

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Optimization of Structures under Load Uncertainties Based on Hybrid Genetic Algorithm

Nianfeng Wang and Yaowen Yang
Nanyang Technological University
Singapore

1. Introduction

Today with the development of industry and research, reliability is more and more emphasized. However in some analysis of engineering problem, uncertainty inevitably exists. For example, wind loading on structures often varies over certain range instead of one determined value. The design of such systems should take into account the intervals of deviation of variables from their nominal values. In some uncertainty cases, the variety in the system parameters is neglected for the convenience of analysis. In certain situation it is possible to obtain the response which is valid and reasonable. However, sometimes the uncertainty analysis of system can not be neglected because the uncertainty would significantly affect the system performance.

Real structures are designed for numerous combinations of loads and load cases. In structure design it is typical that there may be several hundred important design load cases, which implies uncertainty in the loading conditions. A number of approaches have been developed to tackle such problems. According to the different types of uncertainty, several distinct methods were introduced to handle the analysis: fuzzy sets and fuzzy logic, interval analysis, and probabilistic approach. Fuzzy set theory was firstly introduced in (Zadeh 1978). The application of fuzzy sets concept into structural analysis was studied in (Ayyub 1997) systematically. In his work, a fuzzy set was utilized to describe every objective function. Some mechanical systems can be analyzed through this method (Qiu and Rao 2005). Fuzzy set theory has several advantages including the fact that the mathematics is formally defined and it can provide a potentially simple approach. The disadvantages of fuzzy set theory include validation necessary, justification of each step necessary, complexity when using many variables and the fact that the membership function may be heuristically defined. When less information is given and only range of each uncertainty quantity can be specified, interval method is widely used. Bounded uncertainty approach (Ben-Haim and Elishakoff 1990), a convex model of uncertainty was an extension to the interval analysis (Moore 1966). Some disadvantages of interval analysis are: (1) Interval arithmetic computations are slower than floating point operations, roughly by a factor of three, though there are problems that are solved faster when implemented using interval arithmetic; (2) There are no additive or multiplicative inverses for intervals; (3) There is no strict distributive law of multiplication over addition, only sub-distributivity; and (4) the

Source: Evolutionary Computation, Book edited by: Wellington Pinheiro dos Santos,
ISBN 978-953-307-008-7, pp. 572, October 2009, I-Tech, Vienna, Austria

number of required subproblems is much larger than that of the underestimator algorithms (Han, Manousiouthakis et al. 1997; Ju, Vasilios et al. 1997; Agoston 2005). For problems with distribution description of the variety in the system parameters, probabilistic approach is often used (Au 2005). Probabilistic approaches offer an attractive framework for designing structures in the presence of uncertainty, but they require much information about probabilistic models. When such information is inaccurate, large errors could be incurred in the calculation of failure probabilities (Elishakoff 1995).

Anti-optimization technique, on one hand, represents an alternative and complement to traditional methods, and on the other hand, it is a generalization of the mathematical theory of interval analysis (Qiu & Elishakoff 2001). When the available uncertainty data is limited, a probability distribution may not be able to be estimated accurately, but bounds for the uncertain variables may be at least estimated. The designer will generally seek for the least favorable solution for the structure within the domain defined by the bounds on the uncertain variables. This search for the worst condition for a given problem was named anti-optimization (Elishakoff 1995). The term anti-optimization was also used in a more general sense, to describe the task of finding the worst scenario for a given problem. A two species genetic algorithm (GA) was presented effectively reducing the two-level problem to a single level (Venter & Haftka 1996). The maximum strength of laminated composites was optimized under bounded uncertainty of material properties by GA (Maenghyo & Seung Yun 2004).

In recent years, hybrid genetic algorithms (GAs) have become more homogeneous and some great successes have been achieved in the optimization of a variety of classical hard optimization problems. Hybrid algorithms of GAs and those local search algorithms were proposed to improve the search ability of GAs, and their high performance was reported (Ishibuchi & Murata 1998; Deb & Goel 2001; Jaszakiewicz 2003; Wang & Tai 2007; Wang & Tai 2008). In these studies local search algorithms were suggested in order to reach a quick and closer result to the optimum solution. However, this work integrates a simple genetic local search algorithm as the anti-optimization technique with a constrained multi-objective GA proposed in (Wang & Tai 2007). A constrained tournament selection is used as a single objective function in the local search strategy. Section 2 outlines the proposed hybrid GA. And section 3 presents a morphological geometry representation scheme coupled with the GA. Formulations and numerical results of a target matching test problem in the context of structural topology optimization are presented in Section 4. Formulations and numerical results of the structural design problem are presented in Section 5. Finally, concluding remarks are given in Section 6.

2. Proposed algorithm –a hybrid GA

A general constrained multi-objective optimization problem (in the minimization sense) is formulated as:

$$\begin{aligned} &\text{minimize } \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \quad f_2(\mathbf{x}) \quad \cdots \quad f_m(\mathbf{x})] \\ &\text{subject to } g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, q \\ &\quad \quad \quad h_k(\mathbf{x}) = 0, \quad j = 1, 2, \dots, q \\ &\quad \quad \quad \mathbf{x}^L < \mathbf{x} < \mathbf{x}^U \end{aligned} \tag{1}$$

where \mathbf{f} is a vector of m objectives, and $\mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_n]$ is the vector of n design variables. g_j and h_k are the equality and inequality constraints. \mathbf{x}^L and \mathbf{x}^U define the lower bound and upper bound of \mathbf{x} , respectively.

For anti-optimization, the robustness of the objective function can be achieved by maximizing the minimum value of the objective functions. It tends to move the uncertainty parameters to the desirable range and results in higher reliability with respect to uncertainty. Therefore an anti-optimization procedure implemented in this work by local search is employed to search for the values of the worst objective functions. Consider a problem subject to u uncertain variables \mathbf{x}_u and $n-u$ normal design variables \mathbf{x}_n . The formulation can then be written as below:

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && \mathbf{f}(\mathbf{x}) \\ & \underset{\mathbf{x}_u}{\text{maximize}} && \mathbf{f}(\mathbf{x}_n, \mathbf{x}_u) \\ & \text{subject to} && \mathbf{x}_n \in \mathbf{C}_N \\ & && g_j(\mathbf{x}) \leq 0, \quad j=1, 2, \dots, q \\ & && h_k(\mathbf{x}) = 0, \quad k=1, 2, \dots, q \\ & && \mathbf{x}^L < \mathbf{x} < \mathbf{x}^U \end{aligned} \quad (2)$$

where \mathbf{C}_N is a particular set of solutions which are related to generation number N . The formulation considered leads to a nested optimization problem which will be solved by means of a GA for optimization and a local search for anti-optimization. Generally speaking, this is a sort of minmax search, where the "max" part is dealt with by a local search algorithm, and the "min" part is realized by a GA.

2.1 Tournament selection for local search

Since a local search strategy requires a tournament selection between an initial solution and its neighboring solution, a comparison strategy is needed. For multi-objective optimization without constraints, a single objective function converted from multiple objectives can be used. For constrained optimization, constraint handling mechanisms should be given first. In most applications, penalty functions using static (Srinivas & Deb 1994), dynamic or adaptive concepts (Michalewicz & Schoenauer 1996) have been widely used. The major problem is the need for specifying the right value for penalty parameters in advance. The method from (Ray, Tai et al. 2001) incorporates a Pareto ranking of the constraint violations, so it does not involve any aggregation of objectives or constraints and thus the problem of scaling does not arise. Note that Pareto ranking is not well suited for hybridization with local search. For an anti-optimization problem, when Pareto ranking is used, the current solution \mathbf{x} is replaced with its neighboring solution \mathbf{y} (i.e., the local search move from \mathbf{x} to \mathbf{y} is accepted) only when \mathbf{x} dominates \mathbf{y} (i.e., \mathbf{x} is better than \mathbf{y}). That is, the local search move is rejected when \mathbf{x} and \mathbf{y} are non-dominated with respect to each other. However, change of the rank of a given solution may require significant changes of the objective/constraint values, thus, many local moves will not degenerate the rank.

A constraint handling method (Deb 2000) was proposed which is also based on the penalty function approach but does not require the prescription of any penalty parameter. The main idea of this method is to use a tournament selection operator and to apply a set of criteria in the selection process. For an anti-optimization problem, it can be easily changed to:

- i. Any infeasible solution is preferred to any feasible solution.
- ii. Between two feasible solutions, the one having worse objective function value is preferred.
- iii. Between two infeasible solutions, the one having bigger constraint violation is preferred.

