

## A Preliminary Study on the Multi-Objective Topology Design by Genetic Algorithm and Finite Volume Method

Kittipong BOONLONG and Kuntinee MANEERATANA\*

Department of Mechanical Engineering, Faculty of Engineering, Chulalongkorn University,  
Phaya Thai Road, Pathum Wan, Bangkok 10330 Thailand  
Tel: 0-2218-6610, Fax 0-2252-2889, E-mail: kittipong\_toy@hotmail.com, kuntinee.m@chula.ac.th

### Abstract

This paper presents a preliminary investigation on the multi-objective topology design by the genetic algorithm (GA) and finite volume (FV) method. The continuum design divides the available domain into uniform regular parallelepiped grids. By selective filling these grids or leaving the spaces empty, different structural configurations are obtained. Performances of structures are calculated by FV simulations and 4 different multi-objective GA procedures – MOGA, NPGA, NSGA-I and NSGA-II – searches for optimum block arrangements. A 2D heat conduction problem with the domain composed of a structured  $5 \times 4$  grid is used as the test case. The design objective is to obtain lightweight configurations that can dissipate lots of heat into the surroundings. Results of the test run, with prescribed numbers of simulated ad hoc configurations of less than 2% of the total combinatorics, are satisfactory. On average, all 4 GA procedures obtain more than 90% of the true optimum solutions with the 100% performances from NSGA-II.

**Keywords:** Topology design, multi-objective optimisation, genetic algorithm, finite volume method.

### 1. Introduction

The genetic algorithm (GA) and computational mechanics have been combined for structural topology optimisation. The topology design allows the creation of new boundaries and may be divided into 3 main categories – the discrete truss design, unit cell properties, e.g. orientation and porosity in composite materials and the continuum structural design.

In the continuum topology design [1], the space that contains the structure is specified. As this area can be divided into small basic geometrical shapes, structures may be assembled by block buildings. Given design parameters and restrictions, patterns of block arrangements that can superlatively perform specified objective functions can be found. This procedure has the advantages of freely evolved topology and the complete absence of human bias as no a-priori knowledge about the topology is needed.

Hence, performances of structures must be evaluated with minimum human involvement and the numerical modelling fulfils these requirements very well. As most previous studies involves

stress analysis problems, the finite element (FE) method has been employed in virtually all cases [1], [2], [3], [4].

Of these works, most involve single objective problems [1], [4], [5]. Others [1], [2] use the fitness sharing method which, in effect, simply combines normalised objective values into a single variable. This method derates fitness of similar members and limits growth of particular species within the population [6].

This research is based on a previous single-objective study, involving a heat transfer application with the FV modelling [7]. The combined approach is found to handle the problem well. It is clear that a derivative-based optimisation is unsuitable for this application as the objective functions are highly discontinuous. In addition, the FV models do not require the finite element connectivity analysis [1] due to its superlative physical representation as control volumes with only one shared corner vertex are not physically attached.

In this paper, the multi-objective topology design is introduced. Four Pareto optimisations with incorporated GA are described. The slightly enlarged test case is modified such that the load is non-uniform, yielding more complex solutions.

### 2. Genetic Algorithm (GA)

The genetic algorithm (GA), inspired by concepts of natural selection and evolutionary processes [6], is a derivative-free, population-based optimisation method.

The problem-specific knowledge is translated into the GA framework by the encoding scheme, which transforms points in the solution space into binary bit strings or chromosomes. Each chromosome is associated with fitness values. The GA stores a set of points as a population, representing the gene pool for the solution. A group of randomly generated chromosome forms the first generation, from which successive generations are repeatedly evolved through genetic operators – namely selection, crossover, mutation and elitism – towards populations with better fitness values (Figure 1).

The selection operation encourages fitter parent chromosomes to reproduce offspring for the next generation. For each pair of selected parents, two children are generated through the crossover operator. The mutation operator randomly changes values of binary bits in the chromosomes and, thus, introduces new routing solution randomly and preventing stagnation at any local optima. These selection and crossover processes are

\* Corresponding author

repeated until the prescribed number of the new population is obtained. In addition, the elitism principle keeps a certain number of best chromosomes for the next generation. Thus, subsequent generations are generated until a stopping criterion is fulfilled.

