Blucher Mechanical Engineering Proceedings
May 2014, vol. 1 , num. 1
www.proceedings.blucher.com.br/evento/10wccm

10th World Congress on Computational Mechanics
8-13 July 2012 • São Paulo • Brazil

PERFORMANCE ASSESSMENT OF MODERN HEURISTIC ALGORITHMS USED IN STRUCTURAL OPTIMIZATION

O. Begambre[1]

[1]Programa de Ingeniería Civil, Universidad de Santander - UDES, Bucaramanga Colombia (oscarbegambre@hotmail.com)

Abstract. *In the recent literature, a description of the conditions under which an algorithm can be expected to be successful or fail is not often included in the studies. Because of this, in this work we compare the performance, in terms of precision and stability, of five heuristic algorithms in order to obtain valid statistical results. The problem instance we have used to do the comparison is the optimal weight design of a set of two dimensional steel frames. The new Bacterial Foraging Optimization Algorithm (BFOA), the Bees algorithm (BA), the Particle Swarm Optimization (PSO),the Genetic algorithm (GA) and the Simulated Annealing Algorithm (SAA) were tested. This work also provides an initial assessment in terms of the success rate and quality of the solution.*

Keywords: *Structural Optimization, Bacterial Foraging Optimization Algorithm (BFOA), Bees algorithm (BA), the Particle Swarm Optimization (PSO), Simulated Annealing Algorithm (SAA).*

1. INTRODUCTION

During the last forty years several heuristic algorithms for optimization have been developed. The central idea of these procedures is mimicking some natural process [1, 2, 3] in order to solve real life optimization problems. Though these algorithms have been used to resolve various structural optimization problems [4] a complete study about the relation between the selection of its controlling parameters and it success rate is missing. That study is of worth because not all users of this technology are experts in selecting the parameters (for instance, the rate of mutation and crossover in GA´s or the annealing schedule in SA). In this paper we tested five heuristic algorithms in the real life design of steel frames. In the first part we define the basics of the algorithms. Next, we present the structural design problem followed by a performance assessment of the algorithms in the optimization problem. In order to accomplish this goal, we use the concepts of stability, precision and rate of success.

2. **BASIC OPTIMIZATION ALGORITHMS AND ITS PARAMETERS.**

The principal objective of this section is to identify and select (fine tuning) the parameters that control the performance of the five selected algorithms namely (BFOA), (BA), (PSO), (GA) and (SAA). It is worth noting that these optimization procedures are highly dependent on the choice of several constants that affect the final result quality

2.1. Bacterial Foraging Optimization Algorithm

This optimization procedure was introduced by Passino in 2002 [1] and it mimics the behavior of Escherichia coli colonies. The BFO strategy is based in four processes as presented in [2]. The first is Chemo taxis, that basically summarizes the possible movements that an E. coli can make. This process resembles a swarming behavior (tumbling and swimming). The second is swarming, a type of social behavior which is a mechanism to share information between the bacteria closer to the food and the other members of the colony. As in GA´s, the next steps are reproduction and elimination and dispersal. In the reproduction phase the best set of bacteria is selected from the best chemo tactic stages (similar to the GA´s crossover) and finally, the dispersal and elimination step that could place a new set of bacteria to a new ambient (near the food location for instance) or eliminate it.

Now suppose that the minimum of H (θ) is needed, where $\theta \in R^n$ (θ is an n-dimensional vector of real numbers). The chemotactic step can be defined as [2]:

$$\theta^i(i+1,j,k) = \theta^i(i,j,k) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{1}$$

Where θ^i represents i-th bacterium at j-th chemotactic, k-th re (productive and l-th elimination (dispersal) step and C(i) is the size of the step in a random direction (Δ(i)). The swarming process is defined by [2]

$$J_{cc}(\theta, P(j,k,l) = \sum_{i=1}^{s} J_{cc}(\theta, \theta^i(j,k,l) \tag{2}$$

Where $J_{cc}(\theta, P(j,k,l)$ is the objective function to be added to the actual objective function in order to represent a time varying function and S is the total number of bacteria (for more details see reference [2], then, the last healthy bacteria is dived in two that are placed at the same location keeping the size of the swarm constant and finally bacteria could be killed and replaced, at random, in order to simulate real conditions [2]