According to these criteria, the constrained optimization can be constructed as

$$\tilde{f}(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{if } \mathbf{x} \in F \\ f_{\max} + \text{vio}(\mathbf{x}) & \text{otherwise} \end{cases} \quad (3)$$

where $\tilde{f}(\mathbf{x})$ is the artificial unconstrained objective function, F is the feasible region of the design domain, f_{\max} is the objective function value of the worst feasible solution in the population, and $\text{vio}(\mathbf{x})$ is the summation of all the violated constraint function values. However, this approach is only suitable for single-objective constrained optimization problem if no further handling mechanisms for multiple objectives are given. And $\text{vio}(\mathbf{x})$ as summation of constraint values cannot reflect the real relative comparison between them because of different orders of magnitude among the constraints, and in this sense, it is also based on the penalty functions where all the penalty parameters are set to 1.

Extending the basic idea of Deb's method, a technique combining Pareto ranking and weighted sum is suggested in this work for the local search selection process. There are only 3 combinations for the two solutions: both feasible, both infeasible, and one feasible and the other infeasible. The main idea of the technique is to use a tournament selection operator and to apply a set of criteria in the selection process. For an anti-optimization procedure, any infeasible solution is preferred to any feasible solution. When both solutions are feasible, Pareto ranking based on objectives is calculated. The one with bigger rank value is preferred. If the situation still ties, a more sophisticated acceptance rule is used for handling the situation. The fitness function of the solution \mathbf{x} is calculated by the following weighted sum of the m objectives:

$$f(\mathbf{x}) = w_1 f_1(\mathbf{x}) + w_2 f_2(\mathbf{x}) + \cdots + w_m f_m(\mathbf{x}) \quad (4)$$

where $f(\mathbf{x})$ is a combined objective, and w_1, w_2, \dots, w_m are nonnegative weights for the objectives set according to different orders of magnitude among them. Constant weight values are used in this work to fix the search direction based on user's preference. The solution with a bigger $f(\mathbf{x})$ will survive. When both solutions are infeasible, Pareto ranking based on constraints is calculated. The one with bigger rank value is preferred. If the rank is same, the one with worse fitness value survives. A tournament selection criterion can be described as below to decide whether a current solution \mathbf{x} should be replaced by a neighboring solution \mathbf{y} :

- i. If \mathbf{x} is feasible and \mathbf{y} is infeasible, replace the current solution \mathbf{x} with \mathbf{y} (i.e., let $\mathbf{x} = \mathbf{y}$).
- ii. If both \mathbf{x} and \mathbf{y} are feasible, then if $\text{RankObj}_x < \text{RankObj}_y$, then $\mathbf{x} = \mathbf{y}$, else if $\text{RankObj}_x = \text{RankObj}_y$ and $f(\mathbf{x}) < f(\mathbf{y})$, then $\mathbf{x} = \mathbf{y}$.
- iii. If both \mathbf{x} and \mathbf{y} are infeasible, then if $\text{RankCon}_x < \text{RankCon}_y$, then $\mathbf{x} = \mathbf{y}$, else if $\text{RankCon}_x = \text{RankCon}_y$ and $f(\mathbf{x}) < f(\mathbf{y})$, then $\mathbf{x} = \mathbf{y}$.

2.2 Selection of initial solutions

Local search applied to all solutions in the current population in the algorithm is inefficient, as shown in (Ishibuchi, Yoshida et al. 2002). In the proposed algorithm, the computation time spent by local search can be reduced by applying local search to only selected solutions in selected generations. If n is the number of decision variables, the best n number of

solutions from the current population (based on Pareto ranking) are selected. These n number of mutated solutions and elites from N -th generation after local search are then put into the next population. The generation update mechanism in the proposed algorithm is shown in Fig. 1. The implementation of the anti-optimization part is modularized.

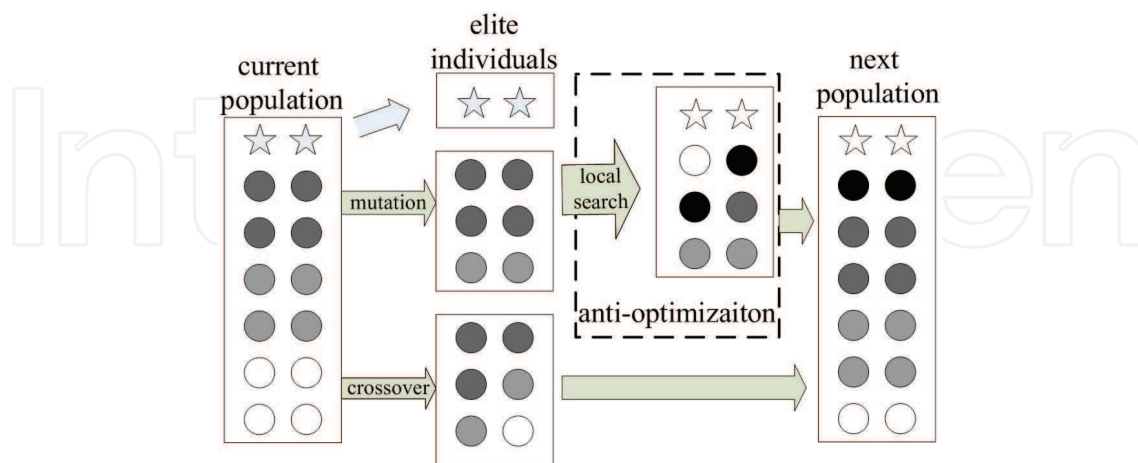


Fig. 1. Generation update mechanism.

2.3 Local search procedure

As explained in the above, a local search procedure is applied to elite individuals and new solutions generated by the mutation in selected generations. Generally, a local search procedure can be written as follows:

- Step 1. Specify an initial solution and its corresponding design variable under uncertainty.
- Step 2. Apply Hooke and Jeeves Method to determine the search path using the tournament selection criteria stated above as the function values.
- Step 3. If the prescribed condition is satisfied, terminate the local search.

2.4 Main algorithm

The overall algorithm uses a framework which combines the method stated in (Wang & Tai 2007) and the local search proposed above. The algorithm is given below:

- Step 1. Generate random initial population P of size M .
- Step 2. Evaluate objective as well as constraint functions for each individual in P .
- Step 3. Compute Pareto Ranking.
- Step 4. Select elite individuals. Elite individuals carried from the previous generation preserve the values of their objective and constraint functions.
- Step 5. Select n best individuals from P , mutate and apply local search procedure in specified generation, then put them into new population P' .
- Step 6. Crossover.
- Step 7. If a prescribed stopping condition is satisfied, end the algorithm. Otherwise, return to Step 2.

3. Enhanced geometric representation scheme for structural topology optimization

The enhanced morphological representation was first introduced in (Tai & Wang 2007), which in turn is an extension of the morphological representation scheme previously

developed in (Tai & Akhtar 2005; Tai & Prasad 2007). As in any structural topology optimization procedure, the geometry of the structure has to be represented and defined by some form of design variables. The enhanced morphological representation efficiently cast structure topology as a chromosome code that makes it effective for solution via a GA. In the proposed scheme, the connectivities and the number of curves used are made variable and to be optimized in the evolutionary procedure. The process of the scheme definition is illustrated as follows.

A square design space shown in Fig. 2(a) is discretized into a 50 by 50 mesh of identical square elements. While it is initially unknown how the design space will be occupied by the structure, there must exist some segments of the structure such as the support and the loading that have functional interactions with its surroundings. The support point is some segment of the structure that is restrained (fixed, with zero displacement) while the loading point is where some specified load (input force) is applied to deform the structure. Collectively, the support and loading points represent the input points of the structure. There is also usually an output point which is some segment of the structure where the desired output behavior is attained. As shown in Fig. 2(a), the problem is defined with four I/O locations, each made up of one element in black. Six connecting curves in the illustration of Fig. 2(b), three of which are active and three of which are inactive, are used such that there is one connecting curve between any two points (i.e. every I/O point is directly connected to the other three). Before continuing, it is important to make a clear distinction between the *active* and *inactive* curves. The active curves are the curves which are in the 'on' state. The structure is generated based only on the active curves. Although the inactive curves, which are in the 'off' state, temporarily contribute nothing to the current structure, they are still very important in subsequent generations because they may be turned 'on' later through the crossover or mutation operations. In Fig. 2(b), the active curves are marked with thick lines and the inactive with thin dotted lines. The connectivity of the I/O points is based on all connecting active curves joining one point to another. Each curve is a Bezier curve defined by the position vector which can be derived from the element number of control point. The set of elements through which each active curve passes form the 'skeleton' (Fig. 2 (c)). Some of the elements surrounding the skeleton are then included to fill up the structure to its final form (Fig. 2 (d)) based on the skeleton's thickness values. Each curve is defined by three control points, and hence each curve has four thickness values. And the union of all skeleton, surrounding elements and I/O elements constitute the structure while all other elements remain as the surrounding empty space.

