IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)
e-ISSN: 2278-1684,p-ISSN: 2320-334X, Volume 14, Issue 3 Ver. V. (May - June 2017), PP 124-130
www.iosrjournals.org

Genetic Algorithm Parameters Effect on the Optimal Structural Design Search

Z. El Maskaoui[1], S. Jalal[2], L. Bousshine[3]
[1, 2, 3](Materials *and Structural Mechanics Team, ENSEM, Hassan II University of CASABLANCA, MOROCCO)*

Abstract: *This paper investigates the effects of genetic algorithm parameters on the performance of optimum structural search. The most significant of these parameters can be grouped according to their biologically-inspired functions: population size, initial population, and crossover and mutation operators. However, since the genetic algorithms use a random search the numerical results presented in this paper show the extent to which the quality of solution depends on the choice of these parameters.*

Keywords: *Optimization, Genetic algorithm, Selection, Crossover, Mutation, Structures engineering*

I. Introduction

Genetic algorithms (GAs) are a class of stochastic relaxation techniques that are applicable to the solution of a wide variety of optimization engineering problems [1–10] by emanating the evolutionary behaviour of biological systems. They are global optimizers due to their population search. They have great advantages over traditional methods for solving optimization problems in design of civil engineering structures, since they can be applied simultaneously with continuous or discrete design variables. In contrast to sequential search methods that, at each iteration, generate a single potential solution from the last, GAs maintain a large population of candidate solutions. Each population is generated from its predecessor by applying a set of stochastic transition operators [11-12]. Three commonly used genetic operators are employed : selection, crossover and mutation . These operators are applied in turn to the solutions in the current generation during the search process.

The selection operator identifies the fittest individuals of the current population to serve as parents of the next generation. Jebari et al [13] are analyzed relative performance of several selection methods often used in GAs and showed the extent to which the quality of solution depends on the choice of the selection method. To explore the search space, the crossover operator consists of choosing a pair of individuals among those survived from a previous generation, and then offspring are generated using a mechanism that inherits valid characteristics of the two parents. Several crossover techniques have been proposed and their relative effectiveness is still continuing [14–16]. As selection and crossover introduce better solutions, mutation is used in hopes of nudging good solutions closer to the best solution. The probability of applying the mutation is low. In contrast, the probability of crossover is usually high. Numerous studies have investigated the optimal setting for the mutation probability [17–23]. However the optimal solution of GAs optimization as well as the convergence of the algorithm depends to a very large extent on the choice of genetic operator's parameters. Hence the present paper examines the effects of these parameters on the performance of optimum structural search like population size, initial population, and crossover and mutation probabilities.

II. Illustrative example

To show the effect of genetic parameters on the performance of optimum structural search, let' s consider a 10-bar truss optimization [24], with a minimum weight as shown in Fig.1. Nodes 5 and 6 are fixed by the hinges. The truss loads $F = 444.822\,kN$ act at nodes 2 and 4.

The material properties are $\rho = 2770\,kg/m^3$ and $E = 6.89\times10^4\,MPa$. The design variables are the cross-section areas of truss members. The list of possible cross sections (Table 1) is taken from the American Institute of Steel Construction Manual. Each design variable will be chosen from 32 values of table 1 that may be feasible solutions. Then the cross-section for members 1, 3, 4, 7, 8 and 9 are chosen from section numbers 11 to 42, the others are chosen from section numbers 1 to 32.

Since there are 10 design variables, and each can take any of the 32 values of the list, the intrinsic size of the search space is 32^{10}. Five bits are required to represent the 32 available sections ($2^5 = 32$), assigning random values from list to the extra codes. Thus, each chromosome is 50 bits long (5 bits/bar x 10 bars).

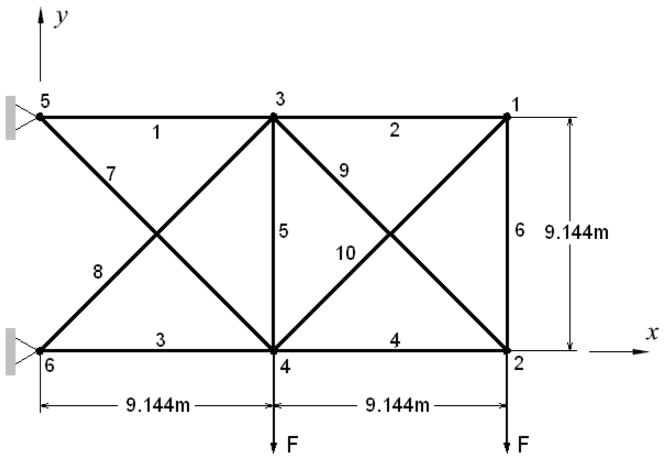

Figure 1 : 10-bar truss

Table 1 : Cross-section list

N°	Area (cm²)	N°	Area (cm²)	N°	Area (cm²)	N°	Area (cm²)
1	10.45	12	21.81	23	30.97	34	103.22
2	11.61	13	22.39	24	32.06	35	109.03
3	12.84	14	22.90	25	33.03	36	121.29
4	13.74	15	23.42	26	37.03	37	128.39
5	15.35	16	24.77	27	46.58	38	141.93
6	16.90	17	24.97	28	51.42	39	147.74
7	16.97	18	25.03	29	74.19	40	170.96
8	18.58	19	26.97	30	87.10	41	193.55
9	18.90	20	27.23	31	89.68	42	216.13
10	19.93	21	28.97	32	91.61	-	-
11	20.19	22	29.61	33	100.00	-	-

III. Problem formulation

We formulate the optimal design problem with a satisfied displacement $u_{y\max} = -5.08cm$ and allowable stress $\sigma_a = \pm 172.37\,MPa$ with a minimized 10 bar truss weight W(A) :

$$W(A) = \rho \sum_{i=1}^{10} A_i L_i \tag{1}$$

The normalized forms of the design constraints are expressed as follows :

$$g_j = \frac{\sigma_j}{\sigma_a} - 1 \leq 0, \quad j = 1,..,10 \tag{2}$$

$$g_k = \frac{u_k}{u_{y\max}} - 1 \leq 0 \qquad k = 1,..,6 \tag{3}$$

where A_i is the cross-section area of the element,
Li is the length of the element,
σ_j is the stress in member j,
u_k is the vertical displacement of node k.

To replace a constrained optimization problem by a unconstrained problem we use a penalty method [12]. The unconstrained problems are formed by adding a penalty function to the objective function that consists of a penalty parameter multiplied by a measure of violation of the constraints. The measure of violation is nonzero when the constraints are violated and is zero in the region where constraints are not violated.

$$\Phi = W\left(1 + C\sum_{i=1}^{n_c} P_i\right) \tag{4}$$

where C constant determined by users,
n_c number of constraints,
P_i penalty function given by :

$$P_i = \begin{cases} g_i & if \quad g_i > 0 \\ 0 & if \quad g_i \le 0 \end{cases} \qquad i = 1,..,n_c \tag{5}$$

GA strategy is a maximization search. This means that the best individuals have a higher probability of surviving. Therefore, we have to transform the objective function minimization problem to the fitness function maximization problem. Consequently, the fitness function F is defined as follows :

$$F_i = [\Phi_{\max} + \Phi_{\min}] - \Phi_i \tag{6}$$

where F_i is the fitness function, $\Phi_{\max}$ and $\Phi_{\min}$ are, respectively, the maximum and minimum value of objective function Φ.

IV. Influence of the initial population

To show the effect of the initial population on the quality of the final results we consider two processes of optimization with tow initial populations Pop1 and Pop2 (Table 2). The initial population Pop1 contains individuals which represent the set of points in the search space while the initial population Pop2 contains only individuals associated with a limited part of the search space. Although the other GA parameters are identical for both processes, the results are different. The diversity of individuals in the Pop1 allows convergence to the best results (Fig. 2).

Table 2 : Initial populations Pop1 and Pop2

	POP1		POP2	
Ind.	Weight (kg)	Section numbers	Weight (kg)	Section numbers
1	2020,1	14-26-30-15-17-21-23-39-41-25	4225,6	42-32-34-42-29-22-42-42-41-26
2	1301,2	16-6-21-22-21-5-33-16-34-25	4220,1	32-32-26-42-32-32-42-42-42-32
3	1849,4	12-15-34-36-11-26-28-35-25-32	4187,1	40-31-42-36-12-32-42-38-42-32
4	1616,3	34-20-36-29-29-8-21-27-25-27	3488,3	34-32-42-30-32-16-37-34-42-32
5	2176,7	42-20-22-32-26-14-12-40-31-18	4203,6	42-32-42-42-32-32-26-42-42-28
6	1080,0	19-9-16-31-1-22-21-16-24-29	4050,2	42-30-34-42-32-32-42-42-26-32
7	1118,0	27-28-19-13-24-6-11-31-27-7	4204,4	42-32-26-42-32-16-40-42-42-32
8	1721,9	35-9-14-32-25-19-29-16-36-27	4306,2	42-28-42-32-32-32-38-42-42-32
9	1799,4	23-17-35-13-17-29-32-40-12-5	3759,3	24-32-41-41-32-32-42-42-34-15
10	1842,9	33-18-16-34-3-27-27-38-31-5	4271,5	41-32-34-42-32-28-42-38-42-31
11	1932,7	37-19-36-31-10-4-23-36-32-3	4190,6	42-32-42-42-32-16-41-42-34-28
12	1859,8	28-3-33-24-19-4-37-36-32-1	4373,7	42-32-42-26-32-32-42-40-42-32
13	2403,4	33-5-13-35-32-2-26-37-40-30	3669,3	42-32-26-41-32-24-26-42-42-30
14	1453,5	39-28-33-31-26-3-21-16-21-3	4228,0	42-32-42-41-26-32-42-34-42-27
15	1340,0	15-12-19-22-13-18-36-14-35-5	3869,4	42-32-42-38-32-32-42-34-38-8
16	996,1	21-25-36-17-22-5-25-15-11-13	3471,7	38-31-40-34-30-32-42-34-29-32
17	1321,5	24-22-19-11-21-29-29-23-32-13	3300,5	42-24-42-25-30-32-34-26-42-30
18	863,0	14-22-19-34-3-6-20-21-14-2	4443,0	38-32-39-42-32-8-42-42-42-32
19	1171,5	15-7-16-15-7-2-34-11-27-29	3683,6	24-24-32-40-32-12-42-42-41-32
20	1691,6	27-21-16-34-18-23-29-39-11-27	3924,2	42-32-42-42-30-24-42-41-24-27

Table 3 : Final results with initial populations Pop1 and Pop2

	Pop1	**Pop2**
Crossover average	173.32	172.63
Mutation average	3.67	3.64
Optimum (kg)	2519.56	2669.79
Section list of optimum	41-1-38-31-1-1-28-40-39-1	42-17-38-30-1-11-35-34-38- 26

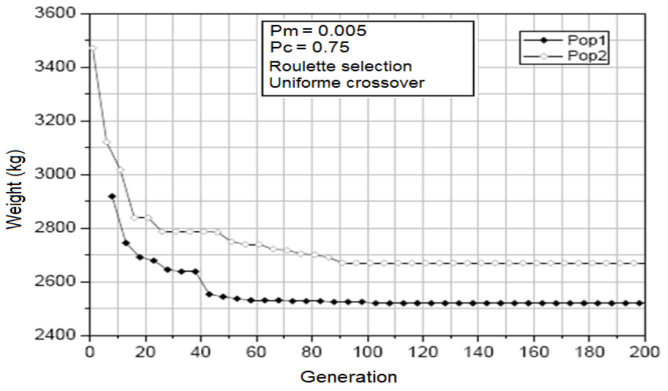

Figure 2 : Initial population effect on the quality of the final results

V. Influence of the population size

Fig. 3 and 4 show comparing the population size effect on the speed and the quality of the final results. We observe the beneficial effect of height population size in terms of quality of the final results. However, a small population size allows an initial faster convergence, but a worse final result. This can be explained because the quality of final solution needs more population diversity -it depends on the population size- to avoid premature stagnation.

