GENERALIZED EXTREMAL OPTIMIZATION APPLIED TO THREE-DIMENSIONAL TRUSS DESIGN

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

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

Abstract. *In this paper, an assessment of the efficiency of a new proposed metaheuristic, the Generalized Extremal Optimization (GEO) algorithm, in dealing with discrete three-dimensional truss optimal design problems is made. GEO is a brand new stochastic optimization tool, developed to be easily applied to nonlinear constrained optimization problems, that may have a non-convex or even disjoint design space with the presence of any combination of continuos, discrete and integer variables. Inspired by Bak-Sneppen´s simplified model of evolution and on the dynamics of Self-Organized Criticality (SOC), GEO has showed to be competitive to popular methods such as the Genetic Algorithms (GAs) and Simulated annealing (SA), with the a priori advantage of having only one free parameter to adjust. The optimization of a 25-bar space truss is used here as test case for comparison of performance between GEO and other methods available in literature. Results demonstrate the potential and competitiveness of GEO on tackling problems with this kind of design space.*

Keywords*: Structural Optimization, Truss optimization, Evolutionary Algorithms, Generalized Extremal Optimization*

1. Introduction

In recent years natural phenomena have inspired the development of many stochastic optimization algorithms. 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. The social behavior of insects, the functioning of brain cells, natural evolution and the annealing of metals are examples of such phenomena. Maybe the two most known and used methods inspired from nature are the Simulated Annealing (SA) (Kirkpatrick, Gellat and Vecchi, 1983) and Genetic Algorithms (GAs) (Eiben and Smith, 2003) which have been applied successfully in many areas of engineering and science.

Recently, a new nature inspired stochastic algorithm was proposed. Called Generalized Extremal Optimization (GEO) (Sousa and Ramos, 2002; Sousa et al, 2003; Sousa et al, 2004a), it was originally developed as an improvement of the Extremal Optimization (EO) method (Boettcher and Percus, 2002), which was devised based on the evolutionary model of Bak-Sneppen (Bak, P. and Sneppen, 1993). GEO was developed 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 (Sousa et al, 2004a; Sousa, Vlassov and Ramos, 2004b; Galski et al, 2004; Galski, Sousa and Ramos, 2005) and shown to be competitive to other stochastic methods (Sousa and Ramos 2002, Sousa et al, 2003, Sousa et al, 2005). 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 three-dimensional truss optimization problem. It has been argued, that such kind of problems would not be suitable to traditional gradient based optimization methods, since treating the design variables as continuous would lead to sub-optimum designs (Rajeev and Krishnamoorthy, 1992; Coello Coello, Rudnick and Christiansen, 1994; Kripka, 2004). The 25-bar space truss problem is used here as a test case to assess the performance of GEO, compared to results from other methods available in literature. This is a follow up study about the performance of GEO on structural optimization problems, that started with the assessment of its performance on the optimization of a 10-bar plane truss (Sousa and Takahashi, 2005).

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 25-bar truss test case and the conclusions.

2. A Brief Introduction to 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 25-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

Kirkpatrick et al (1983) proposed the SA method at the early 1980's. It is based on the Metropolis algorithm (Metropolis et al, 1953) 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:

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

where, K_B is the Boltzmann's constant.

Kirkpatrick et al (1983) 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 a number of 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) (Eiben and Smith, 2003). 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), a binary string that encodes the design variables represents each individual. 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. The crossover and mutation operation is illustrated in Figure 1. The position and occurrence of crossover and the number of bits mutated on each offspring string are probabilistic variables.

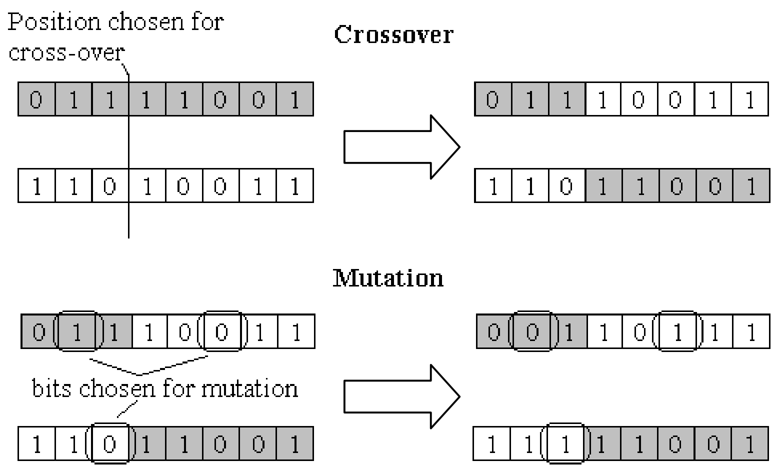

Figure 1. Outline of crossover and mutation in a Simple Genetic Algorithm

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

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 (Bak, 1996). 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:

$$P(s) \approx s^{-\gamma} \quad , \tag{2}$$

where γ is a positive number.

A simple model for a self-organized system is the sand pile (Bak, 1996). 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 (1993) 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 (Gould, S.J., Eldredge, 1993). 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 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, avalanches punctuate the equilibrium (being above the critical level in "stasis") of one or more species, whose occurrence is described by a power law (Bak and Sneppen, 1993). Although the claim that natural evolution may happen in a system that is critically self-organized has been controversial (Solé et al, 1997; Kirchner and Weil, 1998), 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 (2001), 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 (2001). 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 (Sousa and Ramos, 2002; Sousa et al, 2003).

The bit encoding can accommodate any type of design variable and for GEO this could be done as described in Sousa et al (2003). 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 2 is shown the population of species in the canonical GEO and when using a discrete encoding, as for the truss problem. For comparison purposes, it is also shown how the population of individuals is represented in a SGA. Note that in a GA (not only the SGA) the population is composed of individuals, each of them representing a solution in the design space, while in GEO the population is composed of species, each of them representing a bit or a design variable.

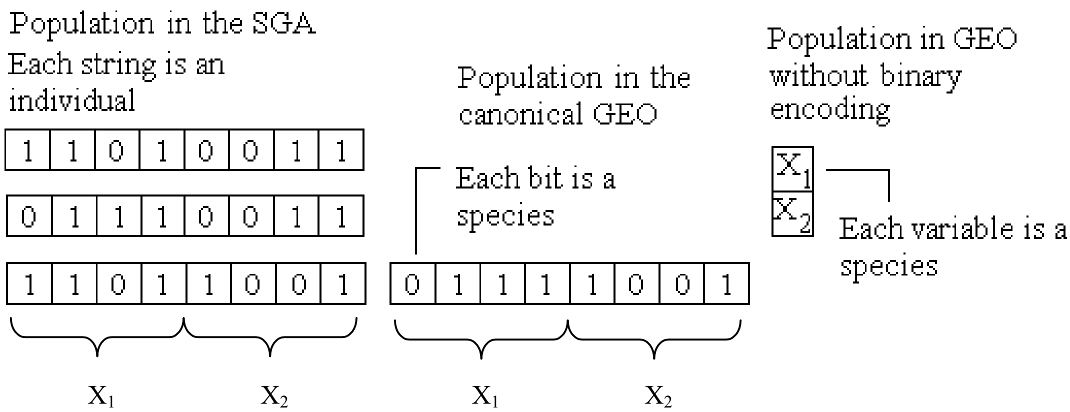

Figure 2. Population as represented in the SGA and GEO for a problem with 2 variables. In this example the population of the SGA has three individuals, while in the canonical GEO has 8 species and 2 species in GEO without binary encoding. For the SGA and the canonical GEO, each variable is encoded in 4 bits.

The main steps for GEO canonical algorithm, and as it was implemented for the 25-bar truss problem is shown in Figure 3.

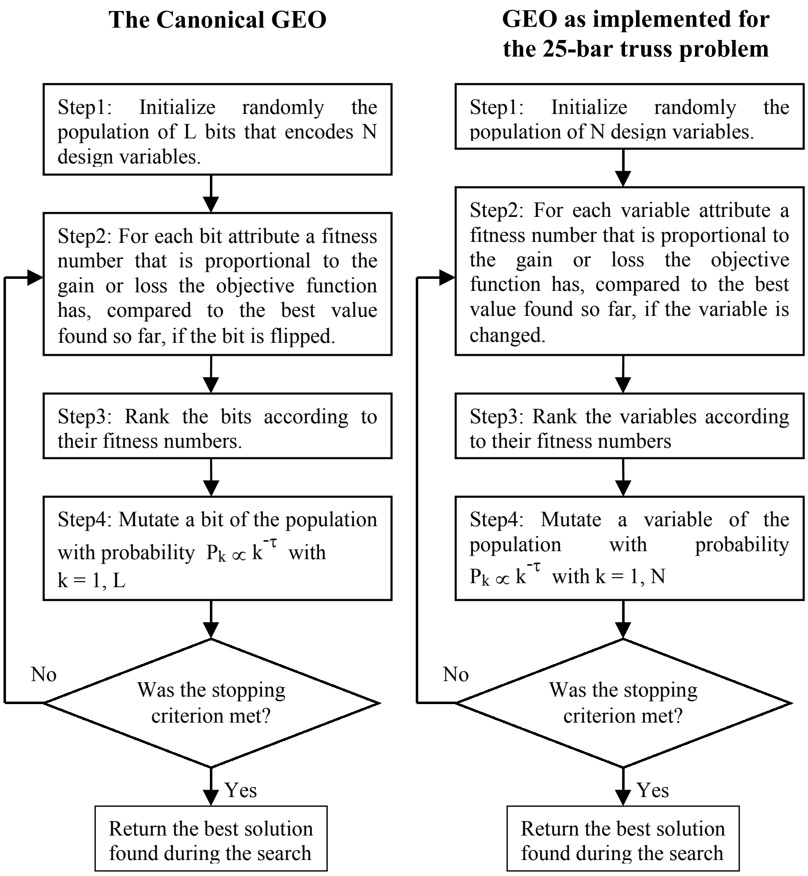

Figure 3. Main steps of the canonical GEO and its implementation for the 25-bar truss optimization problem

In Step 1 the population is initialized randomly with uniform distribution. There is no restriction for the initial solution, it can be an unfeasible one. 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 non-binary implementation) 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 Optimization Problem

The space truss presented in Figure 4 was chosen to evaluate the present optimization algorithm effectiveness. The truss consists of 25 bars L_i long and with the cross sections grouped according to Table 1. Each bar cross-section can assume one of the 30 discrete values of the set: {64.516, 129.032, 193.548, 258.064, 322.58, 387.096, 451.612, 516.128, 580.644, 645.16, 709.676, 774.192, 838.708, 903.224, 967.74, 1032.256, 1096.772, 1161.288, 1225.804, 1290.32, 1354.836, 1419.352, 1483.868, 1548.384, 1612.9, 1677.416, 1806.448, 1935.48, 2064.512, 2193.544} (10^{-6} m^2). Additional problem parameters are the material properties E = 68950 MPa and ρ = 2768 kg/m^3 and the loading condition presented in Table 02.

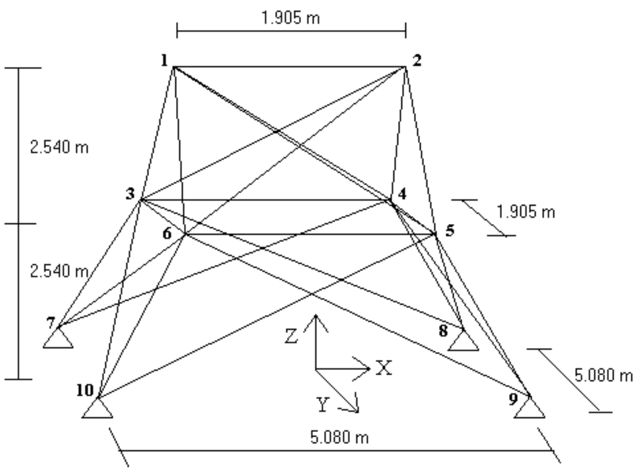

Figure 4. Sketch of the 25-bar space truss

Table 1. Groups of truss elements.

Group Name	Truss Elements (defined by edge nodes)
G_1	1-2
G_2	1-4 ; 2-3 ; 1-5 ; 2-6
G_3	2-5 ; 2-4 ; 1-3 ; 1-6
G_4	3-6 ; 4-5
G_5	3-4 ; 5-6
G_6	3-10 ; 6-7 ; 4-9 ; 5-8
G_7	3-8 ; 4-7 ; 6-9 ; 5-10
G_8	3-7 ; 4-8 ; 5-9 ; 6-10

Table 2. Loading conditions for the 25-bar truss.

Node	Fx (N)	Fy (N)	Fz (N)
1	4453.74	-44537.40	-44537.40
2	0.0	-44537.40	-44537.40
3	2226.87	0.0	0.0
6	2672.24	0.0	0.0

The main goal of the problem is to search a set of cross sections (A_i, i=1,25) minimizing the truss mass (M), as stated in Equation 3, such that, the stress level in each truss element (σ_i, i=1,25) and the displacements at the truss nodes 1 (d_1) and 2 (d_2) remain within the specified limits shown by the Equations 4a and 4b.

$$M = \rho \sum_{i=1}^{25} A_i L_i \tag{3}$$

$$|\sigma_i| \leq 257.6 \text{ MPa for } i = 1,2,\ldots,25 \text{ and;} \tag{4a}$$

$$|d_1| \text{ and } |d_2| \leq 8.89\text{mm} \tag{4b}$$

The stresses and displacements were calculated using a Finite Element Method (FEM) code available in the web site of the Thermal Engineering Laboratory of Chosun University (http://www.chosun.ac.kr/~ygoh/project/source/truss.for).

5. Results

The GEO results for the optimization of the 25-bar space truss were compared with some others 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 proper setting of τ influences the efficiency of GEO on a given application. 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 25-bar truss problem. The range and steps on the values of τ chosen for the search of τ_{best} ([1.00, 10.00], 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. If the search started from an unfeasible design, a high value of mass was assigned as V_{best}, and this value was kept until a feasible design was found. Each run stopped after 10000 function evaluations. Results for the search of τ_{best} are shown in Figure 5.

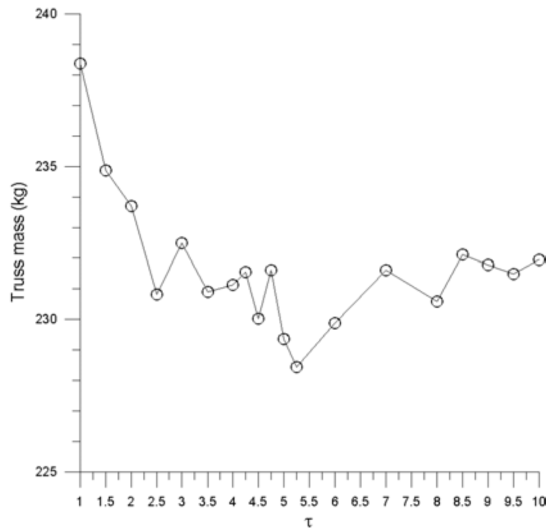

Figure 5. Average of the best values of the objective function in 20 runs for each value of τ.

From Figure 5 it is clear that for this problem GEO is more efficient when τ is set to a value near 5.25. This value was used for the runs on the search for the optimum design. In these runs the stopping criteria used was the same as the one used by Kripka (2004) for tackling the same problem using a SA algorithm: The search stopped after 39201 function evaluations. GEO was executed 20 times from different randomly chosen designs and the best result found is shown in Table 3, together with some other results taken from literature. The evolution of the best value found on each GEO run, as a function of the number of evaluations is shown in Figure 6.

In Table 3 the first and second references used specially devised methods to deal with this kind of problem, while the third, fourth and fifth references used different implementations of the SGA, and Kripka (2004) used the SA method. GEO obtained a better result than all but the SA. However, results from the FEM routine used in this work indicate that Kripka´s result slightly violates the displacement constraint. The average of the best values found in the 20 GEO runs was 221.41 kg with standard deviation of 1.65 kg.

Table 3. Best results for the 25-bar truss problem for GEO and other algorithms from literature.

Reference	Truss Mass (kg)	Section Area of the Group Members (10^{-6} m^2)							
		G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8
(Cai and Thiereu, 1993)	221.08	64.516	64.516	2193.544	64.516	1290.32	645.16	451.612	2193.544
(Thong and Liu, 2001)	220.01	64.516	322.58	2193.544	64.516	1225.804	645.16	258.064	2193.544
(Rajeev and Krishnamoorthy, 1992)	247.67	64.516	1161.288	1483.868	129.032	64.516	516.128	1161.288	1935.48
(Coello Coello, Rudnick and Christiansen, 1994)	224.05	64.516	451.612	2064.512	64.516	903.224	709.676	322.58	2193.544
(Ponteroso and Fox, 1999)	222.65	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
Kripka (2004)	219.69	64.516	258.064	2193.544	64.516	1419.352	645.16	258.064	2193.544
GEO	**219.93**	**64.516**	**193.548**	**2193.544**	**64.516**	**1354.836**	**645.16**	**322.58**	**2193.544**

n.a. – Not available

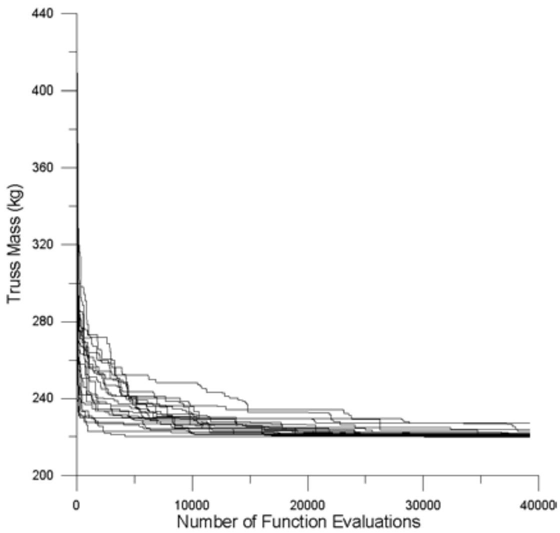

Figure 6. Best value of truss mass as a function of number of evaluations for 20 independent runs of GEO.

5. Conclusions

In this work an assessment on the performance of the Generalized Extremal Optimization algorithm when applied to the optimal design of a three-dimensional truss was made. As a follow up to a similar previous work done for a plane truss, the results shown here confirm that GEO can be a very competitive alternative to other metaheuristics, such as SA and GAs, to be used on discrete structural optimization problems. Being more easily set for the problem at hand than the SA and GAs, GEO would also produce good results quicker and with a lower computational cost. Being a brand new algorithm, it is expected that future developments would improve even better the performance of GEO. 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 comprehensive analysis of its efficiency on solving different types of structural optimization problems is also a topic of our current research.

6. Acknowledgements

7. References

Bak P., "How Nature Works", 1996, Springer-Verlag, New York.

Bak, P. and Sneppen, K., 1993, "Punctuated Equilibrium and Criticality in a Simple Model of Evolution", *Physical Review Letters*, Vol. 71, Number 24, pp. 4083-4086.

Boettcher, S. and Percus, A.G., 2001, "Optimization with Extremal Dynamics", Physical Review Letters, Vol. 86, pp. 5211-5214.

Cai, Jianbo and Thierauf, G., 1993, "Discrete Optimization of Structures Using an Improved Penalty Function Method", Eng. Optimization, Vol. 21, pp. 293-306.

Coello Coello, C.A., Rudnick M. and Christiansen, A.D., 1994, "Using Genetic Algorithms for Optimal Design of Trusses", Proceedings of the Sixth International Conference on Tools With Artificial Intelligence, 6-9 october, New Orleans, USA.

Eiben, A.E. and Smith J.E., 2003, Introduction to Evolutionary Computing. Springer, Berlin.

Finite Element Method Source code: Downloaded from the web site of the Thermal Engineering Laboratory of Chosun University. http://www.chosun.ac.kr/~ygoh/project/source/truss.for

Galski, R. L., Sousa, F. L., Ramos, F. M., Muraoka, I., 2004, "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,.

Galski, R.L, Sousa, F.L. and Ramos, F.M., 2005, "Application of a New Evolutionary Algorithm to the Optimum Design of a Remote Sensing Satellite Constelation", Accepted for publication on the proceedings of the 5th International Conference on Inverse Problems in Engineering: Theory and Practice, Cambridge, UK, 11-15th July, 2005.

Gould, S.J., Eldredge N., "Punctuated equilibrium comes of age", 1993, Nature, Vol. 366, p. 223-227.

Kirchner, J.W. and Weil, A., "No fractals in fossil extinction statistics", 1998, Nature, Vol. 395, pp. 337-338.

Kirkpatrick, S., Gellat, C.D. and Vecchi, M.P., 1983, "Optimization by simulated annealing", Science, Vol. 220, pp. 671-680.

Kripka, M., "Discrete Optimization of Trusses by Simulated Aneealing", 2004, Journal of the Brazilian Society of Mechanical Sciences and Engineering, Vol. XXVI, No. 2, April-June.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E., 1953, "Equation of state calculations by fast computing machines", Journal of Chemical Physics, Vol. 21, pp. 1087-1092.

Osman, I.H. and Kelly, J.P., 1996, "Meta-heuristics: An Overview. In Meta-Heuristics: Theory and Practice", Osman, I.H (Editor), Kluwer Academic Publishers, pp 1-21,.

Ponteroso, P. and Fox, D.S.J., 1999, "Heuristically Seeded Genetic Algorithms Applied to Truss Optimization", Engineering With Computers, Vol. 15, pp. 345-355.

Rajeev, S. and Krishnamoorthy, C.S., 1992, "Discrete Optimization of Structures Using Genetic Algorithms", Journal of Structure Engineering, Vol. 118, No. 5, pp. 1233-1250.

Solé, R.V., Manrubia, S.C., Benton, M. and Bak, P., 1997, "Self-similarity of extinction statistics in the fossil record", Nature, Vol. 388, pp. 764-767.

Sousa, F.L. and Takahashi, W.K. Discrete Optimal Design of Trusses by Generalized Extremal Optimization, 2005, accepted for publication on the proceedings of the 6th Worls Congress of Structural and Multidisciplinary Optimization, Rio de Janeiro, 30 May to 03 June, Brazil.

Sousa, F.L., F.M. Ramos, F.J.C.P. Soeiro and A.J. Silva Neto., 2005, "Application of the Generalized Extremal Optimization Algorithm to an Inverse Radiative Transfer Problem", accepted for publication on the proceedings of the 5th International Conference on Inverse Problems in Engineering: Theory and Practice, Cambridge, UK, 11-15th July,.

Sousa, F. L.; Ramos, F. M., Function Optimization Using Extremal Dynamics, 2002, "Proceedings of the 4th International Conference on Inverse Problems in Engineering", Vol. I, pp. 115-119, Rio de Janeiro, Brasil.

Sousa, F.L., Ramos, F.M., P. Paglione, R.M. Girardi, 2003, "New Stochastic Algorithm for Design Optimization", AIAA Journal, Vol. 41, Number 9, pp. 1808-1818,.

Sousa, F. L.; Ramos, F. M.; Galski, R. L.; Muraoka, I., 2004, "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,.

Sousa, F.L. Vlassov, V. e Ramos, F.M., 2004, "Generalized Extremal Optimization: An application in Heat Pipe Design", Applied Mathematical Modeling, Vol. 28, pp. 911-931.

Tong, W.H. and Liu, G.R., 2001, "An Optimization Procedure for Truss Structures With Discrete Design Variables and Dynamic Constraints", Computers & Structures", Vol. 29, pp. 155-162.