Discrete Planar Truss Optimization by Node Position Variation Using Grammatical Evolution

Michael Fenton, Ciaran McNally, Jonathan Byrne, Erik Hemberg, James McDermott, and Michael O'Neill

***Abstract*—The majority of existing discrete truss optimization methods focus primarily on optimizing global truss topology using a ground structure approach, in which all possible node and beam locations are specified *a priori*. The ground structure discrete optimization method has been shown to be restrictive as it limits derivable solutions to what is explicitly defined. Greater representational freedom can improve performance. In this paper, grammatical evolution is applied. It can represent a variable number of nodes and their locations on a continuum. A novel method of connecting evolved nodes using a Delaunay triangulation algorithm shows that fully triangulated, kinematically stable structures can be generated. Discrete beam-truss structures can be optimized without the need for any information about the desired form of the solution other than the design envelope. Our technique is compared to existing discrete optimization techniques, and notable savings in structure self-weight are demonstrated. In particular, our new method can produce results superior to those reported in the literature in cases in which the problem is ill-defined and the structure of the solution is not known *a priori*.**

***Index Terms*—Civil engineering, computational intelligence, evolutionary computation, genetic algorithms, grammatical evolution (GE), structural engineering.**

I. Introduction

CONTINUUM topology optimization in design (TOD) methods represent the current cutting edge in engineering design optimization [1]–[3]. Continuum TOD is similar in principle to the finite element method of structural analysis [4] in that the system is assumed to be continuous and as such can be discretized into smaller elements. Optimization of individual elements can then be extrapolated to account for the overall design [5], [6]. The advantage of this approach is that it negates the need to know any information about the desired solution, other than the boundary conditions (i.e., loads and reactions). A generalized description of the continuum approach can be seen as a material distribution problem within a design envelope upon which loading and reaction conditions are imposed. Material is then added, removed, or rearranged from within the design space as needed (subject to a specified volume fraction of the original space) in the search for the optimum material layout such that the overall compliance (tendency to deflect) of the structure is minimized [7].

In large-scale civil and structural engineering projects, however, manufacturing solid structures fully optimized using these techniques is generally prohibitively expensive and difficult [1], [8]. While the continuum topology optimization approach has been repeatedly proven to be computationally and structurally more efficient than heuristic forms of discrete truss design [8], [9], it is implicitly cost ineffective to manufacture as nonstandard elements, forms, and construction methods are required [1]. Furthermore, the computational cost of generating a solution increases exponentially with physical size of the structure [8].

Discrete beam structure optimization methods [10] are currently more appropriate for large-scale designs, as they allow regular elements and construction methods to be used, leading to savings in cost and weight over more traditional construction methods [1].

Manuscript received July 8, 2015; revised October 12, 2015; accepted November 18, 2015. Date of publication November 23, 2015; date of current version July 26, 2016. This work was supported in part by the Science Foundation Ireland under Grant 13/IA/1850, and in part by the Graduate Research Education Programme in Sustainable Development through Irish Research Council for Science, Engineering, and Technology and Irish Research Council for Humanities and Social Sciences.

M. Fenton is with the Natural Computing Research and Applications Group, University College Dublin, Dublin, Ireland (e-mail: michael.fenton@ucd.ie).

C. McNally and J. Byrne are with the School of Civil, Structural, and Environmental Engineering, University College Dublin, Dublin, Ireland (e-mail: ciaran.mcnally@ucd.ie; jonathan.byrne@ucd.ie).

E. Hemberg is with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: erik.hemberg@gmail.com).

J. McDermott is with the Lochlann Quinn School of Business, University College Dublin, Dublin, Ireland (e-mail: james.mcdermott2@ucd.ie).

M. O'Neill is with the Natural Computing Research and Applications Group and the College of Business, University College Dublin, Dublin, Ireland (e-mail: m.oneill@ucd.ie).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2015.2502841

II. Motivation

Classical discrete topology optimization follows a ground structure approach, with all possible node and beam locations being specified *a priori* and the algorithm selecting the most appropriate configuration from the given list of options [10]. There are limitations in the techniques employed in discrete optimization, however.

Luh and Lin [11] made particular note of the fact that the most optimal solutions for discrete methods can only be found by simultaneously considering optimization of member sizing, structural shape, and overall topology, as each has an effect on the others. Previous work by Fenton *et al.* [12] has demonstrated not only that simultaneous evolution of structural shape, structural topology, and member sizing is possible, but also that improvements over traditional ground structure methods could be achieved.

Rozvany [13] explicitly stated that globally optimal solutions for discrete optimization problems cannot be found with enforced layouts and a small number of members, i.e., the representation with traditional ground structure approaches is too constrained to effectively find the global optimum of the overall search space. Deb and Gulati [14] confirmed this by showing that even limited variation of node locations in traditional heuristic ground structure approach can lead to improved fitness results. Further unconstrained discrete representations from the literature such as the principal stress line method provide even better results [13], but can only be used specifically with compliance minimization problems [15]. It therefore follows that a minimally constrained discrete optimization method that can be used with any fitness function would be a most useful tool.

Continuum optimization methods can be viewed as being unconstrained in that they require no information about the potential form of the solution. Their only appreciable constraint is in the form of the design envelope [9]. The goal of this paper, therefore, is to replicate the continuum optimization method using discrete optimization techniques. In order to develop such a method, a number of key questions must be asked, as follows.

1) Is it possible to create a minimally constrained discrete representation, given minimal information on the desired solution?
2) Will such an unconstrained representation be able to evolve viable solutions?
3) Are better solutions achievable with well-defined or ill-defined information about the problem?

If even a slight variation in the location of nodes can lead to improvements in fitness, a question must be asked over whether or not methods that use fixed representations can really be considered the most appropriate techniques for discrete optimization. The idea presented here is to effectively invert the traditional process of selecting the arrangement of edges connecting preexisting nodes by selecting and indicating the presence and location of the actual nodes themselves. If these nodes were then to be connected using a simple and repeatable method such that the emphasis was on the nodal locations themselves rather than the interconnectivity between them, would this demonstrate any advantages or disadvantages over the traditional connection-focused ground structure method. This is the central hypothesis that this paper aims to examine.

The remainder of this paper is structured as follows. Section III provides an introduction to the topics discussed in the paper. Section III-A describes the importance of the design envelope in both discrete and continuum optimization. The approach to node generation within this design envelope is discussed in Section III-B, including a number of different methods of describing node locations using a grammatical representation. The method of connecting nodes is discussed in Section III-C. A number of benchmark numerical examples from the literature are detailed in Section IV, including proof-of-concept examples (Section IV-A) and nonregular truss forms (Section IV-D). The implications of these experiments are discussed in Section V and the conclusion is drawn in Section VI.

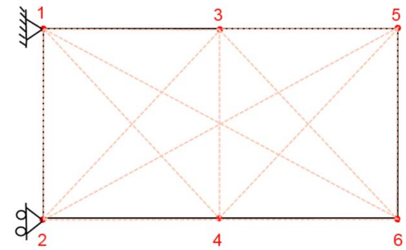


Fig. 1. Traditional 16-member, 6-node ground structure optimization approach [13]. Red lines represent possible edge locations.

III. Problem Definition

A major drawback of classical discrete ground structure optimization approaches is that the entire gamut of possible solutions needs to be specified before evolution begins. The evolutionary process merely selects the most appropriate arrangement from a combination of predefined elements. While this has repeatedly been proven to produce good results [1], [8], [10], it has been noted that this method inherently limits the range and quality of possible solutions due to the fact that potential solutions can only be derived from the given elements and layouts [8]. This set of all potential solutions capable of being represented by the algorithm is typically called the representation space. This representation space is only a very small subset of the overall wider search space [16]. The search space itself encompasses the full set of all possible solutions to the problem (including all those that cannot be generated by the algorithm), regardless of suitability or optimality [16].

Standard practice for ground structure optimization follows a binary-style approach whereby the full set of possible solutions is specified beforehand with the algorithm adding or removing preexisting elements [10] (Fig. 1). However, this limits the representation space to only what is explicitly defined. Potential feasible solutions elsewhere within the search space that lie outside this definition are never considered. A larger representation space can therefore cover a greater proportion of the overall search space, potentially leading to a greater possibility of containing the true global optimum. However, a larger representation space can potentially make the search process more difficult, as there are more candidate solutions to explore [16].

Another important omission to note is that traditional 2-D ground structure optimization methods roundly ignore the fact that coplanar members that intersect by definition create an extra nodal connection. Along with idealized material specifications (including unrealistic stress limits and cross-sectional area optimization that exceeds manufacturing precision capabilities, as discussed in previous work by Fenton *et al.* [12]), there is an apparent assumption that solid members can pass through one another with no structural effect.

Classical ground structure optimization problems, such as the 16-member structure shown in Fig. 1 or the 10-bar truss problem shown in Fig. 2, are therefore unrealistic. In reality,

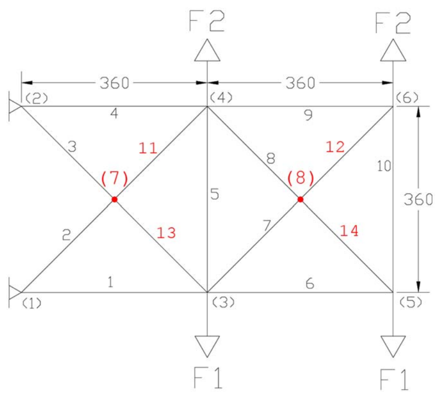


Fig. 2. Ten-bar truss problem [9], [14], [15] can be more accurately described as a 14-bar, 8-node truss problem. Extra nodes and edges are indicated in red.

the 6-node, 10-bar truss from Fig. 2 should be more accurately described as an 8-node, 14-bar truss, with additional nodes 7 and 8, and additional members 11–14 highlighted in red.

This 14-bar truss has completely different characteristics from the 10-bar example, as the four longest members are now effectively halved in length. This has a notable effect on buckling of compression members, as these long diagonal members are now braced at their mid-points. In order to generate and optimize realistic structures, every intersection of multiple members should be treated as a nodal connection, splitting all connected members into shorter members.

A. Design Envelope

The problem definition in continuum optimization generally begins with the definition of the environment in which the solution is evolved: the design envelope. This envelope normally defines the maximum and minimum boundaries in all dimensions for the location of any element of the design. Elements (be they nodes, edges, or design material) that lie outside the design envelope are not considered, and indeed in most cases are not permitted to be generated. Discrete optimization does not technically have a similar concept, but the design envelope could be said to be described by the outermost node locations of the structure [15]. In order to replicate the continuum method using discrete elements, the design envelope must incorporate limits for all possible element locations. These limits can be set by either penalizing designs that fall outside the limits (i.e., constraining designs using the fitness function) or only allowing elements to be placed within (or on the boundary of) the limits (i.e., constraining designs in the representation).

Rozvany [8] provided analytical, theoretically correct solutions to a series of benchmark problems for continuum sections, based on [17]. Therein, it was noted that as its volume fraction approaches zero, the structure of an optimized stressed shell plate tends toward that of a truss. Using this correlation, it is therefore possible to derive optimal 2-D truss topologies from stressed plate structures. Indeed, previous research by Rozvany *et al.* [18] found that a globally optimal structure could be found purely from a given design envelope.

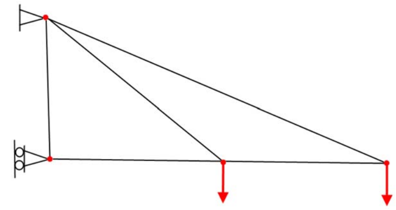


Fig. 3. Stable dual load cantilever truss configuration, using only essential nodes (no nonessential nodes present).

Kawamura *et al.* [19] noted that, as with observable nature, the environment has a significant effect on the evolution of individuals. They argue that any population placed in a well-defined environment should evolve successfully to suit it. While this is not necessarily the case for methods such as developmental systems (which can be used to generate solutions that are reactant to their environment [20], [21]), this paper only focuses on static solutions.

It must be noted that the use of a design envelope necessarily limits the reach of the representation space, thereby limiting the amount of the search space that can be traversed. However, limiting the placement of elements in undesirable locations can overall be helpful to the search process [20]. Furthermore, a design envelope is often an engineering necessity where engineers have hard limits on space.

B. Node Generation

The hypothesis of this paper is that a node location-based discrete approach will be capable of generating fitter solutions than a fixed-node connectivity-based discrete approach given only boundary conditions. A representation therefore must be built with the capability of placing nodes anywhere within the design envelope. Deb and Gulati [14] broke down the node characteristics of an individual into two categories: 1) essential nodes and 2) nonessential nodes (they use the phrases basic and optional). Essential nodes are those nodes that are necessary for the problem definition—all fixed and loaded points in the structure. Any other nodes are viewed as nonessential (i.e., a theoretical solution to a problem can be found using only essential nodes, as shown in Fig. 3).

To use this definition, all fixed and loaded nodes must first be defined as essential. These are the only fundamentally necessary nodes in the structure. If any other nodes are needed they can be subsequently evolved.

Loads can be applied to nodes as a vector of force magnitudes (with x and y directions), allowing for a point load of any size and direction to be placed on any node in the structure. For example, a vertical downward force of 100 N would be described as `[0, -100]`. In this manner, it is possible to specify any point load in any direction for any load, simply by specifying the node location and force vector.

In a similar manner to point loading, fixed-node locations can be indicated by a tuple of Boolean values, indicating fixing in the x- and y-directions. For example, a fully pinned support (i.e., fixed in all directions and free to rotate in any direction) would be indicated by `[True, True]`, while a pinned support with a rolling bearing on the x-axis would

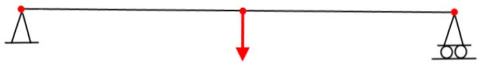


Fig. 4. Kinematically unstable structure resultant from colinear essential nodes.

```
<nodes> ::= <node> | <node>,<nodes>
<node>::= [<x>, <y>]
```

Fig. 5. Simple recursive grammar segment showing potential for multiple node selection through the production rule <nodes>.

```
<node> ::= (<x>, <y>)
<x> ::= <s>/100 * <i_node>[0]
<y> ::= <d>/100 * <i_node>[1]
<i_node> ::= (<percent>, <percent>)
<percent> ::= <n><n>.<n><n>
<n> ::= 0|1|2|3|4|5|6|7|8|9
<s> ::= 18288
<d> ::= 9144
```

Fig. 6. Grammar excerpt showing percentage-based node generation.

be indicated as `[False, True]` (i.e., free to rotate in any dimension and fixed on the *y*-axis, but free to move on the *x*-axis). This methodology can be easily extended to account for other fixing conditions (such as fully fixed or hinged connections), but pinned connections are only considered in this paper. All other nodes in the structure (i.e., those nodes which are neither loaded nor providing reaction support) can therefore be deemed to be nonessential, with one caveat: if all essential nodes were to be colinear (that is, if all essential nodes were to exist on the same line), it would be possible to generate nonoptimal, kinematically unstable 1-D structures (as shown in Fig. 4), since all nodes in the structures are pinned connections and are free to rotate.

Consequently, if all essential nodes are colinear then at least one nonessential node must be specified, which is not colinear with all essential nodes. This ensures that all generated structures are at least 2-D and, as such, are more likely to be kinematically stable (actual fitness notwithstanding).

1) Grammatical Evolution: Grammatical evolution (GE) [22] is employed in this application for a number of reasons.

1) It has been applied successfully to a wide range of evolutionary design and engineering problems [23]–[29].
2) Its recursive capabilities can be employed to allow the grammar to select however many nodes it requires.
3) Bias can be instilled in the grammar to preference particular outcomes.
4) Previous work by the authors in this area, dual optimization in GE (DO-GE) [12], means a framework upon which this paper can be built is well established.

Once essential nodes are defined, the grammar selects any number of nonessential nodes for the interior of the structure. This is achieved through a simple recursive grammar technique, as detailed in the grammar extract in Fig. 5.

The grammar next needs to be able to indicate the actual locations for potential nonessential nodes. Specifically, the grammar needs to be able to select from within a range of numbers (i.e., the boundaries of the design envelope) in such a way that it can effectively cover the entire design envelope. A percentage-based node generation method was chosen as it allows for easy changes to the shape of the design envelope with minimal changes to the grammar. The chosen node generation approach defines nodal positions as a percentage of the overall dimensions of the structure, selecting a percentage of a given maximum value.

The grammar specifies both the *x*- and *y*-values of the node as a percentage _ _._ _% of the maximum permissible value for that dimension (the span <s> or the depth <d> of the structure). The basic grammar excerpt as shown in Fig. 6 illustrates the principle, with span and depth taken from the dimensions of the cantilevered truss as shown in Fig. 2.