2.2. Bees Algorithm (BA)

Since its introduction in 2005 [3] the BA have received great attention by the heuristic community but, similarly to other heuristic optimization procedures, the converge speed of BA is lower than those representative of the most population based algorithms (e.g. PSO and GA´s)

The artificial bee algorithm consists of (as defined in reference [4]): *´´three groups of bees: employed bees, onlookers and scout bees. All bees which are currently exploiting a food* *source are known as employed. Each food source is a possible solution for the problem under consideration and the nectar amount of a food source represents the quality of the solution represented by the fitness value. The number of food sources is the same as the number of employed bees and there is exactly one employed bee for every food source´´.*

The set of equations that control the bees behavior is very similar to the equations used by the classical Particle Swarm Optimization and Genetic algorithms. First, an onlooker bee selects the source using the following probability value (similar to the probability of selection used in genetic algorithms):

$$Prob^{i} = \frac{fobj_i}{\sum_{i=1}^{FS} fobj_i} \tag{3}$$

Where *fobj*$_i$ is the objective function value i; FS is the number of food sources (equal to the number of onlooker bees or employed bees [4]).

Then, the step length or new position for the *i-th* candidate U$_i$ = $U_i = \{u_{i,1}, u_{i,2}, u_{i,3}, \dots, u_{i,p},\}$ is determined from the last position reached Y$_i$ = $Y_i = \{y_{i,1}, y_{i,2}, y_{i,3}, \dots, y_{i,p},\}$ and can be calculated from:

$$u_{i,j} = y_{i,j} + Rand_{i,j}(y_{i,j} - y_{k,j}) \tag{4}$$

Where *p* is the number of variables and *Rand* is a random number in the interval [-1,1]. It is worth noting that equation (3) has the same structure that the expression (6), used in PSO, for update the position of each particle.

In the BA each solution or candidate position is calculated and evaluated and compared to the last position reached. If the new position is of better quality the old is replaced by the recent position else, the old position is maintained. The algorithm finish when a predefined number of iterations is reached. This is an essential factor in defining the BA algorithm.

2.3. The Particle Swarm Optimization (PSO)

The Particle Swarm Optimization Algorithm is a member of the wide category of swarm intelligence methods for solving nonlinear programming problems; the algorithm mimics the social behavior of flocks (for example birds and fishes). It was proposed in 1995 by Kennedy and Eberhart [5]

The particles are manipulated according to the following vectorial equations as stated in reference [5,6]:

$$p_{k+1}^{i} = p_k^i + v_{k+1}^{i} \tag{5}$$

$$v_{k+1}^{i} = v_k^i + C_1 rand_1[b_k^i - p_k^i] + C_{21} rand_2[b_k^g - p_k^i] \tag{6}$$

where k indicates a unit pseudo-time increment. p_k^i represents the position of each particle i (candidate solutions), p_{k+1}^{i} is the position of particle i at time k+1, b_k^i represents the best ever position of particle i at time k (best individual position), b_k^g the best position in the swarm at time k (global best). v_k^i is the velocity of particle i at time k and v_{k+1}^{i} is the update velocity of particle i at time k+1. All vectors in Equations (5) and (6) are of dimensions mx1, where m is the number of optimized parameters. rand$_1$ and rand$_2$ are independent random numbers (with uniform probability) between 0 and 1. The parameters C_1 and C_2 control the flow of information between the current swarm. If $C_2 > C_1$, then the particles puts more trust in the swarm, otherwise, the particle puts more confidence in itself. C_1 and C_2 are known as the cognitive and social parameters respectively. ω is the inertia factor (or inertia weight) that controls the impact of the previous particle velocity on the current particle velocity as explained in [5, 6, 11]. As mentioned above, it it's interesting to compare equation (6) and equation (4) because they have the same function and structure in the algorithms. Finally, the PSO stops if a predefined number of iteration is reached or if a predefined tolerance is attained.

2.4. Genetic Algorithms (GA´s)

The simple GA uses three basic operators [7]: reproduction, selection and crossover and mutation. Reproduction guarantees that individuals are copied according to its function objective value. Crossover and selection assures the union of at least two individuals randomly selected. The information from these individuals is shared in order to obtain new offspring with better fitness than their parents. Finally, the mutation introduces random changes on individuals and helps the algorithm to escape from local optimal.

From the above discussion can be concluded that the performance of any GA strongly depends on the selection of the following three parameters: Mutation probability (MP), Crossover Probability (CP) and Selection strategy (SE). Finally the GA stops when the tolerance or the number of iterations is achieved. The use of the three operators mentioned increase the computational time and have forced the researches to search more efficient optimization techniques. However, in this work, the aim is to establish the rate of success and quality of the solution of each algorithm and for this reason the basic GA (as the others procedures) was selected.

2.5 The Simulated Annealing Algorithm (SAA)

The simulated annealing algorithm is based on the analogy between the simulation of the annealing of solids and the problem of solving large combinatorial optimization problems with discrete and continuous variables [7,8, 11]. The algorithm is essentially an iterative random search procedure that permits uphill moves under the control of a stochastic criterion, avoiding the first local minimum encountered. At the heart of the method is the well-known Metropolis algorithm. The complete procedure can be summarized into four steps:

-Define the objective function F over an n-dimensional continuous variable space

$$F(Y) = F(y_1, y_2, y_{3,} \dots, y_i) \qquad (7)$$

where y_i are the problem variables and i=1…n

-Propose a generator mechanism of random changes in the current configuration (Metropolis criterion).

-At this stage, the control parameter T is lowered according with a decrement rule, known as annealing schedule, of the form:

$$T_{j+1} = \frac{1}{h} T_j \qquad (8)$$

where h is a real valued constant (h>1). With the reduced value of T, a new sequence of points is generated starting from X_{opt} (the current optimum value), until “equilibrium” is reached again (for further details see ref [7]), and finally,

-The procedure is stopped for some small value of T for which no further improvement in F can be expected.

2.6. Basic parameters for the operation of the studied heuristics

From the above discussion, it is evident that each algorithm needs the definition of several parameters in order to achieve an optimum performance. As the performance of the procedures also depends on the problem analyzed, we decide to test the five algorithms in a structural design problem, as described in section 3 and figure 1.

3. OPTIMAL COST / WEIGHT DESIGN OF TWO DIMENSIONAL STEEL FRAME: NUMERICAL EXAMPLE.

The structure to be optimized is the steel frame presented in reference [10]. In this study only linear analysis are carried out, therefore the results obtained here are compared to the results labeled in ref [10] as L (Linear). The problem is to determine the combination of steel sections and connections that render the minimum cost (and the weigh associated) for the structure. For this example we used 100 different steel profiles and we used the same 25 variables as in reference [10]. Here we used the same initial conditions and bounds (as defined in [10]) for the variables in order to control the drift, the global and local effects in the steel members and other problem restrictions. As mentioned above, a first-order analysis, with rigid connection was carried out for live and death loads. In this case, the design variables are not continuous, because the steel members have predefined discrete dimension. Details of the structures can be seen in figure 1. Seismic and nonlinear analyses are out of the scope of this work.

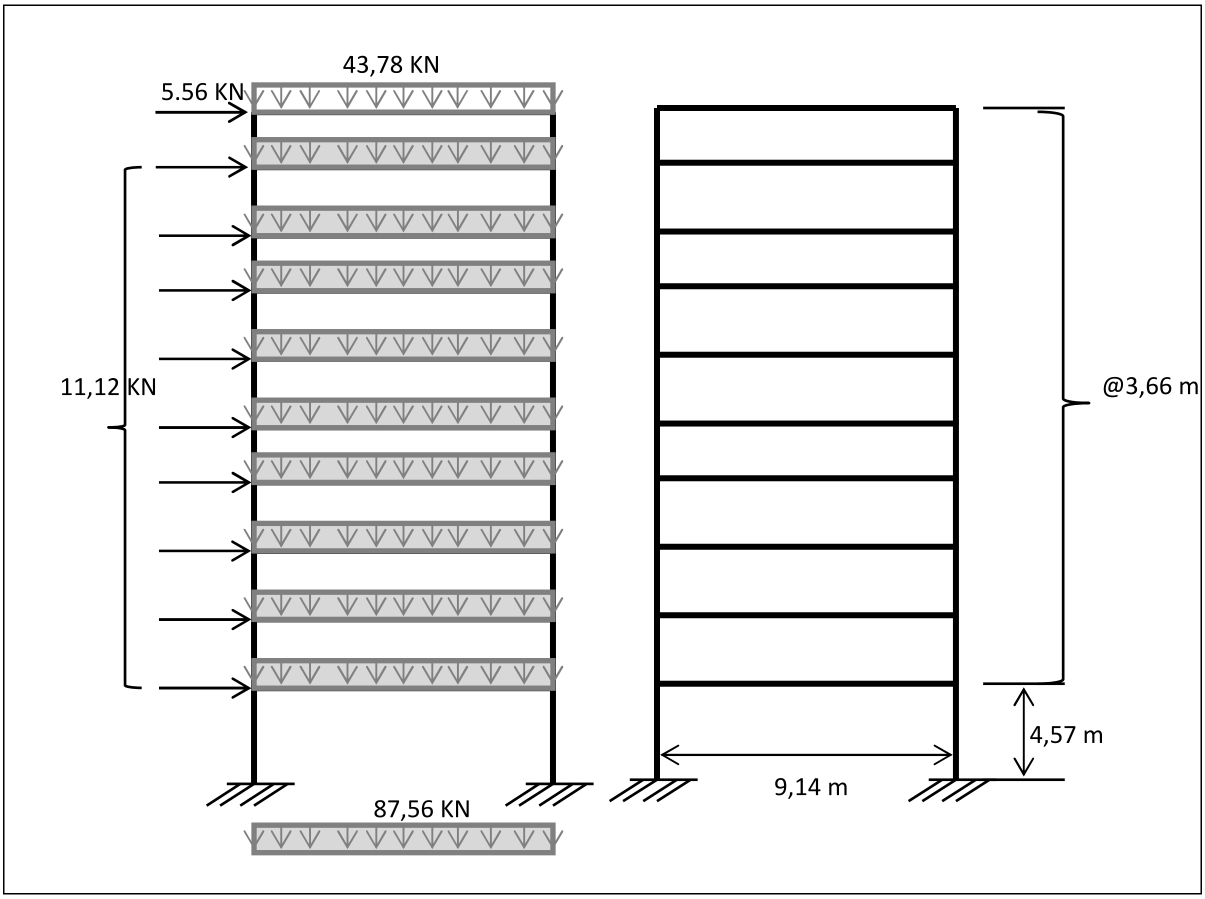

Figure 1. Steel frame studied. Taken from reference [10].

Finally, as the heuristic methods are random by nature, one hundred runs of each algorithm were carried out in order to assess the performance of the algorithms. The results are reported in the next section.

3.1 Heuristic Parameters Selection.

For the numerical example, in table 1 the best parameters found are presented. They were selected after a trial and error study. It it’s worth noting that if another set of parameters is selected the algorithm will fail completely in locating the optimal solution.

Table 1. Heuristic parameter selected for the optimization problem.

Algorithm	Heuristic parameter							
BFO	search space	Population	Number of chemotactic steps	Swimming length	reproduction steps	elimination steps	Elimination probability	C(i)
	9	10	5	5	5	5	0.2	random
BA	search space (p)	food sources (FS)	Maximum number of iteration	tolerance				
	3	10	10	1E-1				
PSO	search space	Initial population	Inertia factor w	Cognitive constant	Social constant	Maximum number of iteration	tolerance	
	9	10	0.7	1.50	2.50	1000	1E-1	
GA	search space	Initial population	Mutation probability	Selection Strategy	Crossover probability	tolerance		
	9	10	0.1	roulette	0.7	1E-1		
SAA	search space	Initial temperature	Annealing schedule, h	Number of cycles before temperature reduction	Stop Temperature	Acceptance criterion		
	9	100	0.8	5	1E-3	Metropolis		

3.2 Numerical Results.

In this section the final weight of the structure was calculated using the five optimization algorithms. In order to evaluate them, the principal statistic parameters were determined. The results can be observed in table 2.

Table 2. Results from 100 runs on the structural design problem of reference [10].

Algorithm tested	Best Weight (Kg)	Worst Weight (Kg)	Mean Weight (Kg)	Standard deviation	Success rate
BFO	29050	29059	29054	4.15	70
BA	28870	28879	28874	4.41	80
PSO	29051	29060	29058	2.54	75
GA	29450	29480	29465	2.85	70
SAA	28860	28865	28862	0.74	92
REF [10]	28860	NR	NR	NR	NR

The algorithm´s stability was measured by its standard deviation. When standard deviation is closer to zero the algorithm can be classified as stable following the procedure defined

I reference [11]. The algorithm´s precision is defined by its mean value. When the mean value is closer to the reference value the algorithm cam be classified as accurate [11]. From table 2 we can conclude that the SAA has the higher precision and stability. Taking as reference the algorithm with the minimum value of standard deviation and the best function value (in this case, the SAA) the success rate was defined as the number of times out of 100 the algorithm reaches an objective function value equals to the standard deviation +/- the mean value (of the reference algorithm). As can be seen from table 2, again the SAA has the better solution. In terms of the above defined parameters we conclude that the quality of the solution depends on the stability, precision and success rate, and in this case, the GA has the solution with less quality when compared to SAA.

4. CONCLUDING REMARKS

A performance test of five optimization algorithms based on computational intelligence was presented. The results indicate that the SAA has high precision and stability than the BFO, BA, PSO and GA. These results strongly depend on the heuristic parameter selection. In this context, it is worth mentioning that the common user of this technology could select the wrong set of constant and therefore the quality of its solution will be poor.

On the other hand, the study presented was based on the linear analysis of a steel frame. The nonlinear behavior under seismic excitation is currently under study by the author. As the complexity of the problem increases (dynamic problem) the optimization solvers must be improved in order to obtain the quality of the solutions.

Acknowledgments

The author wishes to thanks to the University of Santander UDES for the financial support to this work.

5. REFERENCES

[1] K.M. Passino. “Biomimicry of bacterial foraging for distributed optimization and control”. *IEEE Control Systems Magazine*. 52–67, 2002.

[2] S. Das, A. Biswa, S. Dasgupta, and A. Abraham. “Bacterial Foraging Optimization Algorithm: Theoretical Foundations”. *Analysis, and Applications.* tutorial.softcomputing.net/bfoa-chapter.pdf. Accessed 10/01/2012.

[3] D. Karaboga, “An idea based on honey bee swarm for numerical optimization”. *Erciyes University, Technical Report*-TR06, Kayseri, Turkey, 2005.

[4] W. Gao, S. Liu, L. Huang. “A global best artificial bee colony algorithm for global optimization”. *Journal of Computational and Applied Mathematics*. 236, 2741–2753, 2012.

[5] J. Kennedy, R. Eberhart. “Particle swarm optimization”. *Proc. IEEE Int. Conf. Neural Networks.* 4, 1942-1948, 1995.

[6] Y. Shi, R.C. Eberhart. “Parameter selection in particle swarm optimization”. Evolutionary programming VII, Lecture notes in computer science. 1447, 591-600, In: *Porto, V. Saravanan, N., Waagen, D., Eiben, A. (Eds).* Berlin: Springer, 1998.

[7] R. Gabor, E. Aniko. “Genetic algorithms in computer aided design”. *Computer-Aided Desing.* 35 , 709-726, 2003.

[8] Kirkpatrick, S., Gelatt, A., Vecchi, M.P. “Optimization by simulated annealing”. *Science*. 220, 671-680, 1983.

[9] Cerny, V. “Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm”. *J. Opt. Theory Appl.* 45, 41-51, 1985.

[10] G. Sanchez, P. Martinez. “Minimum Cost Design with Advanced Analysis for Elastic Planar Steel Frames”. In: *Proceedings of The Eighth International Conference on Computational Structures Technology. B.H.V. Topping, G. Montero and R. Montenegro* (Editors). Civil-Comp Press, Stirlingshire, UK, Paper 120, 2006.

[11] O. Begambre, J.E. Laier. “A hybrid particle swarm optimization-simplex algorithm (PSOS) for structural damage identification”. *Advances in Engineering Software*. 40, 883-891, 2009.