American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-5, Issue-11, pp-268-272
www.ajer.org
Research Paper Open Access

Comparison between the Performance of GA and PSO in Structural Optimization Problems

Razvan Cazacu[1]
[1](Industrial Engineering and Management Department, Faculty of Engineering / PetruMaior University of TirguMures, Romania)

ABSTRACT: *Genetic Algorithms and Particle Swarm Optimization are two of the most popular heuristic optimization techniques. They belong to the same group of population-based methods and are often regarded as competitors. While there have been previous attempts to compare the two, both methods performances depend heavily on the selection of their parameters. This paper presents a study in which the critical parameters are varied for both techniques and only the best performing sets are compared. The optimization problem chosen as the comparison framework is a benchmark problem in the structural optimization field. The results show that the Genetic Algorithms are generally better than the Particle Swarm Optimization with regard to all performance indicators.*

Keywords: *structural optimization, genetic algorithm, particle swarm optimization, performance, MATLAB*

I. INTRODUCTION

Heuristic optimization is not a new concept. The first Genetic Algorithm (GA) has been developed in 1975 [1], while the Particle Swarm Optimization (PSO) has first been proposed in 1995 [2], but they are considered modern techniques, as shown in [3]. These are two of the most iconic representatives of the most exotic group of optimization techniques, most of them inspired from natural phenomena. They are usually employed for nonlinear problems with large and complex design spaces, or with discontinuous objective functions, problems that are very difficult or impossible to tackle with classic methods.

Both GA and PSO are population-based techniques, working with a group of candidate solutions to the given problem, leading them towards an optimum. GA simulates the natural evolution of species, using bio-evolution mechanisms such as crossover, mutation and selection based on fitness. PSO is based on the social behavior or large groups, such as flying flocks of birds or fish schools.

Structural optimization deals with finding the optimum geometry for a structure that needs to withstand certain loads and has some prescribed boundary conditions. The use of GA [3-9] and PSO [3], [9], [10-12] in structural optimization problems has been an increasingly present preoccupation of researchers in recent years.
There are previous efforts to compare the efficiency of GA and PSO in structural optimization [9], [13-15]. However, there are a number of parameters that strongly influence the behaviour of these methods, their choice being critical to the method's success. This paper proposes a study where these critical parameters are varied over a certain range and only the best performing configurations of both methods are compared for performance.

It is worthy mentioning that certain studies explore the possibility to mix GA and PSO in a single algorithm and thus take advantage of both the evolutionary aspects of GAs and the data exchange capabilities between individuals specific to PSO [14], [15]. As reported by the authors, this integrated approach can lead to more efficient techniques.

II. METHODOLOGY

In both GA and PSO, a group of solutions is randomly generated as the starting point of the algorithm. Each solution in the group (called population) is an individual and is represented by a series of values: the problem parameters.

GA evolves the population over several generations by using specific genetic operators. The most important operator is the mutation, which allows the exploration of the design space in search of fitter individuals. Mutation basically produces random or pseudo-random changes in the individuals. It can be implemented in several ways, but a unified mathematical formulation has been proposed in [16], in the form:

$$x_k^{(t+1)} = x_k^{(t)} + s \cdot f(u,p) \cdot R, \qquad k = 1 \ldots n \tag{1}$$

In equation(1) above, the parameter k selected for mutation in generation $t+1$ changes its value based on 3 values: the sign s which indicates the direction of mutation, $f(u,p)$ which is a function depending on the actual mutation used and R, the feasible range of the parameter.

In PSO, each solution is represented by an individual in a moving group. The group is moving inside the design space in a number of increments, in each step (i) each individual's position (p_i) being updated considering its current speed (v_i), its best known previous position (pO_i) and the groups global known best position (gO_i). A position inside the design space represents a solution, its optimality being assessed with the aid of an objective function.

$$v_i^x = \omega v_i^x + \varphi_p r_p \left(pO_i^x - p_i^x\right) + \varphi_g r_g \left(gO_i^x - p_i^x\right)$$
$$p_i^x = p_i^x + v_i^x \qquad (2)$$

In equation (2) the coefficients r_p and r_g and random numbers in the $(0,1)$ interval. The influence of each component of the speed is weighted with the parameters ω, φ_p, φ_g. Their choice is of critical importance for the effectiveness of the algorithm.

A more detailed theoretical description of both methods is given in [3], [13].

