OPTIMIZATION OF TRUSSES BY SIMULATED ANNEALING TECHNIQUE

Herbert Martins Gomes, herbert@mecanica.ufrgs.br
Franklin de Souza Ferreira, franksfbr@mecanica.ufrgs.br
UFRGS, Federal University of Rio Grande do Sul, Post Graduate Program in Mechanical Engineering, Porto Alegre, Rio Grande do Sul, Brazil. Tel.: (+55 51)3308-3681

Abstract. *. The Simulated Annealing (SA) Technique belongs to a Stochastic Optimization class of algorithms. This technique has been used as a soft computing technique in hard optimization tasks, such as, electronic components allocation, spatial representation of chemical compounds and Travelling Salesman type problems, for a long period. This technique is based on the mathematical description of the experimental cooling technique developed to design stronger crystals (like glass) and metals. In this paper this technique was implemented on a Matlab environment and applied to simple and difficult ones parametric trusses optimization problems with constrains in displacements and stresses. The examples were selected in order to compare results with those presented by related literature. SA performance is compared with those obtained with other heuristic methods like the Genetic Algorithm (GA) and with gradient based Mathematical Programming, such as Sequential Quadratic Programming (SQP). The presented results shows some disadvantages regarding computational time cost with the SA Technique, nevertheless the accuracy and final results shows better or similar results than the other methods.*

Keywords*: simulated annealing, heuristic algorithms, stochastic optimization.*

1. INTRODUCTION

The SA Method is a Technique that has attracted attention due to its application to large optimization problems, especially those where global optimum are hidden among several ‘worst’ local optimum. For practical purposes the SA has solved the well-known traveler-salesman problem where a traveler salesman has to visit ‘N’ cities at most once in an economical way (tracking the small path). Other heuristics methods have been used with success as well. The SA method has been applied with success to design complex integrated circuits boards. The set of hundreds of circuit components in a board is optimized in such a way that the interference between tracks is minimized. Surprisingly the algorithm implementation is relatively simple. Those previously mentioned applications are combinatorial optimization problems. In such cases, as usual, there is an objective function to be optimized; however the search space is not an N dimensional space of continuum variables. On the contrary, the search space is finite and discrete but very large, as the set of the sequence of visited cities or the possibilities of circuit block allocation. The number of possible solutions in the search space is exponentially large enough to prevent any exhaustive searches be worth. In addition, as the search space is discrete, any definition of derivatives is senseless (as the intuition to use the gradient descent methods).

The core of the SA Method is the thermodynamic analogy, specifically in the way the liquids frozen and crystallized or metals anneal as they get cold. At high temperature, as happens in liquids, the molecules move freely between each other. If the liquid is slowly cooled, the thermal mobility is lost. The atoms are often able to align theirselves and a pure crystal that is completely aligned at the least energy configuration is formed. Such crystal presents a minimal energy level for the material. Surprisingly, minimal energy states are found naturally. In fact, if a liquid metal is suddenly cooled and not annealed it does not reach such states, in the opposite it transforms in an amorphous polycrystalline with an energy state higher than those slowly cooled. Thus, the essence of the method is the slowly cooling, allowing the re-distribution of the atoms and molecules as they loose mobility. This is the technical definition for annealing and it is essential to assure that the less energy state will be reach. Although this analogy is not perfect, this avoids some problems like those associated to the gradient descent search. In the molecule level, the well-known Boltzman probability distribution is defined as,

$$P(\Delta E) = e^{-\frac{\Delta E}{kT}} \tag{1}$$

and it expresses the idea that a system thermodynamically in equilibrium at temperature T has its own energy probabilistically distributed in different energy levels. Even at low temperature, there is a little chance that the system was in a high energy state. So, there is a chance that the system could leave from a local minimum, crossing higher energy levels in order to find a better solution far from original position. The Boltzman constant k is the constant that in nature relates the temperature with energy.

In other words, the system make ascents as descents, but at low temperature, uphill excursion are less probably that at higher temperature. In 1953, Metropolis and co-authors firstly incorporated those principles on numerical estimates.

First it was proposed a series of options and a thermodynamically system was assumed to change its energy state configuration from E_1 to E_2 with a probability $p = \exp(-(E_2 - E_1)/kT)$. It should be highlighted that if $E_2 < E_1$ this probability is higher than unity. In this case this change is attributed to the probability $p = 1$, in this case the system always takes this option, i.e., it changes its energy level to a lower one. The whole picture is to take always descent steps and sometimes take ascent steps and these rules became known as Metropolis Algorithm. In order to use the Metropolis Algorithm to different non-thermodynamic systems, some adjustments should be taken into account.

a. It is necessary a precise description of all possible system configurations.

b. A random generator of feasible "perturbed" system configurations should be available.

c. It is necessary to define an objective function *E,* similar to the energy function, which will be minimized by the procedure.

d. A control parameter *T,* similar to the temperature, and an Annealing Scheme must be chosen in order to indicate how temperature decreases with time. Some stop criteria should be elected, such as the maximum allowed number of system perturbations, number of iterations with no change in the objective function *E,* related to a system configuration, to be considered as an optimum.

e. A dimensional parameter *k*, similar to the Boltzman constant, should be used to adjust the probabilities of acceptance for uphill climbs. This parameter will depend on the units of the Energy function as well as the units of the Temperature parameter.

The proof of the SA as a global optimization Algorithm can be found everywhere such as in Delyon (1988), Locatelli(2000), Ingber(1989) and Rajasekaran (1990). It is not intended to develop or to show the proof in this paper, but it can be said that most of the proofs are based on Markov Chain models and the Theory of Probability.

1.2. Algorithm scheme for function minimizations

In the following Fig. it is described the way the Annealing Algorithm behaves for function minimizations with a single variable. In this case x_i is the variable value at iteration *i, E* is the objective function value and Δ is a perturbation applied on the values of the variable x_i.