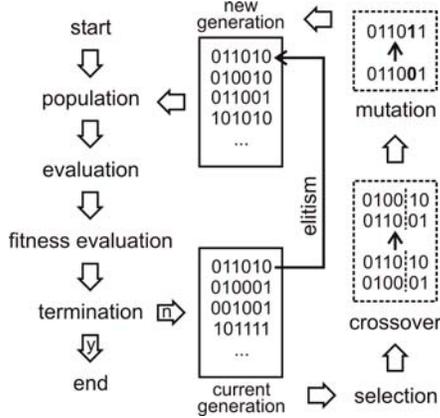


Figure 1 Overview of genetic algorithm evolution

### 3. Evolutionary Multi-Objective Optimisation (EMOO)

In this section, the multi-objective optimisation and GA are combined together. The multi-objective optimisation by Pareto ranking is first outlined, then 4 EMOOs are described.

Multi-objective optimisation searches for decision variables, which simultaneously satisfies constraints and optimised objective functions. Mathematically, for a problem with  $n$  decision variables and  $k$  objective functions, vectors of decision variables  $x_i, i = 1, \dots, n$  are members of the solution space set  $F$ . The optimised decision vector  $x_i^*$  must satisfy the  $m$  inequality constraint  $g_j(x_i) \geq 0, j = 1, \dots, m$  as well as  $p$  equality constraint  $h_j(x_i) = 0, j = 1, \dots, p$ , while simultaneously optimises the objective functions  $f_j(x_i), j = 1, \dots, k$ .

As there is rarely a case that a single point in solution space simultaneously optimises all objective functions, trade-off solutions are instead sought after. For example, if objective functions are to be minimised, a vector of decision variables  $x_i^* \in F$  is *Pareto optimal* if  $f_j(x_i^*) \leq f_j(x_i)$  for all  $j$  and  $f_j(x_i^*) < f_j(x_i)$  in at least one  $j$ .

This concept gives a set of solutions, called the *Pareto optimal set*. The  $x_i^*$ , corresponding to the solutions in the Pareto optimal set, are named *non-dominated vectors*. The plot of the objective functions of the non-dominated vectors is called the *Pareto front*.

Multi-objective optimisation uses the Pareto ranking process to assign the rank of every members of the current population. All non-dominated individuals are assigned rank 1 while ranks of others are equal to one plus the number of its dominating chromosomes. For instance, Figure 2 shows an example of the Pareto ranking in the minimisation of 2 objective functions.

In evolutionary optimisation, the EMOO differs from the single objective genetic algorithm (SGA) mainly in fitness assignment and selection. The SGA straightforwardly employs the values of the single objective function as fitness while an EMOO assigns fitness and selects parent chromosomes through ranking procedures.

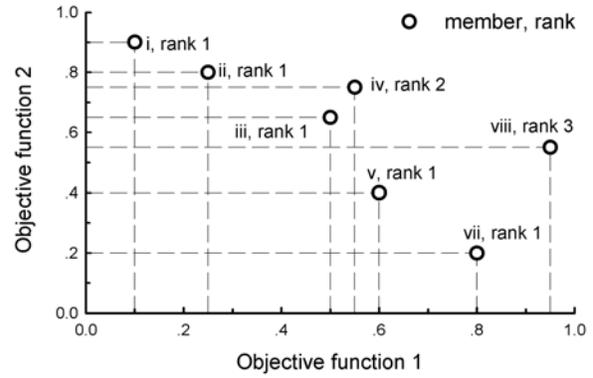


Figure 2 Example of Pareto ranking for objective minimisation

### 3.1 Multiple Objective Genetic Algorithm (MOGA)

MOGA, proposed by Fonseca and Fleming [9], differs from SGA only in fitness assignment. MOGA first sorts all  $N$  members of the current generation into rank 1 to  $r \leq N$  (Figure 2). The Pareto ranking assignment process [6] calculates the dummy fitness values of each chromosome by linear interpolation from rank 1 to  $r$  and assigns block fitness to members with the same rank, equal to their average dummy values, to give the all members in the rank the same chance of selection (Figure 3).

