An Improved Shape Annealing Algorithm For Truss Topology Generation

G. Reddy

J. Cagan

Department of Mechanical Engineering,
Carnegie Mellon University,
Pittsburgh, PA

An improved shape annealing algorithm for truss topology generation and optimization, based on the techniques of shape grammars and simulated annealing, is introduced. The algorithm features a shape optimization method using only simulated annealing with a shape grammar move set; while no traditional gradient-based techniques are employed, the algorithm demonstrates more consistent convergence characteristics. By penalizing the objective function for violated constraints, the algorithm incorporates geometric constraints to avoid obstacles. The improved algorithm is illustrated on various structural examples taking into account stress, Euler buckling and geometric constraints, generating a variety of solutions based on a simple grammar.

1 Introduction

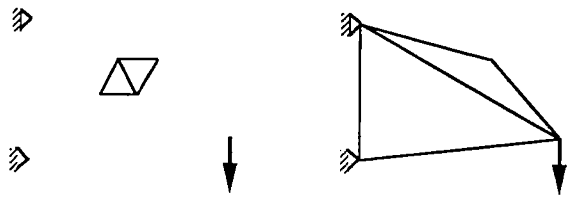
Structural topology optimization has become an important research area. Consistent with shape optimization, the structural topology optimization problem is to minimize an objective function, such as the weight or the cost of the structure, subject to a set of structural constraints: stress constraints, buckling constraints, geometric constraints such as location of loads, support points, and obstacles, as well as a variety of other design criteria. Beyond shape optimization (selection of cross-sectional areas and node locations) of the configuration, the problem becomes the generation of the optimal topology itself; shape optimization of an inferior topology is an inefficient use of resources and so the best configuration of members to shape optimize is desired.

This paper focuses on truss topology optimization. Some limited analytical approaches go back to early in the century (Michell, 1904). In recent years, various research efforts have attempted to use numerical approaches to generate truss topologies. Three such classes of approaches can be classified as ground structures, emergent material distribution structures, and heuristically produced structures. Of the ground structures, a highly connected grid of members is assumed in which members are removed (e.g., Hemp, 1973), or a grid of permissible nodal points are assumed upon which members are generated (e.g., Dorn et al., 1964; Pederson, 1992; Achtziger et al., 1992), often, but not always, using a linear programming approach. See Kirsch (1989) for a thorough discussion of these approaches. These efforts still limit permissible topologies, and do not solve the general topological generation problem of introducing new members and nodes as needed. Within the class of heuristically generated structures, Spillers (1985) introduced the idea of a limited grammatical approach to introduce new members into a structure. Other attempts through knowledge based systems (e.g., Shah, 1988; Rogers et al., 1988) and through theorem proving methods (Lakmazaheri and Rasdorf, 1990) have also generated a class of designs limited by the chosen heuristics.

A more general solution approach to the topology optimization problem can be found in the emergent material distribution structures. Three major approaches to this problem are the homogenization method (Bendsøe and Kikuchi, 1988; also see Bremicker et al., 1991; Diaz and Belding, 1993; and Rodrigues and Fernandes, 1993), genetic algorithms (Chapman et al., 1993) and combinatorial particle/element approaches (Anagnoustou et al., 1992). In these approaches the structure emerges from within the predefined acceptable geometric bounds based on the stress contours; however, the resulting material must be further interpreted to determine the useful structure generated. Papalambros and Chirehdast (1990) developed one such system (the ISOS system) for the homogenization technique where vision algorithms are used to interpret the stress distributions which heuristics convert to usable structural members. These material distribution approaches are also limited by the types of constraints that can be considered; because overall continuity is not recognized, it appears that buckling cannot be modeled during the optimization process.

In a previous paper Reddy and Cagan (1994a) introduced the shape annealing approach to truss design. Modified from an original algorithm introduced by Cagan and Mitchell (1993), the approach optimizes topology based on user defined design preferences by combining shape grammars (Stiny, 1980) and the stochastic global optimization approach of simulated annealing (Kirkpatrick et al., 1983). The shape grammars define permissible design rules for member types and connectivity, while the simulated annealing searches the space of possible configurations generated by the grammar. If all possible topologies can be generated from the grammar

Conributed by the Design Automation Committee for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received March 1994; revised Nov. 1994. Associate Technical Editor: S. Azarm.



(a) (b)

Fig. 1 Example shape space (1*a*) and analogous artifact space (1*b*) truss generation from original 1994 algorithm

then the globally optimal topology can be found, otherwise the optimal solution is limited to the topologies generated from the grammar. In the original shape annealing algorithm, shapes are generated based on repeated application of shape rules from the shape grammar, creating a *shape space* design Fig. 1(*a*), and then *stretched* (i.e., connected) to nearby loads and anchor points to create a valid truss[1] within the *artifact space*, Fig. 1(*b*). Gradient-based shape optimization is then performed on the truss to minimize the objective function (e.g., weight) while satisfying the design constraints (e.g., yield stress and Euler buckling). The objective function and the constraints are calculated through finite element modeling of the truss. In the 1994 paper, a simple grammar that adjacently (serially) adds or removes equilateral triangles to or from the shape design is used.