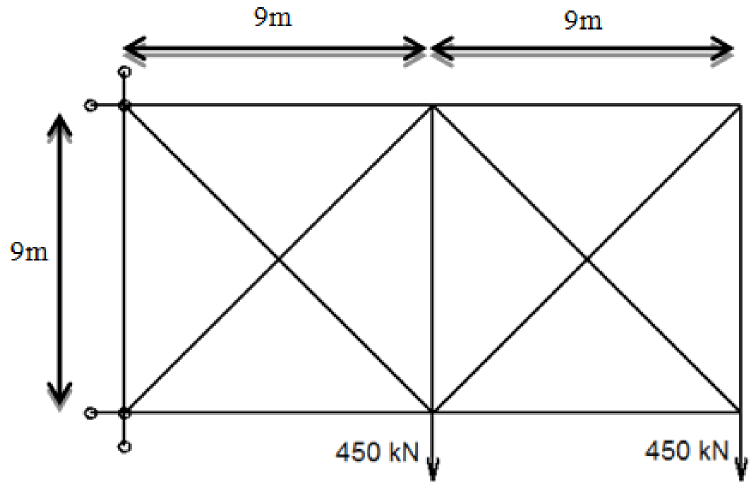

Fig. 1. Benchmark structural optimization problem [13]

The comparative study was performed on the very popular structural optimization benchmark problem described in [5], [9]. It consists in the mass minimization of the aluminum truss depicted in Fig. 1, under the constraints of limit stress (130 MPa) and deformation (50.8 mm).

The optimization tool used for both algorithms is the OOGA MATLAB framework [17], [18]. The framework was developed for the implementation and study of genetic algorithms, but its flexibility and the similarities between the two algorithms allowed the adaptation and implementation of PSO under the same paradigm. Considering OOGA has an object oriented architecture, inherently flexible and extensible, PSO was implemented by extending the key classes of GA, the ones representing the equivalent concepts:

- PSO - main class and start point of the algorithm (inherited from SOGA);
- Particle – class representing a particle (inherited from IndividualW);
- InitializationMixedRndPSO – class for the random initialization of particles in the first iteration (inherited from InitializationBase);
- ReproductionPSO – class responsible for advancing the algorithm over the iterations, calculating the speed and updating positions (inherited from ReproductionBase);
- PSOGraphBestParticleScore – helping class for plotting the evolution of the best solution over iterations (inherited from GraphBase);

For GA, 3 distinct mutation operators were used. The parameter choices and their ranges, for the mutation and the GA itself, are described in [16]. For PSO, the considered population size is 50 and the maximum number of iterations is 51 (including the initial one, randomly generated). These options are consistent with the ones used for the GA implementation.

As stated above, the values of the 3 PSO parameters is crucial for the algorithm success. In order to explore all feasible parameter choices, PSO parameters was configured with the values listed in Table 1.

Table 1. Parameter values for PSO

Parameter	Values				
ω	0.5	0.7	0.8	0.9	1
φ_p	0.5	1	1.5	2	
φ_g	0.5	1	1.5	2	

All parameter combinations were considered, leading to a total of 80 (5x4x4) configurations.
Taking into account that both GA and PSO are stochastic techniques, each configuration was run 50 times to compensate for the inherent randomness of the methods. Thus, the obtained results can be considered statistically reliable.

III. RESULTS

The performance of the algorithms was evaluated using three criteria [16], [19]:

- **reliability** – the average best score of all runs of a given configuration (a measure of the algorithm's capability to reach good solutions consistently);
- **accuracy** – the average best score of the best 20% runs of a given configuration (the algorithm's capability to obtain the best solutions);
- **efficiency** – the total computational time of a given configuration (a measure of how fast the algorithm runs);

As stated above, the GA was considered in 3 variants [16]. As such, there were a total of 4 algorithm choices:

- GA with uniform mutation;
- GA with polynomial mutation;
- GA with Gaussian mutation;
- PSO;

After sorting the results only the best 9 configurations were considered for comparison. Figures 2-4 plot the performance scores for these best 9 configurations of each of the algorithms.

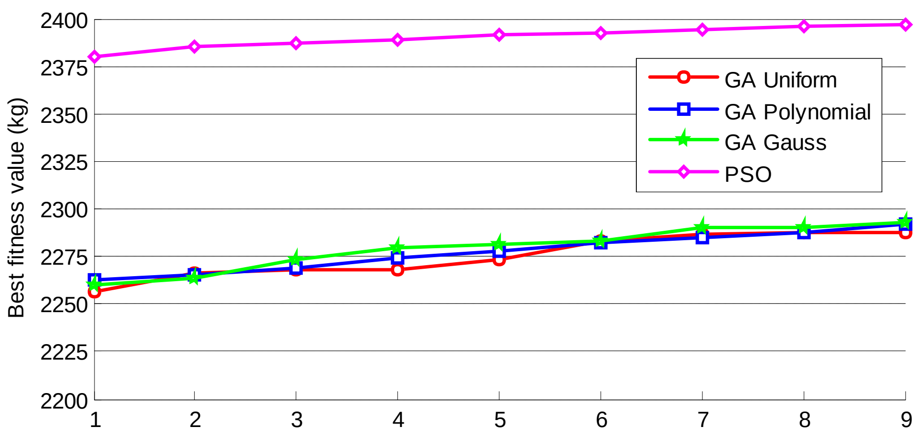

Fig. 2. The average scores of all simulations for the best 9 GA and PSO configurations (reliability)

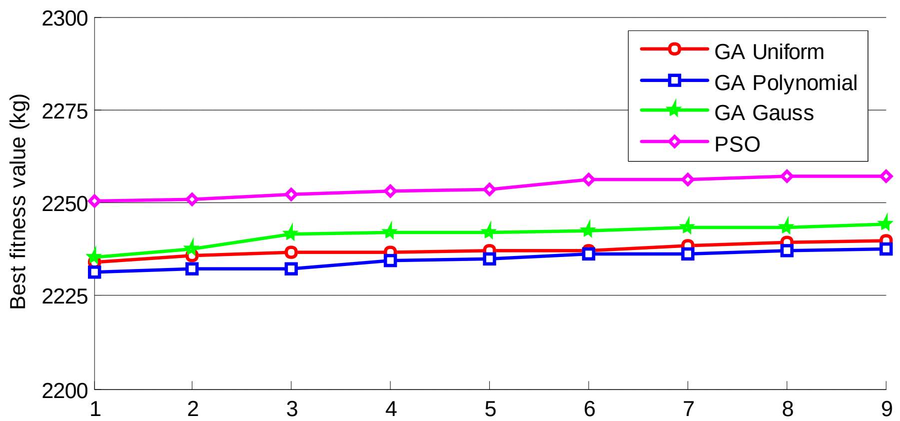

Fig. 3. The average scores of the top 20% simulations for the best 9 GA and PSO configurations (accuracy)

The best fitness scores are the actual masses of the optimum structures while the run times are measured in seconds (total time needed to run all 50 simulations for each configuration).

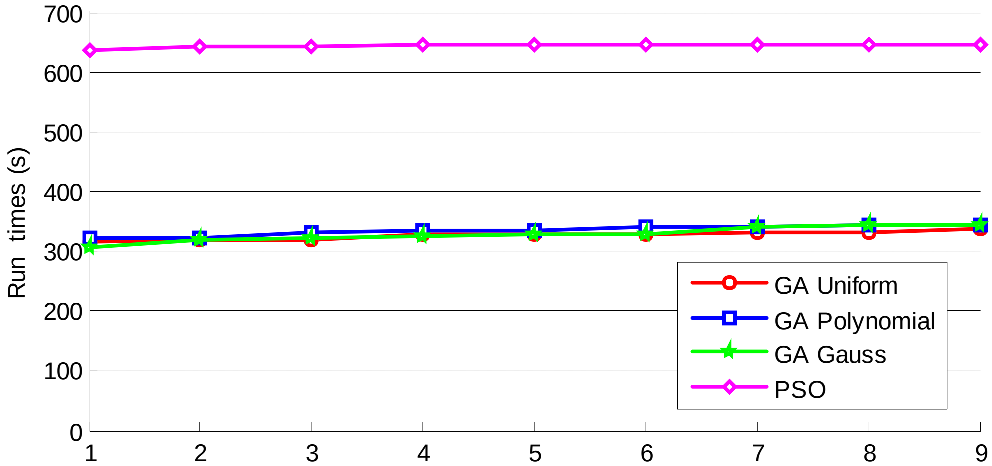

Fig. 4. Total run time for the best 9 GA and PSO configurations

IV. DISCUSSION AND CONCLUSIONS

The graphs plotted in Fig. 2 and Fig. 3 show that for both reliability and accuracy all 3 GA implementations outperform the PSO in all the best configurations. The difference is obvious especially in the case of reliability, the most important efficiency indicator for the situations where the optimization procedure can't be run multiple times due to the high computational effort required by the fitness function evaluation. This is often the case in structural optimization where the fitness is computed using a FEA simulation.

By observing the dynamics of the swarm in the best configurations of the PSO, it was observed that the less fit solutions are due to the fact that the algorithm tends to get stuck in local minima and doesn't have enough kinetic energy to escape them. Higher values of the inertia parameter ω can somehow help the swarm escape local minima but then this fails to properly explore the promising solutions.

By comparing the execution times as well, which show PSO is two times slower than GA, it can be concluded that genetic algorithms are superior to the particle swarm optimization for structural optimization problems, at least in what concerns truss structures. It is also notable that all 3 mutation operators have similar performances, all consistently better that PSO.

REFERENCES

[1] J.H. Holland, *Adaptation in natural and artificial systems* (University of Michigan Press: Michigan, 1975).
[2] J. Kennedy and R. Eberhart, Particle swarm optimization, *Proceedings of the IEEE International Conference on Neural Networks*, 4, 1995, 1942-1948.
[3] R. Cazacu,*Contributions regarding the use of evolutive algorithms for the constructive and technological optimization of prismatic auto parts*, doctoral diss., Universitatea din Oradea, Oradea, 2015.
[4] O. Buiga and C.O. Popa, Optimal mass design of a single-stage helical gear unit with genetic algorithms, *Proceedings of the Romanian Academy, 13(3)*, 2012, 243-250.
[5] R. Cazacu and L. Grama, Steel truss optimization using genetic algorithms and FEA, *Procedia Technology, 12*, 2013, 339–346.
[6] K. Deb and S. Gulati, Design of truss-structures for minimum weight using genetic algorithms, *Finite Elements in Analysis and Design, 37(5)*, 2001, 447-465.
[7] N. Noilublao and S. Bureerat, Simultaneous topology, shape and sizing optimisation of a three-dimensional slender truss tower using multiobjective evolutionary algorithms, *Computers & Structures, 89(23)*, 2011, 2531-2538.
[8] J.E. Rodriguez, A.L. Medaglia and J.P. Casas, Approximation to the optimum design of a motorcycle frame using finite element analysis and evolutionary algorithms, *IEEE Systems and Information Engineering Design Symposium*, 2005, 277-285.
[9] S. Sanchez-Caballero, M.A. Selles, R. Pla-Ferrando, A.V. Martinez Sanz and M.A. Peydro, Recent Advances in Structural Optimization, *Annals of the Oradea University, Fascicle MTE, 11(1)*, 2012, 2.118-2.127.
[10] P.C. Fourie and A.A. Groenwold, The particle swarm optimization algorithm in size and shape optimization, *Structural and Multidisciplinary Optimization, 23(4)*, 2002, 259-267.
[11] R. Perez and K. Behdinan, Particle swarm approach for structural design optimization, *Computers & Structures, 85(19-20)*, 2007, 1579–1588.
[12] S. Saremi, S.M. Mirjalili and S. Mirjalili, Unit Cell Topology Optimization of Line Defect Photonic Crystal Waveguide, *Procedia Technology, 12*, 2014, 174-179.
[13] R. Cazacu and L. Grama, Structural Optimization with Genetic Algorithms and Particle Swarm Optimization, *Annals of the Oradea University, 22(1)*, 2013, 19-22.
[14] R.C. Eberhart and Y. Shi, Comparison between genetic algorithms and particle swarm optimization, *Evolutionary Programming VII Lecture Notes in Computer Science, 1447*, 1998, 611-616.
[15] M.R. Maheri, M. Askarian and S. Shojaee, Size and Topology Optimization of Trusses Using Hybrid Genetic-Particle Swarm Algorithms, *S. Iran J SciTechnol Trans CivEng, 40(3)*, 2016, 179–193.

[16] R. Cazacu, Comparative Study between the Improved Implementation of 3 Classic Mutation Operators for Genetic Algorithms, *Procedia Engineering, 1*, 2016, 1-7.

[17] R. Cazacu, L. Grama and I. Mocian, An OOP MATLAB Extensible Framework for the Implementation of Genetic Algorithms. Part I: The Framework, *Procedia Technology, 19*, 2015, 193-200.

[18] R. Cazacu, L. Grama and I. Mocian, An OOP MATLAB Extensible Framework for the Implementation of Genetic Algorithms. Part II: Case Study, *Procedia Technology, 19*, 2015, 201-206.

[19] P.H. Tang and M.H. Tseng, Adaptive directed mutation for real-coded genetic algorithms, *Applied Soft Computing, 13(1)*, 2013, 600-614.