Available online at www.sciencedirect.com

SCIENCE DIRECT®

Finite Elements in Analysis and Design 42 (2006) 547–566

FINITE ELEMENTS IN ANALYSIS AND DESIGN

www.elsevier.com/locate/finel

Performance enhancement of evolutionary search for structural topology optimisation

Sujin Bureerat*, Jumlong Limtragool

Department of Mechanical Engineering, Khon Kaen University, 40002, Thailand

Received 17 November 2004; received in revised form 4 October 2005; accepted 30 October 2005
Available online 28 December 2005

Abstract

The use of evolutionary algorithms for topological design of structures has been investigated for many years. The methods have disadvantages in that they have slow convergence rate and a complete lack of consistency. In this paper, a number of well-established evolutionary methods including genetic algorithm, stud-genetic algorithm, population-based incremental learning and simulated annealing are reviewed in terms of their philosophical bases. The effective means to deal with topological design problems and prevent checkerboards on a topology are briefly detailed. A new set of design variables employing a numerical technique named approximate density distribution is proposed. The new technique and the classical 1–0 binary variables are applied to the various evolutionary methods and they are implemented on a number of structural topology optimisation problems. The results obtained from the various design strategies are compared, illustrated and discussed. Numerical experiment shows that using the present technique can improve both convergence rate and consistency of the evolutionary algorithms.
© 2005 Elsevier B.V. All rights reserved.

Keywords: Evolutionary algorithms; Topology optimisation; Interpolation; Compliance minimisation

1. Introduction

Structural optimisation is a design problem posed to find structural physical parameters that give the optimum value of design objective whilst fulfilling design constraints. The design parameters or design variables can be mainly categorised as topological, shape and sizing variables. Based upon the types of the design variables the optimum design problems can thus be topology, shape and sizing optimisation respectively. Traditionally, topological design is a structural design problem to be performed in the early stage of the design process. The problem is to find the best material distribution on a given design domain so that design merits are minimised or maximised, while meeting predefined constraints. Practically, such a design process can be carried out by using finite element analysis and optimisation solvers. It is pre-processed by discretising a given design domain into a number of finite elements. The densities or thickness of the elements determine a structural topology in such a way that the elements with near-zero thickness represent holes in the structure while the other elements have material existence as shown in Fig. 1. The topology or material distribution obtained at the optimum point is then taken as the initial configuration for the designed structure.

The classical topological design problem is global stiffness maximisation which is equivalent to compliance minimisation [1]. Another classical design problem is eigenfrequency or dynamic stiffness maximisation [2]. One of the most preferable and popular optimisers for a problem of this kind is the optimality criteria method (OCM) [3] which is arguably the most powerful optimisation technique for the task. The other gradient-based methods such as the method of moving asymptotes (MMA) and sequential linear programming (SLP) are also employed to this design problem successfully [1]. Some research work on the use of evolutionary or population-based optimisation methods like genetic algorithms for topological design has been conducted e.g. in [4–7]. The obtained outcomes indicate that evolutionary algorithms, although said to be successful, are inferior to gradient-based methods in terms of both convergence rate and consistency. This is mainly due to a great

* Corresponding author.
E-mail address: sujbur@kku.ac.th (S. Bureerat).

0168-874X/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.finel.2005.10.011

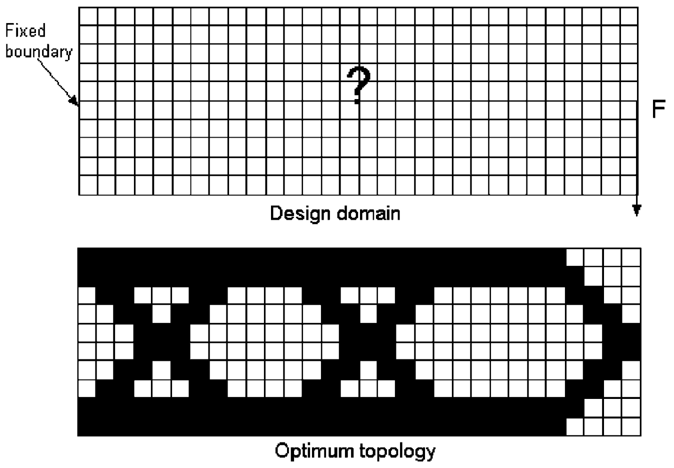

Fig. 1. Sample of topology optimisation.

many topology design variables (approximately as many as the number of structural elements). The gradient-based methods may not be affected by the large number of design variables but the evolutionary methods do suffer because their search mechanisms rely heavily on randomisation. We should not, however, ignore the use of evolutionary methods for topology design since some of their advantages remain attractive to designers. For example, the methods are universal and simple to use, are capable of global optimisation and are derivative-free optimisation methods. Moreover, when applying 1–0 binary bits for representing structural topologies in evolutionary search, the problems of intermediate density or thickness on the topologies can be easily avoided.

In the past, most articles relating to topology optimisation using evolutionary algorithms tended to present only their convergence rate whereas their convergence consistency seems to have been overlooked. Basically, the search consistency of evolutionary algorithms can be measured by performing the method several times and comparing the obtained optimum results. Numerical experiments towards evolutionary algorithms' search consistency for topology design are illustrated in [8,9]. The results show that all the presented evolutionary algorithms have poor convergence consistency, which is clearly the effect of the large number of design variables and random search. The work in this paper is aimed at developing a new technique to enhance the convergence consistency of evolutionary optimisation methods. The selected evolutionary methods reviewed include simulated annealing (SA), genetic algorithm (GA), stud-genetic algorithm (Stud-GA) and population-based incremental learning (PBIL). An approximate density distribution (ADD) technique, which is the applications of surface spline interpolation for approximating the density on structural elements, is presented. The optimisation methods are implemented on four structural compliance minimisation problems with the use of ADD and classical 1–0 binary design variables. The results obtained from the various evolutionary methods are compared, illustrated and discussed. It is shown that the numerical strategy presented can improve the performance of the evolutionary methods in terms of both convergence rate and consistency.

2. Topology optimisation

The standard form of constrained optimisation problem can be written as

$$\min_{\mathbf{x}} f(\mathbf{x}) \tag{1}$$

subject to

$$g_i(\mathbf{x}) \leqslant 0,$$
$$h_i(\mathbf{x}) = 0,$$
$$\mathbf{x} \in \Omega,$$

where $\mathbf{x}$ is the vector of design variables, g_i are inequality constraints, h_i are equality constraints and $\mathbf{x} \in \Omega$ are box constraints.

As previously stated the topological design problem has a large number of design variables and since it is normally implemented at the conceptual design stage, some complicated structural design constraints may be removed. The optimisation problem is simplified to a simple constrained problem which, for a typical compliance minimisation, can be written as

$$\min_{\boldsymbol{\rho}} = c(\boldsymbol{\rho}) = \mathbf{U}^{\mathrm{T}}\mathbf{K}\mathbf{U} \tag{2}$$

subject to

$$V(\boldsymbol{\rho}) - rV_0 \leqslant 0,$$
$$\mathbf{0} < \boldsymbol{\rho}^{\mathrm{l}} \leqslant \boldsymbol{\rho} \leqslant \boldsymbol{\rho}^{\mathrm{u}},$$

where $\boldsymbol{\rho}$ is the elements' density, c is the structural compliance, $\mathbf{U}$ is the vector of structural displacement, $\mathbf{K}$ is a structural stiffness matrix, V is structural volume, $V_0 = V(\boldsymbol{\rho}^{\mathrm{u}})$, $r \in (0, 1)$ is the volume reduction ratio and $\boldsymbol{\rho}^{\mathrm{l}}$ and $\boldsymbol{\rho}^{\mathrm{u}}$ are the lower and upper bounds.

Note that the lower bounds are usually set to be slightly higher than zero so as to prevent singularity in the structural global stiffness matrix. The problem (2) is set for the gradient-based optimisers rather than the evolutionary algorithms. For compliance minimisation using evolutionary algorithms, it is better to alter the problem to be an optimisation problem where the objective is the weighted sum of compliance and volume [8]. The bounds can be dealt with in such a way that each design variable has the value of either '0' or '1' where a '0' design variable is decoded to be the lower bound and a '1' design variable is decoded to be the upper bound of the corresponding variable. This idea is advantageous in that the resulting topology is free from having intermediate densities. Another problem often encountered is having checkerboard patterns on the resulting optimum topologies, which is found to be the cause of numerical instability [1]. A simple way to deal with such a problem is the introduction of a checkerboard penalty function to the optimisation problem [10]. The modified objective function, therefore, is of the form

$$f(\boldsymbol{\rho}) = w_1 c(\boldsymbol{\rho}) + w_2 V(\boldsymbol{\rho}) + w_3 P(\boldsymbol{\rho}), \tag{3}$$

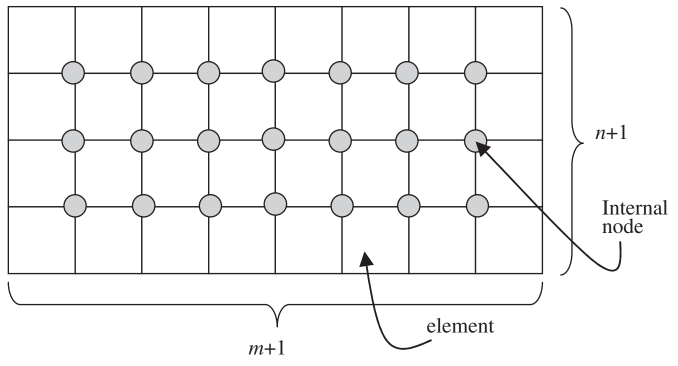

Fig. 2. Internal nodes.

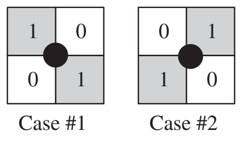

Fig. 3. Patterns to be penalised.

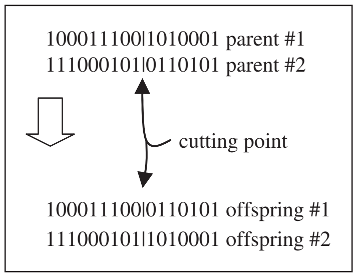
100011100|1010001 parent #1
111000101|0110101 parent #2
cutting point
100011100|0110101 offspring #1
111000101|1010001 offspring #2

Fig. 4. Illustration of GA crossover.

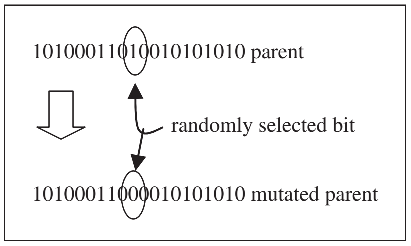
10100011010010101010 parent
randomly selected bit
10100011000010101010 mutated parent

Fig. 5. Illustration of GA mutation.

where P is a checkerboard penalty value computed from

$$P = \sum_{i=1}^{m \times n} p_i. \tag{4}$$

Fig. 2 shows a rectangular design domain being meshed to have $(m + 1) \times (n + 1)$ elements and, thus, $m \times n$ internal nodes. At an ith internal node, with four surrounding elements, if the element densities (or thickness for plate structures) match any pattern shown in Fig. 3, $p_i = 1$, otherwise $p_i = 0$. It has been proven that this simple technique can effectively prevent the checkerboard problem in topological design [10].

The value of w_3 can be defined as high as desired since a checkerboard pattern is unwanted. The values of w_1 and w_2 are set based upon the principle that a higher w_1 value leads to a topology being rather stiff and heavy whereas a higher value of w_2 results in the topology having more or bigger voids and less global stiffness.

3. Evolutionary algorithms

Evolutionary algorithms are sometimes called population-based or random-directed optimisation methods. The optimisation method categorised as an evolutionary method starts its search with a group of initial design solutions which is called a population. The population is then evolved in some manner, mostly relying on randomisation and a selection mechanism, until the optimum solution is reached [11]. The evolutionary methods employed in this paper include GA, Stud-GA, PBIL and SA. The methods will be briefly detailed as follows:

3.1. Genetic algorithm

GA is probably the best known and most popular evolutionary method [12]. The method, like its name, can be best thought of as imitating Darwinian natural selection in that offspring or children are created by mating pairs of their parents. The pairs of parents are chosen at random where their probabilities of being selected are based upon their merit. Some of the population's elite, both parents and children, are then carried on to the next generation (or iteration). This process is repeated iteratively until the optimum point is achieved. The conventional GA operators used to create offspring are crossover and mutation, which are allowed to occur by predefined probabilities. The simple crossover and mutation operators are illustrated in Figs. 4 and 5, respectively. There have been numerous research articles relating to the development of new GA techniques for a variety of engineering and scientific applications, which mostly claim to be better than the original. The simplest form of GA (with crossover, mutation and selection) is, however, still considered as one of the most widely used GAs.

3.2. Stud-genetic algorithm

Stud-GA is a slight modification of GA search which is claimed to be as powerful as the original version [13]. Rather than generating an initial population the search starts with an initial solution or 'stud'. The stud is then mutated to produce a current population. An individual solution in the population is also allowed to be changed by shuffling its bit positions with a given probability. Afterwards, the best individual is selected as the new stud. The procedure carries on until the optimum is achieved.

Probability Vector #1 0.5 0.5 0.5 0.5	Probability Vector #2 0.5 0.5 0.5 0.5	Probability Vector #3 0.75 0.5 0.5 0.25
1 0 0 1	1 1 0 0	1 1 0 0
0 1 0 0	1 0 0 1	1 0 1 0
1 1 1 0	0 1 1 0	1 1 0 0
0 0 1 1	0 0 1 1	0 0 1 1
Corresponding population	Corresponding population	Corresponding population

Fig. 6. Probability vectors and their corresponding populations in PBIL.

3.3. Population-based incremental learning

PBIL is an evolutionary method that has a completely different basis from any evolutionary algorithm using binary codes. Rather than memorising a set of design solutions and passing them on to the next generation, the PBIL uses a so-called probability vector representing a population. The probability vector is the vector having the same size as the number of bits of an individual binary solution. The *i*th element of the probability vector represents the probability of having '1' on the *i*th bit of the set of solutions. Fig. 6 depicts the sample of probability vectors and their corresponding populations. Each row of the population denotes a binary design solution. It is shown that one probability vector can lead to various populations.

Starting with an initial probability vector whose elements are full of '0.5', a corresponding population is then created and function evaluation is performed. The probability vector is then updated based upon the learning rate and the strings of the best individual. The vector can also be mutated with a predefined probability to prevent premature convergence. The vector is updated iteratively until reaching the optimum. For more details, see [14].

3.4. Simulated annealing

SA can sometimes be classified as an evolutionary algorithm. The method can be seen as mimicking the random behaviour of molecules during an annealing process, which involves slow cooling from a high temperature. As the temperature cools, the atoms line themselves up and form a crystal, which is the state of minimum energy in the system. As it is a universal optimisation method, 1–0 binary coding can be applied. The search procedure of SA is somewhat similar to the stud-GA procedure in that it starts with an initial solution, which will be called the parent. The parent is then mutated in some manner leading to a set of children or offspring. The best offspring is said to be a candidate to challenge its parent. For minimisation, if the candidate has a lower objective value than that of the parent, the parent is replaced by the candidate. In cases that the candidate has a higher objective function value than its parent, it still has a chance to replace the parent if accepted by Boltzmann probability, and this makes SA different from the stud-GA. Since on each loop the worse candidate may replace its parent, the best solution and the parent may not be the same solution. Therefore, the best individual on each loop should be kept along with a parent ensuring that the best solution of the search is not lost. One of the key factors of SA search is the way to create children on each iteration. For more details of SA, see [9,11,15,16].

4. Approximate density distribution

The numerical technique termed Approximate Density Distribution (ADD) aims to reduce the number of topological design variables without changing the number of finite elements used. ADD exploits an application of surface spline interpolation for approximating the finite element densities from the given densities at some sampling points [17]. Fig. 7 shows what are called sampling points ('+' sign) and the centres of the rectangular elements ('o' sign).

From the rectangular design domain being meshed into *n* elements as shown, let $\mathbf{r}_j^0$ be the position vectors of *m* sampling points and $\mathbf{r}_k^e$ be the position vectors of the centre points of the *n* elements. The vector of densities at the centre points of the elements are denoted by $\boldsymbol{\rho}^e$ and the vector of densities at the sampling points are denoted by $\boldsymbol{\rho}^0$. Density value at a particular

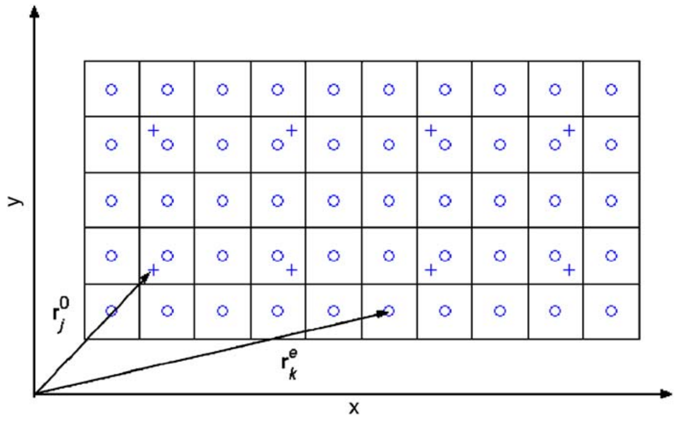

Fig. 7. Position vectors of sampling points and element centre points.

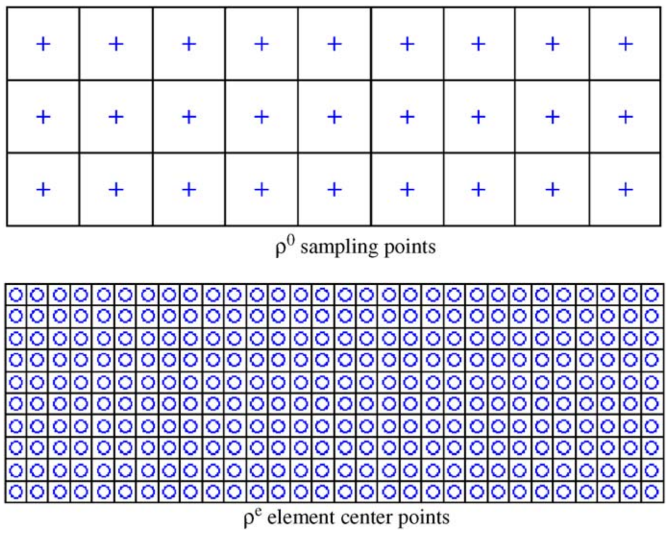

Fig. 8. Sampling points and element centre points.

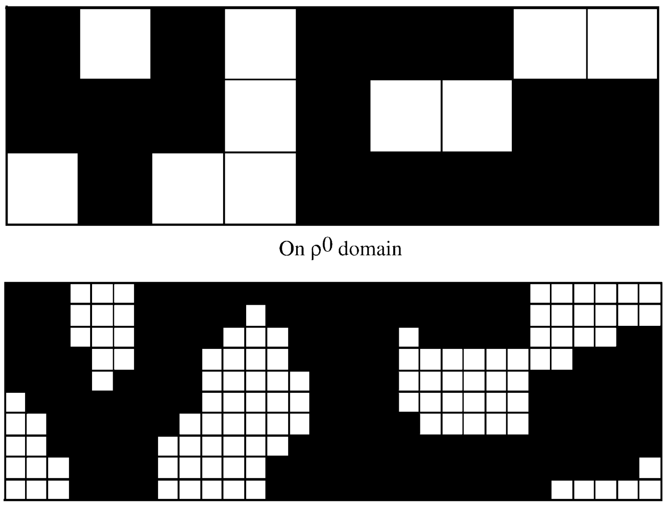

Fig. 9. Mapping of $\boldsymbol{\rho}^0$ and $\boldsymbol{\rho}^e$: Example 1.

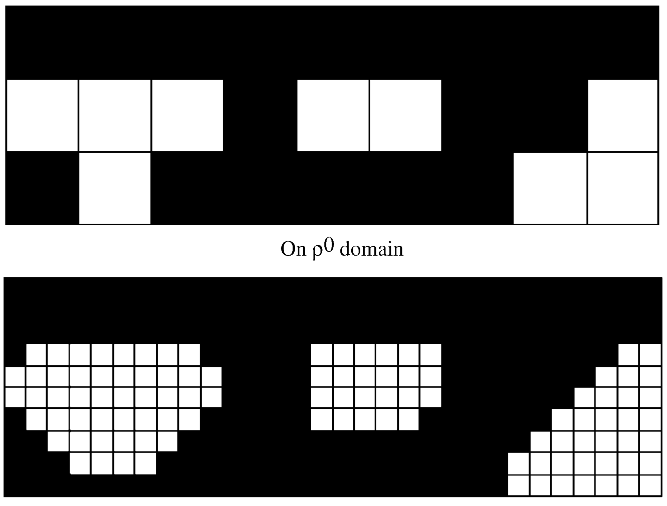

Fig. 10. Mapping of $\boldsymbol{\rho}^0$ and $\boldsymbol{\rho}^e$: Example 2.

point $\mathbf{r}$ on the design domain can be estimated by using the relation

$$\rho(\mathbf{r}) = \sum_{j=1}^{m} b_j f(d(\mathbf{r}, \mathbf{r}_j^0)), \tag{5}$$

where b_j is the coefficient vector to be determined, $f(d)$ is called a radial basis function and $d(\mathbf{r}, \mathbf{r}_j^0)=\sqrt{(\mathbf{r}-\mathbf{r}_j^0)^{\mathrm{T}}(\mathbf{r}-\mathbf{r}_j^0)}$ is the distance between $\mathbf{r}$ and $\mathbf{r}_j^0$.

By using (5), densities at sampling points $\boldsymbol{\rho}^0$ can be computed as

$$\boldsymbol{\rho}^0 = \mathbf{A}\mathbf{b}, \tag{6}$$

where $\mathbf{A} = [a_{ij}]_{m\times m} = [f(d(\mathbf{r}_i^0, \mathbf{r}_j^0))]$ and $\mathbf{b} = [b_i]_{m\times 1}$.

Thus, the densities at the element centre points are

$$\boldsymbol{\rho}^e = \mathbf{C}\mathbf{b}, \tag{7}$$

where $\mathbf{C} = [c_{kj}]_{n\times m} = [f(d(\mathbf{r}_k^e, \mathbf{r}_j^0))]$.

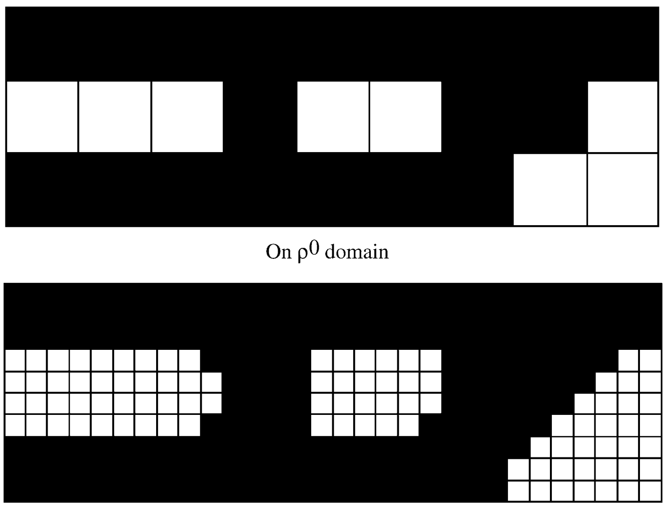

Fig. 11. Mapping of $\boldsymbol{\rho}^0$ and $\boldsymbol{\rho}^e$: Example 3.

Solving (6) and (7) leads to the relation

$$\boldsymbol{\rho}^e = \mathbf{C}\mathbf{A}^{-1}\boldsymbol{\rho}^0 = \mathbf{T}\boldsymbol{\rho}^0. \tag{8}$$

The radial basis function used in this work is

$$f(d_{ij}) = 1 + d_{ij} + d_{ij}^2 + d_{ij}^3. \tag{9}$$

Eq. (8) is the transformation between $\boldsymbol{\rho}^0$ and $\boldsymbol{\rho}^e$. When used with evolutionary algorithms, it is possible that the vector $\boldsymbol{\rho}^e$ calculated from (5) would still have intermediate density values, which are undesirable. Thus, to prevent this, the computed $\boldsymbol{\rho}^e$ will be filtered by

$$\rho_k^e = \begin{cases} \rho_k^u; & \rho_k^e > 0.5, \\ \rho_k^l & \text{otherwise}. \end{cases} \tag{10}$$

By using the transformations (8) and (10), the new design variables $\boldsymbol{\rho}^0$ can be treated to have a smaller size than the number of element densities. Fig. 8 displays a plot of 9×3 sampling points ('+' sign) and 30×10 centre points of elements ('o' sign). Examples of mapping between $\boldsymbol{\rho}^0$ and $\boldsymbol{\rho}^e$ are illustrated in Figs. 9–12. In Fig. 9, it is shown that the density distribution on the $\boldsymbol{\rho}^e$ domain can form a connected structure even if density distribution on the $\boldsymbol{\rho}^0$ domain is disconnected. However, there is no guarantee that the disconnected structures on the $\boldsymbol{\rho}^0$ domain will always result in one connected structure on the $\boldsymbol{\rho}^e$ domain. Figs. 10 and 11 demonstrate that different configurations on the $\boldsymbol{\rho}^0$ domain can lead to similar structural configurations on the $\boldsymbol{\rho}^e$ domain and this is an important feature in improving the convergence consistency in randomising-based methods. Fig. 12 shows a particular density distribution on the 9×5 sampling points and the transformed distribution on the 30×10 centre points. It is shown that the ADD technique helps

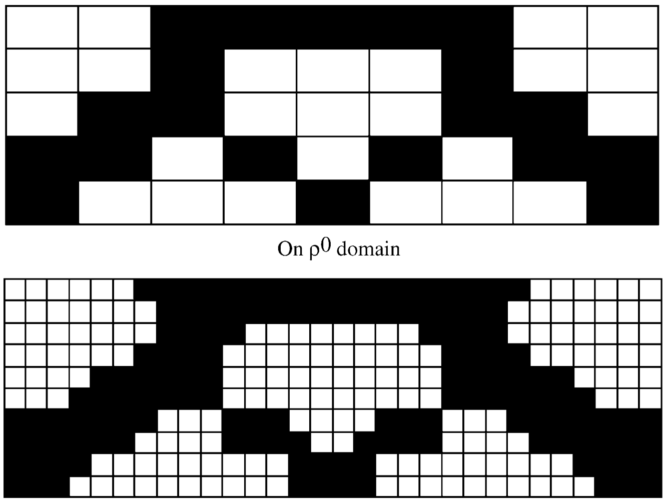

Fig. 12. Mapping of $\boldsymbol{\rho}^0$ and $\boldsymbol{\rho}^\mathrm{e}$: Example 4.

suppressing checkerboard patterns but cannot completely prevent them. Nevertheless, the checkerboard penalty function can be used to cope with this problem.

5. Test problems

Numerical experiments are conducted so that the effectiveness of using design variables with ADD technique and the classical 1–0 binary design variables can be observed and compared. Note that for simplicity the original 1–0 binary design variables will be called DSV1 whereas the design variables used by the ADD technique will be called DSV2. The test design problems are compliance minimisation of a cantilever plate, Michell-type structure, 2D bridge and square plate with a hole. Fig. 13 shows a cantilever plate made up of material having Young's modulus $E = 200 \times 10^9$ N/m^2 and Poisson's ratio $v = 0.3$. The structure is loaded at the right-hand top corner. The rectangular plate, discretised to 20×10 elements, has an aspect ratio of $L/H = 2$. Fig. 14 depicts the positions of $\mathbf{r}_j^0$ and $\mathbf{r}_k^\mathrm{e}$ (represented by '+' and 'o' signs, respectively), which are used to calculate the transformation between $\boldsymbol{\rho}^0$ and $\boldsymbol{\rho}^\mathrm{e}$ for this design

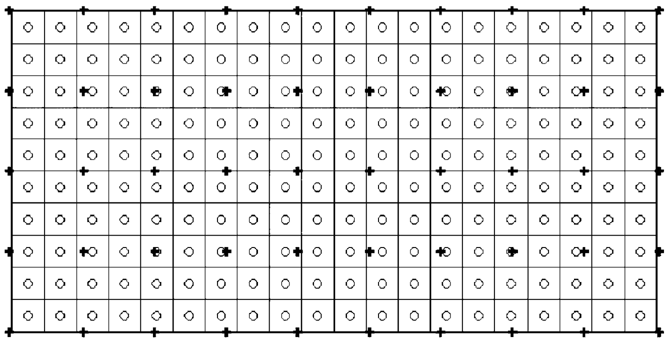

Fig. 14. Sampling points and element centre points of the cantilever plate.

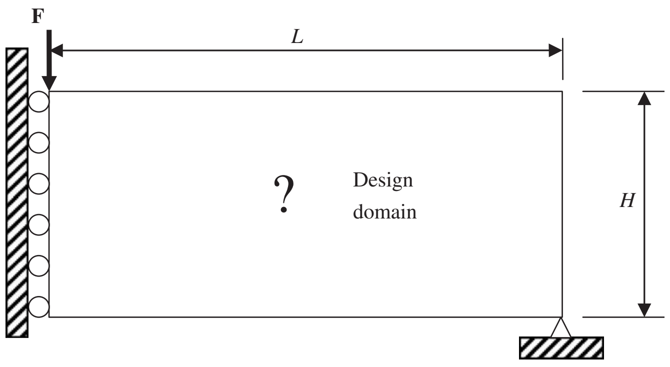

Fig. 15. Michell-type plate.

problem. DSV1 has 20×10 elements and DSV2 has 10×5 elements. The design problem of the plate is termed CASE1.

A half model of the Michell-type structure which is made of the same material as the cantilever plate is illustrated in Fig. 15. The structure has an aspect ratio of $L/H = 3$. It is discretised to be 30×10 elements and loaded by the external force at the left-hand top corner. The positions of $\mathbf{r}_j^0$ and $\mathbf{r}_k^\mathrm{e}$ are displayed in Fig. 16. From the figure, DSV1 has 30×10 elements whereas DSV2 has 15×7 elements. The design problem of this plate is called CASE2.

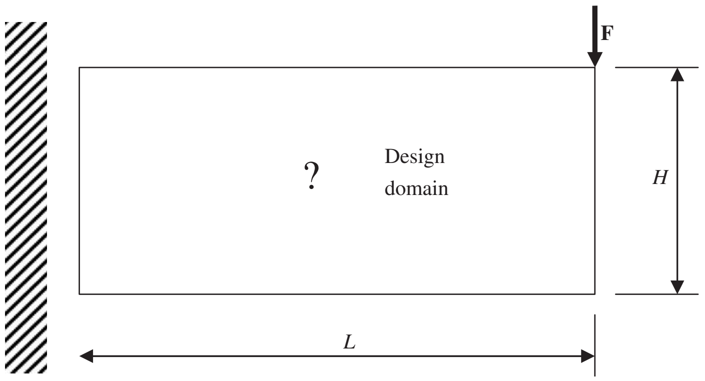

Fig. 13. Cantilever plate.

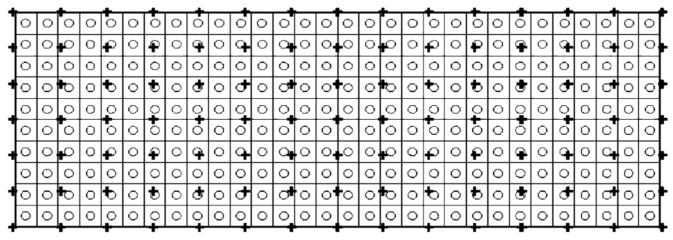

Fig. 16. Sampling points and element centre points of the Michell-type plate.

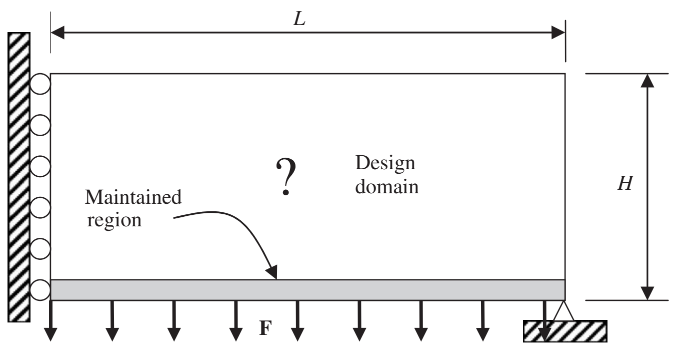

Fig. 17. 2D bridge.

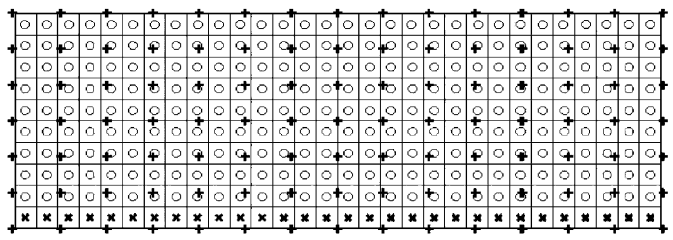

Fig. 18. Sampling points, element centre points and maintained elements of the 2D bridge.

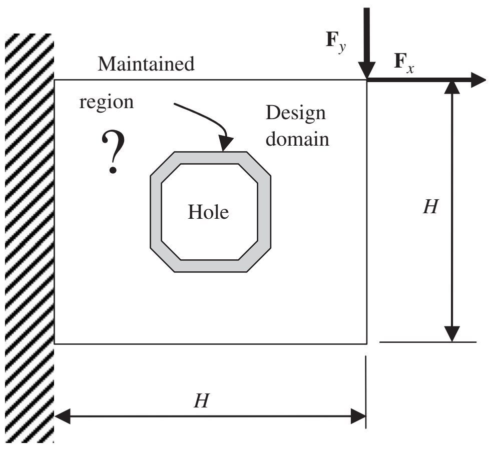

Fig. 19. Plate with the hole.

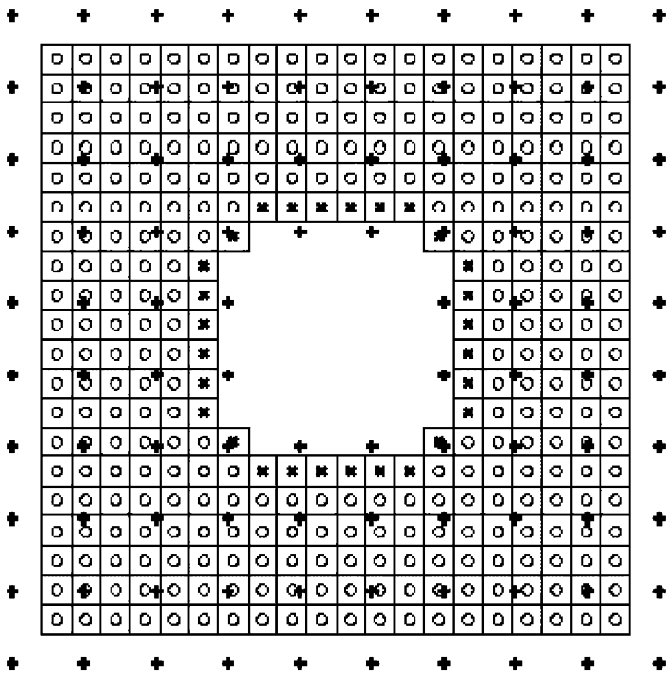

Fig. 20. Sampling points, element centre points and maintained elements of the plate with the hole.

CASE3 is the optimisation of a 2D bridge under distributed load as shown in Fig. 17. The bridge, with an aspect ratio $L/H = 3$, is made of the same material as used in CASE1 and CASE2. Fig. 18 displays the finite element model of the bridge as well as 270 element centre points that are used as DSV1, 105 sampling points for DSV2 and 30 element centre points to be unchanged.

Design CASE4 is the topological design of a square plate with a hole as shown in Fig. 19. The structure is made up of the same material as the previous structures. Fig. 20 shows the finite element model and all the points used in design process. There are 340 finite elements with the thickness of 28 elements being maintained and the thickness of 312 elements being set as DSV1. There are 96 sampling points assigned for DSV2 as shown. It is shown in this case that the sampling points can be placed outside the design domain.

Linear finite element analysis is applied to these four structures. Because it is the case of 2-dimentional structures, element thickness is used instead of element density. As a 4-node quadrilateral membrane element is used, a checkerboard problem is expected to occur. Therefore, the checker board penalty (4) is added to an objective function as in (3) where the weighting factors are $[w_1, w_2, w_3] = [0.5, 8, 4]$. Note that the weighting factors are intuitively selected for benchmarking the effectiveness of the proposed numerical technique. It is expected to have some voids on an optimum topology. The functions $c(\boldsymbol{\rho})$ and $V(\boldsymbol{\rho})$ are normalised by the values of $c(\mathbf{1})$ and $V(\mathbf{1})$, respectively. The evolutionary methods employed here are as follows:

- *GA01*: Genetic algorithm with 0.95 crossover probability, 0.05 mutation probability, 40 design solutions for a population and 50 search iterations.

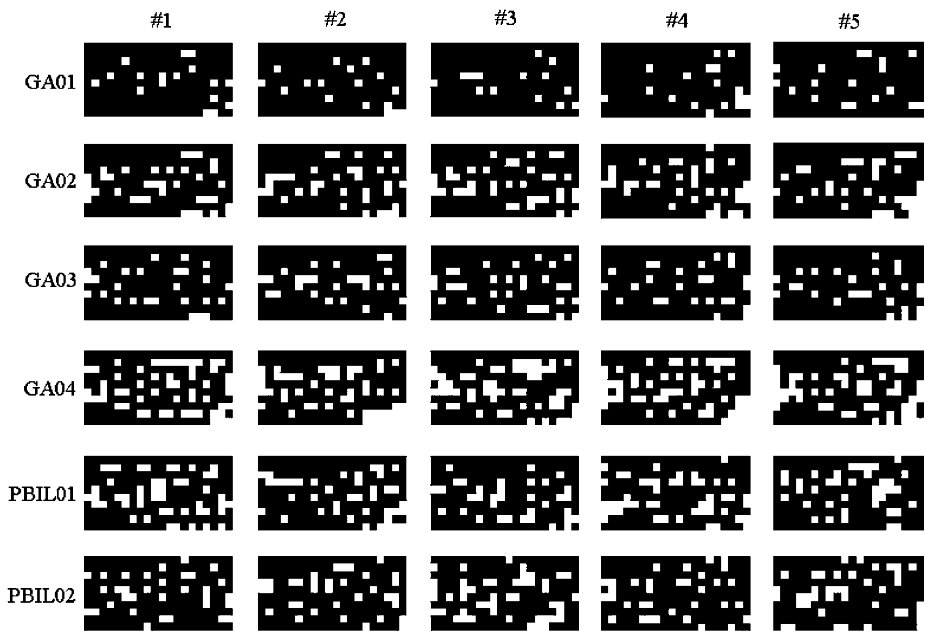

Fig. 21. CASE1 design results from GAs and PBILs with DSV1.

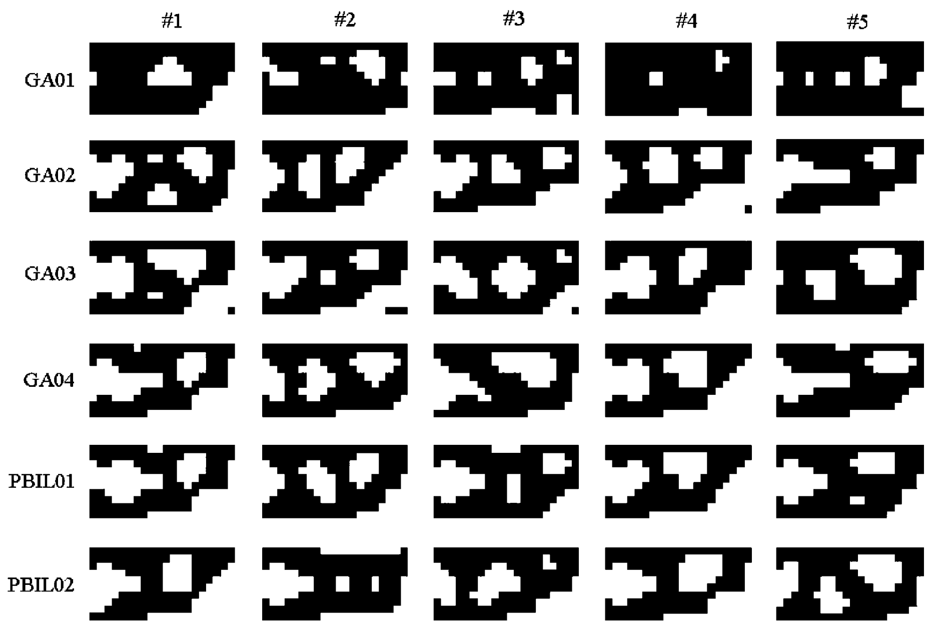

Fig. 22. CASE1 design results from GAs and PBILs with DSV2.

- *GA02*: Genetic algorithm with 0.05 crossover probability, 0.95 mutation probability, 40 design solutions for a population and 50 search iterations.
- *GA03*: Genetic algorithm with 0.95 crossover probability, 0.05 mutation probability, 20 design solutions for a population and 100 search iterations.
- *GA04*: Genetic algorithm with 0.05 crossover probability, 0.95 mutation probability, 20 design solutions for a population and 100 search iterations.
- *PBIL01*: Population-based incremental learning with 0.05 mutation probability, 40 design solutions for a population and 50 search iterations.
- *PBIL02*: Population-based incremental learning with 0.05 mutation probability, 20 design solutions for a population and 100 search iterations.
- *Stud-GA01*: Stud-genetic algorithm with 0.05 shuffling probability, 40 design solutions for a population and 50 search iterations.

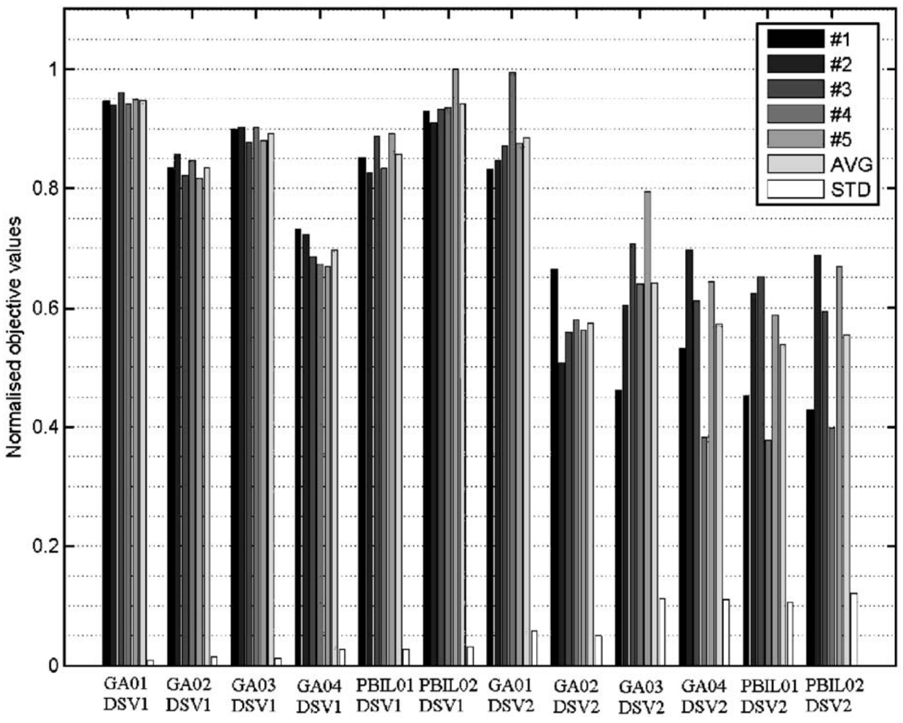

Fig. 23. CASE1 normalised objective values from GAs and PBILs.

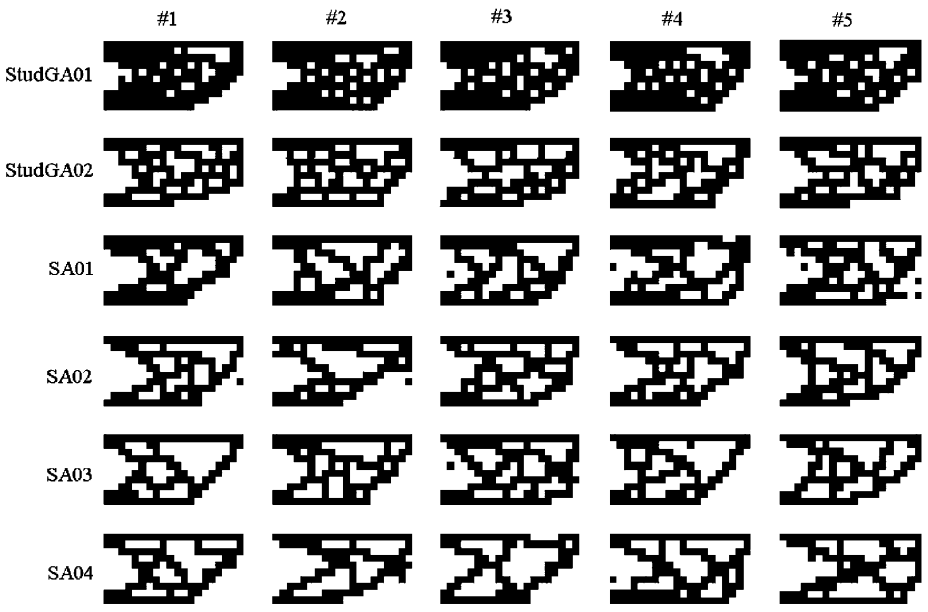

Fig. 24. CASE1 design results from stud-GAs and SAs with DSV1.

- *Stud-GA02*: Stud-genetic algorithm with 0.05 shuffling probability, 20 design solutions for a population and 100 search iterations.
- *SA01*: Simulated annealing with mutation1, 40 candidates being created on one loop and 50 search iterations.
- *SA02*: SA01 with 20 candidates being created and 100 search iterations.
- *SA03*: Simulated annealing with mutation2, 20 candidates being created and 100 search iterations.
- *SA04*: Simulated annealing with mutation3, 20 candidates created on each iteration and 100 search iterations.

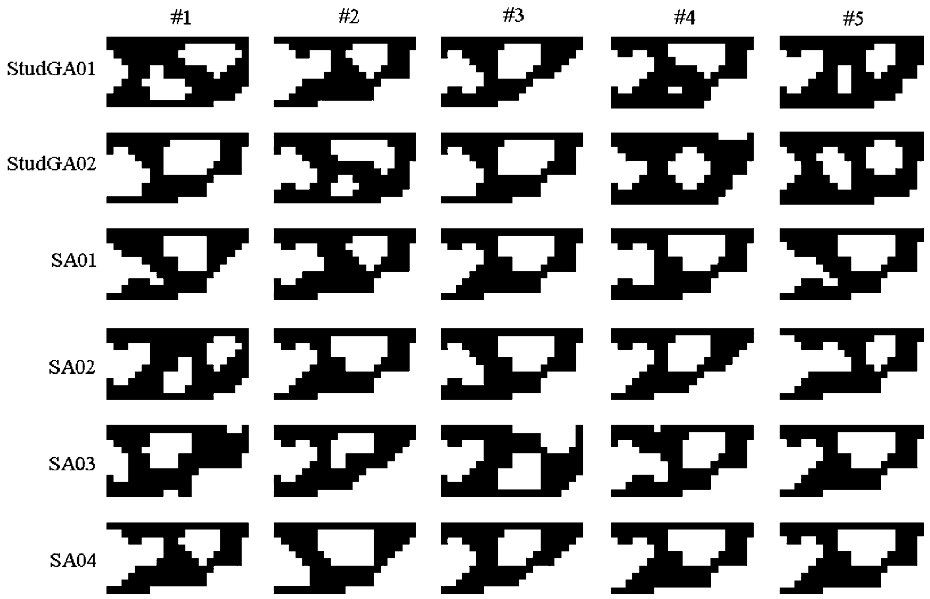

Fig. 25. CASE1 design results from stud-GAs and SAs with DSV2.

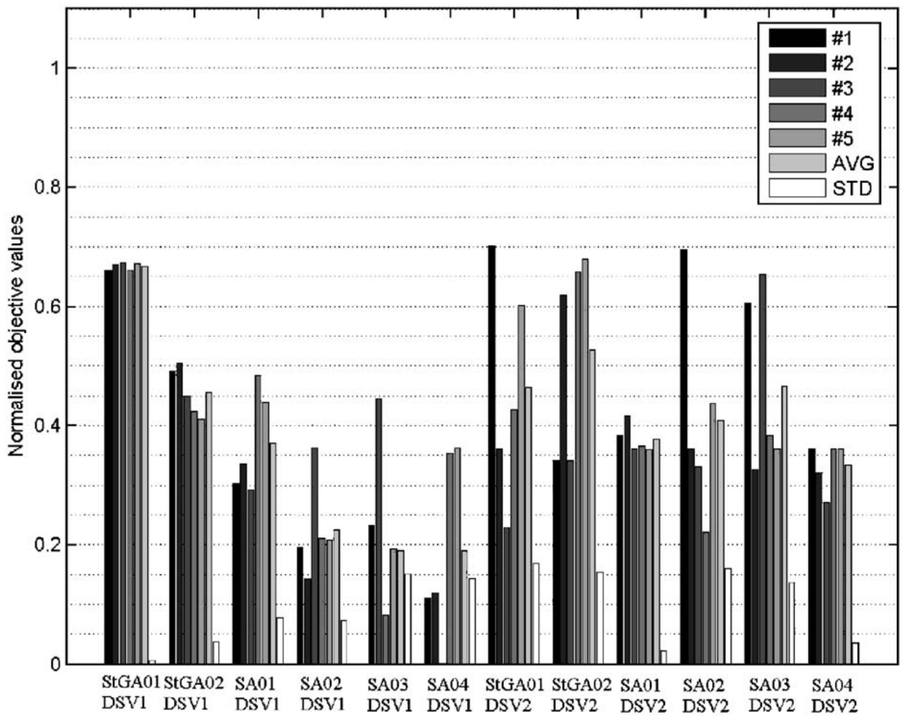

Fig. 26. CASE1 normalised objective values from stud-GAs and SAs.

Three mutation strategies of SAs can be detailed as: mutation1—all of the children are generated from mutating a parent, mutation2—most of the children are created by mutating their parent and a few children are randomly generated and mutation3—half of the children are created by mutating their parent while the other half are from mutating on the current best solution. The initial temperature for SAs is 10 and the final temperature is 0.001.

With the various evolutionary search strategies, the effectiveness of crossover and mutation for structural topology optimisation can be compared, the effect of population size and iteration number on the optimum results can be investigated and a

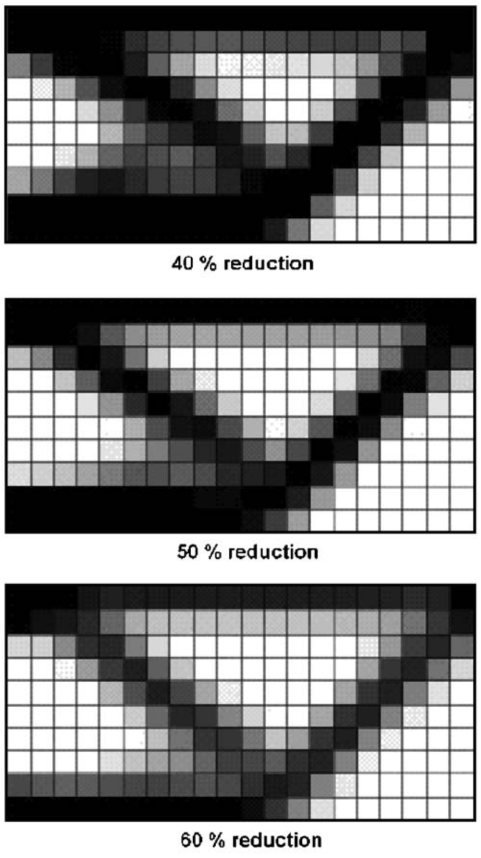

Fig. 27. Optimum topologies of CASE1 form using OCM.

comparison of the SA mutation strategies can be made. Each method needs 2000 function evaluations for one operation. The population size and number of loops are set for measuring the performance of the methods; therefore, it is not guaranteed that all the evolutionary methods will reach the real optimum with those predefined parameters. For each design case, all of the methods start with the same set of initial solutions, and they are implemented to solve the design problem using both DSV1 and DSV2 for five attempts. The best solution of the last generation obtained from a particular method on each run is taken as the optimum result. The convergence rate is measured by the mean value of five optimum function values. The search consistency can be measured by the standard deviation of the five objective values but a more reliable factor is the similarity of the five topologies.

6. Numerical results

The optimum results of CASE1 obtained from using GAs and PBILs with DSV1 are displayed in Fig. 21. Fig. 22 shows the results of CASE1 from using GAs and PBILs with DSV2. Most of the topologies shown in Fig. 22 are better than the topologies shown in Fig. 21. Fig. 23 is a bar chart illustration of the optimum objective values of the topologies in Figs. 21 and 22. Each bar group consists of five optimum objective values, the average and the standard deviation of the five objective values. Note that all the objective values are normalised to the range of [0, 1] to ease in illustration and comparison. The optimum topologies of CASE1 obtained from using stud-GAs and SAs with DSV1 and DSV2 are illustrated in Figs. 24 and 25, respectively. The bar chart of the objective values of the topologies in Figs. 24 and 25 is displayed in Fig. 26. With the exception of stud-GA01, the results from using stud-GA02 and all SAs with DSV1 are better than those obtained from using their counterparts with DSV2. In terms of convergence rate, SA04 with DSV1 is the best while the second best is SA03 with DSV1. The worst method is GA01 with DSV1. When considering the similarity of the five plate configurations obtained from each method, SA04 and SA01 with DSV2 give the best consistency. Fig. 27 shows the optimum topologies from solving the design problem (2) using the optimality criteria method (OCM) where the mass reduction ratio, *r*, is 40%, 50% and 60%. It is shown that the topologies from using OCM are similar to some of the topologies from using DSV2.

The optimum results of CASE2 design problem from using GAs and PBILs with DSV1 and DSV2 are displayed in Figs. 28 and 29, respectively. The bar chart of the normalised objective values of the resulting topologies is illustrated in Fig. 30. Fig. 31 shows the optimum topologies obtained from stud-GAs and SAs with DSV1 whereas the optimum topologies from using the methods with DSV2 are depicted in Fig. 32. The bar chart of their corresponding normalised objective values is shown in Fig. 33. For this design case, the results from all 12 methods with DSV2 are improved compared to their counterparts that use DSV1. SA04 with DSV2 gives the best convergence rate while the second best is SA02 with DSV2. The worst is PBIL02 with DSV1. According to Figs. 28, 29, 31 and 32, the most consistent methods are SA03 and SA04. The optimum topologies of the beam from solving the constrained problem (2) using OCM with reduction ratios as 40%, 50% and 60% are displayed in Fig. 34, which are somewhat similar to some of those shown in Fig. 32.

The optimum topologies of the 2D bridge CASE3 from using GAs and PBILs with DSV1 and DSV2 are illustrated in Figs. 35 and 36. The bar chart illustration of the corresponding normalized objective function values is shown in Fig. 37. The optimum topologies of the bridge obtained from employing stud-GAs and SAs with the use of DSV1 and DSV2 are depicted in Figs. 38. and 39. The bar chart of their normalised objective values is shown in Fig. 40. Similarly to CASE2, the results from using all of the optimisation methods with DSV2 are better than those from using their counterparts with DSV1. The method that gives the best convergence rate is SA04 with DSV2 while the second best is SA02 with DSV2. The method that gives the worst convergence rate is GA01 with DSV1. The most consistent method is SA04 with DSV2. Fig. 41 displays the optimum topologies of the bridge obtained from solving the design problem (2) using OCM. The topologies are comparable to most of those shown in Fig. 39.

The topologies shown in Figs. 42 and 43 are from optimising CASE4 using GAs and PBILs with DSV1 and DSV2 while the bar chart of the objective values is shown in Fig. 44. Figs. 45 and 46 show the optimum topologies of CASE4 obtained from

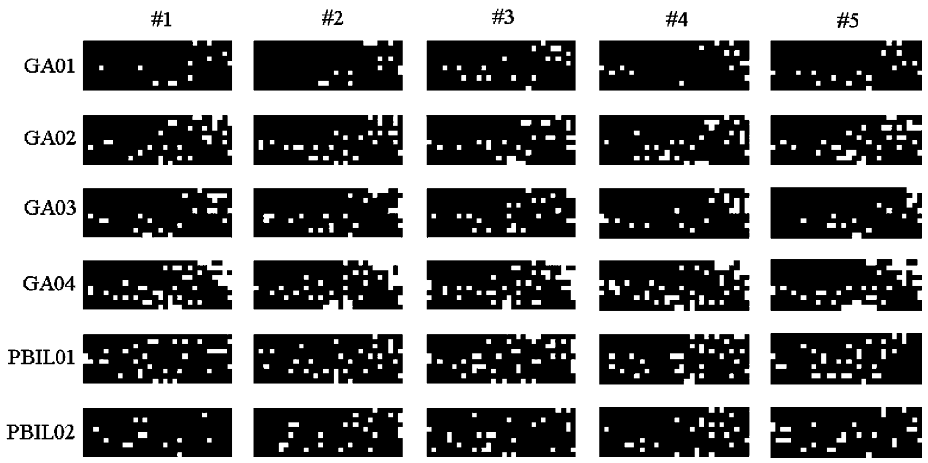

Fig. 28. CASE2 design results from GAs and PBILs with DSV1.

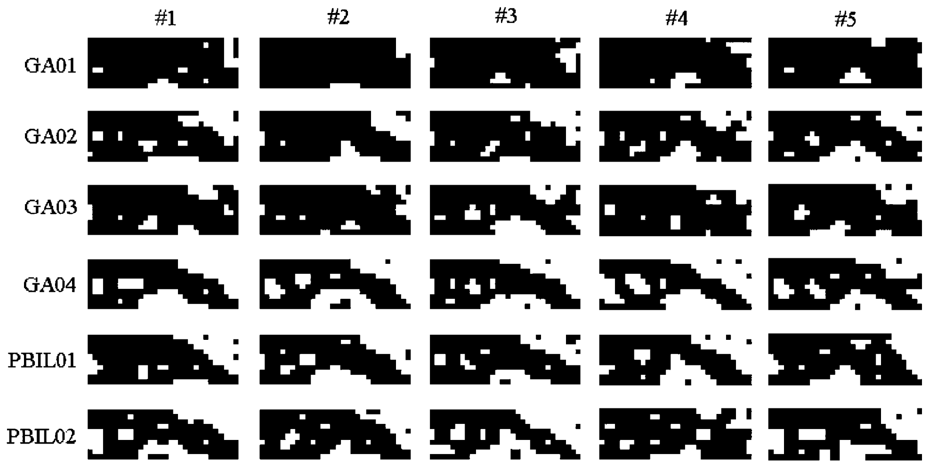

Fig. 29. CASE2 design results from GAs and PBILs with DSV2.

using stud-GAs and SAs with DSV1 and DSV2. In Fig. 47, the bar chart of the objective values of the topologies is displayed. Similarly to CASE2 and CASE3, all the optimisation methods using DSV2 have better convergence rate than their counterparts using DSV1. The best method considering convergence rate is SA03 with DSV2 while the second best is stud-GA01 with DSV2. The most consistent method, according to the similarity of the configurations using five attempts, is SA02 with DSV2. Fig. 48 displays the optimum topologies of the structure in CASE4 from using OCM. It can be said that some of the topologies in Fig. 43 and most of the topologies in Fig. 46 are comparable to those from using the OCM.

When considering all design cases and conditions, GA with crossover as the main operator is inferior to one that uses mutation as the main operator. The smaller population size and larger iteration number lead to better design results when applied to GAs. The same can be said of SA01 versus SA02 except for CASE1 with the use of DSV2. For stud-GAs using DSV1, the smaller population size and greater iteration number result in better design topologies. In contrast, for PBIL search, the resulting topologies are worsened when using smaller population size and larger iteration number with exception of CASE3 using DSV2.