The process is iterative in that a shape is generated, mapped to the artifact space, and evaluated through shape optimization, and then the shape is modified and the process repeated. At each iteration the objective function of the design is compared to the previous design. If the objective function is better, then the new design is accepted as the current design. If, however, the objective function is worse, then there is still a probability that the new design will be accepted. This probability decreases over the number of iterations based on the simulated annealing algorithm.

In shape annealing, shape grammars generate a variety of designs by applying different sequences of rules from the grammar to an existing design. The shape annealing approach models predefined features through the grammar; thus structures that perfectly distribute material based on the stress distributions do not emerge immediately from the algorithm; however, interpretation of distributed material is also not required. Rather, a valid truss characterized by the grammar is always generated.[2] In addition, the approach can include any constraints that can be mathematically or heuristically modeled (including buckling) and is not limited by a preset grid of members or nodes.

The shape annealing algorithm introduced in 1994 showed promise as a technique for topology optimization; however there are limitations to the method: (*1*) The algorithm utilizes a gradient-based technique for shape optimization; with nonconvex constraints, local optima can be generated; further, the algorithm is not able to avoid obstacles. (*2*) Because complete shape optimization is performed after every step, a large amount of computational time is wasted in pursuing patently undeserving topologies. (*3*) The grammar presented in the paper also contributes to its inefficiency; a shape rule application from the grammar does not always make meaningful changes to the design. Figure 1 shows such a rule application where the two outside members, Fig. 1(*b*), do not take any load. (*4*) Although each shape space topology maps to only one truss through stretching, multiple shape space topologies may map to the same truss. Again referring to Fig. 1(*b*), once the design is shape optimized, the two outside members will be removed and the truss will reduce to the same configuration as is achieved with a single triangle. (*5*) Because every shape rule application changes topology, implying a large change in the objective function, simulated annealing is unable to consistently converge to the optimal solution, i.e., the best design produced during the run may be far better than the design which the algorithm reaches in the end. For the same reason, the solutions produced are also inconsistent over a number of runs.

This paper presents an improved shape annealing algorithm that overcomes many of the disadvantages described above. (*1*) Simulated annealing is employed as the optimization technique throughout the entire process (both topology and shape); local gradient-based techniques are not used. Thus nonconvex constraints can be used and geometric obstacles avoided. (*2*) The simulated annealing algorithm penalizes violated constraints during the optimization process until the design converges on a feasible, optimally directed shape and topology. Because a complete shape optimization is not required at each iteration the computational time is greatly reduced. (*3*) Within the grammar, two sets of shape rules are used: one which changes the geometry of the truss and the cross-sectional areas of the truss members for shape optimization along with one for topology modifications. The rule sets work in conjunction with each other and consistently make load bearing changes to the design. (*4*) The shape space and artifact space are now unified and stretching is eliminated; each design generated is already a valid truss (although constraints may still be violated). (*5*) The new shape grammar makes smaller changes to the design in each iteration, keeping the objective function in the same neighborhood so that the designs can more consistently converge to the same class of structures (i.e., designs have more consistent final objective functions). Each of these changes result in better efficiency, consistency, and performance of the algorithm. Note, however, that due to the remaining topology jumps and large space of possible solutions, repetitive convergence to the same configuration is still not likely. Further, as the complexity of the design constraints increase (e.g., buckling along with gross geometric obstacles) along with the topology jumps, convergence and consistency still decreases; as is typical with simulated annealing, the algorithm can still get stuck in a local optimum or not converge. Yet even for complex geometric obstacles, shape annealing is able to eventually find a feasible solution. The approach generates a variety of design topologies of relatively equal quality (based on the objective function evaluation) each of which could solve a given problem.

2 Improved Shape Annealing Algorithm for Truss Generation

This section describes the improved shape annealing algorithm for truss generation. Two sets of rules are used; one permits small moves within a simulated annealing algorithm for shape optimization while the other makes larger moves by modifying the topology; each rule set works in conjunction with the other forming a grammar which describes the language of permissible trusses. Simulated annealing is used to optimize both the topology and shape. Constraint violations are penalized in the objective function; as the solution converges to the optimum, it also converges to a feasible design and the penalties go to zero. This approach to optimization

[1] A valid truss is defined to be a truss which connects all of the load and anchor points.

[2] Note that due to optimization of the truss, however, new characteristics can emerge from the topology as members are driven to zero cross sectional area.