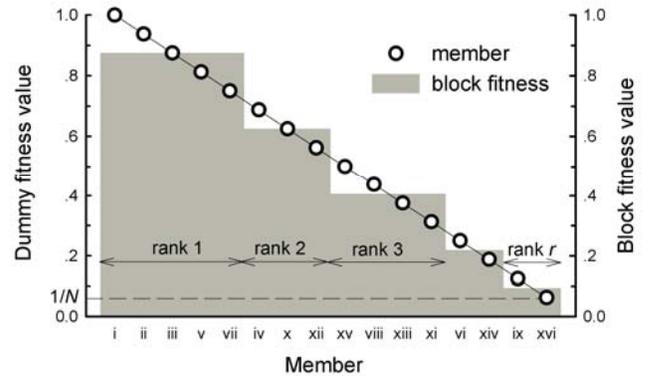


Figure 3 MOGA fitness assignment of members in Figure 2

Since the use of this block fitness may cause a large selection pressure, leading to premature convergence [6], a niche-formation method for population distribution over the Pareto optimal region is used to provide the final fitness values for selection process [10].

MOGA performs sharing in the objective function domain [9] by first calculating the corresponding niche size or sharing factor,  $\sigma$ , which establishes the distance between two individuals that influences the fitness of each other. Using the minimum and maximum values of each objective function,  $m_j$  and  $M_j, j = 1, \dots, k$ , of the current generation,  $\sigma$  can be obtained from:

$$N\sigma^{k-1} - \frac{\prod_{j=1}^k (M_j - m_j + \sigma) - \prod_{j=1}^k (M_j - m_j)}{\sigma} = 0. \quad (1)$$

The population cluster density around a chromosome is represented by its niche count. When only 2 objective functions are considered, the niche of a candidate member may be

visualised as a circle with radius  $\sigma$ , centred on the member. The niche count of a member is computed from:

$$\text{niche count} = 1 + \sum_{q=1}^Q \{1 - (d_q / \sigma)^2\}, \quad (2)$$

where  $Q$  is the number of comparison set member that lies within the circle and  $d_q$  is the distance from the chromosome to uncircled members (Figure 4).

This sharing of individual dummy fitness values help maintaining the diversity of population and discourage the concentration of members in particular regions in the objective domain.

The final fitness value of a member is equal to the block fitness divided by its niche count. In this study, the roulette wheel selection is employed with mating restrictions for crossover [6].

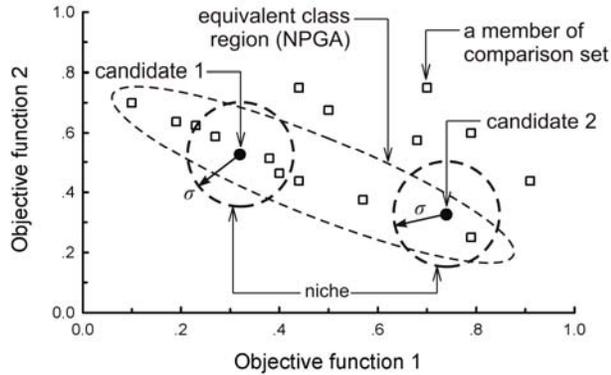


Figure 4 Niche formation method with objective minimisation

### 3.2 Niched Pareto Genetic Algorithm (NPGA)

In NPGA, proposed by Horn, Nafpliotis and Goldberg [11], an updated population with  $N$  members are created from the current population before the chromosome reproduction.

First, all members with rank 1 are copied to the updated population according to the elitism principle. The rest of the updated population are generated by the tournament selection scheme based on Pareto dominance.

In the tournament selection, chromosomes are chosen by comparison with others. Instead of using all population as the comparison set as in MOGA, limited set with  $M$  members, typically around 10% of the population size, is involved in the competition. The  $M$  candidates for selection and the equal number of comparison set members are picked at randomly from the population. Each candidate is then ranked (Figure 2) and compared against the comparison set.