In order to use a GA for the optimization, the topological/shape representation variables have to be configured as a chromosome code. Hence the structural geometry in Fig. 2 (d) can be encoded as a chromosome in the form of a graph as shown in Fig. 3. Each curve is represented by a series of nodes connected by arcs in the sequence of start element number, thickness values alternating with control element number and end point. For identification purpose, the active curves are shown by solid lines and the inactive curves are represented by dotted lines. Altering the curve states can vary the connectivity of the I/O regions, and therefore the representation scheme can automatically decide the connectivity. The resulting scheme therefore increases the variability of the connectivity of the curves and hence the variability of the structure topology.

Two of the important operations in a GA are the crossover and mutation. In this implementation, the crossover operator works by randomly sectioning any single connected

subgraph from a parent chromosome and swapping with a corresponding subgraph from another parent (as shown in Fig. 3.). As a result, two offsprings are produced which have a mix of the topological/shape characteristics of the two parents, and the advantages of the representation (such as no checkerboard patterns and single-node hinge connections) are maintained in the offspring. The 'on' and 'off' state of different curves which are crossed by the loop are also swapped. If the 'on' variables dominate a curve, i.e. when the number of 'on' variables is more than the number of 'off' variables, the curve of in the child chromosome will be active. Otherwise, the child curve will be inactive. As for mutation, the mutation operator works by randomly selecting any vertex of the chromosomal graph and altering its value to another randomly generated value within its allowable range. Mutation about the on-off state is simple, which is altering the state of curves. When the selected curve is active, it will be inactive after mutation, and vice verse.

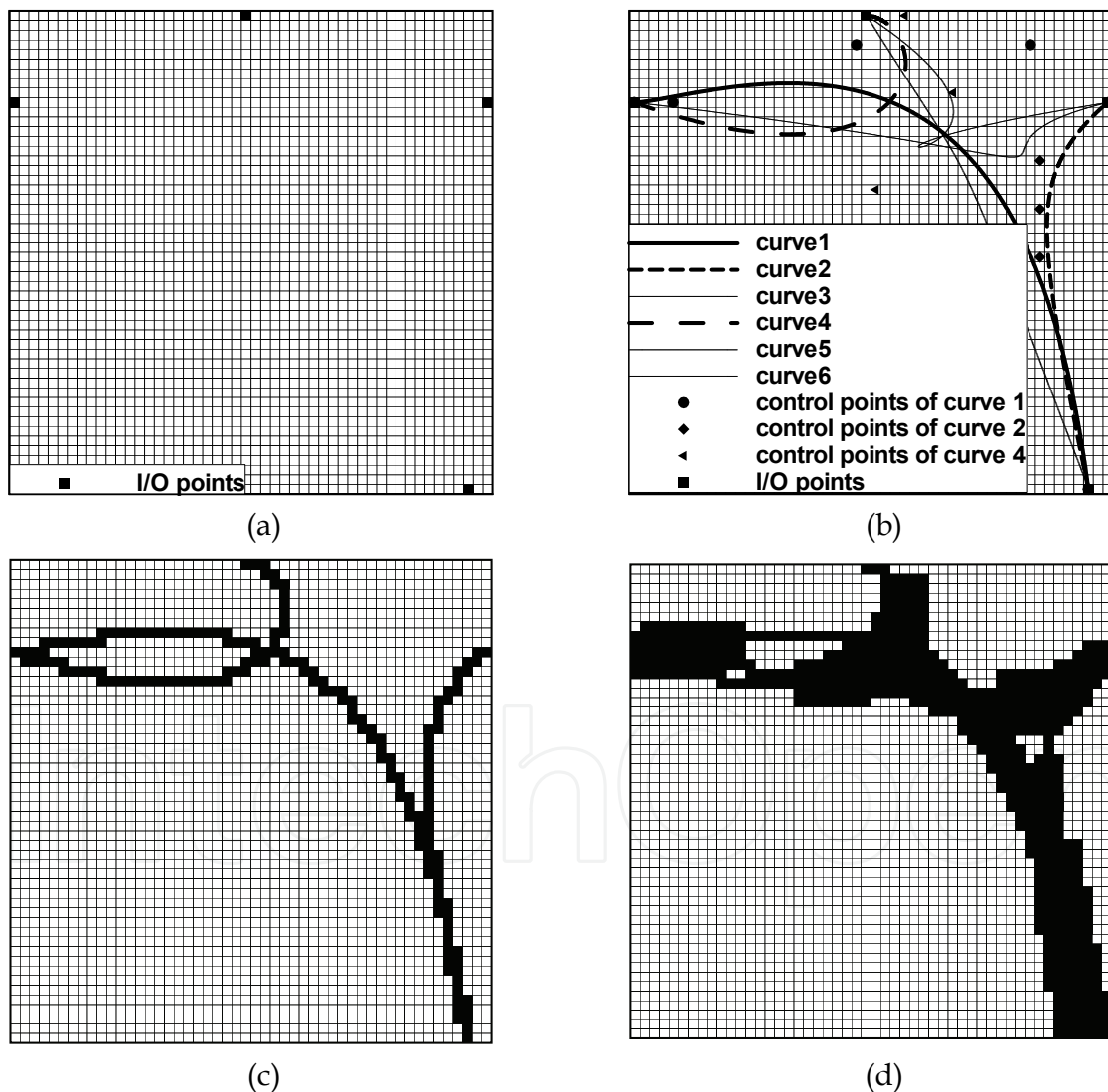


Fig. 2. Definition of structural geometry by enhanced morphological scheme (a) FE discretization of design space (I/O element marked in black) (b) Connecting I/O elements with Bezier curves (c) Skeleton made up of elements along curves (d) Surrounding elements added to skeleton to form final structure.