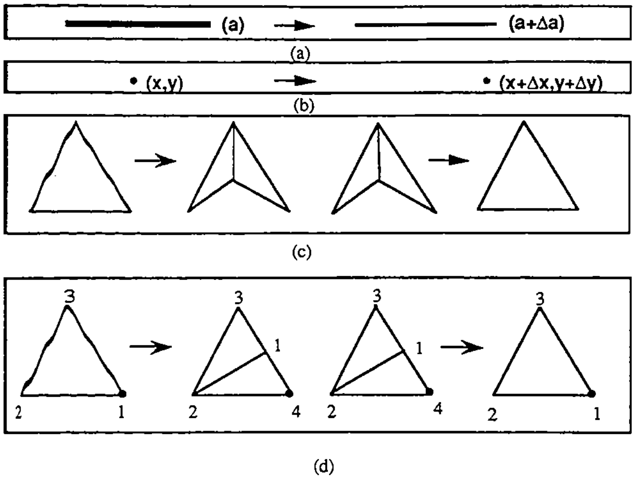


Fig. 2 Shape grammar for sizing (2*a*) shape (2*b*) and topology (2*c*: dividing and 2*d*: adding) topology modifications

allows the algorithm to avoid geometric obstacles. It also leads to more consistent convergence properties.

2.1 The Shape Grammar

2.1.1 Shape Optimization Rules. In this algorithm shape optimization occurs through simulated annealing as opposed to traditional gradient-based algorithms. Simulated annealing works best when moves stay within a neighborhood of the objective function. At first, a larger neighborhood is considered and larger moves are required; over time the move size decreases along with the neighborhood. We introduce a shape rule set, illustrated in Fig. 2, to shape optimize two-dimensional trusses through shape annealing. Size modification rules, Fig. 2(*a*), increase or decrease the cross-sectional area of a member. Shape modification rules, Fig. 2(*b*), move a node in the truss to a different location. If a size modification rule is applied, one of the truss members is selected and its cross-sectional area is either increased or decreased, with equal probability, by a default amount which starts from a finite preset value for the initial iteration and reduces to a value near zero as the iterations progress to the final iteration; this decrease in step size allows the annealer to decrease the neighborhood over time. If a shape modification rule, Fig. 2(*b*), is applied one of the nodes in the truss is selected and moved in a randomly selected direction by a default value which again starts from a finite preset value for the initial iteration, and reduces to a value close to zero by the final iteration, reducing the neighborhood as well. Size and shape modifications move the truss to an optimum geometry through the annealer. For both modifications, the magnitude of the change distance decreases with the annealing temperature; thus the initial modifications can be quite large and random and, as the annealer becomes more deterministic, the moves become much smaller. Note that these rules are continuous rules in that, over a successive number of moves, the design state can repeat a previous state; thus no reversal rules are required.

2.1.2 Topology Rules. The topology modification rules take a different form than the shape and sizing rules. Topology modifications may create disturbances in the objective function by changing the basic configuration of the truss. Here the neighborhood does not decrease over time. Although the grammar presented in Reddy and Cagan (1994a) produces good truss designs, there are configurations which cannot be produced through that grammar; further that grammar only modifies the shape in a serial manner off the last point in the chain of triangles. Each time a modification takes place the shape has to be reconnected to the load and

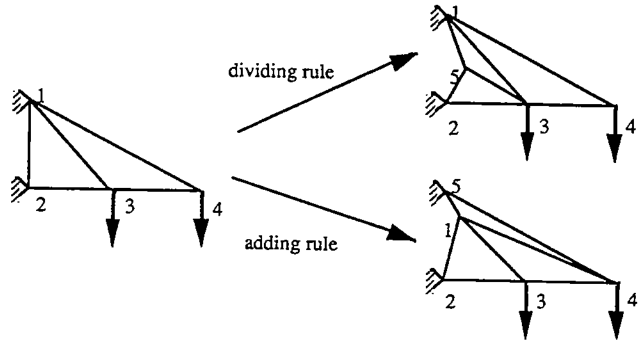


Fig. 3 Application of dividing and adding topology modification rules

anchor points and completely shape optimized. A better approach would have topology modifications occur between shape modifications based on the current shape of the truss; from that point shape optimization continues on the modified topology.

The current grammar takes this approach. Dividing and adding rules are used as shown in the left side of Figs. 2(*c*) and 2(*d*). A dividing rule modifies the topology by dividing an existing triangle in the design into two new triangles as shown in the left side of Fig. 2(*c*). If a dividing rule is to be applied, a random number generator is used to select one of the three sides of the triangle on which to modify the topology. An additive rule is applied at any of the fixed nodes; a fixed node is a node whose geometric location cannot be changed (such as a load or anchor point) as illustrated by the label "·" in Fig. 2(*d*). If an additive rule is applied to the truss, a fixed node is randomly selected [node 1 in the left of Fig. 2(*d*)] and then another node is randomly selected from the adjacent nodes to that fixed node (node 2 in the figure). An additive rule modifies the truss topology by adding a new triangle at the fixed node. Node 1 is disconnected from the fixed node location and is moved by a preset distance. A new node, node 4 in the figure, is connected to the fixed node location and a new triangle, 1-2-4, is formed. Figure 3 shows a truss to which a dividing rule is applied and one to which an additive rule is applied.

In the exploration of the design space, inferior designs may be pursued, with the expectation that they may lead to superior designs. In order to reverse the exploration when the solution path is abandoned, reversal rules are defined for both the dividing and adding rules as shown in the right side of Figs. 2(*c*) and 2(*d*); i.e., a rule can be applied from left-to-right or from right-to-left. Thus the effect of any topology modification rule can be nullified by applying the partner rule for the reverse direction[3]. Note that there is no requirement that a rule be applied only at the last point of rule application; thus the grammar is not limited to serial shape designs as required in the 1994 paper and Cagan and Mitchell's original paper. Although there is no guarantee for completeness of the truss solutions generated from this grammar, the grammar can generate any feasible truss that the original grammar was able to generate.

2.1.3 Grammar Probabilities. The topology modifications may result in large disturbances in the design and evaluation of the objective function. Over time, the shape and sizing modifications result in much smaller disturbances in the objective function. If an annealer spends effort on shape optimization and then makes a change in topology, much of

[3]The shape grammar is required to have a corresponding reversal rule for every discrete shape rule; continuous rules, or rules that discretize a continuum, can generate any solution along its continuum of allowable moves and require no reversal rules.

the effort from the shape optimization in the area where the topology changes is lost; yet, any given topology can have a large range of objective function values for various sizes and dimensions and some indication of the final dimensions must be used to accurately evaluate a topology. Once the optimal topology is determined then the structure need only be shape optimized and no further topology modifications should be applied. Thus the probability of selecting shape and sizing rules versus topology rules should change as the algorithm progresses; toward the start there should be a high probability of selecting a topology modification rule but as time progresses that probability should drop off considerably. At the same time the probability of selecting a shape and sizing rule should increase as the algorithm progresses. An adjustable probability is associated with each rule. Presently, a starting probability of 0.1 is used for topology modifications so that roughly 1 in 10 moves will attempt to modify the topology; the probability is reduced with every iteration so as to reach zero by the final iteration. A probability of 0.45 is used for shape modifications. The probability of a size modification starts at a value of 0.45 and increases linearly to 0.55 at the final iteration. Note that the sum of the probabilities equals 1.0 at any time; our experience finds the current distribution to be effective in both generating topologies and shape optimizing the trusses.

2.2 The Algorithm. The improved shape annealing algorithm requires an initial connected design which provides a path between the loads and the anchor points; the minimum such design of exhaustive nonintersecting connections is currently input by the user (e.g., left side of Fig. 3). Note that the process of stretching is eliminated from this algorithm and the algorithm starts with a fully connected initial shape. All the truss members in the initial design are assigned a default cross-sectional area; any new member added to the design during topology modifications is currently also assigned the same default value.

The shape annealing algorithm, like the simulated annealing algorithm, involves a large number of iterations; every iteration involves a shape rule application and design evaluation as follows: A shape rule is randomly selected according to the probabilities discussed in Section 3.1.3. If a topology modification rule is selected, then the choice between dividing and adding rules as well as the direction of application (left-to-right or right-to-left) is randomly determined. Once a shape rule has been applied, the design is analyzed and the objective function evaluated. As is typical with simulated annealing, at each iteration if the modified design is better in its objective function than the previous design it is accepted as the new design; if it is worse in the objective function then it is accepted with a probability that decreases as the algorithm progresses. This process iterates until convergence is achieved or a limit in number of iterations is reached. The best design generated during the process is also saved.

All analysis is performed through a finite element analysis of the structure; currently only stresses, Euler buckling, and geometric constraints are imposed, although others can easily be added. Note that buckling constraints impose a minimum area on a member; to allow an element in the truss to disappear from the final design through optimization, the buckling constraint is not applied to a member with a cross-sectional area less than a predefined limit. In the final design all remaining elements with area less than that limit are removed, unless they are both required for stability and do not buckle. Obstacles are geometric constraints tested through intersection of the members with the obstruction.

The objective function is evaluated based on the FEM results. The design objective function is calculated. The total objective function is increased based on violations in the design constraints. A penalty is added to the objective function for each constraint violation; the greater the violation of the constraint the larger the penalty added:

$$\text{total\_objective} = \text{design\_objective} + \Sigma(\text{constraint\_violation})^{*}\text{penalty}.$$

As the annealing process progresses, the design is pushed from an infeasible state to a feasible state. For this implementation, the weight of the structure is selected as the design objective function, and the penalty is defined as:

$$\text{penalty} = \text{design\_objective}\left(1 - e^{-K\frac{\text{iteration}}{\text{total\_iterations}}}\right).$$

constraint_violation is the sum of the values of all the violated stress, buckling and geometric constraints. The penalty is set dynamically as a function of the design objective of the current design and the proportion of iterations completed. The penalty is designed so that the constraint violations have little effect during the initial iterations and increasing effect as the iterations progress until it is of the same order of magnitude as the design objective. For this implementation a value of 10 is presently used for K; the larger the value of K selected the larger the rate of increase of the penalty. Any other constraints such as displacement constraints, constraints limiting the cross-sectional area, and frequency constraints could also be incorporated at this stage by assigning appropriate penalties for constraint violation.

Note the effect of this new approach to performing shape annealing. In the 1994 paper, after any iteration, a feasible, shape optimized structure was generated. In the current approach, after each iteration, the structure is not guaranteed to be feasible or shape optimized; it is possible that the only design that will be both feasible and shape optimized is the final design. It is through the simulated annealing optimization that the structure pushes out of the infeasible region. This approach is similar to that used in VLSI layout with simulated annealing: As components are laid out they are allowed to overlap, causing a penalty on the objective function; the annealing process pushes the components away from each other until the overlap disappears (Jepsen and Gelatt, 1983).

3 Examples

The success of the method is now demonstrated with a single load truss problem. First the ability of shape annealing to laterally explore the design space with good convergence under stress and buckling constraints is illustrated. Next, the same problem is formulated with geometric obstacles, illustrating the ability of the method to use geometric constraints. In each example note the variety of solutions generated and the ability of the algorithm to mold the design based on the constraints. Each example runs for up to 100,000 iterations; convergent designs run in fewer iterations. Note that in those problems that were also solved using the 1994 algorithm, the current algorithm solves the problem, with generally better solution objective function values, in one fifth the time. For an example with multiple loads, see Reddy and Cagan (1994b). All structures are optimized with simulated annealing; solutions without geometric obstacles have been verified by optimizing the topology generated by shape annealing with traditional gradient-based shape optimization. For reference the node locations and cross-sectional areas of the bars can be found in Reddy and Cagan (1994c).

3.1 Single Load/10-Bar Truss Specifications. The classic 10-bar truss, single load problem specifications are as shown in Fig. 4(*a*). The material properties are: Young's modulus of 68.8 GPa, allowable stress of 172 MPa, and density of 27 kN/m^3. The shape optimized 10-bar truss topology produces designs of weight 4971 N without buckling and

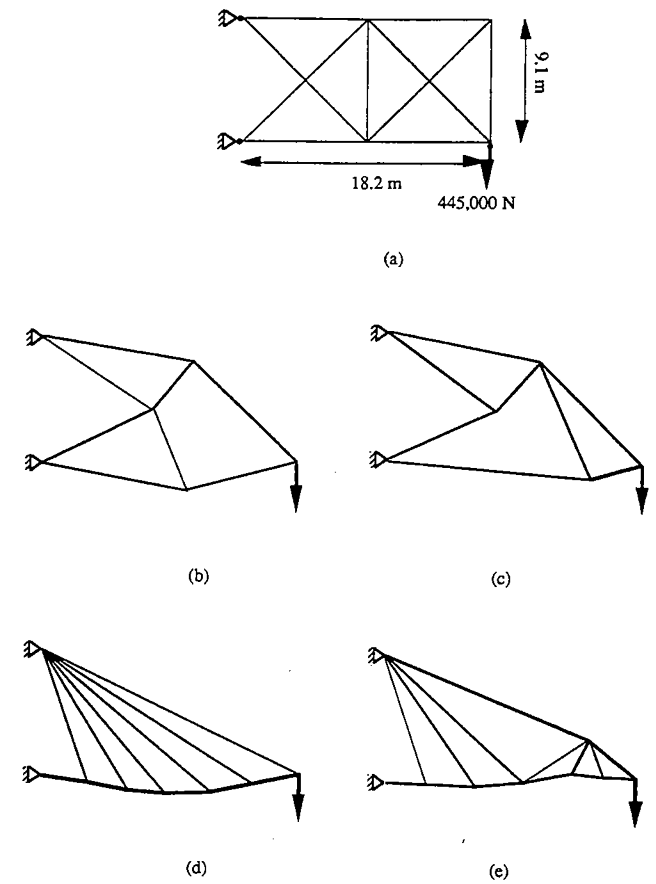


Fig. 4 Single load truss example: input (4*a*), solutions without buckling (4*b* and 4*c*), solutions with Euler buckling (4*d* and 4*e*)

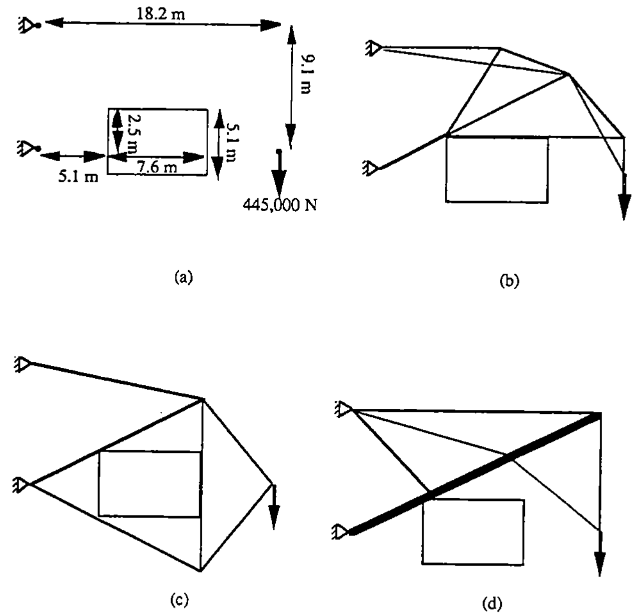


Fig. 5 Single load truss with obstacle 1: input (5*a*), solutions without buckling (5*b* and 5*c*), solution with Euler buckling (5*d*)

