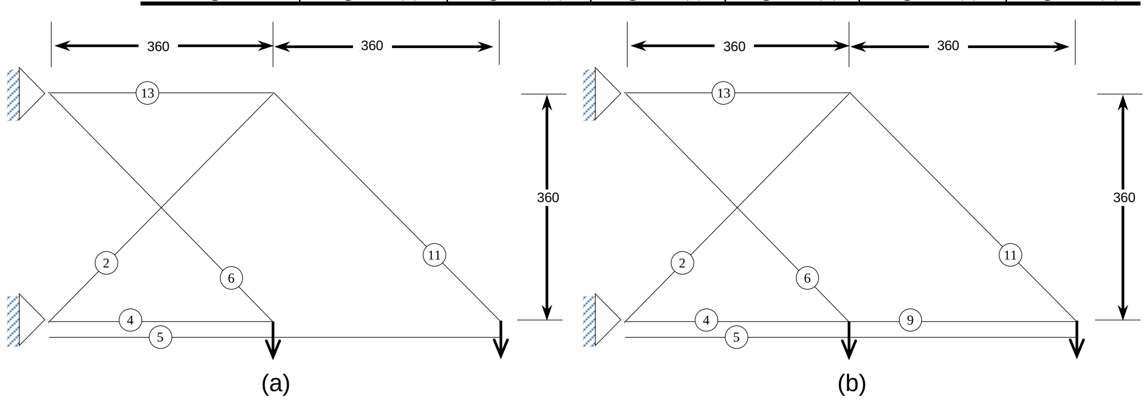

Figure 5. The 15-member and six-node ground structure.

### 4.1.2 Optimal Solutions

Table 4 shows the optimal truss obtained by the proposed algorithm using the default parameters and the species radius ( $r_s = 10$ ) after the generation of 3000. The proposed algorithm has found 4 solutions on a single run and their corresponding connections of nodes are shown as in Figure 6. The best truss (Solution 1) has the same connection as the truss obtained by Deb & Gulati (2001) and Luh and Lin (2008). Only six members shown in Figure 6(a) are necessary in the optimal trusses. Stresses in elements and displacements in nodes of all solutions are tabulated in Table 5 to show how a solution is close to the boundaries of the problem. It is interesting to notice that all solutions lie on the intersections of some constraints. For an example, in Solution 1, the displacements of Node 3 and 5 along the y axis are -1.99999 and -2.00000, that are very close to the allowances.

Table 4. Results of Example 1.

| Element | Deb & Gulati | Luh & Lin | Proposed | | | |
|---|---|---|---|---|---|---|
| No | (2001) | (2008) | Solution 1 | Solution 2 | Solution 3 | Solution 4 |
| 2 | 20.310 | 20.549 | 20.27712 | 21.92580 | 25.46163 | 30.24313 |
| 4 | 5.219 | 5.428 | 5.29368 | 16.33877 | 21.97115 | |
| 5 | 14.593 | 14.308 | 14.33871 | 6.90382 | | 20.84183 |
| 6 | 7.772 | 7.617 | 7.71425 | 6.07553 | 6.08882 | 0.57217 |
| 7 | | | | | | 3.78584 |
| 9 | | | | 7.43428 | 10.37983 | 0.43245 |
| 10 | | | | | | 3.60725 |
| 11 | 20.650 | 20.265 | 21.02456 | 21.39010 | 21.93803 | 21.75424 |
| 13 | 28.187 | 28.876 | 28.15770 | 28.36948 | 31.20734 | 34.99946 |
| Weight (lb) | 4734.34039 | 4730.82397 | 4732.12081 | 4888.80729 | 5011.28834 | 5879.23281 |
| Configuration | Figure 6(a) | Figure 6(a) | Figure 6(a) | Figure 6(b) | Figure 6(c) | Figure 6(d) |



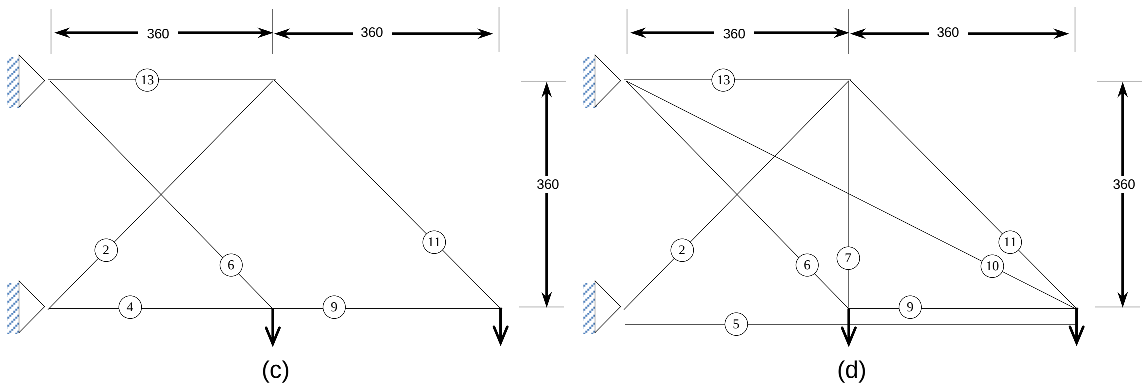



Figure 6. Optimal solutions of Example 1.

Table 4 and Figure 6 also illustrate that the proposed algorithm is capable of searching multiple solutions of truss optimizations on a single run.