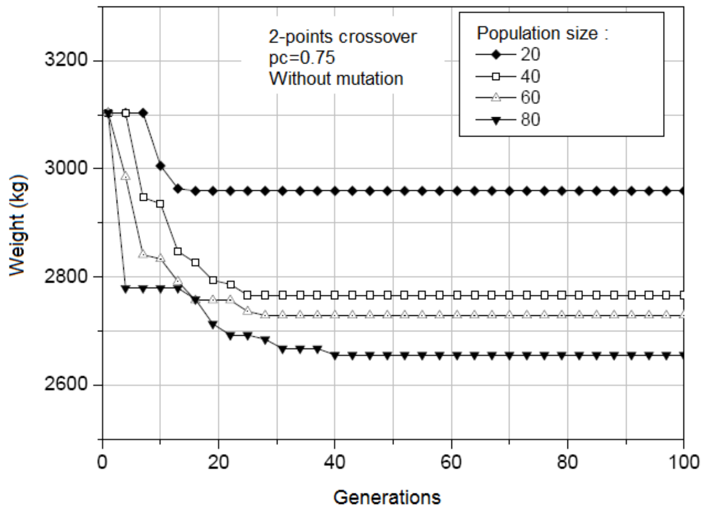

Figure 3 : Best individuals weight evolution (Without mutation)

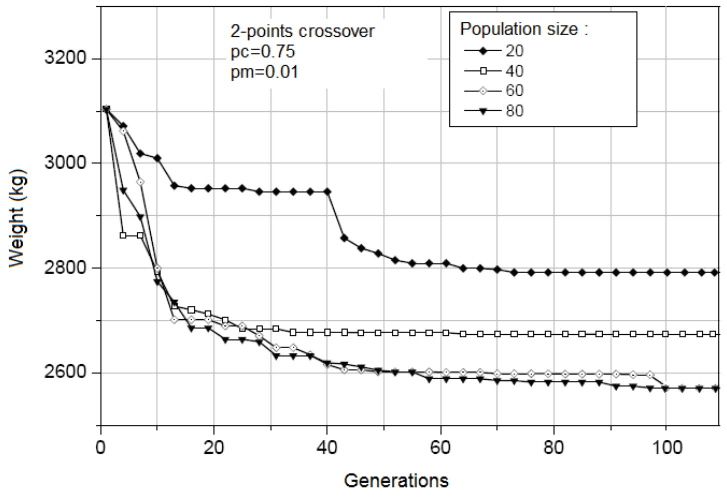

Figure 4 : Best individuals weight (With mutation)

VI. Influence of mutation probability

Mutation is expected to introduce diversity since it is able to insert new individuals into the population. Fig. 5 and 6 illustrate the influence of the mutation probability rate on the quality of the final results. The higher this rate, the greater the population strongly changes. That will keep shaking things up enough so that other parts of the solution space will be explored and the global optimum can be achieved. However, if the mutation probability is large enough to prevents premature convergence it gives poor results.

