Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers and Structures 82 (2004) 1609–1622

Computers & Structures

www.elsevier.com/locate/compstruc

# Graph representation for structural topology optimization using genetic algorithms

S.Y. Wang, K. Tai *

*Centre for Advanced Numerical Engineering Simulations, School of Mechanical and Production Engineering, Nanyang Technological University, 50 Nanyang Ave, Singapore 639798, Singapore*
*Singapore–MIT Alliance, 4 Engineering Drive 3, Singapore 117576, Singapore*

Received 15 September 2003; accepted 6 May 2004

**Abstract**

This paper proposes a graph representation for evolutionary structural topology design. Based on graph theory, a valid topology in the design domain is represented by a connected simple graph and each edge of the graph is defined by a cubic Bézier curve with varying thickness. The graph is defined to have connecting paths between loading regions and support regions of the structure, to ensure a physically meaningful connected structure. Each path is formed by one or more Bézier curves in order to allow more local control of the shape. A real-valued chromosome encoding and decoding scheme and a partition crossover method are also developed based on graph theory. The graph representation GA is applied to structural topology optimization problems and its performance is compared with those of other methods. Compared with the power-law approach, the present graph representation GA can generate clearly defined and distinct geometries and perform a global search, but it requires more computational cost. The numerical results also demonstrate the improved performance of the present graph representation over that of the Voronoi-bar representation in terms of connectivity of the geometry and convergence speed. It is suggested that this graph representation method is both physically meaningful and computationally effective in the framework of topological optimum design using GAs.

*Keywords:* Structural topology optimization; Graph representation; Graph crossover; Design connectivity; Genetic algorithms; Graph theory

## 1. Introduction

Topology optimization of solid structures is one of the main subset approaches of structural optimization which aims to find the best possible structure that meets different multidisciplinary requirements such as functionality and manufacturing [1]. An optimal structural topology can be arrived at by the optimal modifications of holes and connectivities of the structural design domain, which is actually implemented by redistributing material in an iterative and systematic manner. Generally, structural topology optimization is a powerful tool which can help the designer select suitable initial structural topologies and furthermore, it is identified as economically the most rewarding task in structural design [2–4].

Structural topology optimization as a generalized shape optimization problem has received considerable attention recently [2]. Various families of structural

* Corresponding author. Address: Centre for Advanced Numerical Engineering Simulations, School of Mechanical and Production Engineering, Nanyang Technological University, 50 Nanyang Ave, Singapore 639798, Singapore. Tel.: +65-679-04444; fax: +65-679-11859.

*E-mail addresses:* sywang@ntu.edu.sg (S.Y. Wang), mktai@ntu.edu.sg (K. Tai).

0045-7949/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compstruc.2004.05.005

topology optimization methods have been well developed [5]. Bendsøe and Kikuchi [3] proposed a homogenization method, in which the material is represented by a sponge-like material with infinite microscale cells with voids and the material throughout the structure is redistributed by using an optimality criteria procedure. As an important alternative approach of this family of homogenization methods, the power-law approach [6], which is also called the solid isotropic microstructure with penalization (SIMP) method [7] and originally introduced by Bendsøe [8], has got a fairly general acceptance during the past few years [2]. It adopts the element relative densities as the design variables and assumes that the material properties within each element are constant, which are modeled as the relative material density raised to some power times the material properties of solid material [6]. Xie and Steven later introduced an evolutionary structural optimization (ESO) method, in which the material in a design domain which is not structurally active is considered as inefficiently used and can thus be removed by using some element rejection criteria [4]. Both the homogenization method and the ESO method have been further developed by a large number of researchers, leading to extensive exposition and exploration of these two families of methods. Although computationally effective, both cannot perform a global search and thus do not necessarily converge to the global optimal solution for the given objective function and constraints [2,9].

Another emerging family of structural topology optimization methods is the one using the genetic algorithms (GAs), which are based on Darwinian survival-of-the-fittest principle to mimic natural biological evolution. GAs are gradually recognized as a kind of powerful and robust stochastic global search method [2,10], thanks to the seminal work of Holland [11] and the comprehensive study of Goldberg [12]. More recently, GAs have been increasingly employed in the structural topology optimization field in order to perform a global search in the design domain [10,13–23].

It is well known that for the GAs, the definition of the search space (or correspondingly the choice of a representation method) is of vital importance [15]. Currently, the bit-array representation method is the most widely adopted [10,13,14,16,17] topological representation method. It is a straightforward representation and the decoding step is virtually eliminated. A bit-string or bit-array with 1 and 0 is used to represent the two-dimensional design domain discretized by a fixed mesh, where each of the small, square elements contains either material or void and is thus treated as a binary design variable. However, the complexity of the resulting topology is dependent on that of the given mesh. Hence, for a fine mesh, the bit-array representation method not only may require huge computational cost but also may fail to produce properly connected structures for the finite element analysis (FEA) at the initial stage, as demonstrated by Hamda et al. [23]. Another representation method is the Voronoi-based representation [10,14,15,23], first introduced by Schoenauer [14]. A finite number of Voronoi sites being labeled 0 or 1 are used to define the Voronoi diagram and to represent a partition of the design domain into two subsets and thus the Voronoi representation of shapes and topologies does not depend on the mesh that will be used to compute the behavior of the shapes [14]. However, it also cannot suppress the occurrence of the unwanted disconnected topologies, checkerboard pattern and unbounded Voronoi sites. Furthermore, it may be unable to find satisfactory solution for the problem of a cantilever beam, similarly to the bit-array representation [23]. Another relatively new representation method is the morphological representation proposed by Tai and Chee [18,19]. Simple parametric curves (Bézier curves) with varying thickness to connect the input/output (I/O) regions are used to represent the topology and shape in the two-dimensional design domain. The connectivity of the underlying topology can be guaranteed and the problem of undesirable checkerboard patterns is practically eliminated. In spite of its success in solving those problems of topological optimum design of compliant mechanisms [18–21], the morphological representation is essentially an intuitive method without a strong mathematical or theoretical background. Furthermore, the complexity of the resulting topology would greatly rely on the complexity of the connection curves.

In the present study, the morphological representation method is further developed into a graph-theoretic representation method. Based on graph theory, a valid topology in the two-dimensional design domain is represented by a connected simple graph consisting of vertices and simple undirected cubic Bézier curves with varying thickness. Connectivity requirement can be guaranteed and the complexity of the resulting topology mainly depends on the number of vertices and their distribution in the design domain. A partition crossover method is proposed based on the connectivity of each partitioned subgraph. Application of this graph representation into the constrained structural optimization problems by using the GA are implemented and numerical experiments are demonstrated to compare results with those from other methods. It is suggested that this representation is both physically meaningful and computationally effective in the framework of topological optimum design using GAs, and the benefits of the proposed method include the guaranteed connectedness of the structure, a topology complexity independent of the underlying mesh, and a stochastic global optimization approach.

## 2. Graph representation of structural geometry

### 2.1. Graph representation

Generally, for the structural topology optimization problems, the structure must be fully connected to bear loadings and so ensure that the numerical calculations based on the FEA is then possible. Hence, for each loading region of the structure, at least one path must exist to transfer the external loading to the support region. The structure may also include other regions where functional or response conditions may apply. As demonstrated in Fig. 1, the morphological representation scheme [18–21] specifies paths joining one region to another. These paths can therefore be interpreted as load paths. Each path is defined by a Bézier curve with a varying thickness. The complexity of the resulting topology can be controlled by not only the number of paths but also the complexity of the Bézier curve defining each path. Bézier curves are widely used for creating curvilinear shapes in many fields of design nowadays since the Bézier curve is very effective to control the shape of the curve even though the control points do not lie on the curve in general. However, only global control is possible, i.e. if one of the control points are moved, the whole shape of the curve will be affected [24]. Since it does not offer enough local control over the shape of the curve, the Bézier curve is relatively inflexible. Furthermore, the higher-order Bézier curve will become computationally more expensive considering the fact that the order of the Bézier curve is related to the number of control points.