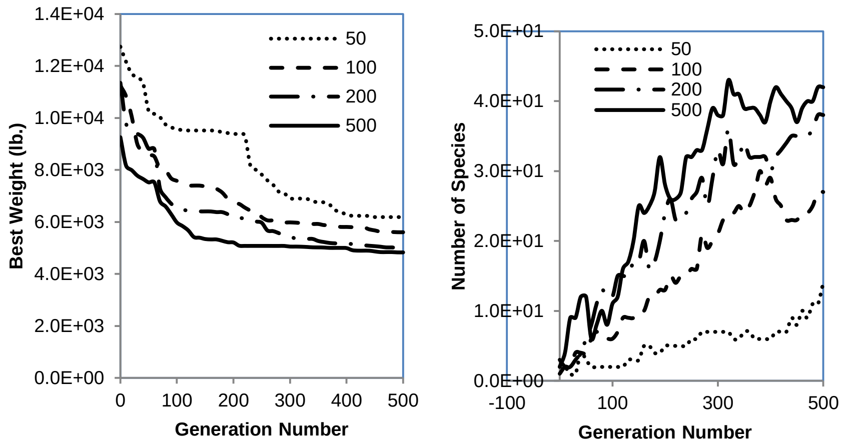
It looks like that Luh & Lin (2008) obtained better results. However, Table 5 shows that the y-axis displacement of Node 3 of the trussed obtained by Luh & Lin (2008) is -2.00001 in, which is little less than the lower allowance (-2in).

Table 5. Stresses and displacements of Example 1.

| Element No. | Stresses in existed elements | | | | | |
|---|---|---|---|---|---|---|
| | Deb & Gulati (2001) | Luh & Lin (2008) | Proposed | | | |
| | | | Solution 1 | Solution 2 | Solution 3 | Solution 4 |
| 2 | -6.96314E3 | -6.88215E3 | -6.97443E3 | -6.45000E3 | -5.55429E3 | -8.59980E3 |
| 4 | -1.91608E4 | -1.84230E4 | -1.88904E4 | -9.00139E3 | -9.10285E3 | |
| 5 | -6.85260E3 | -6.98909E4 | -6.97413E3 | -7.66654E3 | 2.32264E4 | -5.57015E3 |
| 6 | 1.81963E4 | 1.85665E4 | 1.83325E4 | 2.32772E4 | | 1.32389E4 |
| 7 | | | | | | 2.49994E4 |
| 9 | | | | -6.33169E3 | -9.63407E3 | 1.23859E4 |
| 10 | | | | | | 6.65504E3 |
| 11 | 6.84849E3 | 6.97860E3 | 6.72648E3 | 6.61153E3 | 6.44640E3 | 5.80293E3 |
| 13 | 7.09547E3 | 6.92617E3 | 7.10285E3 | 7.04983E3 | 6.40875E3 | 7.80503E3 |
| Node | Displacements at active nodes | | | | | |
| 2 | X: -0.68979<br>Y: -1.99992 | X: 0.66323<br>Y: **-2.00001*** | X: -0.68006<br>Y:-1.99999 | X: -0.32405<br>Y: 2.00000 | X: -0.32770<br>Y: -2.00000 | X: -0.84694<br>Y: -1.80014 |
| 3 | X: -0.49339<br>Y: -1.99870 | X: -0.50321<br>Y: -1.99987 | X: -0.50214<br>Y: -2.00000 | X: -0.55199<br>Y: 2.00000 | X: -0.67453<br>Y: -2.00001 | X: -0.40105<br>Y: -2.00000 |
| 5 | X: 0.25544<br>Y: -0.75678 | X: 0.24934<br>Y: -0.74486 | X: 0.255703<br>Y: -0.75786 | X: 0.25379<br>Y: -0.71819 | X: -0.23071<br>Y: -0.63062 | X: 0.28098<br>Y: -0.90017 |

* this value is out of the given requirement. In the following paper, a ”*” will present that the current constraint is violated.

### 4.1.3 Effects of Population Sizes




(a) Best Weight (b) Number of Species

Figure 7. Effects of generations with different population sizes for Example 1.

Figure 7 illustrates the effects of generations with different population sizes on the best weights of solutions and the numbers of species. As expected, the best weight decreases with the increase of the number of generations. If a higher accurate solution is required, the proposed algorithm has to been run more generations. A population with more individuals likely converges faster, since the searching space is larger. Figure 7(b) demonstrates that only one or two species have been found in initial populations. This is because the truss optimization problem is restricted with a lot of constraints and only a small portion of the area bounded by the lower and upper limits of variables are the feasible region. Once at least one feasible solution has been identified, more and more species can be found in the further evolutionary process. A higher population size will increase the probability of finding species.

### 4.1.4 Efficiency on Objective Evaluations

In this section, the efficiencies of the proposed SCGA are investigated. The proposed SCGA will record the best weight when the number of the objective evaluations ( $N_f$ ) is up to 1000, 5000, $10^4$ and $2x10^4$. For each population size, the proposed algorithm has been run for 20 times. The mean values of the best weights are shown in Figure 8. It is very interesting to notice that the increase of the population size cannot improve the best weights if the maximum number of objective evaluations is fixed. For an example, if the number of objective evaluations is fixed to 20000, the mean best weight is 5437.2 lb when the population size is 50; while the mean best weight is 6099.2 lb when the population size is 400. It looks like that the algorithm is more efficient when the population size is 50 in solving this problem. In the following sub-sections, all the population size is set as 50.