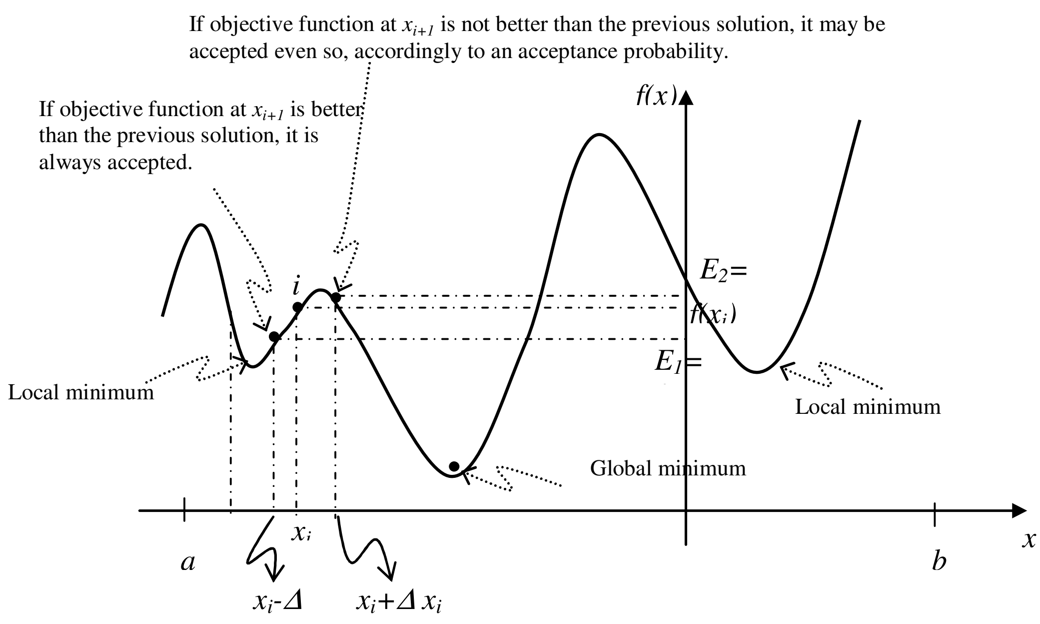

Figure 1. Behavior of the Annealing Algorithm for function minimization with one variable.

The extension for multivariable problems is straightforward. A simplified sketch (pseudo-code) of the algorithm implementation is depicted in the following box.

Table 1. Pseudo-Code for SA Algorithm.

Initialization (Current solution at x_i, current temperature Ti)
Evaluation of the current objective function E_i.
While stop Criteria is not satisfied
    Through perturbation of the current solution x_i, find a new state x_{i+1}
    Evaluation of the objective function at new state E_{i+1}
    If $(E_i\text{-}E_{i+1}) \leq 0$ then
        $x_i = x_{i+1}$
    else
        If $e^{\left(\frac{E_i - E_{i+1}}{T_i}\right)} > radom(0,1)$ then
            this new state is accepted ($x_i=x_{i+1}$)
        else
            this state is rejected
        End of if
    End of if
    Evaluation of the Stop Criteria
    Decrease the Temperature following a cooling schedule
End of while

2. RESULTS

2.1. Example 1 – Optimization of a four bar truss

The structure of a simple four bar truss is optimized by the SA algorithm. This example was solved by Haftka (1991) with the Linear Programming Technique. The applied load is P=10N, the length is L=2,0 m, the Young Modulus of the members is $E=1.0\times10^4$ N/m^2. The constraints in the vertical displacement on node 3 is set as $y_3<3.0\times10^{-6}$L and the stress constraint for all member is set as $\sigma_c<4.833\times10^{-4}$ E (N/m^2) for the compression members and $\sigma_t>-7.73\times10^{-4}$ E (N/m^2) for the tension members. For simplicity the material density is set as $\rho=1.0$ kg/m^3. The sectional areas are treated as the design variables and the allowed range of variation is set from 0.1 m^2 to 10.0 m^2. Figure 2 shows a sketch for the analyzed truss. In this problem, members 1, 2 and 3 have the same cross sectional area and the last one, the 4$^{\text{th}}$, has another cross sectional area, so the optimization problem simplifies to a two design optimization problem. The solution presented in the literature shows a minimal weight of 89.57 kg for the design variables $A_1=A_2= A_3= A_4=9.464$ m^2. In this example, the following parameters were used for the simulations (Tab.2):

Table 2. Parameters used in the four member truss example with the SA Algorithm.

Temperature Schedule	Exponential
Initial Temperature	1.0
Reduction Temperature factor	0.85
Maximum Number of iterations	100000
Tolerance for Convergence	1.0E-3

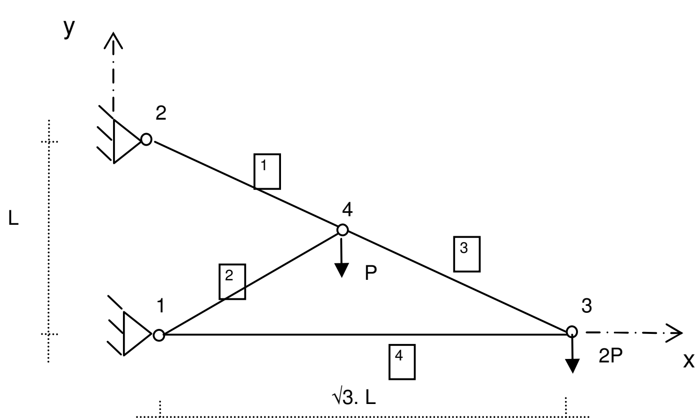

Figure 2. Four member truss sketch.

In the following Fig. 3, it is shown the results obtained with the SA Technique along the iterations. The final obtained design variables were 9.4477 m^2. And the final total weight of the truss was 89.486 kg. Since, in this example, the technique to account for constraints is the penalty technique, only constraints in displacement were slightly violated by 0.1%, neither tension nor compression stresses were violated in this example.