It is well recognized that graph theory is useful for modeling a wide variety of real-world situations and for the efficient analysis of complex practical structures particularly suitable for computational mechanics [25–27]. It has been shown that optimal analysis of skeletal structures or networks is feasible by using various concepts of graph theory in [25–30]. In the present study, a graph representation based on graph theory is proposed to overcome those drawbacks of higher-order Bézier curves. Piecewise connected simpler Bézier curves, rather than a single higher order Bézier curve, are used for defining each joining path. Only the cubic Bézier curve, which is the most common form of Bézier curves, is used for each section of Bézier curves. It consists of two endpoints and two control points and it is the simplest form that allows a curve to have an inflection point [24], as shown in Fig. 2, in which the tangent at the endpoints is along the line to the middle two points. This helps control local conditions, normally changing the position of one control point will affect the whole curve. According to graph theory [31], the joining paths can thus be represented by a graph $G = (V, E)$, where the vertex set $V$ is the set of endpoints and edge set $E$ the set of small sections of cubic Bézier curves. Each edge of $E$ is undirected due to the symmetry property of the Bézier curves [24]. Those vertices and edges construct a connected simple graph $G$ to represent a connected topology in the design domain. Fig. 3 displays a graph representation of those paths as shown in Fig. 1. Path 1 is formed by three edges $ad$, $de$ and $ec$, while path 2 by two edges $bf$ and $fc$, in which $\{a, b, \dots, f\}$ is the vertex set and the support regions are represented by vertices $a$ and $b$ and the loading region by vertex $c$. Each edge is actually a cubic Bézier curve, as shown in Fig. 2. To represent a more complicated topology, more paths joining one region to another and more piecewise cubic Bézier curves for each path may be required. It is evident that a more complicated topology can be represented by a more refined connected simple graph, which can be constructed by the addition of more vertices and paths, though the

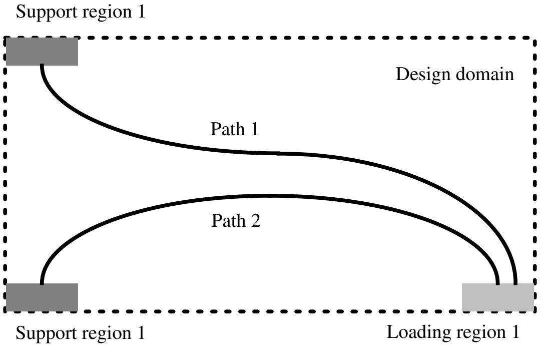


Fig. 1. Morphological representation of structural geometry.

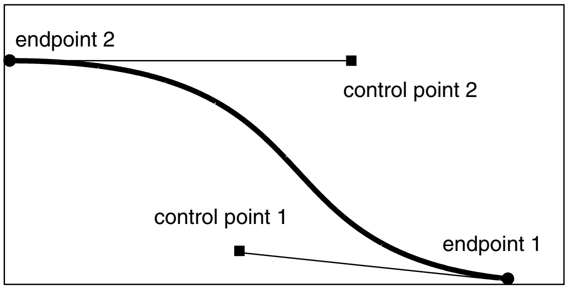


Fig. 2. Cubic Bézier curve.

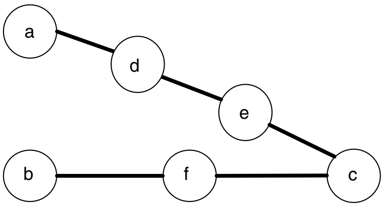


Fig. 3. Graph representation for structural geometry of Fig. 1.

computational cost when using the GA would become much higher.

### *2.2. Real-valued chromosome encoding and decoding scheme*

To implement the above-mentioned graph representation, a real-valued chromosome encoding and decoding scheme is proposed based on the connectivity of a simple graph. The chromosome is assumed to be with a fixed length and thus a variable-length chromosome encoding strategy such as the messy GA developed by Goldberg et al. [32] is beyond the scope of the present study.

Connectivity is an important issue for optimal structural topology design. Shai and Preiss [33] used the isomorphic representation to perform a systematic evaluation of the well-formedness (rigidity and stability) of a structural system before commencing the analysis. Kawamura et al. [34] proposed a methodology based on connected triangular elements to generate stable truss topologies. Bhardwaj and Day [35] presented a technique to improve the partitions by using a carefully calibrated set of element weights to maintain load balance and adding extra subdomains to ensure the face connectivity of the subdomains. Singular subdomain matrices are avoided but the computational cost is too heavy even for the powerful CPLANT platform [35]. Kaveh and Kalatjari [30] developed a penalty scheme to ensure that topologically unstable structures and the corresponding chromosomes are highly penalized for the size optimization design of truss structures. In the present study, it is assumed that a constant connectivity relationship between the loading regions and the support regions exists among all the individuals in the population of any generation and therefore the chromosome encoding and decoding scheme as well as other GA operations can be performed based on the same connectivity matrix. It should be noted that the connectivity matrix is always symmetric [31] since each present graph representation topology is actually a simple undirected graph, as mentioned before.

According to the graph representation afore-mentioned, two groups of design variables should be included in the real-valued chromosome: the vertices consisting of all the endpoints of Bézier curves and the control points and thickness values of each Bézier curve. Hence, for a graph consisting of $N$ vertices and $M$ edges, the corresponding chromosome is encoded as

$$[\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_N, \mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_M] \tag{1}$$

in which $\mathbf{P}_i = (x_i, y_i)$ for $i = 1, 2, \ldots, N$ is the Cartesian co-ordinates of the $i$th vertex, and $\mathbf{B}_j = [t_1^j, x_2^j, y_2^j, t_2^j, x_3^j, y_3^j, t_3^j]$ for $j = 1, 2, \ldots, M$ includes the data of the thicknesses $t_1, t_2$ and $t_3$ (three thickness values, rather than one constant value, are assigned to each Bézier curve [19]) and the Cartesian co-ordinates of the two control points of the $j$th cubic Bézier curve.

To map the real-valued chromosome into a topology in the design domain, a decoding scheme is also developed. Based on the connectivity matrix, the Cartesian coordinate vectors of the endpoints and control points of each cubic Bézier curve can be directly obtained from the real-valued chromosome in Eq. (1) and therefore the cubic Bézier curve can be uniquely determined by

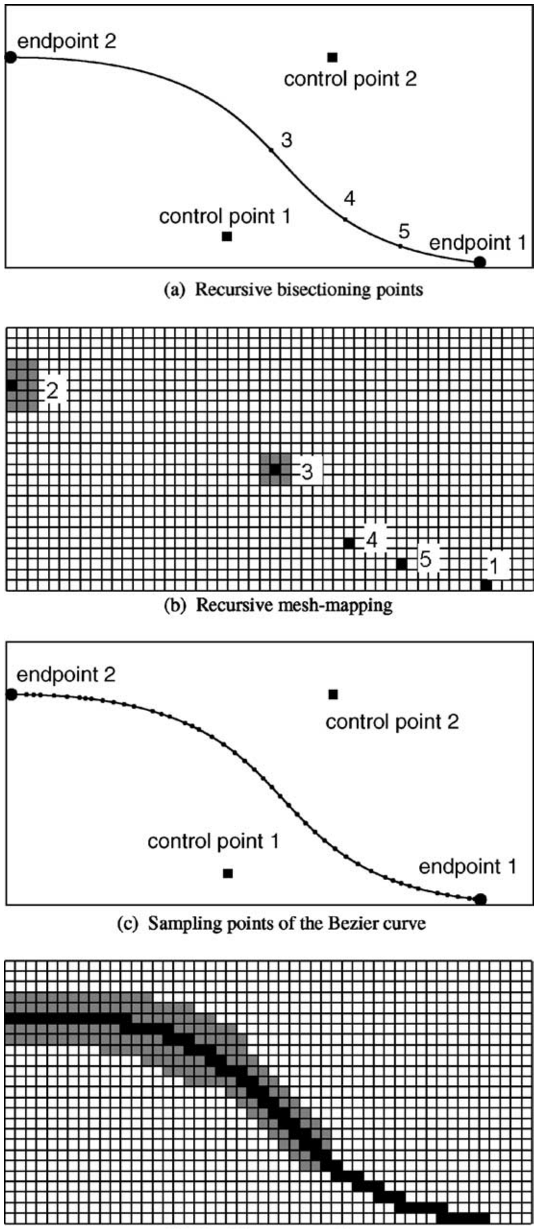
$$\begin{aligned}\mathbf{r} &= \mathbf{r}(u) \\ &= \sum_{k=0}^{3} \frac{6}{k!(3-k)!} u^k (1-u)^{3-k} \mathbf{r}_k, \quad 0 \leqslant u \leqslant 1,\end{aligned} \tag{2}$$

in which $\mathbf{r} = (x, y)$, $\mathbf{r}_0$ and $\mathbf{r}_3$ the co-ordinate vector of the two endpoints, $\mathbf{r}_1$ and $\mathbf{r}_2$ the co-ordinate vector of the two control points, and $u$ the intrinsic co-ordinate along the curve.

However, the decoding step to map the cubic Bézier curve in Eq. (2) with varying thickness into the finite element (FE) mesh is more computationally involved. To avoid mapping the connected curve into disconnected topology in the FE mesh, the zero-thickness Bézier curve is mapped such that each element which it passes through becomes solid. To reduce the computational cost significantly, the number of sampling points of the Bézier curve within each mapped solid element should not be more than 1. To implement this requirement, a recursive bisection mapping method is proposed. As shown in Fig. 4, after mapping the endpoints of the cubic Bézier curve, points that recursively bisect the Bézier curve are computed and mapped (Fig. 4(a) and (b)). The recursion is terminated when the mapped element of the bisection point is the same as that of any current endpoint of the bisected segment. Based on this method, the final sampling points of the cubic Bézier curve are shown in Fig. 4(c). Fig. 4(d) shows the resulting structural geometry after layers of elements (shown in grey for the purpose of illustration only) are added to segments of the Bézier curve and the number of layers is based on the corresponding thickness values [19].

### *2.3. Crossover and mutation operations*

Crossover (recombination) is the most important GA operator since it plays a central role in GAs and constitutes one of the major differences between GAs and other optimization algorithms, as emphasized by Holland's schema theorem [11]. Whether a search space is particularly suited to a GA greatly depends on the power of crossover in generating an appropriate search bias. In the present work, a partition crossover method is proposed in order that each child would become an



(a) Recursive bisectioning points

(b) Recursive mesh-mapping

(c) Sampling points of the Bezier curve

(d) Resulting structural geometry

Fig. 4. Recursive bisection mapping method.

improvement over the parents with relatively high probability.

According to the graph representation, each topology in the design domain is represented by a connected simple graph. Connectivity is also a prerequisite to perform the objective function evaluation based on the FEA for structural optimization. This can be guaranteed in the present graph representation since all the GA operations are based on the same connectivity matrix, as mentioned before. Since it is required that the crossover operator for GAs may generate children superior to both of their parents with high probability, it would be desirable that the children inherit good substructure of their parents properly [36]. The graph crossover is a quite common crossover method for data structures of graphs by breaking edges. Nagata and Kobayashi [36] proposed the edge assembly crossover for the traveling salesman problems. Sen et al. [28] used an elementary tree transformation or cyclic interchange as the crossover operation for the design of sensor network. Globus et al. [37] developed a general-purpose crossover operator for directed and undirected graphs to evolve molecules and circuits as long as there are rings in the population. In the present graph representation, a connected subgraph of the graph, rather than a single vertex as in [21], is chosen as a good substructure for the children to inherit. Since the connected subgraph is randomly selected from the graph, the geometric characteristic that each child can inherit from each parent is not fixed. It is suggested that this crossover would be more flexible and effective to facilitate combination of low-order building blocks to form higher-order building blocks and thus the probability that the children can inherit better properties of both parents is relatively high. The partition crossover procedure is illustrated by an example in Fig. 5, in which child 1 inherits a connected subgraph ($V_2' = \{B_2, D_2, E_2\}$) from parent 2 and child 2 inherits a connected subgraph ($V_1' = \{B_1, D_1, E_1\}$) from parent 1. It should be noted that the exact points at which the subgraph is partitioned from the rest of the whole graph are chosen randomly and the connectivity analysis can be performed by searching for a spanning tree in the subgraph. According to graph theory, a graph is connected if and only if it has a spanning tree [31,38]. To search for a spanning tree, the depth-first search method is used based on the afore-mentioned connectivity matrix. Since outgoing edges of a vertex is considered before any neighbors of the vertex in the search, extremes are searched first and this can be implemented with recursion.

On the other hand, mutation is usually used as a background GA operator to enforce a random walk in the design domain so that the probability of searching any given point in this domain will never be zero and the diversity in the population will be increased. Generally, mutation is randomly applied with low probability (0.001–0.01). However, as suggested by Janikow and Michalewicz [39], the real-coded GA may take advantage of higher mutation rates. The reason is that the real-coded GA does not provide enough diversity through the crossover operation alone. In the present study, a real-value mutation method is adopted and the mutation is achieved by the random selection of a new real value within the allowable range of each design variable with a relatively high mutation rate (0.1–0.5), similar to that of the Breeder genetic algorithm (BGA)

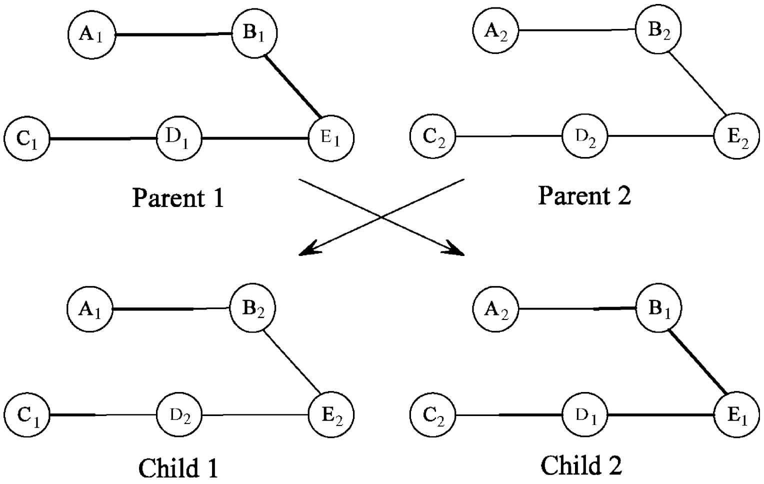


Fig. 5. Partition crossover method.

[40]. The higher mutation rates increase the level of possible exploration of the search space without adversely affecting the convergence characteristics.

## 3. Optimization procedure using genetic algorithms

### *3.1. Artificial unconstrained objective function of constrained optimization*

The classical structural topology optimization problems are with a single objective either to minimize the compliance with a constraint on the volume fraction or to minimize the weight with a constraint on the maximum displacement. Generally, the single-objective constrained structural topology optimization problem can be written as follows:

$$\begin{aligned}\text{Minimize:}\quad & f(\mathbf{x}),\quad \mathbf{x}\in\mathscr{R}\\ \text{Subject to}\quad & g_i(\mathbf{x})\geqslant 0,\quad i=1,2,\ldots,I\\ & h_j(\mathbf{x})=0,\quad j=1,2,\ldots,J,\end{aligned}\tag{3}$$

where $\mathbf{x}$ is the solution vector, $\mathscr{R}$ the design domain, $I$ the number of inequality constraints, $J$ the number of equality constraints, $f(\mathbf{x})$ the objective function, $g_i(\mathbf{x})$ the $i$th inequality constraint function, and $h_j(\mathbf{x})$ the $j$th equality constraint function.

Since GAs are generic search methods, most applications of GAs to constrained optimization problems have used the penalty function constraint handling approach [10,23,41–43]. The major difficulty is to set the right value for penalty parameters in order to obtain feasible best individuals in the population. In the present study, a constraint handling method developed by Deb [41], which is also based on the penalty function approach but does not require any penalty parameter, is adopted. The main idea of this method is to use a tournament selection operator and to apply a set of criteria to decide the selection process:

1. Any feasible solution is preferred to any infeasible solution.
2. Between two feasible solutions, the one having better objective function value is preferred.
3. Between two infeasible solutions, the one having smaller constraint violation is preferred.

According to these criteria, an artificial unconstrained objective function of the constrained optimization in Eq. (3) can be constructed as

$$F(\mathbf{x})=\begin{cases} f(\mathbf{x}) & \text{if } \mathbf{x}\in\mathscr{F},\\ f_{\max}+\mathrm{viol}(\mathbf{x}) & \text{otherwise,}\end{cases}\tag{4}$$

where $F(\mathbf{x})$ is the artificial unconstrained objective function, $\mathscr{F}$ the feasible region of the design domain $\mathscr{R}$, $f_{\max}$ the objective function value of the worst feasible solution in the population, and $\mathrm{viol}(\mathbf{x})$ the summation of all the violated constraint function values. However, if there are no feasible individuals existing in the population, a pre-defined worst value, rather than 0 as recommended in [41], is assigned to $f_{\max}$ in order to allow the GA to use an elitist strategy.

### *3.2. Rank-based fitness assignment*

According to the value of the artificial unconstrained objective function in Eq. (4) of each individual, the population is sorted by using a rank-based fitness assignment method in this study. The fitness assigned to each individual depends only on its position in the individuals rank and not on the actual objective function value. The reproductive range is limited so that no individuals generate an excessive number of offspring. Rank-based fitness assignment usually behaves in a robust manner. In the present study, Baker's linear ranking algorithm [44] with a selective pressure of 2 is used. The fitness of each individual in the population is defined as

$$\widehat{F}(\mathbf{x}_i) = \frac{2(n_i - 1)}{N_{\text{ind}} - 1}, \qquad (5)$$

where $\widehat{F}(\mathbf{x}_i)$ is the fitness of the $i$th individual, $n_i$ the position of the $i$th individual in the individuals rank, and $N_{\text{ind}}$ the population size.

### 3.3. SUS selection method

The stochastic universal sampling (SUS), proposed by Baker [45], is a method for selecting a population in a way to minimize chance fluctuations. SUS can ensure a selection of population which is closer to what is deserved than the roulette wheel selection. SUS also has the advantage of being easier and quicker to implement. Therefore, although the roulette wheel selection has been used by most researchers, SUS is adopted as the preferred selection method in the present study. In the SUS method, the individuals are mapped into contiguous segments of a line, each of which is equal in size to its assigned fitness value exactly. Equally spaced pointers are then placed over the line as many as the required number of individuals to be selected. In the case where $N_p$ individuals are to be selected, the distance between every two adjacent pointers should thus be $1/N_p$ and the position of the first pointer can be given by a randomly generated number in the range $[0, 1/N_p]$.

## 4. Results and discussion

With the introduced GA fitness evaluation, selection, crossover and mutation operators, the proposed graph representation GA can now be applied to evolutionary structural topology design problems. Two examples are presented in order to demonstrate the performance of the present graph representation. Unless otherwise stated, the following settings are used in the numerical experiments presented below: standard simple GA evolution with a population size of 50 and a generation gap of 0.9 and an elitist strategy to keep the fittest individuals; the initial population is generated by a random number generator that uniformly distributes numbers in the allowable range; all runs are stopped after 100 generations; all the runs are carried out using MATLAB; all the finite element analyses are based on Sigmund's corresponding FEA code in [6], in which a planar stress square element is used; and all plots are the results of 20 independent runs. As to the physical property of the plates, it is also assumed that $Et = 1$, where $E$ is Young's elasticity modulus and $t$ the thickness of the plate.

### 4.1. Example 1: minimum compliance design

As shown in Fig. 6, the minimum compliance design problem is a 2 × 1 cantilever plate with the left boundary as the fixed support and a unit point force applied vertically downward at half-height of the right boundary. The plate is discretized by a 24 × 12 FE mesh with square elements, and compliance is minimized subject to the constraint that volume fraction does not exceed 50%. Numerical results from the present graph representation by using the GA are compared with those from the power-law approach using Sigmund's 99 line topology optimization code written in MATLAB [6]. In the graph representation GA, two different graphs are used to perform the numerical experiments, as shown in Fig. 7,

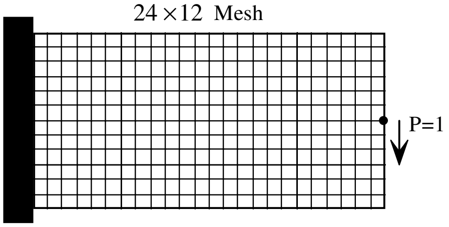
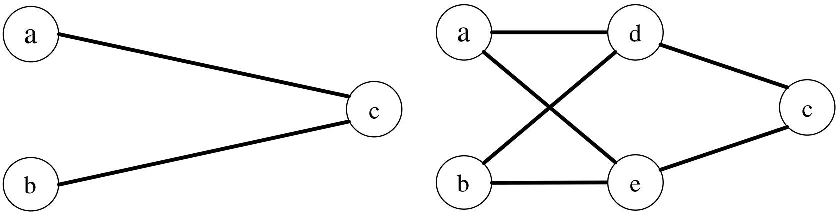


Fig. 6. Cantilever plate for the minimum compliance design.



(a) Graph with 3 vertices and 2 edges

(b) Graph with 5 vertices and 6 edges

Fig. 7. Two graph representations for minimum compliance design.

in which the positions of vertices *a* and *b* are variable only along the left boundary, the position of vertex *c* is fixed (loading point), and other vertices and the control points of each edge are fully variable within the design domain. For the more complicated graph as shown in Fig. 7(b) a larger population size of 100 and maximum generation number of 200 have been used. Fig. 8 displays a sampling of the resulting topologies from the initial generation (note that the population of this generation is randomly generated) based on the 5-vertices graph in Fig. 7(b) These show the diversity of the geometries that can be defined by the graph representation scheme. In the numerical experiments, a mutation rate of 0.3 is used.

Fig. 9 displays the best topology that is obtained from the power-law approach while Figs. 10 and 11 show the best topologies obtained from the present graph representation GA with 3 vertices and 5 vertices, respectively. Their corresponding minimum compliance values are 74.118, 79.279 and 77.742, respectively. As expected, the present graph representation GA can obtain lower minimum compliance by using more vertices. It can be seen that the resulting topology from the power-law approach displays a blurred boundary with gray elements (Fig. 9) while those from the present graph representation are distinct black and white designs (Figs. 10 and 11). The power-law approach of Sigmund's [6] did not eliminate the blurred boundary completely (even though a filtering scheme has been applied to alleviate this problem), since the mass density of each element is assumed to be a continuous design variable and intermediate densities would inevitably appear. Although it is still possible to achieve a distinct black and white design by introducing penalties on intermediate densities, it is questionable that the penalty or truncation techniques involved may not affect the optimal solution significantly since the optimality criteria cannot be satisfied [46]. Usually, it is left to the designer to use a particular approach to finish the post-processing task [6,46]. On the other hand, it can also be seen that the present two graph representations converge to similar optimal topologies with distinct solid material and void regions since the GA is capable of performing a discrete global search. However, the no-free-lunch principle [47], i.e. no method is generally the best, still applies here. The power-law approach converges to its optimal solution after only 30 FEA function evaluations, while the two graph representation GAs reach their best values after 4415 and 13,420 function evaluations (97 and 148 generations), respectively. This is consistent with the observation made by Jakiela et al. [16] that a GA-based solution may require 10–100 times

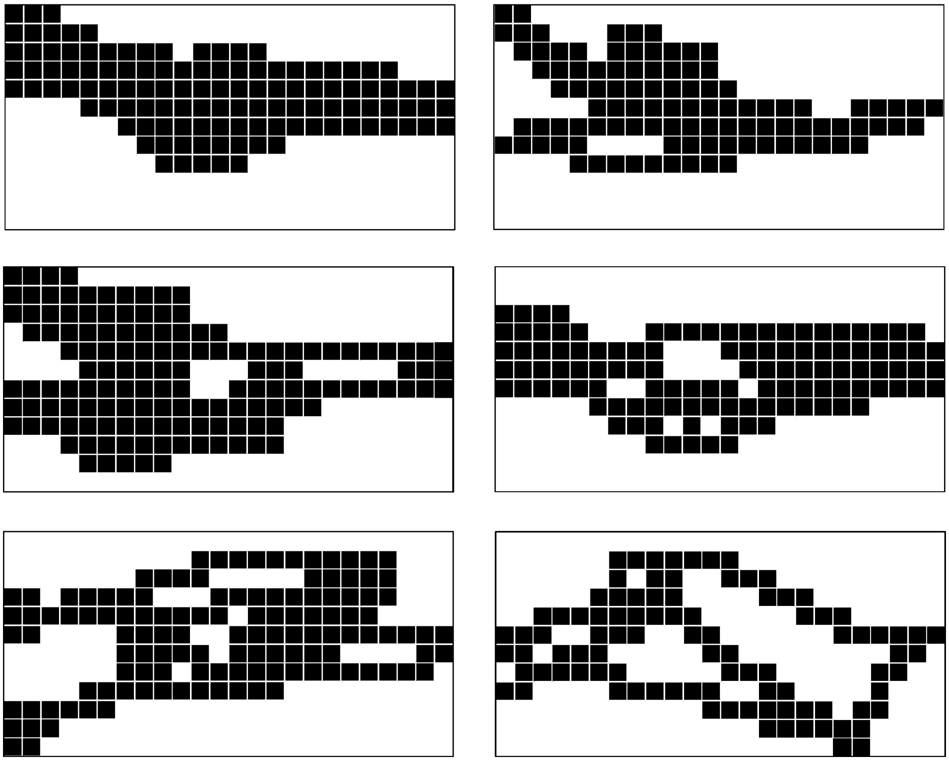


Fig. 8. Sampling of the design topologies of the initial population obtained from the graph representation.

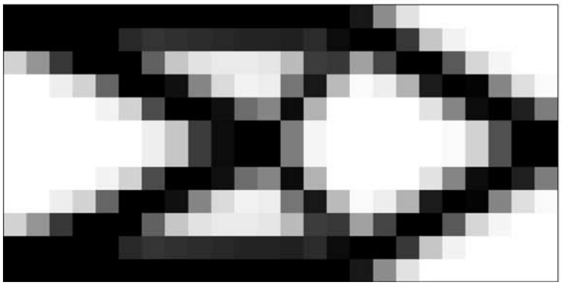


Fig. 9. Best topology from power-law approach.

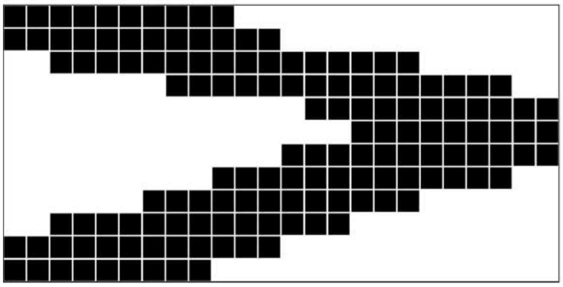


Fig. 10. Best topology using graph with 3 vertices and 2 edges.

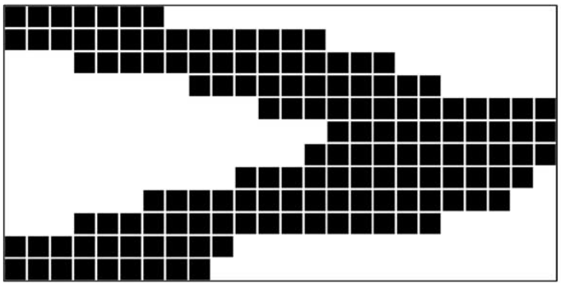


Fig. 11. Best topology using graph with 5 vertices and 6 edges.

the number of function evaluations as would be required by a homogenization-based solution. The CPU time of each FEA function evaluation for a computer with an Intel Pentium 4 processor of 2 GHz is about 0.2 s and therefore the computational cost of the graph representation is significantly higher than that of the power-law approach. Improvement of the GA solution has been observed while using a more complex graph representation (Fig. 7(b)); however, more than three times the number of function evaluations are required. Hence, like all the other numerical methods, a compromise between accuracy and computational cost is usually required when applying the graph representation.

It should be noted that in Sigmund's power-law approach [6], the mass densities of all the elements are initialized with the same value, hence it converges to symmetric topology results for this symmetric problem (Fig. 6). Therefore, this symmetric problem is solved again by using the present graph representation GA and explicitly enforcing symmetry of the resulting geometry, so that a better comparison can be made with the power-law approach. Symmetry is enforced by performing all the GA operations based only on a half of the design domain, but the FEA and other function evaluations are based on the whole structure. With this modification of the problem, a new set of runs is thus performed. Figs. 12 and 13 show the best topologies obtained from the present graph representation GA with 3 vertices and 5 vertices, respectively. Their corresponding minimum compliance values are 75.755 and 75.925 achieved at generation 23 and 75, respectively, and these are corresponding improvements of 4.5% and 2.3% compared to those of the best topologies previously achieved. As expected, using the symmetric geometry condition can improve the optimization results. It can also be seen that although the improvement in the objective function value is small, the topologies have been changed much more significantly. The variation of the best topology using the graph with 5 vertices is shown in Fig. 14, from which it can be seen that the best topology does not change much after 30 generations, though the best initial design is far away from the final solution.

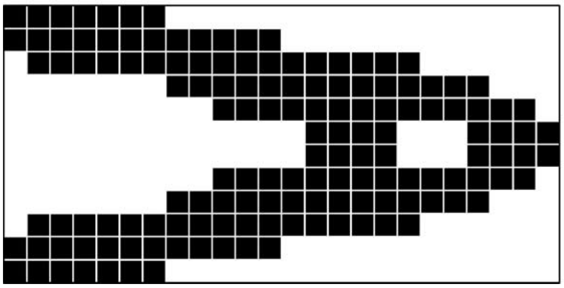


Fig. 12. Best symmetric topology using graph with 3 vertices and 2 edges.

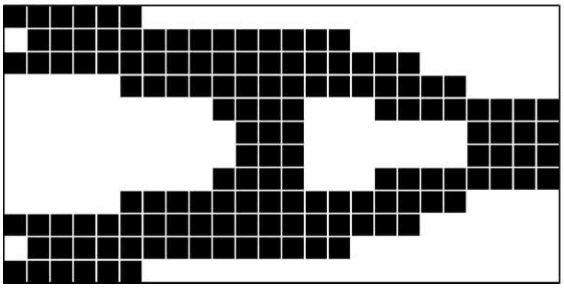
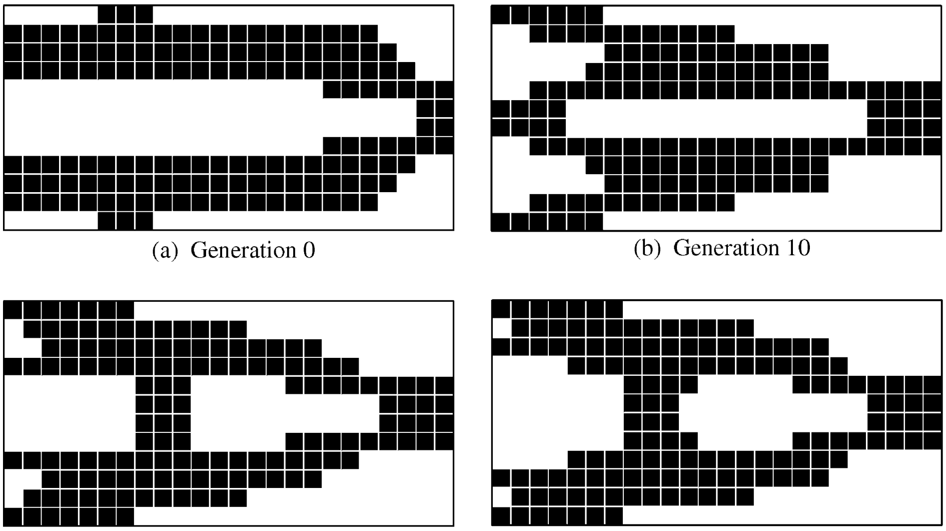


Fig. 13. Best symmetric topology using graph with 5 vertices and 6 edges.



(a) Generation 0

(b) Generation 10

(c) Generation 30

(d) Generation 50

Fig. 14. Evolution history of best symmetric topology (for graph with 5 vertices and 6 edges).

In the preceding examples, the concentrated vertical load $P = 1$ is applied only at a single node. To increase the loading area, a uniformly distributed load is also considered in the present study, which does not change the resultant force (load $P = 1$) but is applied at 7 nodes with distributed loading points along the right vertical boundary, instead of being concentrated at a single node. The connectivity of all individuals in the half domain is shown in Fig. 15, in which 4 vertices and 3 edges are used to construct a simple undirected graph and vertices $a$ and $b$ are allowed only to vary along the left boundary while $c$ and $d$ are inside the loading area along the right boundary only. Figs. 16 and 17 show the best topologies obtained from the power-law approach [6] and the present graph representation GA. Their

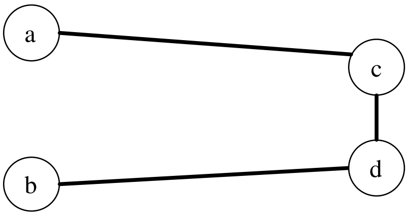


Fig. 15. Graph with 4 vertices and 3 edges (for the distributed load).

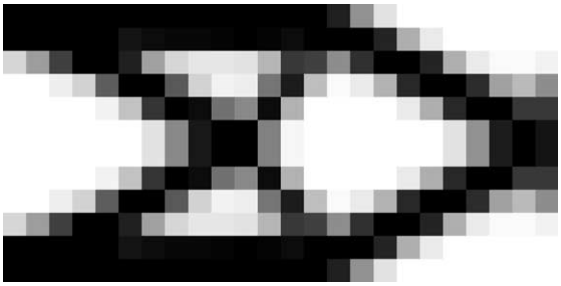


Fig. 16. Best topology for the distributed load (power-law approach).

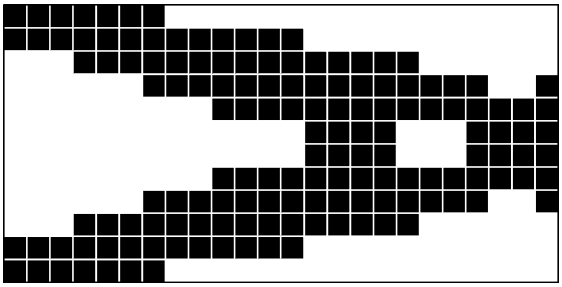


Fig. 17. Best topology for the distributed load (graph representation GA).

Table 1
Effect of mutation rate on the optimal solution (averaged over 20 runs)

| $P_m$ | 0.001 | 0.005 | 0.01 | 0.05 | 0.10 | 0.20 | 0.30 | 0.40 | 0.50 |
|---|---|---|---|---|---|---|---|---|---|
| $C_{min}$ | 87.835 | 83.309 | 84.683 | 81.408 | 80.432 | 79.738 | 78.967 | 79.469 | 79.017 |

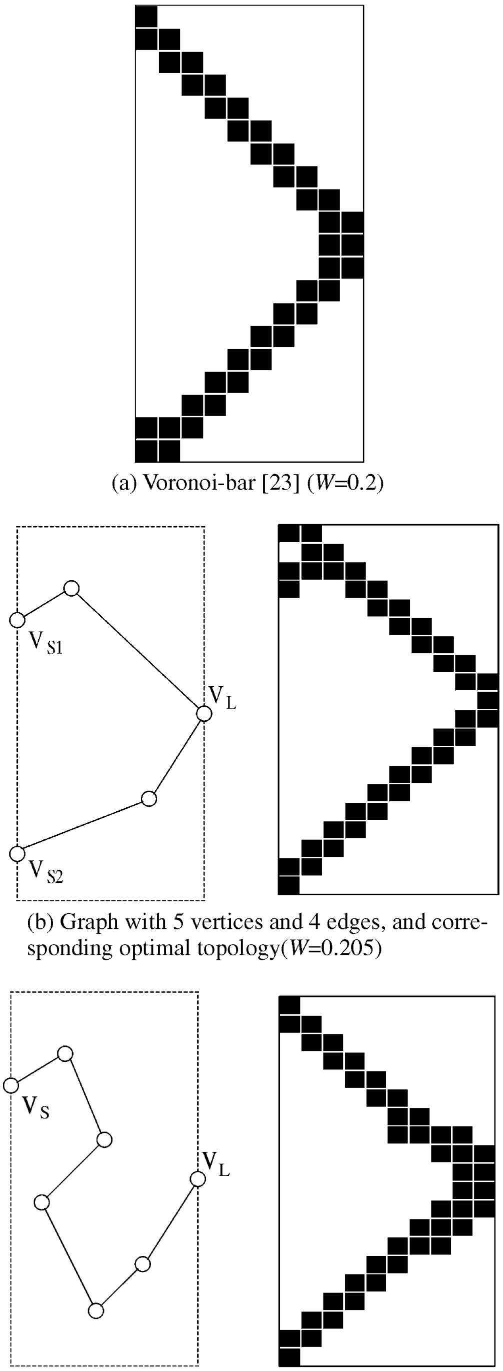
corresponding minimum compliance values are 74.014 and 74.254 (achieved at generation 54), respectively. Again, the present graph representation GA demonstrates better performance in the geometry. Comparing the best topology in Fig. 17 with that in Fig. 12, it can be found that the best topology does not change much with the introduction of the uniformly distributed load since only the local effects will dominate the difference of the response without changing the resultant force. The effect of the mutation rate on the minimum compliance obtained is shown is Table 1, in which $P_m$ is the mutation rate and $C_{\min}$ the minimum compliance. It can be seen that the relatively high mutation rates (0.1–0.5) behaves better than the relatively low mutation rates (0.001–0.005) on the average and the averaged best mutation rate is 0.3. Since the real-value mutation is applied to each real variable only while the bit-value mutation is applied to each bit of a binary string that represent a design variable, it is much more difficult for the real-value representation to make a local move through the mutation operation than for the binary representation. This is also consistent with the observation made by Janikow and Michalewicz [39] that the real-coded GA may take advantage of higher mutation rates, as mentioned previously in Section 2.3.

### 4.2. Example 2: minimum weight design

The minimum weight optimal topology design problems in [23] are adopted in this study to compare the results from the present graph representation with those from the Voronoi-bar representation [23], in which each individual is a list of Voronoi-bars. Two cantilever beams (aspect ratios are 1 × 2 and 2 × 1, respectively) with the left boundary as the fixed support and a unit point force applied vertically downward at half-height of the right boundary are studied. The 1 × 2 and 2 × 1 beams are discretized into 10 × 20 and 20 × 10 meshes, and their weights are minimized subject to the constraint that their maximal displacements are limited to 20 and 220, respectively, as stated in [23]. However, due to the different FE program and FE formulations used in [23], the resulting optimal structures from [23] are analyzed with the FEA code [6] used in this work so as to compute their corresponding maximal displacements. These maximal displacements are then adopted as the displacement constraint limits for the minimum weight problem here, so as to ensure a fair comparison.

#### 4.2.1. The 1 × 2 cantilever beam

In Fig. 18, the best topologies obtained from the present GA using two different prescribed graphs are compared with those from the Voronoi-bar representation GA [23]. It can be seen that the optimal weights achieved by the present GA ($W = 0.205$ and 0.22) are slightly worse than that of the Voronoi-bar, but all the optimal topologies are approximately equivalent to a two-bar V-shape, similar to the theoretical prediction for



(a) Voronoi-bar [23] ($W$=0.2)

(b) Graph with 5 vertices and 4 edges, and corresponding optimal topology($W$=0.205)

(c) Graph with 7 vertices and 6 edges, and corresponding optimal topology($W$=0.22)

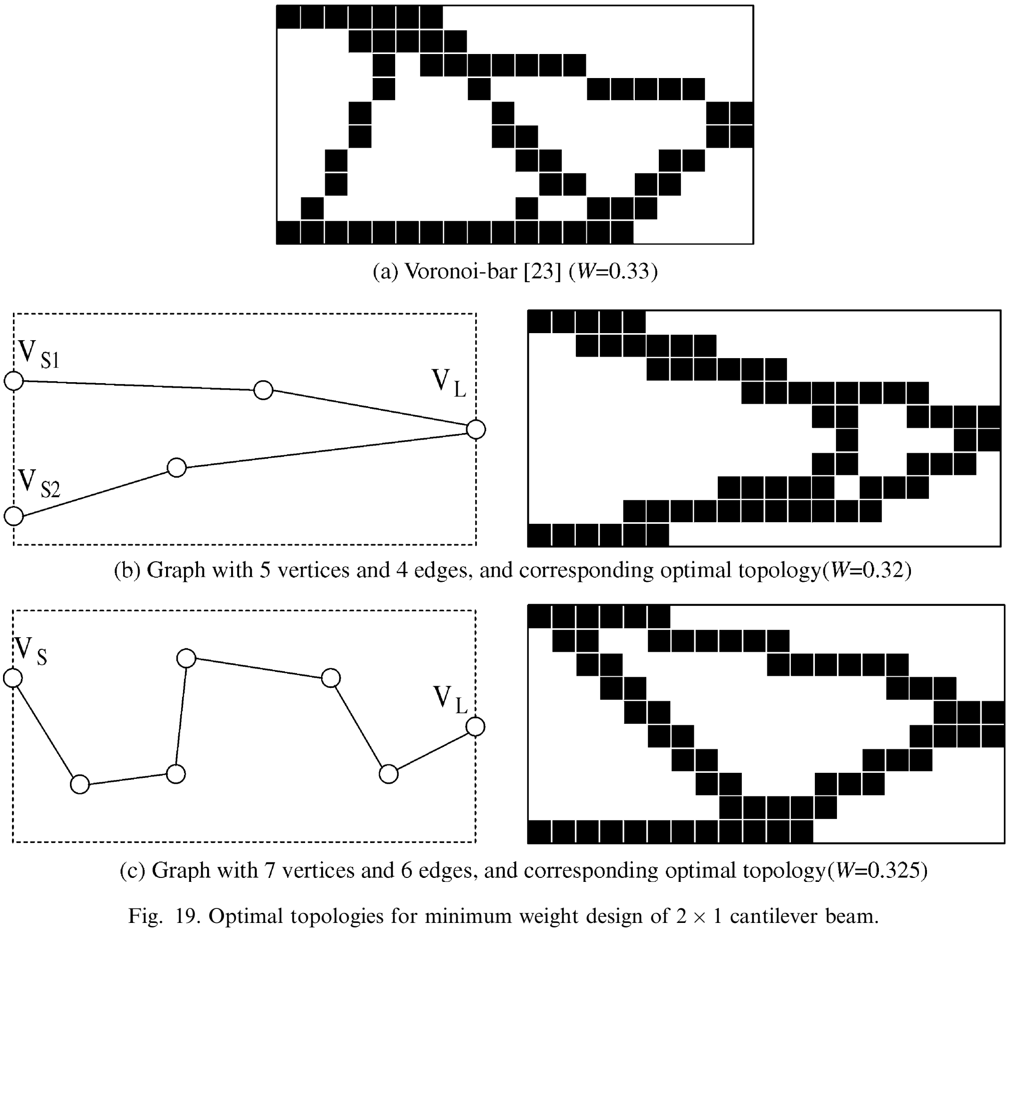
Fig. 18. Optimal topologies for minimum weight design of 1 × 2 cantilever beam.

this problem [23]. Fig. 18(c) shows that the present graph representation can provide optimal topologies almost identical to those from the Voronoi-bar representation even if only one joining path exists in the predefined graph. Note that the locations of all vertices are free to vary anywhere within the design domain except $V_L$ which is the loading point at the fixed location and $V_S$ which is the support point whose location is restricted to vary only along the left support boundary. Hence, good results can be obtained without prior knowledge about the optimal graph to be used or the actual optimal structural topology.

### *4.2.2. The 2 × 1 cantilever beam*

In Fig. 19, the best topologies obtained from the present GA using two different prescribed graphs are compared with those from the Voronoi-bar representation GA [23]. It can be seen that the optimal weights achieved by the present GA ($W = 0.32$ and 0.325) are slightly better than that of the Voronoi-bar. The optimal topologies are quite different, and a similar observation has also been reported in [48] in which the objective function value is almost unchanged from an early step but the topology still changes significantly. In addition, the best topology attained by the Voronoi-bar representation (Fig. 19(a)) is not properly connected as it contains a few one-node hinge connections. Again, Fig. 19(c) shows that the present method with a prescribed graph comprising a single joining path can also find a fairly good solution.

The efficiency of the different methods can be measured by the total number of FE-based function evaluations required in reaching those optimal results since the FE analysis is the most time consuming part of the procedures. In each of the 1 × 2 and 2 × 1 beam examples, the Voronoi-bar representation GA in [23] requires a total of 160,000 FE function evaluations. The reason is that it adopts not only a larger population size (80) and a larger maximum number of generations (2000) but also a small stiffness value for void elements in the design domain so that FE analysis can actually be carried out for the individuals even if they are not properly connected. On the other hand, the present graph representation requires only 4550 function evaluations be-



(a) Voronoi-bar [23] (*W*=0.33)

(b) Graph with 5 vertices and 4 edges, and corresponding optimal topology(*W*=0.32)

(c) Graph with 7 vertices and 6 edges, and corresponding optimal topology(*W*=0.325)

Fig. 19. Optimal topologies for minimum weight design of 2 × 1 cantilever beam.

cause it can guarantee all the individuals are properly connected such that the FE analysis for every individual is possible and no evaluations are wasted on improperly connected individuals. In terms of efficiency, the present graph representation therefore outperforms the Voronoi-bar representation which operates within an unnecessarily larger search space where disconnected structures are allowed.

## 5. Conclusions

A graph representation method for the evolutionary structural optimal topology design is proposed in the present study. Based on graph theory, a valid topology in the design domain is represented by a connected undirected simple graph. The joining paths are defined by segmented cubic Bézier curves with varying thickness. To implement this graph representation, a real-valued chromosome encoding and decoding scheme and a partition crossover method are developed. With the appropriately established GA operations, the graph representation GA is applied to structural topology optimization problems. Compared with the power-law approach, the present graph representation GA has the advantage in generating distinct design geometries and performing a global search in the design domain, but the computational cost will be much higher and a compromise between the accuracy and the computational cost must thus be made carefully. The obtained good results demonstrate an improved performance of the present graph representation over that of the Voronoi-bar representation in terms of the connectivity of the geometry and the convergence speed because of its relatively smaller search space. It is suggested that this graph representation is both physically meaningful and computationally effective in carrying out evolutionary structural topology design.

## Acknowledgements

Part of this work was supported by the Singapore Ministry of Education Academic Research Fund through research grant RG92/98, which the authors gratefully acknowledge.

## References

[1] Akin JE, Arjona-Baez J. Enhancing structural topology optimization. Eng Comput 2001;18(3–4):663–75.

[2] Rozvany GIN. Aims, scope, methods, history and unified terminology of computer-aided topology optimization in structural mechanics. Struct Multidiscip Optim 2001;21(2): 90–108.

[3] Bendsøe MP, Kikuchi N. Generating optimal topologies in structural design using a homogenization method. Comput Meth Appl Mech Eng 1988;71:197–224.

[4] Xie YM, Steven GP. A simple evolutionary procedure for structural optimization. Comput Struct 1993;49(5):885–96.

[5] Bulman S, Sienz J, Hinton E. Comparisons between algorithms for structural topology optimization using a series of benchmark studies. Comput Struct 2001;79(12):1203–18.

[6] Sigmund O. A 99 line topology optimization code written in MATLAB. Struct Multidiscip Optim 2001;21(2):120–7.

[7] Rozvany GIN, Zhou M, Birker T. Generalized shape optimization without homogenization. Struct Optim 1992;4:250–4.

[8] Bendsøe MP. Optimal shape design as a material distribution problem. Struct Optim 1989;1:193–202.

[9] Zhou M, Rozvany GIN. On the validity of ESO type methods in topology optimization. Struct Multidiscip Optim 2001;21(1):80–3.

[10] Kane C, Schoenauer M. Topological optimum design using genetic algorithms. Contr Cybernet 1996;25(5):1059–88.

[11] Holland J. Adaptation in natural and artificial systems. Ann Arbor, MI: University of Michigan Press; 1975.

[12] Goldberg DE. Genetic algorithms in search, optimization and machine learning. Reading, MA: Addison-Wesley Publishing; 1989.

[13] Chapman D, Saitou K, Jakiela MJ. Genetic algorithms as an approach to configuration and topology design. J Mech Des 1994;116(4):1005–12.

[14] Schoenauer M. Shape representation for evolutionary optimization and identification in structural mechanics. In: Winter G, Periaux J, Galan M, Cuesta P, editors. Genetic algorithms in engineering and computer science. Chichester: John Wiley; 1995. p. 443–64.

[15] Schoenauer M. Shape representations and evolutionary schemes. In: Evolutionary Programming: Proceedings of the Fifth Annual Conference on Evolutionary Programming, San Diego, USA, 1996. p. 121–9.

[16] Jakiela MJ, Chapman C, Duda J, Adewuya A, Saitou K. Continuum structural topology design with genetic algorithms. Comput Meth Appl Mech Eng 2000;186(2–4):339–56.

[17] Fanjoy DW, Crossley WA. Topology design of planar cross-sections with a genetic algorithm: Part 1—overcoming the obstacles. Eng Optim 2002;34(1):1–12.

[18] Tai K, Chee TH. 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, UK, 1998. p. 827–36.

[19] Tai K, Chee TH. Design of structures and compliant mechanisms by evolutionary optimization of morphological representations of topology. ASME J Mech Des 2000;122(4):560–6.

[20] Tai K, Cui GY, Ray T. Design synthesis of path generating compliant mechanisms by evolutionary optimization of topology and shape. ASME J Mech Des 2002;124(3):492–500.

[21] Akhtar S, Tai K, Prasad J. Topology optimization of compliant mechanisms using evolutionary algorithm with design geometry encoded as a graph. In: Proceedings of the

ASME International Design Engineering Technical Conferences, Montreal, Canada, 2002 [Paper no. DETC2002/ DAC-34147].

[22] Azid IA, Kwan ASK, Seetharamu KN. A GA-based technique for layout optimization of truss with stress and displacement constraints. Int J Numer Meth Eng 2002;53(7):1641–74.

[23] Hamda H, Jouve F, Lutton E, Schoenauer M, Sebag M. Compact unstructured representations for evolutionary design. Appl Intell 2002;16(2):139–55.

[24] Foley JD, Dan AV, Feiner SK, Hughes JF, Phillips RF. Introduction to computer graphics. Reading, MA: Addison-Wesley; 1994.

[25] Kaveh A. Structural mechanics: graph and matrix methods. 2nd ed. Exeter: Research Studies Press (John Wiley); 1995.

[26] Kaveh A. Optimal structural analysis. Exeter: Research Studies Press (John Wiley); 1997.

[27] Kaveh A. Topological transformation in structural mechanics. In: Topping BHV, editor. Civil and structural engineering computing 2002. Edinburgh: Saxe-Coburg Publications; 2002 [chapter 13].

[28] Sen S, Narasimhan S, Deb K. Sensor network design of linear processes using genetic algorithms. Comput Chem Eng 1998;22(3):385–90.

[29] Igel C, Stagge P. Effects of phenotypic redundancy in structure optimization. IEEE Trans Evol Comput 2002;6(1):74–85.

[30] Kaveh A, Kalatjari V. Topology optimization of trusses using genetic algorithm, force method and graph theory. Int J Numer Meth Eng 2003;58(5):771–91.

[31] West DB. Introduction to graph theory. 2nd ed. Upper Saddle River: Prentice-Hall; 2001.

[32] Goldberg DE, Korb B, Deb K. Messy genetic algorithms: motivation, analysis, and first results. Complex Syst 1989;3:349–530.

[33] Shai O, Preiss K. Isomorphic representations and well-formedness of engineering systems. Eng Comput 1999; 15(4):303–14.

[34] Kawamura H, Ohmori H, Kito N. Truss topology optimization by a modified genetic algorithm. Struct Multidiscip Optim 2002;23(6):467–72.

[35] Bhardwaj MK, Day DM. Modifications to graph partitioning tools for use with FETI methods. In: Herrera I, Keyes DE, Widlund OB, Yates R, editors. Fourteenth International Conference on Domain Decomposition Methods. Mexico: Cocoyoc; 2002. p. 195–202.

[36] Nagata Y, Kobayashi S. Edge assembly crossover: a high-power genetic algorithm for the traveling salesman problem. In: Back T, editor. Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97). Morgan Kaufmann Publishers; 1997. p. 450–7.

[37] Globus A, Lawton J, Wipke T. Graph crossover. In: Spector L, Goodman ED, Wu A, editors. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001), San Francisco, CA, USA, 2001. p. 761.

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

[39] Janikow CZ, Michalewicz Z. An experimental comparison of binary and floating point representations in genetic algorithms. In: Proceedings of the Fourth International Conference on Genetic Algorithms, San Diego, CA, 1991. p. 31–6.

[40] Mühlenbein H, Schlierkamp-Voosen D. The science of breeding and its application to the breeder genetic algorithm (BGA). Evol Comput 1994;1(4):335–60.

[41] Deb K. An efficient constraint handling method for genetic algorithms. Comput Meth Appl Mech Eng 2000;186(2–4):311–38.

[42] Coello CAC, Montes EM. Handling constraints in genetic algorithms using dominance-based tournaments. In: Proceedings of the Fifth International Conference on Adaptive Computing Design and Manufacture (ACDM 2002), vol. 5. Springer-Verlag; 2002. p. 273–84.

[43] Lagaros ND, Papadrakakis M, Kokossalakis G. Structural optimization using evolutionary algorithms. Comput Struct 2002;80(7–8):571–89.

[44] Baker JE. Adaptive selection methods for genetic algorithms. In: Grefenstette JJ, editor. Proceedings of the International Conference on Genetic Algorithms and their Applications. Pittsburgh, PA: Lawrence-Erlbaum Associates; 1985. p. 101–11.

[45] Baker JE. Reducing bias and inefficiency in the selection algorithm. In: Grefenstette JJ, editor. Proceedings of the Second International Conference on genetic algorithms. Cambridge, MA: Lawrence-Erlbaum Associates; 1987. p. 14–21.

[46] Bruns TE, Tortorelli DA. Topology optimization of non-linear elastic structures and compliant mechanisms. Comput Meth Appl Mech Eng 2001;190(26–27):3443–59.

[47] Fogel LJ. Intelligence through simulated evolution (forty years of evolutionary programming). Intelligent systems. John Wiley; 1999.

[48] Belytschko T, Xiao SP, Parimi C. Topology optimization with implicit functions and regularization. Int J Numer Meth Eng 2003;57(8):1177–96.