Structural Topology Optimization Using a Genetic Algorithm and a Morphological Representation of Geometry

Kang Tai, Shengyin Wang, Shamim Akhtar and Jitendra Prasad

Abstract—**This paper describes an intuitive way of defining geometry design variables for solving structural topology optimization problems using a genetic algorithm (GA). The geometry representation scheme works by defining a skeleton that represents the underlying topology/connectivity of the continuum structure. As the effectiveness of any GA is highly dependent on the chromosome encoding of the design variables, the encoding used here is a directed graph which reflects this underlying topology so that the genetic crossover and mutation operators of the GA can recombine and preserve any desirable geometric characteristics through succeeding generations of the evolutionary process. The overall optimization procedure is tested by solving a simulated topology optimization problem in which a 'target' geometry is pre-defined with the aim of having the design solutions converge towards this target shape. The procedure is also applied to design a straight-line compliant mechanism : a large displacement flexural structure that generates a vertical straight line path at some point when given a horizontal straight line input displacement at another point.**

Keywords—**chromosome code, genetic algorithm, morphological geometric representation, topology optimization.**

I. Introduction

STRUCTURAL design optimization is an area that has seen active research over the last three decades due to the natural desire of engineers to build artifacts and structures that not just satisfy their functional requirements, but also perform those functions in an optimal way. Solving optimization problems where the performance of a structure or component is optimized through a variation of its topology is known as topology design optimization. While shape optimization can usually be performed by varying some typical shape parameters (for example, nodal coordinates, arc radii, control points of spline curves or NURBS, etc.), there are no obvious or simple topological parameters suitable for manipulation within an optimization process. Some of the variables that can be used to define topology are the number of holes in a structure and the (yes or no) existence of material at any particular point in the design space. Therefore, unlike the continuous variables treated in shape problems, the design variables in topology optimization are intrinsically discrete in nature and the problem is essentially a discrete optimization problem of material distribution/arrangement. Such discrete problems are hard because they require the enumeration of large portions of the solution space, and computational order can grow enormously with the number of discrete variables to be resolved. Hence the greater reliance on non-exact algorithms based on heuristics and intuition, compared with shape optimization problems which are routinely solved by mathematical programming techniques.

Among methods found in the topology optimization literature, a popular strategy is to initially discretize the allowable design space into finite elements (FE) and define the required loading/boundary conditions. The optimization procedure will then be concerned with determining which elements should contain material (and so form the structure) and which elements are void (and thus represent the surrounding empty space). Based on this strategy, Bendsoe and Kikuchi [1] defined the problem with a composite material represented by each element having material plus a void (hole) inside. The material properties of each element is then dependent on the size and orientation of the void within the element according to a homogenization relationship. A sizing optimization is then performed to optimize the sizes/orientations of the voids of all the elements for a given objective function. Elements with relatively large voids (and hence low material density) will represent empty space while those with smaller voids (and higher material density) denote that material exists and hence form the structure. An alternative but conceptually similar approach is to directly use the material density of each element (instead of voids) as the design variable. An empirical formula is then applied to relate this density with the elastic modulus, without the need for a homogenization formulation. The topology designs produced by this material density approach [2] are similar to those obtained with the homogenization method.

Both the homogenization and material density approaches

Manuscript received November 8, 2002. Part of this work was supported by the Singapore Ministry of Education Academic Research Fund under grant RG92/98.

Kang Tai is with the Singapore-MIT Alliance and School of Mechanical and Production Engineering, Nanyang Technological University, Singapore (phone: +65 67904444; fax: +65 67911859; e-mail: mktai@ntu.edu.sg).

Shengyin Wang is with the Singapore-MIT Alliance, Nanyang Technological University, Singapore (e-mail: sywang@ntu.edu.sg).

Shamim Akhtar is with the School of Mechanical and Production Engineering, Nanyang Technological University, Singapore.

Jitendra Prasad is with the School of Mechanical and Production Engineering, Nanyang Technological University, Singapore (e-mail: jitendra@pmail.ntu.edu.sg).

circumvent the difficulty of discrete optimization by using continuous design variables (void sizes/orientations or material density), turning the problem into a continuous optimization problem. However, since the resulting elements can be of intermediate densities ranging from complete voids to completely filled with material, some interpretation by the designer is still needed to determine the final topology and shape of the structure. It is uncertain if the final useful design interpreted from the results is close to the actual optimum point since the criteria for including/excluding any particular element from the structure tends to be arbitrary (for example, based on an arbitrary threshold density value).

A basically different approach illustrated by Chapman et al. [3] directly addresses the discrete nature of the problem by treating it with a discrete optimization method, namely the genetic algorithm (GA). The strategy of discretizing the design space is used, with all the finite elements forming a one-dimensional binary-coded bit-array chromosome~: 0 if the element is to be empty space and 1 if it is part of the structure. The GA is applied over many generations, and each generation is a population of many individual designs, to attain the optimum chromosome string and hence structure. However, the resulting chromosome string may represent structures with checkerboard patterns (alternating elements of material and void) and `floating' elements (elements `floating' in space and not connected to the main structural body). Such designs may be invalid or impractical, but the chromosome representation allows for them. Consequently the overall procedure may not be efficient because expensive computing resources have to be spent analyzing these undesirable designs. Alternatively a `repair' scheme is needed to detect and alter those checkerboard and `floating' patterns as and when they arise out of the crossover and mutation operations. This may, however, corrupt the transmission of genetic characteristics across the generations.