A modifier, defined in (1), is employed to translate percentage values (nonterminals <x> and <y>) from the range [00.00%, 99.99%] to [00.00%, 100.00%]. This allows the grammar to cover 100% of the design envelope, with an equal bias for all percentage values. Furthermore, this modifier translates the percentage value from the grammar into a nodal coordinate within the design envelope

$$x' = \frac{\text{span}}{99.99} * x. \tag{1}$$

This percentage-based grammar allows for nodes to be placed in any location within the design envelope. More importantly, it exhibits high locality—the coordinates for each node are well defined and the relatively high number of codons required for each coordinate allows for fine-grained mutation and crossover control. This itself allows for local hillclimbing in the search space.

2) Grammatical Bias and Regularity of Structures: One of the major benefits of GE use of a formal grammar structure in defining its individuals is the ability to add bias to the grammar [22]. This allows the user to specify preference toward any particular outcome. For example, the percentage-based grammar described in Fig. 6 is biased toward a uniform distribution of nodes within the design envelope; it is equally likely that nodes can appear anywhere within the structure.

Bias can be added or changed in this grammar by adding extra options to the node generation process. The nonterminal <percent> from Fig. 6 shows only a single terminal option; it will always return a value in the range [00.00%, 99.99%], with an equal probability for all options. This rule could be changed to add bias toward different outcomes

```
<percent> ::= 0 | <n><n>.<n><n> | 99.99.
```

This would bias the grammar toward placing nodes at the boundaries of the design envelope. There is an equal one in three chance that a node will be placed at any of the three options.

With certain well-defined structural optimization problems, the forms of the solutions are well known, as they have been exhaustively covered by the literature. In these cases, this domain knowledge can be used to write a grammar in such a way that it is biased toward a particular set of solutions.

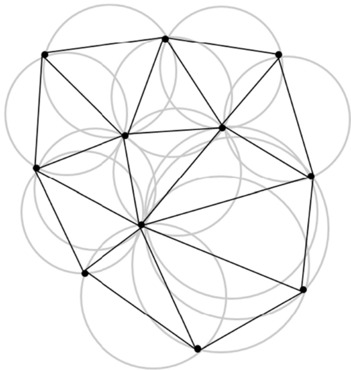


Fig. 7. Delaunay triangulation [33].

C. Node Connection

The recursive capabilities of GE [30]–[32] allow for variations in the number of nodes between individuals. By passing the list of nodes through a Delaunay triangulation algorithm [33], connections between nodes (edges) are defined. Delaunay triangulation operates by triangulating a set of points in a plane such that no point lies within the circum-circle of any triangle (as shown in Fig. 7). Kawamura *et al.* [19] noted that representation of truss structures by combinations of triangles led exclusively to statically determinate, kinematically stable structures with a high degree of optimality, being able to reduce (or even eliminate entirely) the number of unnecessary members. By limiting the representation space exclusively to structures composed of triangles, the evolutionary process is exposed only to those areas of the search space that contain solutions guaranteed to be stable. This serves to focus the evolutionary search process toward desirable solutions.

Although Delaunay triangulation is most widely used in surface mesh generation [34], [35], it has been used in other applications. Notably, Joachimczak and Wróbel [36] used Delaunay triangulation in order to evolve developmental systems to describe the internal structure of multicellular organisms. While nodal locations were not themselves evolved, the resultant layouts were connected via Delaunay triangulation, from which a Gabriel graph [37] was generated, defining the ultimate solution. Although Gabriel graphs can produce cleaner-looking structures [36], their ability to generate elements of more than three sides could lead to an increased possibility of kinematically unstable structures. Since triangulation is guaranteed to result in a statically determinate and kinematically stable structure in pin-jointed planar trusses [4], [19], Gabriel graphs are not used here. A structure is kinematically stable if [4]

$$r + m - 2n \geq 0 \tag{2}$$

where

- r number of reactions;
- m number of members;
- n number of nodes.

It must be acknowledged that the reliance on triangulation to generate nodal connections limits the representation space. In certain cases, it is possible that this could render the global optimum unattainable. For this reason, the ability of the DO-GE method [12] to delete unstressed members from its solutions is retained. Though not an evolutionary feature, it nevertheless allows for the production of irregular quadrilateral shapes that might otherwise be unattainable. While this does not entirely remove the limitations imposed by triangulation, it does mean that the representation space itself can be expanded in certain circumstances to incorporate solutions not otherwise attainable by the grammar. The deletion of unstressed members can only be allowed to occur if the following two conditions are met.

1) Equation (2) must be satisfied.
2) All nodes must have at least two edges attached.

DO-GE introduced the concept of structural evolution using two independent chromosomes to define separate aspects of the structure itself. The first, chromosome A (Ch.A), defines the topological form, or shape, of the structure. This is done by using the genes in Ch.A to map to the locations of any number of nodes within the design envelope, or to select more nodes to add to the structure. Once connections between nodes are made, the second chromosome (Ch.B) is used to set member sizings from a predefined list of available material options. In this way, evolution of the structural topology and shape, and the sizing of individual members in tandem, are possible.

IV. Truss Optimization Examples

Continuum topology optimization deals with minimization of compliance, given a specified weight (a volume fraction of the original design space) [9]. Ground structure (beam) optimization could be considered the opposite, as it generally deals with minimization of self-weight, given specified deflection limit (deflection being linearly related to compliance) [8]. Since specifying a volume fraction or weight using a ground structure approach is not generally possible, the objective of these experiments is the minimization of structure self-weight, given prespecified deflection limits. This new discrete optimization method, with grammar-based node positioning and Delaunay triangulation methods described earlier, is hereafter given the name structural engineering optimization in GE (SEOIGE).

In this section, a number of experiments are carried out to demonstrate the capabilities of the SEOIGE technique. These experiments are divided into three categories.

1) *Proof of Concept:* Simply supported truss example.
2) *Benchmark Tests:* Regular truss forms and problems.
3) Irregular truss forms and problems.

All experiments use the same evolutionary computation settings, as outlined in Table I.

Solutions are evaluated using the free open source finite element analysis program San Le's free finite element analysis [38]. In all figures showing solutions, shades of blue indicate compression and shades of red indicate tension. The fitness value of an individual is determined as the self-weight of that individual. Constraints on stress, deflection, and buckling are applied to all components of solutions in accordance with building codes of practice [12], [39], [40]. Individuals

TABLE I
EVOLUTIONARY PARAMETER SETTINGS AS USED IN ALL EXPERIMENTS

Initialization:	Random
Number of Runs:	100
Population Size:	1000
Number of Generations:	100
Selection:	Tournament
Tournament Size:	1% of population
Replacement:	Generational with elites
Elite size:	1% of population
Crossover Type:	Single Point
Crossover Probability:	75%
Mutation Type:	Integer
Mutation Probability:	1%

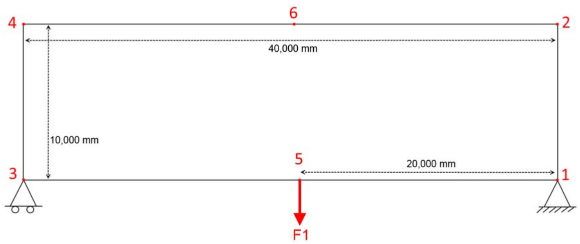


Fig. 8. Design envelope for simply supported truss.

TABLE II
BASIC NODES FOR SIMPLY SUPPORTED TRUSS

Node Index	Node Location	Node Label
1	(20,000, 0)	Pinned Support
2	(20,000, 10,000)	Design node
3	(-20,000, 0)	Rolling support
4	(-20,000, 10,000)	Design node
5	(0, 0)	Loaded (-1,000,000 N)
6	(0, 10,000)	Design node

that fail any constraints have penalty values added to their fitness equal to the sum of all normalized constraint failures. All experiments are compared with the best solutions from the literature, along with solutions evolved by DO-GE [12].

A. *Proof of Concept: Simply Supported Truss*

The first test conducted using the SEOIGE method was a proof-of-concept experiment, with the objective of evolving an arbitrary simply supported truss for minimal self-weight. The ultimate goal of this experiment was to validate the hypothesis of this paper: that it is possible to evolve a viable solution given only a design envelope.

A generic simply supported truss design envelope was created, with a total span of 40 m, a height of 10 m, and a single vertical force acting at the center span (as shown in Fig. 8). Support conditions are pinned roller. In accordance with the findings of Kicinger *et al.* [41], which state that the most effective and efficient method of designing large structures is through the use of symmetry, a simply supported truss structure can be achieved by merely mirroring a cantilevered truss about its central axis. Thus, the method only needs to evolve half of the structure.

