

OTIMIZAÇÃO BASEADA EM CONFIABILIDADE PARA A MINIMIZAÇÃO DO PESO DE TRELIÇAS

Leandro L. Corso^a e Herbert M. Gomes^b

^a*Núcleo de Projeto e Fabricação, Universidade Caxias do Sul, Rua Francisco Getúlio Vargas, 1130
Caxias do Sul, RS, Brasil, lcorso@ucs.br, <http://www.ucs.br>*

^b*Grupo de Mecânica Aplicada, Universidade Federal do Rio Grande do Sul, Rua Sarmento Leite, 425,
Porto Alegre-RS, Brasil, herbert@mecanica.ufrgs.br, <http://www-gmap.mecanica.ufrgs.br>*

Palavras Chave: Confiabilidade, Algoritmos Genéticos, FORM, RBDO.

Resumo. Neste estudo, é apresentada uma metodologia de otimização baseada em confiabilidade. A Neste estudo, é apresentada uma metodologia de otimização baseada em confiabilidade. A metodologia de RBDO (*Reliability Based Design Optimization*) foi implementada para encontrar a melhor configuração em termos de áreas para estruturas treliçadas, com o objetivo de reduzir ao máximo o peso da estrutura respeitando certa confiabilidade (índice de confiabilidade Beta) assim como restrições de tensões nas barras e flechas admissíveis máximas. A confiabilidade é um dado de entrada e funcionará como uma restrição adicional ao problema de otimização. Como Metodologia de otimização é empregado o método dos Algoritmos Genéticos com codificação real de parâmetros. A opção pela escolha recai sobre facilidade de implementação e robustez do método frente ao tratamento de problemas com funções de custo não-lineares, ou onde cálculo de gradientes seja difícil. Um exemplo simples de estrutura treliçada é apresentado e comparado com resultados na literatura. Algumas estruturas utilizadas são hiperestáticas, sendo a função objetivo altamente não-linear. Aqui são testados métodos determinísticos e heurísticos para encontrar o melhor valor para função objetivo. É apresentado o comportamento de cada um dos métodos utilizados para otimização com a restrição em confiabilidade.

1 INTRODUÇÃO

O desejo de se obter projetos ótimos, conseguindo um menor custo e maiores desempenhos e melhor eficiência sempre foi um dos grandes objetivos da engenharia estrutural. Neste trabalho se utiliza a otimização baseada em confiabilidade (RBDO, do inglês *Reliability Based Design Optimization*), onde se visa otimizar a função custo de peso de estruturas simples e ao mesmo tempo garantindo certo nível de confiabilidade para o sistema estrutural.

RBDO deve ser considerada como um importante ingrediente no projeto de estruturas, entretanto quando envolve a análise da resposta de estruturas de grande porte, particularmente com comportamento não-linear podem surgir grandes limitações, geralmente relacionadas com a baixa eficiência computacional segundo [Pereira \(2007\)](#).

[Ponslet et al. \(1995\)](#) descrevem a aplicação do algoritmo genético determinístico e probabilístico para a RBD, visando aumentar o amortecimento de uma estrutura treliçada. A formulação probabilística minimiza a probabilidade de exceder o valor dos limites de certa resposta dinâmica de uma estrutura devido a incertezas nas propriedades do amortecimento.

[Dimou e Koumouisis \(2003\)](#), introduziram a competição entre as populações de um algoritmo genético. O algoritmo possuía diferentes utilizações para diferentes parâmetros. As aplicações foram de otimização baseada em confiabilidade no projeto.

[Padmanabhan et al. \(2006\)](#), apresentaram uma metodologia de otimização baseada em confiabilidade utilizando o método de Monte Carlo, apesar de ser bastante robusto, este método se mostra extremamente oneroso no que diz respeito ao custo computacional. O grande inconveniente de métodos deste tipo é o elevado número de vezes que a função objetivo precisa ser chamada para se poder obter a probabilidade de falha. Quando a simulação envolve análises estruturais pesadas, isso praticamente inibe o uso deste método, ainda mais em casos onde a função objetivo é chamada exaustivamente, como no caso da otimização baseada em confiabilidade. Segundo [Pereira \(2007\)](#), este mesmo problema pode também ocorrer com o método de Amostragem por Importância para cálculo de confiabilidade, por este motivo optou-se em buscar outro método que pudesse atender os objetivos.

Uma boa alternativa é utilizar o método FORM (*First Order Reliability Method*), que é suficientemente capaz de resolver a maioria dos casos práticos a despeito de suas limitações no tratamento de problemas com múltiplas funções de falhas ou funções de falhas não-lineares ou não suaves.

Aqui a metodologia de RBDO foi implementada de forma que se consiga encontrar parâmetros ótimos para estruturas treliçadas. Foram utilizados os métodos FORM para o cálculo do índice de confiabilidade (índice Beta) e o método de Algoritmos Genéticos (AG) com codificação real de parâmetros para realizar a otimização, tal escolha se deve pela facilidade de implementação e robustez do método quando trabalha com funções objetivo com comportamentos não-lineares, ou onde o cálculo do gradiente seja dispendioso.

O objetivo estabelecido é de reduzir o peso da estrutura, respeitando uma confiabilidade pré-determinada. Também são consideradas restrições de tensão nas barras e uma flecha máxima admissível para a estrutura. O índice de confiabilidade da estrutura é um dado de entrada e funcionará como uma restrição adicional ao problema a ser otimizado.

São apresentados exemplos simples da otimização com restrição em confiabilidade de treliças, assim como comparações com resultados da literatura. Uma estrutura hiperestática é testada, em razão do comportamento não-linear que apresenta para a função de custo estabelecida.

2 RBDO (RELIABILITY BASED DESIGN OPTIMIZATION)

Pode-se dizer que é um processo de otimização, onde o objetivo é otimizar uma determinada função objetivo, satisfazendo as restrições de confiabilidade colocadas como condição inicial. Para isso, é necessário realizarmos análises de probabilidade de falha durante o processo de otimização.

A formulação mais simples e usual de implementação da RBDO é aquela dividida em dois níveis: (a) o primeiro nível seria a realização do projeto ótimo, onde são consideradas as variáveis de projeto e (b) no segundo nível é realizado o processo de análise de confiabilidade, mais detalhes podem ser encontrados em [Melchers \(2001\)](#).

Geralmente um modelo de otimização determinística pode ser definido da seguinte maneira (veja equação 1):

$$\begin{aligned} & \text{Minimizar (ou Maximizar) Função objetivo} \\ & \text{(variáveis de projeto)} \end{aligned} \quad (1)$$

Tal que Restrições

Escrevendo da forma matemática, tem-se:

$$\text{Minimizar } f(vp; p) \quad (2)$$

sujeito a:

$$\begin{aligned} g_i(vh; p) &= 0 & i &= 1 \dots nr \\ g_i(vh; p) &\leq 0 & i &= nr + 1 \dots nr \\ vhl < vh &\leq vhu & i &= 1 \dots nv \end{aligned} \quad (3)$$

Onde, as vh são variáveis de projeto, p são os parâmetros fixos com relação ao problema de otimização, g_i é considerada a i -ésima restrição do modelo, vhl e vhu são os limites inferiores e superiores, respectivamente. Fazendo uma analogia a este trabalho, temos que vh para este trabalho se refere às áreas das barras das treliças, sendo que estas apresentam limites mínimos e máximos.

Entretanto, a otimização determinística não considera as incertezas nas variáveis e nem nos parâmetros de projeto. No método RBDO, as restrições da formulação determinísticas são trocadas por restrições de probabilidade. Pela teoria da probabilidade sabe-se que o índice de confiabilidade pode ser escrito em função da probabilidade de falha:

$$\beta = -\Phi(P_f(vp; p)) \quad (4)$$

onde Φ é a função cumulativa normal. Para este trabalho, as restrições ficam:

$$g_i^r(vp; p) = (\beta_c^r - \beta_e^r) \quad \text{onde sabe-se que: } g_i^r(vp; p) \leq 0 \quad i=1 \dots m, \quad (5)$$

$$g^f(vp; p) = (\beta_c^f - \beta_e^f) \quad \text{onde sabe-se que: } g^f(vp; p) \leq 0 \quad (6)$$

onde g_i^r são as restrições de resistência para cada uma das i -ésimas barras da treliça, e g^f é a restrição de flecha para estrutura. Isso quer dizer que se o índice de confiabilidade que estipularmos de entrada (β_e) tanto para resistência, como para flecha deve ser maior que o valor de confiabilidade (β_c) calculado no decorrer da otimização, sendo este β_c para resistência e flecha. Para encontrar o valor do β utilizaremos o método FORM, que será descrito no item 5.

Neste trabalho temos que vh se refere às áreas das barras das treliças, sendo que estas

apresentam limites mínimos e máximos.

3 IMPLEMENTAÇÃO DO MÉTODO DE ALGORITMOS GENÉTICOS

Os Algoritmos Genéticos (GA) representam, atualmente, uma poderosa ferramenta para busca de soluções de problemas com grandes complexidades e não-linearidades. Este método é utilizado na busca por mínimo-máximos de funções, tal busca é baseada na Teoria da Evolução e Sobrevivência, seguindo o que Darwin pressupunha, os indivíduos se modificariam de tal maneira que fosse proveitosa para si mesmo (Holland, 1975).

Os algoritmos genéticos (AGs) são uma classe de algoritmos evolutivos que tiveram sua origem no início da década de 70, através dos trabalhos desenvolvidos na Universidade de Michigan pelo professor John Holland (1975) e seus alunos. A partir dos trabalhos publicados por Holland, principalmente o livro “*Adaptation in Natural and Artificial Systems*” de 1975, a evolução pôde ser encarada como um processo de otimização e os algoritmos genéticos foram utilizados com sucesso em diversos problemas de otimização Fonseca e Neves (2004).

A implementação do método de AGs começa com uma população aleatória de cromossomos. Onde essas estruturas são avaliadas e associadas a uma probabilidade de reprodução de tal forma que as maiores probabilidades são associadas aos cromossomos que apresentam uma melhor (maior ou menor dependendo se maximização ou minimização) função de custo para o problema de otimização do que àqueles que representam uma solução pior. A aptidão da solução é tipicamente definida com relação à população corrente, sendo ela dada por uma função objetivo diretamente ou função de aptidão (uma modificação da função objetivo a fim de atender necessidades do processo de seleção dos indivíduos) maiores detalhes podem ser encontrados em Goldberg (1989). Os GA simulam o processo evolutivo numericamente. Eles representam os parâmetros de um dado problema codificando-s num vetor. Como na genética, genes são constituídos por cromossomos. Similarmente, em GA na sua forma simplificada, os vetores de são codificados em “bits”. Um vetor de “bits” pode ser decodificado no respectivo valor do parâmetro do problema e a avaliação total do vetor de “bits” para um indivíduo pode ser ponderado seguindo-se alguma função de aptidão representativa do fenótipo do indivíduo para aquele conjunto de “bits”.

Um algoritmo genético simples constitui-se de três operadores básicos, sendo eles, reprodução, “crossover” ou recombinação e mutação. O algoritmo começa com uma população de indivíduos cada um deles representando uma possível solução para o problema. Os indivíduos, como na natureza, se utilizam destes três operadores básicos e evoluem em gerações onde a Teoria de Darwin para a esta evolução prevalece, ou em outras palavras, uma população de indivíduos mais adaptados surge como uma seleção natural do processo.

Ao nível da reprodução, a avaliação da função objetivo (aptidão) indica quais indivíduos irão ter mais chances de procriar e gerar uma prole maior. Os valores simplesmente da função objetivo, dependendo do interesse, podem gerar valores negativos, fato este que inviabiliza o processo presente nos algoritmos genéticos, de escolha de indivíduos para procriação e geração dos elementos seguintes, visto que não há probabilidades negativas. Neste caso deve-se lançar mão de uma transformação adequada da função objetivo a fim de que a mesma venha a ter valores preferencialmente positivos. A função objetivo, assim calculada, permite o cálculo da aptidão de cada indivíduo, que fornecerá o valor a ser usado para o cálculo de sua probabilidade de ser selecionado para o processo de reprodução.

Nas operações genéticas, os genes de pares de indivíduos são trocados e como na natureza, esta troca pode dar-se de diversas formas sendo então comumente chamadas de “crossover” ou recombinação.

Diferenças entre técnicas convencionais e os algoritmos genéticos (AG) podem ser resumidas como segue:

- Os AG operam de forma codificada na tarefa de procura dos parâmetros;
- Os AG trabalham com uma população a qual representa os valores numéricos de uma variável e particular;
- Diferentemente na maioria dos algoritmos de otimização os quais requerem avaliações das derivadas da função objetivo, AG somente requer o uso do valor da função objetivo;
- Somente regras probabilísticas e a regra da seleção natural é utilizada com AG.

A representação binária tem uma importância histórica devido ao primeiro uso por Holland (1975). Quando se trabalha com o algoritmo genético na forma binária, cada um dos parâmetros reais b_i a serem otimizados são traduzidos num código binário seguindo a seguinte equação:

$$s = bin_n \left\{ round(2^n - 1) \frac{[b_i(k) - P_{\min}(k)]}{[P_{\max}(k) - P_{\min}(k)]} \right\} \quad (7)$$

onde, bin_n indica a tradução binária para um vetor binário s de n “bits”, n significando o número de “bits”, $P(k)$ significando o máximo e os mínimos valores permitidos para cada variável.

Para transformar o código binário para valores reais a seguinte equação é utilizada como indicado em seqüência:

$$b_i(k) = P(k)_{\min} + bin^{-1}(s) \frac{P(k)_{\max} - P(k)_{\min}}{2^n - 1} \quad (8)$$

onde, $bin^{-1}(s)$ significa a tradução dos valores codificados em binário para os respectivos valores reais. Deve ser notado que com esta formulação está implícito que o mapeamento tem uma resolução de $[P(k)_{\max} - P(k)_{\min}] / (2^n - 1)$. Isto restringe a procura no espaço dos parâmetros reais a valores discretos os quais podem induzir ótimos locais.

Isto pode ser contornado utilizando-se um algoritmo genético com codificação real. Esta aproximação assume valores reais para cada variável. As principais são encontradas no operador de “crossover” (recombinação). Há diversas formas de tratar com a recombinação em algoritmos genéticos com codificação real tais como: a recombinação plana, a recombinação simples, a recombinação aritmética, a recombinação de Wright, a recombinação BGA, etc. Neste trabalho a recombinação BLX- α (“blended crossover”) é utilizada devido ao fato dela operar inicialmente com uma fase exploratória abrangente do campo de parâmetros seguida de uma fase de pesquisa mais detalhada para melhoria da resolução dos parâmetros. Ela pode ser descrita como (veja equação 9):

$$\begin{aligned} \Delta &= \max[b_i(k), b_{i+1}(k)] - \min[b_i(k), b_{i+1}(k)] \\ b(k) &= \text{random}\{\min[b_i(k), b_{i+1}(k)] - \alpha \Delta, \max[b_i(k), b_{i+1}(k)] + \alpha \Delta\} \end{aligned} \quad (9)$$

onde, i e $i+1$ estão referidos a dois cromossomos de indivíduos, α significa um parâmetro de diminuição da exploração e random significa um número aleatório no respectivo intervalo. A Tabela 1 resume os principais passos seguidos por um algoritmo genético com codificação real para maximizar funções.

```

Inicialize o tempo t=0
Inicialize o tamanho da população "m", Probabilidade de Mutação "Pm", Probabilidade de
recombinação "Pc", Número de cromossomos individuais "nc", Limites permitidos para cada
cromossomo, "Pmax(nc), Pmin(nc)".
Geração da população inicial  $B_0 = (b_{1,0}, b_{2,0}, \dots, b_{m,0})$ 
"Laço sobre Gerações"
Enquanto a condição de parada não for satisfeita
  "Seleção Proporcional"
  Laço i=1 até m
    x=random(0,1)
    k=1
    Enquanto k<m e  $x < \sum_{j=1}^k f(b_{j,t}) / \sum_{j=1}^m f(b_{j,t})$ 
      k=k+1
       $b_{i,t+1} = b_{k,t}$ 
    Fim Enquanto
  Fim laço
  "Recombinação em um ponto"
  Laço i=1 até m-1 passo 2
    Se random(0,1)<Pc então
       $\alpha = 0.5$ 
       $\Delta = \max[b_{i,t}(k), b_{i+1,t}(k)] - \min[b_{i,t}(k), b_{i+1,t}(k)]$ 
       $b_{i,t+1}(k) = \text{random}\{\min[b_{i,t}(k), b_{i+1,t}(k)] - \alpha \Delta, \max[b_{i,t}(k), b_{i+1,t}(k)] + \alpha \Delta\}$ 
       $b_{i+1,t+1}(k) = \text{random}\{\min[b_{i,t}(k), b_{i+1,t}(k)] - \alpha \Delta, \max[b_{i,t}(k), b_{i+1,t}(k)] + \alpha \Delta\}$ 
    Fim Se
  Fim Laço
  "Mutação dos filhos"
  Laço i=1 até m
    Se random(0,1)<Pm então
      k=random(0,1)*nc
       $b_{i,t+1}(k) = \text{random}\{P_{\max}(k), P_{\min}(k)\}$ 
    Fim Se
  Fim Laço
Fim Enquanto

```

Tabela 1: Pseudo-código para Algoritmo Genético com codificação real.

4 MÉTODO DA PROGRAMAÇÃO QUADRÁTICA SEQUENCIAL

A programação quadrática sequencial (do inglês, *Sequential Quadratic Programming – SQP*) foi criada na tentativa de resolver problemas não-lineares. Teve a sua primeira publicação em 1970 na Rússia, por Pshenichny *apud* Belegundu e Chandrupatla (1999).

Em cada etapa, um modelo de otimização local do problema é construído e resolvido, a cada solução que se obtém, se avança na direção da solução do problema em questão, até que os critérios de convergência sejam atendidos e um ponto ótimo seja determinado.

Numa minimização sem restrição, apenas a função objetivo precisa ser aproximada, e o modelo local é quadrático. Como por exemplo, na programação não-linear.

O método é similar ao método de Newton para otimização sem restrições. A idéia principal

é a formulação de subproblemas de programação quadrática baseada em uma aproximação quadrática da função Lagrangeana.

$$L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i \cdot g_i(x) \quad (10)$$

São adotados basicamente três passos:

Passo 1: Atualização da Matriz Hessiana:

A cada iteração uma aproximação Quase-Newton definida positiva da Hessiana da função Lagrangeana H é calculada. Onde λ_i ($i=1, \dots, m$) é uma estimativa dos multiplicadores de Lagrange.

Passo 2: Solução de um subproblema de Programação Quadrática;

A cada laço do método SQP, um subproblema de Programação Quadrática é resolvido como é mostrado abaixo:

$$\begin{aligned} \min q(d) &= \frac{1}{2} d^T H d + c^T d \\ A_i d &= b_i \quad i = 1, \dots, m_e \\ A_i d &\leq b_i \quad i = m_e + 1, \dots, m \end{aligned} \quad (11)$$

Passo 3: Line Search e cálculo da função custo.

Mais informações sobre o método podem ser encontradas em [Raftka e Gürdal \(1991\)](#).

5 MÉTODO DE CONFIABILIDADE DE PRIMEIRA ORDEM (FORM)

Uma expressão matemática para a falha (quer seja última ou violação de determinada restrição) de um sistema, em geral, pode ser colocado da seguinte forma:

$$M(x) = g(X_1, X_2, \dots, X_n) \leq 0 \quad (12)$$

onde, M representa a margem de segurança e X as n variáveis que afetam a resistência do material. $M \leq 0$ significa falha e $M > 0$ que o sistema está no domínio de segurança. A probabilidade de falha pode ser calculada usando a função de densidade de probabilidade conjunta $f_x(X_1, X_2, \dots, X_n)$:

$$P_f = \iiint \dots \int_D f_x(X_1, X_2, \dots, X_n) dX_1 dX_2 \dots dX_n \quad (13)$$

onde D é a região onde $M \leq 0$.

Considerando uma estrutura de treliça onde, por exemplo, assume-se uma função de falha do tipo que a tensão de tração em uma barra σ_i da mesma não exceda um determinado valor σ_{lim} . A equação para a função de estado limite pode ser colocada (dentre outras formas) como:

$$M = 1 - \frac{\sigma_i}{\sigma_{lim}} \quad (14)$$

A mesma metodologia pode ser usada para tensões de compressão, flechas, ou qualquer outra condição que indique violação de alguma condição desejada na estrutura.

A integração da Equação (13) é difícil, pois se está trabalhando com diversas variáveis

aleatórias. Além disso, a função $f_x(\mathbf{X})$ geralmente não é conhecida se não houver dados estatísticos suficientes. Dessa forma, para obter um índice de confiabilidade β costuma-se usar o primeiro e o segundo momentos (média e variância) da margem de segurança M .

Quando a função M é linear e as variáveis aleatórias são independentes e normalmente distribuídas, o índice de confiabilidade β é definido como:

$$\beta = \frac{\mu_M}{\sigma_M} \quad (15)$$

onde, μ_M e σ_M representam respectivamente a média e o desvio padrão da função M . Quando a margem de segurança é não-linear, os valores aproximados de μ_M e σ_M são obtidos pela linearização da função M através da expansão da função em série de Taylor truncada no termo linear. O ponto no qual é feito o desenvolvimento da série de Taylor afeta os valores de μ_M e σ_M .

O método conhecido como FOSM (*First Order Second Moment*) utiliza as médias das variáveis aleatórias para desenvolver a série de Taylor, o valor de β pode ser obtido analiticamente, porém seu valor tende a ficar superestimado. Além disso, o método FOSM fornece um valor para o índice de confiabilidade que depende da expressão matemática utilizada para representar a função margem de segurança M , mesmo que as expressões forneçam valores idênticos. Dessa forma, um método que obtém o mesmo valor de β , independentemente da função utilizada para calcular a margem de segurança foi proposto por Hasofer-Lind e é conhecido como AFOSM (*Advanced First Order Second Moment*). No caso de variáveis aleatórias independentes, as variáveis aleatórias X_i são transformadas em variáveis normais padronizadas U_i fazendo-se:

$$U_i = \Phi^{-1}[F_{x_i}(X_i)] \quad (16)$$

onde, $F_{x_i}(X_i)$ e Φ são respectivamente a função de distribuição de probabilidade acumulada de X_i e a função de distribuição normal padrão. Dessa forma, a margem de segurança no espaço real X é transformada para o espaço Gaussiano padrão U :

$$H(\mathbf{U}) = M(\mathbf{X}) \quad (17)$$

A linearização da função de estado limite é feita em um ponto \mathbf{U}^* que tenha a menor distância entre a superfície de falha $H(\mathbf{U}) = 0$ e a origem do espaço \mathbf{U} . O ponto \mathbf{U}^* é chamado de ponto de projeto e β é calculado como a distância entre a origem e este ponto.

$$\beta = \min(\mathbf{U}^{*T} \cdot \mathbf{U}^*)^{1/2} \quad (18)$$

5.1 ALGORITMO DE RACKWITZ – FIESSLER

Para resolver o problema da Equação (18) utiliza-se o algoritmo proposto por Rackwitz – Fiessler (1978), que pode ser descrito da seguinte forma:

Passo 1: Definir a função que representa a margem de segurança M .

Passo 2: Assumir valores iniciais para o ponto de projeto X_i^* $i=1,2,\dots,n$, e calcular o correspondente valor da função M (considera-se aqui o valores iniciais do ponto de

projeto como sendo as médias das variáveis de projeto).

Passo 3: Calcular a média e o desvio padrão, no ponto de projeto, da distribuição normal equivalente para variáveis que não têm distribuição normal. As coordenadas do ponto de projeto no espaço normal padrão são:

$$U_i^* = \frac{X_i^* - \mu_{X_i}^N}{\sigma_{X_i}^N}$$

Passo 4: Calcular as derivadas parciais $\frac{\partial M}{\partial X_i}$ no ponto de projeto X_i^* .

Passo 5: Calcular as derivadas parciais $\frac{\partial M}{\partial U_i}$ no espaço normal padrão equivalente, usando a regra da cadeia.

$$\frac{\partial M}{\partial U_i} = \frac{\partial M}{\partial X_i} \cdot \frac{\partial X_i}{\partial U_i} = \frac{\partial M}{\partial X_i} \cdot \sigma_{X_i}^N$$

Passo 6: Calcular um novo valor para o ponto de projeto U_i^* no espaço normal padrão equivalente através da seguinte expressão:

$$U_{i\ k+1}^* = \frac{1}{|\nabla M(U_{i\ k}^*)|^2} [\nabla M(U_{i\ k}^*)^t U_{i\ k}^* - M(U_{i\ k}^*)] \nabla M(U_{i\ k}^*)$$

Passo 7: Calcular a distância deste novo ponto até a origem e estimar novo índice de confiabilidade

$$\beta = \sqrt{\sum_{i=1}^n (U_i^*)^2}$$

Passo 8: Verificar a convergência de β para um determinado valor de tolerância pré-determinado.

Passo 9: Calcular os valores do ponto de projeto no espaço original fazendo:

$$X_i^* = \mu_{X_i}^N + \sigma_{X_i}^N \cdot U_i^*$$

Passo 10: Calcular o valor de M para o novo ponto de projeto e verificar o critério de convergência para M .

Passo 11: Se ambos os critérios de convergência dos passos 8 e 10 forem satisfeitos, parar o algoritmo. Senão, repetir os passos de 3 a 11.

No algoritmo acima descrito, todas as variáveis do espaço original são assumidas como não correlacionadas. Caso exista correlação entre as mesmas calcula-se, via decomposição Cholesky, uma matriz de covariância que transforma as variáveis correlacionadas em não-correlacionados (Haldar e Mahadevan, 1999).

6 EXPERIMENTOS NUMÉRICOS E RESULTADOS

Com a finalidade de verificar as implementações realizadas, um modelo encontrado na literatura (Eboli e Vaz, 2005) foi utilizado. A estrutura é treliçada e hiperestática, apresentando uma função objetivo para minimização não-linear visto que restrições nas variáveis de projeto, na confiabilidade final da estrutura, tensões e deslocamentos são colocadas.

Abaixo se tem o desenho da treliça analisada:

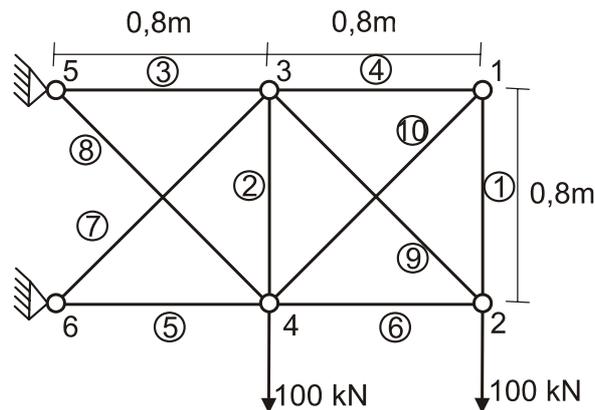


Figura 1: Estrutura treçada utilizada para validação no trabalho.

Os valores médios do módulo de elasticidade para este modelo foram de 200 GPa, com o coeficiente de variação de 0,08. O limite de escoamento considerado para o material foi 400 MPa. O limite máximo para flecha foi de 8 mm no nó 2.

As áreas das barras obedecem a limites de variação definidos pela Tabela 2. Estas são as duas variáveis de projeto do problema, as propriedades do material obedecem a uma distribuição do tipo Lognormal:

Barras	Limite inf. (cm ²)	Limite Sup. (cm ²)
1 a 6	5	10 cm ²
7 a 10	5	10 cm ²

Tabela 2: Limites inferiores e superiores para as áreas das barras.

O comportamento do algoritmo de otimização pode ser resumido da seguinte forma: a) designa-se um índice de confiabilidade (Beta) para a estrutura, o qual será uma restrição a mais para o algoritmo; b) definem-se os limites máximos para flechas assim como tensões nas barras, que também são restrições do problema. Utilizando os métodos de otimização anteriormente citados, os valores das duas variáveis da área são procurados com o intuito de minimizar o peso. Para cada iteração do algoritmo genético as restrições de confiabilidade, flecha e tensões são verificadas e caso haja violações, penalizações são aplicadas.

A função objetivo pode ser compreendida como:

$$\text{Min } P(p) = \sum_{i=1}^n \rho_i A_i l_i \quad (19)$$

Foram estudados três casos, onde em cada um deles, o que irá variar são as restrições, num primeiro momento visou-se a minimização do peso fixando um índice de confiabilidade para resistência do material (β_{cr}), posteriormente um índice de confiabilidade para o deslocamento (β_{cv}) e por final um índice de confiabilidade para a resistência e para o deslocamento ao mesmo tempo. Os três casos estão apresentados abaixo.

6.1 Minimização do peso de uma estrutura treçada utilizando restrição do índice de confiabilidade para tensão.

No primeiro modelo estudado, o índice de confiabilidade para resistência designado por β_{cr} , foi um dado de entrada, sendo que foi atribuído o valor $\beta_{cr} \geq 3.0$ para o mesmo, este valor foi utilizado como restrição do problema de otimização, onde a violação deste valor acarreta

em penalidades para o valor da função objetivo.

Inicialmente foi utilizado Método dos Algoritmos para encontrar o ótimo, o gráfico obtido para a convergência do algoritmo está apresentado na Figura 2, onde se podem analisar os valores dos melhores indivíduos (menores valores neste caso são os menores, pois o problema é de minimização) para cada geração, também se pode analisar o valor médio dos indivíduos a cada geração. Analisando os valores, pode-se concluir que foram necessárias 40 gerações para que o algoritmo convergisse para o ponto considerado ótimo.

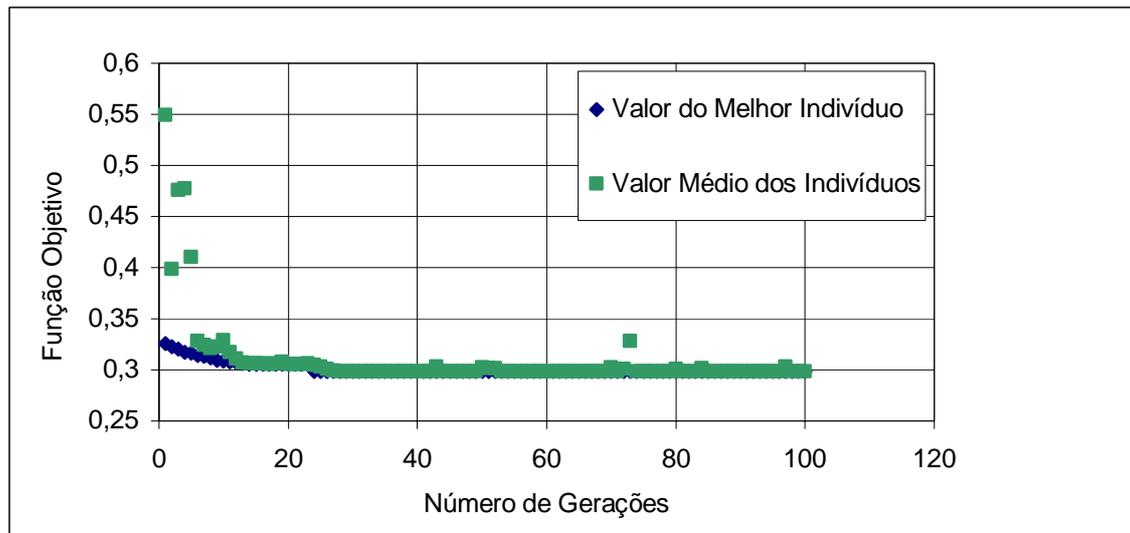


Figura 2: Gráfico de convergência do algoritmo considerando o índice de confiabilidade para deslocamento como restrição.

O valor das variáveis A_1 (área das barras 1 a 6) e A_2 (área das barras de 7 a 10) e o valor do índice de confiabilidade mediante a utilização das áreas encontradas estão apresentadas na Tabela 3.

Variáveis	A_i
A_1	6,52 cm ²
A_2	5,00 cm ²
β_{cr}	3,043

Tabela 3: Valores encontrados utilizando Algoritmos Genéticos

Pode-se verificar que o valor do índice de confiabilidade encontrado ficou muito próximo do índice que foi estipulado como restrição. Posteriormente, o mesmo modelo foi analisado utilizando SQP, foi criada uma Tabela onde se mostram estes valores (Tabela 4), também indicando que o SQP tem um bom comportamento para o tipo de função objetivo. As tolerâncias tanto para as variáveis como para a função objetivo foi de 1.0×10^{-3} , sendo necessárias 16 iterações para a convergência.

Variáveis	A_i
A_1	6,50 cm ²
A_2	5,00 cm ²
β_{cr}	2,999

Tabela 4: Valores encontrados utilizando SQP para o primeiro caso analisado.

6.2 Minimização do peso de uma estrutura treliçada utilizando restrição do índice de confiabilidade para deslocamento.

Inicialmente foi utilizado Método dos Algoritmos Genéticos para encontrar o ótimo. O gráfico obtido para a convergência do algoritmo está apresentado na [Figura 3](#), onde pode-se analisar os valores dos melhores indivíduos (menores valores neste caso são os menores, pois o problema é de minimização) para cada geração, também pode-se analisar o valor médio dos indivíduos a cada geração. Neste exemplo assumiu-se $\beta_{cv} \geq 3.0$. Analisando os valores, pode-se concluir que foram necessárias 6 gerações para que o algoritmo convergisse para o ponto considerado ótimo.

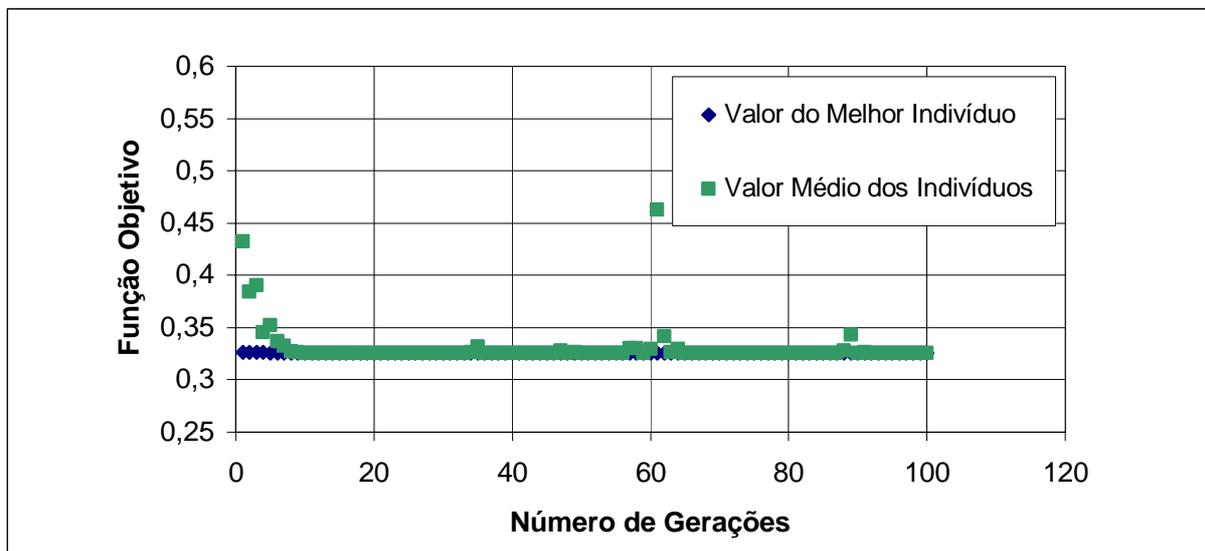


Figura 3: Gráfico de convergência do algoritmo considerando o índice de confiabilidade para tensão como restrição.

O valor das áreas e o valor do índice de confiabilidade mediante a utilização das áreas encontradas estão apresentados na [Tabela 5](#).

<i>Variáveis</i>	<i>A_i</i>
A ₁	7,05 cm ²
A ₂	5,53 cm ²
β_{cv}	3,00

Tabela 5: Valores encontrados utilizando Algoritmos Genéticos

Pode-se verificar que o valor do índice de confiabilidade atingiu a restrição imposta.

Posteriormente, o mesmo modelo foi analisado utilizando SQP, foi criada uma Tabela que mostra estes valores (Tabela 6), também indicando que o SQP tem um bom comportamento para o tipo de função objetivo. As tolerâncias tanto para as variáveis como para a função objetivo foi de 1.0×10^{-3} , sendo necessárias 18 iterações para a convergência.

<i>Variável</i> <i>s</i>	<i>A_i</i>
A ₁	6,71 cm ²
A ₂	5,94 cm ²
β_{cv}	3,00

Tabela 6: Valores encontrados utilizando SQP para o segundo caso analisado.

6.3 Minimização do peso de uma estrutura treliçada utilizando restrição do índice de confiabilidade para tensão e do índice de confiabilidade para deslocamento.

Neste exemplo assumiu-se $\beta_{cv} \geq 3.0$ e $\beta_{cr} \geq 3.0$. Inicialmente foi utilizado Método dos Algoritmos para encontrar o ótimo, o gráfico obtido para a convergência do algoritmo está apresentado na Figura 4, onde se podem analisar os valores dos melhores indivíduos (menores valores neste caso são os menores, pois o problema é de minimização) para cada geração, também se pode analisar o valor médio dos indivíduos a cada geração. Analisando os valores, pode-se concluir que foram necessárias 46 gerações para que o algoritmo convergisse para o ponto considerado ótimo.

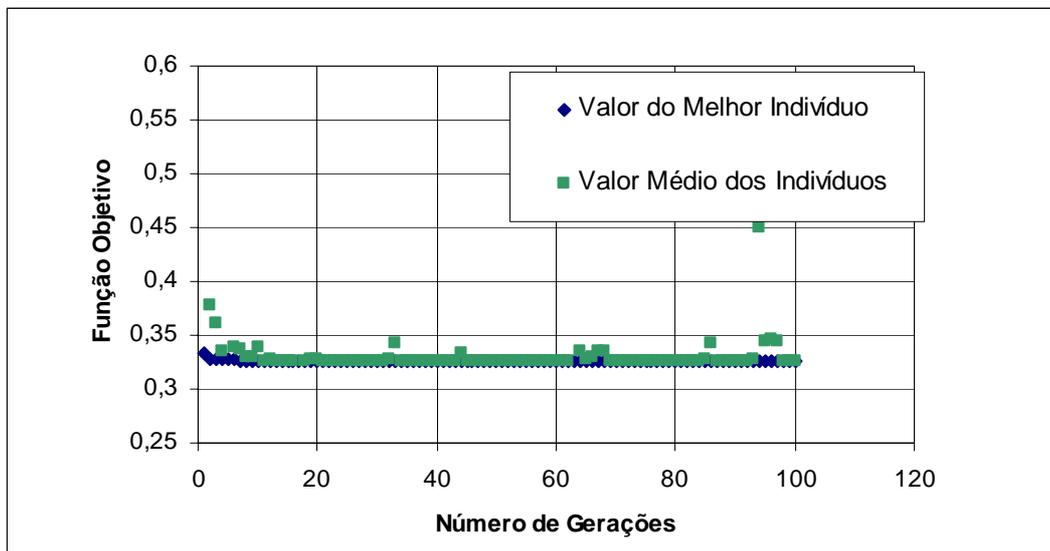


Figura 4: Gráfico de convergência do algoritmo considerando o índice de confiabilidade para tensão e índice de confiabilidade para deslocamento como restrição.

Os valores das áreas e o valor do índice de confiabilidade mediante a utilização das áreas encontradas estão apresentados na Tabela 7.

<i>Variável</i> <i>s</i>	<i>A_i</i>
A ₁	7,26 cm ²

A_2	5,32 cm ²
β_{cr}	4,42
β_{cv}	3,00

Tabela 7: Valores encontrados utilizando Algoritmos Genéticos

Pode-se verificar que o valor do índice de confiabilidade atingiu a restrição imposta no caso do deslocamento e ficou com certa segurança para a função de estado limite de tensão.

Posteriormente, o mesmo modelo foi rodado utilizando SQP, foi criada uma Tabela que mostra estes valores (Tabela 8), também mostrando que o SQP tem um bom comportamento para o tipo de função objetivo. As tolerâncias tanto para as variáveis como para a função objetivo foi de 1.0×10^{-3} , sendo necessárias 15 iterações para a convergência.

<i>Variáveis</i> s	A_i
A_1	6,62 cm ²
A_2	6,07 cm ²
β_{cr}	3,32
β_{cv}	3,00

Tabela 8: Valores encontrados utilizando Algoritmos Genéticos

Comparando-se o resultado obtido pelo Algoritmo Genético e o obtido por SQP, observa-se que o peso total da treliça com os resultados do Algoritmo Genético gera um peso menor que o resultado obtido por SQP. Em outras palavras, apesar do SQP fornecer valores de β mais próximos dos limites inferiores impostos como restrição, as áreas das seções transversais, podem vir a gerar estruturas mais pesadas em relação ao AG. Este caso ficou claro no estudo do terceiro caso, onde foi utilizado como restrição o índice de confiabilidade para tensão e para deslocamento. Na Tabela 9 estão mostrados os valores de volume encontrados para cada um dos casos otimizados.

Casos	Rest. β tensão		Rest. β deslocamento		Rest. β tensão e β desl.	
	AG	SQP	AG	SQP	AG	SQP
Volume (cm ³)	5394	5384	6202	5914	5898	5930

Tabela 9: Valores de volume encontrados para cada um dos casos analisados.

7 CONCLUSÕES

A aplicação de técnicas de otimização para a obtenção de projetos mais econômicos e estruturalmente confiáveis se mostrou bastante eficiente. O comportamento dos algoritmos foi satisfatório tendo em vista que ambos os métodos chegaram a resultados muito próximos, encontrando-se uma configuração de áreas das seções transversais que minimiza o volume mas que ainda assim mantém o índice de confiabilidade utilizado como restrição dentro dos valores estabelecidos (projeto ótimo, mas ainda assim confiável).

AGRADECIMENTOS

Os autores desejam agradecer ao Programa de Pós Graduação em Engenharia Mecânica, PROMEC da UFRGS pelos auxílios indiretos que viabilizaram a realização deste trabalho, bem como a UCS pelo apoio e incentivo a pesquisa.

REFERÊNCIAS

- Belegundu A.D. e Chandrupatla T.R. Optimization Concepts and Applications in Engineering. *P.-H. Inc.*, 1999.
- Dimou C.K. e Koumousis V.K. Competitive genetic algorithms with application to reliability optimal design. *Advances in Engineering Software* 34 / 773–785, 2003.
- Eboli C.R. e Vaz L.E. Optimization of Plane Trusses with Constraints in the Reliability Index Beta. *6th World Congresses of Structural and Multidisciplinary Optimization* Pages: 30 May - 03 June, 2005.
- Fonseca M. e Neves F.A. Algoritmos Genéticos Aplicados à Otimização Discreta de Estruturas Treliçadas Espaciais. *U. F. de Ouro Preto*, 2004.
- Goldberg D.E. Genetic algorithms in search, optimization and machine learning. *R. Addison-Wesley Publishing, MA*, 1989.
- Haldar A. and Mahadevan S. Probability, Reliability and Statistical Methods in Engineering Design. *E. J. W. S. Cap. 3*, 1999.
- Holland J. Adaptation in natural and artificial systems *A. A. University of Michigan Press, MI*, 1975.
- Melchers R.E. Structural Reliability Analysis and Prediction. *C. John Wiley & Sons Ltd, second edition*, 2001.
- Padmanabhan D., Agarwal H., Renaud J.E., e Batill S.M. A study using Monte Carlo Simulation for failure probability calculation in Reliability-Based Optimization. *Springer Science, Business Media, LLC* 7:297–316, 2006.
- Pereira A. Otimização baseada em confiabilidade: aplicação a treliças espaciais. *Departamento de Engenharia Civil - Tese de Doutorado*. 2007.
- Ponslet E., Maglaras G., Haftka R.T., Nikolaidis E., e Cudney H.H. Comparison of probabilistic and deterministic optimizations using genetic algorithms. *Structural Optimization* 10, 247-257 *Springer-Verlag*, 1995.
- Raftka R.A. e Gürdal Z. Elements of structural optimization: solid mechanics and its applications. *Springer*, 1991.