Algoritmos Genéticos

Computação Bioinspirada

Visão geral

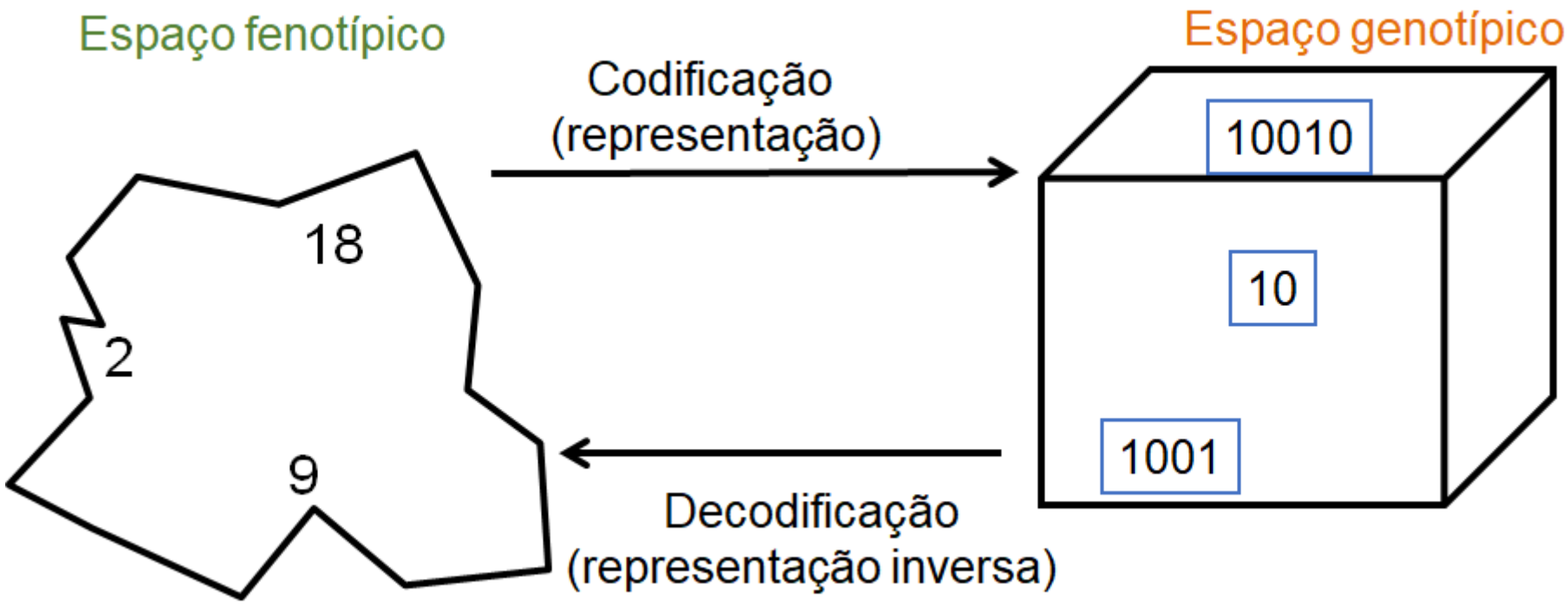
- Desenvolvido: EUA na década de 1970
- Pioneiros: J. Holland, K. De Jong, D. Goldberg
- Normalmente aplicado a:
  - otimização discreta
- Recursos atribuídos:
  - não muito rápido
  - boa heurística para problemas combinatórios
- Características especiais:
  - tradicionalmente enfatiza a combinação de informações de bons pais (cruzamento)
  - muitas variantes, por exemplo, modelos de reprodução, operadores

Algoritmos genéticos

- O GA original de Holland é agora conhecido como algoritmo genético simples (SGA)

- Outros GAs usam diferentes:
  - Representações
  - Mutações
  - Recombinações
  - Mecanismos de seleção

Representação

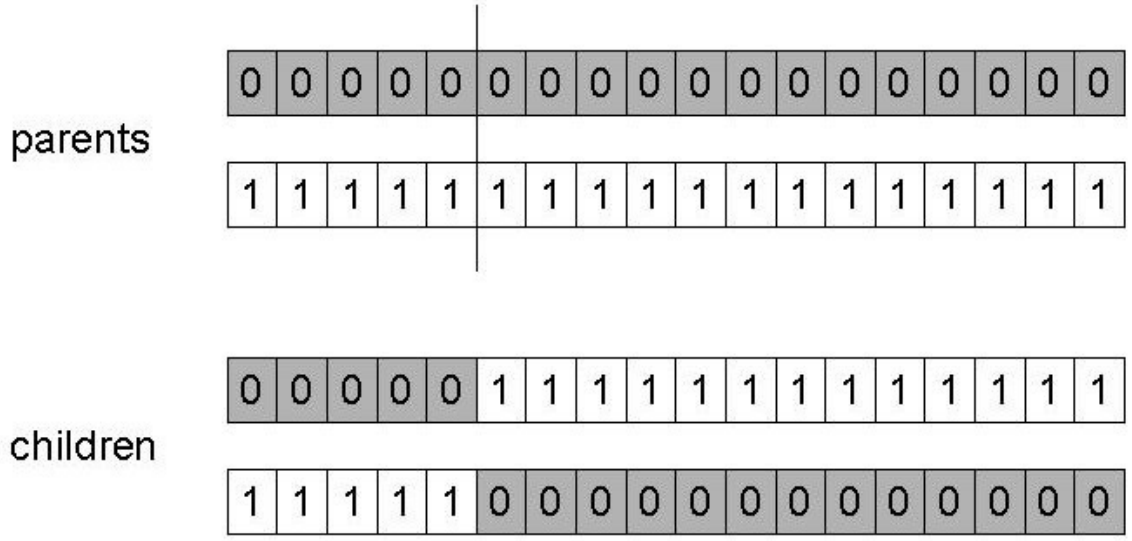


Estrutura básica

1. Selecione os pais para um conjunto (*pool*) de acasalamento
   - (tamanho do conjunto = tamanho da população)
2. Misture os elementos do conjunto
3. Para cada par consecutivo, aplique o cruzamento com a probabilidade p_c, caso contrário, copie os pais
4. Para cada filho, aplique a mutação (*bit-flip* com probabilidade *pm* independentemente para cada bit)
5. Substitua toda a população com a prole resultante

Operadores do SGA: crossover de 1 ponto

- Escolha um ponto aleatório nos dois pais
- Divida os pais neste ponto de cruzamento
- Crie filhos trocando caudas

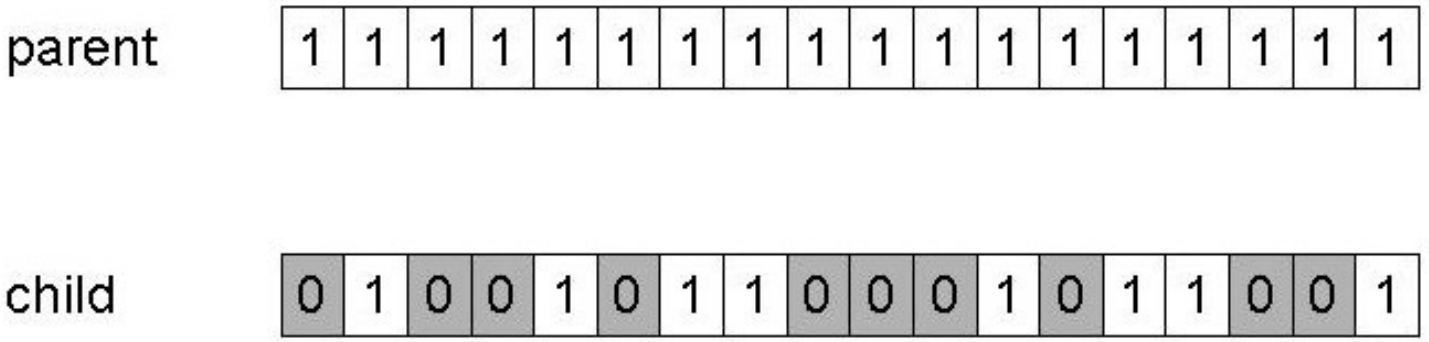


- p_c: tipicamente entre 0.6 e 0.9

Operadores do SGA: mutação

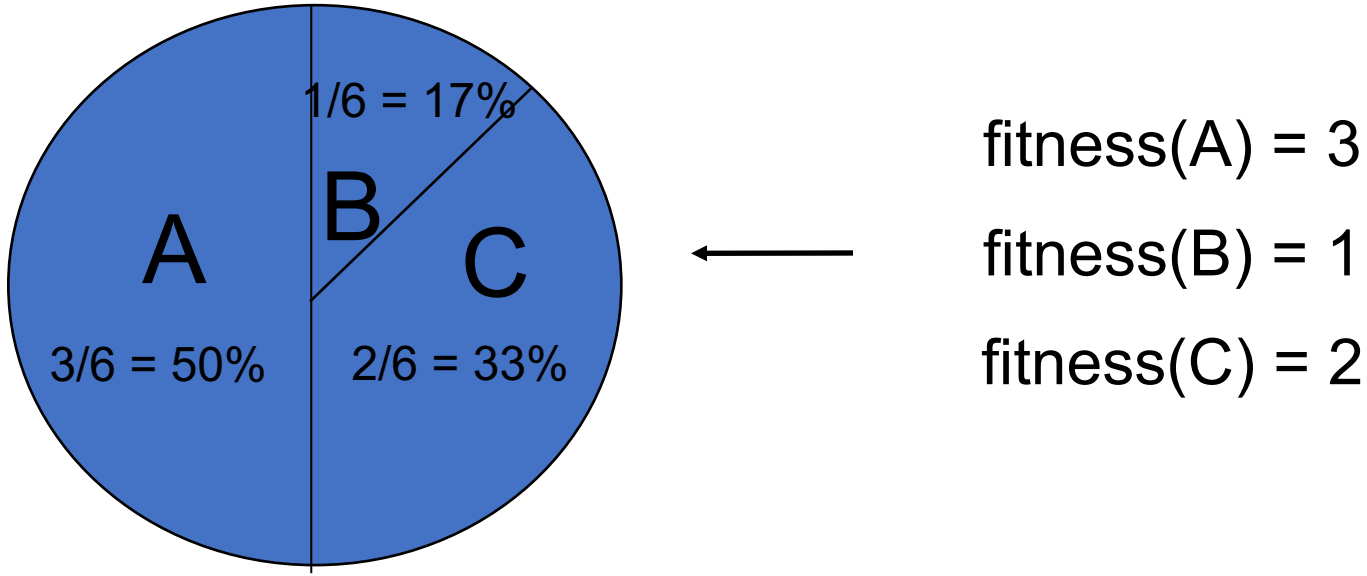
- Altere cada gene independentemente com uma probabilidade p_m
  - p_m é chamada de taxa de mutação

- Tipicamente entre 1/pop_size e 1/chromosome_length



Operadores do SGA: Selecão

- Ideia principal: melhores indivíduos têm maior chance
  - Chances proporcionais ao *fitness*
- Implementação: técnica da roda de roleta
  - Atribua a cada indivíduo uma parte da roda da roleta
  - Gire a roda *n* vezes para selecionar *n* indivíduos



Exemplo: Goldberg, 1989 (1)

- Problema: max x^2 em {0,1,…,31}

- Abordagem do GA:
  - Representação: Código binário, por exemplo: 01101 $\leftrightarrow$ 13
  - Tamanho da população: 4
  - Crossover de 1 ponto (*1-point xover*)
  - Mutação por inversão de bit (*bitwise*)
  - Seleção por Roleta
  - Inicialização aleatória

Exemplo: Goldberg, 1989 (2)

String no.	Initial population	x Value	Fitness $f(x) = x^2$	$Prob_i$	Expected count	Actual count
1	0 1 1 0 1	13	169	0.14	0.58	1
2	1 1 0 0 0	24	576	0.49	1.97	2
3	0 1 0 0 0	8	64	0.06	0.22	0
4	1 0 0 1 1	19	361	0.31	1.23	1
Sum			1170	1.00	4.00	4
Average			293	0.25	1.00	1
Max			576	0.49	1.97	2

Exemplo: Goldberg, 1989 (3)

String no.	Mating pool	Crossover point	Offspring after xover	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 | 1	4	0 1 1 0 0	12	144
2	1 1 0 0 | 0	4	1 1 0 0 1	25	625
2	1 1 | 0 0 0	2	1 1 0 1 1	27	729
4	1 0 | 0 1 1	2	1 0 0 0 0	16	256
Sum					1754
Average					439
Max					729

Exemplo: Goldberg, 1989 (4)

String no.	Offspring after xover	Offspring after mutation	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 0	1 1 1 0 0	26	676
2	1 1 0 0 1	1 1 0 0 1	25	625
2	1 1 0 1 1	1 1 0 1 1	27	729
4	1 0 0 0 0	1 0 1 0 0	18	324
Sum				2354
Average				588.5
Max				729

O GA típico

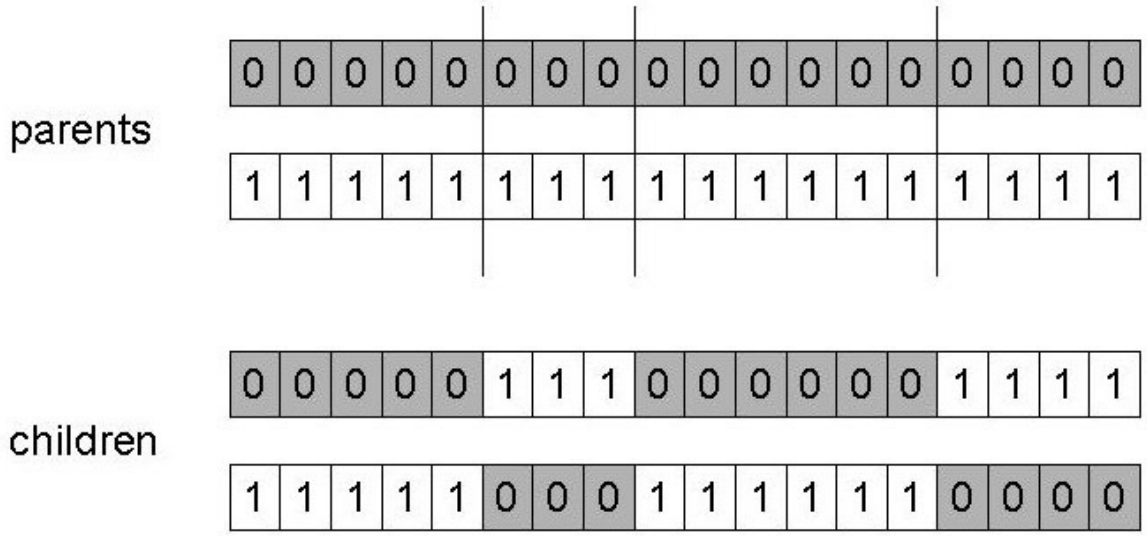
- Tem sido objeto de muitos estudos
  - ainda frequentemente usado como referência para novos GAs
- Mostra muitas deficiências, por exemplo
  - A representação é muito restritiva
  - Mutação e cruzamentos aplicáveis apenas para representações de cadeia de bits e inteiros
  - Mecanismo de seleção sensível para populações convergentes com valores de aptidão próximos
  - O modelo de população geracional (etapa 5) pode ser melhorado com a seleção explícita de sobreviventes

Operadores de Recombinação Alternativos

- O desempenho com cruzamento de 1 ponto depende da ordem em que as variáveis ocorrem na representação
  - mais propensos a manter juntos genes que estão próximos uns dos outros
- Nunca consegue manter juntos genes de extremidades opostas da cadeia
- Isso é conhecido como viés posicional
- Pode ser explorado se conhecermos a estrutura do nosso problema, mas geralmente não é o caso

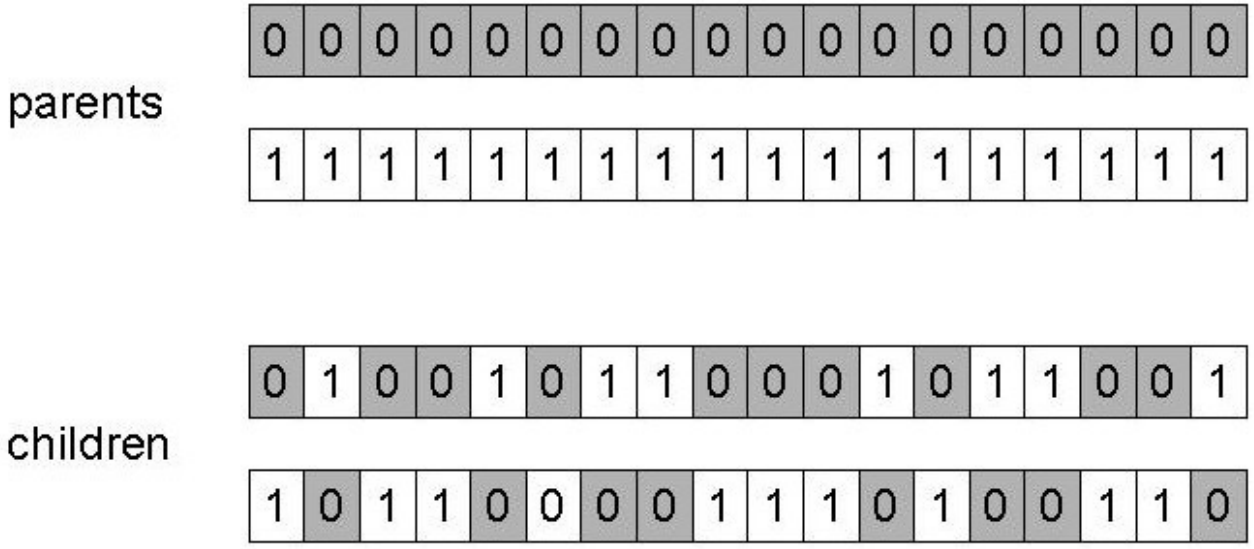
Recombinação de *n* pontos

- Escolha *n* pontos de cruzamento aleatórios
- Divida ao longo desses pontos
- Construa o filho alternando entre os pais
- Generalização de 1 ponto (ainda com algum viés posicional)



Recombinação Uniforme

- Atribua “cara” a um dos pais e “coroa” ao outro
- Jogue uma moeda para cada gene do primeiro filho
- Faça uma cópia inversa do gene para o segundo filho
- A herança é independente da posição



Recombinação ou Mutação?

- Um longo debate: qual é o melhor / necessário / principal?

- Resposta (pelo menos, concordância bastante ampla):
  - depende do problema, mas
  - em geral, é bom ter ambos
  - ambos têm outra função

- EA somente mutação é possível, EA somente com recombinação não funcionaria

Recombinação ou Mutação? (cont.)

- Exploração: descobrir áreas promissoras no espaço de pesquisa, ou seja, obter informações sobre o problema (*exploration*)
- Intensificação: otimizando em uma área promissora, ou seja, usando informações (*exploitation*)
- Há cooperação e competição entre eles
  - O cruzamento é exploratório, ele dá um grande salto para uma área em algum lugar "entre" duas áreas (pai)
  - A mutação é exploradora, ela cria pequenos desvios aleatórios, permanecendo assim perto (na área) dos pais

Recombinação ou Mutação? (cont.)

- Apenas o crossover pode combinar informações de dois pais
- Apenas a mutação pode introduzir novas informações (alelos)
- O cruzamento não muda as frequências de alelos da população
  - Pense: somente 0 no primeiro bit na população
- Para atingir o ponto ideal, você geralmente precisa de uma mutação de “sorte”

Outras Representações

- Hoje em dia, é geralmente aceito que é melhor codificar variáveis numéricas diretamente como
  - Inteiros
  - Ponto flutuante

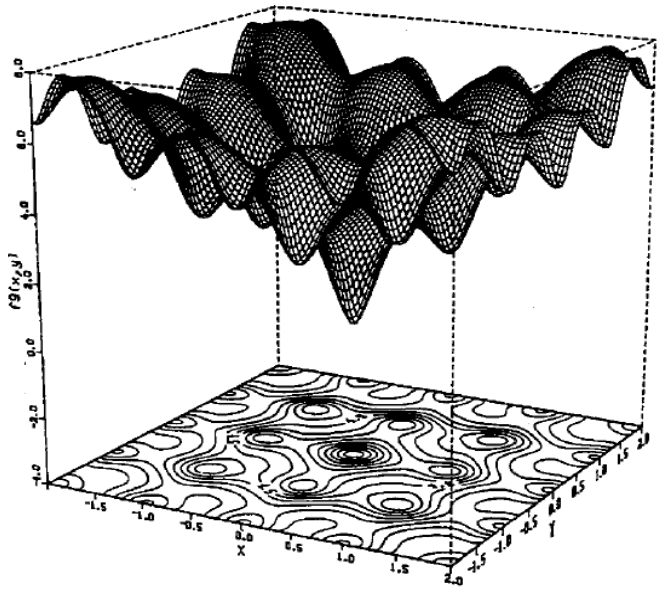
Representação Inteira

- Alguns problemas naturalmente têm variáveis inteiras, por exemplo parâmetros de processamento de imagem
- Outros assumem valores categóricos de um conjunto fixo, por exemplo, {azul, verde, amarelo, rosa}
- Operadores de recombinação permanecem iguais
- É necessário estender a mutação para trazer:
  - “Fluência”, ou seja, mais probabilidade de se mover para um valor semelhante
  - Escolha aleatória (especialmente variáveis categóricas)
  - Para problemas ordinais, é difícil saber o intervalo correto para fluência, então muitas vezes use dois operadores de mutação em conjunto

Representação Real

- Também conhecida como representação em ponto-flutuante
- Muitos problemas têm como entrada valores reais
  - Por exemplo otimização de parâmetro contínuos $f : \mathcal{R}^n \rightarrow \mathcal{R}$

$$f(\overline{x}) = -c_1 \cdot exp\left(-c_2 \cdot \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)$$
$$-exp\left(\frac{1}{n} \cdot \sum_{i=1}^{n} cos(c_3 \cdot x_i)\right) + c_1 + 1$$
$$c_1 = 20,\ c_2 = 0.2,\ c_3 = 2\pi$$



Representação Real (cont.)

- Podemos usar strings de bits e decodifica-las em valores reais

- Riscos:
  - Perda de precisão
  - Cromossomos muito grandes

- Alternativa: usar diretamente cromossomos de valores reais!
  - Seu cromossomo não representa mais um único valor, mas um conjunto de possíveis parâmetros

Mutações em ponto flutuante (1)

- Esquema geral de mutações de ponto flutuante

$$\bar{x} = \langle x_1, \ldots, x_l \rangle \rightarrow \bar{x}' = \langle x_1', \ldots, x_l' \rangle$$

$$x_i, x_i' \in [LB_i, UB_i]$$

- Mutação uniforme:

$$x_i' \text{ gerado uniformemente em } [LB_i, UB_i]$$

- Análogo à inversão de bits (binário) ou redefinição aleatória (inteiros)

Mutações em ponto flutuante (2)

- Mutações não uniformes:
  - Muitos métodos propostos, como intervalo de mudança variável com o tempo, etc.
  - A maioria dos esquemas são probabilísticos, mas geralmente fazem apenas uma pequena alteração no valor
- O método mais comum é adicionar o desvio aleatório a cada variável separadamente, por meio de uma distribuição Gaussiana N(0, σ)
  - Desvio padrão σ controla o total de mudanças
  - 2/3 de desvios estarão na faixa ($-\sigma$, $+\sigma$)

Operadores de recombinação para GAs reais

- Discretos:
  - Cada gene no filho *z* vem de um dos pais (*x,y*) com igual probabilidade: *zi = xi ou yi*
  - Pode ser n-pontos ou uniforme

Operadores de recombinação para GAs reais

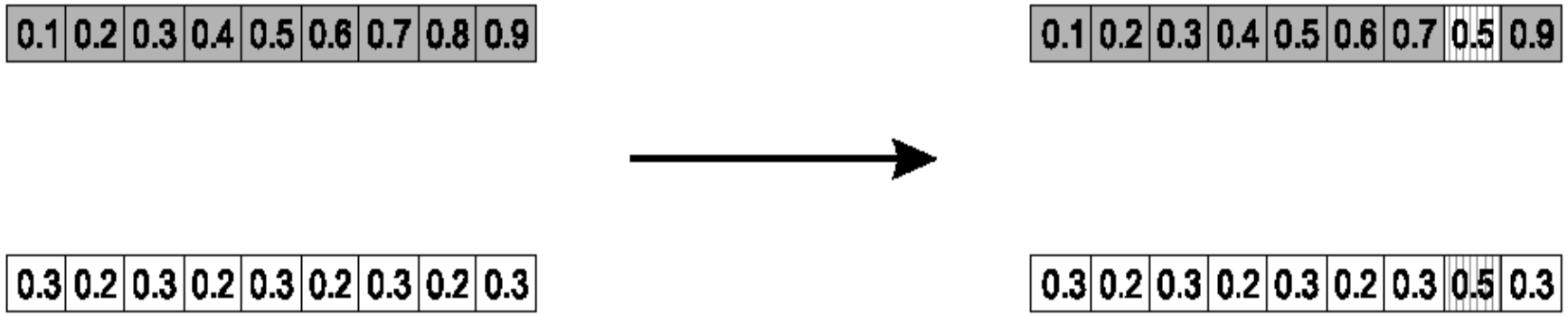
- Intermediários:
  - explora a ideia de criar filhos “entre” pais (daí o nome ***recombinação aritmética***)
  - $z_i = \alpha\, x_i + (1 - \alpha)\, y_i$, com α : $0 \leq \alpha \leq 1$.
  - O parâmetro α pode ser:
    - constante: cruzamento aritmético uniforme
    - variável (por exemplo, depende da idade da população)
    - escolhido aleatoriamente todas as vezes

Recombinação aritmética simples (1 ponto)

- Pais: $\langle x_1,\ldots,x_n \rangle$ e $\langle y_1,\ldots,y_n \rangle$
- Pegue um único gene (*k*) aletoriamente,
- $filho_1$ is:

$$\langle x_1,\ldots,x_{k-1},\alpha \cdot y_k + (1-\alpha) \cdot x_k,\ldots,x_n \rangle$$

- Exemplo: α = 0.5

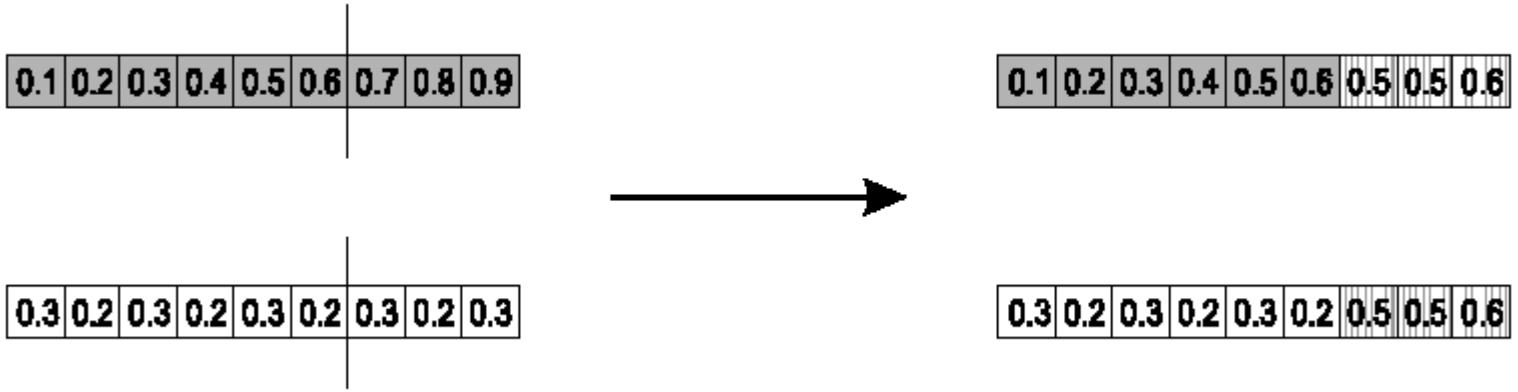


Recombinação aritmética simples (n pontos)

- Pais: $\langle x_1,\ldots,x_n \rangle$ e $\langle y_1,\ldots,y_n \rangle$
- Pegue um único gene (*k*) aletaoriamente,
- filho$_1$ is:

$$\langle x_1,\ldots,x_k,\alpha\cdot y_{k+1}+(1-\alpha)\cdot x_{k+1},\ldots,\alpha\cdot \mathrm{y_n}+(1-\alpha)\cdot x_n\rangle$$

- Exemplo: α = 0.5

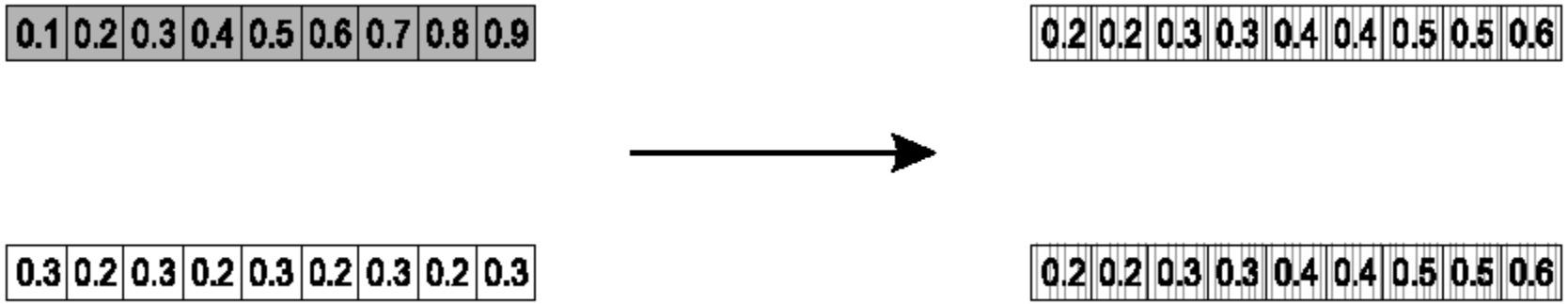


Recombinação aritmética completa (*whole*)

- Mais comumente utilizada
- Pais: $\langle x_1,\ldots,x_n \rangle$ e $\langle y_1,\ldots,y_n \rangle$
- filho$_1$ é:

$$a \cdot \bar{x} + (1-a) \cdot \bar{y}$$

- Exemplo: $\alpha = 0.5$



Modelos Populacionais

- SGA usa um modelo geracional (GGA):
  - cada indivíduo sobrevive por exatamente uma geração
  - todo o conjunto de pais é substituído pela prole
- Em oposição a esses, estão os modelos de estacionário (*Steady-State GA*):
  - uma prole é gerada por geração,
  - um membro da população substituído,
- Diferença entre gerações
  - a proporção da população substituída 1.0 para GGA, 1/pop_size para SSGA

Métodos de Seleção

- A seleção pode ocorrer em dois momentos:
  - Seleção da geração atual para participar do acasalamento (seleção dos pais)
  - Seleção de pais + filhos para ir para a próxima geração (seleção de sobreviventes)
- Operadores de seleção trabalham individualmente
  - ou seja, eles são independentes de representação
- Distinção entre seleção
  - operadores: definir probabilidades de seleção
  - algoritmos: definem como as probabilidades são implementadas

Exemplo de Implementação: SGA

- Número esperado de cópias de um indivíduo *i*

$$E(n_i) = \mu \bullet f(i)/\langle f \rangle$$

(μ = tamanho da pop, f(i) = fitness de i, ⟨f⟩ fitness médio na pop.)

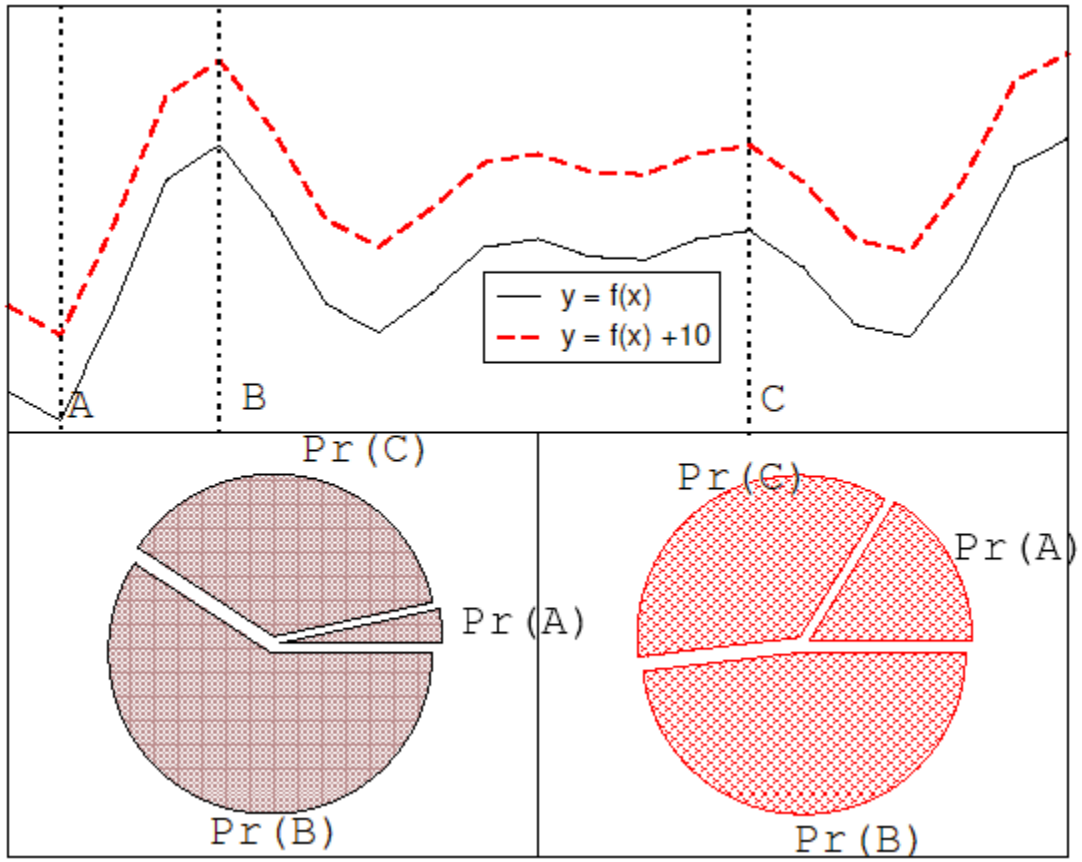
- Algoritmo da roda de roleta:
  - Dada uma distribuição de probabilidade, gire uma roda *n* vezes para fazer *n* seleções
  - Sem garantias sobre o valor real de n_i

Seleção Baseada no Fitness (FPS)

- Os problemas incluem
  - Um membro altamente apto pode assumir rapidamente se o resto da população for muito menos apto: *Convergência prematura*
  - No final das rodadas, quando os condicionamentos são semelhantes, perde a pressão de seleção

- Altamente suscetível à transposição de função

Transposição de Função



Seleção baseada em classificação (*rank*)

- Tenta resolver problemas de FPS baseando as probabilidades de seleção na aptidão relativa, em vez de absoluta
- Classifique a população de acordo com a aptidão e, em seguida, baseie as probabilidades de seleção na classificação em que o mais apto possui a classificação μ e a pior classificação 1
- Isso impõe uma sobrecarga no algoritmo (ordenação), mas geralmente é insignificante em comparação com o tempo de avaliação de aptidão

Seleção por Torneio

- Seleção por classificação ainda depende de estatísticas populacionais globais
  - Pode ser um gargalo, especialmente em máquinas paralelas
  - Depende da presença de função de aptidão externa que pode não existir

- Torneio:
  - Escolha *k* membros aleatoriamente e, em seguida, selecione o melhor deles
  - Repita para selecionar mais indivíduos

Seleção por Torneio (cont.)

- A probabilidade de selecionar *i* dependerá:
  - Da classificação *i*
  - Do tamanho da amostra *k*
    - *k* mais alto aumenta a pressão de seleção
  - Se os concorrentes são escolhidos com substituição
    - A escolha sem substituição aumenta a pressão de seleção
  - Se o competidor mais apto sempre vence (determinístico) ou isso acontece com a probabilidade *p*

Seleção dos Sobrevivente

- A maioria dos métodos acima são usados para a seleção dos pais
- Precisamos, também, selecionar os sobrevivente
- A seleção de sobreviventes pode ser dividida em duas abordagens:
  - Seleção baseada na idade
    - Exemplo: SGA
    - No SSGA, pode-se implementar uma “delete-random” (menos recomendada) ou uma first-in-first-out
  - Seleção baseada em fitness
    - Usando algum dos métodos anteriores

Dois casos especiais

- Elitismo:
  - Amplamente utilizado em ambos os modelos de população (GGA, SSGA)
  - Sempre mantenha pelo menos uma cópia da solução mais adequada até o momento
- Genitor (“delete-worst”):
  - Do algoritmo de estado estacionário original de Whitley
  - Usado com grandes populações