Specified essential nodes (including load and support nodes) are outlined in Table II.

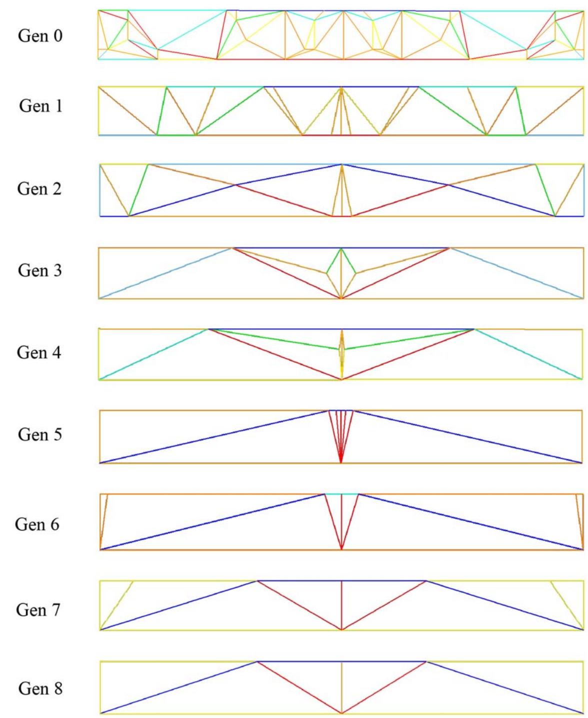


Fig. 9. Evolution of a simply supported truss structure.

Fig. 9 shows a number of elite solutions taken from the initial generations of a single evolutionary run. A clear progression in the evolution of the fittest solutions is visible from generations 0 to 8. Initial generations begin with a random arrangement of internal nodes within the structure. As generations progress, this nodal arrangement becomes less and less random as evolution drives the placement of nodes. This problem in particular is an interesting case in point, as accepted solutions are well established [4], [42] and the problem can be seen as a very basic benchmark example. The eventual evolved solution can clearly be recognized as an efficient simply supported truss structure for a vertical load placed at the midpoint.

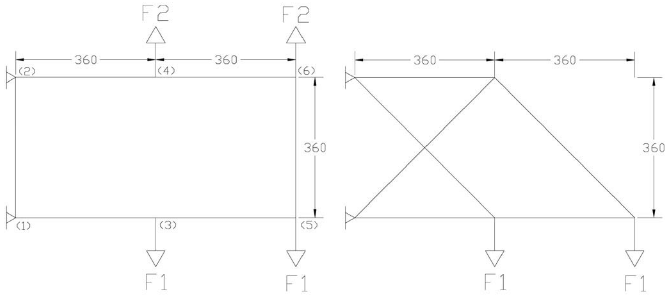
This experiment demonstrates that the SEOIGE method is indeed capable of generating viable solutions given only minimal knowledge of the problem itself. The following sets of experiments take this knowledge and compare solutions generated by SEOIGE for benchmark problems against those from the literature.

B. *Ten-Bar Cantilever Truss*

A cantilevered truss grammar was created to match the dimensions and loads of the 10-bar cantilevered truss example from Fig. 2. A design envelope was generated by setting the maximum node dimensions to a span of 18 288 mm (720 in) and a depth of 9144 mm (360 in) [as shown in Fig. 10(a)]. Maximum deflection limits were set at 50.8 mm (2 in) in both cases.

Two load cases were tested.

1) $F1 = 444\,800$ N (100 kips)
   $F2 = 0$.
2) $F1 = 667\,200$ N (150 kips)
   $F2 = 222\,400$ N (50 kips).



(a) (b)

Fig. 10. Ten-bar cantilevered truss example, dimensions shown in inches. (a) Design envelope. (b) Load case 1 evolved optimal topology from [14] and [43].

TABLE III
MATERIAL PROPERTIES

	10 Bar Truss	17 Bar Truss
	Aluminium solid sections	Steel solid sections
Section sizes	350 sections; CSA from 0.1 to 35 in^2, increments 0.1 in^2	
Young's modulus (ksi)	10000	30000
Density (lb/in^3)	0.100	0.268
Max tensile stress (ksi)	25	50
Max comp stress (ksi)	25	50

TABLE IV
ESSENTIAL NODES FOR 10-BAR CANTILEVERED TRUSS PROBLEM, LOAD CASE 1

Node Index	Node Location	Node Label
1	(0, 0)	Pinned Support
2	(0, 9144)	Pinned Support
3	(9144, 0)	Loaded (444,800 N)
5	(12288, 0)	Loaded (444,800 N)

All 10-bar truss examples from the literature used aluminum solid sections, while those of the 17-bar truss (presented in Section IV-C) used steel solid sections. Materials and section properties are described in Table III.

1) Load Case 1: For load case 1, only four essential nodes were defined (as shown in Table IV). No other information on the design of the desired solution was detailed in the grammar. Options for either the addition of a single node or multiple nodes were given equal bias of 50% each. Infeasible individuals (i.e., individuals that failed constraints) were left unrepaired. This problem was selected because, although there are many optimal solutions [11], [14], [43]–[46], there is considerable freedom in the design envelope and, thus, there is the potential for new solutions to be uncovered.

In terms of benchmark solutions against which to compare, Deb and Gulati [14], Hajela and Lee [43], Luh and Lin [11], and Ruiyi *et al.* [46] tackled the same problem as Li *et al.* [45], Lee and Geem [47], and Kaveh and Talatahari [44], but included a Boolean topological value in their genetic representation, allowing them to select the presence or absence of individual members. Their evolved optimal topologies [as shown in Fig. 10(b), needing only six bars as opposed to the original problem description of ten] resulted in greatly improved results over previous methods (although Luh and Lin [11] only show a figure of the optimal topology rather than giving their results).

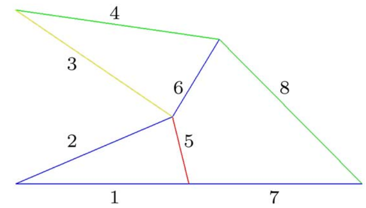


Fig. 11. Ten-bar cantilevered truss, load case 1—best SEOIGE solution.

TABLE V
NONESSENTIAL NODES DEFINED BY SEOIGE FOR 10-BAR TRUSS, LOAD CASE 1

Node 'X' Percentage	Node 'Y' Percentage	Actual Node Coordinate
58.77	83.11	(10749, 7600)
45.26	38.59	(8278, 3529)

The best topological solution as derived by SEOIGE is shown in Fig. 11. While there are still six nodes in the structure, there are now only eight bars, with a substantially different topological and overall shape layout to the original problem description from Fig. 2. For this solution, SEOIGE added two new nodes in addition to the four essential nodes listed in Table IV, as shown in Table V.

It must be noted that there are certain similarities between the evolved result in Fig. 11 and the topological solutions from the literature in Fig. 10(b). Superficially, there appears to be the same number of bars and general topological layout, although the overall shape is distorted somewhat. However, complete comparison of results from SEOIGE and the literature (particularly in terms of cross-sectional areas) cannot be made in this case, as the results from the literature contain only six bars, compared to the eight as shown in Fig. 11. Significantly, this implies that two of the bars from Fig. 10(b) overlap at their midpoints, creating an impossible structure. As discussed in Section III, any intersection of two or more truss members necessarily creates a nodal connection. Since the results from the literature ignore this point, accurate comparisons cannot be made beyond the comparison of overall structural weight.

When compared against previous results from the literature, it can be seen in Table VI that SEOIGE is capable of producing results that are significantly lower than previously published results. SEOIGE was able to reduce the best achieved result of DO-GE from 5056.88 to 4888.84 lb, 24.01 lb lighter than the best discrete solution from Deb and Gulati [14].

