



ANÁLISE DA CONFIABILIDADE E EFICIÊNCIA DO PROGRAMA COMPUTACIONAL OTR

Dogmar Antonio de Souza Junior

Doutorando – Faculdade de Engenharia Mecânica - Universidade Federal de Uberlândia – Rua João Naves de Ávila nº 2121 Campus Santa Mônica – Uberlândia
Professor – Universidade de Uberaba – Rua: Coronel Severiano 251 – Uberlândia
dogmar.souza@uniube.br

Francisco Antonio Romero Gesualdo

Professor – Faculdade de Engenharia Civil – Universidade Federal de Uberlândia - Rua João Naves de Ávila nº 2121 Campus Santa Mônica – Uberlândia
gesualdo@ufu.br

Resumo: *A utilização de um novo programa computacional exige que o mesmo seja amplamente testado. Em se tratando de um programa de otimização, é necessário submetê-lo aos mais variados problemas, cada qual com suas dificuldades específicas para determinação do valor ótimo da função objetivo. Neste trabalho apresentar-se-á os resultados obtidos da análise da confiabilidade e eficiência do software OTR. Ele foi escrito em linguagem DELPHI® e o método de busca utilizado é o algoritmo genético. Neste sentido, foram otimizadas quatro funções matemáticas sem restrições e dois problemas de engenharia, um com restrição e outro sem restrição, descritos na literatura como clássicos. Os resultados obtidos mostram que o software OTR é confiável e que possui robustez em problemas de multimodalidade e com ruído.*

Palavras-chave: *otimização, algoritmo genético, confiabilidade.*

1. INTRODUÇÃO

Nas últimas décadas, o forte crescimento industrial impulsionado pela evolução computacional, tanto no nível de hardware quanto de software, exigiram da ciência novas metodologias de análise e otimização. O objetivo é projetar o sistema com mais eficiência e menor custo. Dessa maneira, para cada problema define-se as variáveis de projeto e uma função objetivo que deverá ser otimizada, ou seja, maximizada ou minimizada, num contexto que pode existir restrições. Neste sentido, a otimização visa determinar a melhor correlação entre as variáveis de projeto sem testar todas as possibilidades possíveis.

Esta busca deve atender uma característica muito importante para qualquer método de otimização: o equilíbrio entre eficiência e eficácia de forma a garantir não só a correta solução do problema, mas também a generalidade do processo (Souza and Mendes Neto, 2001). Neste caso, um processo é dito eficiente, quando ele atinge o seu objetivo com menor esforço computacional possível. A eficácia da busca pelo ótimo está associada a generalidade do método, isto é, não há necessidade de particularização ou alteração do processo de otimização para a resolução dos mais diversos problemas, à custa de adaptações impostas ao processo de otimização inicialmente idealizado. Um processo com estas características é dito robusto.

A otimização é uma das ferramentas mais importantes da atualidade, empregada nas mais diversas áreas da engenharia. Neste contexto, diversas técnicas de otimização têm sido empregadas na resolução dos mais variados problemas. De modo geral, todas elas experimentam, em maior ou menor grau, dificuldades relacionadas a não convergência e à existência de mínimos locais da função custo.

Como vantagens no uso de procedimentos de otimização podem ser citados os seguintes aspectos:

- Diminuição do tempo dedicado ao projeto;
- Possibilidade de tratamento simultâneo de uma grande quantidade de variáveis e restrições de difícil visualização gráfica ou tabular;
- Possibilidade de obtenção de algo melhor (mesmo que não seja o mínimo global);
- Redução de custo em relação aos modelos experimentais.

E como desvantagens no uso da otimização podem ser consideradas algumas dificuldades para a sua aplicação, tais como:

- Aumento do tempo computacional;
- As funções envolvidas são descontínuas (ou gradientes descontínuos) ou de lenta convergência;
- Presença de muitos mínimos locais (mínimo global raramente é obtido);
- Programas de análise não adequados;

Para vencer as desvantagens no uso da otimização foram desenvolvidos diversos métodos de busca. A escolha de um ou outro método de busca deve estar estritamente relacionada com o problema a ser minimizado.

Durante as últimas décadas, surgiram diferentes técnicas relacionadas aos algoritmos evolucionários, ou seja, métodos de busca estocásticos que imitam a evolução biológica natural. Dentre eles encontra-se o método dos AGs inventado por John Holland. Os AGs operam num conjunto de soluções aplicando o princípio da sobrevivência dos indivíduos mais aptos para produzir uma solução cada vez melhor. Dessa forma, geração após geração, um novo conjunto de indivíduos é criado pelo processo de seleção dos indivíduos progenitores, de acordo com o nível de aptidão associado ao problema. A procriação é feita entre os progenitores através de operadores extraídos da genética natural. Este processo leva à evolução de indivíduos que se adaptam melhor ao problema, tal como na adaptação natural. Os AGs modelam os processos naturais, através de operadores básicos tais como seleção, cruzamento, mutação, etc (Goldberg, 1989).

O uso dos AGs não garante uma solução ótima, mas na maioria das vezes encontra-se uma solução quase ótima que, satisfaz os critérios de parada do método. Além disto, são aplicados numa grande variedade de problemas, pois não impõem limitações que são encontradas, na maioria das vezes, nos métodos de busca tradicionais (Castilho et al, 2001). Suas principais vantagens são:

- Trabalham sobre um conjunto de pontos do espaço de busca ao invés de um único ponto;
- Podem trabalhar sobre a codificação dos parâmetros, ao invés de trabalhar sobre os próprios parâmetros;
- Não exigem a continuidade e a diferenciabilidade da função objetivo;
- São fáceis de serem implementados;
- São flexíveis para otimizar múltiplas funções com objetivos conflitantes;

Devido às suas características, os AGs demonstram ser mais robustos que os métodos tradicionais, sendo este o principal motivo do crescente emprego destes algoritmos no estudo de problemas de engenharia.

O programa computacional OTR é baseado nos métodos dos algoritmos genéticos (AGs) e foi desenvolvido na linguagem DELPHI com o objetivo específico de minimizar o consumo de tubos de uma estrutura espacial de cobertura de duas águas, denominada estrutura lamelar de eixo reto. Entretanto, a utilização de um novo programa computacional, sempre requer que este seja submetido a diversos testes para avaliar sua confiabilidade.

Sendo assim, o objetivo deste trabalho é apresentar os resultados obtidos na análise da eficiência e da robustez a multimodalidade e ao ruído, do programa computacional OTR. Para tanto, foram avaliadas quatro funções matemáticas sem restrição e, mais dois problemas de engenharia, descritos na literatura como clássicos, um sem restrição e o outro com restrição, cujas soluções são conhecidas na literatura ou foram retiradas de teses que trabalham com métodos tradicionais de busca.

2. AVALIAÇÃO DA CONFIABILIDADE E EFICIÊNCIA DO SOFTWARE “OTR”

A Tabela 1 apresenta os valores utilizados neste trabalho para os parâmetros do AG.

Tabela 1: Valores para os parâmetros de entrada do AG.

Gerações	5000
População	50
Número de operadores	5
Taxa de Recombinação Discreta	0,80
Taxa de mutação	0,05

Para a execução do programa computacional OTR utilizou-se o seguinte micro-computador:

- Pentium - 1,6 GHz
- 256 Mb de memória RAM
- Sistema Operacional - Windows XP

Em cada problema estudado serão apresentados a função objetivo e o domínio de cada variável. Em situações onde os dados de entrada não coincidem com os valores dispostos na Tabela 1, será feito um comentário explícito da modificação.

2.1 Minimização de uma função matemática – Caso 1

No primeiro caso deseja-se minimizar a função objetivo de duas variáveis:

$$f(x_1, x_2) = 10x_1^4 - 20x_1^2x_2 + 10x_2^2 + x_1^2 - 2x_1 + 5 \quad (1)$$

Adotou-se o seguinte domínio para as variáveis:

$$\begin{cases} -2.00 \leq x_1 \leq 2.00 \\ -2.00 \leq x_2 \leq 2.00 \end{cases}$$

Tabela 2: Resultados da minimização da função matemática (1).

	Teórico	Faria	OTR
f(x)	4	4.001	4.00000
x1	1	0.969	1.00058
x2	1	0.936	1.00119

A Tabela 2 apresenta os resultados obtidos. O problema foi executado apenas uma vez, pois os resultados da primeira tentativa foram bastante satisfatórios. Foram necessárias 413 gerações e o tempo computacional gasto neste caso foi de apenas 2 segundos.

2.2 Minimização de uma função matemática – Caso 2

Para avaliar a robustez do algoritmo genético à multimodalidade estudou-se a minimização de duas funções matemáticas, caso 2 e 3.

$$F(x, y) = 3(x-1)^2 e^{(-x^2-(y+1)^2)} - 10\left(\frac{x}{5} - x^3 - y^5\right) e^{(-x^2-y^2)} - \frac{1}{3} e^{(-(x+1)^2-y^2)} \quad (2)$$

$$\text{no domínio:} \quad \begin{cases} -3 \leq x \leq 3 \\ -3 \leq y \leq 3 \end{cases}$$

O mínimo global desta função neste intervalo é conhecido:

$$x_{min} = 0.2282 \quad y_{min} = 1.6260 \quad f_{min} = -6.55113$$

A Figura 1 ilustra o gráfico da função $F(x,y)$.

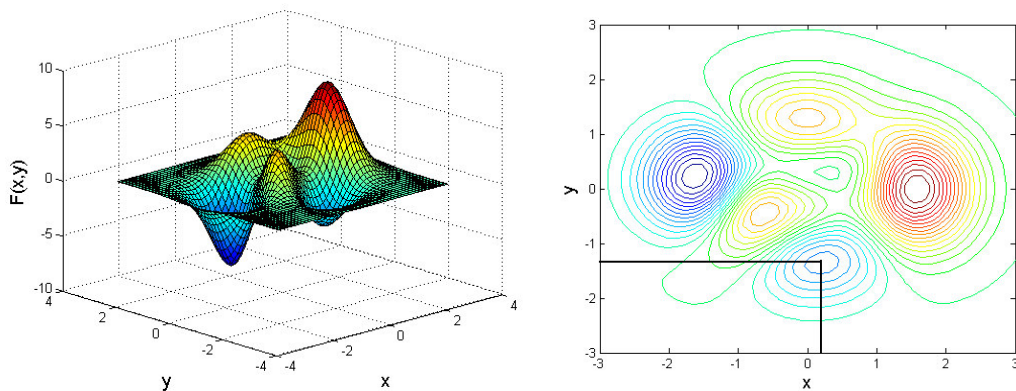


Figura 1: Função utilizada na avaliação da robustez à multimodalidade - Caso2.

Tabela 1: Resultados dos testes de otimização empregando os métodos BFGS e AGs.

	Método tradicional (BFGS)					Algoritmo genético (OTR)			
	xo	yo	xf	yf	F	pop	xf	yf	F
1	-2.0	-2.0	-3.0	-3.0	6.67×10^{-5}	20	0.2281	-1.6267	-6.5511
2	-1.4	0.4	-1.32	0.3	-2.8807	20	0.2271	-1.6255	-6.5511
3	0.2	1.7	3.0	3.0	4.10×10^{-5}	50	0.2283	-1.625	-6.5511
4	2.5	2.5	3.0	3.0	4.10×10^{-5}	50	0.2266	-1.6302	-6.5508
5	-2.0	2.0	0.2282	-1.6260	-6.55113	100	0.2283	-1.6264	-6.5511
6	0.3	0.5	0.2282	-1.6260	-6.55113	-	-	-	-
7	2.5	-2.0	0.2282	-1.6260	-6.55113	-	-	-	-

A Tabela 3 apresenta os resultados obtidos por Silva (1999) utilizando o método tradicional BFGS do otimizador do programa computacional MATLAB® e os resultados obtidos empregando o

software OTR. Foram executados 5 testes empregando o programa computacional OTR com diferentes números de indivíduos para a população. Os resultados mostram que o sucesso dos métodos tradicionais na busca da solução ótima em funções multimodais é fortemente influenciado pelo ponto inicial (x_0) de busca fornecido pelo usuário. Já o software OTR encontrou a solução ótima em todos os testes realizados com um tempo computacional médio de 10 segundos, e sempre com menos de 600 gerações.

2.2 Minimização de uma função matemática – Caso 3

Seja a seguinte função matemática (Haupt e Haupt, 1998):

$$F(x, y) = x \sin(4x) + 1.1y \sin(2y) \quad (3)$$

o objetivo é minimizar esta função no domínio: $\begin{cases} 0 \leq x \leq 10 \\ 0 \leq y \leq 10 \end{cases}$

Pela Figura 2 observa-se que a função possui vários ótimos locais.

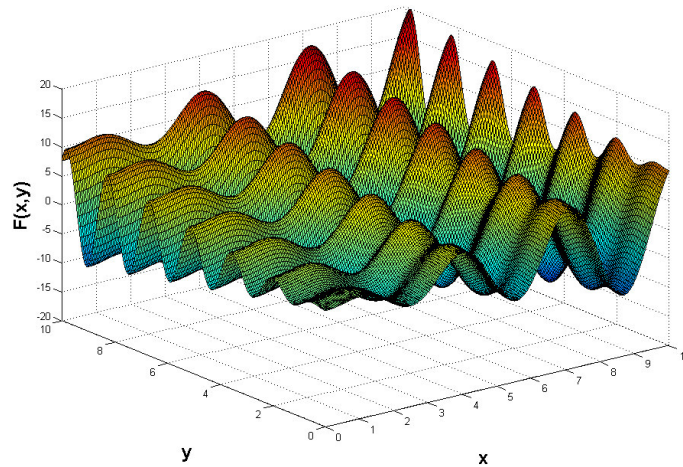


Figura 2: Função utilizada na avaliação da robustez à multimodalidade - Caso 3.

Para o caso 3, todas as execuções foram realizadas com 500 gerações. O tempo médio computacional foi de 14s. Os resultados obtidos com o programa computacional OTR são apresentados na Tabela 4 juntamente com os encontrados na literatura.

Tabela 4: Resultados obtidos para o Caso 3.

	Haupt e Haupt (1998)	GAOT V	OTR
$F(x, y)$	-18.5	-18.5547	-18.55472
x	-	9.0390	9.03884
y	-	8.6682	8.66828

A Figura 3 ilustra os resultados obtidos para 100 execuções. A média obtida foi de -18.15559, com desvio padrão de 0.80734.

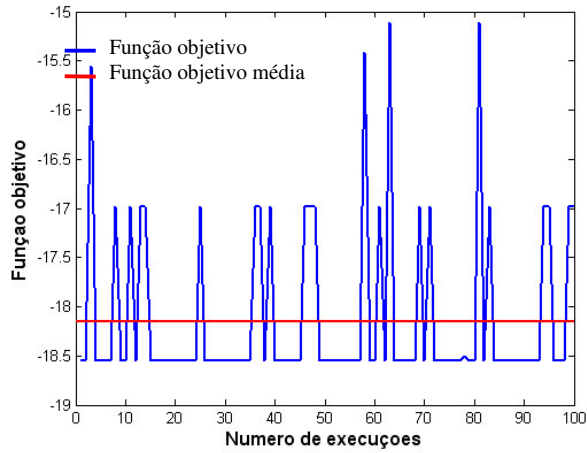


Figura 3: Função objetivo versus Número de execuções para o caso 3.

2.3 Minimização de uma função matemática – Caso 4

Para analisar a robustez ao ruído do programa computacional OTR foi minimizada a função:

$$f(x) = \cos(x) + \cos(2.5x) \quad (4)$$

no domínio:

$$0 \leq x \leq 2.6\pi$$

cujo mínimo é :

$$x_{\min} = 3.6864 \quad \text{e} \quad f_{\min} = -1.8335$$

A Tabela 5 apresenta os resultados obtidos para a função (4). Novamente, percebe-se que a eficiência do método tradicional depende da escolha do ponto inicial de busca. Sendo que das três tentativas obteve-se sucesso apenas em uma delas. O mesmo não acontece com o OTR que encontrou o ponto de mínimo em todos os casos na primeira tentativa com menos de 1s de tempo computacional e com no máximo 53 gerações.

Tabela 5: Resultados obtidos dos testes de otimização - sem ruído.

	Método Tradicional (BFGS)			OTR		
	x_0	x_f	f	pop	x_f	f
1	0.5	3.6864	-1.8335	20	3.68657	-1.83351
2	5.5	6.2832	0	40	3.68641	-1.83351
3	8.0	8.1681	-0.3089	50	3.68620	-1.83351

Denomina-se função perturbadora, a toda função harmônica de alta frequência e baixa amplitude, cujo objetivo é simular as incertezas contidas nos modelos experimentais. Seja a função perturbadora $\varepsilon(x) = 0.2 \cos(60x)$, então a função objetivo contaminada com ruído é:

$$\tilde{f} = f(x) + \varepsilon(x) = \cos(x) + \cos(2.5x) + 0.2 \cos(60x) \quad (5)$$

A Figura 4 ilustra os gráficos da função sem ruído e com ruído.

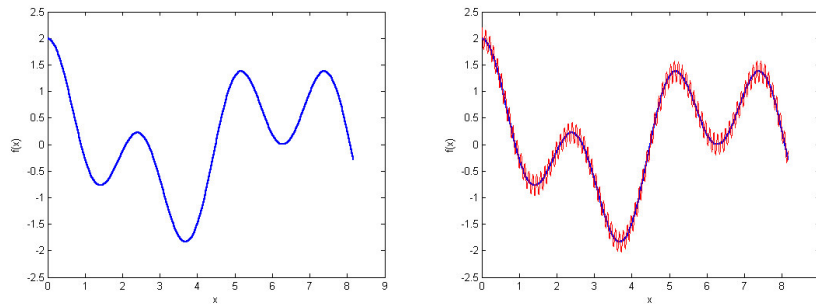


Figura 4: Função utilizada na minimização (a) sem ruído; (b) com e sem ruído.

Tabela 6: Resultados obtidos dos testes de otimização - com ruído.

	Método Tradicional			OTR		
	x_0	x_f	f	pop	x_f	f
1	0.5	0.4751	1.0683	20	3.71725	-2.03015
2	5.5	5.5	0.888	40	3.71731	-2.03015
3	8.0	8.1681	-0.1089	50	3.71725	-2.03015

Nota-se pelos resultados dispostos na Tabela 6 que o método tradicional não teve um bom desempenho com a presença de ruído. Isto acontece devido o método tradicional usar o gradiente da função objetivo, cujo efeito do ruído modifica fortemente o comportamento da função gradiente, como pode ser observado pela Figura 5.5. Por outro lado, como os AGs utilizam somente o valor da função objetivo para guiar sua pesquisa, em todos os testes obteve-se ótimos resultados.

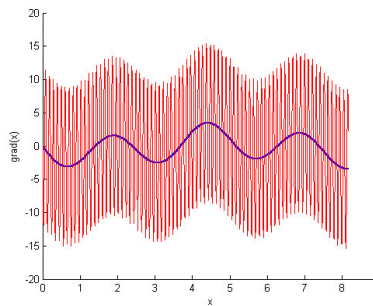


Figura 5: Gradiente da função otimizada.

2.5 Minimização da energia de um sistema de 2 molas

A Figura 6 ilustra um sistema mecânico formado por duas molas submetido a um conjunto de forças P_1 e P_2 . O objetivo neste problema é determinar o ponto de equilíbrio do sistema minimizando a energia potencial (PE). A energia potencial do sistema é dada pela Eq. (6):

$$PE = 0.5K_1 \left[\sqrt{x_1^2 + (l_1 - x_2)^2} - l_1 \right]^2 + 0.5K_2 \left[\sqrt{x_1^2 + (l_2 + x_2)^2} - l_2 \right]^2 - P_1 x_1 - P_2 x_2 \quad (6)$$

O domínio das variáveis é: $\begin{cases} 6.00 \leq x_1 \leq 10.00 \\ 2.00 \leq x_2 \leq 6.00 \end{cases}$

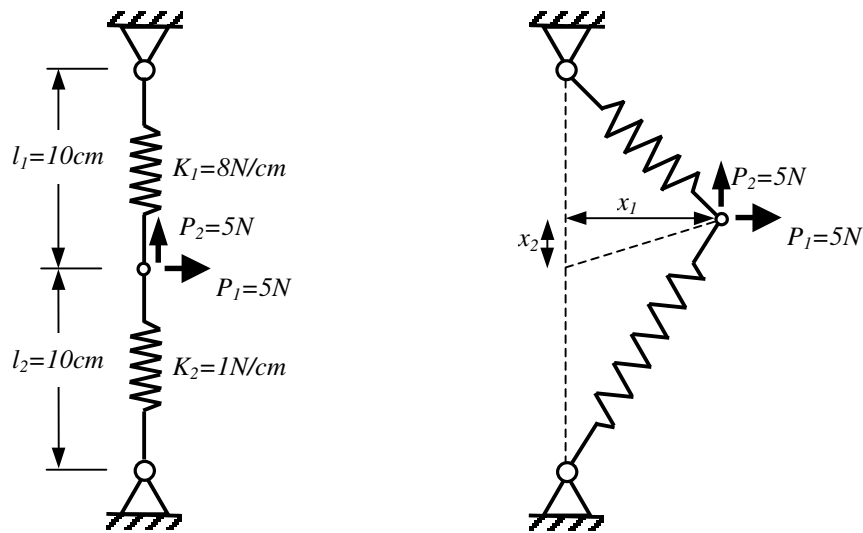


Figura 6: Determinação da posição de equilíbrio estático de um sistema de duas molas.

A Tabela 7 apresenta os resultados obtidos na primeira tentativa com um tempo computacional de 2s e 188 gerações.

Tabela 7: Resultados obtidos da minimização da função energia (PE).

	Teórico	Faria	OTR
PE	-41.81	-41.81	-41.80823
x_1	8.63	8.62	8.63071
x_2	4.53	4.52	4.52982

2.6 Minimização do volume de uma treliça

A Figura 7 ilustra uma treliça plana simétrica formada por três barras solicitada por duas forças P_1 e P_2 . O objetivo neste problema determinar as áreas A_1 , A_2 e A_3 que minimizem o volume da estrutura e satisfaça as restrições de projeto. Impõe-se que $A_1=A_3$. O volume da treliça é dado pela Eq. (7):

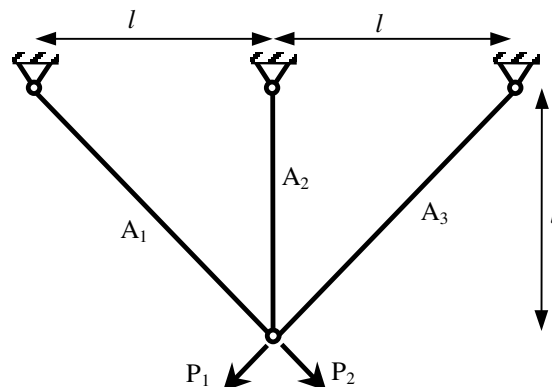


Figura 7: Treliça plana formada por 3 barras.

$$V = \rho l (2\sqrt{2}A_1 + A_2) \quad (7)$$

onde:

ρ = densidade do material da estrutura = $2.768 \times 10^{-3} \text{ kg/m}^3$;

$l = 25.4 \text{ cm}$;

$-10550 \leq \sigma_{ij} \leq 14060 \text{ N/cm}^2$;

$P_1 = P_2 = 89 \times 10^3 \text{ N}$;

Restrições:

$$\frac{\sigma_{ij}}{\sigma_a} - 1 \leq 0$$

$$g(1) = \frac{2A_1 + \sqrt{2}A_2}{2A_1(A_1 + \sqrt{2}A_2)} - 1 \leq 0$$

$$g(2) = \frac{1}{2(A_1 + \sqrt{2}A_2)} - 1 \leq 0$$

A Tabela 8 apresenta os resultados obtidos para a minimização do volume da treliça no seguinte domínio:

$$\begin{cases} 0.01 \leq A_1 \leq 1000 \\ 0.01 \leq A_2 \leq 1000 \end{cases}$$

Tabela 8: Resultados obtidos da minimização do volume da treliça.

	Braga	Faria	OTR
F(x)	1.196	1.17	1.17471
A ₁	5.162	5.09	4.95875
A ₂	2.420	2.63	2.68280

O software OTR necessitou de 1355 gerações realizadas no tempo computacional de 10s para encontrar o ponto ótimo.

3. CONCLUSÃO

Conclui-se que o programa computacional OTR teve um ótimo desempenho em todos os problemas analisados. Sendo capaz de encontrar o ponto ótimo de cada função em um tempo computacional aceitável. Além disso, apresentou robustez em problemas de multimodalidade ou na presença de ruído na função objetivo.

4. REFERÊNCIAS

- Braga, C. G., 1998, "O Uso de Algoritmos Genéticos para Aplicações em Problemas de Otimização de Sistemas Mecânicos", Dissertação de Mestrado, Universidade Federal de Uberlândia.
- Castilho, V. C., Debs, M. K. e Nicoletti, M. C., 2001, "Application of Genetic algorithm for optimizing slabs of prestressed concrete joists", 22th Iberian Ltin-American Congress on Computation Methods in Engineering, Campinas - SP, Brasil.
- Goldberg, D. E., 1989, "Genetic Algorithms in Search, Optimization and Machine Learnig", Reading, MA, Addisson Wesley Pub. Co.
- Faria M. M. L. and Steffen Jr. V., "Uma Técnica para Otimização de Projetos Mecânicos", Revista Ciência e Engenharia, UFU, Ano 1, n° 2, pp 181-197.
- Haupt, R. L. and Haupt, S. E., 1998, "Practical Genetic Algorithms", John Wiley & Sons, Inc.

- Silva, L. A., 1999, "Estudo Avaliativo de Algoritmos Genéticos Aplicados a Problemas de Identificação em Elastodinâmica", Dissertação de mestrado - Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia, Minas Gerais.
- Souza, E. N.; Mendes Neto, F., 2001, "Aplicação de algoritmo genético no dimensionamento de seções transversais retangulares de concreto armado", XXII CILAMCE – 22nd Iberian Latin-American Congress on Computational Methods in Engineering, Campinas – SP, Brasil.

ANALYSIS OF THE RELIABILITY AND EFFICIENCY FOR THE COMPUTATIONAL PROGRAM OTR

Dogmar Antonio de Souza Junior

Doutorando – Faculdade de Engenharia Mecânica - Universidade Federal de Uberlândia – Rua João Naves de Ávila n° 2121 Campus Santa Mônica – Uberlândia

Professor – Universidade de Uberaba – Rua: Coronel Severiano 251 – Uberlândia

dogmar.souza@uniube.br

Francisco Antonio Romero Gesualdo

Professor – Faculdade de Engenharia Civil – Universidade Federal de Uberlândia - Rua João Naves de Ávila n° 2121 Campus Santa Mônica – Uberlândia

gesualdo@ufu.br

Abstract: *The use of a new computational program demands several tests. Being an optimization program, it is necessary to check it to different problems, each one with its specific difficulties for reaching the optimum value of the objective function. In this work it will be present the results from the analysis of the reliability and efficiency of software OTR. The genetic algorithm was the search method implemented in DELPHI® programming language. Classic engineering problems were used to check the performance of the OTR software. They are represented by four mathematical functions without restrictions and two problems of engineering - one with restriction and another one without restriction. The results show that software OTR is reliable and robust in problems of multimodality and even having noise.*

Keywords: *optimization, genetic algorithm, reliability.*