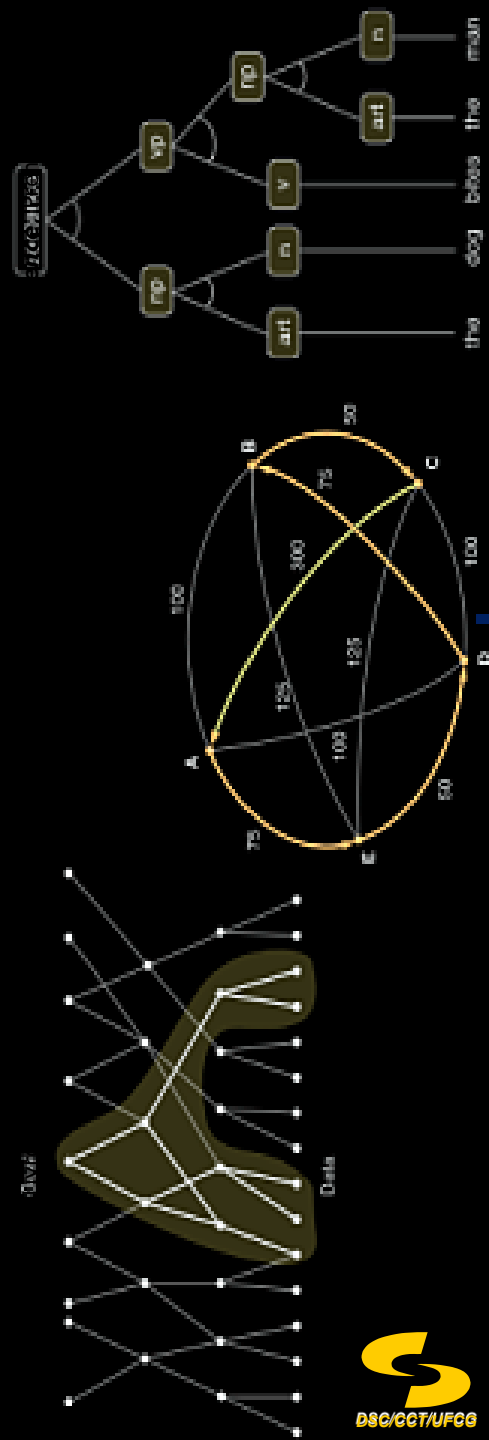


Universidade Federal de Campina Grande
Departamento de Sistemas e Computação
Curso de Pós-Graduação em Ciência da Computação

Inteligência Artificial

Resolução de Problemas (Parte VI)

Prof.^a Joseana Macêdo Fachine Régis de Araújo
joseana@computacao.ufcg.edu.br





Em Busca de Soluções

Tópico

- Algoritmos de Busca Local e Problemas de Otimização



Em Busca de Soluções

- **Modelos estudados:** interesse consistia em, a partir de um estado inicial, buscar sistematicamente uma seqüência de ações que levem ao estado-objetivo.
 - **Solução = caminho ao estado-objetivo**
- Em vários problemas, a própria descrição de estado contém toda informação relevante para a solução e o caminho ao estado-objetivo não interessa:
 - **Ex: problema das 8 rainhas, projeto de circuitos integrados, escalonamento, problemas de roteamento, de otimização de redes de telecomunicação, etc. ⇒ problemas de otimização**

Em Busca de Soluções

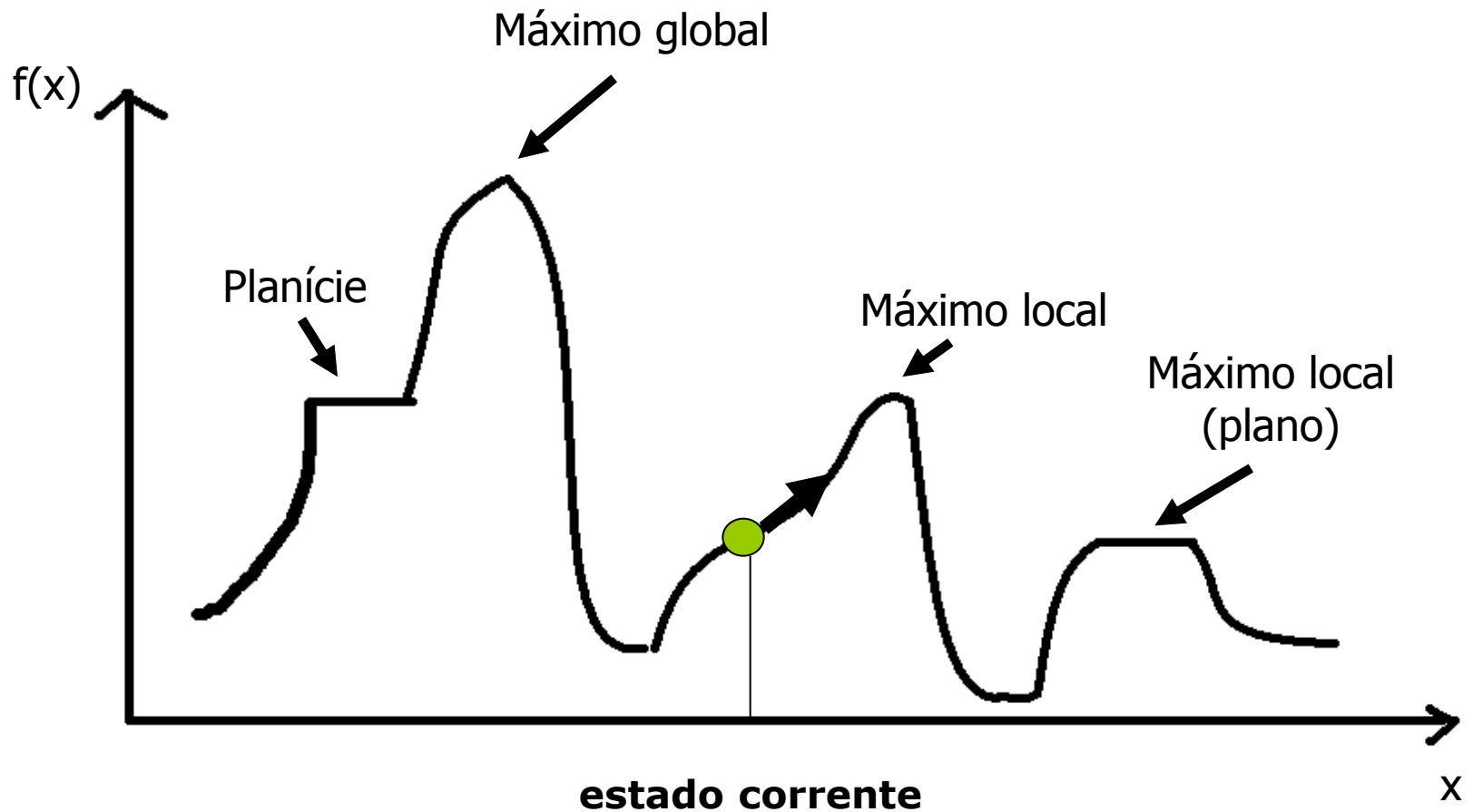
□ Busca Local (ou de melhoria iterativa)

- opera em um único estado e move-se para a vizinhança deste estado.
- Idéia: começar com o *estado inicial* (configuração completa, solução aceitável), e melhorá-lo iterativamente.

□ Vantagens:

- Usa pouquíssima memória;
- Frequentemente, pode encontrar soluções razoáveis em grandes ou infinitos espaços (contínuos) de estados para os quais os algoritmos sistemáticos são inadequados.