The evolved solution as shown in Fig. 11 is an interesting case. It can be seen that there is a quadrilateral shape in the best evolved solution, consisting of members 5–8. In this instance, a ninth structural member was originally included in the structure, consistent with the triangulated nature of node connectivity. However, this member was evolved with very low stresses in it, allowing for SEOIGE to delete it to improve the overall solution. It must be noted that the deletion of this

TABLE VI
TEN-BAR CANTILEVERED TRUSS: EVOLVED MINIMUM TRUSS WEIGHTS (lb)

	Su *et al.* [46]	Hajela & Lee [43]	Deb & Gulati [14]	Lee & Geem [47]	Li *et al.* [45]	Kaveh & Talatahari [44]	DO-GE [12]	SEOIGE
Load Case 1	4962.08	4942.7	4912.85	5057.88	5060.92	5056.56	5056.88	**4888.84**
Load Case 2	N/A	N/A	N/A	4668.81	4677.3	4675.78	**4612.8**	4624.35

TABLE VII
ESSENTIAL NODES FOR 10-BAR CANTILEVERED TRUSS PROBLEM, LOAD CASE 2

Node Index	Node Location	Node Label
1	(0, 0)	Pinned Support
2	(0, 9144)	Pinned Support
3	(9144, 0)	Loaded (667,200 N)
4	(9144, 9144)	Loaded (222,400 N)
5	(18288, 0)	Loaded (667,200 N)
6	(18288, 9144)	Loaded (222,400 N)

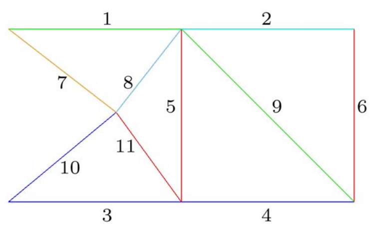


Fig. 12. Ten-bar cantilevered truss, load case 2—best SEOIGE solution.

TABLE VIII
NONESSENTIAL NODES DEFINED BY SEOIGE FOR 10-BAR TRUSS, LOAD CASE 2

Node 'X' Percentage	Node 'Y' Percentage	Actual Node Coordinate
31.15	51.79	(5697, 4736)

TABLE IX
CROSS-SECTIONAL AREAS FOR 10-BAR TRUSS, LOAD CASE 2, SEOIGE SOLUTION

A	1	2	3	4	5	6	7	8	9	10	11
B	15161	65	13032	9161	65	1290	9097	8581	12774	13032	4774

A: Element
B: Area (mm^2)

member leaves a structure that still complies with the rules stated in Section III-C, in that it remains kinematically stable.

2) Load Case 2: Load case 2 of the 10-bar cantilevered truss was not attempted by either Ruiyi *et al.* [46], Hajela and Lee [43], or Deb and Gulati [14], and the only example to be found in the literature of the application of a binary-style approach to this particular problem is that of Fenton *et al.* [12]. As such, the only comparisons that can be made are against those of Fenton *et al.* [12], Lee and Geem [47], Li *et al.* [45], and Kaveh and Talatahari [44]. This problem was chosen as it represents a highly constrained example to which there is only one predominant solution. With a generally fixed topological layout this problem is essentially a material sizing exercise and, thus, poses a difficult challenge for a method such as SEOIGE.

The addition of two more loads to the structure requires the addition of two more essential nodes, for a total of six essential nodes (as described in Table VII), the same number as with the original 10-bar truss model as shown in Fig. 2. All other variables and settings remained the same as the previous load case. The grammar was further changed over that of load case 1 in that an additional option was supplied to the production rule described in Fig. 5: that of generating no nodes at all. Equal 33.33% bias was given to all three options—no nonessential nodes, the generation of a single node, and the generation of multiple nodes (recursion).

The best evolved topology is shown in Fig. 12, with optimization results presented in Table VI. Interestingly, it can be seen that the member with the highest risk of buckling (member 2 from Fig. 2) has been broken into smaller segments to reduce the risk of buckling and to better dissipate the stresses. Although SEOIGE was given the option of adding no nonessential nodes, one extra node was added to this structure in addition to the six essential nodes listed in Table VII, as described in Table VIII.

A different topological layout to those of the literature also makes comparisons between evolved member cross-sectional areas of previous approaches difficult for load case 2. While similarities between Figs. 2 and 12 show both many common members, the original 6-node 10-bar truss problem should be more accurately described as an 8-node 14-bar truss, as detailed in Fig. 2. This sheds uncertainty over the results from the literature and, like load case 1, direct comparisons cannot be made beyond simple contrasting of overall structural weights.

Cross-sectional areas for all members are given in Table IX. It can be seen that members 2 and 5 are of the lowest available cross-sectional area. Examination of the analysis results shows that member 2 is actually unstressed, meaning that potential improvements can be made without that member. However, its deletion from the solution would create a kinematic mechanism whereby member 6 would be free to rotate about its base. For this reason, member 2 must be kept in the solution. Conversely, while member 5 is of the smallest available cross-section, it is a stressed member and its removal would over-stress member 11 such that it would no longer pass the imposed stress constraints.

Comparisons between SEOIGEs results and those of the literature show that although SEOIGE was able to achieve a better result than that of Lee and Geem [47] (besting their solution by 44.46 lb), it was still 11.55 lb off the best achieved result by DO-GE. The reasoning for this is that the number and location of the essential nodes for load case 2 severely

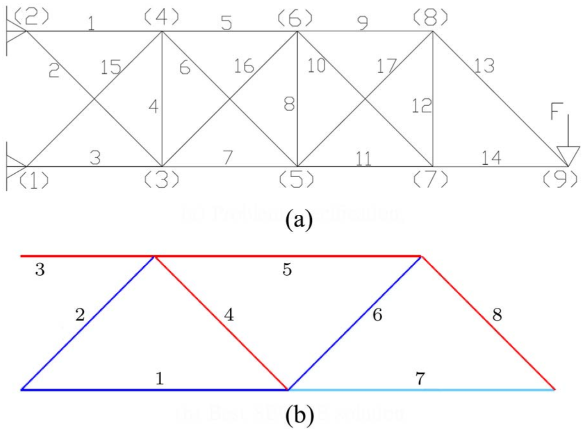


Fig. 13. Benchmark 17-bar truss problem. (a) Problem specification. (b) Best SEOIGE solution.

limits the scope of this problem for the SEOIGE method. Whereas SEOIGE performs well with ill-defined problems when there is little information given about the form of the solution (i.e., in load case 1), the load case 2 problem is well defined and there is consequently very little room for SEOIGE to evolve its own solution as the anticipated form of the solution is preset by the location of the essential nodes. Conversely, DO-GE utilizes topological optimization to obtain its superior results. However, the results of DO-GE are subject to the same flaws as the rest of the literature with regard to overlapping members. This discussion is expanded in Section V.

C. 17-Bar Cantilever Truss

Li *et al.* [45], Khot and Berke [48], Adeli and Kumar [49], and Fenton *et al.* [12] proposed solutions to a 17-bar cantilevered truss problem, as shown in Fig. 13(a). The design envelope for this problem measures 10 160 mm by 2540 mm (400 in by 100 in), with a vertical load of 444 800 N (100 kips) acting at the far end of the structure. Only three essential nodes need to be defined in this instance, with pinned supports at (0, 0) and (2540, 0), and a load of 444 800 N at (0, 10 160).

In this instance, both the shape of the design envelope and knowledge of prior successful solutions provide clues to the potential shape of the optimum solution. The grammar can therefore be biased accordingly such that SEOIGE has a greater chance of generating a fitter solution. In the case of the 17-bar truss problem, previously successful solutions [12], [45], [47]–[49] indicate that nodes placed at quarter points along the x-axis (i.e., at 25%, 50%, and 75% of the total span) will generate an optimum solution. These options were added to the grammar such that the production rule `<node>` had four options, as shown in Fig. 14.

It must be noted that this grammar, though biased toward placing nodes in regularized locations, still retains the capability of placing nodes at any location within the design envelope. In this manner, SEOIGE can determine the optimum topological layout, with hints at what that optimum layout is suspected to be. This problem was chosen, as with

```
<node> ::= 25 | 50 | 75 | <n><n>.<n><n>
<n> ::= 0|1|2|3|4|5|6|7|8|9
```

Fig. 14. Node generation production rule for SEOIGE grammar, 17-bar truss case.

TABLE X
17-Bar Cantilevered Truss: Evolved Minimum Truss Weights

	Khot & Berke [48]	Adeli & Kumar [49]	Li *et al.* [45]	DO-GE [12]	SEOIGE
Weight (lb)	**2581.9**	2594.4	**2581.9**	2595.4	**2581.9**

load case 1 of the 10-bar truss from Section IV-B1, as the design envelope is relatively unconstrained allowing for more freedom of choice in topological layout. However, accepted solutions [12], [45], [47]–[49] conform strongly to known topological layouts for problems of this nature [4], [8], [42]. This indicates that the more optimal areas of the representation space may potentially be constrained to solutions similar to those previously identified.

