

Genetic Algorithm Parameter Effect on 3D Truss Optimization with Discrete Variable



Akshay Kumar, H. K. Rangavittal

Dept. of Mechanical Engineering, BMS College of Engineering, Bull Temple Road, Bengaluru, INDIA

ABSTRACT

The Genetic Algorithm is one of the advanced optimization techniques frequently used for solving complex problems in the research field, and there are plenty of parameters which affect the outcome of the GA. In this study, a 25-bar truss with the nonlinear constraint is chosen with the objective to minimize the mass and variables being the discrete area. For the same, GA parameter like Selection Function, Population Size, Crossover Function, and Creation Function are varied to find the best combination with minimum function evaluation. It is found that the Uniform selection gives the best result irrespective of the creation function, population size or crossover functions. But this is at the cost of a large number of function evaluations, and the other selection function fails to reach the global optimum and has a smaller number of function evaluation count. If the analysis of selection function is done one at a time, it is seen that all Cases performs better in Roulette but, Case A which is non-integer type with 200 population size being computationally cheaper than Case B and C of population size 300. In the Tournament selection, Case A, B with smaller population size and Case C with higher population size performs better. Case C performs better at Remainder selection with smaller population size, and Case A and B for Stochastic Uniform with higher population size. And, it is clear that the function evaluation count increases with the population size in every Case from this study.

Keywords: Genetic Algorithm; 3D Truss; Optimization; Discrete Variable; MATLAB.

*Corresponding Author email:
akshaykumar00325@gmail.com

Article History

Received: 05 September 2018

Revised: 11 November 2018

Accepted: 25 November 2018

Published: 12 December 2018

Student(s)

- Akshay Kumar

Academic Year: 2017-18

Course Level: Bachelor

Course Name: B.E. (Mechanical Engineering)

Course year: 4th Year

Mentor(s)

- H. K. Rangavittal

1 Introduction

Optimization of the structure is one of the design methods followed by the designer to get the adequate result. There are a number of optimization method one can follow while trying for structural optimization, but the classical method breaks down or becomes complex with more number of design variables.

Moving away from classical optimization techniques, there are lots of work done on truss and frame structure optimization using advanced optimization techniques. In early years, C. Camp et al. [1] used a Genetic algorithm(GA) to optimize the frame structure. Rajeev et al. [2] gave a simple GA for optimizing

3D and 2D truss structure. V. Tongan et al. [3] used Adaptive GA where improved crossover and mutation penalty function is suggested and compared with simple GA. T. Dede et al. [4] proposed binary and the value encoded GA for optimization of the space truss. Also new crossover techniques in GA for structural optimization by O. HasancEebi et al. [5]. Other approaches include; C. A. Coello et al. [6] using multi-objective optimization on different 3D truss cases, by taking 3 objective functions. Schutte et al. [7] made use of Particle Swarm Optimization algorithm and M. Kripka [8] used Simulated Annealing algorithm to find the solution for discrete optimization of truss comparing it with GA and other algorithms.

A. Kaveh et al. [9] proposed a hybrid Particle swarm and Ant colony optimization techniques, and Hybrid Big-Bang-Big-Crunch algorithm [10] as well. G. Luh et al. [11] presented 2 stage particle swarm optimization. But GA still remains one of the most used optimization algorithms, due to its flexibility to tune the parameter to get the optimum result.

Moreover, many works are done in recent years such as: finding out the influence of various parameters of the GA effect on structural optimization by Z. El Maskaoui et al. [12] which concentrated on the crossover, mutation probability, and population size. Ramkumar. P. et al. [13] used differential evolution algorithm to optimize the plane truss member cross-section area, D. Neeraja et al. [14] along with size; shape and topology optimization was implemented through MATLAB on plane truss using GA, and H Assimi et al. [15] extended size and topology optimization of planar truss using GA to more benchmark problem and did a comparison. High-performance GA to optimize the space truss, including 25 bar space truss was developed by I. Serpik et al. [16]. V. R. Kalatjari et al. [17] used the method of partitioning the design space to increase optimization speed in GA in the skeletal structure. Most of the studies fail to take into consideration the many parameters which need to be properly selected for optimization of the problem, and not much study of the GA parameter effect has been carried out which gives a research opportunity.

The aim of this study was to compare different cases with each having different Genetic Algorithm parameters like Selection Function, Population Size, Crossover Function, and Creation Function and find the best combination which minimizes the mass and has low function evaluation for 25 bar space trusses with non-linear constraint.

2 25 Bar Truss Optimization

Truss structure has the flexible design space which gives a design engineer number of variables to design. The stress in the truss member and the displacement of each node are the main concern while designing, which leads us to problem definition for optimization.

2.1 Problem Definition

Material Properties [2]:

Modulus of Elasticity (E) = 10^4 ksi = 6.89×10^4 MPa

Density = 0.10 lb/in³ = $2,770$ kg/m³

25 bar Truss element arrangement for this problem is given in the Fig.1.

The table view of node location in mm is as shown in table 1.

