





# Computação Evolutiva



**Prof. Dr. André C. P. L. F de Carvalho**  
**LABIC - ICMSC - USP**


André Ponce de Leon F. de Carvalho - LABIC/USP 1



## Principais Tópicos

- Introdução
- Evolução Natural
- Computação Evolutiva
- Algoritmos Genéticos
- Programação Genética
- Aplicações
- Conclusão


André Ponce de Leon F. de Carvalho - LABIC/USP 2



## Introdução

- Computação Evolucionária
  - Sistemas para a resolução de problemas que utilizam modelos computacionais baseados na teoria da evolução natural
  - Pesquisas tiveram início na década de 50
  - Podem ser agrupados em três grandes categorias:
    - Algoritmos Genéticos
    - Estratégias de Evolução
    - Programação Genética


André Ponce de Leon F. de Carvalho - LABIC/USP 3



## Evolução natural

- Na natureza
  - Indivíduos e populações competem entre si por recursos
    - Alimento
    - Água
    - Abrigo
  - Membros da mesma espécie frequentemente competem para atrair um par


André Ponce de Leon F. de Carvalho - LABIC/USP 4



## Evolução natural

- Na natureza (continuação)
  - Indivíduos mais bem sucedidos na sobrevivência e atração de um par terão, relativamente, mais descendentes
  - Indivíduos mal sucedidos geram poucos ou nenhum descendente
  - Genes dos indivíduos mais adaptados se espalham para um número cada vez maior de indivíduos, geração a geração

André Ponce de Leon F. de Carvalho - LABIC/USP 5



## Evolução natural

- Na natureza (continuação)
  - Combinação de boas características de diferentes ancestrais podem produzir indivíduos ainda mais aptos
    - Espécies evoluem para se tornarem cada vez mais adaptadas aos seus ambiente
  - Genética é a ciência da hereditariedade
    - Estuda mecanismo de transmissão de características de uma espécie, de uma geração para outra

André Ponce de Leon F. de Carvalho - LABIC/USP 6



## Evolução natural

- Provavelmente todos os seres vivos vieram de um mesmo ancestral
  - Forte evidência vem das características compartilhados por espécies que, a primeira vista, parecem não ter nenhuma relação
    - Ex. Asa de pássaros X antebraço dos seres humanos



## Evolução natural

- Asa X antebraço
  - Morfologia geral
    - Virtualmente idêntica relação de ossos e músculos
    - Ancestrais comuns compartilhavam estes elementos
  - Morfologia específica
    - com penas X sem penas
    - sem dedos X com dedos
  - Função
    - vôo X manipulação de objetos



## Fatos históricos

- 1809 Lamarck publica *Philosophie Zoologique*
  - Seres vivos modificam-se através do tempo
    - Lei do uso e desuso
    - Lei da transmissão das características adquiridas
- 1859 Charles Darwin publica o livro *The origin of Species*
  - Mecanismos de transformação das espécies



## Fatos históricos

- 1865 Mendel apresenta experimentos em cruzamento genético de ervilhas
  - Pai da genética
- 1960 início da Computação Evolucionária
  - Holland define princípios básicos de Algoritmos Genéticos
  - Rechenberg propões as estratégias evolucionárias



## Algoritmos Genéticos (AGs)

- Métodos adaptativos que podem ser utilizadas para resolver problemas de busca e otimização
  - São baseados nos processos genéticos de organismos biológicos
    - Depois de várias gerações populações naturais evoluem de acordo com os princípios de seleção natural e sobrevivência dos mais aptos (*Charles Darwin, A Origem das Espécies*)



## Algoritmos Genéticos

- Podem “evoluir” soluções para problemas do mundo real
  - Problemas devem ser adequadamente codificados
  - Deve haver uma forma de avaliar as soluções apresentadas



## Algoritmos Genéticos

- Desenvolvido por John Holland e sua equipe (popularizado por David Goldberg)
- Objetivos:
  - Abstrair e explicar rigorosamente os processos adaptativos dos sistemas naturais
  - Desenvolver sistemas artificiais que conservam mecanismos importantes dos sistemas naturais



## Algoritmos Genéticos

- Baseados na genética e teoria da seleção natural
- Utilizam uma população de soluções candidatas (indivíduos)
- Otimização ocorre em várias gerações
  - A cada geração
    - Mecanismos de seleção selecionam os indivíduos mais aptos
    - Operadores de reprodução geram novos indivíduos

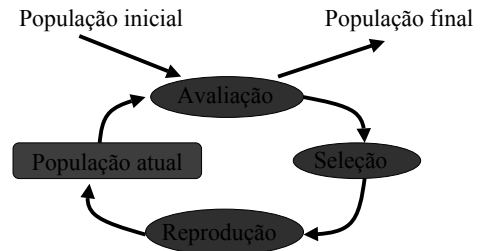


## Algoritmos Genéticos

- Cada indivíduo representa uma possível solução para um dado problema
- A cada indivíduo é associado um escore de aptidão, que mede o quão boa é a solução que ele representa
- Indivíduos mais aptos têm mais oportunidades de serem reproduzidos
  - Produzindo descendentes cada vez mais aptos



## Algoritmos Genéticos



## Princípios básicos

- Indivíduo
- Codificação
- Função de aptidão
- Reprodução
- Convergência



## Indivíduo

- Possível solução para um dado problema
  - Também chamado de cromossomo ou string
- Codificado como vetor de características
- A cada indivíduo é associado um valor de aptidão
  - Mede qualidade da solução que ele representa
- População
  - Conjunto de indivíduos



## Exemplo: preparo de biscoitos

- Otimizar número de colheres de açúcar e farinha de trigo para preparar biscoitos
- Passos
  - Criar população inicial
  - Codificar strings ou cromossomos
  - Definir função de aptidão
  - Reprodução

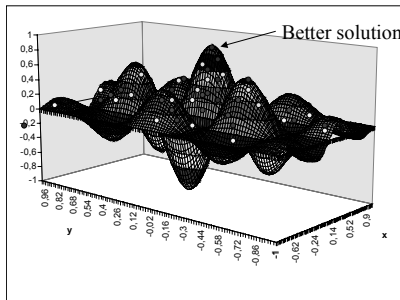


## Exemplo

		0.4	0.6	0.5
Farinha de trigo		0.7	0.9	0.6
		0.5	0.7	0.4
		Açúcar		



## Espaço de Busca



## Codificação

- Cada indivíduo é codificado por um conjunto de parâmetros (genes)
  - Genes podem assumir valores:
    - Binários
    - Inteiros
    - Reais
- Parâmetros são combinados para formar strings ou vetores (cromossomos)



## Codificação

- Genótipo
  - Conjunto de parâmetros representado por um cromossomo
  - Contém informação necessária para construir um organismo (fenótipo)
  - Fenótipo codificado
- Fenótipo
  - Produto da iteração de todos os genes
  - Aptidão de um indivíduo depende do desempenho de seu fenótipo (inferido do genótipo usando função de aptidão)



## Codificação

- Tradicionalmente, os indivíduos são representados por vetores binários
  - 1 = presença
  - 0 = ausência
  - Esta representação é independente do problema
  - Permite a utilização dos operadores de reprodução padrão



## Codificação

- Genes também podem assumir valores inteiros, reais ou de tipos abstratos
- Representações em níveis abstratos mais altos
  - Facilitam sua utilização em determinados domínios, onde essa transformação “fenótipo - genótipo” é muito complexa
  - Necessitam de operadores específicos



## Função de aptidão

- Mede o grau de aptidão de um indivíduo
  - Retorna um valor (índice) de aptidão numérico
    - Proporcional a utilidade ou habilidade do indivíduo
  - Aptidão = probabilidade do indivíduo sobreviver para a próxima geração
- Pode envolver uma (otimização de função) ou mais medidas (otimização multi-objetivo)
  - Ex. projeto de ponte
    - Custo, tempo de construção, capacidade máxima



## Função de aptidão

- É aplicada ao fenótipo do indivíduo
  - Genótipo precisa ser decodificado, recuperando o fenótipo associado
- Define valor (índice) de aptidão
  - Cada aplicação tem sua própria função de aptidão
  - Exemplo
    - Para o caso do biscoito, a função de aptidão poderia ser a nota dada por uma pessoa para o gosto de cada biscoito



## Função de aptidão

- Métodos para calcular valor ou índice de aptidão
  - Padrão
  - Baseada em rank
  - Rank-espaço



## Função de aptidão padrão

- Utiliza apenas informação sobre “qualidade do cromossomo”

$$f_i = \frac{q_i}{\sum_j q_j} \quad q = \text{índice de aptidão do cromossomo}$$



## Exemplo

CROMOSSOMO	GRAU	APT. PADRÃO
1 4	4	0.4
3 1	3	0.3
1 2	2	0.2
1 1	1	0.1



## Aptidão baseada em rank

- Aptidão padrão: escolha infeliz da escala do valor de aptidão pode prejudicar seleção
  - Ex.: valores de aptidão 1, 10, 100
- Aptidão baseada em rank utiliza medida de qualidade apenas para definir um “ranking” de cromossomos por aptidão
  - Utiliza depois um algoritmo para redefinir a aptidão de cada indivíduo



## Aptidão de candidato por rank

1. Classificar os N indivíduos por aptidão
2. Definir um valor  $p$  dentro do intervalo (0,5; 1,0)
3.  $q = 1$ ;  
Repetir até o penúltimo indivíduo
  - 3.1 Selecionar indivíduo de maior aptidão da lista
  - 3.2 Aptidão baseada em rank deste indivíduo =  $p * q$
  - 3.3 Retirar este indivíduo da lista
  - 3.4  $q = q - p * q$ ;
4. Aptidão do último indivíduo = 1 - (soma das aptidões baseadas em rank dos demais indivíduos)



## Exemplo

- Seja  $p = 0.667$

INDIV.	VALOR DE APT.	APT. PADRÃO	RANK	APT. RANK
1 4	4	0.4	1	0.667
3 1	3	0.3	2	0.222
1 2	2	0.2	3	0.074
1 1	1	0.1	4	0.025



## Aptidão rank-espço

- Medidas de aptidão anteriores ignoram diversidade
  - Diversidade mede o quão diferente são os cromossomos de uma população
- Princípio da diversidade
  - Ser diferente pode ser tão bom quanto ser apto



## Aptidão rank-espço

- Medida de diversidade
  - Utiliza rank das medidas:

$$D_i = \sum_j d_{ij}^2$$

- Aptidão rank-espço

$$Rank_i = Rank_{f_i} + Rank_{D_i}$$



## Aptidão rank-espço

1. Selecionar indivíduo com maior aptidão baseada em rank
2. Enquanto a população não estiver completa  
Selecionar indivíduo com maior aptidão rank-espço (diversidade é medida com relação aos indivíduos já selecionados)  
Em casa de empate, desempatar por rank ou diversidade)



## Exemplo: preparo de biscoitos

- População inicial
  - (1, 4), (1, 1), (1,2), (1,1)

### Codificação

- Quantidade de farinha de trigo e de açúcar

### Função de aptidão

$$f_i = \frac{q_i}{\sum_j q_j} \quad q = \text{índice da combinação}$$



## Seleção

- Escolhe preferencialmente, embora não exclusivamente, indivíduos com maiores notas de aptidão
  - Procura de manter a diversidade da população
- Indivíduos mais aptos têm mais oportunidades de serem reproduzidos
  - Produzindo descendentes cada vez mais aptos



## Seleção

- Seleciona uma população intermediária
  - Onde serão aplicados os operadores de reprodução
- Existem vários métodos de seleção
  - Pela roleta
  - Por sorteio
  - Amostragem Universal Estocástica



## Seleção pela roleta

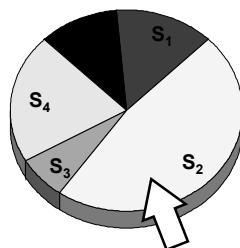
- Método mais simples e utilizado
- Escolhe indivíduos para fazer parte da próxima geração através de um sorteio
- Cada indivíduo da população é representado na roleta por uma fatia proporcional ao seu índice de aptidão
  - Quanto maior o desempenho, maior é a chance de ser selecionado para a próxima geração



## Seleção pela roleta

Método da Roleta baseado em Aptidão Relativa

Indivíduo	Aptidão $f(S_i)$	Aptidão Relativa
$S_1$	10110	2.23
$S_2$	11000	7.27
$S_3$	11110	1.05
$S_4$	01001	3.35
$S_5$	00110	1.69



## Seleção por torneio

- Escolhe  $n$  indivíduos da população aleatoriamente, com a mesma probabilidade
  - Geralmente  $n = 3$
- Cromossomo com maior aptidão dentre estes  $n$  cromossomos é selecionado para a população intermediária
- Processo se repete até que a população intermediária seja preenchida



## Seleção por torneio

Método da Torneio baseado em Aptidão Relativa

Indivíduo	Aptidão $f(S_i)$	Aptidão Relativa	Supondo $n = 3$ Candidatos $\Rightarrow$ vencedor	
$S_1$	10110	2.23	0.14	$S_1, S_2, S_5 \Rightarrow S_2$
$S_2$	11000	7.27	0.47	$S_2, S_4, S_5 \Rightarrow S_2$
$S_3$	11110	1.05	0.07	$S_5, S_1, S_3 \Rightarrow S_1$
$S_4$	01001	3.35	0.21	$S_4, S_5, S_3 \Rightarrow S_4$
$S_5$	00110	1.69	0.11	$S_3, S_1, S_5 \Rightarrow S_1$

André Ponce de Leon F. de Carvalho - LABIC/USP

43



## Seleção por Amostragem Universal Estocástica

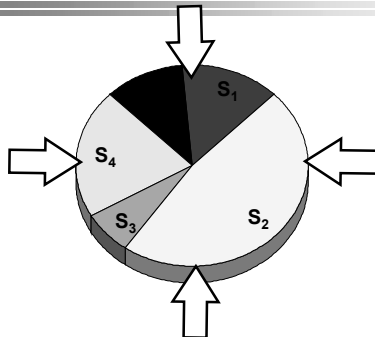
- Conhecido como SUS (do inglês, *Stochastic Universal Sampling*)
- Variação do método da roleta
  - Utiliza  $n$  agulhas igualmente espaçadas ao invés de 1
    - $n$  é o número de indivíduos a serem selecionados para a próxima geração
    - Ao invés de  $n$  vezes, a roleta é girada uma única vez
  - Exibe menos variância que as repetidas chamadas do método da roleta

André Ponce de Leon F. de Carvalho - LABIC/USP

44



## Seleção por Amostragem Universal Estocástica



André Ponce de Leon F. de Carvalho - LABIC/USP

45



## Reprodução

- Permite obtenção de novos indivíduos
  - Cada geração possui, geralmente, indivíduos mais aptos
- Utiliza operadores genéticos
  - Transformam a população
    - Crossover (cruzamento ou recombinação)
    - Mutação

André Ponce de Leon F. de Carvalho - LABIC/USP

46



## Crossover

- Recombinação de características dos pais durante a reprodução
  - Permite que as próximas gerações herdem essas características
- Funcionamento
  - Escolhe dois indivíduos e troca trechos entre eles

André Ponce de Leon F. de Carvalho - LABIC/USP

47



## Crossover

- É o operador genético predominante
  - A taxa de *crossover* deve ser maior que a taxa de mutação
  - Taxa de *crossover*:  $0.6 < P_c < 1.0$
  - Caso *crossover* não seja aplicado, descendentes são iguais aos pais
- É a operação mais importante para exploração rápida do espaço de busca

André Ponce de Leon F. de Carvalho - LABIC/USP

48



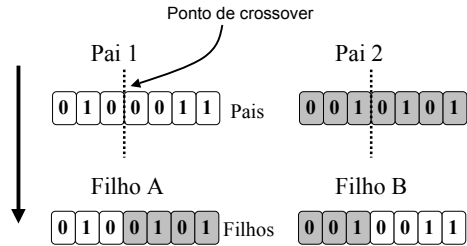


# Crossover

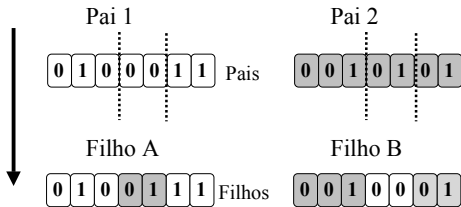
- Diversas variações
  - Um ponto
    - Mais comum
  - Dois pontos
  - Multi-pontos
  - Uniforme



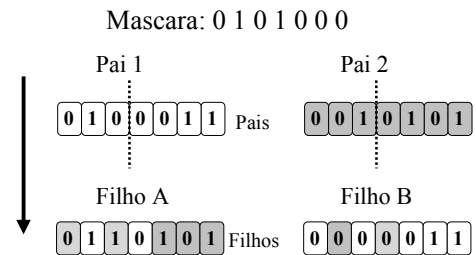
# Crossover 1 ponto



# Crossover de 2 pontos



# Crossover uniforme

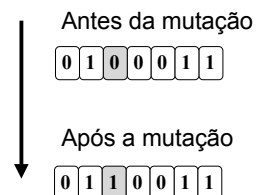


# Mutação

- Introdução e manutenção da diversidade genética
  - Aplicado a cada indivíduo após crossover
- Altera randomicamente um ou mais componentes de uma estrutura escolhida
- Assegura que a probabilidade de atingir qualquer ponto do espaço de busca nunca será zero
- Operador genético secundário
  - Taxa de mutação pequena  $P_m \cong 0.001$



# Mutação





## Elitismo

- Indivíduo de maior desempenho é automaticamente selecionado
- Evita modificações deste indivíduo pelos operadores genéticos
  - Utilizado para que os melhores indivíduos não desapareçam da população pela manipulação dos operadores genéticos



## Observações

- Se o AG estiver corretamente implementado, a população evolui em gerações sucessivas
- Aptidão do melhor indivíduo e do indivíduo médio aumentam em direção a um ótimo global



## Critério de parada

- Tempo de execução
- Número de gerações
- Valor de aptidão mínimo, médio e/ou máximo
- Convergência
  - Nas últimas k iterações não houve melhora nas aptidões



## Convergência

- Convergência é o progressão em direção à uma uniformidade crescente
  - Um gene converge quando 95% da população compartilha o mesmo valor
  - A população converge quando todos os genes tiverem convergido
    - Perda de diversidade



## Escolha de parâmetros

- Escolhidos de acordo com o problema
  - Quantos cromossomos em uma população
    - Poucos ⇒ efeito pequeno do *crossover*
    - Muitos ⇒ aumenta tempo de computação
  - Taxa de mutação
    - Baixa ⇒ mudanças lentas
    - Alta ⇒ traços desejados não são mantidos (caos)



## Escolha de parâmetros

- Outros parâmetros
  - Quantos indivíduos selecionados para reprodução?
  - Quantos pontos de *crossover*?
  - Critério para medir aptidão?
- Manter limites no tamanho da população e complexidade da análise
  - Algoritmo pode se tornar ineficiente



# Um Algoritmo Genético

1. Escolher população inicial de cromossomos
2. Avaliar cada cromossomo da população
3. Enquanto não convergir
  - 3.1 Selecionar indivíduos mais aptos
  - 3.2 Criar novos cromossomos aplicando operadores genéticos
  - 3.3 Avaliar cada cromossomo da população



# Comparação com outras técnicas

- Várias técnicas têm sido propostas para resolver problemas de busca e otimização
  - Todas assumem que o problema é definido por uma função de aptidão que precisa ser maximizada
  - Várias técnicas são aplicadas apenas a domínios específicos (ex. programação dinâmica)



# Exemplo

Utilizando Algoritmos Genéticos, achar o máximo da função :

$$f(x) = x \text{ seno}(10\pi x) + 1,0$$

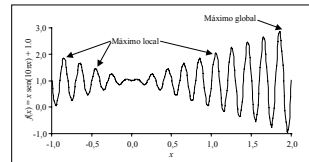
Restrita no intervalo:

$$-1,0 \leq x \leq 2,0$$



# Exemplo

Gráfico da função  $f(x) = x \text{ seno}(10\pi x) + 1.0$



Máximo global:  
 $x = 1,85055$   
 $f(x) = 2,85027$

Pode ser observado que existem vários pontos de mínimo



# Representação dos cromossomos

- Representar o único parâmetro deste problema (a variável  $x$ ) na forma de um cromossomo
  - Binário = 1000101110110101000111
  - Decimal =  $(1000101110110101000111)_2 = 2288967$
  - Valor de  $x$  precisa estar no intervalo  $[-1,0; 2,0]$

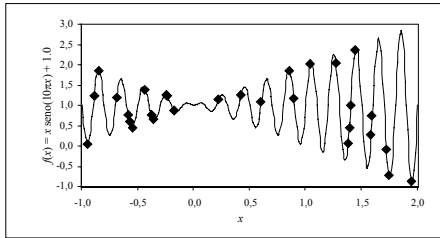


# Geração da população inicial

Rand. $t$	Cromossomo $s_t$	$x_t$	Função objetivo $f(x_t)$	Aptidão $f_t$	Aptidão acumulada $\sum_{i=1}^t f_i$
1	11010000001110110111	1,43891	2,35231	2,30000	2,30000
2	1100001110100011111	1,29925	2,04416	1,91101	5,31101
3	1010111001010110010000	1,04101	2,01192	1,86201	7,17301
4	0001100001100100000000	0,05237	1,84862	1,79701	8,97001
5	1001101101110000111000	0,84820	1,84706	1,72410	9,11033
6	0000110011110100101100	-0,44792	1,84610	1,65511	10,76543
7	0011000000010110101010	-0,51970	1,39548	1,26962	12,51722
8	0111100101000001011000	0,42098	1,25739	1,17252	14,06897
9	0100000001000110100000	-0,24761	1,24850	1,04820	15,51722
10	01000001000111011110	-0,24141	1,21822	1,17911	16,89633
11	0000101000000110101010	-0,09170	1,23144	1,01810	18,00909
12	0001101001100010101111	-0,09070	1,19704	1,24130	19,44823
13	1010000110011000011011	0,49970	1,17252	1,17241	20,62069
14	01010000010110000100	0,22292	1,44899	1,01840	21,24144
15	1000100011100010000111	0,60870	1,09033	1,01440	22,51862
16	110011001000100001110	1,19980	0,99843	0,90550	23,24144
17	0100011001000110101101	-0,11633	0,88140	0,89651	24,23099
18	00110100011101001001000	-0,37961	0,77140	0,82750	25,46224
19	00100010100110011000	-0,36631	0,75322	0,76661	26,00000
20	1101110101101111111111	1,59892	0,74988	0,68960	26,79960
21	0011001001000110011010	-0,33970	0,66321	0,62000	27,51722
22	00100100000111100111	-0,37440	0,58121	0,51722	28,00977
23	10001111010001010000	1,18870	0,64310	0,68220	28,51722
24	00100100100100100111	-0,34992	0,45000	0,41370	28,93100
25	11011100100100100000	1,84890	0,27101	0,44480	29,31033
26	11000011010001000000	1,37891	0,06101	0,36000	29,69033
27	00001000010010010001	-0,93140	0,04933	0,20900	29,93133
28	111010000100000000000	1,72400	-0,08120	0,17901	29,81033
29	111010100000000000000	1,74400	-0,22390	0,08490	30,00000
30	111101000000000000000	1,94401	-0,36710	0,00000	30,00000



## Geração da população inicial



## Exemplo 2

- Problema:  $\min f(x) = x^2$ ,  $x \in [-10, +10]$  e inteiro
  - Representação → cada inteiro como um cromossomo de 4 bits
    - 0 = (0,0,0,0)
    - 7 = (0,1,1,1)
  - População inicial → aleatória
  - População de tamanho 4
  - Função de aptidão →  $f(x)$



## Exemplo 2

- População inicial gerada:

	x	f(x)	
$A_1 \rightarrow$	1101	-5	25 ← Pai 1
$A_2 \rightarrow$	1111	-7	49
$A_3 \rightarrow$	0010	2	4 ← Pai 2
$A_4 \rightarrow$	0011	3	9

- Escolha dos pais com maior aptidão pela roleta
- Gera 2 filhos, que substituirão indivíduos com menor aptidão



## Exemplo 2

	x	f(x)	
$\text{Pai}_1 = A_1 \rightarrow$	1101	-5	25
$\text{Pai}_2 = A_4 \rightarrow$	0010	2	4
$\text{Filho}_1 \rightarrow$	1110		
$\text{mutação}_1 \rightarrow$	1100	-4	16
$\text{Filho}_2 \rightarrow$	0001	1	1

	x	f(x)	
$A_1 \rightarrow$	1100	-4	16
$A_2 \rightarrow$	0001	1	1
$A_3 \rightarrow$	0010	2	4
$A_4 \rightarrow$	0011	3	9



## Exercício

- Encontrar de x para o qual a função  $f(x) = x^2 - 3x + 4$  assume o valor mínimo
  - Assumir que  $x \in [-10, +10]$
  - Codificar X como vetor binário
    - Usar 5 bits, primeiro bit é o sinal (1-positivo, 0-negativo)
  - Criar uma população inicial com 4 indivíduos
  - Utilizando mutação em apenas um gene da população e *crossover de um ponto*
  - Definir o valor mínimo após no máximo 10 gerações



## Aplicações

- Otimização de função numérica
- Otimização combinatoria
  - Problema do caixeiro viajante
  - Problema de empacotamento
  - Alocação de recursos (*job shop scheduling*)
- Projetos
  - Projeto de pontes
- Aprendizado de Máquina
  - Jogos



## Programação Genética

- Um dos principais desafios da Computação:
  - Conseguir que computadores façam as tarefas requeridas sem que a necessidade de informá-los como fazê-las
  - Programação Genética, GP
    - Originada na década de 80
    - Fornece um método para criação automática de programas
      - A partir de uma descrição em alto nível do problema a ser atacado



## Programação Genética

- Evolui uma população de programas utilizando princípios da genética e evolução natural
- Indivíduos de uma população
  - Programas de computador ao invés de cadeias de valores binários ou reais
    - Representados por árvores, ao invés de linhas de código
    - Podem apresentar tamanhos e formatos diferentes



## Programação Genética

- Avaliação de cada indivíduo ocorre através da execução do programa representado pelo indivíduo
- População inicial
  - Conjunto de programas de computador, usualmente gerados de forma aleatória
    - Devem ser programas sintaticamente válidos, que podem ser executados



## Programação Genética

- Papel de cada nó da árvore no programa depende de sua posição na árvore
  - Nós internos (conjunto funcional)
    - Funções aritméticas, funções lógicas operadores condicionais, atribuições e funções específicas do domínio onde o programa será utilizado
  - Nós terminais ou folhas (conjunto terminal)
    - Variáveis, constantes ou entradas externas (operações de leitura)



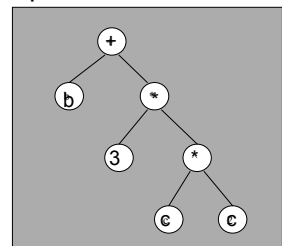
## Programação Genética

- Programas mais complexos
  - Cada chamada de função pode ser representado por um nó interno da árvore
    - Nós descendentes deste nó podem ser os argumentos da função



## Exemplo

- Programa que calcula o valor de:  $b + 3 * c^2$





## Programação Genética

- Aptidão de um indivíduo
  - Determinada executando o programa que ele representa com um conjunto de dados de entrada
    - O desempenho do programa define sua aptidão
  - Após várias gerações, pode ser obtido um programa que resolve completa ou aproximadamente o problema apresentado



## Programação Genética

- Operadores de reprodução
  - Adaptados para lidar com a estrutura de árvores
  - Basicamente, três operadores genéticos são utilizados:
    - Cruzamento
      - Operador genético predominante
    - Mutação
    - Alteração de arquitetura



## Cruzamento

- Troca sub-árvores de indivíduos
  - Pais geralmente possuem tamanhos e formatos diferentes
    - Um ponto de cruzamento é escolhido aleatoriamente em cada pai
    - A sub-árvore do pai 1 é substituída pela sub-árvore do pai 2
    - Operação inversa ocorre para o pai 2

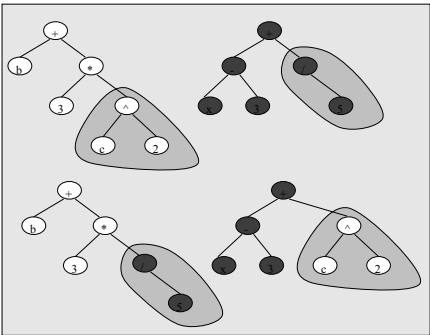


## Cruzamento

- Filhos diferentes podem ser produzidos por pais iguais
  - Basta que os pontos de cruzamento selecionados sejam diferentes



## Cruzamento

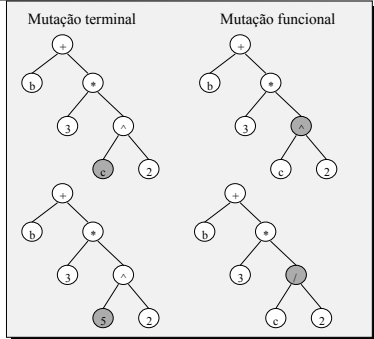


## Mutação

- Dois tipos de operação de mutação:
  - Substituição de um nó da árvore por um outro nó
    - Terminal por terminal
    - Funcional por funcional
  - Substituição de uma sub-árvore por outra sub-árvore
    - Gerada aleatoriamente
    - Geralmente, PG não utiliza operadores de mutação



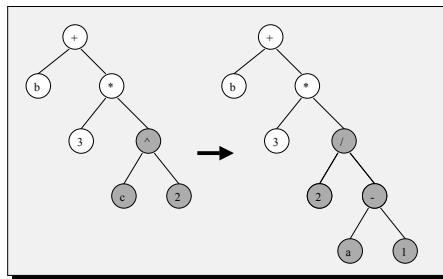
## Mutação



85



## Mutação



André Ponce de Leon F. de Carvalho - LABIC/USP

86



## Exercício

- Achar a expressão aritmética que implementa a função  $f(x)$ :

x	y	$f(x, y)$
0	1	-2
0	0	-4
1	0	5
2	1	4
1	2	-3
2	2	0

André Ponce de Leon F. de Carvalho - LABIC/USP

87



## Exercício

- Nós folhas podem assumir os valores:
  - $T = \{x, y, 1, +2, +3, +4, +5\}$
- Nós internos podem assumir os valores:
  - $N = \{+, -, *\}$
- Tamanho máximo da árvore: 3 arestas

André Ponce de Leon F. de Carvalho - LABIC/USP

88



## Conclusão

- Conceitos básicos
- Evolução Natural
- Algoritmos genéticos
  - Codificação
  - Função de aptidão
  - Operadores Genéticos
  - Reprodução
- Programação Genética

André Ponce de Leon F. de Carvalho - LABIC/USP

89