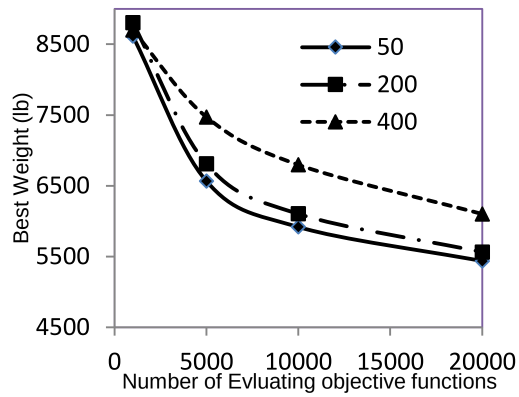
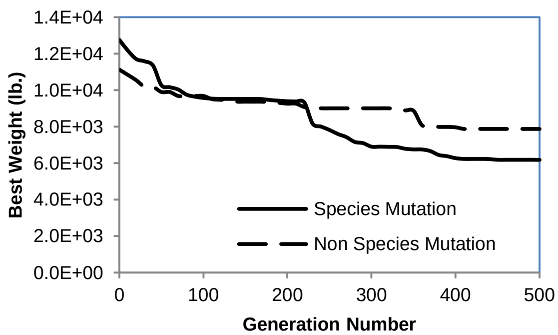
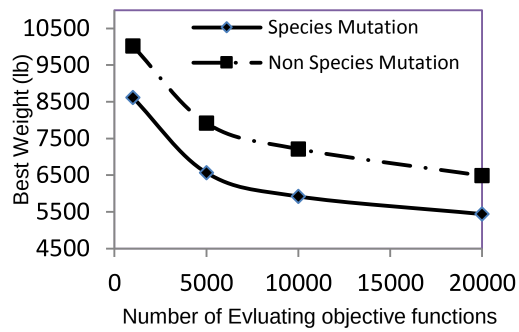



Figure 8. Effects of objective evaluations with different population sizes.

### *4.1.5 Effects of Species Mutation*




(a) Performances of species mutation




(b) Best weight vs. given objective evaluations

Figure 9. Effects of species mutation.

Figure 9 (a) illustrates the process of the best weights of two optimization procedures with and without the species mutation. The SCGA with the species mutation looks like converges faster than without this technique.

Each procedure is run for 20 times. The results are summarized in Figure 9 (b). The mean value of the best weights obtained with the species mutation is 5437.2 lb when the number of objective evaluations is up to 20000, which is much better than the weight (6483 lb) obtained without the species mutation.

### *4.1.6 Effects of Mutation Techniques*

In the proposed SCGA, the mutation is the combination of a uniform mutation and a neighborhood mutation. In this section, the effects of this combination will be analyzed. The proposed algorithm will output the best weights, numbers of species and the improvement ratios after each 50 generations.

The improvement ratio is defined as:

$$r_i = \frac{w_i - w_{i-1}}{w_{i-1}} \tag{19}$$

Where $w_i$ is the current best weight.

Figure 10 illustrates the effects of the mutation threshold ($t_m$):

- Case 1: $t_m = 1$. Only uniform mutation is used.
- Case 2: $t_m = 0.5$. 50% is uniform mutation and 50% is neighbourhood mutation.

- Case 3: $t_m = 0$. Only neighbourhood mutation is used.

Figure 10(a) shows that the number of species remains to be constant in Case 3, and the uniform mutation (Case 1) is capable of finding more and more species. In Case 3 the neighborhood mutation looks hard to find new species. This is because the neighborhood mutation radius is 10% of the species radius and the algorithm just search the local area around an individual. Of course, if the mutation radius is increased, it will have more chance to find global solutions.

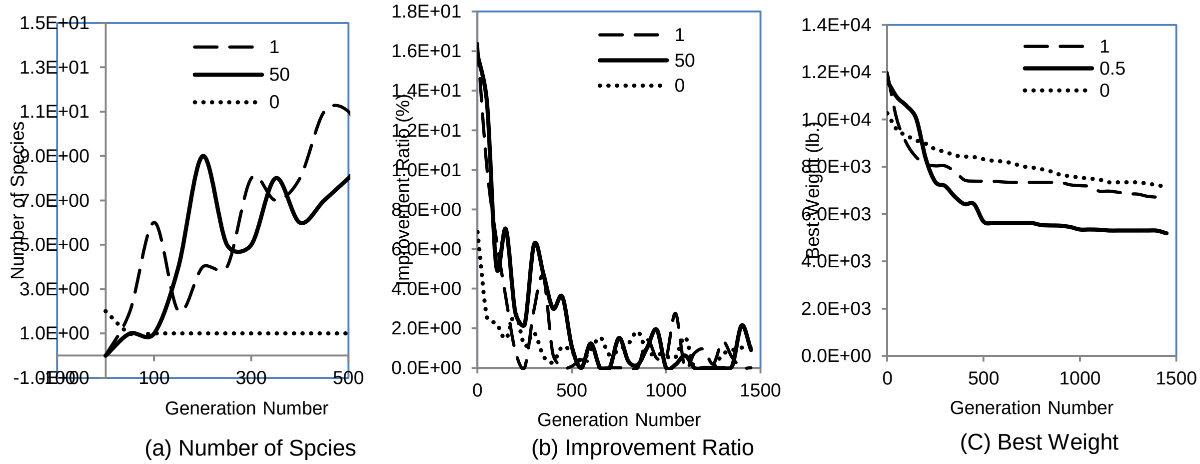



Figure 10. Effects of mutation techniques.

Figure 10(b) illustrates that the uniform mutation is very efficient in the early stage. Its improvement ratio drops very quickly and is close to zero in the further calculation. For the neighborhood mutation, the improvement ratio is better in the middle stage than the uniform mutation and the best weight of the truss continually decreases.

Figure 10 shows that the proposed algorithm with this combination (Case 2) out-performs other two cases.