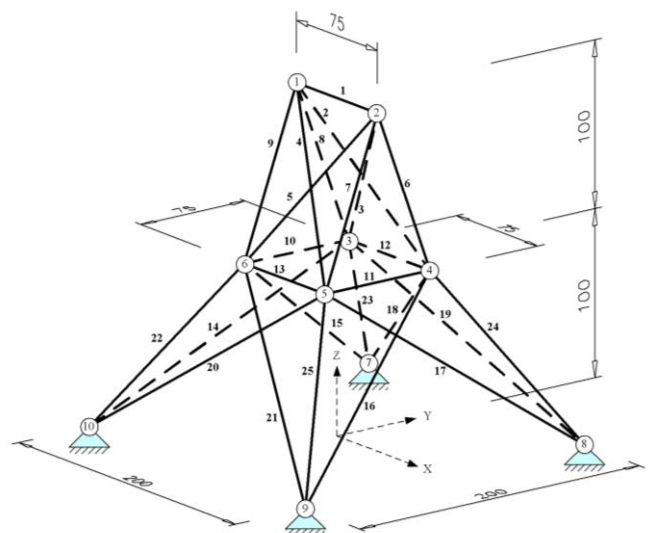


Figure 1: Dimensions of 25 Bar Truss in inches

Table 1: Node Location of 25 Bar Truss in mm

| | | Location in mm | | | | | Location in mm | | |
|---------|--------|----------------|------|---------|--------|--------|----------------|--|--|
| Node No | X | Y | Z | Node No | X | Y | Z | | |
| 1 | -952.5 | 0 | 5080 | 6 | -952.5 | -952.5 | 2540 | | |
| 2 | 952.5 | 0 | 5080 | 7 | -2540 | 2540 | 0 | | |
| 3 | -952.5 | 952.5 | 2540 | 8 | 2540 | 2540 | 0 | | |
| 4 | 952.5 | 952.5 | 2540 | 9 | 2540 | -2540 | 0 | | |
| 5 | 952.5 | -952.5 | 2540 | 10 | -2540 | -2540 | 0 | | |

The 25 elements are divided into 8 groups and each group will have a particular area truss, the element numbers are as shown in table 2.

Table 2: Element Grouping of 25 Bar Truss

| Element No | Node1 | Node2 | Group No | Area No | Element No | Node1 | Node2 | Group No | Area No |
|------------|-------|-------|----------|---------|------------|-------|-------|----------|---------|
| 1 | 1 | 2 | 1 | A1 | 14 | 3 | 10 | 6 | A6 |
| 2 | 1 | 4 | 2 | A2 | 15 | 6 | 7 | 6 | A7 |
| 3 | 2 | 3 | 2 | | 16 | 4 | 9 | 6 | |
| 4 | 1 | 5 | 2 | | 17 | 5 | 8 | 6 | |
| 5 | 2 | 6 | 2 | | 18 | 4 | 7 | 7 | |
| 6 | 2 | 4 | 3 | A3 | 19 | 3 | 8 | 7 | A8 |
| 7 | 2 | 5 | 3 | | 20 | 5 | 10 | 7 | |
| 8 | 1 | 3 | 3 | | 21 | 6 | 9 | 7 | |
| 9 | 1 | 6 | 3 | A4 | 22 | 6 | 10 | 8 | A8 |
| 10 | 6 | 3 | 4 | | 23 | 3 | 7 | 8 | |
| 11 | 4 | 5 | 4 | | 24 | 4 | 8 | 8 | |
| 12 | 3 | 4 | 5 | A5 | 25 | 5 | 9 | 8 | A8 |
| 13 | 6 | 5 | 5 | | | | | | |

Node number 7, 8, 9, and 10 is completely fixed. And, the loading on the structure is as shown in table 3.

Table 3: Loading on Nodes in 25 Bar Truss

| Node | Force in N | | |
|------|------------|----------|----------|
| | X | Y | Z |
| 1 | 4453.74 | -44537.4 | -44537.4 |
| 2 | 0 | -44537.4 | -44537.4 |
| 3 | 2226.87 | 0 | 0 |
| 6 | 2672.244 | 0 | 0 |

2.2 Optimization Problem Statement

Objective - Minimization of the Mass

Variables - Area of the elements (8 Variables)

Constraints-

Stress Constraints: $\sigma_i \leq \pm 257.6\text{MPa}$ (40ksi) $i=1,2,3\dots 25$

Displacement Constraints: $\delta_j \leq \pm 8.89\text{mm}$ (0.35 in) in x and y direction, $j=1,2$

Area can take only discrete values of 0.1, 0.2, ..., 3.4 in² [2]

3 Genetic Algorithm and Parameters

In GA, there are many components which define the end result, especially in nonlinear constraint problem such as this problem, these components are widely known as genetic operators. Important ones are explained below.

3.1 Population Size

Population size is the number of individuals per generation, if the population size increases then the total number of generations needed will be decreased to reach the optimum, but the number of function evaluation or computational time will increase.

3.2 Population Type

- i. Integer: Every individual of the population is an integer number, also known as a discrete variable, mainly useful in times where the result should have a standard size. Example-diameter of the bolt, thickness of the sheet.
- ii. Non-Integer: Every individual of the population can have any real number, this is also known as continuous variable type, it will lead to the global optimum, but manufacturing could be a challenge.

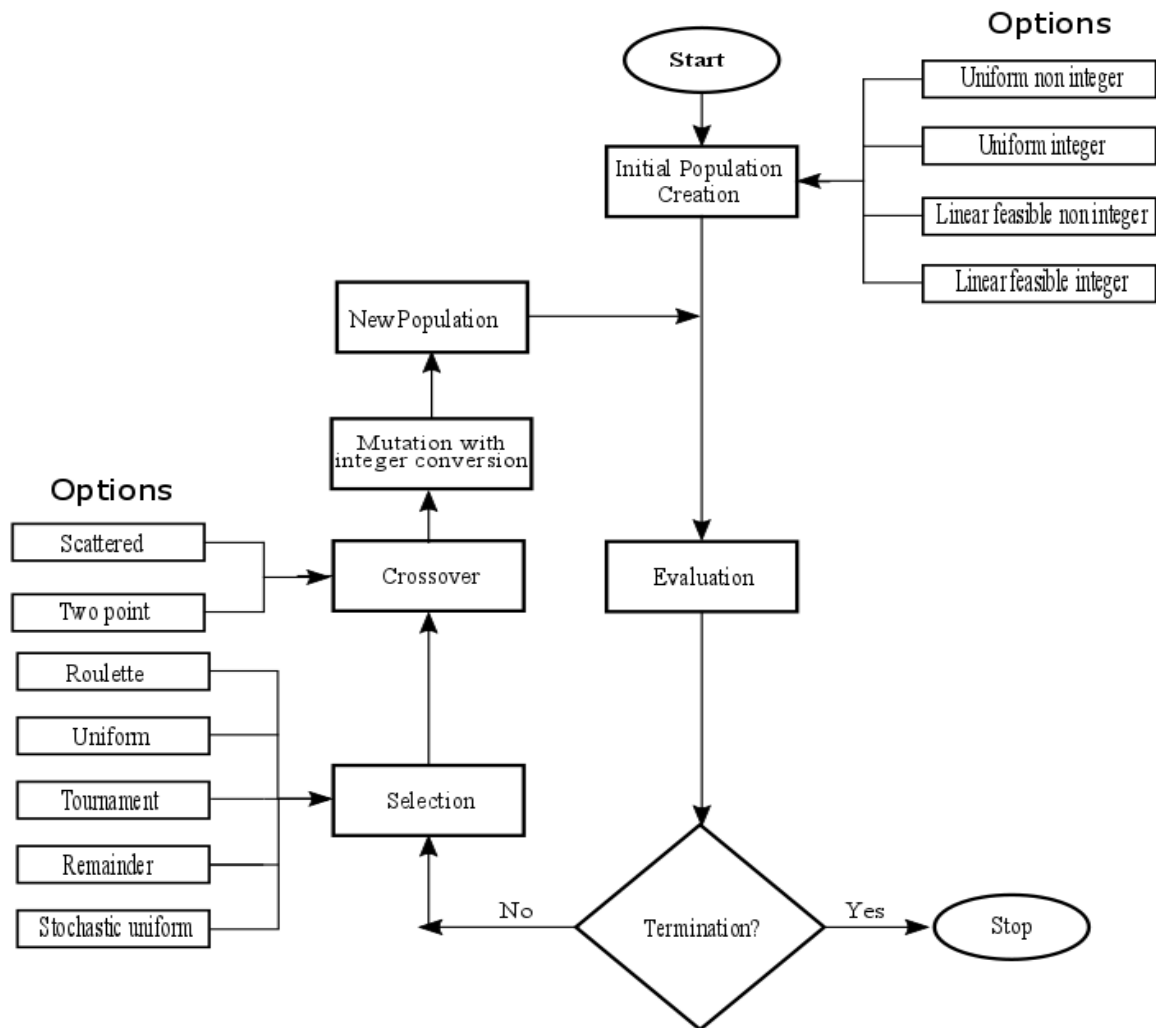


Figure 2: Genetic Algorithm Flow Chart for this study used in MATLAB

3.3 Initial Population Creation

- i. Uniform: This creates a random initial population with a uniform distribution. The uniform distribution is in the initial population range given by upper and lower bounds on the variables.
- ii. Linear feasible: This generates a random initial population that is bounded and satisfies linear constraints if it exists. And with linear constraints, the Feasible population will produce a large number of individuals on the boundaries of the constraint region and generates a scattered population. Also, feasible population neglects the initial range.

