RELIABILITY BASED DESIGN OPTIMIZATION OF TRUSSES BY HEURISTIC ALGORITHMS

Leandro Luis Corso, llcorso@ucs.br
Núcleo de Projeto e Fabricação, Universidade Caxias do Sul, Rua Francisco Getúlio Vargas, 1130, Caxias do Sul, RS, Brasil, http://www.ucs.br

Herbert Martins Gomes, herbert@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: *It is presented in this study a methodology for the optimization of trusses with reliability constraints. The RBDO (Reliability Based Design Optimization) was implemented in order to find the best set of bar areas that lead to a minimum truss weight, taking into account constraints in reliability index of the structure, strengths of material and service limits as maximum allowed displacements on nodes. In addition, some material properties are treated as stochastic ones. So the reliability index will be an additional constraint to the optimization problem. As methodology, a Genetic Algorithm (GA) with floating point codification, a Simulated Annealing algorithm (SA) and Sequential Quadratic programming (SQP a traditional mathematical programming technique) were used in the optimizations. The choice for the Heuristic Algorithms is justified by the ease in the implementation of such methods and its robustness when faced with highly non-linear cost functions. A simple example of a planar truss is modeled including statistical parameters regarding the material and cross sectional areas. This example is used as means for comparison of performance between the methods. The results are compared with similar ones in the literature. The average results shows that the heuristic algorithms performed better than the SQP.*

Keywords*: Reliability Based Design Optimization, Heuristic Algorithms, SQP.*

1. INTRODUCTION

The desire to reach optimal designs associated to production low cost and high efficiency were one of the main objectives chased by in the structural engineering. In this paper it is used the Reliability Based Design Optimization concept (RBDO) in order to optimize cost functions like the structural weight and, at the same time, ensuring a high level of structural reliability The reliability level have to be considered as an important ingredient in the structural design, however when this means structural analysis of large structures with a non-linear behavior, then some disadvantages may occur, most of time related to low computational efficiency, according to Pereira (2007). Ponslet and Maglaras (1995) described the application of a deterministic and a probabilistic genetic algorithm to RBDO in order to reduce the structural damping in a truss structure. The probabilistic formulation minimizes the probability to exceed some dynamic responses of the structure due to uncertainties in the damping properties. Dimou and Koumousis (2003) introduce competition between populations of a genetic algorithm. The algorithm had different parameters depending on the used as well as the reliability based design. Padmanabhan and Agarwal (2006) presented a methodology for optimization based on reliability using a Monte Carlo procedure. In spite of being robust, the method presented computationally expensive. The main inconvenience is the high number of functions evaluations in order to compute the failure probability. When the simulations are related to heavy structural analysis this makes the use of the method worthless, mainly when the structural analysis are required exhaustively as in the RBDO. According to Pereira (2007), this same problem can happen with the Importance Sampling technique for the reliability evaluations, for this reason it was chosen another methods to serve the objectives. A good alternative is to use FORM (First Order Reliability Method), which is capable to solve the majority of the practical cases, where the problem has multiple failure modes but well-known linearly and smooth functions. In this paper, the RBDO methodology was implemented in order to find the optimum parameters for minimum weight, maximum allowable deflections and limited stress in truss members. It was used the FORM method for the reliability index (Beta index) calculations and a Genetic Algorithm with floating point codification for the optimization. Such choice was taken in order to investigate the algorithm behavior and to ease the numerical implementation. Such method deals with cost functions with non-linear behavior, non-smooth or with functions where gradients are not defined. It is presented three simple examples of reliability based design optimization of trusses, as well as comparisons with literature results.

2. RELIABILITY BASED DESIGN OPTIMIZATION (RBDO)

RBDO is an optimization process which aims at the minimization/maximization of cost function and that satisfies reliability constraints as initial conditions. For this reason, it is necessary perform probability analysis during the optimization process. Besides, the design variables can be probabilistic parameters so that the optimization task became more complex.

The simpler and common formulation in the RBDO implementation is separated in two levels:

(a) An outer loop to perform the optimization, where the design variables are taken into account;

(b) An inner loop to perform the reliability analysis.

Generally speaking, an optimization model can be defined in the following way: Minimize (or Maximize) a cost function subjected to constraints. In a mathematical notation:

$$\text{Minimize } f(vp;\, p) \tag{1}$$

$$\begin{aligned} &\text{Subjected to} && \\ &g_i(vh;\, p) = 0 && i = 1...nr \\ &g_i(vh;\, p) < 0 && i = nr + 1...nr \\ &vhl < vh < vhu && i = 1...nv \end{aligned} \tag{2}$$

where *vh* are the design variables, *p* area the constant parameters of the problem, g_i is the *i-th* model constraint, *vhl* and *vhu* are respectively the lower and upper limit for the design variables. The deterministic optimization does not consider uncertainties in the design variables. Using the RBDO methodology, the constraints in the deterministic formulation are changed by probabilistic constraints. By the probability theory, it is well-known that the reliability index can be written as function of the failure probability as:

$$\beta = -\Phi(P_f(vp;p)) \tag{3}$$

where Φ is the cumulative probability function. In this work, the constraints can be stated as:

$$g_i^r(vp;p) = (\beta_c^f - \beta_e^f) \text{ where } g_i^r(vp;p) \le 0 \qquad i=1...m \tag{4}$$

$$g^f(vp;p) = (\beta_c^f - \beta_e^f) \text{ where: } g^f(vp;p) \le 0 \tag{5}$$

where g_i^r are the strength constraints for each of the i-th truss member, g_i^f is the displacements constraints for the structure. This means that the reliability index pre-set in the input parameters (*βe*) for both stress and displacements must be greater than the calculated reliability index (*βc*) during the optimizations. In order to find *β* it is used FORM. In this work, *vh* refers to cross sectional areas of truss members. *vhl* and *vhu* are the corresponding lower and upper limits for member's cross section.

3. GENETIC ALGORITHM IMPLEMENTATION

Genetic Algorithms (GA) are optimization techniques based on the Darwin's Theory of evolution and survival of the fittest. The Darwin's Theory of Natural Selection (1859) *apud* Goldberg (1989) says that "… any being, if it varies slightly in any manner profitable to itself, will have better chance of surviving…". GA simulates the evolutionary process numerically. They represent the parameters in a given problem by encoding them into a string. As in genetics, genes are constituted by chromosomes. Similarly, in simple GA, encoded strings are composed of bits. A string of bits can be decoded to the respective problem parameter value and the total evaluation of the string of bits for and individual may be weighted following some fitness function representing the phenotype to that string of bits.

A simple genetic algorithm consists of three basic operations (Holland, 1975), these being reproduction, crossover and mutation. The algorithm begins with a population of individuals each of them representing a possible solution of the problem. The individuals, as in nature, perform the three basic operations and evolve in generations where the Darwin's Theory prevails, or in other words, a population of individuals more adapted emerges as natural selection.

The floating point Coded Genetic Algorithms assumes real values to each variable. The main differences in this method are found on the crossover operator. There are several methods to deal with the floating point Coded Genetic Algorithms crossover such as flat crossover, simple crossover, arithmetical crossover, Wright's crossover, linear BGA crossover, etc. In this paper the BLX-α was used because it uses an initial exploration of the parameters field followed by an exploitation phase to improve resolution. It may be described by:

$$\begin{aligned} &\Delta = max[b_i(k), b_{i+1}(k)] - min[b_i(k), b_{i+1}(k)] \\ &b(k) = random\{min[b_i(k), b_{i+1}(k)] - \alpha\Delta,\ max[b_i(k), b_{i+1}(k)] + \alpha\Delta\} \end{aligned} \tag{6}$$

where, *i* and *i+1* are referred to two parents' chromosomes, α means a decreasing exploration parameter and random means a random number in the respective interval. Figure 2 summarizes the main steps followed by a Floating Point Coded basic Genetic Algorithm to maximize functions.

As indicated by Fig. 1, the algorithm starts generating a random set of individual that will form the population. In the following, individual are selected and picked on pairs according its fitness (objective function). This is accomplished by a probabilistic raffle called roulette wheel. At this point crossover will occurs, mixing chromosomes from two individual generating two offspring with characteristics inherited from its parents. The reproduction will be promising with a probability of 'Pc'. At last, from the offspring population, some chromosomes of some individual will suffer mutation under a probabilistic rate of 'Pm' usually set as a low value (1% or less) as found in nature. Then, eventually, some best individual belonging to the parent set will bypass the natural selection and will be introduced in the offspring set through an elitism procedure and exchanged by the less fitted offspring. This procedure assures that best solutions are hold and not lost in the probabilistic selection. The generations will succeed until a convergence criterion being reached. In this paper the stopping criteria is the diversity on the population set evaluated by the standard deviation of the objective function (fitness).

```
Step i) Initialize Time t=0, Initialize Population size: "m", mutation probability: "Pm", crossover probability: "Pc", number of individual  chromosomes: "nc", allowed limits for each chromosome: "Pmax(nc), Pmin(nc)".
Step ii) Generate Initial Population: $B_0 = (b_{1,0}, b_{2,0}, ..., b_{m,0})$
Step iii) While Stopping Condition is not fulfilled
  Step iii-1) "Proportional Selection"
  Loop i=1 to m
    x=random (0,1)
    k=1
    While k<m and x< $\sum_{j=1}^{k} f(b_{j,t}) / \sum_{j=1}^{m} f(b_{j,t})$
            k=k+1
            $b_{i,t+1} = b_{k,t}$
     End While
  End Loop
  Step iii-2) "One Point Crossover"
  Loop i=1 to m-1 step 2
   If random (0,1)<Pc then
     $\alpha = 0.5$
     $\Delta = max[b_{i,t}(k), b_{i+1,t}(k)] - min[b_{i,t}(k), b_{i+1,t}(k)]$
     $b_{i,t+1}(k) = random\{min[b_{i,t}(k), b_{i+1,t}(k)] - \alpha\Delta,\ max[b_{i,t}(k), b_{i+1,t}(k)] + \alpha\Delta\}$
     $b_{i+1,t+1}(k) = random\{min[b_{i,t}(k), b_{i+1,t}(k)] - \alpha\Delta,\ max[b_{i,t}(k), b_{i+1,t}(k)] + \alpha\Delta\}$
   End If
  End Loop
  Step iii-3) "Offspring Mutation"
  Loop i=1 to m
   If random (0,1)<Pm then
    k=random (0,1)*nc
     $b_{i,t+1}(k) = random\{P_{max}(k), P_{min}(k)\}$
   End if
  End Loop
End Loop
```

Figure 1. Pseudo-Code for the floating point genetic algorithm codification.

3. SEQUENTIAL QUADRATIC PROGRAMMING (SQP)

The Sequential Quadratic Programming (SQP) was developed in order to solve non linear-problems not solved by the Linear Programming (LP). The first paper published regarding the method was presented by Pshenichny in (1970) *apud* Belegundu and Chandrupatla (1999). Generally speaking, in each stage, a local quadratic model of optimization is built and solved and the partial results are used as start point for the next stage. This is performed till convergence criteria is met and optimum design variables are found.

The method is similar to the Newton's Method to unconstrained optimizations. The main idea is starts in the sub problem formulation in the quadratic programming that is based on quadratic approximations of the Lagrangean function.

$$L(x,\lambda) = f(x) + \sum_{i=1}^{m} \lambda_i . g_i(x) \tag{7}$$

The three main steps in the method are:

Step 1: Hessian matrix update:

At each iteration an Quasi-Newton Positive definite approximation (*H*) of the Lagrangean function is defined. And the Lagrangean multipliers are estimated λ*i* (*i*=1,...m).

Step 2: Solve Quadratic Sub-Problem;

At each SQP method iteration, the quadratic sub-problem, as stated below, is solved:

$$\begin{aligned} &\min q(d) = \frac{1}{2} d^T H d + c^T d \\ &\quad A_i d = b_i \quad i = 1,\ldots,m_e \\ &\quad A_i d \le b_i \quad i = m_e + 1,\ldots,m \end{aligned} \tag{8}$$

Step 3: *Line Search* and Cost Function Evaluation.

More details about the method can be found in Raftka and Gürdal (1991).

5. SIMULATED ANNEALING PRINCIPLES

The core of the Simulated Annealing 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 (Czeny, 1985). 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,

$$P(\Delta E) = e^{-\frac{\Delta E}{kT}} \tag{9}$$

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.

In the beginning 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$(E_2\text{-}E_1)/kT$. 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 (Kirkpatrick *et al.,* 1998).

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 Simulated Annealing 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.

A simplified sketch (pseudo-code) of the algorithm implementation is depicted in the following Figure.

```
Initialization (Current solution at $x_i$, current temperature $T_i$)
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 ($E_i$-$E_{i+1}$) ≤0 then
        $x_i = x_{i+1}$
    else
        If $e^{(\frac{E_i - E_{i+1}}{T_i})} > radom(0,1)$ 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 Stopping Criteria
    Decrease the Temperature following a cooling schedule $T_{i+1}$ =$R$ $T_i$
End of while
```

Figure 2. Pseudo-Code for Simulated Annealing Algorithm.

6. FIRST ORDER RELIABILITY METHOD (FORM)

A mathematical expression for the failure of a system may be stated in the following way:

$$M(x) = g(X_1, X_2, ..., X_n) \leq 0 \tag{10}$$

where M means the safety margin and X are the vector of n probabilistic variable that affects the material strength. $M \leq 0$ means failure and $M > 0$ means that the system is in the safe domain. The failure probability can be calculated using the jointly probability density function $f_x(X_1, X_2, ..., X_n)$ as (Melchers, 2001):

$$P_f = \iint \cdots \int_D f_x(X_1, X_2, ..., X_n) dX_1 dX_2 ... dX_3 \tag{11}$$

where D is the domain where $M \leq 0$. Taking into consideration a truss structure, where the failure function is defined by the stress σ_i in a member related to the allowed stress $\sigma_{\lim}$, the safety margin (limit state function) is written as:

$$M = 1 - \sigma_i / \sigma_{\lim} \tag{12}$$

This methodology can be used to compression stresses, displacements or any other function that indicates undesirable condition violations. Integration of Eq. (11) is difficult. Besides, some times statistical parameters of $f_x(\mathbf{X})$ are not known *a priori*. For this reason, it is usual use the first and the second moments (mean value and standard value) to calculate the reliability index β of the safety margin *M*. The known FORM (*First Order Second Moment*) uses an approximation of the limit state function in the vicinity of the design point in order to evaluate the β index. Besides, the FORM allows calculating the reliability index independent of the expression used as safety margin. For non-correlated variables, the random variables X_i can be transformed into standard non-correlated variables U_i taking:

$$U_i = \Phi^{-1}\left[F_{x_i}(X_i)\right] \tag{13}$$

where $F_{x_i}(X_i)$ and Φ are the cumulative distribution function and the Standard cumulative distribution function for variable X_i, respectively. In this way the safety margin in the real space can be transformed to the non-correlated standard space U :

$$H(\mathbf{U}) = M(\mathbf{X}) \tag{14}$$

A first order approximation of the limit state function on point $\mathbf{U}^*$ can be drawn and the gradient descent method can be used to find the smallest distance from approximated limit state function $H(\mathbf{U})=0$ to the origin of the non-correlated standard space $\mathbf{U}$. The $\mathbf{U}^*$ point is called design point and the reliability index β can be calculated as the Euclidian distance to the origin of the non-correlated standard normal space as:

$$\beta = \min(\mathbf{U}^{*^T} \cdot \mathbf{U}^*)^{1/2} \tag{15}$$

In order to solve this problem Eq. (16) it is used an iterative solution proposed by Rackwitz – Fiessler (1978) *apud* Haldar and Mahadevan (1999), which can be described as:

$$U^*_{i\,k+1} = \left[\nabla M(U^*_{i\,k})^t U^*_{i\,k} - M(U^*_{i\,k})\right] \nabla M(U^*_{i\,k}) / \left|\nabla M(U^*_{i\,k})\right|^2 \tag{16}$$

where ∇M is the limit state function gradient (safety margin), U is the vector of probabilistic variables in the non-correlated normal space.

In Eq.(16) all the variables in the real space are considered non-correlated. If there exist any correlation it may be used a Cholesky decomposition of the covariance matrix in order to transform from real space to non-correlated Normal space (Haldar and Mahadevan (1999)).

7. SOME TESTS AND RESULTS

A simple example, found in the literature Eboli and Vaz (2005), was modeled and solved in order to evaluate the results of the proposed methodology. The structure is a simple truss. It is intended to minimize the truss weight taking into account displacements and stress constraints and minimum reliability index as well. In this example there are 3 probabilistic variables, which are the Young Modulus of two materials and the corresponding material stress limit. All the variables are assumed non-correlated and having LogNormal distribution type. The design variables are the cross sectional areas for two sets of members of the truss.

Figure 3 shows a sketch of the analyzed structure:

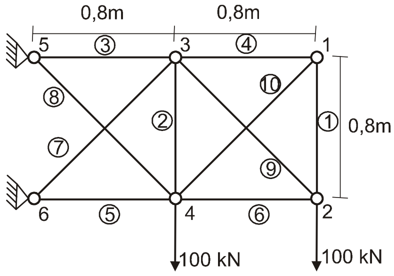

Figure 3. Simple truss structure analyzed in this work.

The model's mean values for Young Modulus were set as E_1= *200 GPa* and E_2=*210GPa* to members set 1 to 6 and 7 to 10, respectively. It was assumed a coefficient of variation of *COV*= 0,08 for the Young Moduli. The mean material stress limit was considered as 400 MPa and a coefficient of variation of *COV*= 0,08 and assumed equal for all the members of the structure. As limit state function it was assumed a limit vertical displacement value about 8 mm for node 2. The cross sectional areas comply with the upper and lower limits stated on Tab. 1. These two variables are the design variables for the optimization problem:

Table 1. Lower and upper limits for design variables.

Members	**Lower Limit (cm²)**	**Upper Limit (cm²)**
1 to 6	5	10
7 to 10	5	10

The behavior of the proposed algorithm can be summarized as: It is set a reliability index for the limit state function performance (β), which will become a new constraint to the optimization problem. The lower and upper limits for the design variables are design constraints, which will naturally treated by the optimization algorithms previously shown.

Using any of the cited algorithms for optimization the cross section areas values are searched in order to minimize the truss weight. At each genetic algorithm iteration, for example, the reliability index constraints, displacements and stress are checked to ensure that there will not exist violation, if so a penalty function is applied. The cost function to be minimized can be stated as:

$$\begin{aligned} &Min\, P/P_0(A_i) = \sum_{i=1}^{n} \rho_i A_i l_i / \sum_{i=1}^{n} \rho_i A_{i0} l_i \\ &Sujeito\ a \quad 1-\sigma_i/\sigma_{\max} < 0 \qquad i=1,\ldots,10 \\ &\qquad A_{\min} < A_i < A_{\max} \qquad i=1,\ldots,10 \\ &\qquad 1-d_{vertical\,node2}/8mm < 0 \\ &\qquad \beta_{\lim} - \beta \le 0\ (for\ stress\ limit\ \ or\ \ displcament\ limit) \end{aligned} \tag{17}$$

where A_{i0} are the member's cross sectional areas, ρ is the material density (as indicated by the reference it is assumed as unity since the cost function is defined as P/P_0), d means displacement and σ means stress. $\beta_{\lim}$ means limit value for the reliability index.

It was studied three cases. Each one of then it was changed the constraints for the reliability index. At first it was defined a reliability constraint for the stress limit in the members (*βcr*), latter a reliability index was set for the displacements limit state function (*βcv*) and at last it was set a reliability index for both stress and displacements limit simultaneously. For each of the cases it was analyzed the performance of the three optimization methods described previously, GA, SA and SQP.

In order to the better perform the optimization, it should be carefully chosen the heuristic parameters for the methods GA and SA. These are intrinsic parameters for each algorithm and problems in order to allow convergence and to present reliable results (non scattered results). Table 3 shows the used parameters for GA and SA algorithms.

Table 3. Heuristic parameters used by GA and SA.

Algorithm	P_m	**Crossover Method**	P_c	**Crossover percent**	**Max. Generations**	**Elitism (%)**	***n***	***R***	**Boltzman Constant**	**Stopping Creteria**	**Max. No. Func. Eval.**
GA	0.01	Heuristic	0.8	0.3	100	1	30	-	-	$1x10^{-4}$	20000
SA	-	-	-	-	-	1	-	0.85	1.0	$1x10^{-4}$	20000

7.1. Truss weight minimization using stress reliability index constraint.

In this first studied model, it was set the reliability index for strength, defined as *βcr*. It was required that *βcr*≥3.0 It was first used the genetic algorithm to find the optimum weight. The following Fig. 3 was obtained for the convergence of the method. Figure 3 shows mean value and best values into the population during generations for the truss weight. It was necessary 40 generations to reach a stable population and achieve convergence of the method, however the method was allowed to iterate further.

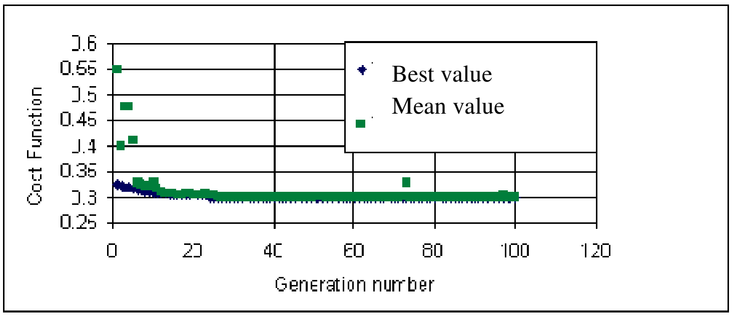

Figure 3. Convergence Figure for genetic algorithm and stress reliability index constraints.

As final optimum values it was found A_1 (cross sectional area for member 1 to 6) and A_2 (cross sectional areas for member 7 to 10) and the final stress reliability index for the design as indicated by Tab. 4. It can be verified that the stress reliability index converged to values close from the previously set. Latter the same problem was analyzed using do SQP and SA. For the SQP convergence it was necessary 16 iterations. It was necessary 14225 iterations for the SA convergence.

Table 4. Comparisons for the three used methods for stress reliability index constraint.

Method	**AG**		**SA**		**SQP**	
Área (cm^2)	**A1**	**A2**	**A1**	**A2**	**A1**	**A2**
	6.503	5.000	6.511	5.003	6.503	5.000
Volume (cm^2)	5385		5389		5384	
β_{cr}	3.043		3.000		2.999	

7.2. Truss weight minimization using displacements reliability index constraint.

First it was used the genetic algorithm to find the optimum cross sectional areas for minimum weight taking into account the displacements reliability index constraint. The obtained Fig. 4 shows the convergence iterations, indicating the best and mean values of truss weight into population for the GA method. In this example it was set $\beta cv \geq 3.0$. Analyzing the values it can be noticed that was necessary about just 6 generations to achieve convergence; however the method was allowed to iterate further.

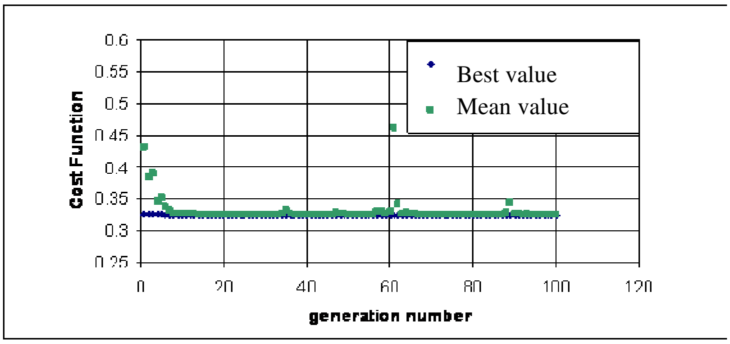

Figure 4. Convergence Figure for Best and Mean values of truss weight for GA Method and displacements reliability index constraint.

The cross sectional areas and the reliability index for this design are shown by Tab. 5. As in the previous example, the design variables converged to values close to the lower limit. It was necessary 18 iterations for SQP convergence and 14389 iterations for SA convergence.

Table 5. Comparisons for the three used methods for displacements reliability index constraint.

Method.	**AG**		**SA**		**SQP**	
Area (cm^2)	**A1**	**A2**	**A1**	**A2**	**A1**	**A2**
	7.059	5.532	7.093	5.498	6.718	5.945
Volume (cm^2)	6202		5893		5914	
β_{cv}	3.000		3.002		3.000	

7.3. Truss weight minimization using displacements and stress reliability index constraints.

In this example both displacement and stress reliability index constraints will be used in the truss weight minimization It is assumed $\beta cv \geq 3.0$ and $\beta cr \geq 3.0$. Firstly, the optimization was performed by the GA. Figure 5 shows best and mean values of truss weights during GA iterations (generations). It was necessary about 16 generations for the GA convergence, however the method was allowed to iterate further.

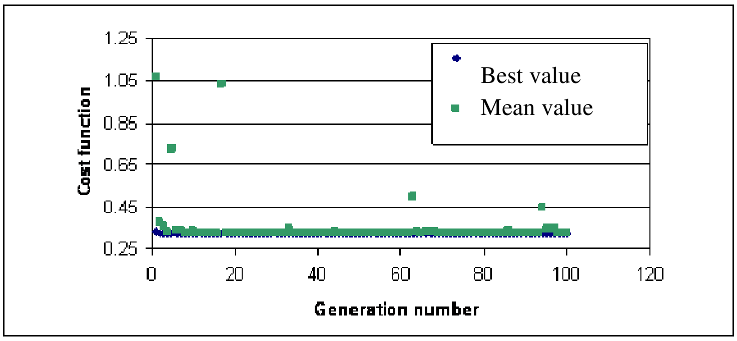

Figure 5. Convergence Figure for Best and Mean values of truss weight for GA Method and both displacements and stress reliability index constraint.

It can be noticed that the reliability index achieved the imposed constraint in the displacement failure mode and maintain a higher safety margin in stress failure mode. Table 6 shows the results for this last case. It was necessary about 15 iterations for the SQP convergence and about 13846 iterations for SA convergence.

Table 6. Comparisons for the three used methods for both displacements and stress reliability index constraint.

Method	**AG**		**SA**		**SQP**	
Area (cm²)	A1	A2	A1	A2	A1	A2
	7.065	5.525	7.049	5.543	6.872	5.744
Volume (cm²)	5898		5891		5930	
β ***cr***	4.420		4.02		3.320	
β ***cv***	3.000		3.000		3.000	

3. CONCLUSIONS

The application of the non-conventional optimization techniques to obtain optimum and reliable designs exhibited promising. The algorithms behavior was suitable since all the methods achieved very close results even for the literature results. The cost function is a linear function but the reliability imposes a high non linear constraint to the problem In this case, as shown by 1st. 2nd and 3rd examples, the SA and GA performed better than SQP, presenting light truss structures and keeping the probabilistic constraint without violation. Besides the SQP method converged to higher truss weights and sometimes with the probabilistic constraint violated (not complying constraint).

Comparing the obtained results, it can be noticed that in average SA achieved the best results in spite of the slight difference compared with the other methods. The genetic algorithm presented the second better performance, followed by the SQP method. Therefore the truss weight was lower using SA. Although presenting values of reliability index close to those previously set, the values of cross sectional areas arrangement suggested by the SQP method may result on heavy truss. This is clearly seen on the 3rd example where it was considered stress and displacements constraints.

3. ACKNOWLEDGEMENTS

The authors wish to acknowledge the financial support by the Brazilians Research Councils CAPES and CNPq.

4. REFERENCES

Belegundu, A. D., Chandrupatla, T. R., 1999, "Optimization Concepts and Applications in Engineering". P.-H. Inc.

Cerny, V., 1985, "Thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm". Journal Optimum Theory Applied, No.34.

Delyon, B., 1988, "Convergence of the Simulated Annealing Algorithm", LIDS, MIT, Cambridge, 01239, MA, USA, LIDS-P-1808, 16p.

Dimou, C. K., Koumousis, V. K., 2003. "Competitive genetic algorithms with application to reliability optimal design". Advances in Engineering Software, No. 34, pp.773–785.

Eboli, C. R., Vaz, L. E., 2005, "Optimization of Plane Trusses with Constraints in the Reliability Index Beta". 6th World Congresses of Structural and Multidisciplinary Optimization, pp. 30-36.

Goldberg, D. E., 1989, “Genetic Algorithms in search, optimization and machine learning”. R. Addison-Wesley Publishing, MA.

Haldar, A., Mahadevan, S., 1999, “Probability, Reliability and Statistical Methods in Engineering Design”. E. J. W. S.

Haldar,A., Mahadevan, S., 1999, “Probability, Reliability and Statistical Methods in Engineering Design”, Ed. Jonh Wiley & Sons

Holland, J., 1975, “Adaptation in natural and artificial Systems”. University of Michigan Press, MI, USA.

Ingber, A. L., 1989, “Very fast simulated re-annealing”, J Mathematical Computational Modelling, V. 12, pp. 967-973.

Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., 1998, “Optimization by Simulated Annealing”, Science, No. 45, pp.671-80.

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.

Melchers, R. E., 2001, “Structural Reliability Analysis and Prediction”. C. John Wiley & Sons Ltd, 2nd Edition.

Padmanabhan, D., H., Agarwal, A., 2006, “A study using Monte Carlo Simulation for failure probability calculation in Reliability-Based Optimization”. Business Media, Springer Science, No.7, pp.297–316.

Pereira, A., 2007, “Otimização baseada em confiabilidade: aplicação a treliças espaciais”.(in Portuguese). Doctoral Thesis, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Dept. de Engenharia Civil.

Ponslet, E., Maglaras, G., 1995, “Comparison of probabilistic and deterministic optimizations using genetic algorithms”. Structural Optimization, Springer-Verlag, No. 10, pp.247-257.

Raftka, R. A., Gürdal, Z., 1991, “Elements of structural optimization: solid mechanics and its applications”. Springer-Verlag.

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.

5. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.