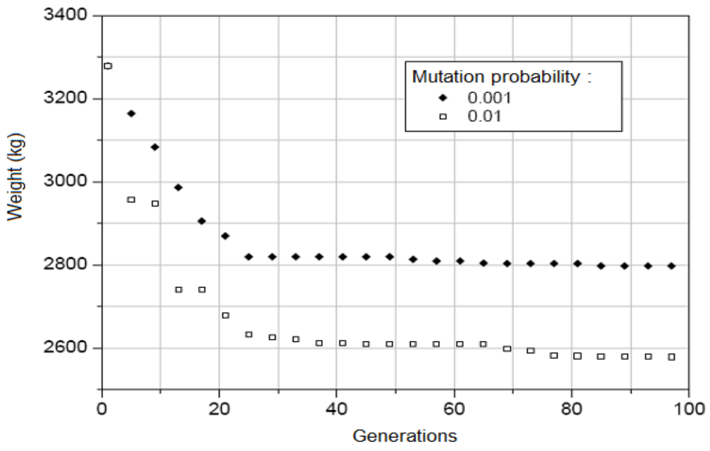

Figure 5 : Weight evolution of the best individuals

VII. Influence of crossover implementation

Crossover provides an exchange of design characteristics between paired individuals. In this paper, a comparative study is carried out between a single, 2-points and uniform crossovers technique for selected proportions of generations with a fixed generation number. Fig. 7 sows that a uniform crossover single point resulted in better solutions when compared to the single and 2-points crossovers. The uniform crossover gives a higher chance to individuals in exchanging the foremost genes of their chromosomal strings with respect to tow other crossovers technique. Fig. 8 illustrates the influence of the crossing probability rate, without mutation, on the problem convergence. The lower the rate, the less the population is destroyed. A large enough of crossover probability leads to suboptimal solutions.

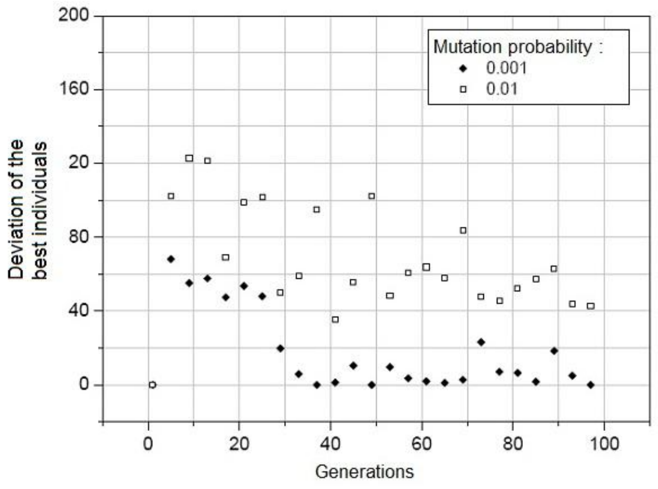

Figure 6 : Mutation probability effect on the weight standard deviation of the best individuals