In this present work, the topological optimization problem is also treated as a discrete problem using the GA. However, a recently developed and more effective geometry representation scheme [4] is used together with a graph-theoretic chromosome encoding scheme to improve the formulation of topology problems for GA. It is believed that the coding scheme can better represent the topological/shape characteristics of a structure (and does not create invalid designs), and also helps preserve any desirable characteristics through the evolutionary process.

In this paper, the effectiveness of the techniques developed here are evaluated and compared with that of the binary-coded bit-array chromosome representation. This is done through solving a simulated topology design optimization problem (by way of a 'target' geometry matching problem). The overall optimization procedure is then tested in the design of a compliant (flexural) mechanism. Compliant mechanisms are flexible structures which achieve some desired motion by way of elastic deformation instead of through rigid linkages/joints as in rigid-body mechanisms. To achieve any specific displacement or path, the required structural geometry can best be discovered through a process of design optimization [5].

II. Optimization by a Genetic Algorithm

The optimization method used is a genetic algorithm which has a basic working procedure that starts with an initial randomly-generated population of design solutions, and this population evolves over many generations through selection of parent solutions and then reproduction (crossover and mutation) processes with the aim of improving individual designs according to some measure of fitness. This fitness measure corresponds directly to the objective function of the optimization problem, and the algorithm ranks designs according to this fitness [6]. Each individual in the population is actually the representation of a candidate design solution, and the GA operates on this representation rather than on the solution itself. Hence the representation and the operations that go with it (the crossover and mutation operation) are crucial to the effectiveness and success of the algorithm. In the following sections, the representation is described by how the design geometry is encoded in a chromosome code which is subject to the reproduction operations of crossover/mutation.

III. Morphological Representation of Design Geometry

In this representation scheme, a design problem is characterized by a set of input/output regions which denote regions of the structure where support, loading and functional/response conditions are applied or required. A representative illustration with three input/output regions is shown in Fig. 1. The design space (as denoted by the dashed lines) is the assigned space within which the structure must lie and cannot exceed. While it is still unknown how the rest of the design space will be occupied by the structure, the input/output regions are pre-defined by the designer to exist because any structure must have parts which interact with its surroundings by way of support conditions, loading and/or some other functional interactions.

The design space is next discretized into a mesh of quadrilateral finite elements as shown in a representative illustration in Fig. 2, with the three elements in black corresponding to the input/output regions. For a valid structural design, all input/output regions must be connected to one another (either directly or indirectly) in order to form one single connected structure. A total of three curves have been used in this example, connecting all three regions to one another (i.e. there is a direct connection between any two regions). Hence this scheme is based on specifying connecting curves joining one region to another. Each curve is a Bezier curve defined by its start and end points plus a number of control points in between (Bezier curves have been chosen for use because of their simplicity). The start and end points are the elements of the respective start and end input/output regions. The set of elements through which each curve passes form the `skeleton' connecting the two regions (Fig. 3). Some of the elements surrounding the skeleton are then included to fill up the structure to its final form (Fig. 4). These additional elements represent the `flesh' around the skeleton, and the union of all skeleton, flesh and input/output elements constitute the structure while all other elements remain as the surrounding empty space. The amount of flesh elements to be added is determined by a thickness value. This is done by considering each skeleton element in turn and adding an all-round layer of elements to it [7], with the layer thickness according to the thickness value which can range from zero to some prescribed (integer) maximum.