3.4 Selection function

- i. Roulette: This selects parents by simulating a roulette wheel, where the area of the section of the wheel is directly proportional to the individual's fitness value of the individual. Hence the best individual will pass the gene to reproduce.
- ii. Uniform: This selects parents using the expectations and number of parents. And it is not a very effective search strategy, but it works.
- iii. Tournament: This selects parent by creating Tournament size set and then choosing the best individual out of that to be a parent. Tournament size, in this case, will be 4.
- iv. Remainder: This selects parents deterministically from the integer part of each individual's scaled value and then uses Roulette selection on the remaining fractional part. The probability that a parent is chosen is proportional to the fractional part of scaled values.
- v. Stochastic uniform: This lays out a line in which each parent corresponds to a section of the line of length proportional to its scaled value. The algorithm moves along the line in steps of equal size. At each step, the algorithm allocates a parent from the section it lands on. The first step is a uniform random number less than the step size.

3.5 Crossover function

- i. Scattered: This creates a random binary vector and selects the genes where the vector is a 1 from the first parent and the genes where the vector is a 0 from the second parent, which is then combined to form the child.

For example, if P_1 and P_2 are the parents

P_1 = [A B C D E F G H]

P_2 = [1 2 3 4 5 6 7 8]

And if the binary vector is [1 0 0 0 1 0 1 0], the function returns the following child:

C_1 = [A 2 3 4 E 6 G 8]

- ii. Two-point: This selects two random integers m and n between 1 and the number of variables. The function selects vector entries numbered less than or equal to m from the first parent vector entries numbered from m+1 to n, inclusive, from the second parent vector entries numbered greater than n from the first parent. The algorithm then concatenates these genes to form a single gene.

For example, if P_1 and P_2 are the parents

P_1 = [A B C D E F G H]

P_2 = [1 2 3 4 5 6 7 8]

And if the crossover points are 2 and 4, the function returns the following child.

C_1 = [A b 3 4 5 F G H]

Other crossover functions are single point, arithmetic, intermediate, heuristic, etc.

3.6 Mutation function

The mutation function chosen here is an adaptive feasible function. Here it randomly generates directions that are adaptive with respect to the last successful or unsuccessful generation. The mutation chooses a direction and step length that satisfies bounds and linear constraints.

The GA procedure is independently repeated 20 times for each case as per the GA flow chart Fig. 2, with objective function and constraints values evaluated from Finite Element Model code of 25 bar Truss in MATLAB. After this independent run, the solution having the minimum value of objective function among these 20 solutions is accepted.

Table 4: Different Cases used for the Study

| | Case A | Case B | Case C |
|------------------------|--|-----------|-----------|
| Creation function | Uniform/Linear feasible | | |
| Creation function type | Non-Integer | Integer | Integer |
| Crossover function | Scattered | Scattered | Two-point |
| Mutation function | Adapt feasible | | |
| Mutation function type | Integer | | |
| Population size | 100/200/300 | | |
| Selection functions | Roulette/Uniform/Tournament/Remainder/Stochastic Uniform | | |

4 Results

4.1 Case A:

From the Fig. 3, it is evident that Uniform creation gives the minimum mass irrespective of the population size and creation function, which is better than other GA discrete variable approach [2][4][16] and heuristic search algorithm [8][18][10] till date, whereas Roulette selection shows local minimum for population size 200, and deteriorates for population size 300. Tournament selection objective function value deteriorates with increases in the population size, this is mainly due to non-integer creation function. Both Remainder and Stochastic Uniform selection result show improvement with an increase in the population size, with Remainder showing significant change compared to Stochastic uniform selection.

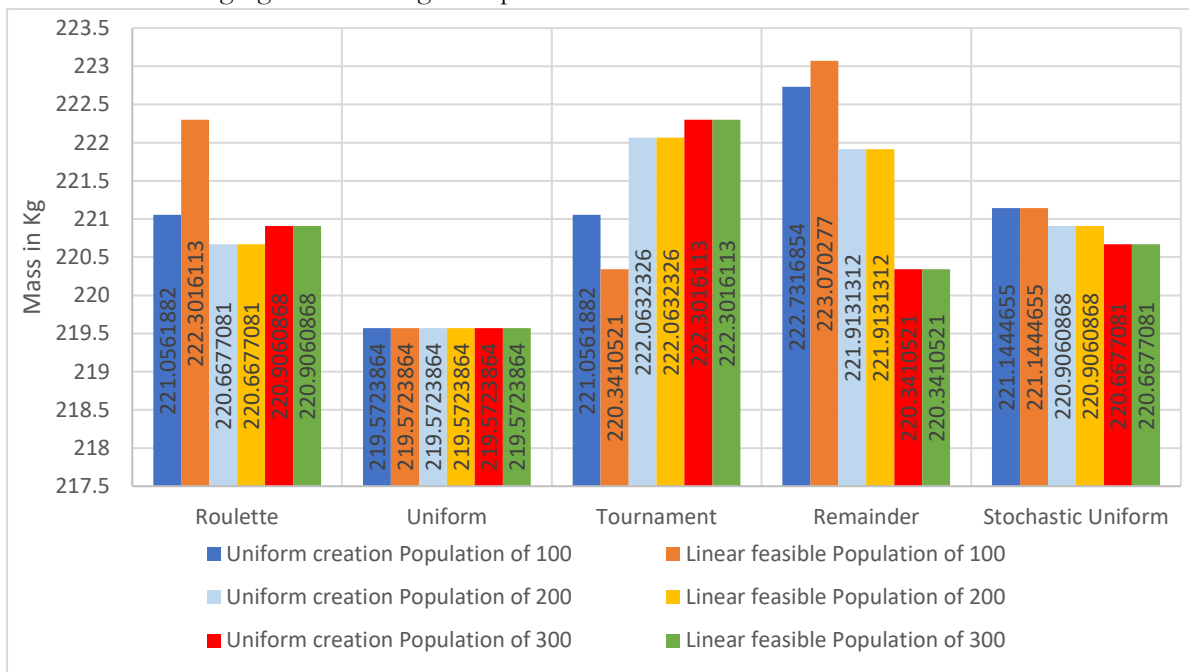


Figure 3: Case A- Minimum Mass Result for different GA Parameters

4.2 Case B:

Referring to Fig. 4, the Uniform selection function is outperforming others, also it shows no effect of creation function or population size. But there is a slight improvement in the objective function value with increase in the population size for other selection methods, except Tournament where Uniform creation

function with the population size of 100 has better objective function value than its counterparts. Also, the Uniform creation function with the population size of 100 gives much better result than Linear feasible function with the population size of 100.

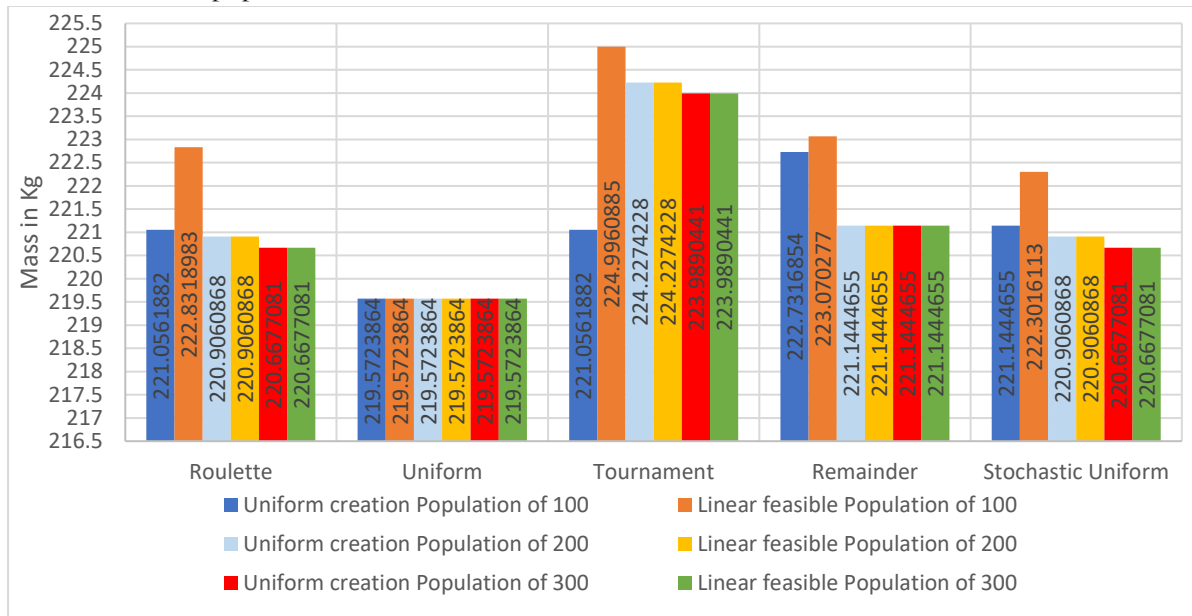


Figure 4: Case B- Minimum Mass Result for different GA Parameters

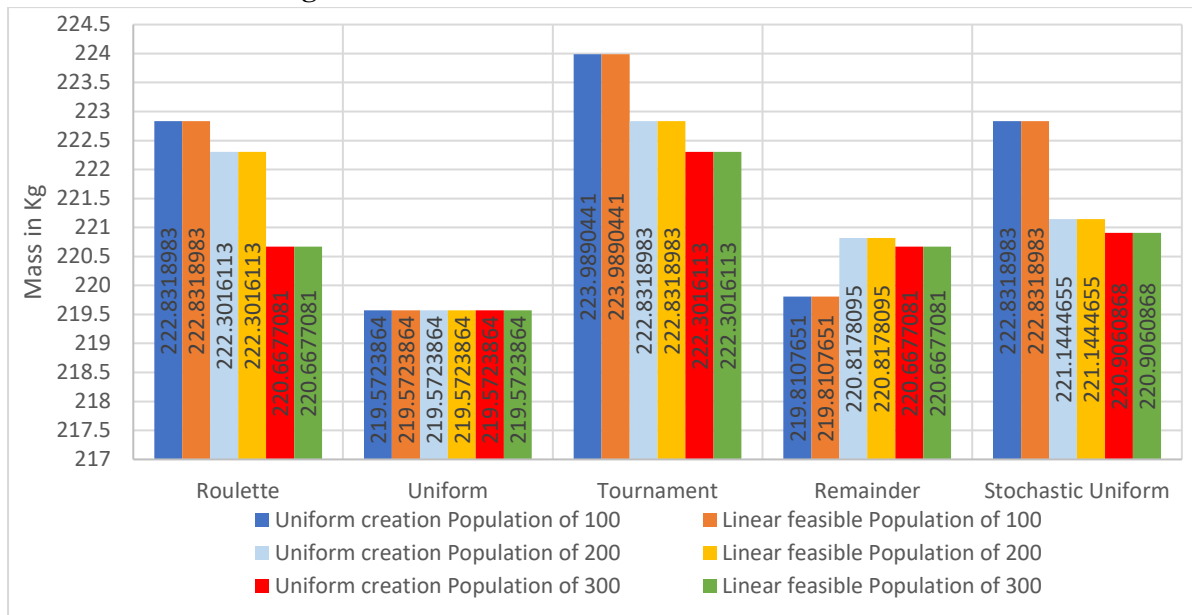


Figure 5: Case C- Minimum Mass Result for different GA Parameters

4.3 Case C:

As shown in Fig. 5, Uniform selection function performs better in this case as well, and the Roulette selection function result shows significant change with population size compared to Case A, and B, which is because of two-point crossover. A similar trend has been observed in the Tournament and Stochastic Uniform selection, and opposite effect with the Remainder selection. But two-point crossover performs better with the Remainder selection, rather than scattered crossover which is shown in Fig. 4.

4.4 Comparison

In all 3 Case, Uniform selection function performance is found to be the global optimum. Roulette selection function gives a better result with the population of 300, for integer creation type, and with the

population of 200 for non-integer creation type, with no effect of creation function and crossover function, but still fail to reach a global minimum. The Tournament selection function works best with the Linear creation function of non-integer type, with 100 population size. Whereas, Remainder selection function gives a result which is very close to global optimum with two-point crossover, and integer creation type of population size 100, the creation function has no effect. Stochastic Uniform performs better with high population size, and scattered crossover, and shows no effect of creation function and creation function type.

4.5 Function Count Result:

This is the count of the objective function evaluated by the Genetic algorithm, and for all three cases shown in Fig 6, 7, and 8, it can be seen that as the population size increases the function count also increases. But the Uniform selection shows a large increase compared to the other, which is the main reason for better results compared to other selection function. By considering the Genetic Algorithm stopping criterion as one more GA parameter, which this study has not considered, there could be an improvement in the result of other selection functions.

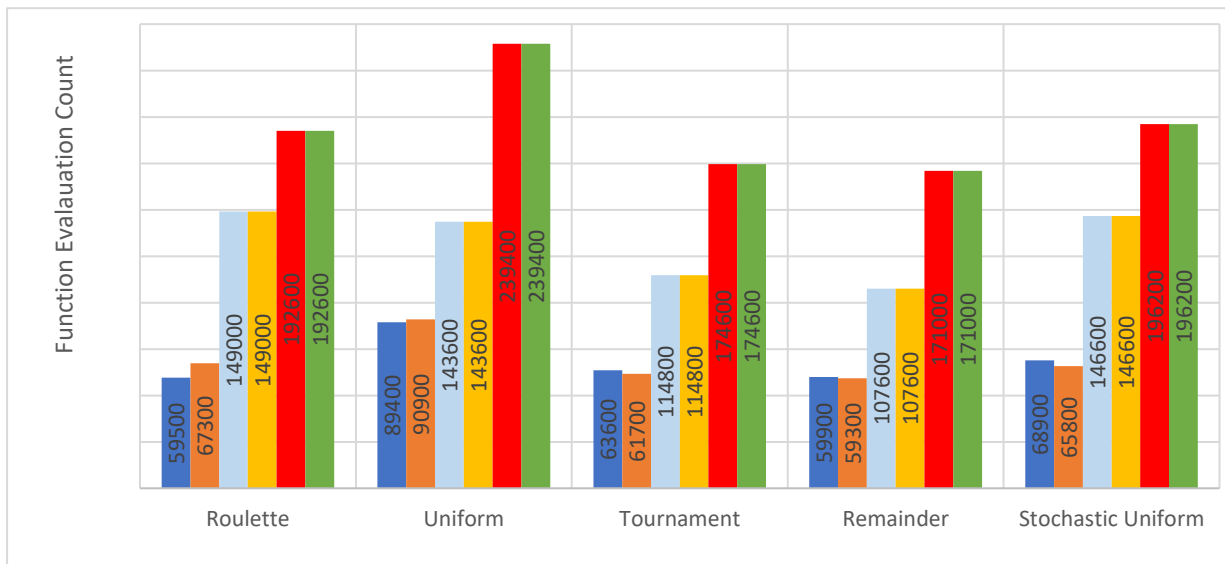


Figure 6: Case A- Function Evaluation Count for different GA Parameters

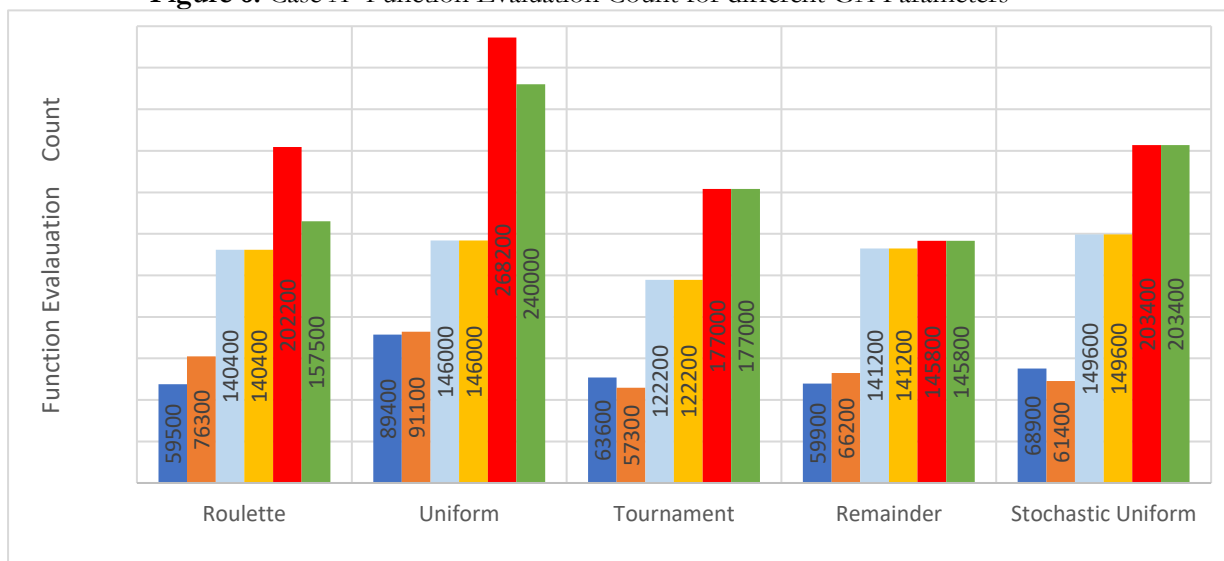


Figure 7: Case B- Function Evaluation Count for different GA Parameters

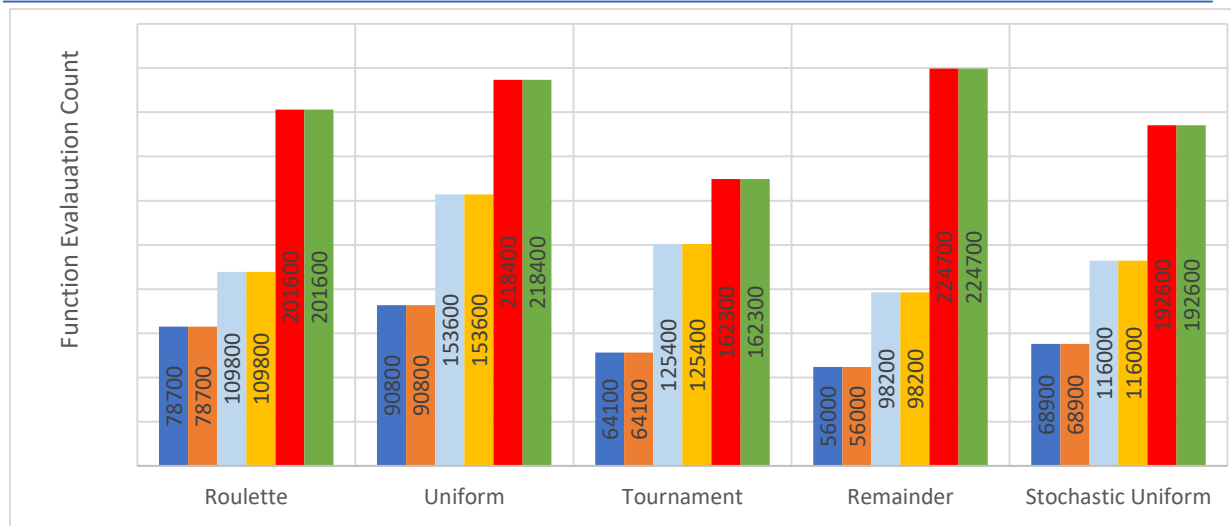


Figure 8: Case C- Function Evaluation Count for different GA Parameters

5 Conclusions

The 25-bar truss with the nonlinear constraint is taken with the objective to minimize the mass and variable being the discrete area of each element. And GA parameters like Selection Function, Population Size, Crossover Function, and Creation Function are varied to find the effect on the optimum result and the function evaluation. It is found that the Uniform selection gives the best result, irrespective of the creation function, population size or crossover function. But this is at the cost of a large number of function evaluations, hence computationally costlier. And the other selection function fails to reach the global optimum and has a smaller number of function evaluation count. If the analysis of selection function is done one at a time, it is seen that all Cases performs better in Roulette but, Case A which is non-integer type with 200 population size being computationally cheaper than Case B and C of population size 300. In the Tournament selection, Case A, B with smaller population size and Case C with higher population size performs better. Case C performs better at Remainder selection with smaller population size, and Case A and B for Stochastic Uniform with higher population size. And, it is clear that the function evaluation count increases with the population size in every Case from this study. It has to be noted that the stopping criteria for all the three cases are same, which can be the cause for the lower function evaluation count for other than Uniform selection function, which leads to the local optimum result, rather than the global optimum. Hence the effect of stopping criteria need to be taken into consideration in future studies.

How to Cite this Article:

A. Kumar and R. K., "Genetic Algorithm Parameter Effect on 3D Truss Optimization with Discrete Variable", *Adv. J. Grad. Res.*, vol. 5, no. 1, pp. 61-70, Dec. 2018. Doi: 10.21467/ajgr.5.1.61-70

Reference

- [1] C. Camp, S. Pezeshk, and G. Cao, "Design of framed structures using a genetic algorithm," in *Proceedings of the US-Japan Joint Seminar on Structural Optimization*, 1997. ISBN:0784402213, 9780784402214
- [2] Rajeev S. and C. S. Krishnamoorthy, "Discrete optimization of structures using genetic algorithms," *J. Struct. Eng.*, vol. 118, no. 5, pp. 1233–1250, 1992.
- [3] V. Toğan and A. T. Daloğlu, "Optimization of 3d trusses with adaptive approach in genetic algorithms," *Eng. Struct.*, vol. 28, no. 7, pp. 1019–1027, 2006.
- [4] T. Dede, S. Bekiroglu, and Y. Ayvaz, "Weight minimization of trusses with genetic algorithm," *Appl. Soft Comput. J.*, vol. 11, no. 2, pp. 2565–2575, 2011.
- [5] 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.
- [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. F. Schutte and A. A. Groenwold, "Sizing design of truss structures using particle swarms," *Struct. Multidiscip. Optim.*, vol. 25, no. 4, pp. 261–269, 2003.
- [8] M. Kripka, "Discrete optimization of trusses by simulated annealing," *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 26, no. 2, pp. 170–173, 2004.
- [9] A. Kaveh, and S. Talatahari, "A hybrid particle swarm and ant colony optimization for design of truss structures," *Asian J. Civ. Eng.*, vol. 9, no. 4, pp. 329–348, 2008.
- [10] A. Kaveh, and S. Talatahari, "Size optimization of space trusses using Big Bang-Big Crunch algorithm," *Comput. Struct.*, vol. 87, no. 17–18, pp. 1129–1140, 2009.
- [11] G. C. Luh and C. Y. Lin, "Optimal design of truss-structures using particle swarm optimization," *Comput. Struct.*, vol. 89, no. 23–24, pp. 2221–2232, 2011.
- [12] Z. El Maskaoui, S. Jalal, and L. Bousshine, "Genetic Algorithm Parameters Effect on the Optimal Structural Design Search," *IOSR J. of Mech and Civ Eng*, vol. 14, no. 3, pp. 124–130, 2017.
- [13] R. P and S. C.R, "Optimal Design of Plane Truss Structures Using Differential Evolution Algorithm," *Jordan J. Civ. Eng.*, vol. 11, no. 1, pp. 91–96, 2017.
- [14] D. Neeraja, T. Kamireddy, P. S. Kumar, and V. S. Reddy, "Weight optimization of plane truss using genetic algorithm," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 263, no. 3, 2017.
- [15] H. Assimi, A. Jamali, and N. Nariman-zadeh, "Sizing and topology optimization of truss structures using genetic programming," *Swarm Evol. Comput.*, vol. 37, pp. 90–103, 2017.
- [16] I. N. Serpik, A. V. Alekseytsev, and P. Y. Balabin, "Mixed approaches to handle limitations and execute mutation in the genetic algorithm for truss size, shape and topology optimization," *Period. Polytech. Civ. Eng.*, vol. 61, no. 3, pp. 471–482, 2017.
- [17] V. R. Kalatjari and M. H. Talebpour, "Optimization of skeletal structure using improved genetic algorithm based on proposed sampling search space idea," *Int. J. Optim. Civil Eng*, vol. 8, no. 3, pp. 415–432, 2018.
- [18] A. Kaveh and S. Shojaee, "Optimal design of skeletal structures using ant colony optimization," *Int. J. Numer. Methods Eng.*, no. October 2006, pp. 1885–1891, 2006.

Publish your research article in AIJR journals-

- ✓ Online Submission and Tracking
- ✓ Peer-Reviewed
- ✓ Rapid decision
- ✓ Immediate Publication after acceptance
- ✓ Articles freely available online
- ✓ Retain full copyright of your article.

Submit your article at journals.aijr.in

Publish your books with AIJR publisher-

- ✓ Publish with ISBN and DOI.
- ✓ Publish Thesis/Dissertation as Monograph.
- ✓ Publish Book Monograph.
- ✓ Publish Edited Volume/ Book.
- ✓ Publish Conference Proceedings
- ✓ Retain full copyright of your books.

Submit your manuscript at books.aijr.org