Only one of the candidates is chosen for the updated population. If only one candidate is non-dominated, it is automatically chosen. Otherwise, if some of the candidates are non-dominated or all are dominated, they are equally preferable from a Pareto point of view and it is likely that they belong to the same equivalent class (Figure 4). These equivalent classes can be labelled *equally* fit, any candidate may be chosen and no form of fitness degradation is implemented.

In the interest of maintaining diversity of the gene pool, however, niches are formed and the winning candidate is chosen for its lower niche count value by the equivalent class sharing. In

Figure 4, for instance, candidate 2 is chosen rather than candidate 1.

When the new population is produced, a simple random selection chooses parent chromosomes for reproduction as the preference for fitter members is already incorporated in the form of duplicated copies.

### 3.3 Non-Dominated Sorting Genetic Algorithm I (NSGA-I)

NSGA-I, proposed by Srinivas and Deb [13], is based on the *front*, which is obtained from several layers of individual sorting. Before selection, the population is sorted on the basis of Pareto ranking domination (Figure 2). All non-dominated chromosomes, or those with rank 1, are assigned front 1. Then this group of classified individuals is removed from the population and another layer of non-dominated chromosomes from the remainder of the population are obtained and assigned the next front. This procedure is repeated until all members are sorted (Figure 5).

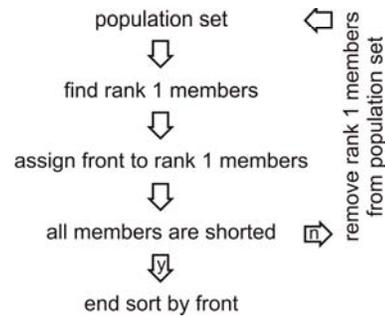


Figure 5 Member classification by front

The dummy fitness values, proportional to the front, are then assigned to every member by linear interpolation from front 1 to the last front  $f \leq N$  as shown in Figure 6. This interpolation is quite similar to the rank interpolation in Figure 3, but with front instead of rank and the linear relationship of dummy fitness between front, rather than members.

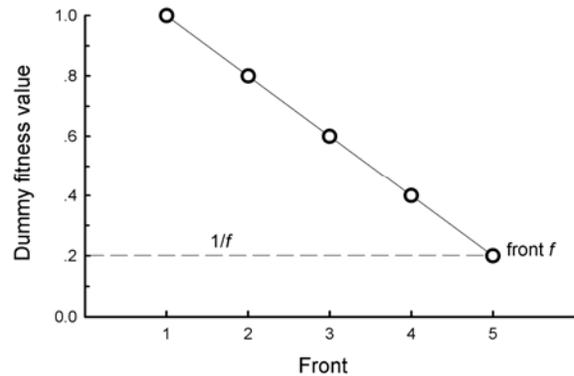


Figure 6 NSGA-I dummy fitness assignment

To maintain the diversity of population and help distributing population over the objective domain, the dummy fitness values members of the same fronts are shared. The final fitness values, used to choose parent chromosomes in the roulette wheel selection, are calculated with the same sharing factor and niche size as in MOGA [9].

### 3.4 Non-Dominated Sorting Genetic Algorithm II (NSGA-II)

NSGA-II, also by Deb [14], proposed a new population sorting procedure. The sharing is replaced with a crowd comparison method, which sort the population according to the crowding distances of objective values.

In a generation  $t$ , the current parent population  $P_t$  with  $N$  members produces the equal number of offspring  $Q_t$  by the simple random selection. The parent  $P_t$  and children  $Q_t$  are then combined together to form the merged population  $R_t$ , half of which are chosen as the parent population of the next generation  $P_{t+1}$  as shown in Figure 7.

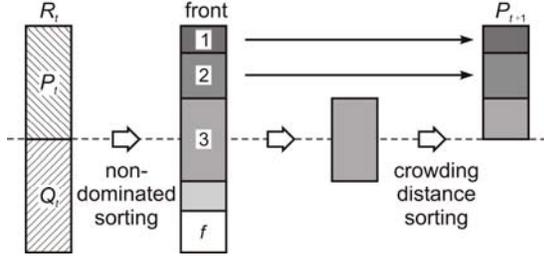


Figure 7 NSGA-II crowd comparison procedure

The choosing procedure starts with the classification of members of  $R_t$  into fronts from 1 to  $f \leq N$ . Chromosomes of better fronts are added en block into the new population  $P_{t+1}$  until the number of selected members is no less than  $N$ . If the population number is greater than  $N$ , members of the last front are lined up in descending order of their crowding distances and those with lower values are removed from the population until the population number is equal to  $N$ .

In the calculation of the crowding distance, the objective values are first normalised and members located at both end spectrums of the normalised values, shown as grey-filled circle in Figure 8, are automatically chosen in order to maintain the diversity of the gene pool.

For the rest, a crowding distance of a member  $n$  for an objective function  $j$ ,  $d_i^n$ , is equal to the difference of normalised objective values of its adjacent members. The summation of  $d_i^n$  from all objective functions  $j = 1, \dots, k$  yields the overall crowding distance for  $n$ . A high value of crowding distance indicates that the chromosome locates apart from others in the objective domain and is prized for its gene diversity.

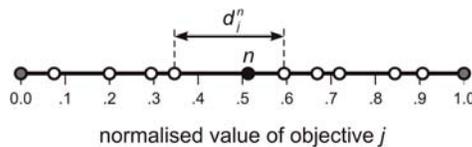


Figure 8 Crowding distance of member  $n$  (filled circle) for objective  $j$

### 4. Objective Evaluation by the Finite Volume Method

The heat dissipating capability of configurations are evaluated by the FV method. The energy conservation with the Fourier's law is employed as the governing equation. For a body in thermal equilibrium without internal heat sources, the model is:

$$\int_s q_i dS_i = \int_s -k \frac{\partial T}{\partial X_i} dS_i = 0, \quad (3)$$

where the normal, outwards vector  $S_i$  presents the surface that bounds the body and  $q_i$  is the heat flux across the surface,  $k$  is the thermal conductivity,  $T$  is the temperature and  $X_i$  is the position vector.

The employed boundary conditions are prescribed temperature and surface convection. The Newton's law of cooling for heat flux  $q^b$  leaving the fluid into solid per unit area is:

$$q^b = h(T^a - T^b), \quad (4)$$

where  $h$  is the heat transfer coefficient,  $T^b$  is the temperature at the solid boundary and  $T^a$  is the ambient fluid temperature.

The mathematical model is discretised by a cell-centred FV technique for arbitrarily shaped control volumes [12]. The spatial domain is divided into a finite number of control volumes or cells. Figure 9 shows a typical cell, bounded by faces  $m$ . The computational node  $P$  locates at the cell centroid. In addition, there are non-computational boundary nodes that are introduced for the specification of boundary conditions.

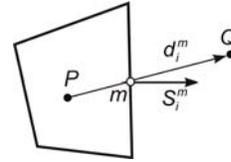


Figure 9 A typical control volume

The 2nd-order accurate spatial distribution is assumed. Gradients of a quantity  $\phi$  at node  $P$  are obtained by ensuring a least square fit of  $\phi$  through  $P$  and its adjacent nodes. Thus, the diffusion flux through the cell face  $m$  into a neighbouring node  $Q$  is approximated using the orthogonal correction method as:

$$\int \frac{\partial T}{\partial X_i} dS_i^m \approx \left( \frac{T^Q - T^P}{d^m} + \left( \frac{\partial T}{\partial X_i} \right)^m \left( \frac{S_i^m}{S^m} - \frac{d_i^m}{d^m} \right) \right) S^m \quad (5)$$

For the convection boundary, the boundary temperature  $T^b$  is calculated by substituting equations (3) and (4) into (5):

$$T^b \left( h + \frac{k}{d^m} \right) \approx k \left( \frac{T^P}{d^m} - \left( \frac{\partial T}{\partial X_i} \right)^m \left( \frac{S_i^m}{S^m} - \frac{d_i^m}{d^m} \right) \right) + h T^a. \quad (6)$$

Then, the temperature at a boundary cell face is incorporated into the cell equation by substituting  $T^Q$  by  $T^b$  in (5).