The best evolved topology for this problem can be seen in Fig. 13(b). What is interesting to note is that this solution is visually identical to the previous solution from DO-GE [12]. Even though the grammar retained the capability of placing nodes at any location, the most optimal node locations as determined by the grammar were at quartile points, as predicted.

The results of these experiments are shown in Table X. It can be seen that SEOIGE was only able to match the best achieved results from the literature (those of Li *et al.* [45] and Khot and Berke [48]), but was not able to improve upon them. However, the results from the literature (excluding those from [12]) suffer from the same intersecting members issue as previously described. SEOIGE was able to match their performance with a more accurately modeled structure. This is discussed more thoroughly in Section V.

D. Nonrectangular Truss Forms

Since the SEOIGE method has been validated and benchmarked against standard forms and tests from the literature in the above experiments, it is useful to examine its capacity to optimize nonstandard truss envelopes. Constraints on structural design will often be driven by physical dimensions and limitations such as space, budget, and services. Architects desire the freedom to design form over function, but are often limited by the nature and capacity of regular structural support mechanisms. Conversely, engineers are routinely pushed to design solutions for the ever-more-complex problems that arise from pushing the boundaries of shape and form. A design approach that could provide structural support solutions for arbitrary shapes and forms would therefore be of great value to both engineer and architect. This is a particular area where the SEOIGE method has the potential to excel.

For exterior design envelopes with nonrectangular shapes, it is possible to set limits on either the x- or y-values by using an

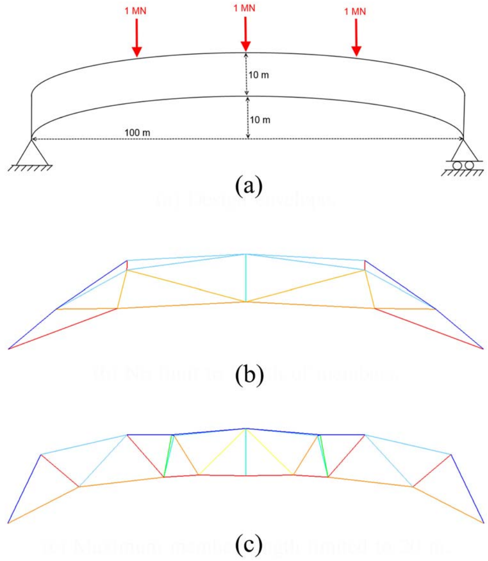


Fig. 15. Large span curved truss with high loading. (a) Design envelope. (b) No limit to length of members. (c) Maximum member length limited to 20 m.

equation that defines the boundary conditions in the form of

$$y = f(x). \tag{3}$$

In order to generate a structure based on a nonrectangular exterior design envelope, the SEOIGE method first selects node locations in the same manner as described in Section III-B. For the argument of the function described in (3), the modifier from (1) translates the percentage defined by the grammar into an actual location in the design envelope as before. However, in order to compute the output of the function described in (3), the modifier function from (1) is amended to account for the nonstandard nature of the exterior design envelope

$$y' = f(x) + \frac{\text{depth}}{99.99} * y. \tag{4}$$

In this manner, a design envelope for any exterior shape that can be defined by an equation can be described in such terms.

The drawback of this method is that it is not currently possible to specify complex internal structures such as voids within the design envelope, other than by penalizing solutions that are undesirable. Furthermore, while the method described above allows for the placing of nodes within a nonstandard design envelope, nodal connections that lie outside this envelope are capable of being generated. The SEOIGE method currently relies on postprocessing measures to remove any such connections, but techniques such as Gabriel graphs [36], [37] can be employed for these purposes in future applications. This area is earmarked for future research.

Fig. 15(a) shows a design envelope for a long-span curved truss, with a span of 100 m and three point loads of 1 000 000 N, at both quarter points and at midpoint.

Since the design envelope only limits the placement of nodes, elements connecting nodes within the design envelope may stray outside of it, as shown in the solution in Fig. 15(b). It is possible to include a maximum length limit for members in these structures. While this is not necessary (long members in tension pose no risk, while long members in compression have been evolved such that they are not at risk of buckling), the results are interesting.

Fig. 15(c) shows a sample solution for the same design envelope as Fig. 15(b), but with an imposed maximum member length limit of 20 m. Solutions that contain members with lengths above this limit are given a penalized fitness in a similar manner to all other failed constraints. It can be seen that the imposition of member length limits results in a solution that more closely aligns with the original nontrivial design envelope itself.

These experiments demonstrate the capacity of the SEOIGE method to generate viable solutions for problems in which there is little to no information known about the form of the solution other than the external acting forces, and the traditional methods of engineering design cannot readily be applied.

V. Discussion

A comparison of results from the various experiments performed in this paper yield interesting observations. It can be seen from the benchmark experiments in Sections IV-B and IV-C that, in all cases, the SEOIGE method was able to match or improve on the results from ground structure approaches in the literature. SEOIGE was also able to improve on the results evolved by previous methods from Fenton *et al.* [12] in two of the three benchmark tests attempted. For load case 1 of the 10-bar cantilevered truss benchmark problem, it can be seen that the SEOIGE solution represents an improvement over both DO-GE (by 168.04 lb) and the results found in the literature (improving on the results from [14] by 24.01 lb). For load case 2, SEOIGE was able to improve on the results from the literature (improving on the results from [47] by 44.46 lb), but remained 11.55 lb shy of the best achieved weight from DO-GE. While SEOIGE was able to improve on the best result of DO-GE for the 17-bar truss (by 13.5 lb), it was only able to match the best results from the literature (Li *et al.* [45] and Khot and Berke [48]) at 2581.9 lb.

SEOIGE can be seen to perform well where there is both a broad design envelope area and where little information is known about the optimal solution. In the instance of load case 1 for the 10-bar truss, there are four essential nodes and a design envelope where 1-D is twice the length of the other, creating a broad area over which new nodes can be placed (and therefore a broad area over which new structures can be evolved). Since the SEOIGE method operates by addition and subsequent connection of newly defined nodes above and beyond predefined essential nodes, there is plenty of scope for the addition of new nodes and structure. This is reinforced by the fact that there is little information given about the structure of potential solutions. An example of an unintuitive yet highly successful solution evolved using this method can be seen in Fig. 11; this solution is very different to the previously accepted solutions to the problem, as shown in Fig. 10(b).

However, the SEOIGE method does not perform quite as well when the problem is well defined (i.e., in cases with known solutions and little flexibility in the design space). Such is the case with both load case 2 for the 10-bar truss problem and the 17-bar cantilevered truss problem. Since SEOIGE operates by adding new nodes to form a structure, it performs best when it has control over not only the topological layout but more importantly the overall shape of the structure. With load case 2 for the 10-bar truss, the shape of the structure is predetermined by the fact that the essential nodes completely describe the boundary of the design envelope. SEOIGE therefore does not have the freedom it needs to define its own structural shape. Less flexible existing methods such as [12] are capable of doing so well in this instance that SEOIGE has no opportunity to outperform them.

The disparity between the performance of regular ground structure methods from the literature and the SEOIGE method can further be explained by the fact that SEOIGEs Delaunay triangulation approach to node connections is algorithmic in nature and lacks an appreciation for specific topological efficiency. The Delaunay triangulation method cannot specify which connections are required. With load case 2 of the 10-bar truss, there are six essential nodes defined. Since the overall shape cannot be changed (an area where SEOIGE excels), the only logical course of action in this case is to optimize the topology of the existing structure. This means that this particular problem is well suited to a traditional ground structure approach, such as those described by [14], [43], [45], and [46]. A method that varies the number and location of nodes, and automatically connects nodes in an algorithmic fashion, lacks the fine topological control that this particular problem requires and is consequently less effective.

A similar situation is seen with the 17-bar truss problem. In this instance, however, the extreme dimensions of the design envelope (with 1-D being four times greater than the other) coupled with the fact that the most appropriate solutions are known (structures with regular node locations and elements) means that again SEOIGEs options are limited. Although SEOIGE is always given the option of deciding its own node locations in the regular fashion described in Section III-B (and thus retaining its full representation space capabilities), the inclusion of the regular quartile node locations at 25%, 50%, 75%, and 100% of the total span allows the program to sample a wide range of potential solutions while hinting at the potential correct solution. This is an example of incorporation of domain knowledge into the grammar. The fact that SEOIGE evolves near-optimal solutions using only these suggested quartiles and not using any nodes generated using the regular percentage-based method suggests that the problem can be considered well defined in that the solution of the structure is generally known.

