6th World Congresses of Structural and Multidisciplinary Optimization
Rio de Janeiro, 30 May - 03 June 2005, Brazil

Discrete Optimal Design of Trusses by Generalized Extremal Optimization

Fabiano Luis de Sousa and Walter Kenkiti Takahashi

Instituto Nacional de Pesquisas Espaciais, Av. dos Astronautas 1758, São José dos Campos, SP, 12227-010, Brasil
fabiano@dem.inpe.br, kenkiti@dem.inpe.br

Abstract
In this paper the first results of the application of the Generalized Extremal Optimization (GEO) algorithm to a discrete structural optimization problem is shown. GEO is an evolutionary brand new algorithm, devised to be easily applicable to a broad class of nonlinear constrained optimization problems, with the presence of any combination of continuos, discrete and integer variables. So far, it has been applied successfully to real optimal design problems with continuos design variables and shown to be competitive to other stochastic methods such as the Genetic Algorithms (GAs) and the Simulated Annealing (SA). Having only one free parameter to adjust, it can be easily set to give its best performance for a given application. This is an *a priori* advantage over methods such as the SA and GAs since each of them have at least three parameters to be set, making their tuning to a particular application more prone to be computationally expensive and becoming a problem in itself. In this work, the 10-bar truss problem is used as a test case and the performance of GEO, compared to results from other methods available in literature.
Keywords: Structural Optimization, Truss Optimization, Evolutionary Algorithms, Generalized Extremal Optimization.

1. Introduction
Stochastic algorithms inspired by natural phenomena have been increasingly used to tackle optimization problems. The motivation behind this trend may be the observation that, either to save energy, reduce waste or produce fitter individuals, nature has “developed” robust, self-regulating mechanisms, that tend to produce efficient solutions for complex problems. Other motivations are that these methods are usually easy to implement and are very robust to complex features of the design space, like multiple local optima. Simulated Annealing (SA) [1] and Evolutionary Algorithms (EAs) [2], particularly the Genetic Algorithms (GAs), are probably the most used of such methods, having being applied in many areas of engineering and science.

Recently, a new evolutionary algorithm was proposed. Called Generalized Extremal Optimization (GEO) [3, 4, 5], it was originally developed as an improvement of the Extremal Optimization (EO) method [6], which was inspired by the evolutionary model of Bak-Sneppen [7]. GEO was devised to be easily applicable to a broad class of nonlinear constrained optimization problems, with the presence of any combination of continuos, discrete and integer variables. So far, it has been applied successfully to real optimal design problems with continuos design variables [4, 8, 9, 10], and shown to be competitive to other stochastic methods in test functions [3, 4]. Having only one free parameter to adjust, it can be easily set to give its best performance for a given application. This is an *a priori* advantage over methods such as the SA and GA since each of them have at least three parameters to be set, making their tuning to a particular application more prone to be computationally expensive and becoming a problem in itself.

Being a brand new algorithm, many features of GEO are still to be explored, including the assessment of its performance on different types of design spaces. In this context, this work brings the first results of the application of GEO to a discrete structural optimization problem. It has been argued, that such kind of problems would be more efficiently tackled by methods such as GAs and SA since they would deal directly with the discrete variables, while the design space has to be treated as continuos when using traditional gradient based methods [11, 12, 13, 14]. The 10-bar truss problem is used here as a test case to assess the performance of GEO, compared to results from other methods available in literature.

In the following Sections a brief description of the SA and GA is made followed by a detailed explanation of GEO, the results from the 10-bar truss test case and the conclusions.

2. Simulating Annealing and Genetic Algorithms
Probably the most known and used stochastic optimization methods inspired by nature are the Simulated Annealing and Genetic Algorithms. They have been used for tackling structural optimization problems and here we compared some of their results for the 10-bar truss problem found in literature, with the ones from GEO. A very brief description of the canonical implementations of the SA and GA is provided below. More detailed information on them can be found in [2, 15, 16].

2.1 Simulated Annealing
The SA method was proposed by Kirkpatrick et al [15] at the early 1980’s. It is based on the Metropolis algorithm [17] that was developed to simulate a collection of atoms in equilibrium at a given temperature (T). Submitting one of such atoms to a small random displacement, the variation ΔE of the system energy is calculated and if $\Delta E \leq 0$ the move is accepted. Otherwise, the move is accepted with probability:

$$P(\Delta E) = \exp\left(\frac{-\Delta E}{k_B T}\right) \tag{1}$$

where, K_B is the Boltzmann’s constant.

Kirkpatrick et al [15] adapted this procedure to an optimization algorithm making the variation in energy be a variation in the value of the objective function. Moreover, they introduced a schedule to the temperature. Starting from a given point in the design space and setting an initial temperature T_0, a random perturbation is applied to the design variables. If the new solution implies a decrease on the objective function it is accepted. If not, the new design is accepted with probability proportional to (1). This process is repeated for some iterations and then the temperature is decreased. The procedure is repeated following a temperature schedule (annealing) until a given stopping criterion is met. The probability of moves that increase the value of the objective function is controlled by T and is usually high at the beginning of the search. As $T \to 0$ only moves that decrease the objective value are accepted and the search becomes deterministic. Adjustable parameters of the SA are the temperature schedule, how the design variables are changed, the number of iterations at a given temperature level and the acceptance probability distribution.

2.2 Genetic Algorithms

Genetic Algorithms are part of a more general category of methods called Evolutionary Algorithms (EAs) [2]. The functioning of these algorithms is based on Darwin's theory of survival of the fitness. Beginning with a population of individuals (solutions in the design space), operations of selection, reproduction and mutation are applied through a given number of generations such that the average population fitness, based on the value of the objective function, is consistently improved. In the Simple Genetic Algorithm (SGA), each individual is coded in a binary string. The fitter individuals are probabilistically selected for reproduction and "mate" by exchanging bits of each other strings, in a process called crossover. A mutation operation is applied to the resulting offspring and the new generation is formed. This process is repeated for a given number of generations. Parameters to be set in a typical GA are the size of the population, the selection procedure, the probability and scheme of crossover and mutation, and the number of generations.

3. The Generalized Extremal Optimization Algorithm

The theory of Self-Organized Criticality (SOC) has been used to explain the power law signatures that emerge from many complex systems in such different areas as geology, economy and biology [18]. It states that large interactive systems evolve naturally to a critical state where a single change in one of its elements generates "avalanches" that can reach any number of elements in the system. The probability distribution of the sizes "s" of these avalanches, is described by a power law in the form:

$$P(s) \approx s^{-\hat{o}} \tag{2}$$

A simple model for a self-organized system is the sand pile [18]. After some time adding up grains of sand, the pile reaches a critical slope and from that on even a small perturbation (a single grain) could cause avalanches that may reach, in a probabilistic sense, the whole pile. After the avalanche, the system would then recover to the critical state. Bak and Sneppen [7] suggested that a similar dynamic behavior could explain the bursts of evolutionary activity observed in the fossil record and that has been given the name of punctuated equilibrium [19]. In the Bak-Sneppen model, M species $\mathbf{e}_i$ (i = 1, M) are represented in a lattice with periodic boundary conditions and, for each of them, is assigned randomly a fitness number in the range [0, 1]. The evolution is simulated forcing the least adapted species, the one with the least fitness, and their side neighbors, to change (it can "evolve" or be "extinct" and replaced by a new one, that not necessarily has a better fitness). This is done by assigning new fitness numbers, randomly, to these species. Since the less adapted species are constantly forced to change, the average fitness value of the ecosystem increases and, eventually, some time after initialization all species have a fitness number above a "critical level". However, as even good species may be forced to change (if they are neighbors of the least adapted one), it happens that a number of species may fall below the critical level from time to time. That is, the equilibrium (being above the critical level in "stasis") of one or more species is punctuated by avalanches, whose occurrence is described by a power law [7]. Although the claim that natural evolution may happen in a system that is critically self-organized has been controversial [20, 21], an optimization heuristic based on the Bak-Sneppen model may evolve solutions quickly, systematically mutating the worst individuals, while preserving throughout the search process the possibility of probing different regions of the design space (via avalanches).

Inspired by the Bak-Sneppen model, Boettcher and Percus developed the EO method [6], which has been applied successfully to combinatorial optimization problems in which a fitness number is associated with the design variables. However, as they pointed out, in some cases this may become an ambiguous or even impossible task [6]. GEO was devised to overcome this problem, in a way that it could be easily applicable to a broad class of optimization problems, with any kind of design variable, either continuous, discrete or a combination of them, on a design space that may be multimodal or even discontinuous and subject to any kind of constraints.

In the canonical GEO the species are represented by a string of L bits that encodes N design variables. That is, each bit is considered a species. For each of them is associated a fitness number that is proportional to the gain (or loss) the objective function value has in mutating (flipping) the bit. All bits are then ranked from 1, for the least adapted bit, to L for the best adapted. A bit is then mutated according to the probability distribution $P \approx k^{-\tau}$, where k is the rank of a selected bit candidate to mutate, and τ is a free control parameter. If $\tau \to 0$ any bit has the same probability to be mutated, whereas for $\tau \to \infty$, only the least adapted bit will be mutated. It has been observed that the best value of τ for a given application (τ_{best}, i.e., the one that yields the best performance of the algorithm on the application at hand) generally lies within the range [1, 10], which makes the setting of τ a relatively easy task.

On a variation of the canonical GEO, one bit per variable is mutated at each iteration of the algorithm. In this case the ranking is done separately for each design variable. This approach has shown to be more efficient than the canonical GEO for problems that has only bound constraints [3, 4].

The bit encoding can accommodate any type of design variable and for GEO this could be done as described in [4]. Nevertheless, GEO may also work with other kinds of encoding. In fact, for the truss problem, we worked with the design variables directly, treating each variable as a species. In Figure 1 is shown the population of species as represented in the Bak-Sneppen model, in the canonical GEO and on the truss problem.

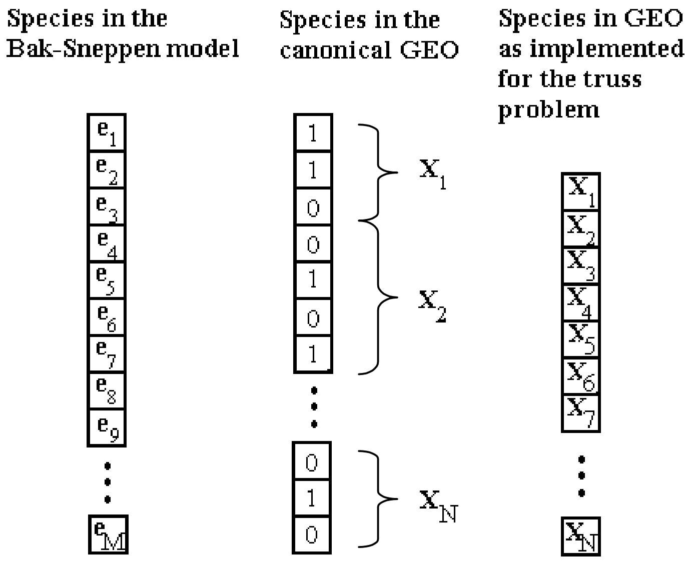

Figure 1. Population of species in the Bak-Sneppen model, in the canonical GEO and in GEO as implemented for the truss problem.

As shown in Figure 1, each design variables in the canonical GEO may be encoded with a different number of bits. In fact, the number of bits necessary for encoding each variable will be dictated by the precision desired for it. GEO has been applied successfully for optimal design problems with continuos variables using binary coding [4, 8, 9, 10]. Although a binary coding could also be used for the truss problem, we decided to tackle the problem using the variables directly. The modification on the canonical algorithm was minimum, as can be seen on Figure 2.

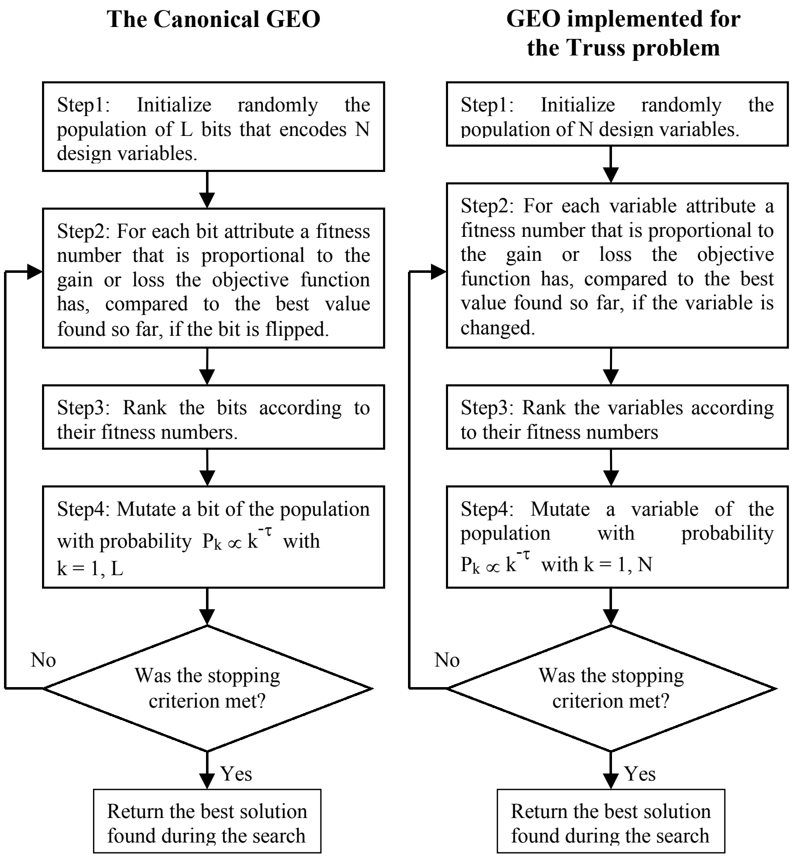

Figure 2. Pseudo-codes for the canonical GEO and as implemented for the truss problem.

In Step 2 the fitness attributed to each bit (or variable in the truss problem) is given by $\Delta V_i = (V_i - V_{best})$, where V_i is the value of the objective function if the bit i is flipped and V_{best} is the best value of the objective function found so far. Note that in this Step, after the value of ΔV_i is calculated, the bit is returned to its original value. For the truss problem the change in the value of the variables in this step is done randomly. In Step 3 the bits are ranked from k = 1, for the least adapted one (for a minimization problem, the one with the least value of ΔV_i), to k = L (or N) for the best adapted one. In Step 4 a candidate bit is chosen randomly to mutate and the value of P_k calculated (k is the value of the rank of the chosen candidate bit - variable). If $P_k \geq$ RAN (a randomly generated number in the interval [0, 1]), then the bit is accepted to mutate. This process is repeated until a bit is confirmed to mutate. Note that in the truss implementation of GEO, the mutation of the candidate variable is done to the value randomly generated in Step 2 (for the bit encoding this is done automatically as the bit is flipped).

Constraints are easily taken into account in GEO. In the canonical algorithm, boundary constraints are incorporated directly by the binary encoding. The discrete coding also takes into account the boundary constraints directly, since only the discrete variable values are available to be used. In the canonical GEO, inequality and equality constraints are taken into account by assigning a high fitness for the bit that, when flipped, leads the algorithm to an unfeasible region of the design space. Note that this move is not prohibit, it only has a low probability to happen. In fact, the algorithm can even be initialized from an unfeasible design. The same approach was used for the truss problem, i.e., a high fitness value was assigned to the variable that when changed in Step 2, lead the algorithm to an unfeasible design.

4. Formulation of the Truss Optimization Problem

The standard 10-bar planar truss consists of 10-hinged bars with the elements connected as shown in Figure 3. The translation of hinges 5 and 6 is constrained in the plane of the truss and it is subjected to two vertical loads applied at hinges 2 and 4. The main goal of the problem is to minimize the mass of the truss (M), varying only the cross sections of the truss members, as long as the geometry of the truss cannot be altered.

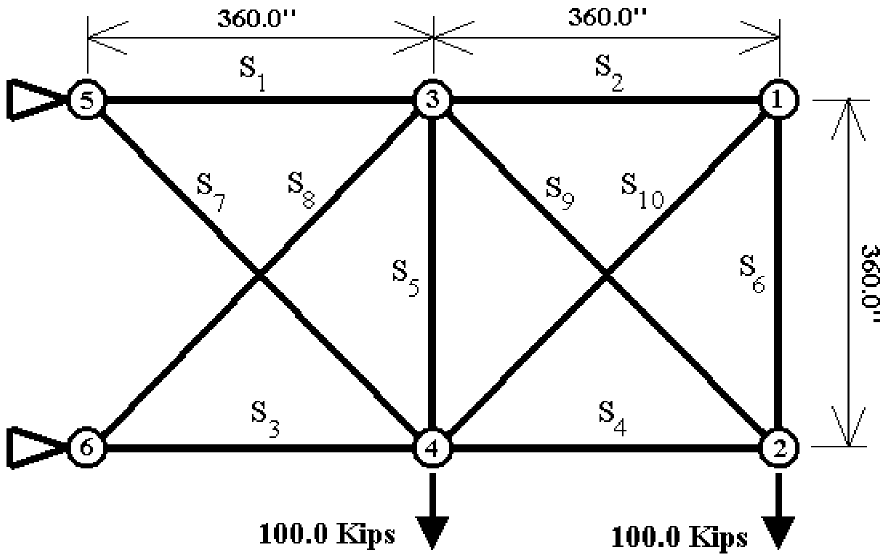

Figure 3 – Geometry of the Standard 10-Bar Truss.

Forty-two cross-sectional areas were taken from AISC (American Institute for Steel Construction) Manual to be used as possible discrete values the truss members may take. These values are presented in Table 1.

Table 1. Possible Bar Cross Section Areas S_i (in^2).

1.62	1.8	1.99	2.13	2.38	2.62	2.63	2.88	2.93
3.09	3.13	3.38	3.47	3.55	3.63	3.84	3.87	3.88
4.18	4.22	4.49	4.59	4.8	4.97	5.12	5.74	7.22
7.97	11.5	13.5	13.9	14.2	15.5	16.00	16.9	18.8
19.9	22.00	22.9	26.5	30.00	33.5			

The mechanical properties of the bars' material are:

E = 10,000 ksi,
ρ = 0.1 lb/in^3 and
Maximum Allowable Stress (σ_A) = ±25 ksi.

The maximum allowable truss displacement (u_A) is 2 in. Due to the small number of elements, the Method of Consistent Deformations was used to determine the axial forces acting in the truss bars and the displacement of the hinges.

The optimization problem is formulated as:

Minimize

$$W = \sum_{i=1}^{10} \rho S_i L_i \,, \tag{3}$$

where S_i and L_i are the cross section area and the length of the *ith* truss bar, respectively.

Subject to

$$\frac{\sigma_i}{\sigma_A} - 1 \le 0 \quad \text{and} \quad \frac{u_j}{u_A} - 1 \le 0 \,, \tag{4}$$

where σ_i (i=1,…10) is the stress of the *ith* bar and u_j (j=1…6) is the displacement *jth* hinge.

5. Results

The efficiency of GEO on tackling the 10-bar truss problem was compared with 7 other results taken from literature, including the one that, as far as we know, is the best result found so far for this problem.

As mentioned in Section 3, the efficiency of GEO on a given application is influenced by the proper setting of τ. A strategy that has shown to be efficient in finding the value of τ_{best} is to run the algorithm a few times, for a fewer number of function evaluations that the one intended for the main runs at different values of τ, and chose the value that yields the best average results. This procedure was done for the 10-bar truss problem. The range and steps on the values of τ chosen for the search of τ_{best} ([0.75, 8.00], 0.25), was based on our previous experience with the algorithm. At each value of τ, 20 runs were performed, each one from a randomly chosen point in the design space. Due to the stochastic nature of the algorithm, some runs did not find a feasible design after the stopping criterion was reached (10000 function evaluations per run) on the search for τ_{best}. Hence, a high value of mass (100000 lbs) was assigned to the runs that end up in unfeasible designs, penalizing the values of τ that were less efficient in leading the search to feasible designs. Results for the search of τ_{best} are shown in Figure 4.

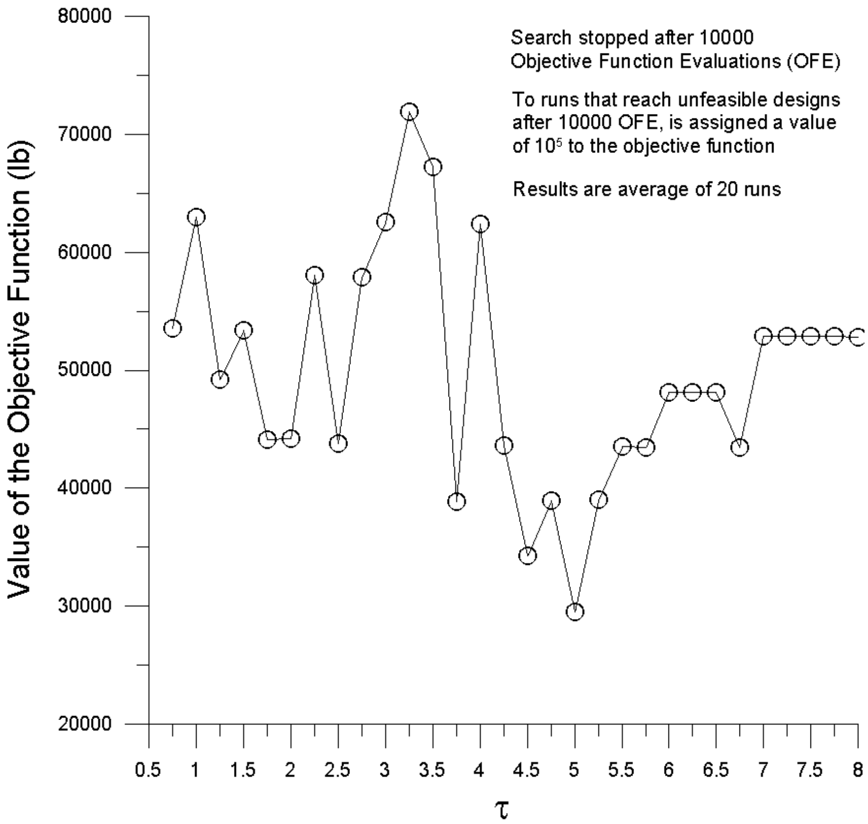

Figure 4. Average of objective function values found in 20 runs of GEO as a function of τ.

From Figure 4 it can be seen that better results are obtained with $\tau = 5.00$, and this value was used in the search for the best truss mass using GEO. Twenty runs were performed from randomly chosen points in the design space, and each run stopped after 6×10^5 function evaluations (this stopping criteria was the same used in reference [13] for a GA with real coding). The best result from these 20 runs is shown in Table 2, together with results from other methods.

Table2. Best result found for the truss problem from different references.

Reference	Truss Mass (lbs)	Sections (in^2)									
		S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
[22]	5491.71	33.50	1.62	22.90	15.50	1.62	1.62	7.97	22.00	22.00	1.62
[23]	5491.71	33.50	1.62	22.90	15.50	1.62	1.62	7.97	22.00	22.00	1.62
[11]	5613.84	33.50	1.62	22.00	15.50	1.62	1.62	14.20	19.90	19.90	2.62
[12]	5586.59	30.00	1.62	22.90	13.50	1.62	1.62	13.90	22.00	22.00	1.62
[24]	5528.09	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
[13]	5491.71	33.50	1.62	22.90	15.50	1.62	1.62	7.97	22.00	22.00	1.62
[14]	5490.74	33.50	1.62	22.90	14.20	1.62	1.62	7.97	22.90	22.00	1.62
GEO	**5525.04**	**33.50**	**2.13**	**22.00**	**13.90**	**1.62**	**1.99**	**7.97**	**22.90**	**22.90**	**1.62**

n/a: not available.

In Table 2 references [22] and [23] used deterministic methods specially devised for this type of problem. References [11, 12, 13, 24] used different implementations of Genetic Algorithms and reference [14] used the Simulated Annealing. While the result from GEO is not the best, it did performed better than improved implementations of the Simple Genetic Algorithm (References [11, 12, 24]) and got close to the results of the floating point coded GA [13] and the SA [14], which got the best result. All runs found feasible designs and the average of the best values found on them was 5547.93, with standard deviation of 14.37. The evolution of the best value found for each GEO run, as a function of the number of evaluations is shown in Figure 5. It can be seen the "anytime behavior" [2] typical of EAs, i.e., major improvements on the objective function taking place in the beginning of the search.

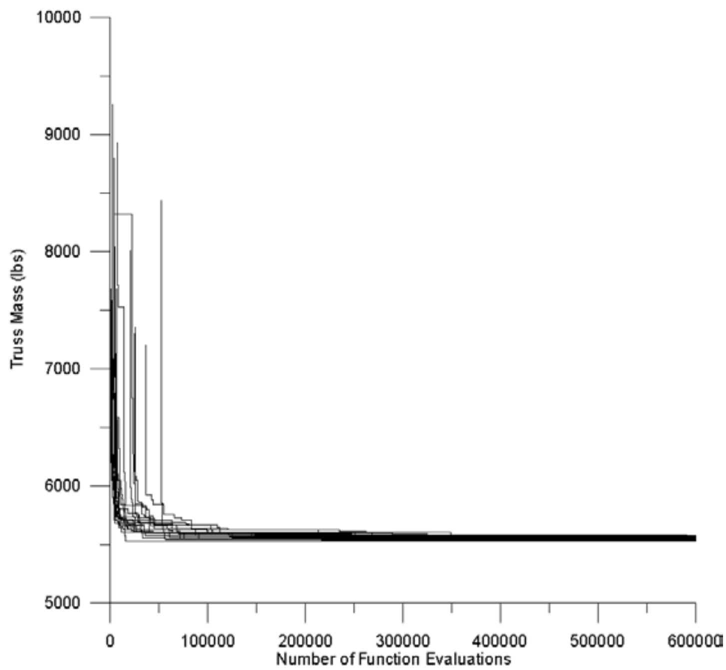

Figure 5. Best value of the objective function as a function of the number of evaluations, for 20 runs of GEO.

6. Conclusions

The results of this preliminary study indicate that GEO can be a competitive alternative to popular meta-heuristics such as GAs and SA for discrete structural optimization problems. In fact, GEO has been recently developed and improvements on the algorithm are an ongoing area of research. Some topics of current investigation are the influence of the variation of τ during the search, parallelization of the algorithm and hybridization with other methods. A more detailed analysis of the efficiency of GEO in solving structural optimization problems is also being carried out at the moment.

7. References

1. S. Kirkpatrick, C.D. Gellat and M.P. Vecchi, Optimization by simulated annealing, *Science*, 1983, Vol. 220, pp. 671-680.
2. Eiben, A.E. and Smith J.E. Introduction to Evolutionary Computing. Springer, Berlin, 2003.
3. Sousa, F. L.; Ramos, F. M., Function Optimization Using Extremal Dynamics, *Proceedings of the* 4^{th} *International Conference on Inverse Problems in Engineering*, Vol. I, pp. 115-119, Rio de Janeiro, Brasil, 2002.
4. Sousa, F.L., Ramos, F.M., P. Paglione, R.M. Girardi, New Stochastic Algorithm for Design Optimization, *AIAA Journal*, Vol. 41, Number 9, pp. 1808-1818, 2003.
5. Sousa, F. L.; Ramos, F. M.; Galski, R. L.; Muraoka, I., Generalized Extremal Optimization: A New Meta-heuristic Inspired by a Model of Natural Evolution, *In Recent Developments in Biologically Inspired Computing,* De Castro, L. N. & Von Zuben, F. J. (editors), Idea Group Inc, 2004.
6. Boettcher, S. and Percus, A.G. Optimization with Extremal Dynamics, *Physical Review Letters*, Vol. 86, pp. 5211-5214, 2001.
7. Bak, P. and Sneppen, K. Punctuated Equilibrium and Criticality in a Simple Model of Evolution, *Physical Review Letters*, Vol. 71, Number 24, pp. 4083-4086, 1993.
8. Sousa, F.L. Vlassov, V. e Ramos, F.M., Generalized Extremal Optimization: An application in Heat Pipe Design, *Applied Mathematical Modeling*, Vol. 28, pp. 911-931, 2004.
9. Galski, R. L., Sousa, F. L., Ramos, F. M., Muraoka, I. Application of a New Hybrid Evolutionary Strategy to Spacecraft Thermal Design. Proceedings of the Genetic and Evolutionary Computational Conference (GECCO-2004), Application of Hybrid Evolutionary Algorithms to Complex Optimization Problems Workshop (in CDROM), June 26 to July 01, Seattle, USA, 2004.
10. Galski, R.L, Sousa, F.L. and Ramos, F.M. Application of a New Evolutionary Algorithm to the Optimum Design of a Remote Sensing Satellite Constelation. Accepted for presentation and publication on the proceedings of the 5th International Conference on Inverse Problems in Engineering: Theory and Practice, Cambridge, UK, 11-15th July, 2005.
11. Rajeev, S. and Krishnamoorthy, C.S. Discrete Optimization of Structures Using Genetic Algorithms, Journal of Structure Engineering, Vol. 118, No. 5, pp. 1233-1250, 1992.
12. Coello Coello, C.A. Discrete Optimization of Trusses Using Genetic Algorithms. In Expert Systems Applications and Artifficial Intelligence EXPERSYS-94. International Technology Transfer Series, pp. 331-336.
13. Turkkan, N. Discrete Optimization of Structures Using a Floating Point Genetic Algorithm, Proceedings of the Annual Conference of the Canadian Society for Civil Engineering, Moncton, Nouveau-Brunswick, Canada, pp. GCM-134-1/8, 2003.
14. Kripka, M. Discrete Optimization of Trusses by Simulated Aneealing, Journal of the Brazilian Society of Mechanical Sciences and Engineering, Vol. XXVI, No. 2, April-June, 2004.
15. Kirkpatrick, S., Gellat, C.D. and Vecchi, M.P. Optimization by simulated annealing, *Science*, 1983, Vol. 220, pp. 671-680.
16. Osman, I.H. and Kelly, J.P. Meta-heuristics: An Overview. In Meta-Heuristics: Theory and Practice. Osman, I.H (Editor), Kluwer Academic Publishers, pp 1-21, 1996.
17. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E. Equation of state calculations by fast computing machines, *Journal of Chemical Physics,* 1953, Vol. 21, pp. 1087-1092.
18. Bak P., *How Nature Works,* Copernicus, Springer-Verlag, New York, 1996.
19. Gould, S.J., Eldredge N., Punctuated equilibrium comes of age. *Nature*, 1993, Vol. 366, p. 223-227.
20. Solé, R.V., Manrubia, S.C., Benton, M. and Bak, P. Self-similarity of extinction statistics in the fossil record. *Nature*, 1997, Vol. 388, pp. 764-767.
21. Kirchner, J.W. and Weil, A. No fractals in fossil extinction statistics. *Nature*, 1998, Vol. 395, pp. 337-338.
22. Cai, Jianbo and Thierauf, G. Discrete Optimization of Structures Using an Improved Penalty Function Method, Eng. Optimization, Vol. 21, pp. 293-306, 1993.
23. Tong, W.H. and Liu, G.R. An Optimization Procedure for Truss Structures With Discrete Design Variables and Dynamic Constraints, Computers & Structures, Vol. 29, pp. 155-162, 2001.
24. Ponteroso, P. and Fox, D.S.J. Heuristically Seeded Genetic Algorithms Applied to Truss Optimization, Engineering With Computers, Vol. 15, pp. 345-355.