### *4.2 Example 2: Ten-node, 2-D truss*

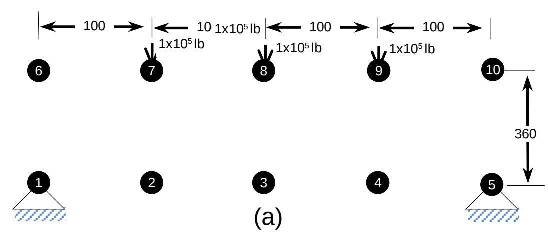



Figure 11. Ten nodes of Example 2.

The coordinates of 10 nodes are shown in Figure 11. The corresponding ground structure has a total of 45 (or $C_{10}^2$) members. All the parameters are the same as before, except that the lower and upper bound of cross-sectional areas are -1.0 and 1.0 in$^2$. This is a very hard problem. Deb and Gulati (2001) applied a big population size of 1800 to solve this problem. In this section, a small population size (50) is applied. After 8000 generations, the proposed SCGA found 2 solutions. The solutions are listed in Table 6 and the corresponding elements are illustrated in Figure 12. The best weight (43.99993 lb) in this paper is similar to the structure obtained by Deb and Gulati (2001). It is quite interesting to notice that the best truss obtained by the proposed algorithm has different connections to the solution by Deb and Gulati(2001).

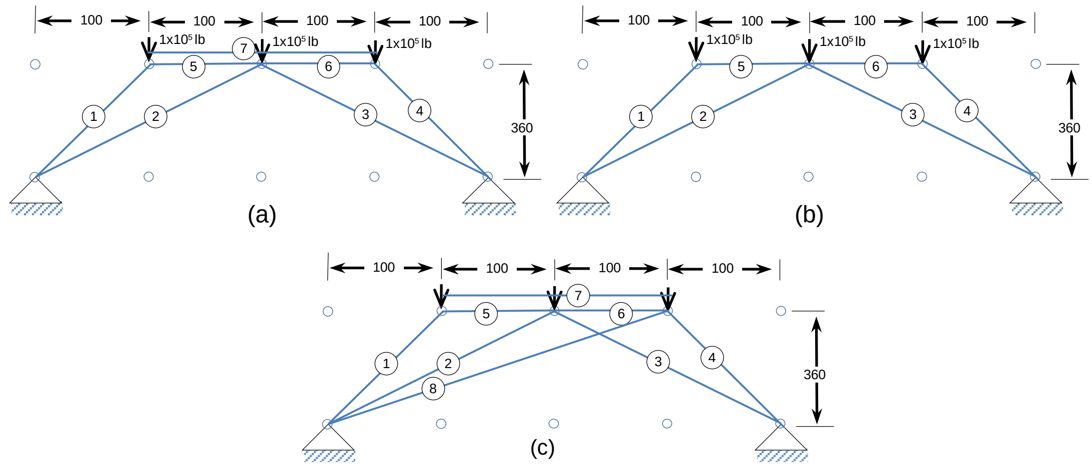



Figure 12. Optimal 10 node ground structuer of Example 2.

Table 6. Results of Example 2.

| **Element** No. | Deb & Gulati (2001)0 | Proposed | |
|---|---|---|---|
| | | Solution 1 | Solution 2 |
| 1 | 0.566 | 0.565684 | 0.56586 |
| 2 | 0.477 | 0.447213 | 0.42752 |
| 3 | 0.477 | 0.447213 | 0.47031 |
| 4 | 0.566 | 0.565684 | 0.66963 |
| 5 | 0.082 | 0.400000 | 0.21976 |
| 6 | 0.080 | 0.400000 | 0.32669 |
| 7 | 0.321 | | 0.25187 |
| 8 | | | 0.08181 |
| Weight (lb) | 45.38098 (44.033*) | 43.99993 | 50.63751 |
| Connections | Figure 12(a) | Figure 12(b) | Figure 12(c) |

With the same parameters, the proposed SCGA has run for 20 times and the best weights are recorded when the number of objective evaluation is up to the given numbers. The results are summarized in Table 7. The same observations can be obtained. The menu value and the standard deviation of the best weights decrease with the increase of the number of objective evaluations. When the objective is up to 20000, the mean value of the best weight is about 87.9077.

Table 7. Effects of number of evaluating objective functions (Example 2).

| $N_f$ | 50 |
|---|---|
| 1000 | 2.2167E2 $\pm$36.6698 |
| 5000 | 1.4115E2 $\pm$28.6506 |
| 10000 | 1.0509E2 $\pm$15.8803 |
| 20000 | 87.9077 $\pm$14.4118 |

## *4.3 Example 3: Two-tier truss*

A two-tier, 39-member and 12-node ground structure is shown in Figure 13. Symmetry along the middle nodes is used to reduce the number of variables to 21. All parameters of the problem are the same as before, except that the lower and upper bounds of the cross area of members are -2.25 and 2.25 in$^2$, the allowable strength is 20 ksi, and the critical area is 0.05 in$^2$.

Table 8 and Figure 14 illustrate the results obtained using the proposed SCGA on a single run after 5000 generations with the default algorithm parameters, except the species radius ($r_s = 0.3$). 3 solutions have been found and have different connections of nodes. In the best truss (Solution 1), only 17 members and 10 nodes are retained.

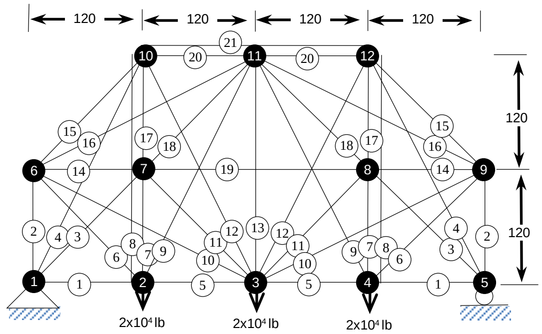



Figure 13. Two-tie-member and 12-node ground structure.