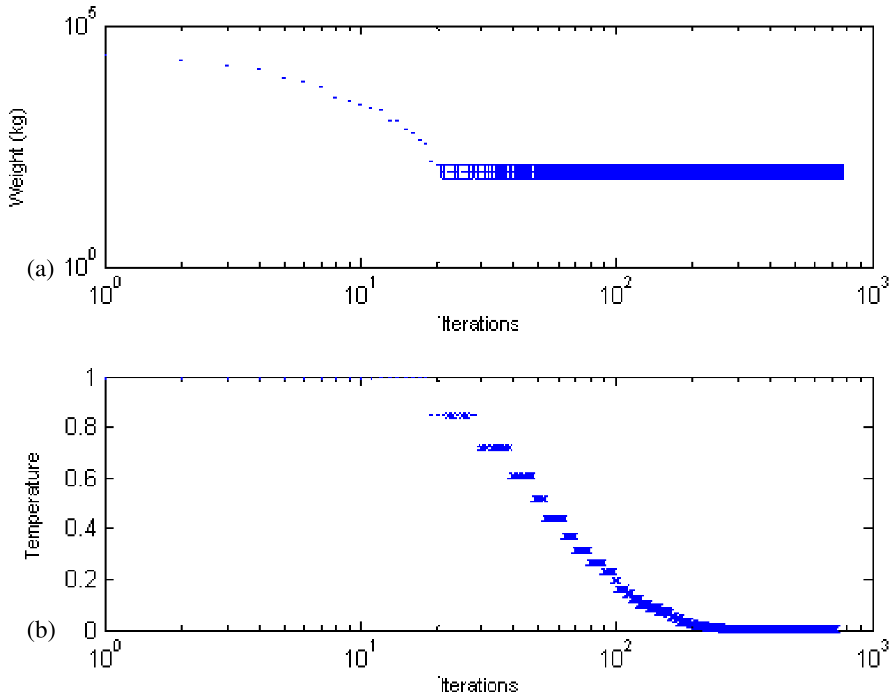

Figure 3. Objective Function Values and Temperature versus iteration for example 1. (a) Objective function (b) Temperature.

The total elapsed time using a Pentium 4 computer with 1 GB RAM and 1.8MHz CPU was about 185 seconds. The total number of function evaluations was 3201. Less than a half of the total number of evaluations (1367) was accepted (uphill moves).

The same problem solved with the Genetic Algorithm Toolbox presented a final weight of 89.49 kg with the final cross sectional areas of $A_1=A_2=A_3=9.399$ m^2 and $A_4=9.53$ m^2 with a displacement constraint violation about 10^{-4}. And the same problem solve with the Sequential Quadratic Programming (SQP) technique presented the final weight of 89.55 kg and the final cross sectional areas of $A_1=A_2=A_3=9.462$ m^2 and $A_4=9.463$ m^2 with a displacement constraint violation about 10^{-5}.

2.2. Example 2 – Optimization of a ten member truss

In this example, the weight of a ten member truss is optimized using the SA Algorithm. This example was also solved using Sequential Nonlinear Approximate Optimization by Haftka and Gürdal (1991). The applied load at nodes 4 and 2 is $P=4.448\times10^5$ N [100 Kips]. Both horizontal and vertical member length is L = 9.144 m (360 in).

The material Young Modulus is $E=6.8958\times10^6$ N/m^2 (1 ksi), for all members there is a tension/compression limit constraint of $|\sigma_c|=|\sigma_t|<1.724\times10^8$ N/m^2 (25 ksi), but for the ninth member, those limits are modified to $|\sigma_c|=|\sigma_t|<5.171\times10^8$ N/m^2 (75 ksi). The material mass density is set as $\rho=2.768\times10^3$ kg/m^3 (0.1 lbm/in^3). All the cross sectional areas are design variables and can range from 6.452×10^{-5} m^2 to 6.452×10^{-3} m^2 (0.1 to 10.0 in^2). So the problem is a 10 design variable Optimization Problem. Figure 4 shows a sketch of the truss.

The same parameters used in Example 1 in the SA Algorithm were used in this example. Haftka and Gürdal (1991) indicates the following best solution for this optimization task: $A_1=5.097\times10^{-3}$ m^2(7.90 in^2), $A_2= 6.452\times10^{-5}$ m^2 (0.10 in^2), $A_3=5.226\times10^{-3}$ m^2 (8.10 in^2), $A_4=2.516\times10^{-3}$ m^2 (3.90 in^2), $A_5=6.452\times10^{-5}$ m^2 (0.10 in^2), $A_6=6.452\times10^{-5}$ m^2 (0.10 in^2), $A_7=3.472\times10^{-3}$ m^2 (5.80 in^2), $A_8=3.555\times10^{-3}$ m^2 (5. 51 in^2), $A_9=2.374\times10^{-3}$ m^2 (3.68 in^2), $A_{10}=9.032\times10^{-5}$ m^2 (0.14 in^2), with a minimum weight of 679.028 kg (1.497 lb).

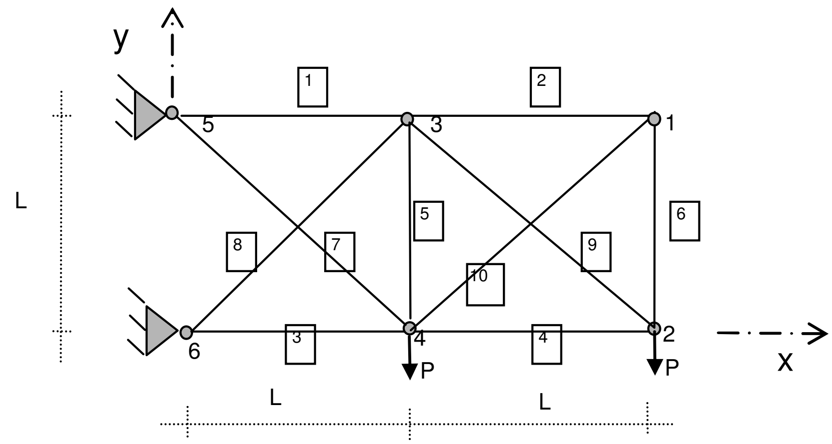

Figure 4. Sketch of the ten member truss.