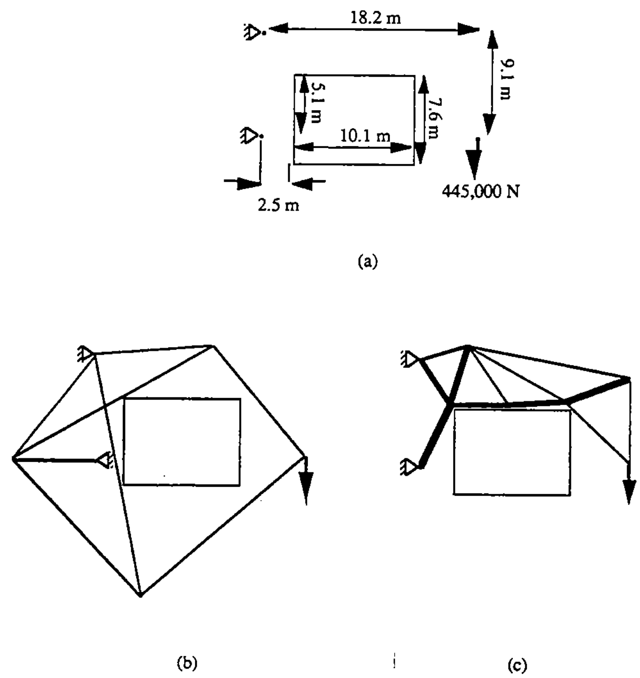


Fig. 6 Single load truss with obstacle 2: input (6*a*), solution without buckling (6*b*), solution with Euler buckling (6*c*)

20884 N when Euler buckling is included. The original algorithm when applied to the same specifications produces a 16 bar topology weight of 4793 N without buckling and a 12 bar topology weight of 12064 N with buckling criteria. The new algorithm presented in this paper, when applied to the same specifications, produces the designs in Figs. 4(*b*) and 4(*c*) without buckling, both 8 bar trusses with weights of 4748 N and 4824 N respectively, and the designs in Figs. 4(*d*) and 4(*e*) with Euler buckling, a 12 bar truss of weight 9025 N and an 11 bar truss of weight 9470 N respectively. The new algorithm produces designs with similar weights, but different topology, to the one produced by the original algorithm when buckling is ignored; however, it produces clearly superior designs when Euler buckling is included. Note that although Fig. 4(*e*) shows a topology with similar characteristics to that generated by the original algorithm, the topology of the truss in Fig. 4(*d*) is quite different with compression bars held by string-like tension bars to the upper support point; the design is lighter as well. The shape annealing algorithm proposes significantly different and superior topologies to the 10 bar topology proposed by the designer in this scenario.

3.2 Single Load Specifications with Obstacles. Three different obstacles are now placed within the specifications for the single load truss problem given above. The first, a single rectangular object, is placed as shown in Fig. 5(*a*); the algorithm finds solutions shown in Figs. 5(*b*) and 5(*c*) with weights of 5407 N and 5595 N without buckling, and Fig. 5(*d*) with buckling having a weight of 20626 N[4]. A larger obstacle shown in Fig. 6(*a*), making the lower support point difficult to reach, leads to shape annealed solutions shown in Fig. 6(*b*) without buckling having a weight of 8834 N, and Fig. 6(*c*) with buckling having a weight of 25823 N. Note how the truss in Fig. 6(*b*) moves behind the supports to gain more space to work with. Figure 7(*a*) demonstrates a tall obstacle, making the path between the load and either support difficult to connect; shape annealing negotiates the obstacle, generating the crane-like design shown in Fig. 7(*b*) without buckling having a weight of 8157 N. For the obstacle of Fig. 7(*a*) with buckling included, shape annealing, after several tries, is able to generate the truss shown in Fig. 7(*c*) having a weight of 28600 N. Simulated annealing starts by violating the obstacles and pushes the solution into the feasible space.

[4]Obstacle intersection in the figures is due to the exaggerated line thickness; the actual member does not intersect the obstacle.

Concluding Remarks

Both the original shape annealing algorithm for truss gen-

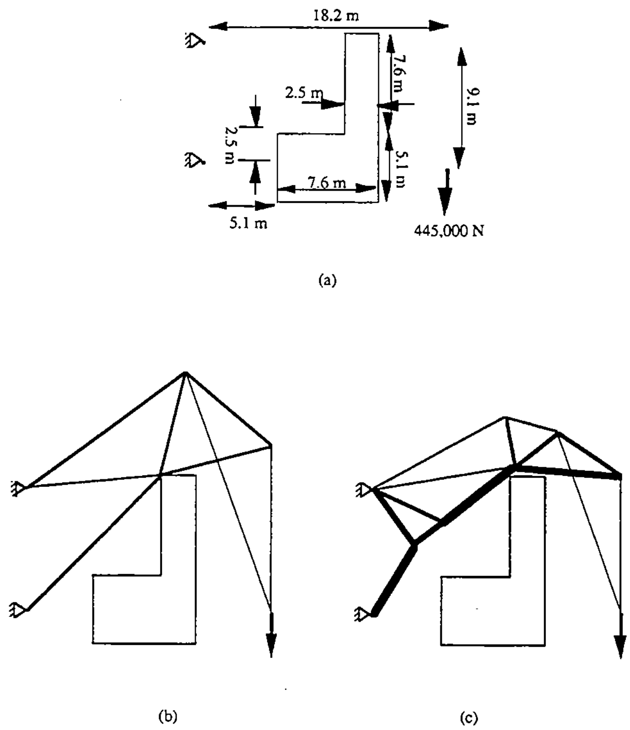


Fig. 7 Single load truss with obstacle 3: input (7*a*), solution without buckling (7*b*), solution with Euler buckling (7*c*)

Table 1 Means (top) and standard deviations (bottom), in N, of runs over single load specifications

no obstacles -stress only	no obstacles-stress & buckling	small obstacle-stress only	small obstacle-stress & buckling	medium obstacle-stress only	medium obstacle-stress & buckling	tall obstacle-stress only	tall obstacle-stress & buckling
4913	13007	6043	21672	10102	67510	8958	56066
18	2123	294	5100	1829	30474	1015	23127

eration and the algorithm introduced in this paper demonstrate the feasibility of shape annealing to generate truss structures. Of interest is the large variety of feasible structures the algorithm is able to generate from the simple shape grammar for classes of objective function values. There are three major directions this research is taking:

(*1*) One direction is the development of efficient grammars. By investigating existing truss configurations (indicating designers' preferences) and the optimal designs generated by shape annealing, we expect to further refine the current grammar. In addition, some understanding of the completeness of the grammar would be useful. Due to the generality of the method, other elements (frames and plates) and three-dimensional structures can be modeled with the shape grammars and used within shape annealing to generate more interesting structures.

(*2*) Convergence of the algorithm is still incomplete. For designs without buckling and geometric obstacles, the convergence is excellent; when buckling or simple obstacles are included convergence is good; when both buckling and simple obstacles are included together, convergence is fair; when buckling and gross geometric obstacles are included, convergence and consistency of the algorithm is poor, although a solution can be found. In almost every run a feasible solution is found even though the algorithm does not always converge on the best solution. By examining the best solutions found these convergent characteristics can be seen. Table 1 shows the mean and standard deviation of the best solutions for 20 runs of the algorithm for the single load specification with and without obstacles; the medium obstacle (Fig. 6) and tall obstacle (Fig. 7) with buckling did not find a feasible solution each run resulting in only 16 data points each. Note that the medium obstacle converges with less consistency due to the tight space near the bottom support.

Solution of the more complex problems is difficult; however, we believe consistent convergence can be obtained. One approach to improving the algorithm is improving the annealing schedule, as discussed below. Another aspect of the problem is the jump in the objective function due to the topology modifications; simulated annealing works best with smooth changes in the objective function as the solution is perturbed. We are exploring topology modification rules which further decrease the jump in objective function value. Completeness of the algorithm also requires additional analysis as well as more thorough structural stability calculations.

(*3*) The other aspect of the algorithm which greatly affects the quality of the solutions is the annealing schedule. This implementation uses a simple schedule where the temperature is reduced by a fixed amount at each iteration. In related work we have found that dynamic annealing schedules (Huang et al., 1986) and move sets from the VLSI community can greatly improve the quality and consistency of the annealing solutions (Szykman and Cagan, 1994). However, due to the jumps in the objective function as the topology changes, this problem may not be as well suited for such standard schedules. An annealing schedule better suited for this problem is being investigated.

Acknowledgments

The authors wish to thank Bill Mitchell, Rob Rutenbar, Kristi Shea, Glenn Sinclair, and Simon Szykman for their discussions on this work and the Engineering Design Research Center, an NSF Center at Carnegie Mellon University, as well as the National Science Foundation under grant DDM-9258090 for supporting this work.

References

Achtziger, W., Bendsøe, M., Ben-Tal, A., and Zowe, J., 1992, "Equivalent Displacement Based Formulations for Maximum Strength Truss Topology Design," *Impact of Computing in Science and Engineering*, Vol. 4, pp. 315–345.

Anagnoustou, G., Rønquist, E. M., and Patera, A. T., 1992, "A Computational Procedure for Part Design," *Computer Methods in Applied Mechanics*, Vol. 97, pp. 33–48.

Bendsøe, M., and Kikuchi, N., 1988, "Generating Optimal Topologies in Structural Design using Homogenization Method," *Computer Methods in Applied Mechanics and Engineering*, Vol. 71, pp. 197–224.

Bremicker, M., Chirehdast, M., Kikuchi, M., and Papalambros, P. Y., 1991, "Integrated Topology and Shape Optimization in Structural Design," *Mechanics of Structures and Machines Int. J.*, Vol. 19, No. 4.

Cagan, J., and Mitchell, W. J., 1993, "Optimally Directed Shape Generation by Shape Annealing," *Environment and Planning B*, Vol. 20, pp. 5–12.

Chapman, C., Saitou, K., and Jakiela, M. J., 1993, "Genetic Algorithms as an Approach to Configuration and Topology Design," published in proceedings: DE-Vol. 65-1, *Advances in Design Automation*, ASME, Albuquerque, NM, September 19–22, 1993, Vol. 1, pp. 485–498.

Diaz, A. R., and Belding, B., 1993, "On Optimum Truss Layout by a Homogenization Method," ASME Journal of Mechanical Design, Vol. 115, pp. 367–373.

Dorn, W. S., Gomory, R. E., and Greenberg, H. J., 1964, "Automatic Design of Optimal Structures," *Journal de Mècanique*, Vol. 3, No. 1, pp. 25–52.

Hemp, W. S., 1973, *Optimum Structures*, Clarendon, Oxford.

Huang, M. D., Romeo, F., and Sangiovanni-Vincentilli, A., 1986, "An Efficient Cooling Schedule for Simulated Annealing," *Proceedings of 1986 IEEE International Conference in CAD*, Nov., pp. 381–384.

Jepsen, D. W., and Gelatt, Jr., C. D., 1983, "Macro Placement by Monte Carlo Annealing," *Proceedings of the IEEE International Conference on Computer Design*, November, pp. 495–498.

Kirkpatrick, S., Gelatt, Jr., C. D., and Vecchi, M. P., 1983, "Optimization by Simulated Annealing," *Science*, Vol. 220, No. 4598, pp. 671–679.

Kirsch, U., 1989, "Optimal Topologies of Structures," *Applied Mechanics Review*, Vol. 42, No. 8, pp. 223–238.

Lakmazaheri, S., and Rasdorf, W. J., 1990, "The Analysis and Partial Synthesis of Truss Structures via Theorem Proving," *Engineering with Computers*, Vol. 6, pp. 31–45.

Michell, A. G. M., 1904, "The Limits of Economy of Materials in Frame Structures," *Philosophical Magazine*, S.6, Vol. 8, No. 47, pp. 589–597.

Papalambros, P. Y., and Chirehdast, M., 1990, "An Integrated Environment for Structural Configuration Design," *Journal of Engineering Design*, Vol. 1, No. 1, pp. 73–96.

Pedersen, P., 1992, "Topology Optimization of Three Dimensional Trusses," *Topology Design of Structures*, NATO ASI Series—NATO Advanced Research Workshop (Bendsøe, M. P. and C. A. Mota Soares, eds.), Sesimbra, Portugal, June 20–26, Kluwer Academic Publishers, Dordrecht.

Reddy, G., and Cagan, J., 1995, "Optimally Directed Truss Topology Generation Using Shape Annealing," ASME Journal of Mechanical Design, Vol. 117, No. 1, pp. 206–209.

Reddy, G., and Cagan, J., 1994a, "An Improved Shape Annealing Method For Truss Topology Generation," *Proceedings of ASME Design Theory and Methodology Conference*, Minneapolis, MN, September 11–14.

Reddy, G., and Cagan, J., 1994b, "An Improved Shape Annealing Algorithm For Truss Topology Generation—with Detailed Examples," *EDRC Report 24-115-94*, Engineering Design Research Center, Carnegie Mellon University, Pittsburgh, PA 15213.

Rodrigues, H. C., and Fernandes, P. A., 1993, "Generalized Topology Optimization of Linear Elastic Structures Subjected to Thermal Loads," published in proceedings: DE-Vol. 65-1, *Advances in Design Automation*, ASME, Albuquerque, NM, September 19–22, Vol. 1, pp. 769–777.

Rogers, J., Feycock, S., and Sobieszczanski-Sobieski, J., 1988, "STRUTEX—A Prototype Knowledge Based System for Initially Configuring a Structure to Support Point Loads in Two Dimensions," *Artificial Intelligence in Engineering: Design*, (Gero, J. S., ed.), Elsevier publications, Amsterdam, pp. 315–335.

Shah, J. J., 1988, "Synthesis of Initial Form for Structural Shape Optimization," ASME *Journal of Vibration, Acoustics, Stress, and Reliability in Design*, Vol. 110, pp. 564–570.

Spillers, W., 1985, " Shape Optimization of Structures," *Design Optimization*, (Gero, J. S., ed.), Academic Press Inc., Orlando, pp. 41–70.

Stiny, G., 1980, "Introduction to Shape and Shape Grammars," *Environment and Planning B*, Vol. 7, pp. 343–351.

Szykman, S., and Cagan, J., 1995, "A Simulated Annealing Approach to Three Dimensional Component Packing," ASME Journal of Mechanical Design, Vol. 117, No. 2(A), June, pp. 308–314.