ALGORITMOS GENÉTICOS

Adair Santa Catarina
Curso de Ciência da Computação
Unioeste – Campus de Cascavel – PR

Fev/2018

Introdução

Algoritmos Genéticos são algoritmos heurísticos de busca, que utilizam regras baseadas numa metáfora do processo evolutivo proposto por Charles Darwin, operando sobre um espaço de soluções codificado.

Holland, 1975; Goldberg, 1989.

Os AGs simulam o mecanismo evolucionário dos sistemas biológicos naturais, onde os indivíduos mais adaptados têm maior probabilidade de sobreviver e se reproduzir, gerando descendentes.

Histórico dos Algoritmos Genéticos

Pesquisas realizadas por John Holland, no início da década de 1970.

Os AGs de Holland manipulavam cadeias de 0 e 1 (cromossomos) e realizavam a evolução simulada de populações destes cromossomos. Manipulando o material contido nos cromossomos obtinham-se indivíduos mais aptos.

Em 1983, Goldberg conseguiu o primeiro sucesso em aplicação industrial de AGs, resolvendo um problema de otimização que envolvia um sistema de bombeamento de combustível.

Características Gerais dos AGs

1. Operam em um conjunto de pontos, denominado população, e não a partir de pontos isolados.

2. Trabalham com soluções codificadas (cromossomos) e não com os próprios parâmetros. Nos cromossomos não está representada informação alguma sobre o problema a resolver.

Goldberg e Luna, 2000.

3. Possuem como informação, acerca do problema que resolvem, apenas o valor da aptidão (*fitness*) estimado pela função de avaliação.

4. Usam transições probabilísticas e não determinísticas.

Um AG Clássico

Um AG, como proposto por Goldberg (1989), inicia com um conjunto de soluções aleatórias codificadas em cadeias de dígitos binários (cromossomos) formando uma população.

Cada indivíduo da população tem o nome de cromossomo e representa uma solução candidata para o problema.

A avaliação de cada indivíduo determina sua aptidão (*fitness*). Indivíduos mais aptos têm maior probabilidade de sobreviver e gerar filhos.

Os cromossomos evoluem através de iterações sucessivas chamadas de gerações.

Um AG Clássico

Processos de seleção, cruzamento e mutação são os responsáveis por criar as novas gerações.

A seleção escolhe indivíduos de acordo com sua aptidão, para depois combiná-los através do operador de cruzamento.

O operador de cruzamento combina características de cromossomos selecionados, gerando novos indivíduos que mantém características de seus pais.

Posteriormente estes novos indivíduos podem ser modificados pelo operador de mutação.

Um AG Clássico

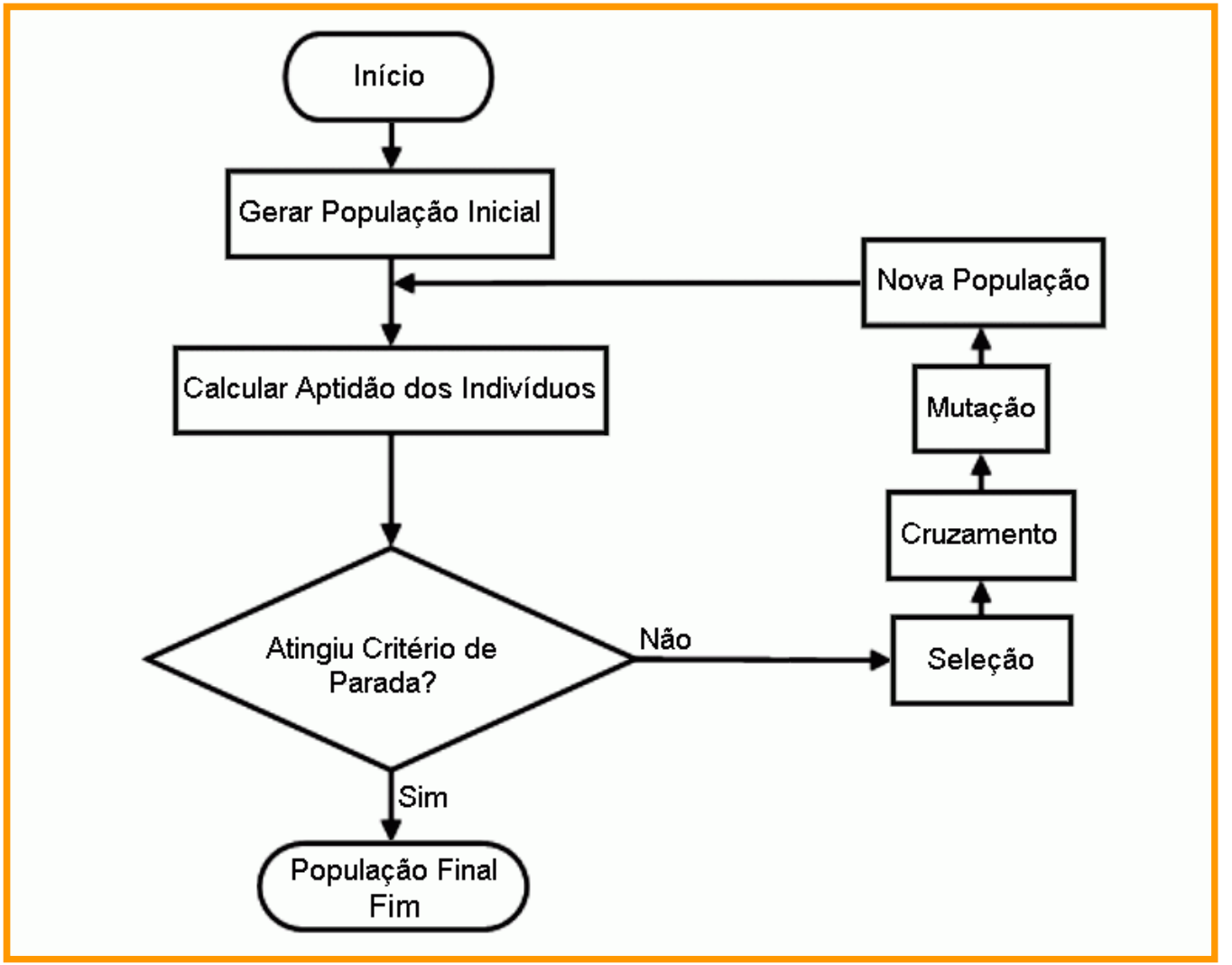
Após várias gerações o AG pode produzir soluções aceitáveis para o problema.

O elemento de ligação entre o AG e o problema a ser resolvido é a função de aptidão. Esta toma como entrada um cromossomo e retorna um número, ou uma lista de números, que representam a medida de performance do cromossomo com relação ao problema a ser resolvido.

O bom desempenho de um AG está condicionado à qualidade da avaliação da aptidão dos indivíduos.

Oliveira, 2000.

Fluxograma de um AG Clássico



Funções dos operadores genéticos

1. Seleção: escolher os indivíduos da população para reprodução. O método mais usado é a roleta, pois prioriza a seleção dos indivíduos mais aptos.

2. Cruzamento: combinar as características de dois pais selecionados na população. Realiza a busca nas imediações (explotação).

3. Mutação: inserir modificações aleatórias nos indivíduos. Contribui na exploração do espaço de busca, na tentativa de evitar ótimos locais.

4. Elitismo: preservar o(s) melhor(es) indivíduo(s) da população, evitando sua perda nos processos evolutivos. Acelera a convergência do AG.

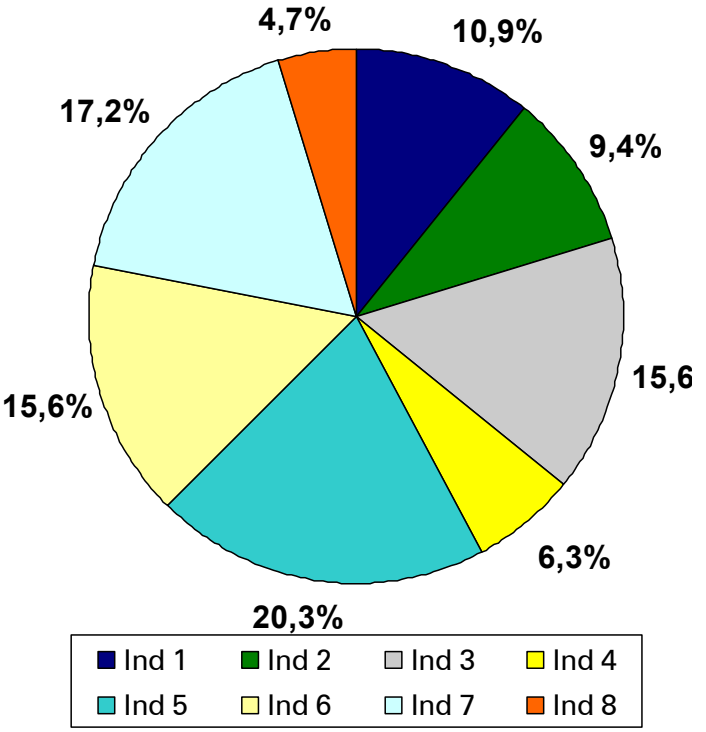
Parâmetros Genéticos

1. Tamanho da população: deve garantir uma boa cobertura do espaço de busca.
2. Taxa de cruzamento: entre 60% e 80% dos pais selecionados devem se recombinar.
3. Taxa de mutação: entre 0,5% e 5%. Taxas maiores tornam o processo em uma busca aleatória.
4. Tamanho da elite: número de melhores indivíduos a serem preservados. Um a cada 50 indivíduos.
5. Intervalo de geração: número de indivíduos que serão substituídos a cada geração.

Exemplo de um Algoritmo Genético

- Encontrar a cadeia de 20 bits com o maior número de bits 1.
- População inicial: 8 indivíduos aleatórios.

Ind.	População inicial	Fitness
1	11001000101000100100	7
2	00011000101010000010	6
3	11001010101010100101	10
4	00001000101000000100	4
5	11001110101011101101	13
6	10101000101001110110	10
7	10111010111000100101	11
8	11000000000000000001	3



Exemplo de um Algoritmo Genético

- Seleção: método da roleta;
  - Sorteiam-se tantos números (0 a ΣFitness) quantos forem os indivíduos na população;
  - Através do Fitness acumulado localizam-se os indivíduos selecionados.

Indivíduo	Fitness ac.
1	7
2	13
3	23
4	27
5	40
6	50
7	61
8	64

Sorteio	Indivíduo	Cromossomo
20	3	11001010101010100101
48	6	10101000101001110110
59	7	10111010111000100101
4	1	11001000101000100100
35	5	11001110101011101101
42	6	10101000101001110110
40	5	11001110101011101101
17	3	11001010101010100101

Exemplo de um Algoritmo Genético

- Cruzamento:
  - Um ponto aleatório (pode ser de dois pontos).
  - Taxa de cruzamento varia de 60 a 80%;
  - Adotada = 80%.

Sorteio	Indivíduo	Cromossomos pais	Cromossomos filhos
50	3	11001010101010100101	11001010101010110110
	6	10101000101001110110	10101000101001100101
37	7	10111010111000100101	10111000101000100100
	1	11001000101000100100	11001010111000100101
27	5	11001110101011101101	11001110101001110110
	6	10101000101001110110	10101000101011101101
81	5	11001110101011101101	11001110101011101101
	3	11001010101010100101	11001010101010100101

Exemplo de um Algoritmo Genético

- Mutação:
  - Taxa de mutação varia de 0,5% a 5%;
  - Adotada = 2%.

Indivíduo	Cromossomos filhos	Cromossomos (mutação)
1	11001010101010110110	11101010101010000101
2	10101000101001100101	10101000101001110110
3	10111000101000100100	11111010111000100101
4	11001010111000100101	11001100101000100100
5	11001110101001110110	11001110101001110110
6	10101000101011101101	10101000101011101111
7	11001110101011101101	11001110101011101101
8	11001010101010100101	11001110101010100101

Exemplo de um Algoritmo Genético

- Após a 1ª geração:

Ind.	População inicial	Fitness	População 1ª geração	Fitness
1	1100100001010001001 00	7	1110101010101000010 1	10
2	0001100010101000001 0	6	1010100010100111011 0	10
3	1100101010101010010 1	10	1111101011100010010 1	12
4	0000100010100000010 0	4	1100110010100010010 0	8
5	1100111010101110110 1	13	1100111010100111011 0	12
6	1010100010100111011 0	10	1010100010101110111 1	12
7	1011101011100010010 1	11	1100111010101110110 1	13
8	1100000000000000001	3	1100111010101010010 1	11
	Média =	8	Média =	11

- Repetir até alcançar a condição de parada.

Convertendo Binário em Real

A codificação binária de variáveis numéricas tornam obrigatória a conversão de binário/inteiro e inteiro/real.

Além disso, cadeias binárias curtas acarretam em perda de precisão numérica. Neste caso, a solução é aumentar o comprimento da cadeia binária, o que acarreta em maior tempo para conversão binário/inteiro.

Como converter um valor binário em real?

Use uma transformação de autoescala!

Convertendo Binário em Real

Seja um número real, no intervalo (X_{min} = -3, X_{max} = 5), a ser representado em uma cadeia binária com Nb = 12 dígitos.

Assumindo que a cadeia binária de X seja:
X_b = 010011011011
A conversão binário/inteiro para X_b resulta em:
X_i = 1243.

X_i pode ser escalonado para X_r por:

$$X_r = X_{min} + [X_i/(2^{Nb} - 1)]*(X_{max} - X_{min})$$

$$X_r = -3 + [1243/(2^{12} - 1)] * (5 - (-3)) = -0,5716728$$

Portanto $010011011011_b = -0,5716728_r$

Representação cromossômica em Reais

Para evitar a conversão binário/inteiro e a autoescala, podemos representar variáveis numéricas pelo seu próprio valor real.

Lucasius e Kaeman, 1989; Davis, 1989.

Neste caso, os operadores de cruzamento e mutação dever ser ajustados para a nova representação.

Operadores de cruzamento aritméticos: média aritmética, média geométrica, BLX-α, aritmético e heurístico.

Operadores de mutação aritméticos: Uniforme, gaussiana, creep, não-uniforme e não-uniforme múltipla.

Cruzamentos para Codificação Real

O cruzamento média aritmética consiste em gerar um novo filho através da média aritmética de seus pais. Seja P_1 e P_2, os pais selecionados. O filho N será dado por:

$$N = (P_1 + P_2)/2$$

Através da média geométrica, o filho N será dado por:

$$N = (P_1 * P_2)^{1/2}$$

Cruzamentos para Codificação Real

O cruzamento *BLX-*α estima o filho N, a partir do cruzamento dos pais P_1 e P_2, através da expressão:

$$N = P_1 + \beta * (P_2 - P_1)$$

$\beta \in \boldsymbol{U}(-\alpha, 1+\alpha)$. α é um pequeno valor que estende os limites para a definição de N e ***U*** é a distribuição de probabilidade uniforme.

O cruzamento aritmético gera dois filhos N_1 e N_2, a partir dos pais P_1 e P_2, através da expressão:

$$N_1 = \beta * P_1 + (1 - \beta) * P_2$$
$$N_2 = (1 - \beta) * P_1 + \beta * P_2$$

$\beta \in \boldsymbol{U}(0, 1)$

Cruzamentos para Codificação Real

O cruzamento heurístico gera um filho a partir da interpolação linear entre os pais usando a informação da aptidão, através da expressão:

$$N = P_1 + r * (P_1 - P_2)$$

$$r \in \boldsymbol{U}(0, 1)$$

$$f(P_1) > f(P_2)$$

Mutações para Codificação Real

A mutação uniforme consiste em substituir o gene selecionado por outro gerado aleatoriamente, segundo uma distribuição uniforme, entre os limites mínimo e máximo.

A mutação gaussiana consiste em substituir o gene selecionado por outro gerado a partir da distribuição $N(\text{pi}, \sigma^2)$, onde pi é o valor do gene e σ^2 é a variância definida pelo usuário. O valor de σ^2 pode diminuir com o passar das gerações.

A mutação *creep* consiste em acrescentar ou subtrair do gene selecionado um valor aleatório obtido de uma distribuição $N(0, \sigma^2)$, com σ^2 pequena. Usada na explotação do espaço de busca.

Mutações para Codificação Real

A mutação não-uniforme consiste em substituir o gene selecionado por outro número extraído de uma distribuição não-uniforme.

A mutação não-uniforme múltipla consiste em aplicar a mutação não-uniforme em todos os genes de um cromossomo selecionado.

Referências Bibliográficas

DAVIS, L. Adapting operator probabilities in genetic algorithms. In: International conference on genetic algorithms, 3, 1989. Fairfax. Proceedings... San Francisco: Morgan Kaufmann, Jun. 1989. p. 61-69.

GOLDBARG, M. C.; LUNA, H. P. L. Otimização combinatória e programação linear: modelos e algoritmos. Rio de Janeiro : Campus, 2000.

GOLDBERG, D. E. Genetic algorithms in search, optimization & machine learning. Reading : Addison-Wesley, 1989.

HOLLAND, J. H. Adaptation in natural and artificial systems. Cambridge: MIT Press, 1992. 228 p.

LUCASIUS, C. B.; KATEMAN, G. Application of genetic algorithms in chemometrics. In: International conference on genetic algorithms, 3, 1989. Fairfax. Proceedings... San Francisco: Morgan Kaufmann, 1989. p. 170-176.

OLIVEIRA, J. R. F. O uso de algoritmos genéticos na decomposição morfológica de operadores invariantes em translação aplicados a imagens digitais. 2000. 110 p. Tese (Doutorado em Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos.