

Paper

Int'l J. of Aeronautical & Space Sci. 14(4), 341–349 (2013)
DOI:10.5139/IJASS.2013.14.4.341

Particle Swarm Assisted Genetic Algorithm for the Optimal Design of Flexbeam Sections

Manoj Kumar Dhadwal*, **Kyu Baek Lim*** and **Sung Nam Jung****

Department of Aerospace Information Engineering, Konkuk University, Seoul, Korea

Tae Joo Kim***

Rotor Department, Rotorcraft Program Office, Korea Aerospace Research Institute, Daejeon, Korea

Abstract

This paper considers the optimum design of flexbeam cross-sections for a full-scale bearingless helicopter rotor, using an efficient hybrid optimization algorithm based on particle swarm optimization, and an improved genetic algorithm, with an effective constraint handling scheme for constrained nonlinear optimization. The basic operators of the genetic algorithm, of crossover and mutation, are revisited, and a new rank-based multi-parent crossover operator is utilized. The rank-based crossover operator simultaneously enhances both the local, and the global exploration. The benchmark results demonstrate remarkable improvements, in terms of efficiency and robustness, as compared to other state-of-the-art algorithms. The developed algorithm is adopted for two baseline flexbeam section designs, and optimum cross-section configurations are obtained with less function evaluations, and less computation time.

Key words: Beam Section Optimization, Real-Coded Genetic Algorithm, Particle Swarm Optimization, Rank-based Multi-Parent Crossover, Flexbeam

1. Introduction

In recent decades, evolutionary computation has emerged as an efficient and powerful technique to solve real world design optimization problems. Numerous population-based algorithms based on the theory of evolution, such as the genetic algorithm (GA) and particle swarm optimization (PSO), are being routinely used in many engineering applications. Despite their popularity, there are various challenges, such as the premature convergence, and high computational load required to achieve global optimum solutions. The real world problems can be posed as highly nonlinear, convex or non-convex, and smooth or non-smooth, along with a large number of design variables and constraints. These often involve multiple local optima.

The classical derivative-based algorithms utilize the gradient information of a continuous function to search for

the local optima, which has been shown to be efficient. The derivative-free algorithms require function evaluations, instead of computing the gradients. For discontinuous functions, the derivatives may not exist, and hence the derivative-based algorithms may fail to reach the optimum solution. Moreover, these algorithms often depend on the choice of the starting points, and therefore cannot guarantee convergence to a global optimum. Further, the computation of the derivative, either exact or an approximation, adds to the computational cost. Population-based, derivative-free algorithms are considered effective to overcome such difficulties, and lead to the reaching of global optimum solutions.

Large-scale complex constrained problems can be tackled using the metaheuristic algorithms (e.g. PSO and GA), which are suitable to locate the global optimum [1]. The two main features of the metaheuristic algorithms are intensification

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

© * Graduate Research Assistant
** Professor, Corresponding author: snjung@konkuk.ac.kr
*** Senior Engineer

(exploitation), and diversification (exploration) [2]. Diversification is intended to explore the search space globally, whereas intensification is focused on directing the search to a small region of the search space. The convergence rate of the algorithm can be ensured, by maintaining the balance between the two features.

There are various optimization approaches available in the literature for the optimal design of beam sections. Murugan and Ganguli [3] presented the optimal design of composite box beams, using a real-coded genetic algorithm (RCGA). The design optimization was decomposed into two levels; a global level, consisting of cross-sectional dimensions, and the local level, consisting of ply orientations. Suresh et al. [4] used a particle swarm optimization (PSO)-based approach to optimize the box-beam section, and proved it to be superior to the real-coded genetic algorithm. Although the above-mentioned methods can be applied to find global optimum solutions, both methods suffer from drawbacks. The real-coded genetic algorithm, though better than the binary GA, has a slow convergence rate; whereas, the PSO may converge to a local optimum. To avoid these drawbacks, a hybrid of RCGA and PSO can be considered effective, in the design of optimum beam sections.

In the present study, a hybrid of particle swarm and a modified real-coded genetic algorithm, hereby called a particle swarm assisted genetic algorithm (PSGA), is applied to search for optimum solutions of the flexbeam cross-

section of a full-scale bearingless helicopter. The efficiency and efficacy of the proposed algorithm are investigated on several constrained engineering optimization problems. Numerical simulations are conducted, to demonstrate the promising application of the PSGA algorithm to different types of flexbeam sections.

2. Proposed Approach

The flowchart of the proposed PSGA is shown in Fig. 1. In the particle swarm assisted genetic algorithm, the population is randomly initialized over the search space, with a uniform distribution. The population is then moved through two sequential phases, to find the best feasible solution. The first phase involves the enhancement of the population with worst fitness, using the PSO. The global-local best inertia weight is used, to preserve the previous velocity. The objective function and constraint violation of the population are evaluated, and the population is ranked, using a pair-wise comparison. The population is then moved to the genetic algorithm (GA) phase. To improve the convergence and stability of the genetic algorithm, a new rank-based multi-parent crossover (RMPC) operator is introduced, based on the parents' ranks, partially adopted from the differential evolution programming. The operator is based on the mutation operator of differential evolution, and the heuristic crossover operator. In RMPC, the scaling

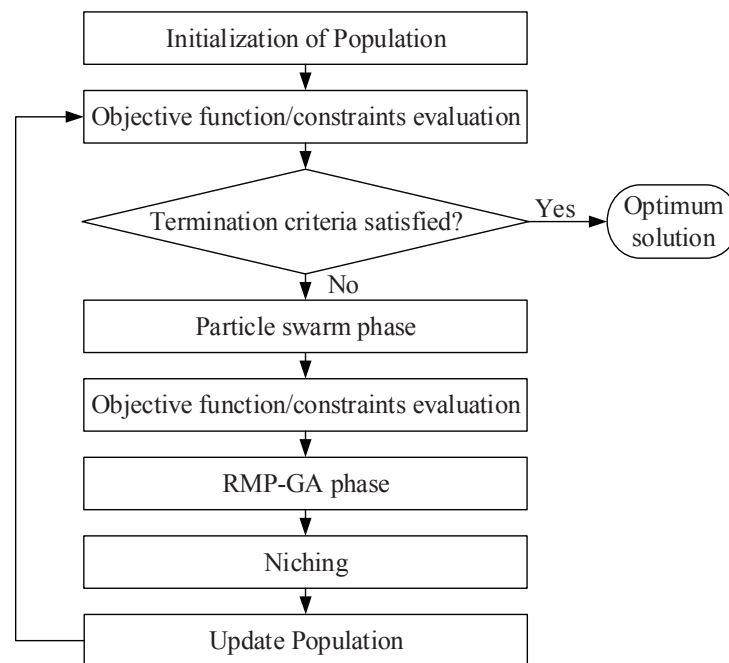


Fig. 1. Flowchart of the PSGA

factor and crossover rate for an individual depend on its rank in the population. First, three parents are selected, using the binary tournament selection scheme. The individuals are then ranked relative to each other. The parents are ranked in the order with the best parent having the lowest rank ($R_1 < R_2 < R_3$). For a parent with i -th index in the population, and rank R_i , the scaling factor S_i and crossover rate cr_i are computed as follows.

$$S_i = S_{min} \left(\frac{R_i - 1}{N - 1} \right)^2 + S_{max} \quad (1)$$

$$cr_i = cr_{min} \left(\frac{R_i - 1}{N - 1} \right) + cr_{max} \quad (2)$$

where, N is the population size. The offsprings ($y_j^{(1)}, y_j^{(2)}, y_j^{(3)}$) are then generated by adding the position of each parent to the difference of positions of the other two parents, multiplied by a scaling factor.

$$y_j^{(1)} = x_j^{(1)} + S_i (x_j^{(2)} - x_j^{(3)}), j = 1, \dots, n \quad (3)$$

$$y_j^{(2)} = x_j^{(2)} + S_i (x_j^{(3)} - x_j^{(1)}) \quad (4)$$

$$y_j^{(3)} = x_j^{(3)} + S_i (x_j^{(1)} - x_j^{(2)}) \quad (5)$$

where, x_j is the n -dimensional position vector of the parent.

It can be deduced from the above equations that the parent with a better rank has a lower value of the scaling factor and higher crossover rate, and *vice versa*. The new offspring thus generated by a parent with a better rank will be closer to the parent, whereas the offspring generated by a parent with a worst rank will be farther from the parent. This improves the efficiency of the search, while maintaining the diversity in the population.

To introduce additional randomness and preserve diversity in the algorithm, the polynomial based mutation operator developed by Deb [5] is used for the mutation in the GA phase. After the GA, the population is again ranked,

according to the values of the objective function, and constraint violation. The iteration loop is continued, until the termination criteria are met. The termination criteria can be the maximum number of function evaluations, or no improvement in the objective function for a successive number of generations.

The constraints are handled using the separation of feasible population. The constraint relaxation scheme proposed by Zhang [6] has been used in the current approach. At the initial generation, the individuals with constraint violation less than the median constraint violation of the population are considered as feasible. As the generations increase, the relaxation of the constraint violation is decreased, pushing the individuals towards the feasible region. The unique feature of the constraint-handling scheme is that the parameters are self-adaptive, and updated automatically with the generations. To maintain the diversity in the population and re-evaluate the same area in the search space, a niching strategy is applied, based on the Euclidean distance between the parent and the offspring. If the offspring lies within the critical distance from the parent, the offspring is rejected.

3. Results

The parameters used in the PSGA are as follows. The total population size is 20, with 10 each for the particle swarm phase, and the RMP-GA phase. The parameters set for the PSO are: $c_1 = c_2 = 2.0$. For rank-based multi-parent crossover operator, the scaling factors are set to $S_{min} = 0.6$, $S_{max} = 0.95$, and the crossover rates are set to $cr_{min} = 0.85$, $cr_{max} = 0.95$. The polynomial based mutation operator is used for the GA mutation. The mutation probability is self-adaptive, which varies with the number of generations. The maximum number of function evaluations is limited to 5,000, whereas the maximum number of generations is set to 250. The optimizations are performed on a computer with

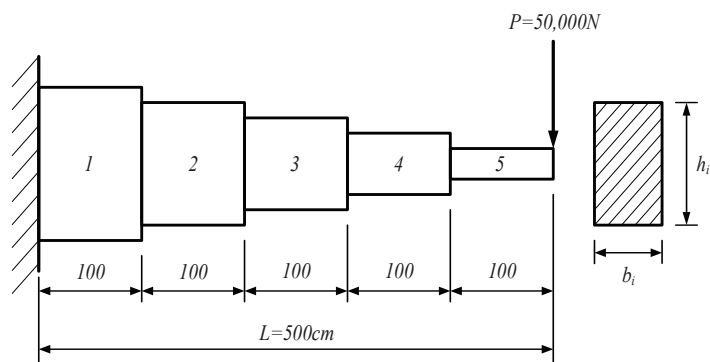


Fig. 2. Stepped cantilever beam

3.40 GHz Intel Core i7 CPU and 8 GB RAM. The algorithm is implemented in Fortran 90.

3.1 Numerical Example: Stepped Cantilever Beam

To evaluate the performance of the algorithm, a mixed discrete stepped cantilever beam design problem has been considered. The objective is to minimize the volume of the stepped cantilever beam under the vertical tip force, shown in Fig. 2. The height and width of the rectangular cross section of each step form the design variables. The problem consists of 10 design variables, and 11 constraints. The width b_1 and height h_1 of the first step are integer values; the widths b_2 , b_3 of the second and third steps are discrete values, chosen from a set of {2.4, 2.6, 2.8, 3.1}; the heights h_2 , h_3 of the second and third steps are discrete values, chosen from a set of {45.0, 50.0, 55.0, 60.0}; and the rest of the design variables are continuous. Thirty independent runs were carried out, to test the robustness of the algorithm.

To solve this problem, Erbatur et al. [7] used a program called genetic algorithm based optimum structural design (GAOS), and Lemonge and Barbosa [8] used the genetic algorithm with an adaptive penalty method (APM). Bernardino et al. [9] used an artificial immune system with

genetic algorithm. Later, Bernardino et al. [10] used the improved version of AIS-GA, and compared the results with the genetic algorithm with a stochastic ranking (SR). The optimal results are presented in Table 1, and the statistical results are shown in Table 2. The convergence histories of the objective function and constraint violation corresponding to the best solution obtained using PSGA are shown in Fig. 3. The PSGA converges toward the near-optimal solution of 64593.00 in nearly 60 generations, requiring 1,377 function evaluations. The feasible global optimum solution is reached in 163 generations, with 3,746 function evaluations. The solution obtained using GAOS violates one of the constraints. A better objective value is reported by Bernardino et al. [10], obtained using the stochastic ranking (SR) with a genetic algorithm, which was obtained in 35,000 function evaluations. As compared to other approaches, PSGA finds the feasible global optimum in nearly 62.5-89.3% less number of function evaluations. The PSGA thus obtains an accurate solution, efficiently and consistently outperforming other algorithms.