Em Busca de Soluções





Em Busca de Soluções

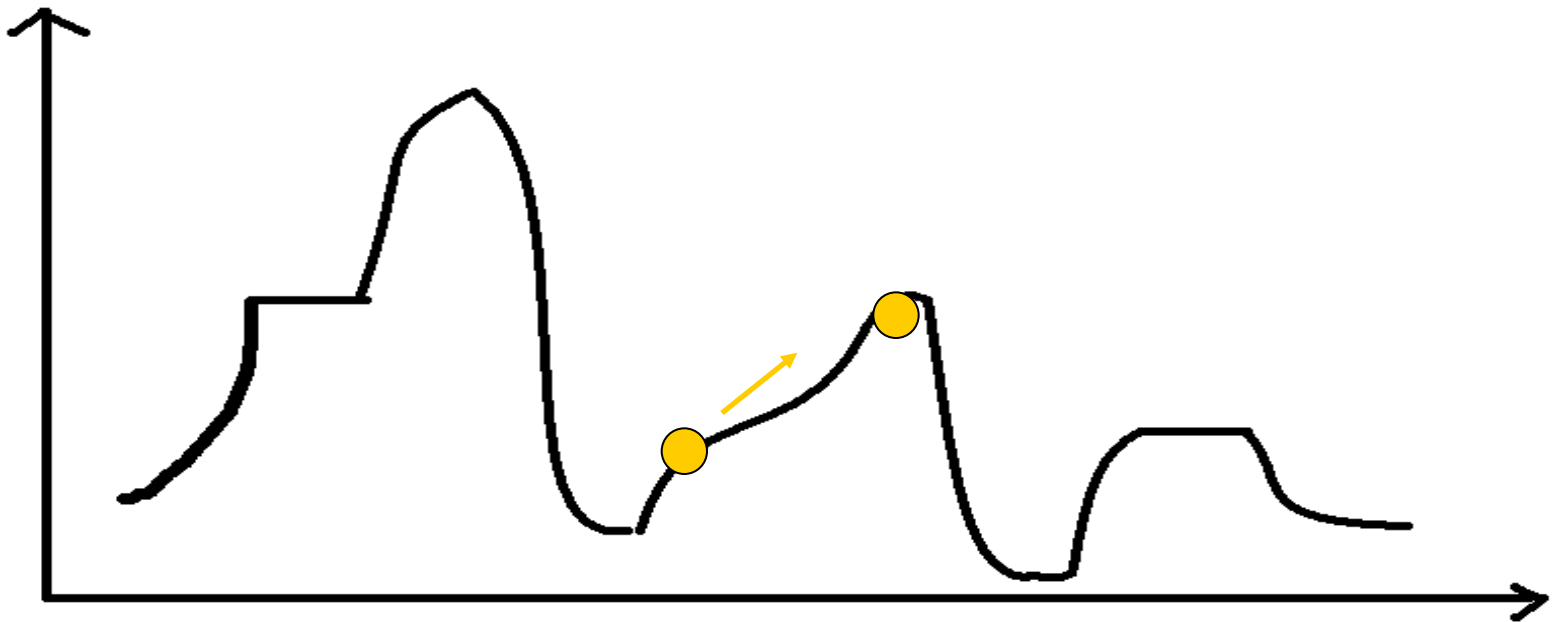
Tipos de Busca local

- ❑ **Hill-Climbing:** Subida pela Encosta mais Íngrime (ou Busca Local Gulosa)
 - só faz modificações que melhoram o estado atual.
- ❑ **Simulated Annealing:** Têmpera Simulada
 - pode fazer modificações que pioram o estado no momento, para possivelmente melhorá-lo no futuro.
- ❑ **Local Beam Search:** Busca em feixe local
 - Mantém k estados em vez de um único.
- ❑ **Algoritmos Genéticos (GA)**
 - É uma busca subida pela encosta, estocástica, na qual uma grande população de estados é mantida e novos estados são gerados por mutação e/ou cruzamento.

Em Busca de Soluções

Busca de Subida pela Encosta mais Íngreme

- O algoritmo procura o pico onde nenhum vizinho tem valor mais alto.





Em Busca de Soluções

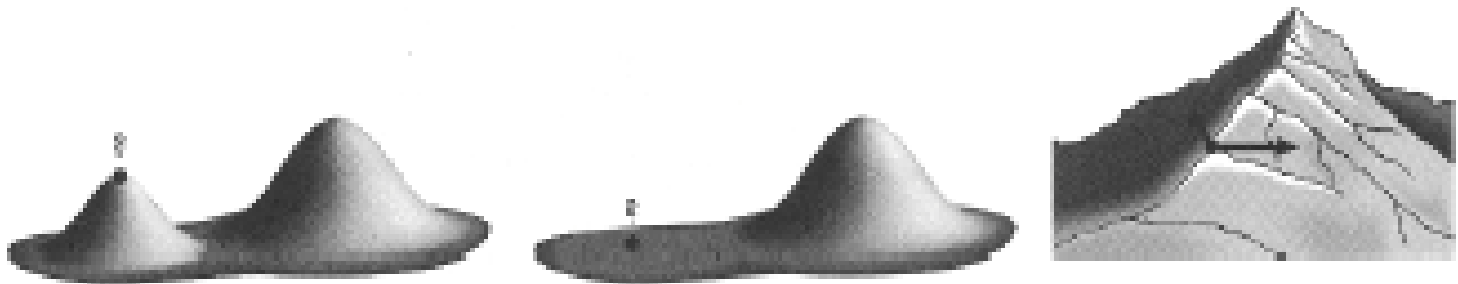
Busca de Subida pela Encosta mais Íngreme

