PEF 5710 - Otimização Estrutural

1º Quadrimestre 2022

Semana 10

Algoritmo Genético

Prof. Marcelo Araujo da Silva
marcelo.araujo@ufabc.edu.br

Professor Titular da Disciplina:
Dr. Reyolando M. L. R. F. Brasil

Algoritmo Genético

Configuração do Problema Para Variáveis Discretas

Os problemas de otimização estrutural para variáveis discretas podem ser representados na seguinte forma:

Determine **b** ∈ $\Re^n$ que minimize a função objetivo

$$f(\boldsymbol{b},T) = \bar{f}\,(\boldsymbol{b},T) + \int_0^T \tilde{f}\,(\boldsymbol{b},\boldsymbol{z},\dot{\boldsymbol{z}},\ddot{\boldsymbol{z}},t)dt.$$

Sujeito às restrições estáticas:

$$g_i = \bar{g}_i(\boldsymbol{b},T) + \int_0^T \tilde{g}_i(\boldsymbol{b},\boldsymbol{z},\dot{\boldsymbol{z}},\ddot{\boldsymbol{z}},t)dt \begin{cases} = 0 \; para \; i = 1,\ldots,l \\ \leq 0 \; para \; i = l+1,\ldots,m \end{cases}.$$

Configuração do Problema Para Variáveis Discretas

Sujeito às restrições dinâmicas:

$$g_i = \tilde{g}_i(\boldsymbol{b}, \boldsymbol{z}, \dot{\boldsymbol{z}}, \ddot{\boldsymbol{z}}, t) \begin{cases} = 0 \; para \; i = m+1, \ldots, l' \\ \leq 0 \; para \; i = l'+1, \ldots, m' \end{cases} \; para \; t \in [0, T] \, .$$

Com

$$b_i \in \boldsymbol{b} \equiv \{ b_{i1} \; b_{i2} \; \ldots \; b_{iN_{Ei}} \} \, ,$$

onde b_{i1}, b_{i2}, …, $b_{i\,NEi}$ são os N_{Ei} possíveis valores discretos para a variável b_i.

Configuração do Problema Para Variáveis Discretas

Nesses problemas, a variável de estado ***z***(*t*), o vetor deslocamento, deve satisfazer as equações do movimento:

$$\boldsymbol{M}\ddot{\boldsymbol{z}} + \boldsymbol{C}\dot{\boldsymbol{z}} + \boldsymbol{K}\boldsymbol{z} = \boldsymbol{p}(t), \ \forall t \in [0, T]\,,$$

com as condições iniciais:

$$\boldsymbol{z}(0) = \boldsymbol{z}_0 \qquad \text{e} \qquad \dot{\boldsymbol{z}}(0) = \dot{\boldsymbol{z}}_0\,.$$

É necessário resolver a equação do movimento, que precisa ser integrada numericamente. Para tal, o intervalo [*0*, *T*] precisa ser discretizado.

Algoritmos Genéticos

Baseados na *seleção natural* e no *processo de Monte Carlo*, nos algoritmos genéticos, AG, os projetos gerados aleatoriamente vão de “geração” em “geração”, com a introdução de pequenas mudanças aleatórias em suas características.

Algumas definições precisam ser feitas:

- **População** - É o conjunto de pontos, cada um deles um projeto diferente gerado aleatoriamente. N_p é o número desses projetos ou o tamanho da população;
- **Geração** - É uma iteração do algoritmo;
- **Cromossomo** - É o vetor de valores das variáveis de projeto de um determinado ponto. Também chamado de projeto ou cadeia genética;
- **Gene** - É um valor particular (um escalar) de uma das variáveis de projeto, um componente do cromossomo;
- **Função de fitness** - Define a importância relativa de um projeto. Um valor maior implica em um projeto melhor.

Algoritmos Genéticos

Outras definições:

- **Reprodução** - É um operador em que um projeto antigo é repassado para uma nova geração de acordo com seu nível de aptidão. É o processo de seleção;
- **Mating pool** - Parte da população que vai participar do processo de *reprodução*, escolhidos entre os membros mais aptos do mesmo, avaliados pela *função de aptidão*;
- **Mating** - É o processo pelo qual membros selecionados de uma nova população trocam características entre si;
- **Mutação** - É uma mudança aleatória em qualquer característica (gene)
- **Critério de parada** - Se a melhoria na melhor função objetivo for menor do que um dado pequeno valor para as últimas iterações consecutivas (em um número definido), ou o número de as iterações excedem um valor especificado, o algoritmo é encerrado;
- **Imigração** - Introdução de projetos totalmente novos na população, em busca de diversidade, em algumas iterações quando a convergência é lenta.

Exemplo 1 - Criação de uma Cadeia Binária *B*

Considere um problema de otimização discreta com duas variáveis de projeto: b_1 e b_2. A variável b_1 pode assumir qualquer valor do grupo representado pelo vetor

$$\boldsymbol{b}_1^t = [b_{11} \ b_{12} \ b_{13} \ b_{14} \ b_{15} \ b_{16}] \, ,$$

já a variável b_2 pode assumir qualquer valor do grupo representado pelo vetor

$$\boldsymbol{b}_2^t = [b_{21} \ b_{22} \ b_{23} \ b_{24} \ b_{25}] \, .$$

Para definir um projeto nesse problema de otimização, é necessário escolher um valor para b_1 e outro para b_2, dentre os valores definidos por $\boldsymbol{b_1}$ e $\boldsymbol{b_2}$.

Exemplo 1 - Criação de uma Cadeia Binária *B*

A escolha é baseada na numeração da posição de cada valor que pode ser assumido por cada variável de projeto. A *Tabela 1* exemplifica o procedimento.

Tabela 1 - Criação de cadeias binárias para representar uma variável de projeto

Posição	Variável			
	b_1		b_2	
	Cadeia	Valor	Cadeia	Valor
1	000	b_{11}	000	b_{21}
2	001	b_{12}	001	b_{22}
3	010	b_{13}	010	b_{23}
4	011	b_{14}	011	b_{24}
5	100	b_{15}	100 à 111	b_{25}
6	101 à 111	b_{16}		

Então, por exemplo, o projeto $\boldsymbol{b}^t = [b_{13}\ b_{21}]$ é representado pela cadeia **B** = {010000}.

Exemplo 1 - Criação de uma Cadeia Binária *B*

Como um procedimento para a determinação de **B** (cadeia binária) é definido, uma primeira população com membros N_p é criada usando N_p cadeias **B**.

O passo seguinte é definir um critério de reprodução, baseado em uma *função de aptidão*. Os membros mais apropriados da população são selecionados para reprodução e agrupados num *mating pool* de onde os membros são selecionados para *reprodução* e *mutação*.

O próximo operador, *cruzamento*, é executado entre dois projetos *parentes*. Para tal, dois projetos são selecionados aleatoriamente. Então, os projetos relacionados são divididos em segmentos e alguns destes são trocados entre seus correspondentes de outro parente. A mutação muda arbitrariamente os valores dos genes (de 0 para 1 e vice-versa).

Exemplo 1 - Criação de uma Cadeia Binária *B*

No AG originalmente desenvolvido por Arora et al (1994), projetos inviáveis não são rejeitados e violações das restrições são usadas para definir a *função de penalidade*:

$$p_i = f_i + RK_{bi},$$

onde f_i é a função objetivo para o i-ésimo projeto, $R > 0$ é o parâmetro de penalidade e K_{bi} é a máxima violação da restrição do i-ésimo projeto. R é dado por

$$R = \frac{\sum_{i=1}^{N_p} f_i}{N_p \varepsilon},$$

onde ε é o valor aceitável para a máxima violação da restrição.

Exemplo 1 - Criação de uma Cadeia Binária *B*

Baseado na definição da *função de penalidade*, a função de aptidão para o *i*-ésimo projeto é definida como:

$$F_i = (1+\vartheta)p_{max} - p_i \, ,$$

onde $\vartheta > 0$ é um número pequeno usado para forçar a convergência e p_{max} é o maior valor da penalidade para a primeira geração.

Na implementação do *mating*, dois projetos são selecionados aleatoriamente e um número aleatório entre 0 e 1 é gerado. Se esse número é menor que P_c, então o cruzamento acontece. Caso contrário, outro par é selecionado. Aqui, $P_c = 0{,}5$.

Exemplo 1 - Criação de uma Cadeia Binária *B*

Para cada *bit* da população toda um número aleatório é selecionado, e, se esse número for maior que a probabilidade de mutação P_m, a mutação ocorre na cadeia **B**.

Ao invés de escolher um número aleatório para cada *bit*, o número de mutações é calculado. O número de mutações é calculado pela equação:

$$N_m = P_m N_p N_b,$$

onde N_b é o número de *bits* que representa um projeto. No nosso exemplo, $N_b = 6$ e $P_m = 0{,}3$.

Exemplo 1 - Criação de uma Cadeia Binária *B*

É necessário determinar o tamanho da população N_p, um *critério de parada* e um limite para o número de iterações. O possível número de projetos para um problema é:

$$N_E = \prod_{i=1}^{n} N_{E_i} \, ,$$

onde N_{Ei} é o número de valores discretos para a *i*-ésima variável e *n* é o total de variáveis de projeto. Neste exemplo $N_{E1} = 6$ e $N_{E2} = 5$ então $N_E = 30$. Define-se, então

$$\chi = \frac{N_E}{n_1} .$$

Exemplo 1 - Criação de uma Cadeia Binária *B*

O seguinte procedimento determina o valor de N_p:

- se $N_E \leq n_2 n$ então
  - $N_p = N_E$
- senão
- se $\chi < n_3 n$ então
  - $N_p = n_3 n$
- senão
- se $\chi > n_4 n$ então
  - $N_p = n_4 n$
- senão
  - $N_p = \chi$

Nesta implementação $n_1 = 1000$, $n_2 = 6$, $n_3 = 2$ e $n_4 = 8$.

Exemplo 1 - Criação de uma Cadeia Binária *B*

Um dos *critérios de parada* para o presente algoritmo é baseado na mudanças de valores da *função de aptidão* e pode ser representado pela inequação:

$$\frac{\left[\max\limits_{1\leq i\leq N_p} p_i\right]_{k+1} - \left[\max\limits_{1\leq i\leq N_p} p_i\right]_{k}}{\left[\max\limits_{1\leq i\leq N_p} p_i\right]_{k=1}} \leq \varepsilon',$$

Onde *k* é a *k*-ésima iteração e *ε'* é um número pequeno adotado como 10^{-3}. Um segundo critério de parada neste exemplo é

$$k < 3n.$$

Outros critérios de parada podem ser obtidos de Arora et al (1994).

O Algoritmo Genético

Em resumo, o algoritmo genético é como se segue:

Passo 1 - Defina uma cadeia binária **B** para representar o dado projeto.

Passo 2 - Gere N_p cadeias aleatórias (membros da população). Faça $k = 0$.

Passo 3 - Defina as *funções de penalidade* e *de aptidão*.

Passo 4 - Calcule valores de aptidão para todos os projetos. Faça $k = k + 1$.

Passo 5 - *Reprodução*:

- **5.1** - Selecione um líder (projeto) da geração anterior. Salve esse projeto duas vezes. Copie um para a próxima geração e mande o outro para o *mating pool*.

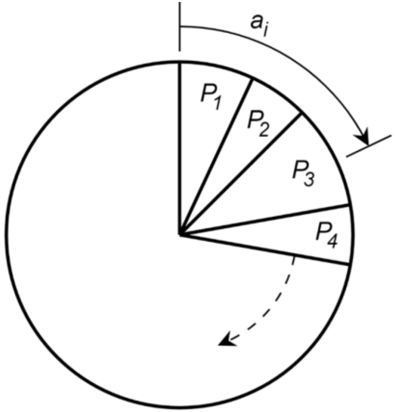
O Algoritmo Genético

Passo 5 - *Reprodução*:

- **5.2** - Calcule a probabilidade de selecionar cada projeto usando

$$P_i = \frac{F_i}{\sum_{j=1}^{Np} F_j}$$

e monte o esquema da roleta de área unitária abaixo:



O Algoritmo Genético

Passo 5 - *Reprodução*:

- **5.3** - Selecione aleatoriamente N_P*-1* projetos gerando N_P*-1* números aleatórios a_i entre [*0, 1*].

Passo 6 - *Mating*:

- **6.1** - Selecione dois projetos (um par para reprodução) do *mating pool*.
- **6.2** - Gere um número aleatório. Se for menor que a probabilidade de *mating*, P_c, faça o *mating*: selecione dois *bits* consecutivos em uma cadeia que representa um dos projetos e mude pelo *bit* correspondente do outro projeto. Volte ao **Passo 6.1** e repita o processo até que o todos os projetos da população tenham sido selecionados ao menos uma vez.

O Algoritmo Genético

Passo 7 - *Mutação*:

- **7.1** - Calcule o possível número de mutações com a equação:

$$N_m = P_m N_p N_b.$$

- **7.2** - Escolha N_m projetos do *mating pool*. Para cada projeto, selecione a posição na cadeia e troque 0 para 1, ou vice versa.

Passo 8 - Se os critérios de parada forem satisfeitos, ou seja, se as inequações

$$\frac{\left[\max\limits_{1\leq i\leq N_p} p_i\right]_{k+1} - \left[\max\limits_{1\leq i\leq N_p} p_i\right]_k}{\left[\max\limits_{1\leq i\leq N_p} p_i\right]_{k=1}} \leq \varepsilon', \qquad \text{e} \qquad k < 3n$$

forem satisfeitas, então pare o processo, caso contrário volte ao **Passo 4**.

Exemplo 2

Discutiremos, agora, a determinação do perfil ótimo de uma barra de treliça, tal como uma barra da torre de telecomunicação da *Figura 1*, submetida à tração.

Figura 1 - Típica torre de telecomunicação

Foto: Marcelo Araujo da Silva

Exemplo 2

As barras trabalham tipicamente com tração ou compressão, dependendo da direção do vento, tal como mostra a *Figura 2*.

Figura 2 - Esquema das cargas e comportamento das barras

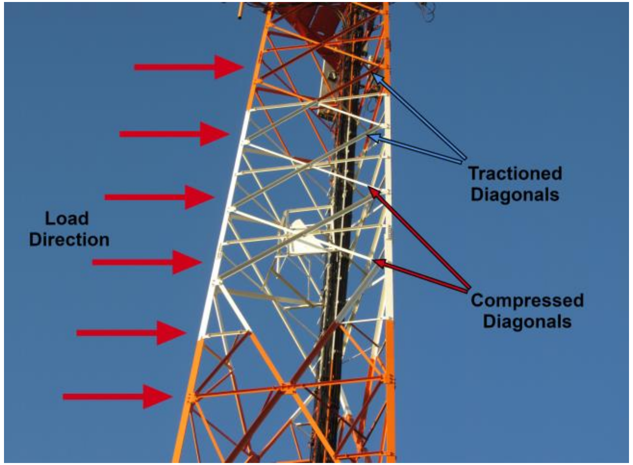
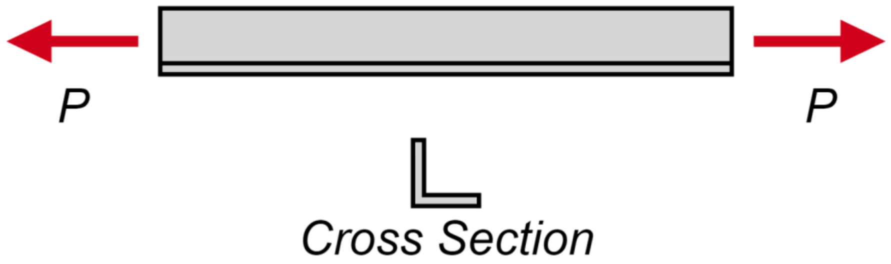


Foto: Marcelo Araujo da Silva

Exemplo 2

A tração atuando na barra não pode exceder um valor admissível. A cantoneira da *Figura 3* está submetida à uma carga de tração $P = 10\ tf$. Feita em aço com $\sigma_a = 2{,}25\ tf/cm^2$, a variável de projeto e a função objetivo é a área A da seção transversal.

Figura 3 - Barra sob tração



Fonte: Elaborado pelos autores

Exemplo 2

O problema de otimização é escrito como se segue:

Determine $b_1 = A$ que minimize $f(b_1) = b_1$, sujeito à restrição

$$g_1 = \frac{P}{A} - \sigma_a \leq 0.$$

A variável de projeto b_1 deve assumir um dos valores mostrados na *Tabela 2*.

Tabela 2 - Valores que podem ser assumidos para o projeto da seção transversal

Posição	Variável		
	b_1		
	Cadeia	Ângulo	Área (cm^2)
1	00	L 2'' x 1/8''	3,13
2	01	L 2'' x 3/16''	4,62
3	10	L 2'' x 1/4''	6,04
4	11	L 2'' x 5/16''	7,43

Exemplo 2

Antes de mais nada, pode-se concluir que o projeto 01 é o de menor área e que satisfaz a restrição g_1. Resolvendo o problema com o AG calcula-se $N_p = 4$, e a população inicial gerada é mostrada na *Tabela 3*.

Tabela 3 - População inicial ($k = 0$)

Posição	Variável b_1			Cálculo da Função de Penalidade				Função de Aptidão		P_i	P_i Acumulado
	Cadeia	Ângulo	Área	f_i	R	Kb_i	p_i	p_{max}	F_i		
1	00	L 2’’ x 1/8’’	3,13	3,13	5305	0,945	5015,8	5015,8	5,016	0,03%	0,03%
2	01	L 2’’ x 3/16’’	4,62	4,62	5305	0,000	4,620	5015,8	5016,2	33,3%	33,4%
3	10	L 2’’ x 1/4’’	6,04	6,04	5305	0,000	7,430	5015,8	5013,3	33,3%	66,7%
4	11	L 2’’ x 5/16’’	7,43	7,43	5305	0,000	6,040	5015,8	5014,7	33,3%	100%

A *Figura 4*, no slide seguinte, ilustra a roleta para a população inicial.

Exemplo 2

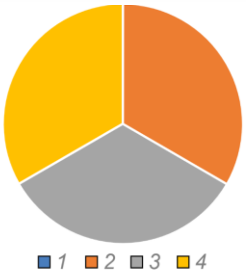


Figura 4 - Roleta de probabilidade

Fonte: Elaborado pelos autores

O projeto escolhido como líder é o 01. A *Tabela 4* mostra o *mating pool*.

Tabela 4 - *Mating pool* (*k = 1*)

Posição	Variável b_1			Cálculo da Função de Penalidade				Função de Aptidão	
	Cadeia	Ângulo	Área	f_i	R	Kb_i	p_i	p_{max}	F_i
1	01	L 2'' x 3/16''	4,62	4,62	5305	0,000	4,620	5015,8	5016,2
2	01	L 2'' x 3/16''	4,62	4,62	5305	0,000	4,620	5015,8	5016,2
3	10	L 2'' x 1/4''	6,04	6,04	5305	0,000	6,040	5015,8	5014,7
4	10	L 2'' x 1/4''	6,04	6,04	5305	0,000	6,040	5015,8	5014,7

Exemplo 2

A *Tabela 5* mostra que os projetos relacionados às cadeias 01 e 10 foram escolhidos para o grupo de *mating*, enquanto os outro, não. Agora, aplicando o cruzamento e a mutação, e mantendo o líder, tem-se a segunda geração (*Tabela 5*).

Tabela 5 - População de segunda geração ($k = 1$)

Posição	Variável			Cálculo da Função de Penalidade				Função de Aptidão		P_i	P_i Acumulado
	b_1										
	Cadeia	Ângulo	Área	f_i	R	Kb_i	p_i	p_{max}	F_i		
1	01	L 2'' x 3/16''	4,62	4,62	5305	0,000	4,620	5015,8	5016,2	33,3%	33,3%
2	00	L 2'' x 1/8''	3,13	3,13	5305	0,945	5015,8	5015,8	5,016	0,03%	33,4%
3	11	L 2'' x 5/16''	7,43	7,43	5305	0,000	7,430	5015,8	5013,3	33,3%	66,7%
4	11	L 2'' x 5/16''	7,43	7,43	5305	0,000	7,430	5015,8	5014,7	33,3%	100%

O algoritmo convergiu em uma única iteração e o projeto ótimo era exatamente a cadeia 01, que é equivalente ao ângulo *L 2'' x 3/16''* com área da seção de *4,62 cm²*.

Dúvidas?

Obrigado!