3.2 Flexbeam Section Optimization

The cross sectional analysis program KSec2D [11] has been

Table 1. Optimal results for the stepped cantilever beam design

	GAOS [7]	APM [8]	AIS-GA [9]	AIS-GA [10]	SR [10]	Present PSGA
b_1	3	3	3	3	3	3
b_2	3.1	3.1	3.1	3.1	3.1	3.1
b_3	2.6	2.6	2.6	2.6	2.6	2.6
b_4	2.270	2.2894	2.3107138	2.2947	2.2837	2.280881
b_5	1.750	1.7931	2.2254148	1.8250	1.7532	1.749772
h_1	60	60	60	60	60	60
h_2	55	55	60	55	55	55.0
h_3	50	50	50	50	50	50.0
h_4	45.250	45.6256	43.18571	45.2153	45.5507	45.617623
h_5	35.000	34.5931	31.250282	35.1191	35.0631	34.995353
V	64447	64698.56	66533.47	64834.70	64599.65	64578.22

Table 2. Statistical results for the stepped cantilever beam design

	<i>fevals</i>	Best	Mean	Worst	SD
GAOS [7]	10,000	64447	NA	NA	NA
APM [8]	35,000	64698.56	68107.05	73931.359	NA
AIS-GA [9]	35,000	66533.47	71821.69	76852.86	NA
AIS-GA [10]	35,000	64834.70	76004.24	102981.06	6.93E+03
SR [10]	35,000	64599.65	83968.45	71240.03	3.90E+03
Present PSGA	3,746	64578.22	67123.60	70568.17	2.11E+03

used to compute the section properties of section stiffnesses, section offsets, and mass per unit length. MSC. Patran is used as a pre-processor, to generate the geometry and mesh. The geometry and mesh generation are automated using Patran command language (PCL). The input from Patran is generated in neutral file format, and then translated into KSec2D input format.

The section optimization is demonstrated for two different configurations of the flexbeam section. The goal of the optimization is to minimize the mass per unit length of the section. Only the geometric dimensions are considered as design variables. It is worth mentioning that all the design variables are considered as discrete or integer multiples, thus considering manufacturing constraints. Note that the flexbeam sections are considered as symmetric. The side constraints are imposed on the section stiffnesses, and the upper limit constraint is imposed on the maximum failure index (FI_{max}), obtained from the *von-Mises* failure criterion. The material properties are given in Table 3. The flexbeam sectional loads in the hover flight condition are shown in Table 4. The section optimization problem is formulated as follows:

Minimize mass/length

Subject to:

$$0.95EA_0 \leq EA \leq 1.1EA_0$$

$$0.95EI_{y0} \leq EI_y \leq 1.1EI_{y0}$$

$$0.95EI_{z0} \leq EI_z \leq 1.1EI_{z0}$$

$$0.95GJ_0 \leq GJ \leq 1.1GJ_0$$

$$FI_{max} < 0.8$$

where, EA is the axial stiffness, EI_y and EI_z are the flap and lag bending stiffnesses, respectively, GJ is the torsional stiffness, and FI_{max} is the maximum failure index. The subscript '0' indicates the baseline properties.

The parameterized geometric model of flexbeam section-A is shown in Fig. 4. The baseline configuration of section-A is illustrated in Fig. 5, which shows the unidirectional glass region filled with two separate regions, and a skin layer of glass fabric core. There are eight design variables, which include the distance of core regions from the horizontal and vertical center-lines, radius of the fillet, and skin thickness. Table 5 presents the values of the design variables for the baseline configuration, and the upper and lower limits, along with the minimum precision. The optimal results obtained from the PSGA are presented in Table 6. The best solution converged to the optimum point after 18 generations. The optimized configuration is shown in Fig. 6. It can be observed that there is a 2.33% reduction in mass per unit length, owing to the increase in area of the glass fabric core regions, which have lower density than unidirectional glass. Additionally, the flap bending stiffness approaches the lower limit of the constraint boundary, due

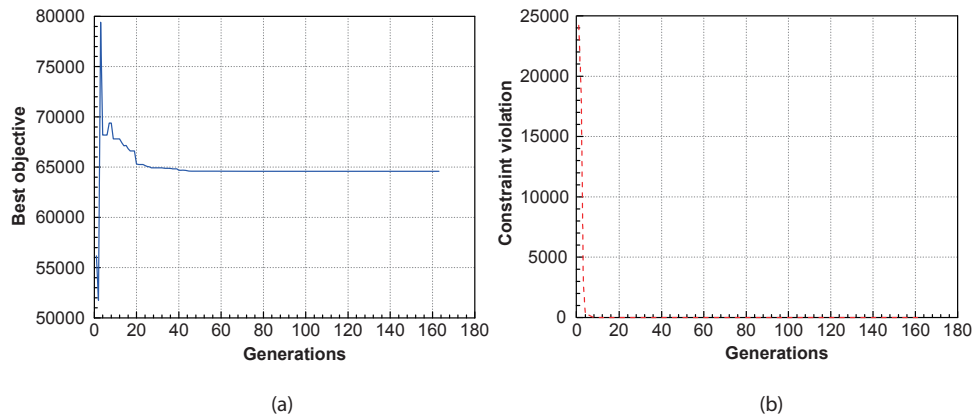


Fig. 3. Convergence of the (a) objective function, and (b) constraint violation of the best solution for the stepped cantilever beam design

Table 3. Material properties

	Glass UD	Glass Fabric
$E_1=E_2$ (GPa)	51.5	23.6
G_{12} (GPa)	4.8	3.28
Density, ρ (kg/m ³)	2014	1730
Ultimate strength (MPa)	1656	452

Table 4. Flexbeam sectional loads in hover flight condition

	Section-A	Section-B
Axial force, F_x (kN)	185.9	188.7
Torsional moment, M_x (N.m)	160.0	233.1
Flap bending moment, M_y (N.m)	663.6	-225.6
Lag bending moment, M_z (N.m)	-1590	-6520

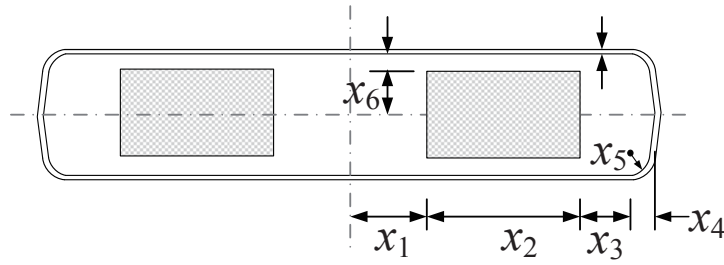


Fig. 4. Flexbeam section-A

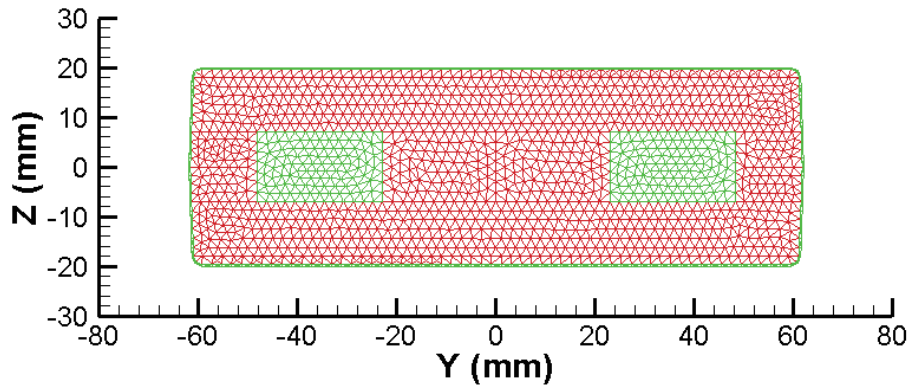


Fig. 5. Baseline configuration of flexbeam section-A

Table 5. Design variables for flexbeam section-A

Design variables	Baseline	Lower bound	Upper bound	Precision
x_1 (mm)	22.96	5.0	30	0.1
x_2 (mm)	25.32	10.0	50.0	0.1
x_3 (mm)	11.1	5.0	20.0	0.1
x_4 (mm)	2.31	2.2	5.0	0.1
x_5 (mm)	1.68	1.5	2.2	0.1
x_6 (mm)	7.0	4.0	15.0	0.1
x_7 (mm)	12.68	4.0	20.0	0.1
x_8 (mm)	0.32	0.32	3.2	0.32

Table 6. Optimal results for flexbeam section-A

Design variables	Baseline	Optimum	Difference (%)
m (kg/mm)	9.69607E-03	9.47010E-03	-2.33
EA (N)	2.31185E+08	2.23610E+08	-3.28
EI_Y (N.mm ²)	3.24367E+10	3.08179E+10	-4.99
EI_Z (N.mm ²)	2.89743E+11	2.77824E+11	-4.11
GJ (N.mm ²)	9.82883E+09	9.49865E+09	-3.36
FI_{max}	0.0804	0.0831	3.36

to the change in the section shape. The maximum failure index is slightly higher than the baseline, but well within the desired limit. The optimal results were obtained after 411 function evaluations, and nearly 5 minutes of CPU time was utilized.

Figure 7 shows the parameterized geometry of

flexbeam section-B, having double-H shape. The inner region consists of unidirectional glass, with an outer skin layer of glass fabric. There are thirteen design variables, including the skin thickness, fillets radii, and the horizontal and vertical distance of the H-shaped regions from the center-lines. The values of the baseline section,

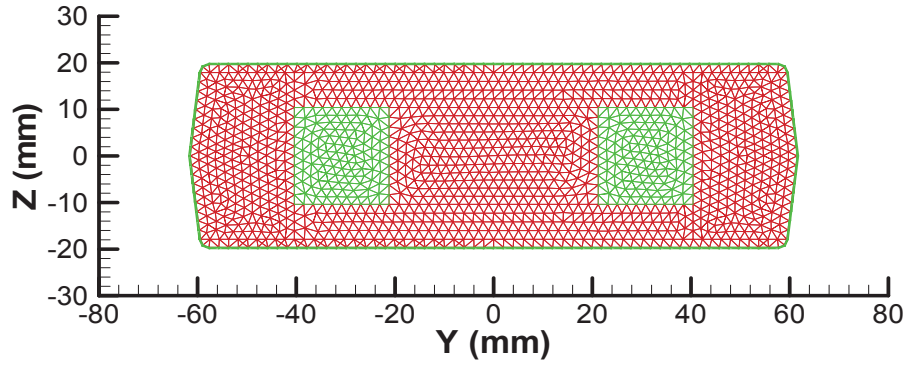


Fig. 6. Optimized configuration of flexbeam section-A

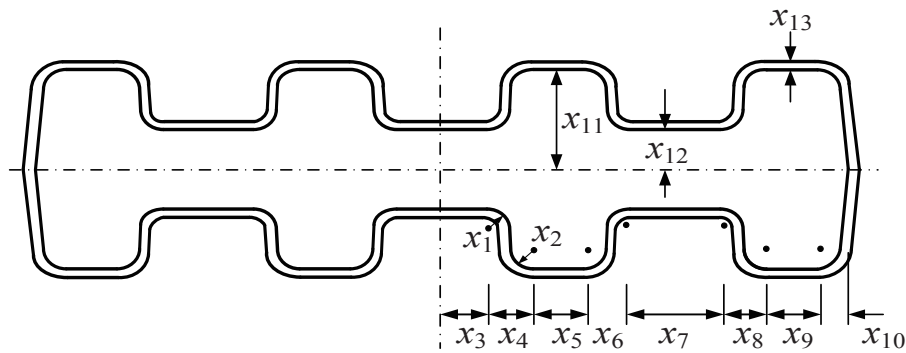


Fig. 7. Flexbeam section-B

Table 7. Design variables for flexbeam section-B

Design variables	Baseline	Lower bound	Upper bound	Precision
x_1 (mm)	2.32	2.0	2.5	0.1
x_2 (mm)	1.68	1.5	2.0	0.1
x_3 (mm)	7.05	5.0	10.0	0.1
x_4 (mm)	4.26	3.0	5.0	0.1
x_5 (mm)	5.85	5.0	20.0	0.1
x_6 (mm)	4.26	3.0	5.0	0.1
x_7 (mm)	12.1	5.0	20.0	0.1
x_8 (mm)	4.26	3.0	5.0	0.1
x_9 (mm)	5.97	5.0	20.0	0.1
x_{10} (mm)	2.14	2.0	5.0	0.1
x_{11} (mm)	14.68	12.0	25.0	0.1
x_{12} (mm)	3.43	3.0	11.0	0.1
x_{13} (mm)	0.32	0.32	1.60	0.32