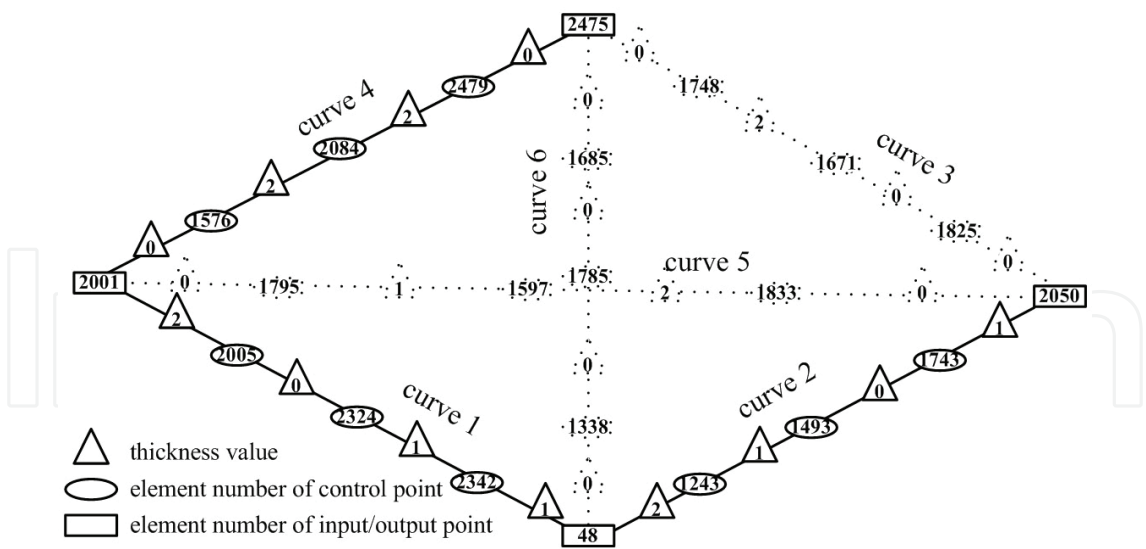


Fig. 3. Chromosome of final structure

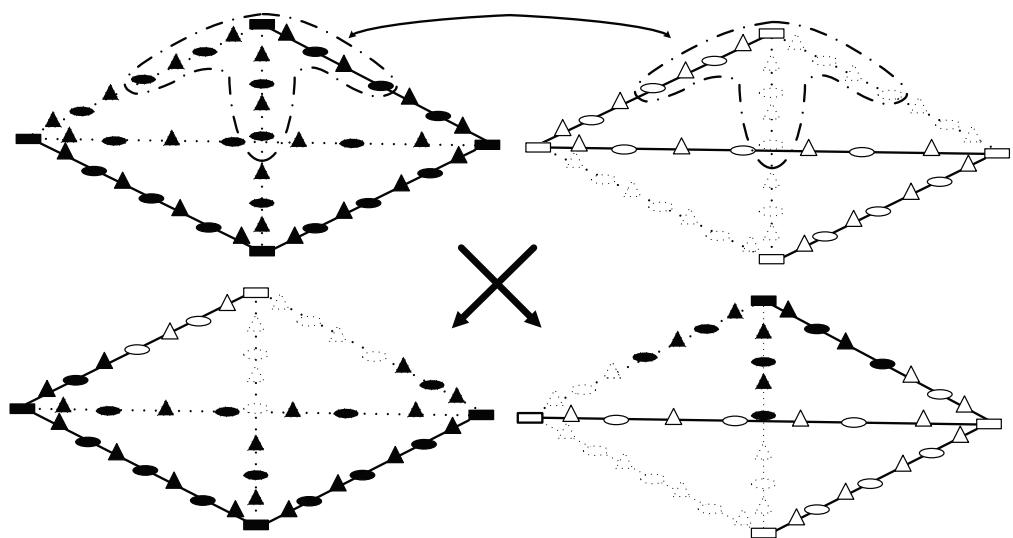


Fig. 4. Illustration of crossover operation

In summary, this morphological representation scheme uses arrangements of skeleton and surrounding material to define structural geometry in a way that will not render any undesirable design features such as disconnected segments, checkerboard patterns or single-node hinge connections because element edge connectivity of the skeleton is guaranteed, even after any crossover or mutation operation. Any chromosome-encoded design generated by the evolutionary procedure can be mapped into a finite element model of the structure accordingly.

4. Target matching problem

Before a GA is relied upon for solving a structure design problem with unknown solutions, it is important that the performance of the GA be tested and tuned by using it to solve a problem with known solutions. Various kinds of test problems (Michalewicz, Deb et al. 2000; Schmidt & Michalewicz 2000; Martin, Hassan et al. 2004) have been established for

testing multi-objective GAs. They were created with different characteristics, including the dimensionality of the problem, number of local optima, number of active constraints at the optimum, topology of the feasible search space, etc. However, all of these test problems have well-defined objectives/constraints expressed as mathematical functions of decision variables and therefore may not be ideal for evaluating the performance of a GA intended to solve problems where the objectives/constraints cannot be expressed explicitly in terms of the decision variables. In essence, a GA is typically customized to tackle a certain type of problem and therefore 'general-purpose' test problems may not correctly evaluate the performance of the customized GA. The test problem should, therefore, ideally suit (or be customized to) the GA being used.

In numerous real-life problems, objectives/constraints cannot be expressed mathematically in terms of decision variables. One of such real-life problems is structural topology optimization, where a procedure (structure geometry representation scheme) first transforms decision variables into the true geometry of the designed structure and then finite element analysis of the designed structure is carried out for evaluating the objectives/constraints. The GA solving such problems may have special chromosome encoding to suit the structure geometry representation used and there may also be specially devised reproduction operators to suit the chromosome encoding used. As such, the structure geometry representation scheme, the chromosome encoding and the reproduction operators introduce additional characteristics to the search space and, therefore, they are very critical to the performance of the GA. The test problem for such GAs, therefore, must use the same structure geometry representation scheme, chromosome encoding and reproduction operators. The conventional test problems found in literature cannot make use of the GA's integral procedures such as structure geometry representation scheme and therefore they are not suitable for testing such GAs.

Ideally, the test problem should emulate the main problem to be solved. The test problem should be computationally inexpensive so that it can be run many times for the GA parameters to be changed or experimented with and the effect thereof can be studied for the purpose of fine-tuning the GA. However, the main problem in the present work, being a structural topology optimization problem under uncertainty, requires structural analysis which consumes a great deal of time. Taking the running time into consideration, the test problem needs to be designed without any need for structural analysis. A test problem emulating structural topology optimization does not necessarily need structural analysis as the main aim of topology optimization is to arrive at an optimal structural geometry. Without using structural analysis, if a GA is successfully tested to be capable of converging the solutions to any arbitrary but predefined and valid 'target' structural geometry, then it may be inferred that the GA would be able to converge design solutions to the optimal structural topology when solving an actual topology optimization problem. Based on this inference, a test problem can be designed such that simple geometry-based (rather than structural analysis based) objectives/constraints help design solutions converge towards the predefined target geometry. This type of test problem may be termed as "Target Matching Problem", which is capable of using exactly the same GA (including structure geometry representation scheme, chromosome encoding and reproduction operators) as that intended for solving the actual topology optimization problem. The present problem is similar to the Target Matching Problem solved in (Wang & Tai 2007; Tai et al. 2008; Wang & Yang 2009). The target matching problems are defined here as multiobjective optimization problems

under uncertainty which are more difficult (e.g. more nonlinear problem) and computationally intensive.

4.1 Formulation

The test problem makes use of the design space shown in Fig.5, which has one support point, two loading points and one output point. The problem does not represent a structural analysis problem, but the original terms ‘support’, ‘loading’ and ‘output’ are still used for ease of reference. The loading point 1 is positioned anywhere along the left boundary and loading point 2 is positioned anywhere along the right boundary. The position of output point is fixed as shown in Fig.5. The support point is positioned in a specified area marked as “under uncertainty” and its position is random in the area. In this problem, the target geometry is shown in Fig. 6. The aim is therefore to evolve structures that match as closely as possible this target geometry. The problem presented here is more difficult than the original problem described in (Wang & Tai 2007), since the support point is under uncertainty that makes the geometry more complex and not easy to converge to the target.

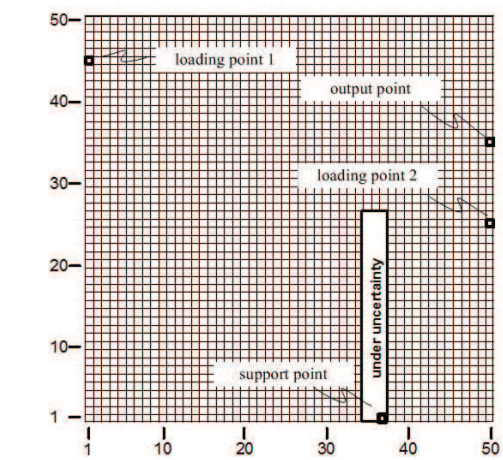


Fig. 5. Design space of Target Matching Problem.

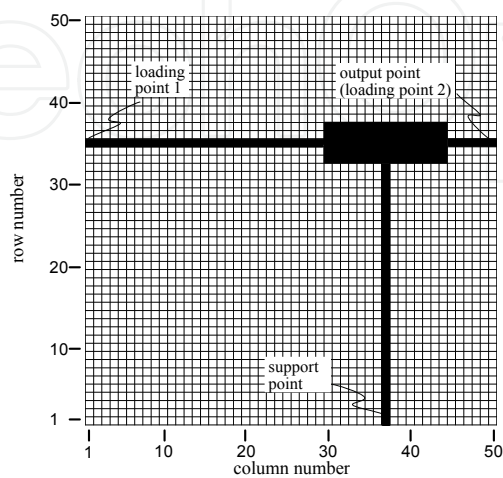


Fig. 6. Target geometry.