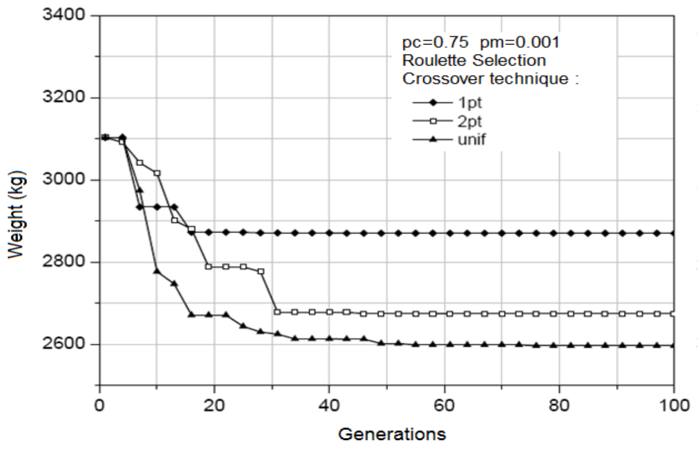

Figure 7 : Crossovers technique effect

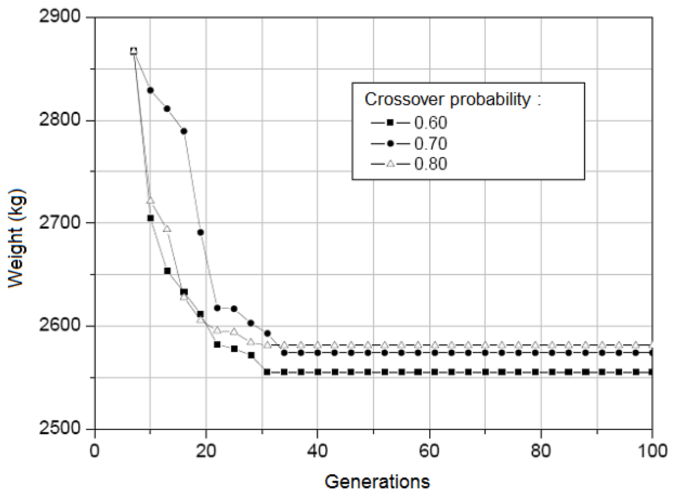

Figure 8 : Crossover probability effect on the weight of the best individuals

VIII. Stability of the solution

Design structures optimization by using GAs requires to ensure the results convergence obtained by several process run with the same starting population and the identical parameters. Fig. 9 shows the optimal weight obtained after each optimization process launches. Although the GAs parameters are identical, the optimization process behaves differently. This is because GAs use random factors. The previous results show that it is advisable to remain cautious about the absolute value of an optimal solution and that the process should preferably be executed several times. They also show that of the 100 solutions obtained, 29 of them do not differ from the optimum by more than 3%.

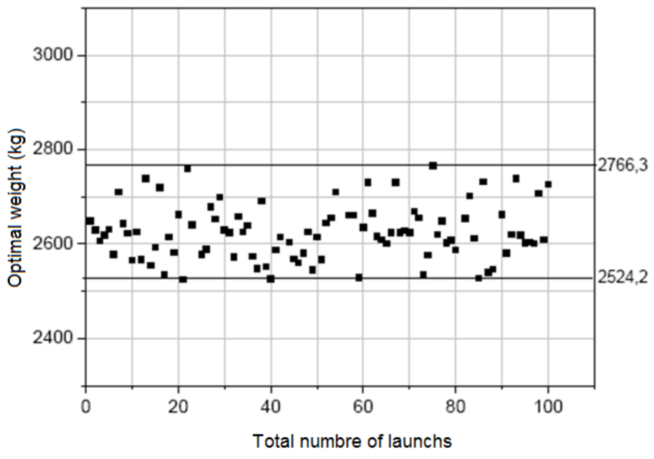

Figure 9 : Optimal weight for each total number launch

Table 4 : Optimization results after several runs

Run	Optimal (kg)	Section numbers	Mutation average	Crossover average
1	2649,26	42-1-39-35-1-1-30-40-33-17	3,89	363,63
20	2661,94	40-1-42-30-1-6-37-35-36-13	3,57	358,31
40	2525,86	42-1-39-33-1-1-29-38-37-1	3,52	348,16
60	2730,62	41-3-41-32-17-3-39-30-37-16	3,86	358,45
80	2933,74	42-9-42-26-1-21-39-33-40-20	3,75	354,09
100	2726,67	39-23-42-35-1-17-37-36-34-19	3,53	350,95

IX. Conclusion

This paper has discussed and evaluated the effect of GAs parameters on the quality of the final optimization results. First, the quality of final solution needs more population diversity -it depends on the population size- to avoid premature stagnation. However, a small population size allows an initial faster convergence, but a worse final result. Then an initial population with individuals which represent the set of points in the search space allows convergence to the best results.

On the other hand, uniform crossover gives a higher chance to individuals in exchanging the foremost genes of their chromosomal strings with respect to single and 2-points crossovers technique. Large enough of crossover probability leads to suboptimal solutions. Finally, large or small enough mutation probability prevents premature convergence and gives poor results. Good use of these parameters increases the chance that the genetic algorithm will find the optimum solution, and improves the value of the best solution found even when the optimum solution is not found.

References

[1] M. A. Rao, J. Srinivas, and B. S. N. Murthy, "Damage detection in vibrating bodies using genetic algorithms," Comput. Struct., vol. 82, pp. 963–968, 2004.

[2] S. Y. Wang and K. Tai, "Graph representation for structural topology optimization using genetic algorithms," Comput. Struct., vol. 82, no. 20–21, pp. 1609–1622, 2004.

[3] C. Xu, S. Lin, and Y. Yang, "Optimal design of viscoelastic damping structures using layerwise finite element analysis and multi-objective genetic algorithm," Comput. Struct., vol. 157, pp. 1–8, 2015.

[4] E. S. Kameshki and M. P. Saka, "Optimum design of nonlinear steel frames with semi-rigid connections using a genetic algorithm," Comput. Struct., vol. 79, no. 17, pp. 1593–1604, 2001.