and the upper and lower bounds of the design variables, along with minimum precision, are shown in Table 7. The baseline configuration of flexbeam section-B is shown in Fig. 8. The optimal results are presented in Table 8. The best solution converged to the optimum point after 29 generations. The optimized section is shown in Fig. 9. The change in the shape of the double-H section can be noted, compared to the baseline. A mass reduction of 2.88% can be observed. Furthermore, the lag bending and torsion stiffnesses approach the lower limit of the constraint boundaries. The increase in the maximum failure index is approximately 4%, however, well below the design limit. The optimization algorithm required 664 function evaluations to reach the optimal solution, and consumed nearly 5 minutes of CPU time.

4. Conclusion

An advanced particle swarm assisted genetic algorithm PSGA has been developed by the hybridization of particle swarm optimization, and an improved real-coded genetic algorithm, for constrained optimization problems. A rank-based multi-parent crossover operator is proposed for the genetic algorithm, based on the mutation operator of differential evolution. An effective parameter-free constraint-handling scheme is implemented. The population with only worst fitness is enhanced through the particle swarm phase, to improve the efficiency. The performance of PSGA has been demonstrated for the design of a stepped cantilever beam, with mixed discrete-continuous design variables. The following remarks can be made, based on the obtained

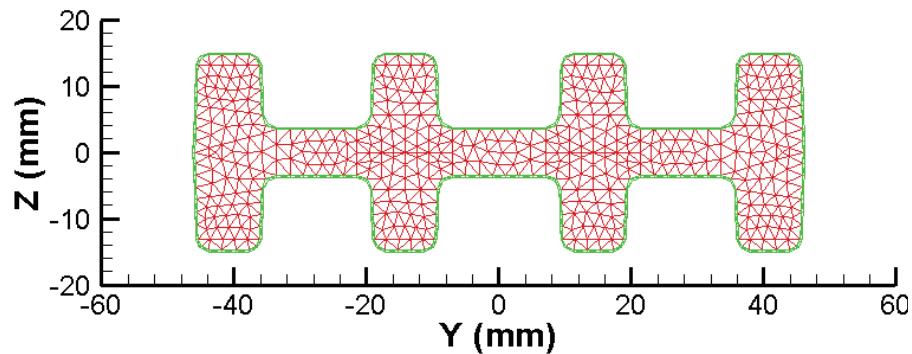


Fig. 8. Baseline configuration of flexbeam section-B

Table 8. Optimal results for flexbeam section-B

Design variables	Baseline	Optimum	Difference (%)
m (kg/mm)	3.19884E-03	3.10663E-03	-2.88
EA (N)	7.94924E+07	7.71275E+07	-2.98
EI_Y (N.mm ²)	4.30482E+09	4.13903E+09	-3.85
EI_Z (N.mm ²)	6.68440E+10	6.35769E+10	-4.89
GJ (N.mm ²)	2.57213E+08	2.44642E+08	-4.89
FI_{max}	0.3998	0.4162	4.10

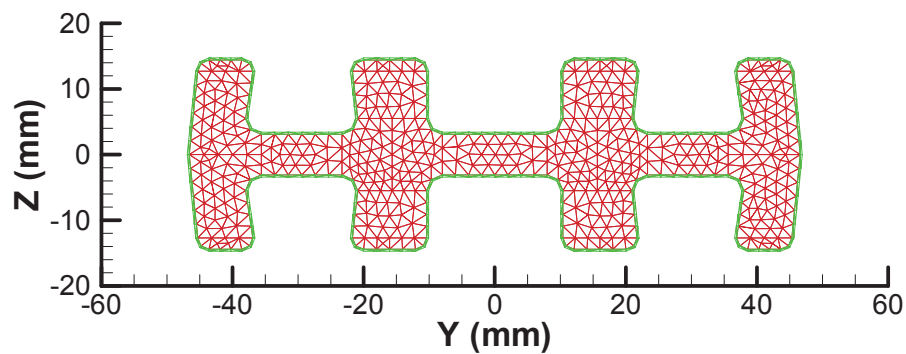


Fig. 9. Optimized configuration of flexbeam section-B

results:

1) The PSGA is found to be more efficient than other state-of-the-art algorithms. The efficiency of PSGA is attributed to the (a) enhancement of population with only worst fitness in the particle swarm phase, and (b) rank-based multi-parent crossover operation in the genetic algorithm phase, which involves the improvement due to the scaling factor and crossover rate.

2) The PSGA always finds the best solution in the feasible domain. This is due to the rank-based multi-parent crossover, which helps guarantee the feasibility of the solution, by shifting the solutions toward the feasible region. In addition, the feasibility-based relaxation scheme in the constraint-handling technique ensures that the optimum solution does not get trapped in local optima.

3) Multiple independent runs verify the reliability of the PSGA in obtaining the consistent optimum solution, which can be measured through the standard deviation. The parameters do not require any additional tuning, based on the type of the problem.

The PSGA is successfully employed to design the flexbeam sections of a full-scale bearingless helicopter, with all discrete design variables. Although the problem size may be small, the number of function evaluations (section analyses) required to reach the optimal solution are reasonable, within very less CPU time. The optimization is performed on two different configurations. The dependency of the optimal solution and the optimum objective function on the section constraints can easily be identified. The PSGA can thus be efficiently utilized for the optimal design of composite rotor blade components, such as flexbeam, torque tube and blade airfoil sections.

Acknowledgement

This research was supported by KARI under the Technology Development of Bearingless Main Rotor Hub System funded by the Ministry of Knowledge Economy (MKE) of Korea.

References

- [1] Talbi, E. G., *Metaheuristics: From Design to Implementation*, John Wiley & Sons, 2009.
- [2] Blum, C., and Roli, A., "Metaheuristics in combinatorial optimization: overview and conceptual comparison", *ACM Computing Surveys*, Vol. 35, No. 3, 2003, pp. 268-308.
- [3] Murugan, M. S., Suresh, S., Ganguli, R., and Mani, V., "Target vector optimization of composite box beam using real-coded genetic algorithm: a decomposition approach", *Structural and Multidisciplinary Optimization*, Vol. 33, 2007, pp. 131-146.
- [4] Suresh, S., Sujit, P. B., Rao, and A. K., "Particle swarm optimization approach for multi-objective composite box-beam design", *Composite Structures*, Vol. 81, Issue 4, 2007, pp. 598-605.
- [5] Deb, K., "An efficient constraint handling method for genetic algorithms", *Computer Methods in Applied Mechanics and Engineering*, Vol. 186, Issues 2-4, 2000, pp. 311-338.
- [6] Zhang, H., *New strategies for global optimization of chemical engineering applications by differential evolution*, Ph.D. Thesis, National University of Singapore, 2012.
- [7] Erbatır, F., Hasançebi, O., Tütüncü, İ., Kılıç, H., "Optimal design of planar and space structures with genetic algorithms", *Computers and Structures*, Vol. 75, Issue 2, 2000, pp. 209-224.
- [8] Lemonge, A. C. C., and Barbosa, H. J. C., "An adaptive penalty scheme for genetic algorithms in structural optimization", *International Journal for Numerical Methods in Engineering*, Vol. 59, Issue 5, 2004, pp. 703-736.
- [9] Bernardino, H. S., Barbosa, H. J. C., and Lemonge, A. C. C., "A hybrid genetic algorithm for constrained optimization problems in mechanical engineering", *Proceedings of the IEEE Congress on Evolutionary Computation (CEC2007)*, Singapore, 2007, pp. 646-653.
- [10] Bernardino, H. S., Barbosa, H. J. C., and Lemonge, A. C. C., Fonseca, L. G., "A new hybrid AIS-GA for constrained optimization problems in mechanical engineering", *Proceedings of the IEEE Congress on Evolutionary Computation (CEC2008)*, Hong Kong, China, 2008, pp. 1455-1462.
- [11] Park, I. J., Dhadwal, M. K., Jung, S. N., and Kim, D. H., "Experimental validation of cross-sectional analysis for composite rotor blades", *Proceedings of the 18th International Conference on Composite Materials (ICCM18)*, Jeju Island, Korea, 21-26 August, 2011.