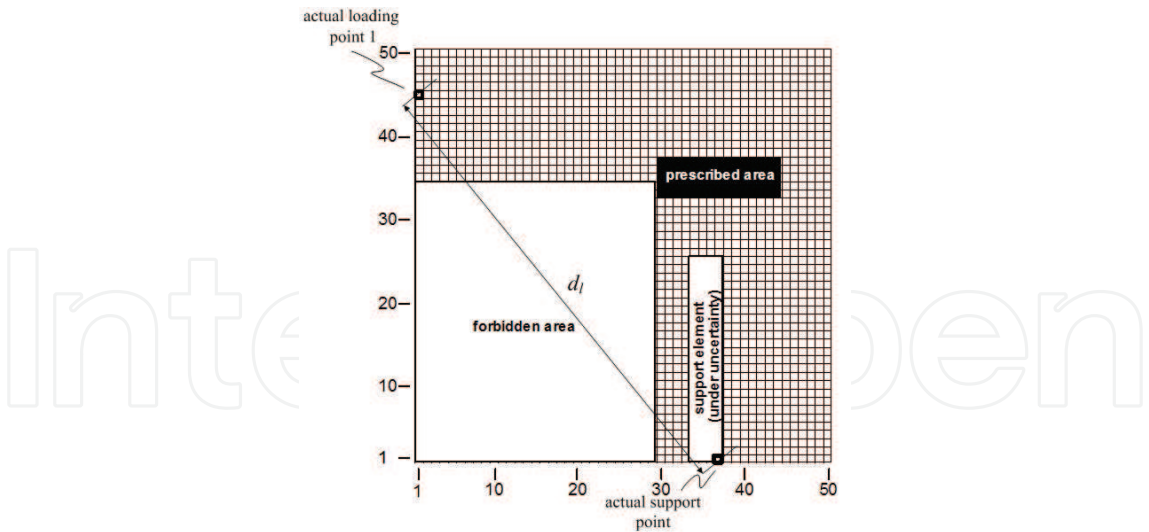


Fig. 7. Formulation of Target Matching Problem 1.

The problem is formulated with the following two objectives and two constraints: distance objective, material objective, forbidden area constraint and prescribed area constraint. Such a problem is defined with the help of Fig.7. The distance objective is given by

$$f_{distance} = d_l \tag{5}$$

where d_l is the centroid-to-centroid Euclidean distance between the actual loading point 1 and the actual support point.

The material objective is given by

$$f_{material} = \sum_{i=1}^n x_i \tag{6}$$

where x_i is the material density of the i -th element in the design space, with a value of either 0 or 1 to represent that the element is either void or material (solid), respectively. n is the total number of elements in the discretized design space. In other words, this objective function is defined as the summation of the material density of all elements in the current geometry.

The forbidden area constraint can be written as

$$g_{forbidden} = \sum_{i=1}^{n_f} y_i \leq 0 \tag{7}$$

where y_i is the material density of the i -th element in the forbidden area, and n_f is the total number of elements in the forbidden area. In other words, the summation of the material density of elements in the forbidden area is required to be less than or equal to zero.

The prescribed area constraint can be written as

$$g_{prescribed} = n_p - \sum_{i=1}^{n_p} z_i \leq 0 \tag{8}$$

where z_i is the material density of the i -th element in the prescribed area, and n_p is the total number of elements in the prescribed area. In other words, the summation of the material density of elements in the prescribed area is required to be more than or equal to the total number of elements in that area.

4.2 Main results

The target matching problem to be solved is defined by the design space shown in Fig. 5. The optimization was run for 1001 generations with a population size of 100 per generation. The local search procedure is triggered once every ten generations. By the end of evolutionary process 132,113 objective function evaluations have been performed. One of the solutions at the end of 1001 generations is shown in Fig. 8. It is the same as the target solution shown in Fig. 6.

As can be seen from the result, the support point is on the extreme point where the element number is 37. The following Fig. 9 illustrates how the solution shown in Fig. 8 is obtained by applying local search. Apply the Hooke and Jeeves method to determine the search path using the tournament selection criteria stated in Section 2. Each data point is labeled with its index where some indexes are coincided. At the start point, $f_{distance}$ is 37.6 and $f_{material}$ is 118. After the local search, the worst case labeled as 9 is obtained with a distance objective $f_{distance}$ value of 49.5 and a material objective $f_{material}$ value of 142. This figure demonstrates the Hooke and Jeeves direct search method for function maximization.

Fig. 10 shows a plot of the best distance objective $f_{distance}$ and the corresponding solution's $f_{material}$ versus generation number. $f_{distance}$ and $f_{material}$ values on the plot corresponding to any particular generation number belong to that generation's non-dominated feasible solution having the best distance objective. The plot starts at generation number 6, as until this generation there is no feasible solution in the population.

Fig. 11 shows plots of the best material objective $f_{material}$ and the corresponding solution's distance objective $f_{distance}$ versus generation number.

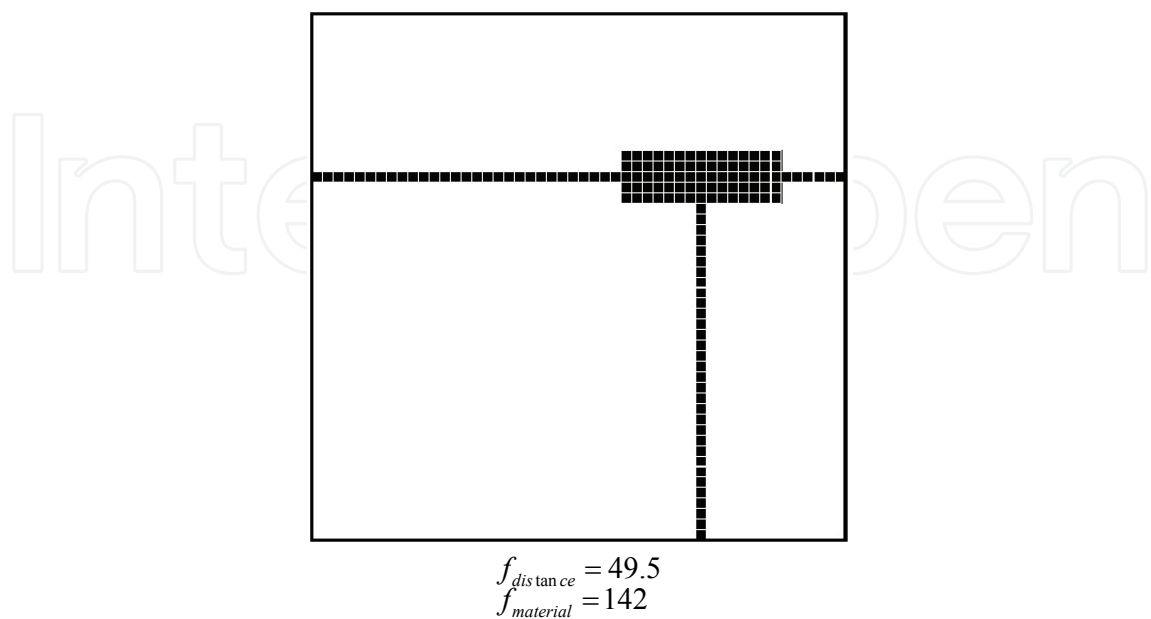


Fig. 8. The optimal solution at 1001st generation.

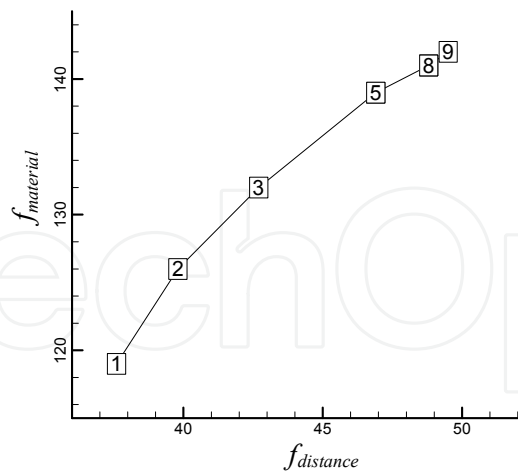


Fig. 9. Path of the local search.

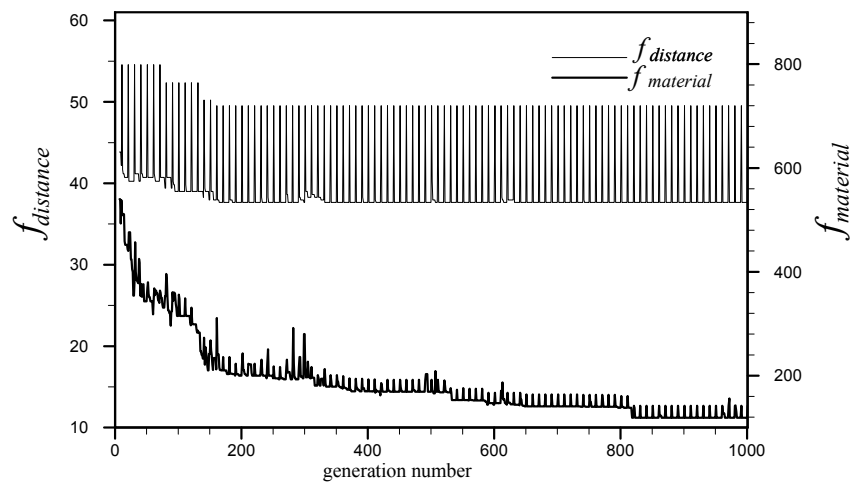


Fig. 10. History of the best distance objective ($f_{distance}$).

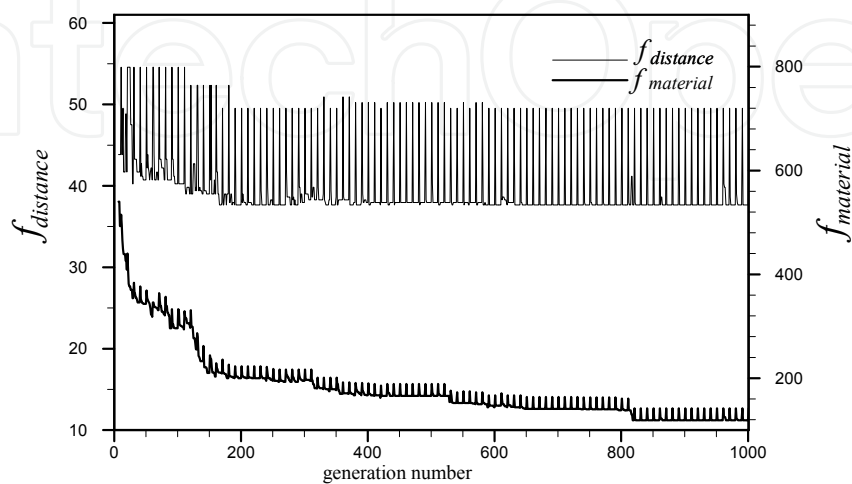


Fig. 11. History of the best material objective ($f_{material}$).

Fig.12 shows a plot in objective space, where the solid shape markers are used to denote the feasible non-dominated solutions at any particular sample generation, viz. the 51st, 101st, 301st, 501st and 1001st generation. Although Fig.12 shows all the non-dominated solutions at any particular generation, only one or two distinct points (in the objective space) can be seen for that generation. However, a few distinct solutions in the design variable space may have the same objective function values and therefore, such solutions would coincide in the objective space. The number shown in parenthesis next to every point marker indicates the total number of such coincident solutions.

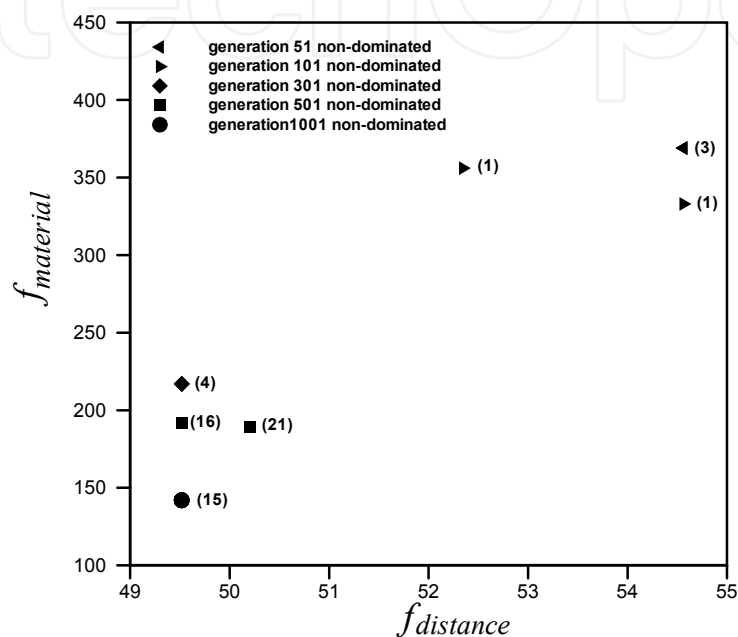


Fig. 12. History of the best distance objective ($f_{material}$).

4.3 Discussion

For the test problem, the results are summarized in Section 4.2. The hybrid GA proves its efficiency by converging the two objective function values ($f_{distance}$ and $f_{material}$) to the optimal values. The recurrent fluctuations in Fig. 10 and Fig. 11 show the effect of local search to the hybrid algorithm. Fig. 9 shows how the local search works.

5. Optimization of structures under load uncertainties

In this work, the structural optimization problem is defined in a design space shown in Fig. 13 using 4-noded quadrilateral elements. The present problem is similar to design problem solved in (Wang, Yang et al. 2008). The characteristics of the material aluminium used are as follows: the Young's Modulus E is 68948 MPa, and the mass density ρ is 2768 kg/m³. The maximum allowable vertical displacement $d_{2,allowable}$ at loading point 2 is 0.000635m. Loading point 1 is subjected to a vertical load P_1 while loading point 2 is subjected to both a vertical load P_2 and a horizontal load P_3 as shown in Fig. 13. P_2 and P_3 are normal loads with constant values of 222N and 44.4N respectively while P_1 is an uncertain load varying between 44.4N and 55.5N.

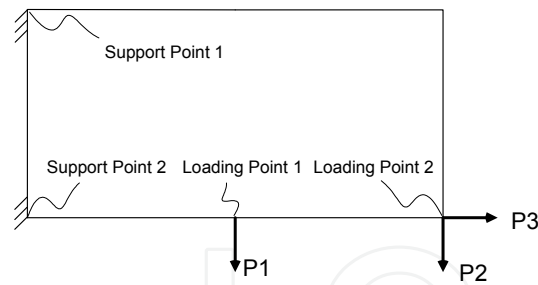


Fig. 13. Design domain.

The optimization problem can be formulated as follows:

$$\begin{aligned}
 & \underset{(\mathbf{x}_n, \mathbf{p})}{\text{minimize}} \quad \{w(\mathbf{x}_n, \mathbf{p}) \quad d(\mathbf{x}_n, \mathbf{p})\} \\
 & \underset{\mathbf{p}}{\text{maximize}} \quad \{w(\mathbf{x}_{n_0}, \mathbf{p}) \quad d(\mathbf{x}_{n_0}, \mathbf{p})\} \\
 & \text{subject to} \quad g_d = d - 0.000635 \leq 0 \\
 & \quad \quad \quad g_{stress} \leq 0
 \end{aligned} \tag{9}$$

where w is the weight of the structure and d is the displacement of the loading point 2. \mathbf{p} is the vector of loads under uncertainty, that is P_i in this problem. Local search which is triggered every 10 generations in this work is only applied to selected C_N solutions. A constraint on the vertical displacement, g_d , is used to prevent big deformations which are supposed likely to occur. A constraint on the maximum stress in the structure (to prevent fatigue or failure) is important. A dimensionless expression for the stress constraint may be written as

$$g_{stress} = \frac{\sigma_{peak-von-Mises} - \sigma_y}{\sigma_y} \leq 0 \tag{10}$$

where $\sigma_{peak-von-Mises}$ is the peak von Mises stress and σ_y is the tensile yield strength of the material.

The optimization procedure and finite element analysis have been implemented through a C++ program running in the Windows environment of a PC. Values of the objective and constraint functions for every design are derived from the results of a FE analysis of the designed structure.

The optimization was run for 501 generations (with a population size of 100 per generation), by the end of which 46,415 objective function evaluations have been performed. The values of w_1 and w_2 in Equation 4 are 1 and 100 respectively. Three of the non-dominated solutions at the end of 501 generations are shown in Fig. 14. Fig. 14 (a) shows the solution with best weight objective under the worst load case where P_i is 55.5N. Fig. 14 (c) shows the solution with the best displacement objective under the worst load case where P_i is 55.4N. One solution with median weight and displacement objective is given in Fig. 14 (b).

Fig. 15 shows a plot of the best weight objective (w) and the corresponding solution's displacement objective (d) versus generation number. w and d values on the plot corresponding to any particular generation number belong to that generation's non-dominated feasible solution which has the best weight objective.