- Não examina antecipadamente valores de estados além dos vizinhos imediatos do estado corrente.
- **O algoritmo *não* mantém uma árvore de busca:**
 - Guarda apenas o estado atual e sua avaliação
 - É simplesmente um ciclo que move o estado (solução) na direção crescente da função de avaliação (muda o estado para o melhor vizinho).
- É como tentar alcançar o cume do Monte Everest em meio a um nevoeiro denso durante uma crise de amnésia.

Em Busca de Soluções

Busca de Subida pela Encosta mais Íngreme

- **3 tipos de problemas que podem ser acarretados:**
 1. Máximos locais
 2. Planícies (platôs)
 3. Encostas e picos (“costelas”)





Em Busca de Soluções

Busca de Têmpera Simulada

- ❑ Este algoritmo é semelhante à Subida da Encosta, porém oferece meios para se escapar de máximos locais.
- ❑ T: “temperatura” - abaixa com o tempo de execução (com o tempo, fica igual ao Subida da Encosta).
- ❑ Assim, no início, movimentos “ruins” ocorrem com maior frequência. Apesar de aumentar o tempo de busca, essa estratégia consegue escapar melhor dos máximos locais.

Têmpera – Em metalurgia é o processo usado para temperar ou endurecer metais e vidro aquecendo-os a alta temperatura e depois esfriando-os gradualmente.



Em Busca de Soluções

Busca de Têmpera Simulada

- Usada inicialmente de forma extensiva para resolver problemas de layout de VLSI no começo dos anos 80.
- Foi amplamente aplicada ao escalonamento industrial e a outras tarefas de otimização em grande escala.

Em Busca de Soluções

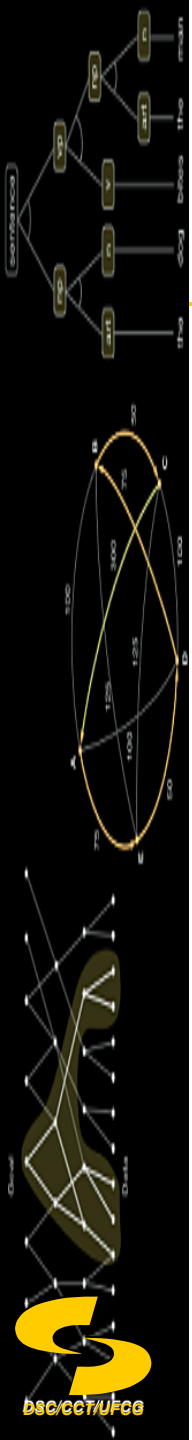
Busca em Feixe Local

- ❑ Começa com k estados gerados aleatoriamente.
- ❑ Em cada passo, são gerados todos os sucessores de todos os k estados.
- ❑ Se um dos sucessores for o objetivo, o algoritmo pára; caso contrário, escolhe os k melhores sucessores a partir da lista completa.
 - Isso NÃO corresponde à execução de k reinícios aleatórios em paralelo (*random start*)!
 - Somente k estados são considerados como estados atuais na busca.

Em Busca de Soluções

Algoritmos Genéticos

- ❑ Estados sucessores são gerados por meio da combinação de dois estados antecessores (pais), em vez de serem gerados pela modificação de um único estado.
- ❑ Os “sucessores” (descendentes) de um “estado” (organismo) ocupam a próxima geração de acordo com o seu “valor” (adaptação ou *fitness*).





Algoritmos Genéticos

- ❑ **Algoritmos genéticos (AG)** são um ramo dos algoritmos evolucionários.
- ❑ Podem ser definidos como uma técnica de busca baseada numa metáfora do processo biológico de evolução natural.
- ❑ Os algoritmos genéticos são técnicas heurísticas de otimização global.
- ❑ São algoritmos de busca baseados nos mecanismos de seleção natural e genética.