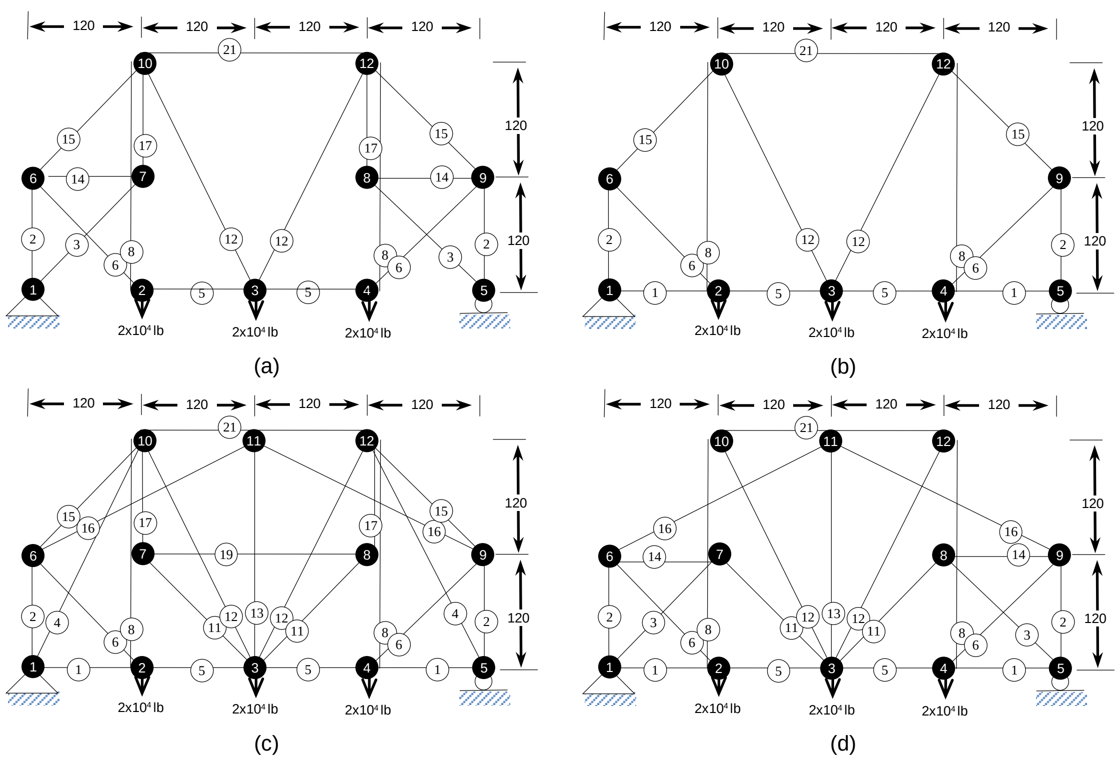



Figure 14. Two-tie-member and 12-node ground structure.

Table 8. Results of two-tie-member and 12-node ground structure.

| Element No. | Deb & Gulati (2001) | Luh & Lin (2008) | Proposed | | |
|---|---|---|---|---|---|
| | | | Solution 1 | Solution 2 | Solution 3 |
| 1 | | 0.05100 | 0.05100 | 0.11871 | 0.05233 |
| 2 | 1.50200 | 1.50200 | 1.50112 | 1.29328 | 1.61043 |
| 3 | 0.05100 | | | | 0.05242 |
| 4 | | | | 0.37023 | |
| 5 | 0.75100 | 0.75100 | 0.75011 | 0.78345 | 1.13691 |
| 6 | 1.06100 | 1.06200 | 1.06108 | 0.96180 | 1.41511 |
| 7 | | | | | |
| 8 | 0.25100 | 0.25000 | 0.25059 | 0.32237 | 0.40567 |
| 10 | | | | | |
| 11 | | | | 0.25734 | 0.05269 |
| 12 | 0.55900 | 0.560000 | 0.56042 | 0.32186 | 0.05032 |
| 13 | | | | 0.13004 | 1.00078 |
| 14 | 0.05100 | | | | 0.13970 |
| 15 | 1.06300 | 1.06300 | 1.06090 | 0.78199 | |
| 16 | | | | 0.14240 | 1.11898 |
| 17 | 0.05200 | | | 0.42001 | |

| | | | | | |
|---|---|---|---|---|---|
| 19<br>21 | 1.00500 | 1.00000 | 1.00157 | 0.15312<br>0.80552 | |
| Weight (lb) | 196.53303<br>(196.54600) | 193.47364 | 193.41720 | 217.07692 | 228.38496 |
| Connections | Figure 14(a) | Figure 14(b) | Figure 14(b) | Figure 14(c) | Figure 14(b) |

With the same parameters, the proposed SCGA has run for 20 times and the best weights are recorded when the number of objective evaluation is up to the given numbers. The results are summarized in Table 9. When $N_f$ increases from 1000 to 20,000, the mean value of the best weight drops from 420.08 lb to 237.55 lb, while the standard deviation decreases from 70.4719 to 20.2995.

Table 9. Effects of number of evaluating objective functions.

| $N_f$ | 50 |
|---|---|
| 1000 | 4.2008E2 ±70.4719 |
| 5000 | 2.7384E2+35.7052 |
| 10000 | 2.4846E2+26.6211 |
| 20000 | 2.3755E2+20.2995 |

### 4.4 *Example 4: 3-D, 39 member, 10 node truss*

In this section, the proposed SCGA is applied to solve a couple of three-dimensional trusses. The node positions of a 39-member ground structure are shown in Figure 15. In order to make design simple, members are grouped by considering the symmetry on opposite sides and the number of variables, which is listed in Table 10, reduces to 11. The material properties are listed in Table 11.

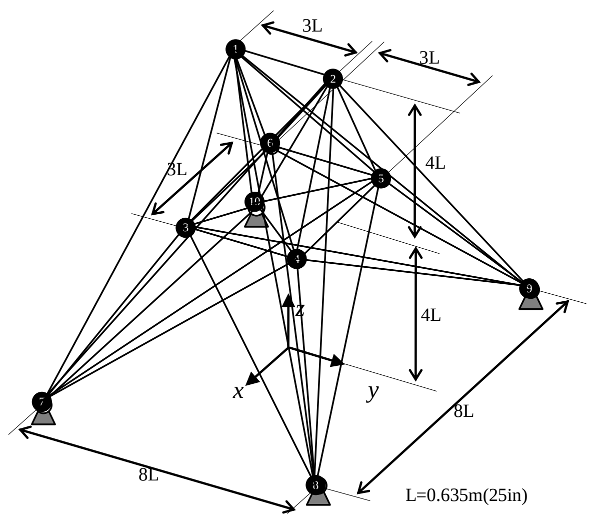



Figure 15. The ground strucuture for the 3-D, 39-member and 10 node truss.

Table 10. Groups of members for Example 4.

| No | Members (node connections) |
|---|---|
| 1 | 1-2 |
| 2 | 1-5, 1-4,2-6, 2-3 |
| 3 | 1-3, 1-6, 2-4, 2-5 |
| 4 | 3-4, 4-5, 5-6, 3-6 |
| 5 | 6-7, 3-10,4-9,5-8 |
| 6 | 4-7, 3-8, 6-9,5-10 |
| 7 | 3-7, 4-8, 5-9, 6-10 |
| 8 | 3-9, 4-10, 5-7, 6-8 |
| 9 | 3-5, 4-6 |
| 10 | 1-7,1-10, 2-8,2-9 |
| 11 | 1-8, 1-9, 2-7, 2-10 |

Table 11. Material parameters for Example 4.

| Young's modulus | $10^4$ ksi |
|---|---|
| Density | 0.1 lb/in$^3$ |
| Allowable compressive strength | 40 ksi |
| Allowable tensile strength | 40 ksi |
| Allowable displacement | 0.35 in |
| Limits of cross-area | [-3,3]in$^2$ |
| Critical area | 0.005 in$^2$ |

### *4.4.1 Example 41: Load 1*

Two downwards loadings of 500 lb are applied at each top node. With the default parameters, $R_f = 0.98$, $R_s = 0.05$ ,and $N_{ms} = 300$, the proposed SCGA have been found 2 solutions, which are listed in Table 12. The best solution (Solution 1) is better than the truss obtained by Luh and Lin (2008), which is better than the truss obtained by Deb and Gulati (2001). Figure 16 illustrates that the connections of nodes in Solution 1 are different to the truss obtained by Deb & Gulati (2001) and Luh & Lin (2008).

Table 12. Results of Example 41.

| Element No. | Deb&Gulati (2001) | Luh&Lin (2008) | Proposed | |
|---|---|---|---|---|
| | | | Solution 1 | Solution 2 |
| 1 | 0.16600 | 0.00500 | | 0.81593 |
| 2 | | | | 0.00673 |
| 3 | | | | 0.43978 |
| 6 | | | | 0.00502 |
| 7 | | | | 0.00526 |
| 10 | 0.40900 | 0.01300 | 0.01631 | 0.08971 |
| 11 | 0.07100 | 0.01500 | 0.00896 | |
| Weight (lb) | 46.68419 (47.93000) | 2.81982 | 2.45568 | 34.23413 |
| Connections | Figure 16(a) | Figure 16(a) | Figure 16(b) | Figure 16(c) |

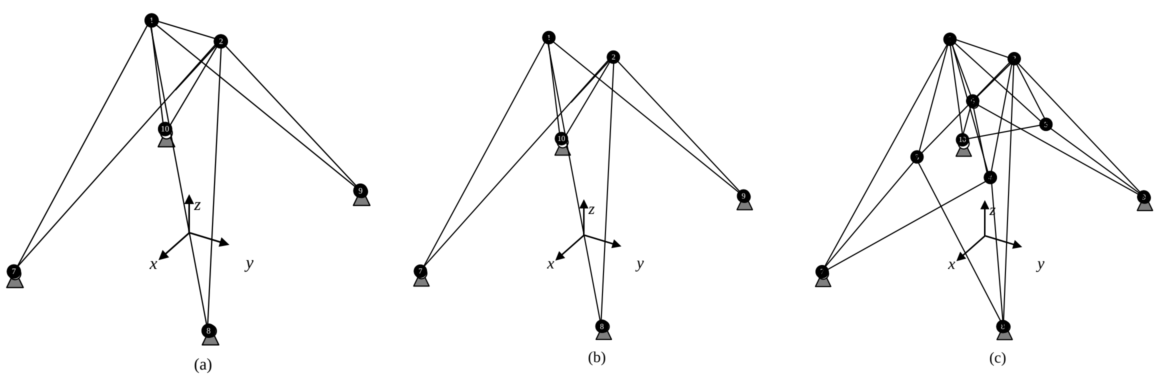



Figure 16. Results of Example 41.

Table 13. Effects of number of evaluating objective functions.

| $N_f$ | 50 |
|---|---|
| 1000 | 95.9335+33.2552 |
| 5000 | 59.7104+13.1297 |
| 10000 | 19.2668+10.2156 |
| 20000 | 12.1557+5.8889 |

With the same parameters, the proposed SCGA has run for 20 times and the best weights are recorded when the number of objective evaluation is up to the given numbers. The results are summarized in Table 13. When $N_f$ increases from 1000 to 20,000, the mean value of the best weight drops from 95.9335 lb to 12.1557 lb, while the standard deviation decreases from 33.2552 to 5.8889.

### *4.4.2 Example 42: Load 2*

In this section, two loads at each top node are increased from 500lb to 5000 lb.

With the default parameters, with $R_f = 0.95$ and $R_s = 0.05$ $N_{ms} = 300$, the proposed SCGA have been found 3 solutions, which are listed in Table 14 after 1000 generations. All solutions have different connections and are still better than the result obtained by Deb and Gulati (2001) (Noticing that their loadings are 500lb).

Table 14. Results of Example 42.

| Element No | Deb & Gulati (2001) | Proposed | | |
|---|---|---|---|---|
| | | Solution 1 | Solution 2 | Solution 3 |
| 1 | 0.16600 | | | |
| 2 | | | 0.00511 | 0.07990 |
| 3 | | | 0.23927 | |
| 4 | | | 0.07210 | 0.26800 |
| 5 | | | | 0.06750 |
| 6 | | | 0.00725 | |
| 7 | | | 0.35253 | |
| 10 | 0.40900 | 0.16409 | | 0.16682 |
| 11 | 0.07100 | 0.08893 | | 0.04959 |
| Weight (lb) | 46.68419 (47.93000) | 24.57757 | 31.99669 | 37.80225 |

Table 15. Effects of number of evaluating objective functions (Example 42).

| $N_f$ | 50 |
|---|---|
| 1000 | 1.1691E2+28.0974 |
| 5000 | 42.6398+8.2059 |
| 10000 | 42.6398+8.2059 |
| 20000 | 34.7847+6.8561 |

With the same parameters, the proposed SCGA has run for 20 times and the best weights are recorded when the number of objective evaluation is up to the given numbers. The results are summarized in Table 15. When $N_f$ increases from 1000 to 20,000, the mean value of the best weight drops from 116.91 lb to 34.7847 lb, while the standard deviation decreases from 28.0974 to 6.8561.

### *4.4.3 Example 43: Load 3*

Two loading conditions are applied in this case. In Case 1, a force vector (0, 20000, -5000) lb is applied on Node 1 and a force vector (0,-20000,-5000) lb is applied on Node 2. In Case 2, four vectors are applied: (1000,10000,-5000) on Node 1, (0, 10000,-5000) on Node 2 and (500,0,0) on Node 3 and Node 6.

Table 16 lists 5 results found with the proposed SCGA with the same parameters after 1000 generations. The best weight is 224.66666 lb.

Table 16. Results of Example 43.

| Element No. | Solution 1 | Solution 2 | Solution 3 | Solution 4 | Solution 5 |
|---|---|---|---|---|---|
| 1 | 0.55227 | 0.48337 | 1.30170 | 1.33709 | 0.57277 |
| 2 | 0.07919 | 0.00684 | 1.13942 | | 1.25613 |
| 3 | | 0.02029 | | 0.00545 | 0.31464 |
| 4 | | | | | 0.31148 |
| 5 | 0.04090 | 0.04069 | 0.18467 | | |
| 6 | | | 0.10137 | 0.04305 | |
| 7 | 0.09024 | | 1.04358 | | 1.44984 |
| 8 | 0.01154 | 0.01176 | | 0.07550 | |
| 9 | | | | | 0.00868 |
| 10 | 1.00820 | 1.01622 | 0.98331 | 1.18062 | 0.79572 |
| 11 | 1.08541 | 1.17183 | | 1.04378 | |
| Weight (lb) | 224.66666 | 226.24537 | 237.00101 | 239.22657 | 244.13667 |

With the same parameters, the proposed SCGA has run for 20 times and the best weight are recorded when the number of objective evaluation is up to the given numbers. The results are summarized in Table 17. The same observations can be obtained. The menu value drops from 390.38 lb to 234.55 lb and the standard deviation decreases from 68.3054 to 3.2043.

Table 17. Effects of number of evaluating objective functions (Example 43).

| $N_f$ | 50 |
|---|---|
| 1000 | 3.9038E2+68.3054 |
| 5000 | 2.6204E2+14.5454 |

| | |
|---|---|
| 10000 | 2.4234E2+7.3295 |
| 20000 | 2.3455E2+3.2043 |

## 5. Conclusions

In this paper, the species conserving genetic algorithm (SCGA) has been improved to solve truss topology optimization problems. In order to improve the quality of solutions, some techniques have been added into the SCGA. The mutation consists of the combination of a uniform mutation and a neighborhood mutation technique. The uniform mutation can explore globally in the whole search space, while the neighborhood mutations can explore near an individual. Species mutation technique is used to improve the fitness of the found species.

Due to lots of constraints in a truss optimization, only a small portion of the space bounded by the lower and upper limits of variables is within the feasible region. In order to improve the efficiency of the proposed algorithm, only individuals in the feasible region are included in the population. Therefore, the initializing process will be stopped when at least one feasible point has been found. A new individual generated by genetic operators, such as crossover and mutation, will be replaced with its parent when it is out of the feasible region.