Figure 5 shows the evolution of the weight and temperature values during iteration. The total elapsed time for this example last 1264 s on the same computer architecture indicated in example 1. The obtained solution with SA was A_1=4.759x10^{-3} m^2(7.37 in^2), A_2= 4.00x10^{-4} m^2 (0.62 in^2), A_3=5.555x10^{-3} m^2 (8.61 in^2), A_4=2.183x10^{-3} m^2 (3.38 in^2), A_5=6.458x10^{-5} m^2 (0.10 in^2), A_6=6.452x10^{-5} m^2 (0.10 in^2), A_7=4.21x10^{-3} m^2 (6.52 in^2), A_8=3.08x10^{-3} m^2 (4.774 in^2), A_9=2.053x10^{-3} m^2 (3.18 in^2), A_{10}=5.658x10^{-5} m^2 (0.87 in^2), with a minimum weight of 693.2 kg (1.5283 lb).

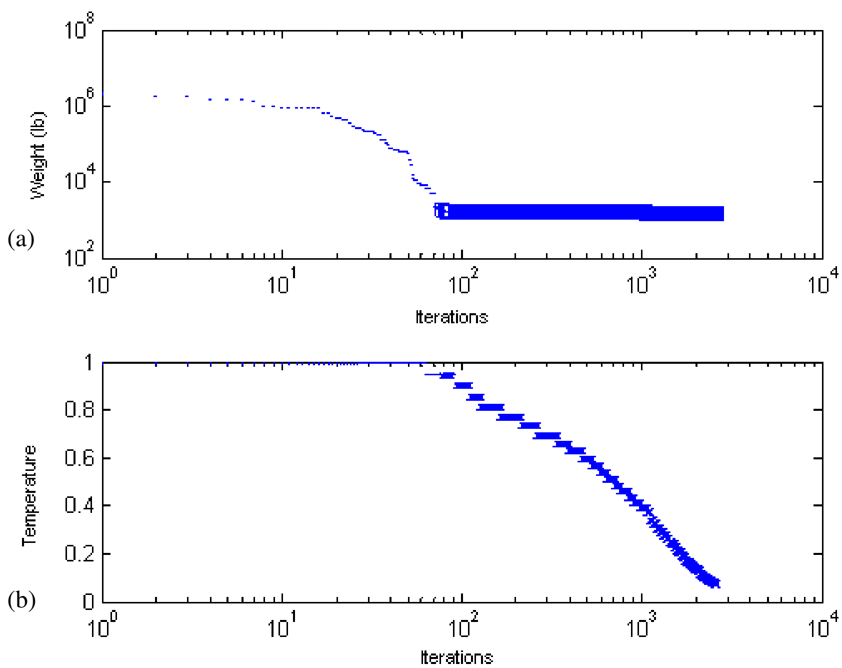

Figure 5. Objective Function Values and Temperature versus iteration for example 2. (a) Objective function. (b) Temperature.

It can be noticed that the results were slightly worse, since the weight was 2.1% greater than the indicated by the literature. The same problem was solved by Teles (2007) using Genetic Algorithm and the results for the cross sectional areas were A_1= 4.623x10^{-3} m^2(7,16583 in²), A_2= 6.452x10^{-4} m^2(1,00383 in²), A_3= 5.79510^{-3} m^2 (8,98245 in²), A_4=2.192x10^{-3} m^2 (3,39835 in²), A_5= 8.903x10^{-6} m^2 (0,01381 in²), A_6= 6.354x10^{-4} m^2 (0,98495 in²), A_7= 4.485x10^{-3} m^2 (6,95241 in²), A_8= 2.888x10^{-3} m^2 (4,47655 in²) , A_9= 2.1543x10^{-3} m^2 (3,33849 in²), A_{10}= 7.872x10^{-4} m^2 (1,22012 in²), with a minimum weight of 721.1 kg (1.5897 lb).

Teles (2007) solved the same problem by the SQP algorithm and the algorithm did not converged due to excessive function evaluations (>1000). The obtained solution for the cross sectional areas in the last iteration were A_1= 5.096x10^{-3} m^2(7,899 in²), A_2= 6.452x10^{-5} m^2(0.10 in²), A_3= 5.225x10^{-3} m^2 (8,098 in²), A_4=2.516x10^{-3} m^2 (3,90 in²), A_5=

6.452×10^{-5} m^2 (0,100 in²), A_6= 6.452×10^{-5} m^2 (0,100 in²), A_7= 3.740×10^{-3} m^2 (5.797 in²), A_8= 3.558×10^{-3} m^2 (5.5151 in²) , A_9= 2.372×10^{-3} m^2 (3,6763 in²), A_{10}= 9.123×10^{-5} m^2 (0,1414 in²), with a minimum weight of 679.22 kg (1497.43 lb) with displacements constraint violation of 10^{-6}. This result is better than the previous ones with Annealing and Genetic Algorithm.

2.3. Example 3 – Optimization of a Twenty Five Member Truss

This example intends to optimize the weight of the 25 member truss sketched in Fig. 6. This truss has groups of members with the same cross sectional area. The 1st. group includes just bar No. 1, the 2nd. group includes bars 2, 3, 4 and 5. The 3rd. group includes bars 6, 7, 8 and 9. The 4th. group includes bars 10 and 11, the 5th. group includes bars 12 and 13, the 6th. group includes bars 14, 15, 16 and 17, the 7th. group includes bars 18, 19, 20 and 21 and the last group includes bars 22, 23, 24 and 25. Each of the group member area is allowed to vary between 3.226×10^{-5} m^2 to 2.581×10^{-3} m^2 (0.05 to 4.0 in^2), so this example is an 8 design variable Optimization Problem. The mass density is assumed as $\rho=2.768\times10^3$ kg/m^3 (0.10 lbm/in^3) and the Young Modulus E= 6.895×10^{10} N/m^2 (1.0×10^4 ksi). The loads are applied as indicated by Tab. 3.