[5] M. Y. Rafiq and C. Southcombe, "Genetic algorithms in optimal design and detailing of reinforced concrete biaxial columns supported by a declarative approach for capacity checking," Comput. Struct., vol. 69, no. 4, pp. 443–457, 1998.

[6] C. A. Coello and A. D. Christiansen, "Multiobjective optimization of trusses using genetic algorithms," Comput. Struct., vol. 75, no. 6, pp. 647–660, 2000.

[7] J.-H. Chou and J. Ghaboussi, "Genetic algorithm in structural damage detection," Comput. Struct., vol. 79, no. 14, pp. 1335–1353, 2001.

[8] F. Erbatur, O. Hasançebi, İ. Tütüncü, and H. Kılıç, "Optimal design of planar and space structures with genetic algorithms," Comput. Struct., vol. 75, no. 2, pp. 209–224, 2000.

[9] C. Xu, S. Lin, and Y. Yang, "Optimal design of viscoelastic damping structures using layerwise finite element analysis and multi-objective genetic algorithm," Comput. Struct., vol. 157, pp. 1–8, 2015.

[10] R. de P. Garcia, B. S. L. P. de Lima, A. C. de C. Lemonge, and B. P. Jacob, "A rank-based constraint handling technique for engineering design optimization problems solved by genetic algorithms," Comput. Struct., vol. 187, pp. 77–87, 2017.

[11] J. H. Holland, Adaptation in Natural and Artificial Systems, vol. Ann Arbor. 1975.

[12] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. 1989.

[13] K. Jebari and M. Madiafi, "Selection Methods for Genetic Algorithms," Int. J. Emerg. Sci., vol. 3, no. 4, pp. 333–344, 2013.

[14] O. Hasançebi and F. Erbatur, "Evaluation of crossover techniques in genetic algorithm based optimum structural design," Comput. Struct., vol. 78, no. 1, pp. 435–448, 2000.

[15] C. Modelling, H. Hirabayashi, K. Katayama, and H. Narihisa, "Analysis of crossovers and selections in a coarse-grained parallel genetic algorithm," Math. Comput. Model., vol. 38, no. 11, pp. 1275–1282, 2003.

[16] A. Wu, R. Lindsay, and R. Riolo, "Empirical observations on the roles of crossover and mutation," Ann Arbor, no. 1989, pp. 1–26, 1997.

[17] J. Lis, "Genetic algorithm with the dynamic probability of mutation in the classification problem," Pattern Recognit. Lett., vol. 16, no. 12, pp. 1311–1320, 1995.

[18] R. L. Haupt and S. E. Haupt, "Optimum Population Size and Mutation Rate for a Simple Real Genetic Algorithm that Optimizes Array Factors," Appl. Comput. Electromagn. Soc. Newsl., vol. 15, no. 2, pp. 94–102, 2000.

[19] J. Smith and T. C. Fogarty, "Self adaptation of mutation rates in a steady state genetic algorithm," in Proceedings of IEEE International Conference on Evolutionary Computation, 1996, pp. 318–323.

[20] I. De Falco, A. Della Cioppa, and E. Tarantino, "Mutation-based genetic algorithm: Performance evaluation," Appl. Soft Comput., vol. 1, no. 4, pp. 285–299, 2002.

[21] K. K. Mishra, S. Tiwari, A. Kumar, and A. K. Misra, "An approach for mutation testing using elitist genetic algorithm," in Proceedings - 2010 3rd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2010, 2010, vol. 5, pp. 426–429.

[22] A. Saitoh, R. Rahimi, and M. Nakahara, "A quantum genetic algorithm with quantum crossover and mutation operations," Quantum Inf. Process., vol. 13, no. 3, pp. 737–755, 2014.

[23] D. Smullen, J. Gillett, J. Heron, and S. Rahnamayan, "Genetic algorithm with self-adaptive mutation controlled by chromosome similarity," in Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014, 2014, pp. 504–511.

[24] S. Rajeev and C. S. Krishnamoorthy, "Genetic Algorithms-Based Methodologies for Design Optimization of Trusses," Journal of Structural Engineering, vol. 123, no. 3. p. 350, 1997.