Topology optimization and size optimization are solved simultaneously with the proposed SCGA by using a real-coded vector to present the cross-sectional area of elements of the ground structure as decision variables. A member will exist if its area is bigger than the critical area that is a very small positive number. Generally, the lower limits are less than the critical area.

In this paper, finite elements were used to both analyze the kinematic stability of structure and compute stresses and displacements. The developed FEA model can also identify if the stresses of existing elements are within the compressive and tensile strengths, and if the displacements of nodes satisfy the allowable limits.

The proposed SCGA has been used to solve a series of truss topology optimization problems. In each case, the proposed SCGA has been run for 20 times and the averages of the best weights over a series of given objective evaluations are summarized for future researchers to develop more efficient algorithms in solving truss topology optimization problems. The numerical results show that the proposed algorithm can find in one run multiple solutions including different topology trusses and have obtained better or similar solutions than the results obtained with existing topology evolutionary algorithms. In the future, the proposal SCGA will be used to solve large and more challenge truss optimization problems.

The results in this paper also demonstrate that when an engineering optimization problem is very complex or there is no knowledge on the number of solutions, species conserving genetic algorithms (SCGA) is a tool to explore all possible global solutions in the design space.

## References

Achtziger W. 2007. "On simultaneous optimization of truss geometry and topology", *Structural and Multidisciplinary Optimization*, 33: 285–304.

Deb K., R.B. Agrawal. 1995. “Simulated binary crossover for continuous search space”, Complex Systems. 9: 115-148.

Deb K. and S. Gulati. 2001. “Design of truss-structures for minimum weight using genetic algorithms”, *Finite Elements in Analysis and Design*, (37): 447-465.

Eschenauer H. A. 2001. "Topology optimization of continuum structures: A review", *Applied Mechanics Reviews*, (54) 4. DOI: 10.1115/1.1388075.

Fourie P.C. and A.A. Groenwold, 2002. "The particle swarm optimization algorithm in size and shape optimization", *Structure Multidisciplinary Optimization*, 23: 259–267.

Grieson D. E. and W. H. Pak. 1993. “Optimal sizing, geometrical and topological design using a genetic algorithm *Structural optimization* , 6: 151-159.

Hajela P. and E. Lee, 1995. "Genetic algorithms in truss topological optimization", *International Journal of Solids and Structures*, (32) 22: 3341-3357.

Imai K. and L.A. Schmith. 1981. “Configuration optimization of trusses”, *Journal of the Structural Division*, (107)5: 745-756.

Kaveh A., B. Farhmand Azar and S. Talatahari, 2008. "Ant colony optimization for design of space trusses", *International Journal of Space Structures*, (23)3: 167-181.

Kaveh A. and V. Kalatjari. 2003. "Topology optimization of trusses using genetic algorithm, force method and graph theory", *International Journal for Numerical Methods in Engineering*, 58: 771–791.

Kawamura H., H. Ohmori and N. Kito. 2002. "Truss topology optimization by a modified genetic algorithm", *Structure Multidisciplinary Optimization*. 23: 467–472.

Kicinger R., T. Arciszewski, K. De Jong. 2005. "Evolutionary computation and structural design: A survey of the state-of-the-art", *Computers and Structures*, (83): 1943–1978.

Krisch U. 1989. "Optimal topology optimization of trusses", *Method Appl. Mech. Eng.*, (72): 15-28.

Lamberti L., 2008. "An efficient simulated annealing algorithm for design optimization of truss structures", *Computers and Structures*, 86: 1936–1953.

Li J.-P., M.E. Balazs, G.T. Parks and P.J. Clarkson. 2002. "A genetic algorithm using species conservation for multimodal function optimization", *Evolutionary Computation*, 10(3): 207-234.

Li J.-P., F. Campean and A. Wood. 2010. "Reliability-Inspired Species Conservation for Multimodal Functions", *10th Annual Workshop on Computational Intelligence (UKCI2010)*.

Luh G.-Ch. and Ch.Y. Lin, 2008. "Optimal design of truss structures using ant algorithm", *Structure Multidisciplinary Optimization*, 36: 365–379.

Ohsaki M. 1995. "Genetic algorithm for topology optimization of trusses", *Computers & Structures*, (57)2: 219-225.

Ohsaki M. 1998. "Simultaneous optimization of topology and geometry of a regular plane truss". *Computers and Structures*. 66(1): 69–77.

Ringertz U. T. 1985. "On topology optimization of trusses", *Engineering Optimization*, (9): 209-218.

Ringertz U. T. 1986. "A branch and bound algorithm for topology optimization of truss structures", *Engineering Optimization*, (10): 111-124.

Rozvany G. I. N. 1996. "Difficulties in truss topology optimization with stress, local buckling and system stability constraints", *Structural Optimization*, 11: 213-217.

Sonmez M., 2011. "Artificial bee colony algorithm for optimization of truss structures", *Applied Soft Computing*, 11: 2406–2418.

Su R.-Y., L.-J.Gui, and Z,-J. Fan. 2009. "Truss topology optimization using genetic algorithm with individual identification technique", Proceedings of the World Congress on Engineering, vol. II, WCE 2009, London, U.K.

Tang W.-Y., L.-Y. Tong and Y.-X. Gu, 2005. "Improved genetic algorithm for design optimization of truss structures with sizing, shape and topology variables", *International Journal for Numerical Methods in Engineering*, 62: 1737–1762.

Tuma J. J. and Ronald A. Walsh, 1997. *Engineering Mathematics Handbook*, McGraw-Hill Prof Med/Tech.

Wang D., W. H. Zhang and J. S. Jiang. 2002. "Combined shape and sizing optimization of truss structures", *Computational Mechanics*, (29) 4-5: 307-312.