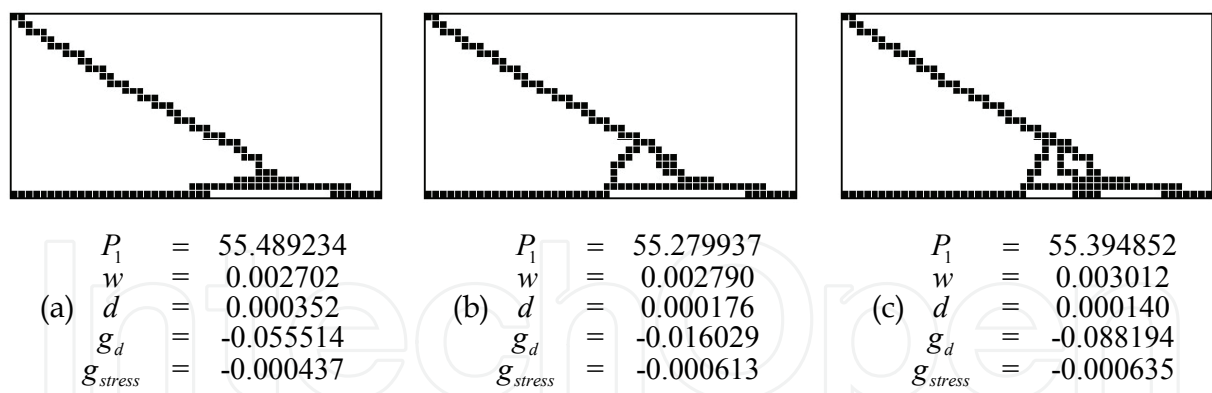


Fig. 14. Three non-dominated solutions at 501st generation.

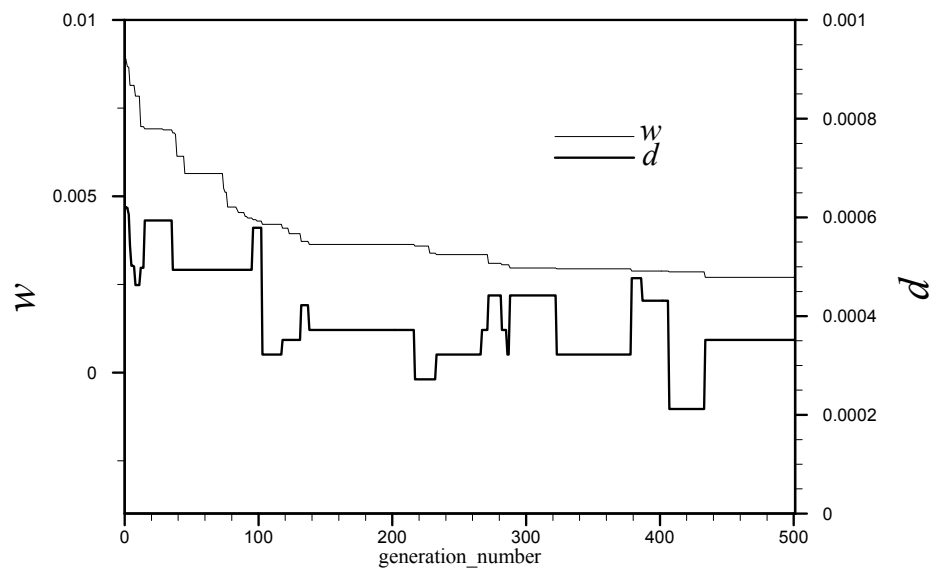


Fig. 15. History of the best weight objective (w).

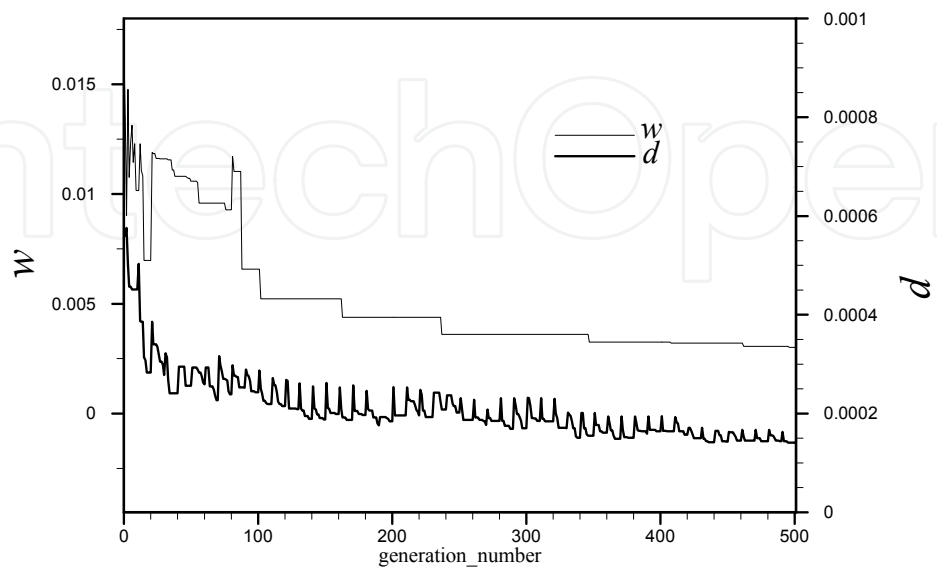


Fig. 16. History of the best displacement objective(d)

Fig. 16 shows plots of the best displacement objective (d) and the corresponding solution's weight objective (w) versus generation number. As can be seen from Fig. 15 and Fig. 16, there are some fluctuations because of anti-optimization.

Fig. 17 shows a plot in the objective space, where the solid shape markers denote the feasible non-dominated solutions at any particular sample generation.

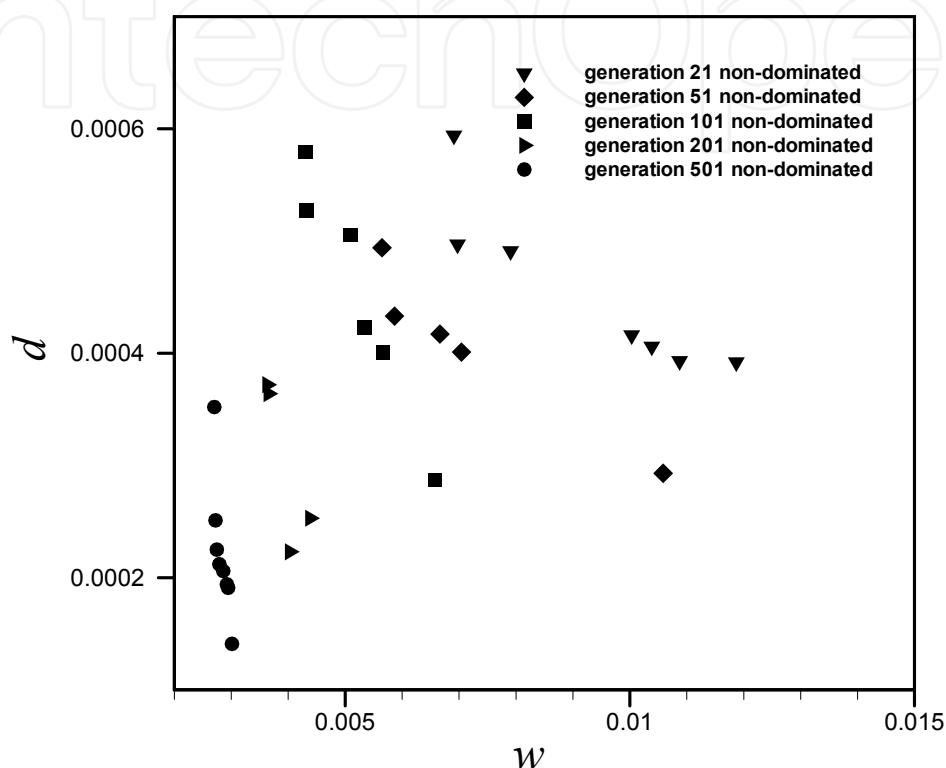


Fig. 17. Plot of non-dominated solutions and elites at some sample generations.