7. Conclusions, discussion and future work

For both DSV1 and DSV2 implementation, SAs are superior to the other evolutionary methods in terms of convergence rate whereas the second best is stud-GAs. The results obtained from using SAs with DSV2 are said to be comparable to those obtained from using the OCM. Among the SA02, SA03 and SA04, their performances are equally good, although SA04 is slightly better in most cases. It can be concluded that the mutation operator is superior to the crossover operator in the topological design of plates as all of the optimisation methods that use mutation as the main evolutionary operator can search for better topologies than those

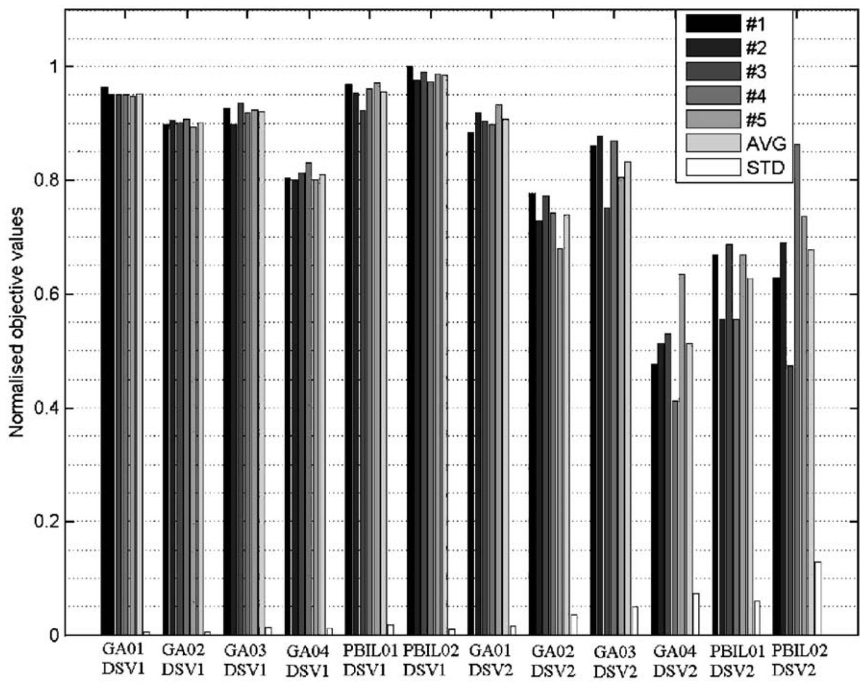

Fig. 30. CASE2 normalised objective values from GAs and PBILs.

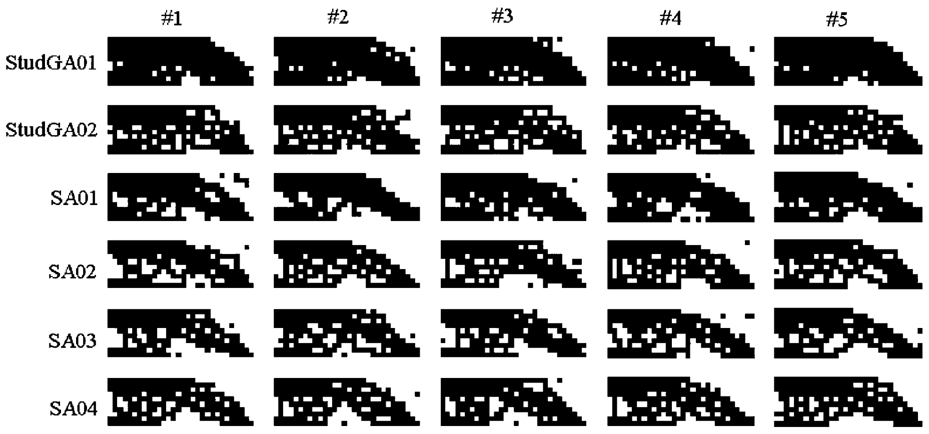

Fig. 31. CASE2 design results from stud-GAs and SAs with DSV1.

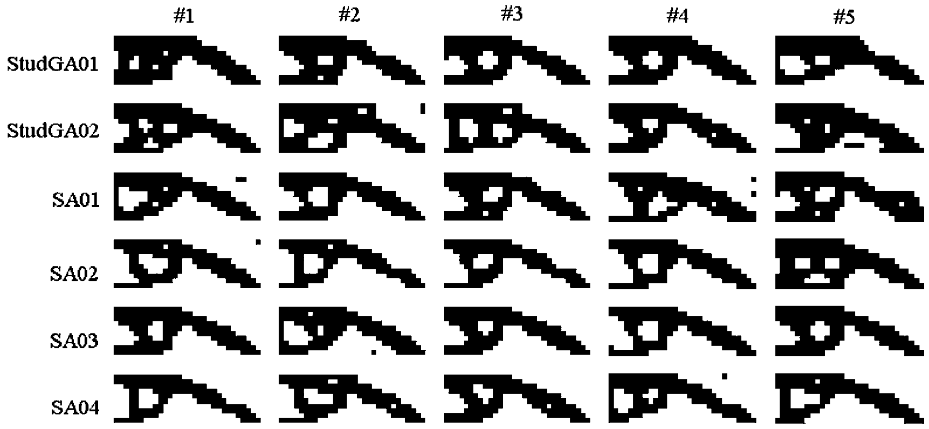

Fig. 32. CASE2 design results from stud-GAs and SAs with DSV2.

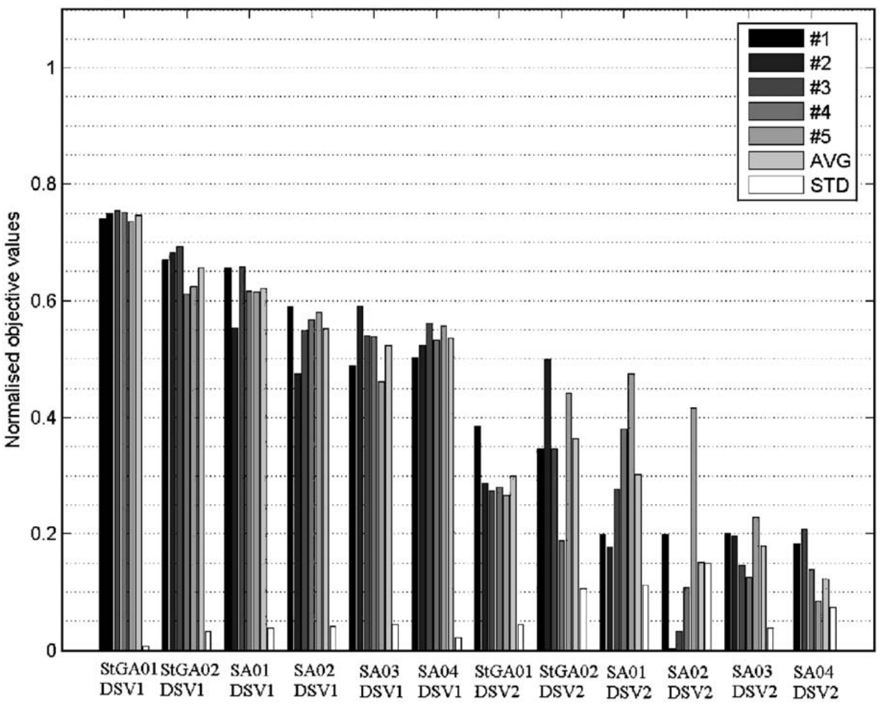

Fig. 33. CASE2 normalised objective values from stud-GAs and SAs.

obtained from crossover-dominated GAs. Each time that mutation takes place, it can be best thought of as either digging a plate or filling a hole, and this makes the mutation more suitable for topological design. The other form of evolutionary method, PBIL, has a fairly good convergence rate. The reduction of population size can weaken its search performance. The search performance of GAs and SAs can be enhanced if a smaller population size and greater iteration number are used. For the main investigation in this work, using DSV2 or design variables with the application of the ADD technique can enhance both convergence rate and consistency of all of the evolutionary methods with the exception of the convergence rate of SAs in CASE1. With 200 variables of DSV1, which is not a really large scale problem, SAs can reach the optimum using DSV1 with 2000 function evaluations. The optimum objective value of higher resolution DSV1 is normally better than that of the lower resolution DSV2, and this causes SAs with DSV1 to have better convergence rate than SAs with DSV2 in the design CASE1. The advantages of using a mutation-based evolutionary algorithm with the ADD technique are that it can be applied to all kinds of topology design problem, evaluation of function derivative is not required, and the optimum topology without intermediate can be achieved using the method. The disadvantage is that it still has slow convergence when compared to some classical gradient-based approach.

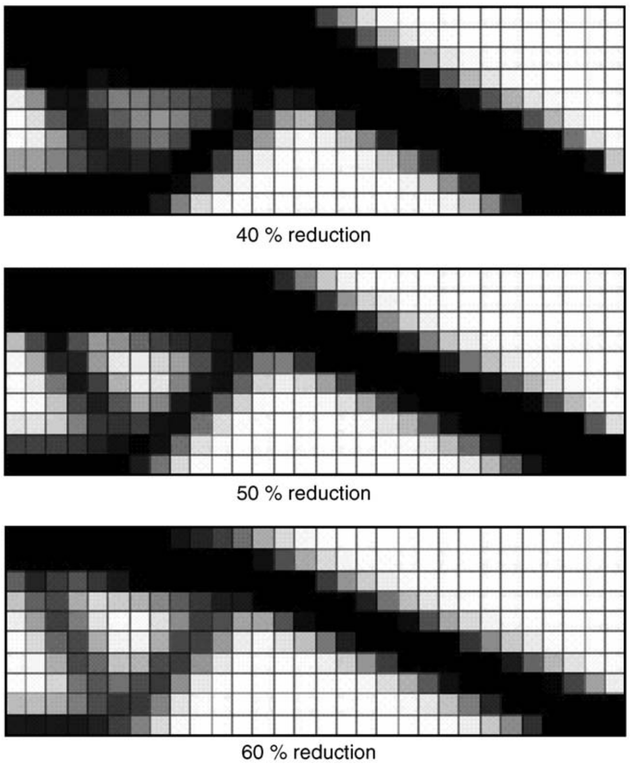

Fig. 34. Optimum topologies of CASE2 using OCM.

Some further improvement can be made. As seen from some of the resulting topologies, there are some floating bits that cannot be prevented by the checkerboard penalty. This problem occurs due to using a random-based mutation operator. A new mutation operator could be developed in such a way that floating bits on a mutated solution can be avoided. Moreover, the weighting factors used in (5) for a particular design problem will have to be selected based upon

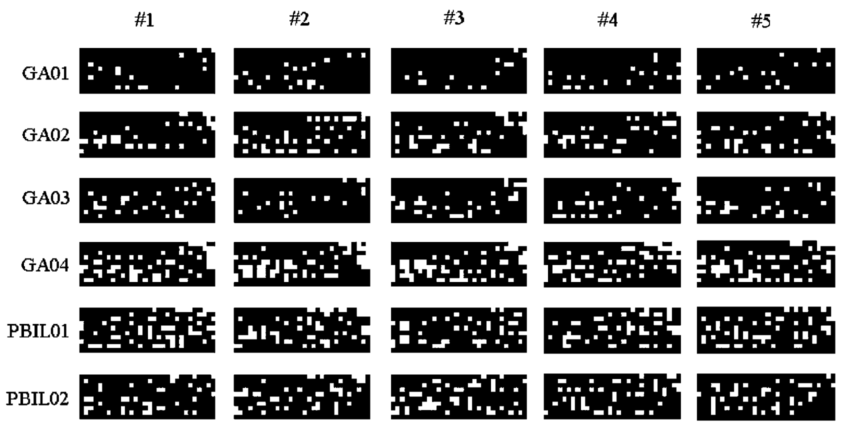

Fig. 35. CASE3 design results from GAs and PBILs with DSV1.

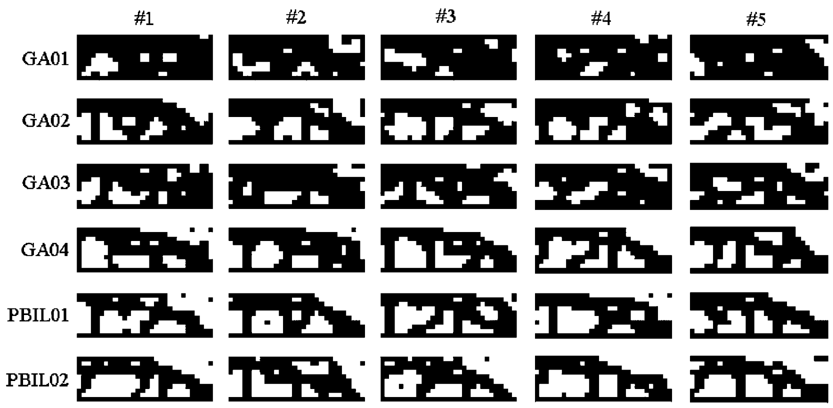

Fig. 36. CASE3 design results from GAs and PBILs with DSV2.

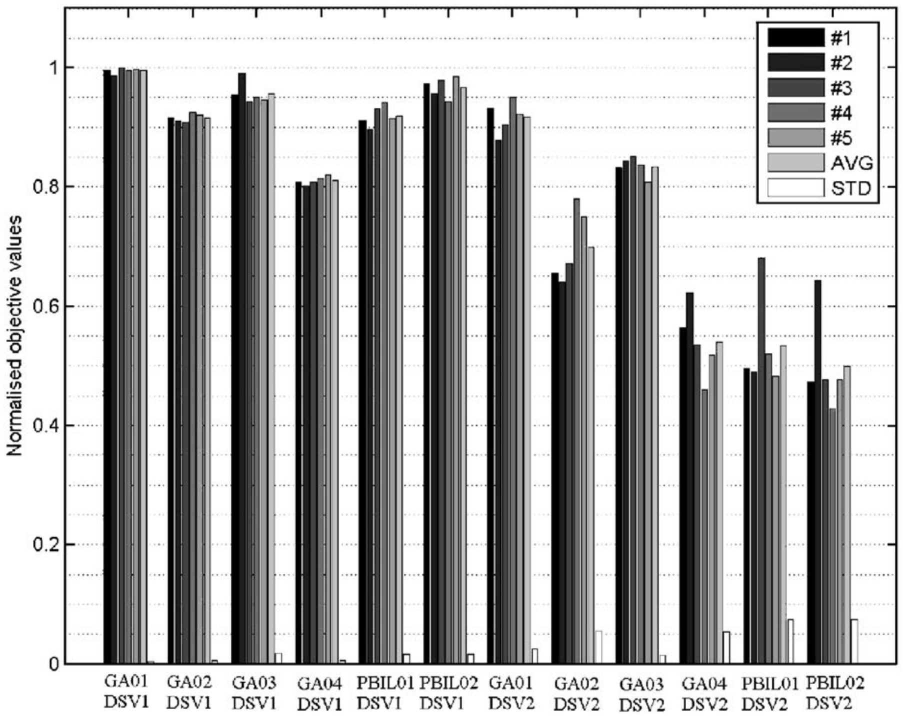

Fig. 37. CASE3 normalised objective values from GAs and PBILs.

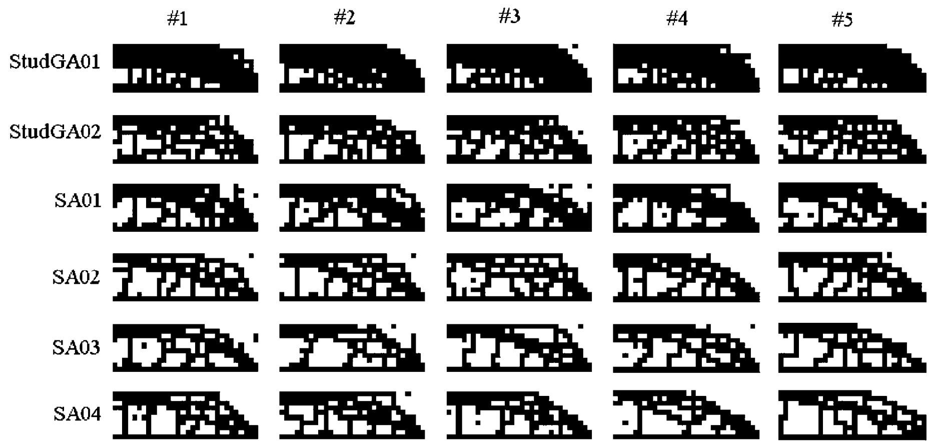

Fig. 38. CASE3 design results from stud-GAs and SAs with DSV1.

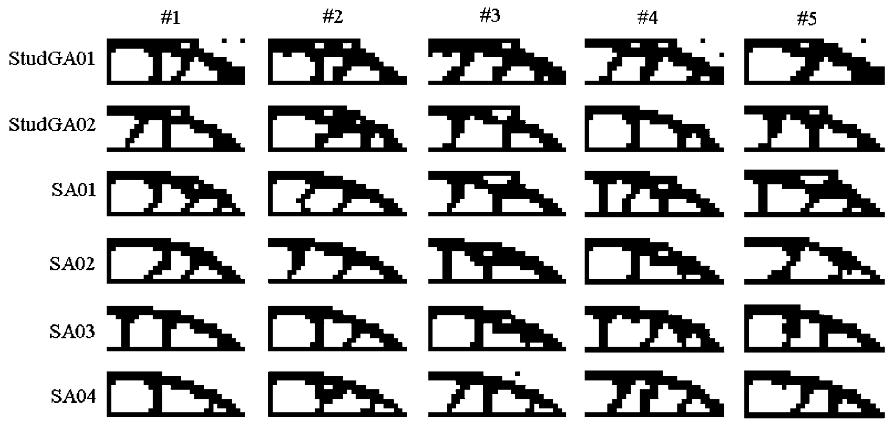

Fig. 39. CASE3 design results from stud-GAs and SAs with DSV2.

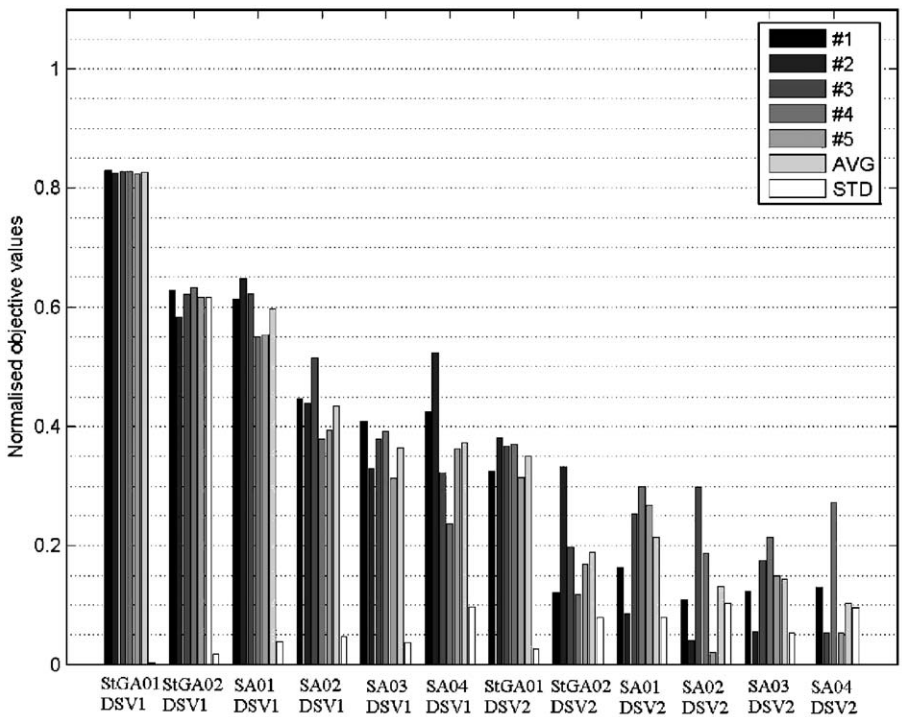

Fig. 40. CASE3 normalised objective values from stud-GAs and SAs.

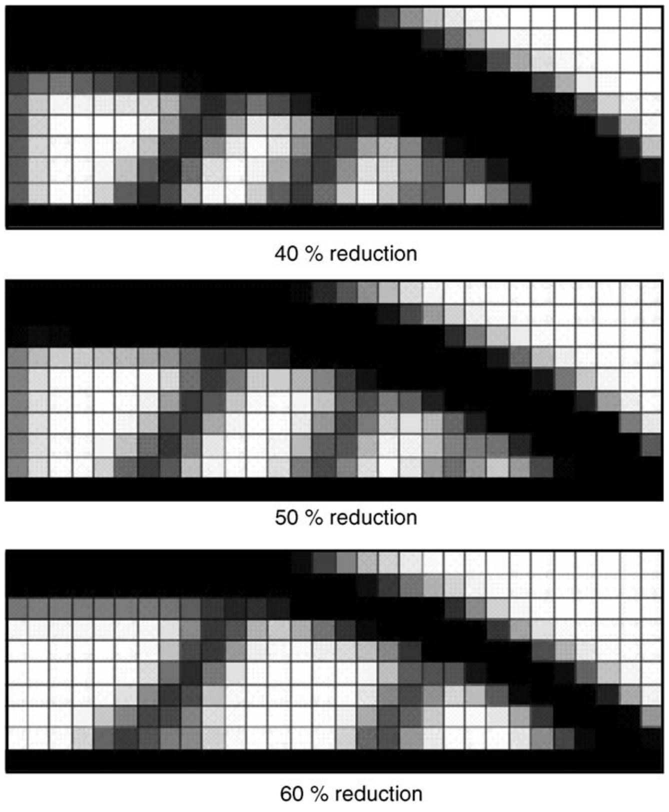

Fig. 41. Optimal topologies of CASE3 from using OCM.

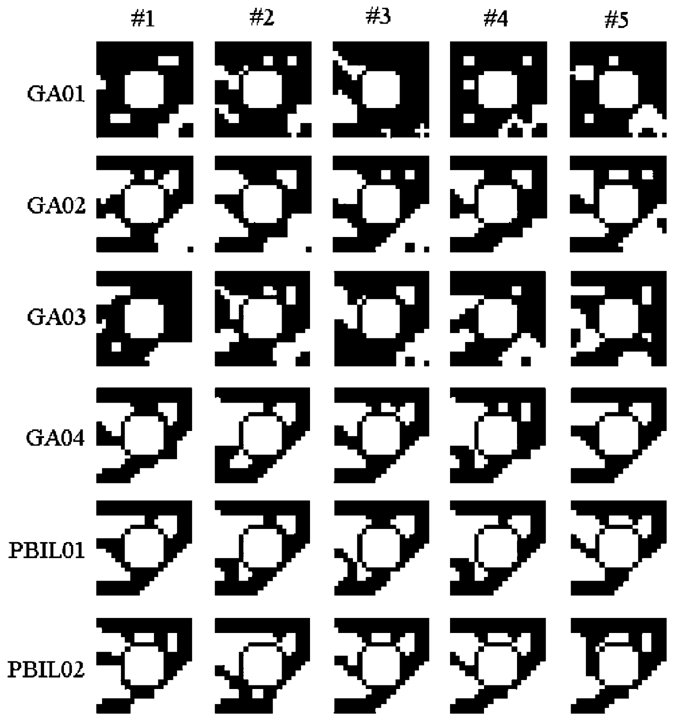

Fig. 43. CASE4 design results from GAs and PBILs with DSV2.

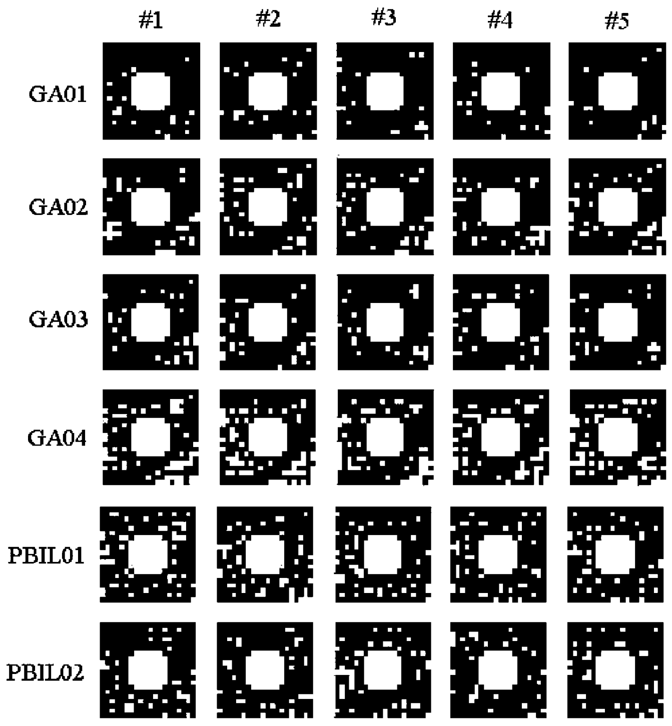

Fig. 42. CASE4 design results from GAs and PBILs with DSV1.

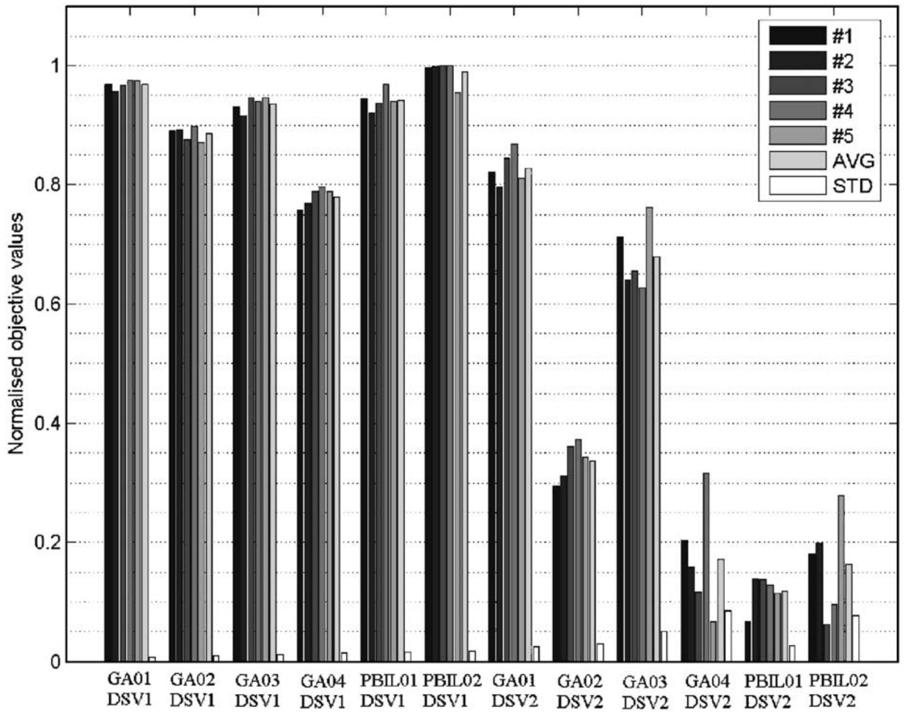

Fig. 44. CASE4 normalised objective values from GAs and PBILs.

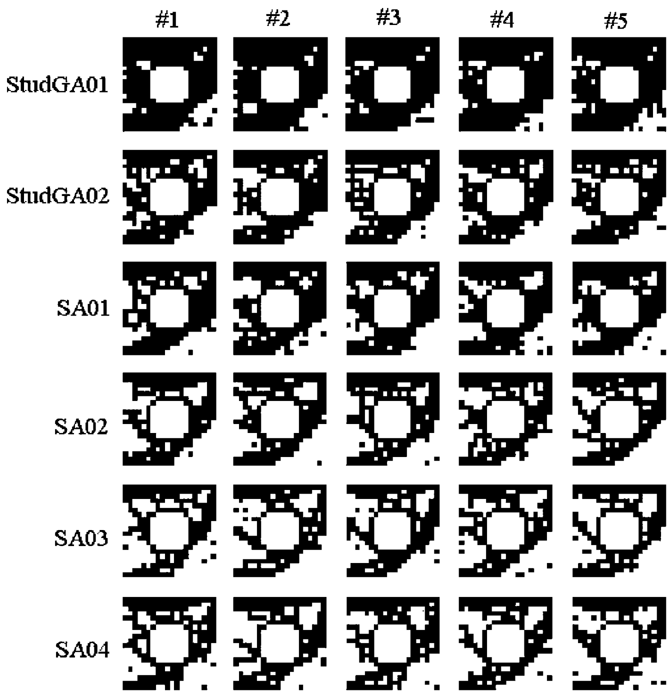

Fig. 45. CASE4 design results from stud-GAs and SAs with DSV1.

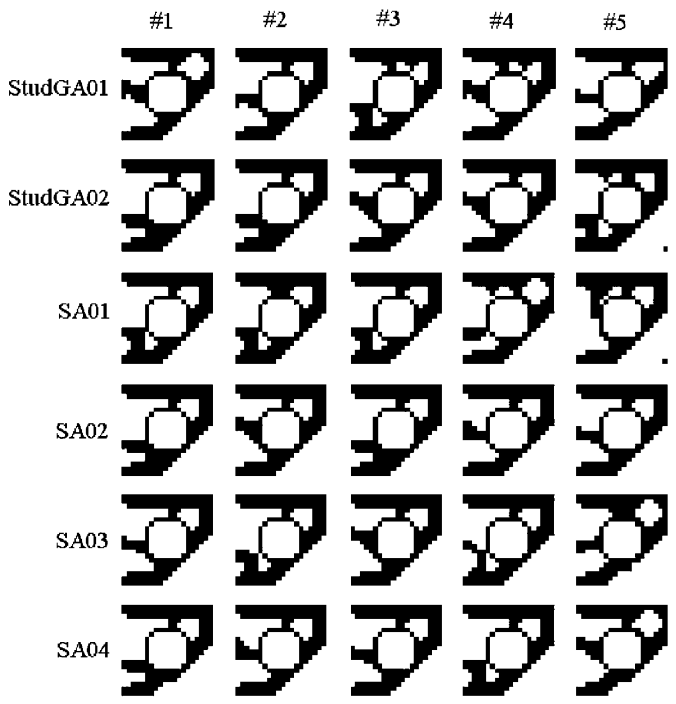

Fig. 46. CASE4 design results from stud-GAs and SAs with DSV2.

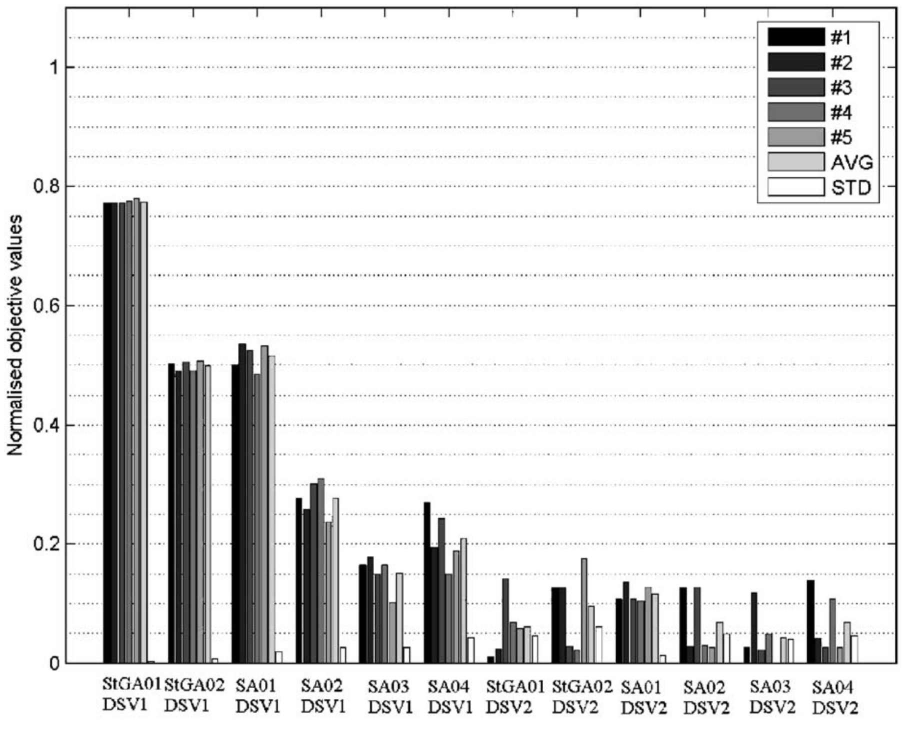

Fig. 47. CASE4 normalised objective values from stud-GAs and SAs.

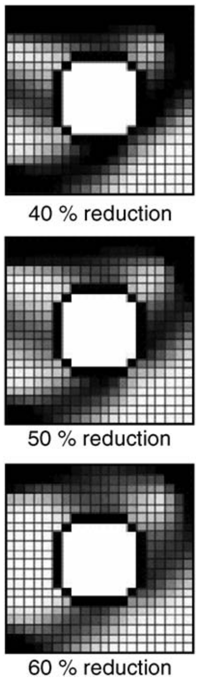

Fig. 48. Optimum topologies of CASE4 from using OCM.

designer experience. The better approach could be to use a multi-objective evolutionary optimiser (having mutation as the main operator) so that multiple aspects of topologies can be obtained within a single run of the method and the problem of selecting weighting factors would be avoided.

Acknowledgements

The authors are grateful for support from the Thailand Research Fund (TRF).

References

[1] M.P. Bensøe, O. Sigmund, Topology Optimization Theory, Method and Applications, Springer, Berlin, Heidelberg, 2003.
[2] N.L. Pederson, Maximization of eigenfrequencies using topology optimization, Struct. Multidiscip. Optim. 20 (2000) 2–11.
[3] O. Simund, A 99 Line topology optimization code written in MATLAB, J. Struct. Multidiscip. Optim. 21 (2001) 120–127.
[4] C. Kane, F. Jouve, M. Schoenauer, Structural topology optimization in linear and nonlinear elasticity using genetic algorithms, 21st ASME Design Automatic Conference, Boston, 1995.
[5] C. Kane, M. Schoenauer, Topological optimum design using genetic algorithms, J. Control Cybernet. 25 (5) (1996) 1059–1088.
[6] M. Jakiela, C. Chapman, J. Duda, A. Adewuya, K. Saitou, Continuum structural topology design with genetic algorithms, J. Comput. Methods Appl. Mech. Eng. 86 (2) (2000) 339–356.
[7] S.Y. Wang, K. Tai, Graph representation for structural topology optimization using genetic algorithms, Comput. Struct. 82 (2004) 1609–1622.
[8] T. Kunakote, S. Bureerat, Structural topology optimisation using evolutionary algorithms, 17th ME-NETT Conference, Prajeenburee, Thailand, 2003, pp. 109–115 (in Thai).
[9] T. Kunakote, Structural topology optimisation using evolutionary algorithms: comparison of the methods and checkerboard suppression, Master Thesis, Department of Mechanical Engineering, Khon Kaen University, 2003.
[10] S. Bureerat, T. Kunakote, A simple checkerboard suppression technique for topology optimisation using simulated annealing, 17th ME-NETT Conference, Prajeenburee, Thailand, 2003, pp. 85–90.
[11] S. Bureerat, J.E. Cooper, Evolutionary methods for the optimisation of engineering systems, Conference on Optimization in Control: Methods and Applications, IEE, London, 1998, pp. 1/1–1/10.
[12] G. Lindfield, J. Penny, Numerical Methods Using MATLAB, Ellis Horwood, Chichester, UK, 1995.
[13] W. Khatib, P. Fleming, The stud GA: a mini-revolution, 5th International Conference on Parallel Problem Solving From Nature, 1998.
[14] S. Baluja, Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning, CMU_CS_163, 1994.
[15] S. Kirkpatrik, C.D. Gelatt Jr., M.P. Vecchi, Optimization by simulated annealing, J. Sci. 220 (4598) (1983) 671–680.
[16] W.A. Benage, A.K. Dhingra, Single and multiobjective structural optimization in discrete-continuous variables using simulated annealing, Int. J. Numer. Methods Eng. 38 (1994) 2753–2773.
[17] S. Bureerat, J. Limtragool, Structural compliance minimisation using evolutionary algorithms with surface spline interpolation, 18th ME-NETT Conference, Khon Kaen, Thailand, 2004, pp. 319–324.