The equation for each control volume may be rearranged as  $a^P T^P - \sum a^Q T^Q = b$ , where  $a^P$  and  $a^Q$  are respectively the coefficients of the cell and its neighbours and  $b$  is the source terms. By assembling equations of all cells, a system of algebraic equations is obtained with nodal temperature as unknown. The resulting system is linearised, segregated and iteratively 'solved' by the incomplete Cholesky conjugate gradient (ICCG) solver until a certain level of convergence is reached. The updated results are then used to adjust the non-linear terms; and the new system is 'solved' again. This procedure is repeated until implicit solutions are obtained.

### 5. Test Case

A 2D heat transfer problem with convective boundary is used as the case study. Given a wall with distributed temperature profile, a limited space is available for attaching a lightweight solid protruding configuration that allows high heat loss from the wall. The dissipated heat is calculated from the combined heat

flux from the wall into the protruding structures and the surroundings while the number of blocks is used to indicate the weight of the configuration.

In this paper, the dimension of available space with unit thickness is 50 mm x 40 mm. This structure-containing space is divided into 5 x 4 mm rectangles of equal size (Figure 10). Other parameters are prescribed as follows: the temperature of the wall varies linearly from 0 °C to 100 °C, the ambient temperature  $T^b = 0$  °C, the thermal conductivity of the protruding solid  $k = 50$  W/mK and the heat transfer coefficient from the wall and the protruding body into the surrounding  $h = 25$  W/m<sup>2</sup>K. It is also assumed that the air circulation is so good that the ambient air temperature in close proximity to the wall and the protruding body remains unchanged.

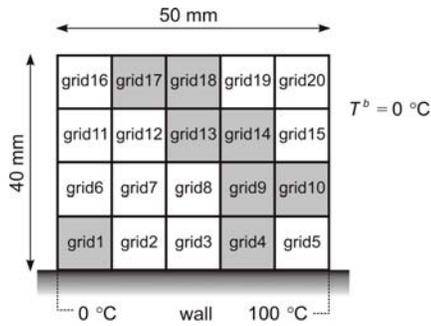


Figure 10 Problem descriptions and block placements

Configurations of solid block in the available space are encoded into 20-bit binary chromosomes in the order illustrated in Figure 10. The block insertion on a grid, shown as a shaded area, is represented by '1' whilst '0' signifies a void in the corresponding location. For instance, the configuration in Figure 10 may be encoded into '10010 00011 00110 01100'.

It is noted that in this preliminary work, a solid-filled grid is modelled with only one control volume and no grid independency is considered.

## 6. Results

Altogether, there are  $2^{20}$  or about 1.05 million possible configurations, much more than 4096 patterns from 5 x 4 grid in the previous work [7]. For comparison, the performances of all configurations are first calculated by FV to provide the true Pareto front and optimal sets as shown in Figure 11 and Figure 12, respectively.

The highest heat loss values of configurations with 14 blocks and over are less than that of the 13-block structure, hence the non-dominated vectors contain solutions with 13 blocks or less as shown in the Pareto front (Figure 11).

Figure 12 shows that all 19 optimum configurations have 2 common characteristics, the wall-structure contact position and surface exposure. The truly influential wall-structure contact position is the right hand side as the prescribed temperature is highest. Due to the good conductivity, temperature distribution in the structure is fairly high and any additional contact of the structure and the wall induces the unwelcome heat transfer into the wall. For the second feature, the structure tries to expand outwards to maximise the exposure area.