Fig. 1. Design space with input/output regions shaded

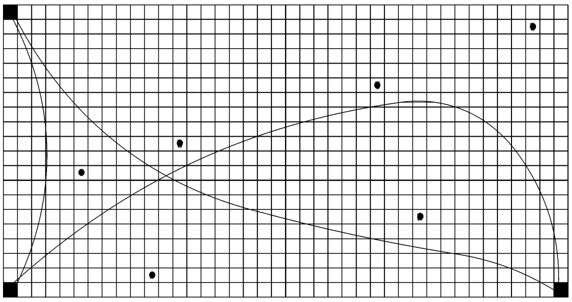


Fig. 2. Connecting input/output elements with curves

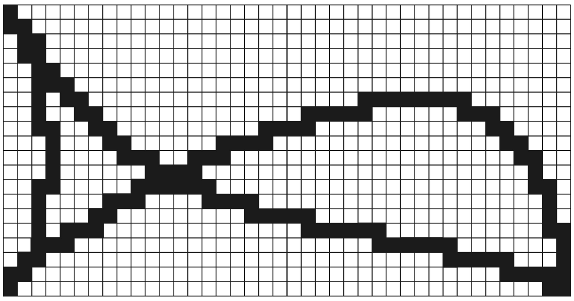


Fig. 3. Skeleton made up of elements along curves

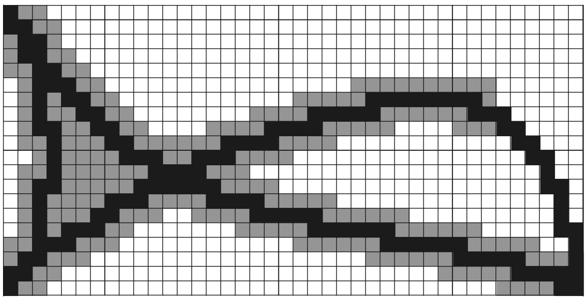


Fig. 4. 'Flesh' elements added to skeleton to form final structure

IV. Chromosome Encoding of Design Geometry

Using a GA for the optimization, the topological/shape representation has to be cast into a chromosome code. Hence the structural geometry illustrated in the preceding section can be encoded as a chromosome in the form of a directed graph [8] as shown in Fig. 5. There is a number in each vertex of the graph, and the vertices are connected by arcs depicted by the arrows in Fig. 5. The vertices and arcs here are the terminology as used in graph theory [9] and not the terms commonly understood in geometric modeling. The numbers in Fig. 5 are only illustrative and may not exactly correspond to the design shown in Fig. 4, and they denote the information shown in Table I.

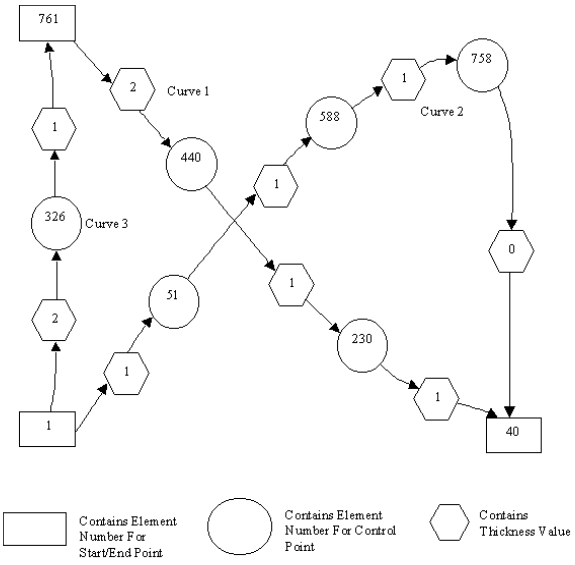

761
2
Curve 1
1
440
326
Curve 3
588
1
758
Curve 2
1
0
2
51
1
230
1
1
1
40
Contains Element Number For Start/End Point
Contains Element Number For Control Point
Contains Thickness Value

Fig. 5. Chromosome code

TABLE I
TYPE OF INFORMATION IN CHROMOSOME CODE

	curve 1	curve 2	curve 3
start input/output point	761	1	1
1st thickness value	2	1	2
1st control point	440	51	326
2nd thickness value	1	1	1
2nd control point	230	588	-
3rd thickness value	1	1	-
3rd control point	-	758	-
4th thickness value	-	0	-
end input/output point	40	40	761

Each start, control or end point of a Bezier curve is taken to be at the centre of the element containing it, so its location is actually referenced by the number of the element. This necessitates a numbering scheme for uniquely labeling each element in the mesh, and in this work, the elements are simply numbered consecutively from left to right along each row, and row by row from bottom to top (any numbering scheme can be used as long as a systematic one-to-one mapping is maintained between every element and its position in the design space). It can be seen from the code that each curve is defined by control points alternating with thickness values. For example, if a curve is defined by two control points, there will be three thickness values (as in curve 1 of this example). With three thickness values, the curve will correspondingly be divided parametrically [7] into three segments with each segment having its thickness determined by the respective thickness value.

It is evident that the topology and shape of a structure defined using this scheme is determined by the arrangement of the skeleton and the flesh. In other words, the topology/shape emerges from the interaction among the curves and their respective thicknesses, and so is not obvious from the chromosome code itself. Slight changes in some thickness values or small shifts in control point positions can lead to topological changes with entire openings (holes) created or destroyed. It is a rich geometric representation that can define a wide variety of topological and shape configurations, using only a fairly compact chromosome code. In fact, the geometric complexity achievable in the design can be controlled by the designer by choosing the number of curves to be used for defining the geometry (subject to a minimum limit to ensure connectivity of the structure) and the number of control points to be used for each curve (the larger the number of points the higher the order of the curve). However, it is clear that the greater the complexity, the larger the number of design variables and so the larger the optimization problem. In addition, the topology/shape of any structure is more intrinsically captured/embodied in this scheme than in a simple material point representation or boundary representation, thus assisting in the transmission of topology/shape characteristics through the evolutionary process. This is achieved by the fact that the chromosome (encoded as a graph) is in a form that reflects the topology/connectivity of the structural geometry. Furthermore, the representation will not render any invalid designs with disconnected `floating' elements/segments or undesirable designs with `checkerboard' patterns. All these advantages with respect to structural validity, disconnected elements and checkerboard patterns are preserved in any child generated from the crossover and mutation operations. Here, the crossover operator devised to work with this chromosome representation is based on randomly selecting an input/output point and sectioning off a random length of each and every curve starting or ending at that input/output point. This is illustrated in Fig. 6 through three example graphs (chromosomes) with different topologies. The dashed line cutting across portions of the chromosome represents the sectioning line or the crossover line, which is the equivalent of the crossover point in a basic crossover operation for a one-dimensional chromosome string [10]. Applying the same crossover line to any given pair of parent chromosomes and swapping the information will produce two children. This crossover operator facilitates the exchange (between two parents) of portions of the skeleton radiating from some input/output point, together with the accompanying thickness values. Intuitively, it can better combine the topological/shape characteristics from the two parents into each child (without destroying these characteristics in the process), thereby helping to transmit any desirable geometry characteristics through the generations. In other words, each resulting child inherits geometric characteristics from both parents.

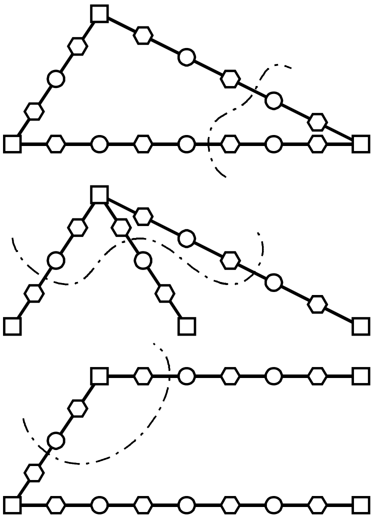


Fig. 6. Crossover line

V. Evaluation and Comparison of Effectiveness of Chromosome Encoding

A. Target matching problem

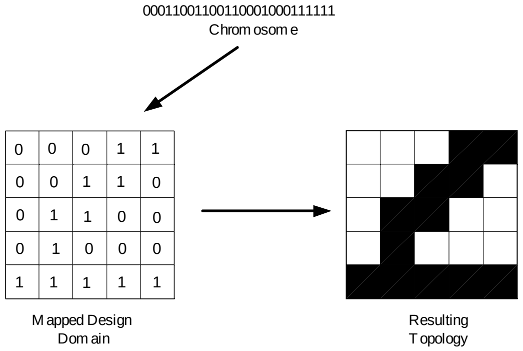
A target geometry matching problem is proposed to evaluate and compare the performance of the bit-array method and the aforementioned morphological representation method for representing geometry within a GA optimization procedure. A predefined structural geometry is assumed as the target geometry. This target should be finally reached (the error is minimized) by any effective chromosome representation method with the appropriate evolutionary optimization and the performance of the chromosome representation methods can be evaluated by their efficiency in solving this problem. Mathematically, the objective function to be minimized in this optimal target matching problem can be expressed as

$$f\left(\boldsymbol{x}\right)=\sum_{i=1}^{n}\left|x_i-\left(x_i\right)_O\right| \qquad (1)$$

where the design variable vector $\boldsymbol{x}$ is composed of the material density of each element, x_i , with a value of either 0 or 1 to represent that the element is either void or material, respectively. In other words, the objective function is defined as the summation of the absolute error in the material density of each element in the current geometry with respect to the target one, where $\left(x_i\right)_O$ is the material density of the *i*-th element of the target geometry.

B. Implementation of the bit-array method

The bit-array representation method uses one dimensional binary string as the chromosome to represent the distribution of material and void within a geometry. To convert such a bit-array genotype into a phenotype, the chromosome is first mapped into the design domain which is discretized into elements. The elements with allele values of 1 become material while those with allele values of 0 become void [11], as shown in Fig. 7.



0	0	0	1	1
0	0	1	1	0
0	1	1	0	0
0	1	0	0	0
1	1	1	1	1

Fig. 7. Bit-array chromosome and the resulting topology

The bit-array representation method, which is based on a given mesh, is a straightforward and natural representation method. Furthermore, the variable decoding step is nearly eliminated compared with that of other topological representation methods. However, this method does not prevent the formation of unanalyzable structures and checkerboard patterns. Additional steps must be taken to bias the formation of connected structures during the GA operations to allow the structural analysis based on the finite element method. In this present study, connectivity analysis is not introduced because this target matching optimization problem is a simulation of the structural topology optimization problem to evaluate the performance of the GA operations, and finite element analysis of the resulting structural geometry is not performed.

C. *Results and Discussion*

The target structural geometry shown in Fig. 8 is adopted as the target to evaluate the performance of each representation method. The population size is 400 and maximum number of generations 100. A generation gap of 0.1 and an elitist strategy are used.

Because the GA is a stochastic search method, the evaluation and comparison based on results from one single execution of the GA would become meaningless. In the present study, all the results are obtained by averaging the results from 10 independent runs.

The bit-array representation method with the uniform crossover and a randomly initialized population is tested first. Table II displays the effect of the mutation rate on the GA's performance based on the minimum error achieved by 100 generations, in which the relative error is the lowest error value (from Eq. (1)) divided by the total number of elements in the design domain. It can be seen that the best performance can be achieved when the mutation operator is vanished because the uniform crossover itself can create enough diversity of the chromosomes. The corresponding best topology and the convergence speed achieved without the mutation operation during these 10 runs are shown in Figs. 9 and 10. It can be seen from Fig. 9 that the best topology is very similar to the target one, but there are many elements isolated from the main body of the structure (which cannot be included if any structural analysis is performed) and also many instances of small voids alternating with material within the main body. It can also be noted that the relative error will converge with the increase of the number of generations.

TABLE II
MUTATION RATE EFFECT ON THE PERFORMANCE OF THE BIT-ARRAY GA

Mutation Rate	0	0.01%	0.1%	1%	10%
Relative Error	5.3%	6.7%	8.1%	21.6%	40.8%

The morphological representation method is next tested. Four Bezier curves are used, each of which has two control points only. The effects of the mutation rate on the GA's performance are shown in Table III.

TABLE III
MUTATION RATE EFFECT ON THE PERFORMANCE OF THE GA WITH MORPHOLOGICAL REPRESENTATION

Mutation Rate	10%	20%	30%	40%	50%
Relative Error	5.6%	4.4%	3.1%	3.2%	4.8%

It can be observed that only the higher mutation rates (10%-50%) are listed in Table III because they are more effective than the lower ones. The real-coded GA may take advantage of higher mutation rates which allow a greater diversity of the real-coded chromosomes, since the proposed crossover just allows limited diversity. This is consistent with Janikow and Michalewicz's finding [12], based on the comparison of binary and floating point representations in GAs. Comparing the results in Table III achieved by the GA using the morphological representation method with those in Table II achieved by the GA using the bit-array method and considering the fact that the bit-array method does not prevent the occurrence of isolated elements and alternating patterns of small voids (holes), it can be concluded that the morphological representation can outperform the bit-array representation and this encoding scheme is thus more effective to converge in a given number of generations. The corresponding best topology and the convergence speed achieved with a mutation rate of 30% during these 10 runs are shown in Figs. 11 and 12. It can be seen from Fig. 11 that the best topology is very similar to the target one and, different from the best topology achieved by the bit-array representation, there are no elements isolated or patterns of small voids, which means that the topology can be used directly for the structural analysis. It can also be noted that the relative error will converge with the increase of the number of generations, but the convergence speed is quite slow after the first 50 generations.

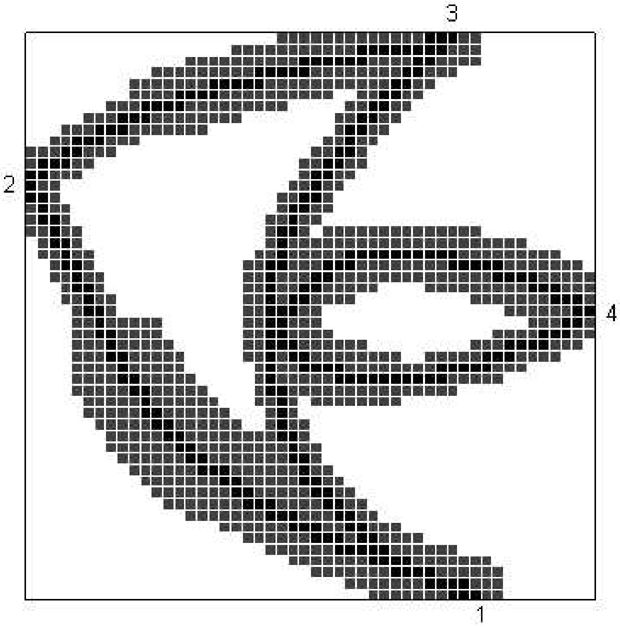


Fig. 8. Morphological representation of a structural geometry

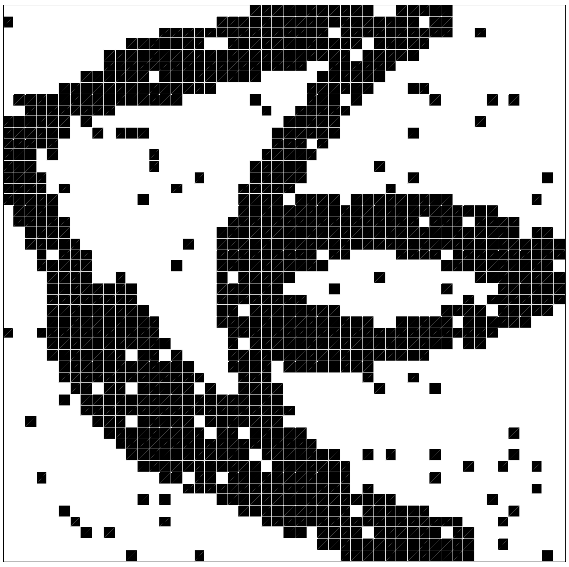


Fig. 9. Best Topology achieved by the GA with the bit-array representation

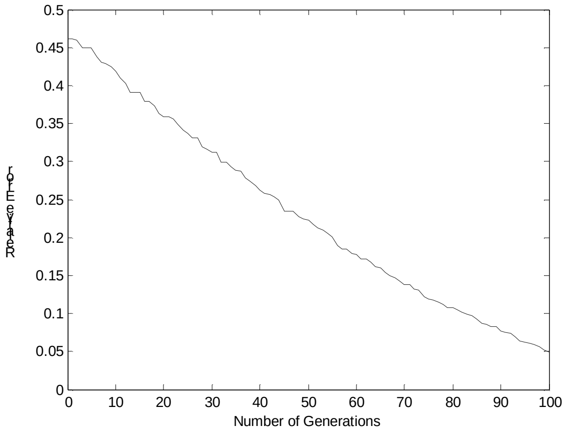


Fig. 10. Convergence speed of the GA with the bit-array representation

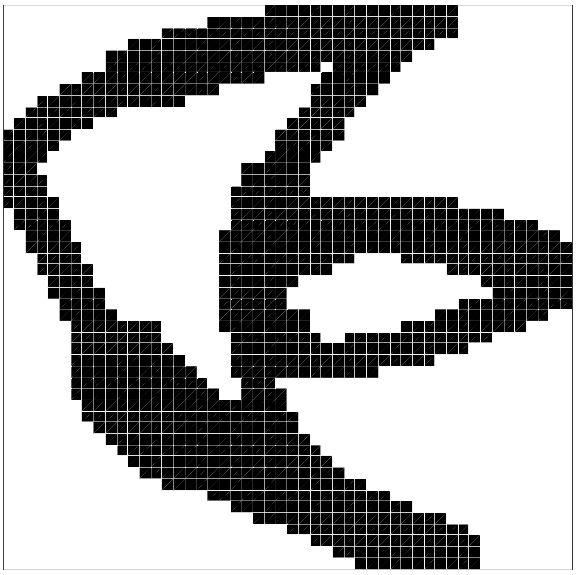


Fig. 11. Best Topology achieved by the GA with the morphological representation

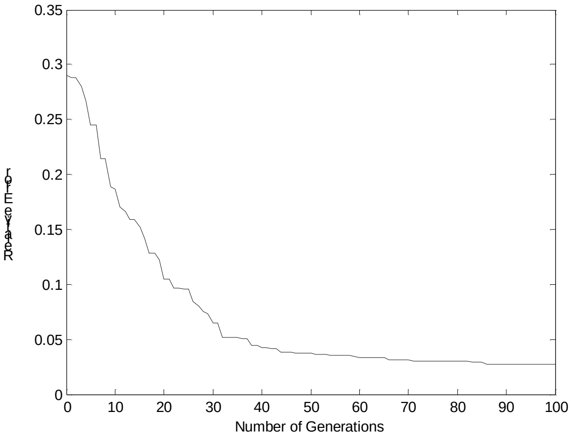


Fig. 12. Convergence speed of the GA with the morphological representation

VI. Results for Design of a Compliant Mechanism

The problem is to design a structure which deforms, when some part of it is given a prescribed displacement, such that another part is displaced along some desired path. This path generating mechanism problem to be solved is defined by a 100 by 100mm design space with the output point positioned as shown in Fig. 13. There is one support and one loading point making up a total of two input points, and the positions of these are variable but confined to the boundary (the support point is positioned anywhere along the bottom boundary while the loading point is positioned anywhere along the left boundary). The position of the output point is fixed (non-variable). Three connecting curves are used such that there is one connecting curve between any two points (i.e. every input/output point is directly connected to the other two), with each curve defined by three control points. All thickness values are not allowed to vary but are fixed at zero because the structures are expected to be thin and therefore flexible in order to achieve the large displacement and, furthermore, this helps to reduce the number of design variables and hence the solution space of the optimization problem. Wherever the input point is located (somewhere along the left boundary), a horizontal straight line input displacement of magnitude 30mm is to be applied in the direction shown in Fig. 13 (with displacement in the vertical direction restrained), while the desired path of the output point is a 30mm vertical straight line. This is partly similar to the path generating characteristics of the four-bar double-slider mechanism shown in Fig. 14.

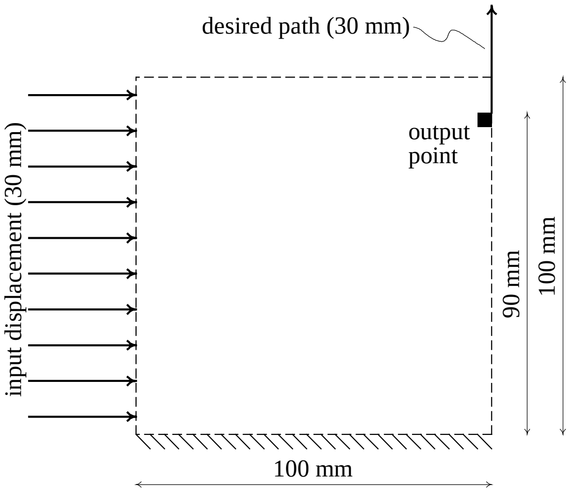


Fig. 13. Definition of design space

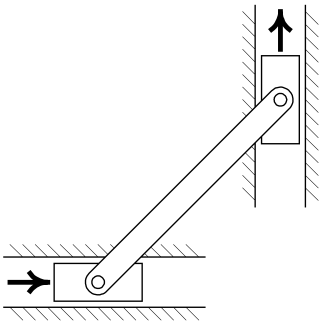


Fig. 14. Four-bar double-slider

For any given structural design, a geometrical non-linear (large displacement) FE analysis is performed to compute the incremental displacement steps at the output point which trace out the actual path. To evaluate how closely this actual path follows the desired path, the average deviation (error) between the two is evaluated. This is done by first dividing the desired path into N number of segments (where N is the total number of analysis steps resulting from the FE procedure) and in the same length proportions as that of the actual path (Fig. 15).

The distances d_i between corresponding points of the desired and actual paths are summed up and averaged and this average distance d_{ave} represents the deviation :

$$d_{ave} = \frac{1}{N}\sum_{i=1}^{N} d_i \qquad (2)$$

This d_{ave} value is taken directly as the objective function that is to be minimized in the optimization problem. The problem is thus solved with little pre-conceived notion of the geometric shape or topology, except for the position of the output point which is fixed relative to the design space, and the relative positions of the input and support points (which are variable but restricted to be along the boundary). The material assumed for the structure is polypropylene because of its ductility and high strength-to-modulus ratio, properties which are advantageous to applications in compliant mechanisms [13]. The Young's modulus assumed is 1140MPa with Poisson's ratio of 0.4 (and the yield strength of polypropylene is about 34MPa). The design space is discretized into a 50 by 50 mesh of 4-noded quadrilateral elements, and any chromosome-encoded design generated by the evolutionary process can be translated into an FE model of the structure according to the morphological representation. The large displacement FE analysis is performed using the ABAQUS/Standard software.

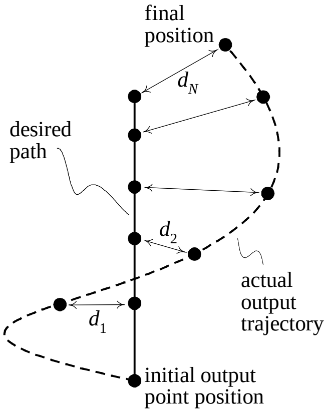
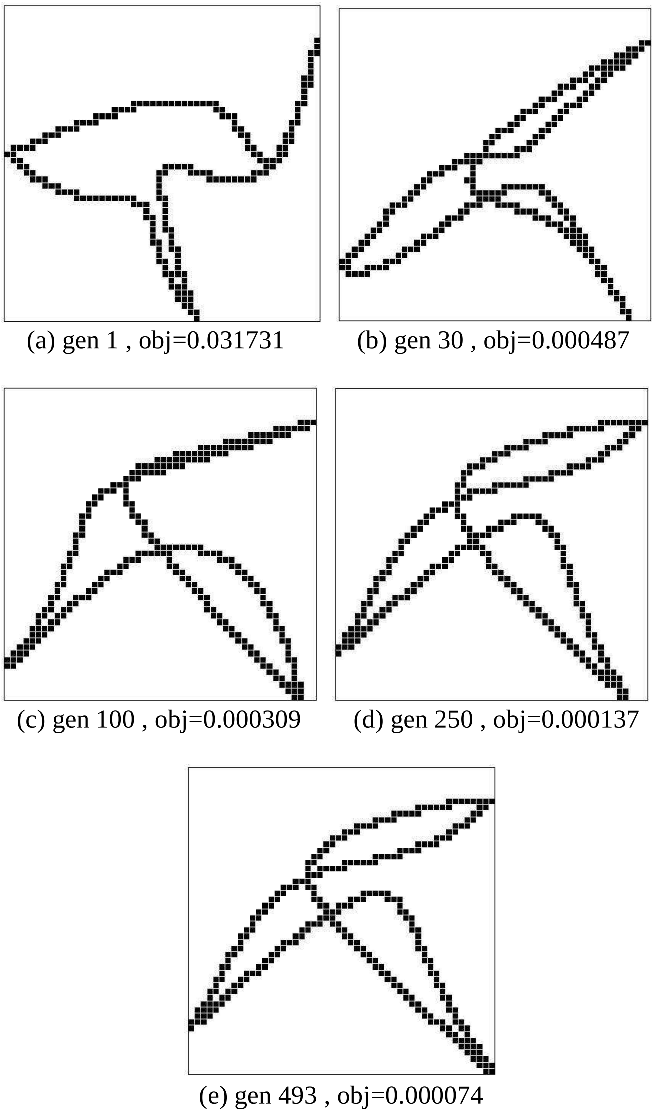


Fig. 15. Deviation between desired and actual paths

The evolutionary optimization was run for 500 generations (with a population size of 200 per generation), by the end of which 10935 objective function evaluations have been performed. The average CPU time consumed for each function evaluation (FE analysis) is between 15 to 20 seconds. The optimum design was attained at the 493rd generation, with an objective function value of 0.000074. The force needed to exert the input displacement is 3.35N, and the peak von Mises stress is 45.3MPa which occurs at the support point. A glimpse of the evolution history is illustrated in Fig. 16 by a sampling of the designs obtained at their respectively indicated generation, with their corresponding objective function (deviation) values. Fig. 16(a) shows a sample design from the initial (1st) generation, (b) to (d) are the best designs achieved up to the respectively indicated generations, and (e) is the final optimum at generation 493. Fig. 17 shows that the actual path achieved by the optimum design is very close to the required straight line desired path. Fig. 18 shows the deformed and undeformed geometries of the optimum design. The resulting designs obtained here are clearly defined due to the discrete nature of the geometry representation, unlike in the homogenization or material density methods of topology optimization which require the prescription of some threshold point to interpret whether the resulting material density values in the elements indicate solid material or void. However, the resulting designs obtained are jagged-edge structures due to the finite element discretization/grid. Therefore future work that needs to be done include the implementation of some form of curve-fitting or smoothing techniques to parameterize the structural boundaries. A secondary shape optimization may then be performed (starting from the optimum design) to fine-tune the shape. This may be necessary as the peak von Mises stress of 45.3MPa (at the support point) exceeds the yield strength (34MPa) of the assumed material. The high stresses may be reduced by fine-tuning the shape and thickness near the support region. Another alternative is to solve the more complex constrained optimization problem (with constraints on stresses) by incorporating constraint handling capability into the GA procedure.


(a) gen 1 , obj=0.031731

(b) gen 30 , obj=0.000487

(c) gen 100 , obj=0.000309

(d) gen 250 , obj=0.000137

(e) gen 493 , obj=0.000074

Fig. 16. Evolution of designs

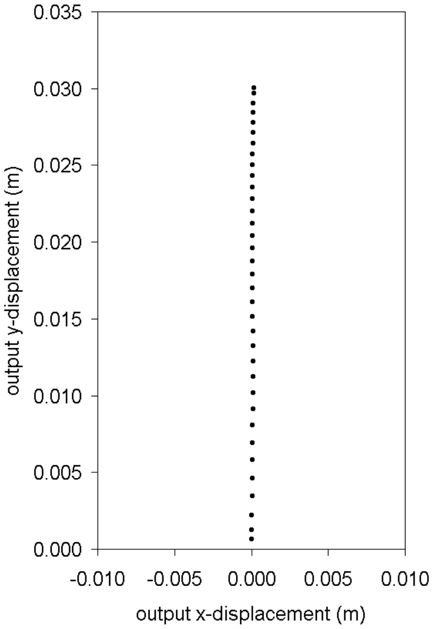


Fig. 17. Plot of actual path

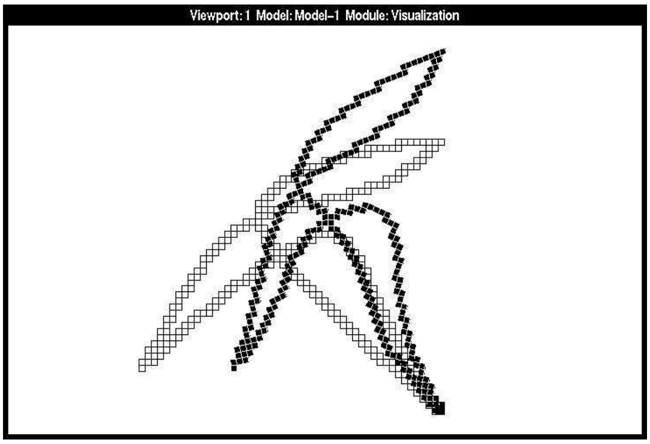


Fig. 18. Deformed and undeformed geometry of optimum design

VII. CONCLUDING REMARKS

An effective topology/shape design optimization methodology has been evaluated via a simulated topology optimization problem and then applied to an actual structural problem of automatically synthesizing a path generating compliant mechanism which produces a vertical straight line output path when given a horizontal straight line input. The success of the method hinges on the use of an efficient morphological representation scheme which defines practical and valid structural geometries, a compatible chromosome encoding system together with genetic operators of crossover and mutation that capture the topology/connectivity of the structures, and a genetic algorithm to treat the problem as a discrete optimization problem.

REFERENCES

[1] M. P. Bendsoe and N. Kikuchi, "Generating optimal topologies in structural design using a homogenization method," *Comput. Meth. Appl. Mech. Engng*, vol. 71, pp. 197-224, 1988.

[2] R. J. Yang and C. H. Chuang, "Optimal topology design using linear programming," *Computers & Structures*, vol. 52, pp. 265-275, 1994.

[3] C. D. Chapman, K. Saitou, and M. J. Jakiela, "Genetic Algorithms as an Approach to Configuration and Topology Design," *ASME Journal of Mechanical Design*, vol. 116, pp. 1005-1012, 1994.

[4] K. Tai and T. H. Chee, "Genetic Algorithm with Structural Morphology Representation for Topology Design Optimization," in *Mechanics in Design - Proceedings of the 2nd International Conference on Mechanics in Design*, Nottingham, U. K., 1998, pp.827-836.

[5] K. Tai, G. Y. Cui, and T. Ray, "Design synthesis of path generating compliant mechanisms by evolutionary optimization of topology and shape," *ASME Journal of Mechanical Design*, vol. 124, pp. 492-500, 2002.

[6] T. Ray, K. Tai, and K. C. Seow, "Multiobjective Design Optimization by an Evolutionary Algorithm," *Engineering Optimization*, vol. 33, pp. 399-424, 2001.

[7] K. Tai and T. H. Chee, "Design of structures and compliant mechanisms by evolutionary optimization of morphological representations of topology," *ASME Journal of Mechanical Design*, vol. 122, pp. 560-566, 2000.

[8] S. Akhtar, K. Tai, and J. Prasad, "Topology optimization of compliant mechanisms using evolutionary algorithm with design geometry encoded as a graph," presented at the 2002 ASME Design Engineering Technical Conferences, Montreal, Canada, September 29 - October 2, 2002, Paper DETC2002/DAC-34147.

[9] J. Gross and J. Yellen, *Graph theory and its applications*. Boca Raton: CRC Press, 1999.

[10] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed. Berlin: Springer, 1996.

[11] C. D. Chapman and M. J. Jakiela, "Genetic algorithm-based structural topology design with compliance and topology simplification considerations," *ASME Journal of Mechanical Design*, vol. 118, pp. 89-98, 1996.

[12] C. Z. Janikow and Z. Michalewicz, "An experimental comparison of binary and floating point representations in genetic algorithms," *Proc. ICGA 4*, pp. 31-36, 1991.

[13] L. L. Howell, *Compliant Mechanisms*. New York: John Wiley, 2001.