A particular note must be made about nodal connections with regard to the examined literature. As described in Section III, overlapping members necessarily create nodal connections in truss structures. All connections in traditional trusses are assumed to be pinned. As such, all members only carry axial stresses; no moment forces can be present in traditional truss members [4]. If members overlap at a certain point, a rigid connection will be created, which is capable of transferring moment forces. Any traditional truss structure with overlapping members must treat such overlaps as pinned nodal connections [4]. Failure to do so will create a truss structure that either contains internal moments (if rigid connections are assumed from overlapping members) or is in itself an impossible structure (if no connection between overlapping members is assumed). All literature examined in this regard made no mention of such overlapping members, but made a point of noting pinned truss connections. As such, the veracity of their results comes into question.

1) Note on 3-D Structures: While the scope of this paper only covers planar trusses, the SEOIGE method can readily be extended to 3-D surface structures. Delaunay triangulation is first and foremost a surface mesh generator, and as such its primary use is for discretizing surfaces, such as in graphical applications [35] or the finite element method of automated structural analysis [4], [34]. As such, it is currently only possible to use the presented method in three dimensions for generation of 3-D surface structures such as facades or exterior envelopes. While there are architectural methods for generating such surfaces [50], there is very little structural optimization literature in this area.

One current deficiency of the SEOIGE method is its inability to optimize 3-D volumetric structures. This is wholly the result of the authors' decision to use Delaunay triangulation to connect evolved nodes. This decision was made as fully triangulated pin-jointed planar structures are guaranteed to be both statically determinate and kinematically stable [4], [19]. This then ensures that the representation space covers only those areas of the overall search space that consist of stable structures. While the Delaunay method can technically be extended to 3-D for volumetric structures, resultant structures do not necessarily exhibit the same degree of optimality that similar planar/surface structures do. Unlike with the planar implementation of Delaunay triangulation, there is no guarantee that 3-D volumetric structures will be kinematically stable.

VI. Conclusion

Traditional ground-structure-based discrete topology optimization methods have been proven capable of deriving optimal solutions to benchmark problems [11]. However, the literature has suggested that the limited representation capabilities of ground structure methods limit their efficiency [8], [13], [14]. It was theorized that a method that could evolve the number and placement of nodes rather than the connectivity between fixed nodes would provide increased representation capabilities over traditional ground structure methods, thereby allowing for a wider search and consequently the potential for superior solutions. It was also theorized that minimal information about the structure of the solution would lead to better results.

By evolving the number and placement of nodes and generating nodal connections via a Delaunay triangulation algorithm, it was shown that the SEOIGE method presented in this paper is able to outperform various methods from the literature, producing equal or superior solutions for an array of 2-D benchmark problems. As such, the research questions posed in Section II have been successfully answered, and the central hypothesis of the paper was confirmed: placing more emphasis on nodal locations rather than nodal connectivity can lead to improved results for pin-jointed planar truss structures.

However, while the SEOIGE method can produce near-optimal solutions for problems in which the solution is not known, it can be seen that it works best when both the structure of the solution and the design envelope (and, more importantly, the shape of the potential solution in the form of the number and locations of essential nodes) are ill defined. The fewer essential nodes that exist (i.e., the looser the design brief), the better the solution SEOIGE can evolve. In cases in which the structural shape is already set and connectivity between preexisting nodes is paramount to finding the global optimum, SEOIGE has difficulty in improving upon existing solutions due to its lack of fine topological control over member connections.

An important point to note is that results from the literature were found to be misleading as all observed methods ignore the effects overlapping members would have on the overall structure. Accurate comparisons between the SEOIGE method presented in this paper and those methods from the literature are therefore severely limited.

Future work will focus on improving the existing method by examining different connectivity methods, as well as fully extending the SEOIGE method to 3-D. Paramount to this is the implementation of a connectivity method that provides similar benefits for 3-D structures in terms of kinematic stability guarantees to the current planar implementation. This is the subject of ongoing work by the authors.

Acknowledgment

The authors would like to thank the members of the Natural Computing Research and Applications Group, University College Dublin, for their invaluable help with this paper.

References

[1] R. Kicinger, T. Arciszewski, and K. A. De Jong, "Evolutionary computation and structural design: A survey of the state-of-the-art," *Comput. Struct.*, vol. 83, nos. 23–24, pp. 1943–1978, 2005.

[2] X. Huang and Y. M. Xie, *Evolutionary Topology Optimization of Continuum Structures: Methods and Applications*. Chichester, U.K.: Wiley, 2010.

[3] J. D. Deaton and R. V. Grandhi, "A survey of structural and multidisciplinary continuum topology optimization: Post 2000," *Struct. Multidiscipl. Optim.*, vol. 49, no. 1, pp. 1–38, 2014.

[4] A. Ghali, A. M. Neville, and T. G. Brown, *Structural Analysis: A Unified Classical and Matrix Approach*, 6th ed. New York, NY, USA: Spon, 2009.

[5] X. Huang and Y. M. Xie, "A new look at ESO and BESO optimization methods," *Struct. Multidiscipl. Optim.*, vol. 35, no. 1, pp. 89–92, 2008.

[6] M. Zhou and G. I. N. Rozvany, "On the validity of ESO type methods in topology optimization," *Struct. Multidiscipl. Optim.*, vol. 21, no. 1, pp. 80–83, 2001.

[7] M. P. Bendsøe, "Optimal shape design as a material distribution problem," *Struct. Optim.*, vol. 1, no. 4, pp. 193–202, 1989.

[8] G. I. N. Rozvany, "A critical review of established methods of structural topology optimization," *Struct. Multidiscip. Optim.*, vol. 37, no. 3, pp. 217–237, 2009.

[9] X. Huang and Y. M. Xie, "A further review of ESO type methods for topology optimization," *Struct. Multidiscipl. Optim.*, vol. 41, no. 5, pp. 671–683, 2010.

[10] W. S. Dorn, R. E. Gomory, and H. J. Grenberg, "Automatic design of optimal structures," *J. Mechanique*, vol. 3, no. 1, pp. 25–52, 1964.

[11] G.-C. Luh and C.-Y. Lin, "Optimal design of truss-structures using particle swarm optimization," *Comput. Struct.*, vol. 89, no. 23–24, pp. 2221–2232, 2011.

[12] M. Fenton *et al.*, "Automatic innovative truss design using grammatical evolution," *Autom. Construct.*, vol. 39, pp. 59–69, Apr. 2014.

[13] G. I. N. Rozvany, "Exact analytical solutions for some popular benchmark problems in topology optimization," *Struct. Optim.*, vol. 15, no. 1, pp. 42–48, 1998.

[14] K. Deb and S. Gulati, "Design of truss-structures for minimum weight using genetic algorithms," *Finite Elem. Anal. Design*, vol. 37, no. 5, pp. 447–465, 2001.

[15] Y. Li and Y. Chen, "Beam structure optimization for additive manufacturing based on principal stress lines," in *Proc. Solid Freeform Fab. Symp.*, Austin, TX, USA, 2010, pp. 666–678.

[16] T. Jones, "Evolutionary algorithms, fitness landscapes, and search," Ph.D. dissertation, Dept. Comput. Sci., Univ. New Mexico, Albuquerque, NM, USA, 1995.

[17] A. G. M. Michell, "The limits of economy of material in frame structures," *Philosoph. Mag.*, vol. 8, no. 47, pp. 589–597, 1904.

[18] G. I. N. Rozvany, N. Olhoff, M. P. Bendsøe, T. G. Ong, and W. T. Szeto, "Least-weight design of perforated elastic plates," Dept. DCAMM, Tech. Univ. Denmark, Lyngby, Denmark, Tech. Rep. 306, 1985.

[19] H. Kawamura, H. Ohmori, and N. Kito, "Truss topology optimization by a modified genetic algorithm," *Struct. Multidiscipl. Optim.*, vol. 23, no. 6, pp. 467–472, 2002.

[20] T. Kowaliw, P. Grogono, and N. Knarma, "Environment as a spatial constraint on the growth of structural form," in *Proc. 9th Annu. Conf. Genet. Evol. Comput. (GECCO)*, London, U.K., 2007, pp. 1037–1044.

[21] T. Kowaliw, P. Grogono, and N. Knarma, "The evolution of structural design through artificial embryogeny," in *Proc. IEEE Symp. Artif. Life (CI-ALife)*, Honolulu, HI, USA, 2007, pp. 425–432.

[22] M. O'Neill and C. Ryan, "Grammatical evolution," *IEEE Trans. Evol. Comput.*, vol. 5, no. 4, pp. 349–358, Aug. 2001.

[23] J. Byrne *et al.*, "Combining structural analysis and multi-objective criteria for evolutionary architectural design," in *Applications of Evolutionary Computation, Part II* (LNCS 6625), C. D. Chio *et al.*, Eds. Berlin, Germany: Springer, 2011, pp. 204–213.

[24] M. O'Neill *et al.*, "Evolutionary design using grammatical evolution and shape grammars: Designing a shelter," *Int. J. Design Eng.*, vol. 3, no. 1, pp. 4–24, 2010.

[25] J. Byrne, M. Fenton, E. Hemberg, J. McDermott, and M. O'Neill, "Optimising complex pylon structures with grammatical evolution," *Inf. Sci.*, vol. 316, pp. 582–597, Sep. 2015.

[26] J. Byrne, P. Cardiff, A. Brabazon, and M. O'Neill, "Evolving parametric aircraft models for design exploration and optimisation," *Neurocomputing*, vol. 142, pp. 39–47, Oct. 2014.

[27] J. McDermott *et al.*, "String-rewriting grammars for evolutionary architectural design," *Environ. Plan. B Plan. Design*, vol. 39, no. 4, pp. 713–731, 2012.

[28] M. O'Neill *et al.*, "Shape grammars and grammatical evolution for evolutionary design," in *Proc. 11th Annu. Conf. Genet. Evol. Comput. (GECCO)*, Montreal, QC, Canada, 2009, pp. 1035–1042.

[29] M. Fenton, D. Lynch, S. Kucera, H. Claussen, and M. O'Neill, "Load balancing in heterogeneous networks using an evolutionary algorithm," in *Proc. IEEE Conf. Evol. Comput.*, Sendai, Japan, 2015, pp. 70–76.

[30] M. O'Neill and C. Ryan, *Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language*. Boston, MA, USA: Kluwer Academic, 2003.

[31] J. Hugosson, E. Hemberg, A. Brabazon, and M. O'Neill, "Genotype representations in grammatical evolution," *Appl. Soft Comput.*, vol. 10, no. 1, pp. 36–43, 2010.

[32] I. Dempsey, M. O'Neill, and A. Brabazon, *Foundations in Grammatical Evolution for Dynamic Environments*. Berlin, Germany: Springer, 2009.

[33] B. Delaunay, "Sur la sphère vide," *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk*, vol. 7, no. 6, pp. 793–800, 1934.

[34] P.-L. George and H. Borouchaki, *Delaunay Triangulation and Meshing*. Paris, France: Hermes, 1998.

[35] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*. Cambridge, MA, USA: MIT Press, 1993.

[36] M. Joachimczak and B. Wróbel, "Co-evolution of morphology and control of soft-bodied multicellular animats," in *Proc. 14th Annu. Conf. Genet. Evol. Comput. (GECCO)*, Philadelphia, PA, USA, 2012, pp. 561–568.

[37] K. R. Gabriel and R. R. Sokal, "A new statistical approach to geographic variation analysis," *Syst. Biol.*, vol. 18, no. 3, pp. 259–278, 1969.

[38] S. Le. (2010). *San Le's Free Finite Element Analysis*. [Online]. Available: http://slffea.sourceforge.net/
[39] *Structural Use of Steelwork in Building*, British Standards Institution Standard BS 449-2, 1969.
[40] *Structural Use of Steelwork in Building*, British Standards Institution Standard BS 5950-1, 2000.
[41] R. Kicinger, T. Arciszewski, and K. A. De Jong, "Evolutionary design of steel structures in tall buildings," *J. Comput. Civil Eng.*, vol. 19, no. 3, pp. 223–237, 2005.
[42] F. Cobb, *Structural Engineer's Pocketbook*, 2nd ed. Oxford, U.K.: Butterworth–Heinemann, 2009.
[43] P. Hajela and E. Lee, "Genetic algorithms in truss topological optimization," *Int. J. Solids Struct.*, vol. 32, no. 22, pp. 3341–3357, 1995.
[44] A. Kaveh and S. Talatahari, "Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures," *Comput. Struct.*, vol. 87, nos. 5–6, pp. 267–283, 2009.
[45] L. J. Li, Z. B. Huang, F. Liu, and Q. H. Wu, "A heuristic particle swarm optimizer for optimization of pin connected structures," *Comput. Struct.*, vol. 85, nos. 7–8, pp. 340–349, 2007.
[46] S. Ruiyi, G. Liangjin, and F. Zijie, "Truss topology optimization using genetic algorithm with individual identification technique," in *Proc. World Congr. Eng.*, vol. II. London, U.K., 2009, pp. 1089–1093.
[47] K. S. Lee and Z. Geem, "A new structural optimization method based on the harmony search algorithm," *Comput. Struct.*, vol. 82, nos. 9–10, pp. 781–798, 2004.
[48] N. S. Khot and L. Berke, "Structural optimization using optimality criteria methods," in *New Directions in Optimum Structural Design*, E. Atrek, R. H. Gallagher, K. M. Ragsdell, and O. C. Zienkiewicz, Eds. New York, NY, USA: Wiley, 1984.
[49] J. Adeli and S. Kumar, "Distributed genetic algorithm for structural optimization," *J. Aerosp. Eng.*, vol. 8, no. 3, pp. 156–163, 1995.
[50] M. Hemberg, U.-M. O'Reilly, and P. Nordin, "GENR8—A design tool for surface generation," in *Proc. GECCO Late Break. Papers*, San Francisco, CA, USA, 2001, pp. 160–167.

Michael Fenton received the M.E. and Ph.D. degrees in structural engineering in 2010 and 2015, respectively, from the School of Civil, Structural, and Environmental Engineering, University College Dublin (UCD), Dublin, Ireland.

He is a Post-Doctoral Researcher with the Natural Computing Research and Applications Group, UCD. His research interests include structural engineering, evolutionary computation, problem solving, and music.

Ciaran McNally received the Ph.D. degree from University College Dublin, Dublin, Ireland, in 2001. His Ph.D. studies were in alkali aggregate reaction in Irish concretes.

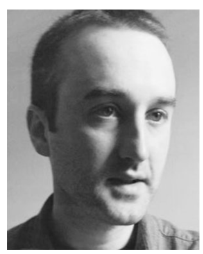
He is a Civil Engineer with University College Dublin, Dublin, Ireland. He has been active in several European research projects as partner and coordinator and has over 75 research publications. His research interests include materials engineering and structural optimization.

Jonathan Byrne received a degree in computer science and software engineering from Maynooth University, Ireland, in 2004 and the Ph.D. degree in evolutionary design optimization from University College Dublin, Dublin, Ireland, in 2011.

He is a Research Scientist with the Urban Modelling Group, University College Dublin, Dublin, Ireland. He is the Director of the Additive Manufacturing Research Centre, University College Dublin, Dublin, Ireland, where he is developing techniques that enable the manufacture and validation of digitally optimized designs. His research interests include design optimization, mesh reconstruction, and additive manufacturing.

Erik Hemberg received the Ph.D. degree in computer science from University College Dublin, Dublin, Ireland, in 2010.

He is a Post-Doctoral Researcher with AnyScale Learning For All Group, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA. He performs research regarding scaleable machine learning. His research interests include tax evasion and physiological time series prediction.



James McDermott received the Ph.D. degree from University of Limerick, Limerick, Ireland, in 2008. His Ph.D. studies were in evolutionary computation applied to sound synthesis.

He was an Irish Research Council—Marie Curie Post-Doctoral Fellow with Massachusetts Institute of Technology, Cambridge, MA, USA. He is currently a Lecturer and the M.Sc. Programme Director of Business Analytics with Lochlann Quinn School of Business, University College Dublin, Dublin, Ireland. His research interests include evolutionary design and genetic programming.

He is an Associate Editor of *ACM SIGEvolution Newsletter*.

Michael O'Neill received the Ph.D. degree from University of Limerick, Limerick, Ireland, in 2001.

He is the ICON Chair of Business Analytics with the School of Business, University College Dublin (UCD), Dublin, Ireland, and the Founding Director of the Natural Computing Research and Applications Group, UCD. He has published over 250 peer-reviewed publications and co-authored a number of successful funding applications with a total value of over nine million euros.