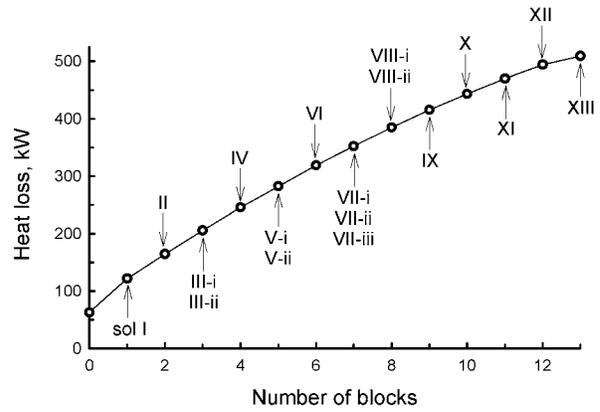


Figure 11 True Pareto front of the test case

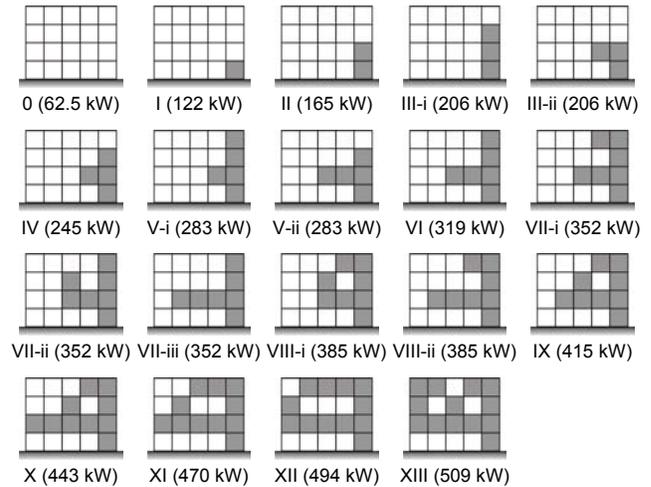


Figure 12 Optimum solutions of the test case

This study uses a simple unoptimised GA algorithm. Its control parameters are simply chosen as follows: maximum number of generation = 200, the population size in each generation = 100, 1-point crossover at the rate of 1.0, bit flip mutation with 0.01 mutation rate and rank 1 chromosomes are automatically elite. Hence, in each run, around 1.9% of the combinatorics, albeit with many repeated appearances, are simulated.

For comparison, the design modellings for each EMOO are repeated for 20 times. The numbers of true Pareto optimum solutions that are found by the GA in each run as well as the number of generations required for the algorithm to find all true optimum solutions in the run are shown in Table 1 while the average values are shown in Table 2.

It is found that NSGA-II gives the best performances as it finds all true optimum solutions in every modelling, followed by MOGA at 50% of the runs. The NPGA and NSGA-I yield not too different results since they can search for all complete set of true optimum solutions in less than 25% of all running times.

In addition, NSGA-II, with 100% performance for true optimum solutions, on average finds them in 124.4 generation, each with 100 members or about 1.2% of the total possible configurations. Even though MOGA can search for all solutions in only half the times, when it does, it is a little faster at 117.3 generations on average. In comparison, the number of complete

solution runs for NPGA and NSGA-I are quite small and are either very quick or near the end of the run.

Table 1 Comparisons of various EMOO solutions

Run	Generation no. that all true optimum solutions are found				No. of true optimum solutions found in each run			
	MOGA	NPGA	NSGA-I	NSGA-II	MOGA	NPGA	NSGA-I	NSGA-II
1	95	67	N/A	189	19	19	16	19
2	N/A	N/A	N/A	144	17	17	18	19
3	97	N/A	161	84	19	18	19	19
4	N/A	N/A	N/A	180	18	18	18	19
5	N/A	N/A	N/A	108	18	17	17	19
6	192	N/A	N/A	67	19	18	16	19
7	N/A	165	N/A	121	18	19	16	19
8	104	N/A	N/A	92	19	18	16	19
9	N/A	N/A	N/A	67	18	18	18	19
10	N/A	78	N/A	80	18	19	17	19
11	99	N/A	N/A	188	19	18	15	19
12	73	N/A	N/A	167	19	18	17	19
13	84	N/A	N/A	194	19	17	15	19
14	N/A	N/A	N/A	76	18	17	18	19
15	N/A	198	N/A	101	18	19	17	19
16	N/A	N/A	N/A	72	18	18	17	19
17	179	N/A	N/A	88	19	17	17	19
18	N/A	N/A	N/A	119	16	18	16	19
19	97	N/A	196	196	19	18	19	19
20	153	N/A	N/A	155	19	18	18	19

Table 2 Overall comparisons of various EMOO solutions

Indicator	MOGA	NPGA	NSGA-I	NSGA-II
Average generation no. that all true optimum solutions are found	N/A	N/A	N/A	124.4
Average no. of run that all true optimum solutions are found	10 (50%)	4 or (20%)	2 (10%)	20 (100%)
Average no. of true optimum solutions found in each run	18.35 (96.6%)	17.95 (94.5%)	17.00 (89.5%)	19.00 (100%)

## 7. Discussions and Conclusions

A preliminary multi-objective topology design procedure using combined GA and FV method is investigated. The concept of the structural design by composition of basic units is outlined. Four different Pareto-based EMOOs, namely MOGA, NPGA, NSGA-I and NSGA-II, are investigated.

EMOO may not be able to find all true optimum solutions under the given GA parameters of 200 generations, each with 100 members. Only NSGA-II finds all true optimum solutions in all runs while MOGA, NPGA and NSGA-I respectively achieves the same in 50%, 20% and 10% of the test runs.

The inability to find all true solutions does not imply that the performance is much more inferior; only one or two solutions may elude the search. On average, MOGA finds 97% of the true optimum solutions; NPGA finds 95% and NSGA-I 90%.

In future development, the problem sizes are much larger, resulting in incredibly large combinatorics and all true optimum solutions may never be known. Hence, it is not expected that all true solutions are obtained. Uncompleted near optimum, but diversified, solutions are generally acceptable and useful for practical proposes.

Consequently, all 4 EMOOs are considered satisfactory, with over 90% of the true optimum solutions are found. Tentatively, NSGA-II gives the best performances, probably due to the

absence of sharing assignment in the procedure. The sharing is very sensitive to niche sizes and is considered a weak point of EMOO [8].

## 8. Acknowledgement

This work is partially supported by TRF for the Senior Research Scholar Prof. Pramote Dechaumphai.

## References

- [1] M.J. Jakiela, C. Chapman, J. Duda, A. Adewuya and K. Saitou, "Continuum structural topology design with genetic algorithms", *Computer Methods in Applied Mechanics and Engineering*, vol. 186, 2000, pp. 339-356.
- [2] K. Deb and T. Goel, "A hybrid multi-objective evolutionary approach to engineering shape design", *Evolutionary Multi-Criterion Optimization, EMO 2001, LNCS 1993*, Springer-Verlag, 2001, pp. 385-399.
- [3] E. Sandgren, E. Jensen and J. Welton, "Topology design of structural components using genetic optimization methods", *Proceedings of the Winter Annual Meeting of ASME*, 1990, pp. 31-43.
- [4] F. Belblidia and E. Hinton, "Fully integrated design optimization of plate structures", *Finite Elements in Analysis and Design*, vol. 38, 2002, pp. 227-244.
- [5] T.-Y. Chen and C.-Y. Lin, "Determination of optimum design spaces for topology optimization", *Finite Elements in Analysis and Design*, vol. 36, 2000, pp. 1-16.
- [6] D.E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley, 1989.
- [7] K. Maneeratana, "A feasibility study on the continuum topology design by genetic algorithm and finite volume method", *Proceedings of ME-NETT 16*. Phuket, 2002, pp. 256-261.
- [8] C.A.C. Coello, "A short tutorial on evolutionary multiobjective optimization", *Proceedings of EMO'01, Zurich, Switzerland*, 2001, pp. 21-40.
- [9] C.M. Fonseca and P.J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization", *Proceedings of the 5th ICGA, Morgan Kaufmann, San Mateo, CA*, 1993, pp. 416-423.
- [10] K. Deb and D. E. Goldberg, "An investigation of niche and species formation in genetic function optimisation", *Proceedings of the 3rd ICGA, Virginia*, 1989, pp. 42-50.
- [11] J. Horn, N. Nafpliotis and D.E. Goldberg, "A niched pareto genetic algorithm for multiobjective optimisation", *Proceedings of the 1st ICEC 1994*, vol. 1, 1994, pp. 82-87.
- [12] K. Maneeratana and A. Ivankovic, "Finite volume method for structural applications involving material and geometrical non-linearities", *Proceedings of ECCM'99, Munich, Germany*, 1999, no. 325.
- [13] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms", *Evolutionary Computation*, vol. 2, no. 3 1994, pp. 221-248.
- [14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, 2002, pp. 182-197.