6. Conclusion

The versatility and effectiveness of the topology optimization methodology developed in this work rest on three key components: an efficient morphological geometry representation that defines practical and valid structural geometries, a compatible graph-theoretic chromosome encoding and reproduction system that embodies topological and shape characteristics, and a multiobjective hybrid GA with local search strategy as the worst-case-scenario technique of anti-optimization. The use of local search strategy helps to direct and focus the genetic search in uncertainty design variable space. A multiobjective target matching problem with known solutions has been formulated and solved to demonstrate the validity of presented algorithm. Simulation results of the structure optimization under load uncertainty are encouraging, which indicates that the hybrid algorithm integrates local search as anti-optimization is applicable. The proposed tournament constrained selection method works well and the computation cost is reasonable.

8. References

- Agoston, M. K. (2005). *Computer Graphics and Geometric Modeling*, Springer-Verlag New York, LLC.
- Au, S. K. (2005). "Reliability-based design sensitivity by efficient simulation." *Computers & Structures* 83(14): 1048-61.
- Ayyub, B. M. (1997). *Uncertainty Modeling and Analysis in Civil Engineering*, John Wiley.
- Ben-Haim, Y. and I. Elishakoff (1990). *Convex models of uncertainty in applied mechanics*, Dordrecht: Elsevier Science Publishers.
- Deb, K. (2000). "An efficient constraint handling method for genetic algorithms." *Computer Methods in Applied Mechanics and Engineering* 186(2-4): 311-338.
- Deb, K. and T. Goel (2001). *A hybrid multi-objective evolutionary approach to engineering shape design*, Berlin, Germany, Springer-Verlag.
- Elishakoff, I. (1995). "Essay on uncertainties in elastic and viscoelastic structures: from A. M. Freudenthal's criticisms to modern convex modeling." *Computers and Structures* 56(6): 871-871.
- Elishakoff, I. (1995). "An idea on the uncertainty triangle." *Editors Rattle Space, The Shock and Vibration Digest* 22(10): 1.
- Han, J., V. Manousiouthakis, et al. (1997). "Global optimization of chemical processes using the interval analysis." *Korean Journal of Chemical Engineering* 14(4): 270-276.
- Ishibuchi, H. and T. Murata (1998). "A multi-objective genetic local search algorithm and its application to flowshop scheduling." *Systems, Man and Cybernetics, Part C, IEEE Transactions on* 28(3): 392-403.
- Ishibuchi, H., T. Yoshida, et al. (2002). Balance Between Genetic Search And Local Search In Hybrid Evolutionary Multi-criterion Optimization Algorithms. *Proceedings of the Genetic and Evolutionary Computation Conference*, New York, July 9-13, 2002
- Jaszkiewicz, A. (2003). "Do multiple-objective metaheuristics deliver on their promises --A computational experiment on the set-covering problem." *IEEE Transactions on Evolutionary Computation* 7(2): 133-43.
- Ju, H., H. Vasilios, et al. (1997). "Global optimization of chemical processes using the interval analysis." *Korean Journal of Chemical Engineering* 14(4): 270-276.
- Maenghyo, C. and R. Seung Yun (2004). "Optimization of laminates with free edges under uncertainty subject to extension, bending and twisting." *International Journal of Solids and Structures* 41(1): 227-45.
- Martin, E. T., R. A. Hassan, et al. (2004). "Comparing the N-branch genetic algorithm and the multi-objective genetic algorithm." *AIAA Journal* 42(7): 1495-500.
- Michalewicz, Z., K. Deb, et al. (2000). "Test-case generator for nonlinear continuous parameter optimization techniques." *IEEE Transactions on Evolutionary Computation* 4(3): 197-215.
- Michalewicz, Z. and M. Schoenauer (1996). "Evolutionary algorithms for constrained parameter optimization problems." *Evolutionary Computation* 4(1): 1.
- Moore, R. E. (1966). *Interval analysis*, Prentice-Hall, Englewood Cliffs, NJ.

- Qiu, Y. and S. S. Rao (2005). "A fuzzy approach for the analysis of unbalanced nonlinear rotor systems." *Journal of Sound and Vibration* 284(1-2): 299-323.
- Qiu, Z. and I. Elishakoff (2001). "Anti-optimization technique - A generalization of interval analysis for nonprobabilistic treatment of uncertainty." *Chaos, Solitons and Fractals* 12(9): 1747-1759.
- Ray, T., K. Tai, et al. (2001). "Multiobjective design optimization by an evolutionary algorithm." *Engineering Optimization* 33(4): 399-424.
- Schmidt, M. and Z. Michalewicz (2000). Test-case generator TCG-2 for nonlinear parameter optimisation. *Proceedings of the 2000 Congress on Evolutionary Computation*, Vols 1 and 2: 728-735.
- Srinivas, N. and K. Deb (1994). "Multi-objective function optimization using nondominated sorting genetic algorithms." *Evolutionary Computation* 2(3): 221-- 248.
- Tai, K. and S. Akhtar (2005). "Structural topology optimization using a genetic algorithm with a morphological geometric representation scheme." *Structural and Multidisciplinary Optimization* 30(2): 113-27.
- Tai, K. and J. Prasad (2007). "Target-matching test problem for multiobjective topology optimization using genetic algorithms." *Structural and Multidisciplinary Optimization* 34(4): 333-45.
- Tai, K. and N. Wang (2007). An enhanced chromosome encoding and morphological representation of geometry for structural topology optimization using GA, 2007 *IEEE Congress on Evolutionary Computation*, Singapore, 25-28 September 2007.
- Tai, K., N. F. Wang, et al. (2008). Target geometry matching problem with conflicting objectives for multiobjective topology design optimization using GA, 2008 *IEEE Congress on Evolutionary Computation*, Hong Kong, China, 1-6 June 2008.
- Venter, G. and R. T. Haftka (1996). Two species genetic algorithm for designing composite laminates subjected to uncertainty, *Proceedings of the 37th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Salt Lake City, Utah, April, 1996
- Wang, N. and K. Tai (2007). Handling objectives as adaptive constraints for multiobjective structural optimization, 2007 *IEEE Congress on Evolutionary Computation*, Singapore, 25-28 September 2007.
- Wang, N. and K. Tai (2007). A hybrid genetic algorithm for multiobjective structural optimization, 2007 *IEEE Congress on Evolutionary Computation*, Singapore, 25-28 September 2007.
- Wang, N. F. and K. Tai (2008). "Design of grip-and-move manipulators using symmetric path generating compliant mechanisms." *Journal of Mechanical Design* 130(11): 112305 (9 pp.).
- Wang, N. F. and Y. W. Yang (2009). Target Geometry Matching Problem for Hybrid Genetic Algorithm Used to Design Structures Subjected to Uncertainty, 2009 *IEEE Congress on Evolutionary Computation*, Trondheim, Norway, 18-21 May 2009.
- Wang, N. F., Y. W. Yang, et al. (2008). Optimization of structures under load uncertainties based on hybrid genetic algorithm, 2008 *IEEE Congress on Evolutionary Computation*, Hong Kong, China, 1-6 June 2008.

Zadeh, L. A. (1978). "Fuzzy sets as a basis for a theory of possibility." *Fuzzy Sets and Systems* 1(1): 3-28.

IntechOpen

IntechOpen



Evolutionary Computation

Edited by Wellington Pinheiro dos Santos

ISBN 978-953-307-008-7

Hard cover, 572 pages

Publisher InTech

Published online 01, October, 2009

Published in print edition October, 2009

This book presents several recent advances on Evolutionary Computation, specially evolution-based optimization methods and hybrid algorithms for several applications, from optimization and learning to pattern recognition and bioinformatics. This book also presents new algorithms based on several analogies and metafores, where one of them is based on philosophy, specifically on the philosophy of praxis and dialectics. In this book it is also presented interesting applications on bioinformatics, specially the use of particle swarms to discover gene expression patterns in DNA microarrays. Therefore, this book features representative work on the field of evolutionary computation and applied sciences. The intended audience is graduate, undergraduate, researchers, and anyone who wishes to become familiar with the latest research work on this field.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Nianfeng Wang and Yaowen Yang (2009). Optimization of Structures under Load Uncertainties Based on Hybrid Genetic Algorithm, Evolutionary Computation, Wellington Pinheiro dos Santos (Ed.), ISBN: 978-953-307-008-7, InTech, Available from: <http://www.intechopen.com/books/evolutionary-computation/optimization-of-structures-under-load-uncertainties-based-on-hybrid-genetic-algorithm>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen