II CONGRESSO NACIONAL DE ENGENHARIA MECÂNICA
II NATIONAL CONGRESS OF MECHANICAL ENGINEERING
12 a 16 de Agosto de 2002 - João Pessoa – PB

MÉTODOS NATURAIS DE OTIMIZAÇÃO: ALGORITMOS GENÉTICOS E APLICAÇÕES

Romes Antonio Borges rborges@mecanica.ufu.br
Cleves Mesquita Vaz mesquitav@mecanica.ufu.br
Valder Steffen Jr. vsteffen@mecanica.ufu.br
Universidade Federal de Uberlândia
Faculdade de Engenharia Mecânica
Caixa Postal 593 - CEP 38400-902 - Uberlândia - MG

Resumo. *Otimização tem a ver com a determinação da melhor configuração de projeto de um dado sistema, sem que seja necessário, entretanto, testar todas as possibilidades. Além disso, com tal ferramenta consegue-se reestruturar o desempenho global de um projeto. Otimizar significa partir de uma configuração inicial e caminhar no sentido de melhorar um dado critério de desempenho, através da modificação de parâmetros que o caracterizam. Para um problema de otimização pode-se ter várias soluções diferentes. O conceito de projeto "melhor", depende do problema, do método de solução e da tolerância adotada. Nos últimos 25 anos tem surgido excelentes algoritmos de otimização usando métodos "naturais", como os algoritmos genéticos e o simulated annealing (recozimento simulado), dentre outros. Os algoritmos genéticos modelam o processo de seleção natural relacionado ao à "luta pela vida", enquanto o simulated annealing modela o processo metalúrgico de recozimento. Ambos procuram encontrar novos pontos dentro do espaço de busca, aplicando operadores estatísticos para os pontos atuais e obtendo novos pontos que orientam a busca dos valores ótimos dentro deste espaço. Estas técnicas têm encontrado muito sucesso na solução de problemas de diversas áreas do conhecimento. Neste trabalho é feita uma abordagem introdutória restrita aos algoritmos genéticos, explorando algumas aplicações para fins de ilustração.*

Palavras-chave: *Otimização, algoritmos genéticos, simulated annealing, técnicas evolucionárias.*

1. INTRODUÇÃO

Seleção é o nome que se dá ao ajustamento (ou à adaptação) das populações aos ambientes em que se encontram.Assim, pode-se dizer que seleção natural é a modificação diferencial na freqüência relativa de genótipos devida a diferenças na capacidade que tem os fenótipos correspondentes de obter representação na geração seguinte. Esse processo opera continuamente (Back, 1995).

O conceito de *Darwin* sobre a evolução foi baseado em observações onde, na maioria das populações, poucos organismos sobrevivem e se reproduzem.

Contudo, apesar desse princípio ser verdadeiro, observa-se que continua havendo seleção mesmo numa população que se expande sem limites, pois alguns indivíduos são mais férteis que outros e deixariam maior número de representantes do seu próprio genótipo.

Existem três requisitos para o processo de seleção:

- A existência de organismos capazes de reproduzirem-se.
- A existência de características hereditárias entre tais organismos.
- E, principalmente, a relação desses organismos com o ambiente.

Os tipos de seleção natural se classificam em:

1) *Seleção estabilizadora*: Processo que opera continuamente em todas as populações, é a eliminação de indivíduos extremos.

2) *Seleção disjuntiva*: Ocorre quando dois tipos extremos de uma população aumentam à custa de formas intermediárias.

3) *Seleção direcional:* Esse tipo de seleção favorece as características fenotípicas extremas. Tende então a produzir substituição gradual de um alelo por outro no *pool* genético. Esta seleção é aquela exercida pelos criadores de animais e plantas em busca de certas características. A seleção direcional resulta em mudanças genotípicas e fenotípicas na população, modificações adaptativas na anatomia, na fisiologia e no comportamento.

Uma das mais potentes forças de seleção direcional é a fertilidade relativa. Este tipo de seleção é caracterizada os fatores que tornam um protozoário, uma planta, ou um animal aptos a produzirem maior número de descendentes vivos do que outro organismo de mesma espécie.

Outra forma de seleção natural importante é o *acasalamento não aleatório*. Estudos feitos mostram que os alelos tendem a permanecer em equilíbrio na população se o acasalamento ocorrer ao acaso. Entretanto, o acasalamento não aleatório é a regra na maioria das populações naturais. As cores brilhantes das flores, os evidentes sinais em muitos peixes e aves e os complexos padrões de comportamento sexual além da territorialidade e hierarquias de dominância que asseguram a exclusão de alguns indivíduos da classe procriadora são fatores de indução a este tipo de acasalamento.

O algoritmo genético está arraigado tanto na genética natural quanto na computação científica, isto é, traz os conceitos e fundamentos da natureza para serem processados pela moderna computação digital.

A grosso modo, a variável de projeto (ou de decisão) de um sistema genético artificial é análoga a um cromossomo de um sistema biológico. Em um sistema natural, um ou mais cromossomos se combinam para dar a prescrição genética necessária para a formação de um ou mais organismos (Davis, 1987).

Em um sistema natural o pacote genético total é chamado *genótipo* enquanto que em um sistema genético artificial este conjunto de variáveis é chamado de *estrutura.* No sistema natural o conjunto formado pela interação de um pacote genético total com seu ambiente é chamado *fenótipo,* enquanto que no sistema genético artificial, a estrutura é decodificada para a forma particular de um *conjunto de parâmetros, solução alternativa,* ou simplesmente *pontos* no espaço de solução.

O projeto de um sistema genético artificial possui uma variedade de alternativas para codificar parâmetros numéricos e não numéricos. No meio natural dizemos que os cromossomos são compostos por genes os quais podem ser classificados em alguns valores numéricos chamados alelos. Na genética, a posição de um gene é identificada separadamente para cada função. Podemos, portanto, falar de um gene particular, por exemplo, num dado animal, o gene que determina a cor de seus olhos está na posição 10, e seu valor de alelo vale por exemplo, “olhos castanhos”.

Na genética de procura artificial diz-se que as variáveis são compostas de *características* as quais são tomadas em diferentes valores. Características podem ser localizadas em diferentes posições das variáveis. Assim, num primeiro instante, não se tem a distinção das características particulares de um gene e de sua posição; a posição de um bit em uma variável tem significado determinado uniformemente ao longo de uma população e do tempo.

A correspondência entre a linguagem natural e artificial é mostrada na Tab. (1)

Tabela 1. Relação entre a genética natural e a artificial.

Natural	Algoritmo Genético
Cromossomo	Variável
Gene	Característica
Alelo	Valores Característicos
Posição (lócus)	Posição Variável
Genótipo	Estrutura
Fenótipo	Conjunto de Parâmetros
Epistasis	Não-linearização

2. ALGORITMOS GENÉTICOS BINÁRIOS

Os algoritmos genéticos podem ser escritos usando parâmetros contínuos ou parâmetros codificados em binário. Ambas as abordagens seguem a mesma regra para modelar a recombinação genética e a seleção natural (Bean *et al*, 1992).

Tanto a evolução biológica quanto os algoritmos genéticos binários começam com uma população inicial constituída por membros gerados aleatoriamente.

Por exemplo, suponha que queremos criar cachorros que sejam "bons corredores". Então são selecionados alguns dos melhores cães (suponhamos os quatro melhores) e esses quatro cachorros são levados a procriar. As características desses melhores *"corredores"* são codificadas em uma seqüência de números binários a eles associados. Desta população, dois são selecionados de forma aleatória para criar dois novos filhotes. Estes filhotes terão grande probabilidade de serem *"bons corredores"*, pois ambos os pais possuem genes que originam esta característica. A nova seqüência binária dos filhotes contém porções das seqüências binárias dos pais. Estes novos filhotes substituem os dois cães que não são *"bons corredores"* e por isso foram descartados.

Filhotes são reproduzidos até que se tenha uma população com tamanho da original. O processo iterativo leva à condição de se ter cachorros cada vez melhores corredores.

2.1. Componentes de um Algoritmo Genético Binário

Os Algoritmos Genéticos começam como qualquer outro algoritmo de otimização, definindo os parâmetros de otimização, a função custo e o próprio custo. A finalização também é realizada como em outros métodos de otimização, ou seja, através de testes de convergência. Todavia os Algoritmos Genéticos diferem bastante de outras técnicas de minimização, especialmente por dispensarem o uso de derivadas da função custo para determinar a direção de busca. Outro aspecto importante é que não investem todo o esforço computacional num único ponto, mas sim operam sobre uma população de pontos. Um esquema básico para os Algoritmos Genéticos é mostrado na Fig. (1):

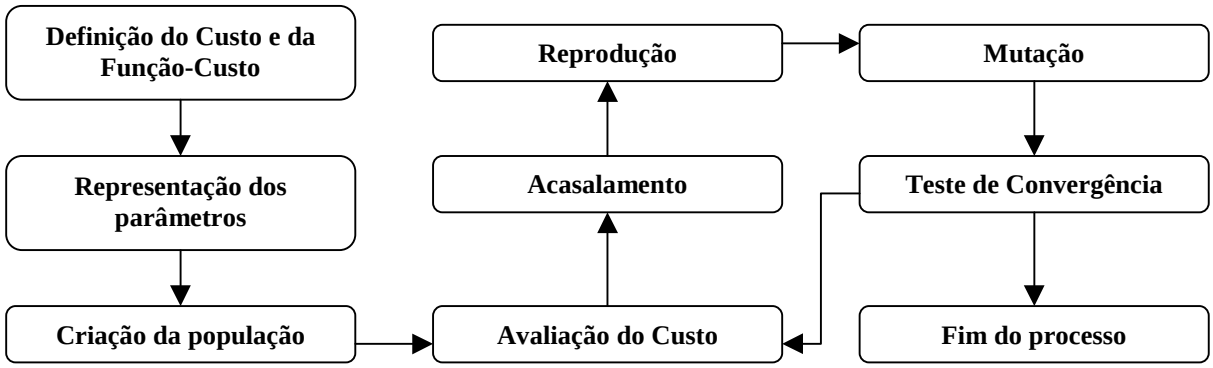

Figura 1. Fluxograma para os algoritmos genéticos binários

2.2.Seleção de Parâmetros e da Função Custo

A função custo gera uma saída para um conjunto de parâmetros de entradas. Esta função pode ser caracterizada por uma função matemática, por um experimento, por um jogo, ou por um código computacional. O objetivo é modificar a saída de alguma maneira, pela atualização dos parâmetros do problema. Isto é feito encontrando valores apropriados para os parâmetros de entrada. Por exemplo, ao encher uma banheira com água, onde o custo é a diferença entre a temperatura atual e a temperatura desejada, observa-se que a função custo é relacionada a resultados experimentais ao sentir a temperatura da água, combinando a abertura das torneiras quente e fria (parâmetros de entrada).

Observa-se que determinar a função custo apropriada e decidir quais os parâmetros escolher são dois aspectos intimamente relacionados.

Os Algoritmos Genéticos começam pela definição de uma cadeia de valores de parâmetros a serem otimizados(cromossomos). Se o cromossomo tem N_{par} parâmetros (problema de otimização com dimensão N^{par}) dados por $p_1, p_2, ..., p_{N\,par}$, então o cromossomo é escrito como um vetor com N_{par} elementos. Assim:

$$cromossomo = [p_1, p_2, ..., p_{N_{par}}] \tag{1}$$

.

Como exemplo, seja o caso em que o ponto de máxima elevação num mapa topográfico deve ser encontrado. Isto requer uma função custo com parâmetros de entrada de longitude (*x*) e de latitude (*y*). Dessa forma:

$$cromossomo = [x, y], \text{ onde } \quad N_{par} = 2 \tag{2}$$

Cada cromossomo tem um custo encontrado pelo cálculo de uma certa função custo *f*. Ou seja, a formulação ficará:

$$p_1, p_2, ..., p_{N_{par}}: \quad custo = f(cromossomo) = f(p_1, p_2, ..., p_{N_{par}}) \tag{3}$$

Então, para encontrar o pico no mapa descrito anteriormente, a função custo pode ser escrita como o valor negativo da elevação. Na forma de um problema de minimização tem-se:

$$f(x, y) = -elevação \tag{4}$$

Freqüentemente a função custo é muito complicada e tem-se que decidir quais parâmetros do problema são importantes. Um número excessivo de parâmetros não facilita o processo de busca.

Na maioria das vezes, a escolha do número correto e de quais são os parâmetros, é feita pela experiência ou por tentativa. Em outras situações, pode-se ter uma função analítica, com os parâmetros sendo as próprias variáveis da função. Assim, na função custo definida por:

$$f(w, x, y, z) = 2x + 3y + \frac{z}{100000} + \frac{\sqrt{w}}{9876}, \tag{5}$$

Com os parâmetros variando entre 1 e 10, pode-se fazer uma simplificação com a ajuda de um algoritmo de otimização. Como w e z são termos extremamente pequenos na região de interesse, poderão ser descartados. Esta função custo quadri-dimensional é adequadamente modelada com apenas dois parâmetros na região de interesse.

A maioria dos problemas de otimização requer restrições ou valores limites para os parâmetros envolvidos. Os parâmetros de restrição podem ser especificados:

a) Limites impostos aos parâmetros;
b)Variáveis com restrição, transformadas em novas variáveis, porém sem restrição;
c)Conjunto finito de valores dos parâmetros na região de interesse.

Na literatura sobre Algoritmos Genéticos a interação entre parâmetros é chamada *epistasis* (termo biológico para a interação entre genes).

2.3. Representação dos Parâmetros

Os Algoritmos Genéticos binários trabalham com um número finito de parâmetros (porém extremamente grande). Esta característica faz o Algoritmo Genético ser ideal para otimizar um custo no qual os parâmetros assumem um número finito de valores.

Se um parâmetro é continuo, então tem que ser *quantificad*o. Para isto, a faixa de valores possíveis é dividida em níveis de *quantificação* iguais. Qualquer valor caindo dentro de um dos níveis é ajustado igual ao valor médio, alto, ou baixo daquele nível. Em geral, fixar o valor do parâmetro ao valor médio do nível de quantificação é a melhor estratégia, pois o maior erro possível é a metade de um nível. Arredondando para o valor baixo ou para o valor alto nos conduz a um *Erro Máximo* igual ao nível de quantificação. As fórmulas matemáticas para codificação e decodificação binária do parâmetro p_n são dadas a seguir.

$$\text{Codificando: } \begin{cases} p_{norm} = \dfrac{p_n - p_{lo}}{p_{hi} - p_{lo}} \\ gene[m] = \text{round}\{p_{norm} - 2^{-m} - \sum_{p=1}^{m-1} gene[m]2^{-p}\} \end{cases} \tag{6}$$

$$\text{Decodificando: } \begin{cases} p_{quant} = \sum_{m=1}^{N_{gene}} gene[m]2^{-m} + 2^{-(m+1)} \\ q_n = p_{quant}(p_{hi} - p_{lo}) + p_{lo} \end{cases} \tag{7}$$

Onde:

- p_{norm} = Parâmetro normalizado, $(0 \le p_{norm} \le 1)$;
- p_{lo} = Menor valor do parâmetro;
- p_{hi} = Maior valor do parâmetro;
- $gene[m]$ = Versão binária de p_n;
- $round\{\cdot\}$ = Arredondamento para o inteiro mais próximo;
- N_{gene} = Número de *bits* no gene;
- p_{quant} = Versão quantificada de p_{norm};
- q_n = Versão quantificada de p_n.

Os Algoritmos Genéticos trabalham com codificação binária, mas a função custo requer freqüentemente parâmetros contínuos. Neste caso, cada um dos parâmetros é representado como numero de ponto flutuante. Mais detalhes podem ser encontrados em . *Haupt and* . *Haupt* (1998).

2.4. População Inicial

O Algoritmo Genético começa com uma grande comunidade de cromossomos, conhecida como *população inicial*. Esta população inicial tem N_{ipop} cromossomos e é uma matriz $N_{ipop} \times N_{bits}$ pre-

enchida com 1 e 0 gerados a partir de:

$$IPOP = round\{random(N_{ipop}, N_{bits})\} \quad (8)$$

onde a função $random(N_{ipop}, N_{bits})$ gera uma matriz $N_{ipop} \times N_{bits}$ de números aleatórios uniformes entre 0 e 1. Este tipo de função pode ser calculada a partir de procedimentos matemático-computacionais. A função *round*{} arredonda o número para um inteiro mais próximo.

Cada linha da matriz é um cromossomo. Com isso os parâmetros são passados para a função custo, que pode agora ser calculada.

2.5. Seleção natural

A população inicial é geralmente grande demais para ser tratada por todos os passos iterativos do Algoritmo Genético. Assim, boa parte da população é descartada através da seleção natural, dentro do esquema de sobrevivência dos mais aptos.

Primeiro a população inicial (N_{ipop}) e os cromossomos associados a ela são ordenados em ordem crescente da função custo. Então, somente os melhores membros $N_{pop} \leq N_{ipop}$ desta população são mantidos para cada iteração e os outros são descartados. A seleção natural ocorre em cada geração ou iteração do algoritmo. Dentre os cromossomos N_{pop} de uma geração, só os melhores (N_{good}) sobrevivem para o cruzamento; os piores (N_{bad}) são descartados para abrir espaço a uma nova descendência gerada no cruzamento.

Decidir quantos cromossomos selecionar é uma tarefa quase sempre arbitrária, deixando sobreviver para a próxima geração os genes disponíveis para a descendência. Se muitos cromossomos forem mantidos, tem-se uma performance ruim para esta geração. No processo de Seleção Natural são mantidos mais ou menos 50% desses cromossomos.

2.6. Emparelhamento

São selecionados dois dos N_{good} cromossomos para acasalar e produzir dois descendentes. Estes irão substituir os N_{bad} cromossomos anteriormente mencionados. O acasalamento dos cromossomos nos Algoritmos Genéticos é um pouco diferente do acasalamento em uma espécie animal. A bibliografia recomenda diferentes tipos de emparelhamento (*Haupt and Haupt, 1998*).

2.7. Cruzamento

Os pais selecionados no processo de emparelhamento irão agora se cruzar e, com isso, ter um ou mais descendentes. O cruzamento é a primeira forma pela qual o Algoritmo Genético explora uma superfície de custo.

Chama se a isto de *"exploração"* porque o algoritmo genético usa combinações dos bits já presentes nos cromossomos. As formas mais comuns de cruzamento envolvem dois pais que irão produzir dois descendentes. O ponto de cruzamento é selecionado entre o primeiro e o último bit dos cromossomos dos pais. Isto ocorre de forma que o Pai_1 passa seu código binário da esquerda para o $Filho_1$, enquanto que o Pai_2 passa seu código binário da esquerda para o $Filho_2$. Com isso o código genético da direita do Pai_1 irá para o $Filho_2$ e o da direita do Pai_2 irá para o $Filho_1$. Consequentemente, os filhos vão ter porções dos códigos binários de ambos os pais. Os pais vão produzir uma quantidade N_{bad} de filhos, isto é, a população retorna novamente a seu número de elementos inicial, N_{pop}.

2.8. Mutação

Mutações randômicas alteram uma pequena porcentagem dos bits da lista de cromossomos. Mutações são a segunda forma do algoritmo genético explorar uma superfície de custo. Este operador genético é capaz de introduzir características não encontradas na população original e impede o algoritmo de convergir prematuramente. A mutação de ponto único modifica "1" para "0" e vice versa, no caso da codificação binária.

Os pontos de mutação são selecionados aleatoriamente de um total de $N_{pop} \times N_{bits}$ na matriz da população. Ao aumentar o número de mutações, aumenta-se a liberdade do algoritmo em buscar alternativas fora da região em que se encontra no espaço de parâmetros. Tipicamente faz-se a mutação numa taxa de 1 a 5% do total de bits por iteração (geração).

Cabe salientar que não se faz mutações ao final de uma iteração completa. Evidentemente não se aplica a mutação às melhores soluções, consideradas *"de elite"*. Assim 5% da população sofre mutação, exceto os melhores cromossomos.

2.9. Geração Futura

Depois de feitas as mutações, calcula-se os custos associados aos descendentes e aos cromossomos que sofreram mutação e dá-se seqüência ao processo iterativo.

2.10. Convergência

O número de gerações do processo evolutivo em curso depende do momento em que a solução desejável é alcançada ou se chega a um número máximo de iterações. Depois de um certo tempo todos os cromossomos e seus respectivos custos se tornam os mesmos (a menos de mutações); neste ponto o algoritmo deve ser interrompido.

A maioria dos algoritmos genéticos mantém informações sobre os aspectos estatísticos da população na forma de média, desvio padrão, custo mínimo. Estas informações podem servir como teste de convergência (Holland, 1975).

3. O TRATAMENTO DE RESTRIÇÕES NOS ALGORITMOS GENÉTICOS

Uma dificuldade que ocorre na utilização dos algoritmos genéticos tem a ver com as funções de restrição que são absolutamente necessárias em muitas aplicações. Na sua formulação original, os algoritmos genéticos não consideram restrições. Uma alternativa bastante viável para se incluir restrições é o uso de funções de penalidade. Dessa forma, a função objetivo restrita é transformada numa função pseudo-objetivo, esta última, sem restrição por exemplo.

Minimizar: $F(x)$

Sujeito a: $g_i(x) \geq 0,\ i = 1,2,...,n$. Onde *x* é um vetor de *m* coordenadas.

Para transformar este problema em outro, sem restrição, cria-se uma função pseudo-objetivo

$$\Phi(x, r_p) = F(x) + r_p\, p(x)$$

Onde *P(x)* é a função de penalidade e r_p é o coeficiente de penalização.

Aplicando este principio ao caso em tela tem-se o novo problema de otimizaçao irrestrito.

Minimizar: $\Phi(x, r_p) = F(x) + r_p \sum_{n=1}^{n} [0, g_i(x)]^2$

A solução do problema irrestrito converge para aquela do caso com restrição para valores elevados de r_p.

4. APLICAÇÕES NUMÉRICAS

4.1. Combinação de Notas Musicais

Foi implementada uma sub-rotina para encontrar as notas musicais da primeira estrofe de *"Luar do Sertão"*, canção popular bastante conhecida.

A estrofe é a seguinte: *"Não há, ó gente, ó não, luar como este do sertão"* cuja partitura é mostrada na fig. (2) abaixo. As notas musicais são dadas na Tab. (2), juntamente com sua representação numérica. Para este caso usamos algoritmos genéticos com parâmetros contínuos.

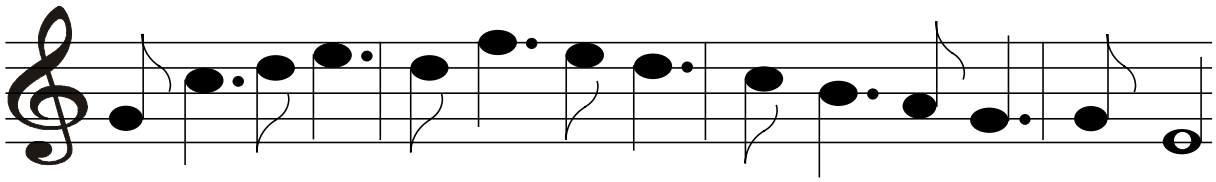

Figura 2. Partitura de uma estrofe da musica Luar do Sertão.

Tabela 2. Notas Musicais e sua correspondência numérica.

Notas	**Dó**	**Re**	**Mi**	**Fá**	**Sol**	**La**	**Si**	***Hold***
Correspondência Numérica	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**

Tabela 3. Valores obtidos na Trigésima Oitava Geração

Geração 38	***5.0(5.0)***	***1.0(1.0)***	***7.9(8.0)***	***1.9(2.0)***	***3.3(3.0)***	***7.9(8.0)***	***2.1(2.0)***
	4.0(4.0)	***7.9(8.0)***	***3.0(3.0)***	***2.3(2.0)***	***7.9(8.0)***	***1.0(1.0)***	***6.5(7.0)***
	7.9(8.0)	***6.1(6.0)***	***5.1(5.0)***	***7.9(8.0)***	***3.9(4.0)***	***3.0(3.0)***	***8.0(8.0)***

4.2.Problema de otimização com restrição

Seja o problema:

Minimizar $F(x) = (x_1 - 1)^2 + (x_2 - 1)^2$
Sujeito a $x_1 + x_2 - 1 \leq 0$ e $x_1 \geq 0$

Foi criada uma função pseudo-objetivo, conforme explicado na seção 3. O espaço de projeto foi assim definido: $-5 \leq x_1 \leq +5$ e $-5 \leq x_2 \leq +5$. O processo evolutivo considerou 500 indivíduos ao longo de 300 gerações. A tab. (4) mostra os resultados obtidos.

Tabela 4. valores ótimos das variáveis de projeto

x_1	**0.7777**
x_2	**0.7777**
Função Pseudo-Objetivo	**-0.2222**

5. CONCLUSÕES

Este trabalho apresentou os fundamentos dos métodos heurísticos de otimização, com enfoque nos algoritmos genéticos. Os aspectos mais importantes foram destacados e procurou-se ressaltar o caso em que restrições são incorporadas ao problema de otimização. Neste caso, são utilizadas funções de penalidade para criar uma função pseudo-objetivo que é então otimizada como um problema irrestrito. Dois exemplos foram apresentados como ilustração. No primeiro, o objetivo foi o de mostrar a versatilidade das técnicas aplicadas, ao se procurar fazer convergir às notas musicais, organizando-as de forma a se chegar a uma canção popular conhecida, o que ocorreu após 38 gerações. O segundo exemplo trata de um problema minimização de uma função analítica, onde uma restrição tem que ser respeitada. Mostrou-se que através do uso de uma função de penalidade pode-se chegar à solução do problema. Neste exemplo, os números de indivíduos e de gerações utilizados foram maiores do que o necessário, uma vez que resultados similares são obtidos para configurações bem mais econômicas do ponto de vista computacional. De uma maneira geral, pode-se considerar que as técnicas apresentadas são bastante eficientes e encontram potencialmente largas aplicações nos diferentes ramos das ciências e da engenharia. Como perspectiva para trabalhos futuros, outras técnicas estão sendo testadas com vistas a aplicações em engenharia, como por exemplo, o recozimento simulado e a chamada *Tabu Search*. Problemas complexos incluindo várias restrições e situações onde a função objetivo é conhecida apenas de maneira implícita em relação às variáveis de projeto, além de ser gerada numericamente, estão sendo investigados.

6. REFERÊNCIAS

Haupt,R.L. and Haupt, S.E., **"**Practical Genetic Algorithms", John Wiley,1998.

Vanderplaats, G.N., "Numerical Optimization techniques for engineering design", McGraw-Hill Book Company,1984.

Back, T., 1995, "Evolutionary Algorithms in Theory and Practice", Oxford University, New York.

Bean, J.F. and Hadj-Alouane, A. B., 1992, "A Dual Genetic Algorithm for Bounded Integer Programs", Department of Industrial and Operations Engineering, The University of Michigan, TR 92-53.

Borges, R.A., 2001, "Introdução aos Algoritmos Genéticos", Relatório Final de Estudo Dirigido, FEMEC, Universidade Federal de Uberlândia, Uberlândia, Brasil.

Davis, L., 1987, (Editor), "Genetic Algorithms and Simulated Annealing", Pitman, London.

Holland, J., 1975, "Adaptation in Natural and Artificial Systems", Ann Arbor: University of Michigan Press.

NATURAL METHODS OF OPTIMIZATION: GENETIC ALGORITHMS AND APPLICATIONS

Romes Antonio Borges rborges@mecanica.ufu.br
Cleves Mesquita Vaz mesquitav@mecanica.ufu.br
Valder Steffen Jr. vsteffen@mecanica.ufu.br
Federal University of Uberlândia
School of Mechanical Engineering
P.O. Box 593 – ZIP 38400-902 - Uberlândia – MG – Brazil

Abstract. *Optimization is related with the determination of the best design configuration of a given system, without testing all possible alternatives. Optimization means to start from on initial design configuration and evolve in the sense of improving a defined performance criterion, through the modification of the system characterizing parameters. An optimization problem may have several different solutions. To choose the "best" design configuration depends on the solution method and the tolerance adopted. In the last 25 years excellent optimization algorithms based on natural phenomena have appeared, such as the genetic algorithms and simulated annealing. The genetic algorithms model the process of natural selection related to evolution (Survival of the fittest), while simulated annealing mimics the annealing process found is metallurgy. Both the processes seek to find new points in the search space, through the application of natural operators at the actual points. This way, new regions can be explored in the search space and the optimum can be found iteratively. These techniques have shown to be very effective in a wide range of problems in different areas. This paper presents an introduction to genetic algorithms and some applications are presented for illustration.*

Keywords**:** *Optimization, genetic algorithms, heuristic techniques.*