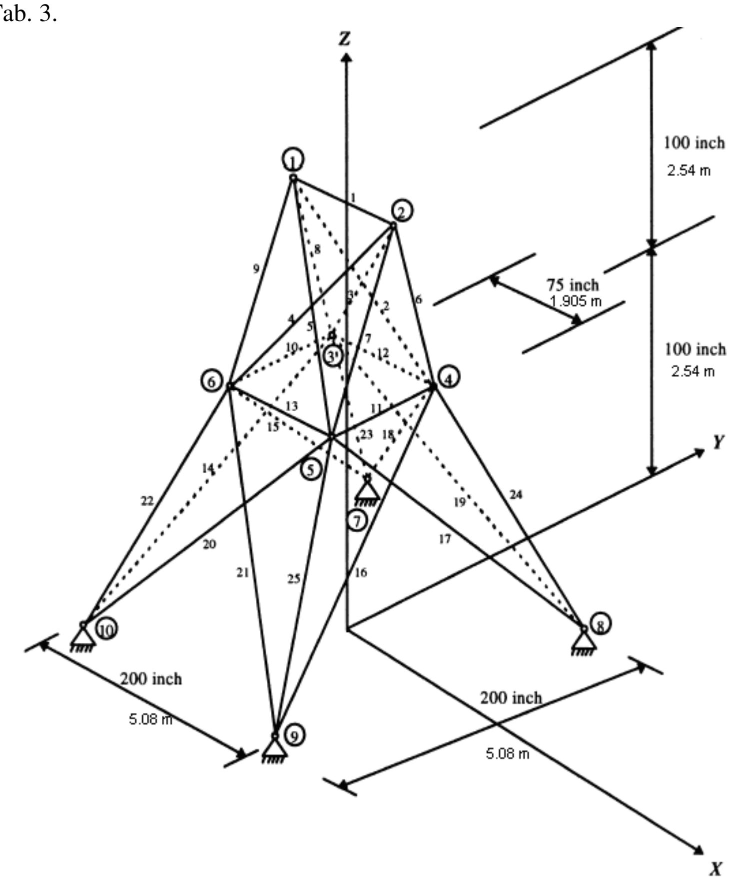

Figure 6. Sketch of the 25 member truss.

Table 3. Applied loads on nodes for the 25 member truss.

Node	Fx	Fy	Fz
1	4.448×10^3 N (1000 lbf)	-4.448×10^3 N (-1000 lbf)	-4.448×10^3 N (-1000 lbf)
2	0.0 N (0.0 lbf)	-4.448×10^3 N (-1000 lbf)	-4.448×10^3 N (-1000 lbf)
3	2.224×10^3 N(500.0 lbf)	0.0 N (0.0 lbf)	0.0 N (0.0 lbf)
6	2.669×10^3 N (600.0 lbf)	0.0 N (0.0 lbf)	0.0 N (0.0 lbf)

Only constraints on displacements and stresses are set. The values are the following: $|x_1|<8.89\times10^{-3}$(0.35in) $|x_2|<8.89\times10^{-3}$ (0.35in) , $|y_1|<8.89\times10^{-3}$ (0.35 in), $|y_2|<8.89\times10^{-3}$ (0.35 in), $|z_1|<8.89\times10^{-3}$(0.35in), $|z_2|<8.89\times10^{-3}$(0.35in), $|\sigma_c|<2.758\times10^8$ N/m^2(40.0 ksi), $|\sigma_t|<2.758\times10^8$ N/m^2 (40.0 ksi).

Rizz *apud* Pyrz and Zawidska (2001) presented the following best solution for this optimization task: $A_1=6.452\times10^{-6}$ m^2(0.01 in^2), A_2= 1.283×10^{-3} m^2 (1.988 in^2), $A_3=1.93\times10^{-3}$ m^2 (2.991 in^2), $A_4=6.452\times10^{-6}$ m^2 (0.01 in^2),

A_5=6.452x10^{-6} m^2 (0.010 in^2), A_6=4.413x10^{-4} m^2 (0.684 in^2), A_7=1.081x10^{-3} m^2 (1.676 in^2), A_8=1.717x10^{-3} m^2 (2.662 in^2), with a minimum weight of 247.28 kg (545.16 lb).

This example took 1127.8 seconds to reach the optimal solution with the proposed SA Algorithm with the same previous computer architecture. The obtained solution with SA was A_1=3.028x10^{-6} m^2(4.694x10^{-3} in^2), A_2= 2.888x10^{-5} m^2 (4.477x10^{-2} in^2), A_3=2.345x10^{-3} m^2 (3.635 in^2), A_4=6.452x10^{-7} m^2 (1x10^{-3} in^2), A_5=1.286x10^{-3} m^2 (1.993 in^2), A_6=5.013x10^{-4} m^2 (0.777 in^2), A_7=1.026x10^{-4} m^2 (0.159 in^2), A_8=2.527x10^{-3} m^2 (3.917 in^2), with a minimum weight of 210.707 kg (464.53 lb). The total amount of Function evaluation was 9601 with 3628 uphill acceptances. Only displacements constraint violations of about 1% in were noticed for this solution.

The same problem was solved by Teles (2007) using Genetic Algorithm and the results for the cross sectional areas were A_1= 3.99x10^{-5} m^2(0.062 in²), A_2= 3.74x10^{-5} m^2(0.058 in²), A_3= 2.15x10^{-3} m^2 (3.33 in²), A_4=6.73x10^{-5} m^2 (0.104 in²), A_5= 1.222x10^{-3} m^2 (1.894 in²), A_6= 4.744x10^{-4} m^2 (0.74 in²), A_7= 4.66x10^{-5} m^2 (0.072 in²), A_8= 2.468x10^{-3} m^2 (3.825 in²), with a minimum weight of 204.04 kg (449.83lb).

The same problem was solved by Teles (2007) using SQP and the results for the cross sectional areas were A_1= 3.23x10^{-5} m^2(0.05 in²), A_2= 3.23x10^{-5} m^2(0.05 in²), A_3= 2.43x10^{-3} m^2 (3.77 in²), A_4=3.23x10^{-5} m^2 (0.05 in²), A_5= 1.285x10^{-3} m^2 (1.992 in²), A_6= 5.015x10^{-4} m^2 (0.777 in²), A_7= 1.021x10^{-4} m^2 (0.158 in²), A_8= 2.527x10^{-3} m^2 (3.917 in²), with a minimum weight of 211.22 kg (465.66 lb).

It is clear that in this case, the results from GA showed better results than SA. This last method (SA) in turn showed better results than the SQP algorithm. In the following Fig. it is shown the behavior of the weight and temperature with iterations.

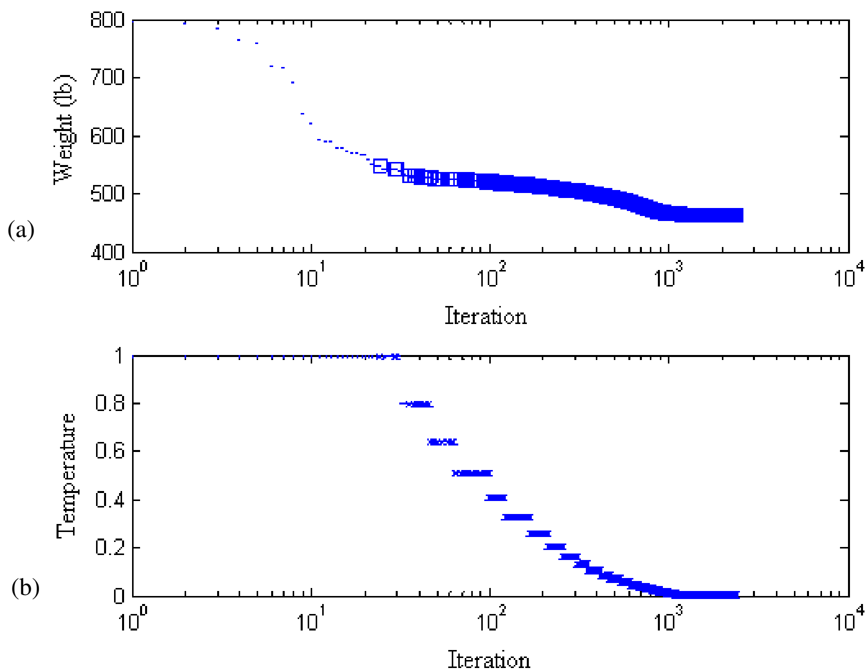

Figure 7. Objective Function Values and Temperature versus iteration for example 3. (a) Objective function. (b) Temperature.

2.4. Example 4 – Optimization of a Seventy Two member Truss

In this example, the weight of a seventy two member truss is optimized. This problem was analyzed by Erbatur *et al.*(2000) using GA. Figure 8 shows a sketch of the analyzed truss structure. The mass density is assumed as ρ=2.768x10^3 kg/m^3 (0.10 lbm/in^3) and the Young Modulus E= 6.895x10^{10} N/m^2(1.0x10^4 ksi). The loads are applied as indicated by Tab. 3.

Table 4. Applied loads on nodes for the 72 member truss.

Node	Fx	Fy	Fz
1	2.224x10^4 N (5000 lbf)	2.224x10^4 N (5000 lbf)	-2.224x10^4 N (-5000 lbf)

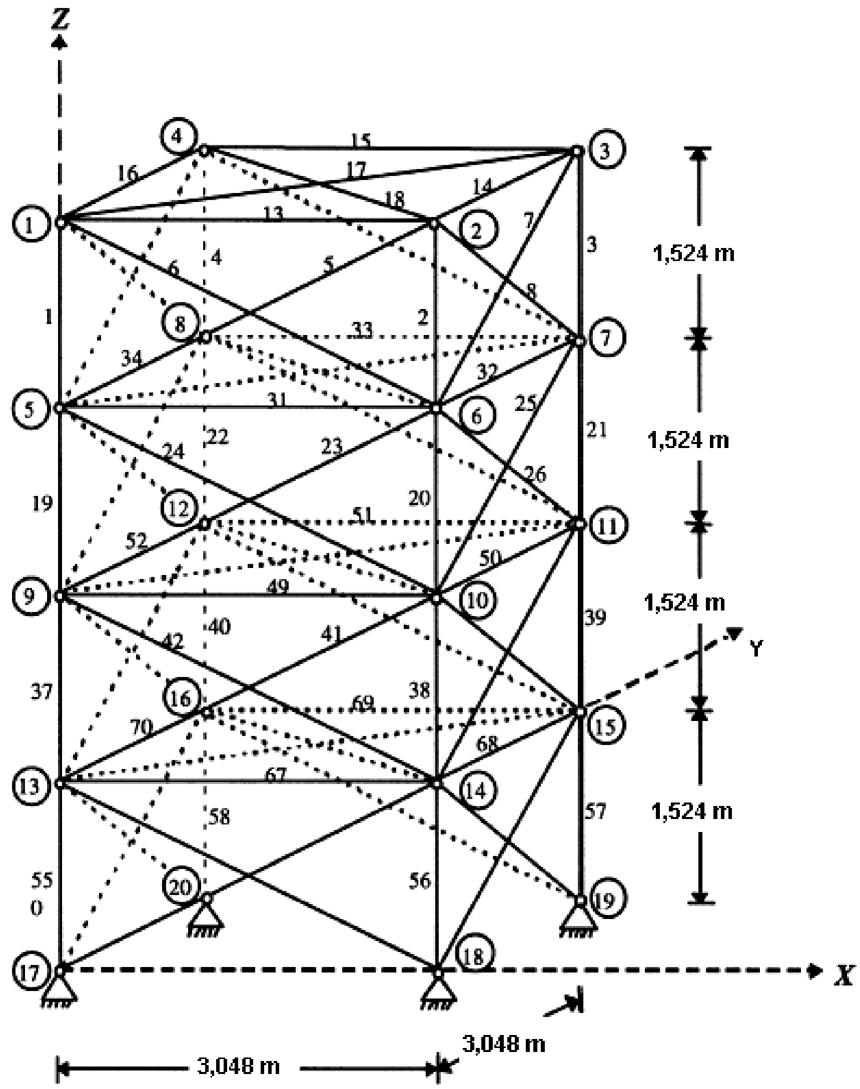

Figure 8. Sketch of the 72 member truss.

Table 5 shows the member groups. There are 16 groups, so this is a 16 design variable optimization problem.

Table 5. Member groups for the 72 member truss.

Group	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
Member	1-4	5-12	13-16	17-18	19-22	23-30	21-24	35-36
Group	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}
Member	37-40	42-48	49-52	53-54	55-58	59-66	67-70	71-72

Constraints on displacements and stresses are set such as in *x* and *y* directions the displacements on all nodes should not exceed $6.35x10^{-3}$m (0.25 in.). Furthermore $|\sigma_c|$ and $|\sigma_t|<1,7237x10^8$ N/m^2 (25 ksi). All member groups cross sectional areas are allowable to assume values greater than $6.452x10^{-5}$(0.1 in^2).

Erbatur *et al.*(2000) presented the following best solution for this optimization task: $A_1=1.0x10^{-4}$ m^2(0.155 in^2), $A_2=$ $3.5x10^{-4}$ m^2 (0.535 in^2), $A_3=3.1x10^{-3}$ m^2 (0.480 in^2), $A_4=3.4x10^{-4}$ m^2 (0.52 in^2), $A_5=3.0x10^{-4}$ m^2 (0.460 in^2), $A_6=3.4x10^{-4}$ m^2 (0.53 in^2), $A_7=0.81x10^{-4}$ m^2 (0.12 in^2),$A_8=1.1x10^{-4}$ m^2 (0.165 in^2), $A_9=7.5x10^{-4}$ m^2 (1.155 in^2), $A_{10}=3.8x10^{-4}$ m^2 (0.585 in^2),$A_{11}=0.6x10^{-4}$ m^2 (0.1 in^2), $A_{12}=0.6x10^{-4}$ m^2 (0.1 in^2), $A_{13}=11.3x10^{-4}$ m^2 (1.755 in^2),$A_{14}=3.3x10^{-4}$ m^2 (0.505 in^2), $A_{15}=0.7x10^{-4}$ m^2 (0.105 in^2), $A_{16}=1.0x10^{-4}$ m^2 (0.155 in^2), with a minimum weight of 174.98 kg (385.76 lb).

This example took 1351 seconds to reach the optimal solution with the proposed SA Algorithm with the same previous computer architecture. The obtained solution with SA was: A_1=6.47x10^{-5} m^2(0.1 in^2), A_2= 3.72x10^{-4} m^2 (0.576 in^2), A_3=2.523x10^{-4} m^2 (0.391 in^2), A_4=3.263x10^{-4} m^2 (0.5058 in^2),A_5=3.493x10^{-4} m^2 (0.5414 in^2), A_6=3.408x10^{-4} m^2 (0.5283 in^2), A_7=6.47x10^{-5} m^2 (0.1 in^2), A_8=6.47x10^{-5} m^2 (0.1 in^2), A_9=8.31x10^{-3} m^2 (1.288 in^2), A_{10}=3.67x10^{-4} m^2 (0.57 in^2), A_{11}=6.581x10^{-5} m^2 (0.102 in^2), A_{12}=6.581x10^{-5} m^2 (0.102 in^2), A_{13}=9.626x10^{-4} m^2 (1.492 in^2), A_{14}=3.074x10^{-4} m^2 (0.4764 in^2), A_{15}=6.47x10^{-5} m^2 (0.1 in^2), A_{16}=6.516x10^{-5} m^2 (0.101 in^2), with a minimum weight of 169.621 kg (373.95 lb). The total amount of Function evaluation was 8035 with 4021 uphill acceptances. Again, only displacements constraint violations less than 1% were noticed for this solution. Figure 9 shows the behavior of the Weight and Temperature along iterations.

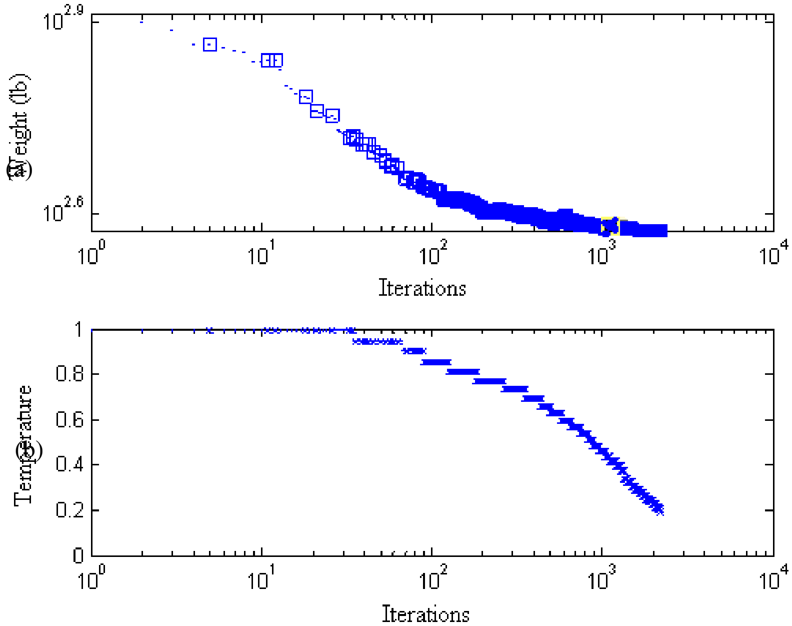

Figure 9. Objective Function Values and Temperature versus iteration for example 4. (a) Objective function. (b) Temperature.

The same problem was solved by Teles (2007) using Genetic Algorithm and SQP. The results for the cross sectional areas and total weight were listed in the following Tab. 6.

Table 6. GA Solution for the 72 member truss.

Group	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
Area m^2 (in^2)	0.7x10^{-4} (0.109)	3.7x10^{-4} (0.574)	3.2x10^{-4} (0.496)	3.5x10^{-4} (0.543)	3.5x10^{-4} (0.543)	3.4x10^{-4} (0.527)	0.6x10^{-4} (0.093)	0.7x10^{-4} (0.109)
Group	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}
Area m^2 (in^2)	7.1x10^{-4} (1.101)	3.4x10^{-4} (0.527)	0.9x10^{-4} (0.14)	0.8x10^{-4} (0.124)	10.2x10^{-4} (1.581)	3.2x10^{-4} (0.496)	0.7x10^{-4} (0.109)	0.7x10^{-4} (0.109)

Total Weight=166.37 kg (366.78 lb)

Table 7. SQP Solution for the 72 member truss.

Group	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
Area m^2 (in^2)	6.7x10^{-4} (1.0)	3.9x10^{-4} (0.61)	3.9x10^{-4} (0.61)	5.7x10^{-4} (0.88)	6.7x10^{-4} (1.0)	3.9x10^{-4} (0.61)	3.9x10^{-4} (0.61)	3.9x10^{-4} (0.61)
Group	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}
Area m^2 (in^2)	6.7x10^{-4} (1.0)	3.9x10^{-4} (0.61)	3.9x10^{-4} (0.61)	3.9x10^{-4} (0.61)	6.7x10^{-4} (1.0)	4.2x10^{-4} (0.65)	3.9x10^{-4} (0.61)	4.4x10^{-4} (0.68)

Total Weight=260.62 kg (574.57 lb)

In this example the GA performed better that the SA and this last one, in turn, performed better that the SQP Method.

3. FINAL REMARKS

This paper described the principles of the SA technique as a Heuristic tool. This algorithm was implemented in a Matlab Code and joined with an open source finite element code to perform optimizations. The objective function was the total weight of trusses. It was shown that SA may be used as an Optimization tool for weight minimization in trusses structures. The main advantage of this class of algorithms relies in the fact that it is not necessary to evaluate functions gradients.

The developed programs were applied to 4 examples that range from simple trusses to complex ones. It was intended to highlight the behavior of the algorithm in the optimizations. The same problems were compared with results from literature using GA, SQP and LP. One noticed disadvantage in the SA algorithm is the slowly convergence rate. In the average, the SA behaved similar to GA and in some cases gave better results than the SQP gradient based method.

4. ACKNOWLEDGEMENTS

The authors are grateful for the financial support by the Brazilians Research Councils CAPES and CNPq.

5. REFERENCES

Carrillo, O. J. B., 2007, “Algoritmo Híbrido para Avaliação da Integridade Estrutural: uma Abordagem Heurística”, (in Portuguese) Doctorial Thesis, Federal University of São Carlos, Brazil, 152p.
Delyon, B., 1988, “Convergence of the Simulated Annealing Algorithm”, LIDS, MIT, Cambridge, 01239, MA, USA, LIDS-P-1808, 16p.
Dréo, J., Pétrowski, A., Siarry, P., Taillard, E., 2003, “Metaheuristics for Hard Optimization”, Springer, New York, , 369p.
Erbatur,F., Hasançebi, O., Tütüncü, I., Hakan, K., 2000, “Optimal Design of Planar and Space Structures with Genetic Algorithms”, Computers and Structures, Cambridge, No. 75, p. 209-224.
Haftka, R. A., Gürdal, Z., 1991, “Elements of structural optimization: solid mechanics and its applications”, Springer, December, 500p.
Hasançebi¸O., Erbatur, F. , 2002 “Layout optimization of trusses using simulated annealing”, Advances in Engineering Software, V. 33, pp. 681–696.
Hasangebi, O.; Erbatur, F., 2002, “On efficient use of simulated annealing in complex structural optimization problems”, Acta Mechanica, No. 157, pp.27- 50.
Ingber, A. L., 1989, “Very fast simulated re-annealing”, J Mathematical Computational Modelling, V. 12, pp. 967-973.
Kirkpatrick, S., Gelatti Jr., C. D., Vecchi, M. P., 1983, “Optimization by Simulated Annealing”, Science, V. 220, No. 4598, pp. 671-680.
Locatelli, M., 2000, “Simulated Annealing Algorithms for Continuous Global Optimization: Convergence Conditions”, Journal Of Optimization Theory And Applications, V. 104, No. 1, pp. 121–133.
Pyrz, M.; Zawidska, J., 2001, “Optimal Discrete Truss Design Using Improved Sequential and Genetic Algorithm”, Engineering Computations, Bingley, V. 18, n. 8, p. 1078-1090.
Rajasekaran, S., 1990, “On the Convergence Time of Simulated Annealing”, Dept.. of Computer & Information Science, Technical Reports (CIS) MS-CIS-90-89, University of Pennsylvania, 19p.
Teles, M. L., 2007, “Avaliação de Desempenho de Algoritmos Genéticos Aplicados à Otimização Paramétrica de Estruturas de Engenharia”. (in Portuguese), Final Course Work, Mechanical Engineering Department, Porto Alegre, 20p.
Varanelli, J. M., 1996, “On the Acceleration of Simulated Annealing”, Doctor of Philosophy (Computer Science) University of Virginia, 133p.
Weck, O., Jilla, C., 2005, “Simulated Annealing: a Basic Introduction”, Massachusetts Institute of Technology, Internal Report, 44p.

6. RESPONSIBILITY NOTICE