Algoritmos Genéticos

Características

- ❑ Podem trabalhar com uma codificação do conjunto de parâmetros ou com os próprios parâmetros.
- ❑ Trabalham com uma população e não com um único ponto.
- ❑ Utilizam informações de custo ou recompensa.
- ❑ Utilizam regras de transição estocásticas e não determinísticas.



Algoritmos Genéticos

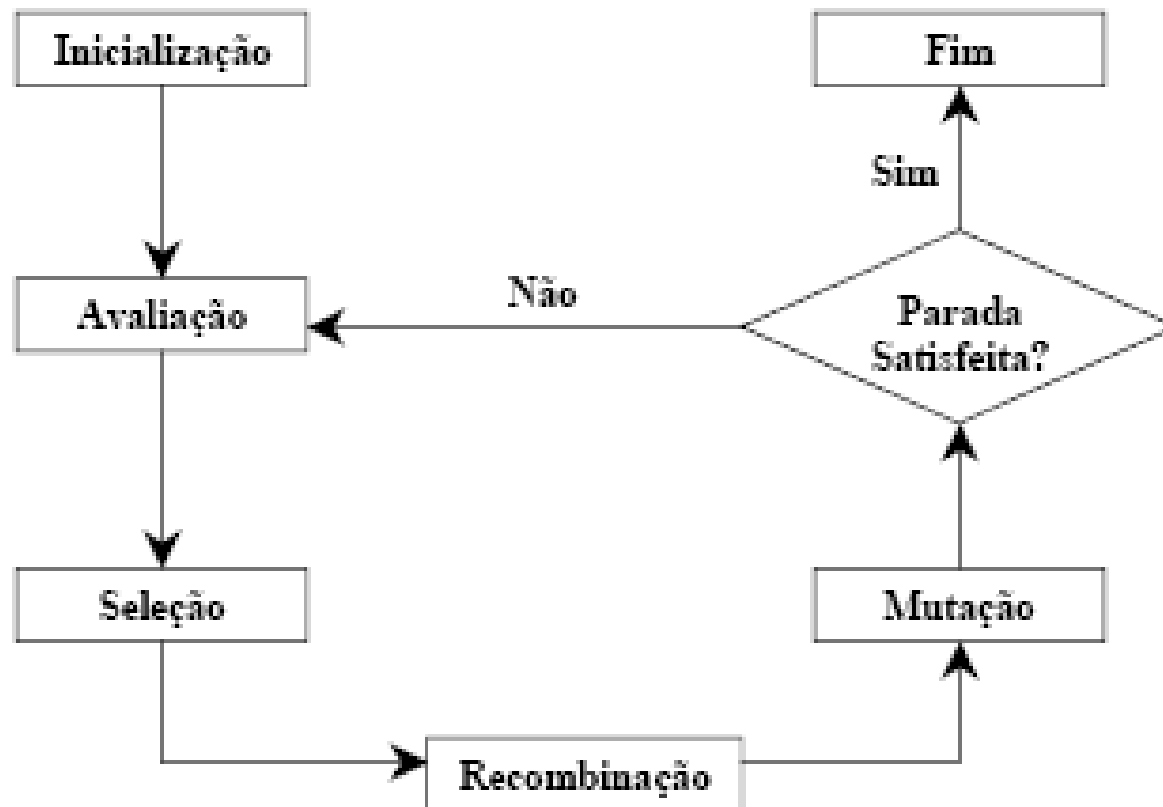
- ❑ Os parâmetros do problema são representados como genes em um cromossomo.
- ❑ Cada gene pode assumir valores específicos, sendo cada um destes valores chamados de alelo do gene.
- ❑ Um cromossomo representa um indivíduo, sendo composto por uma configuração de alelos.
- ❑ A posição de um gene num cromossomo corresponde a um locus gênico.

Algoritmos Genéticos

Evolução Natural	Problema Computacional
<i>Indivíduo</i>	Solução de um problema
<i>População</i>	Conjunto de soluções
<i>Cromossomo</i>	Representação de uma solução
<i>Gene</i>	Parte da representação de uma solução
<i>Crossover</i>	Operador de Busca
<i>Mutação</i>	Operador de Busca
<i>Seleção natural</i>	Reutilização de boas aproximações

Algoritmos Genéticos

□ Estrutura Básica





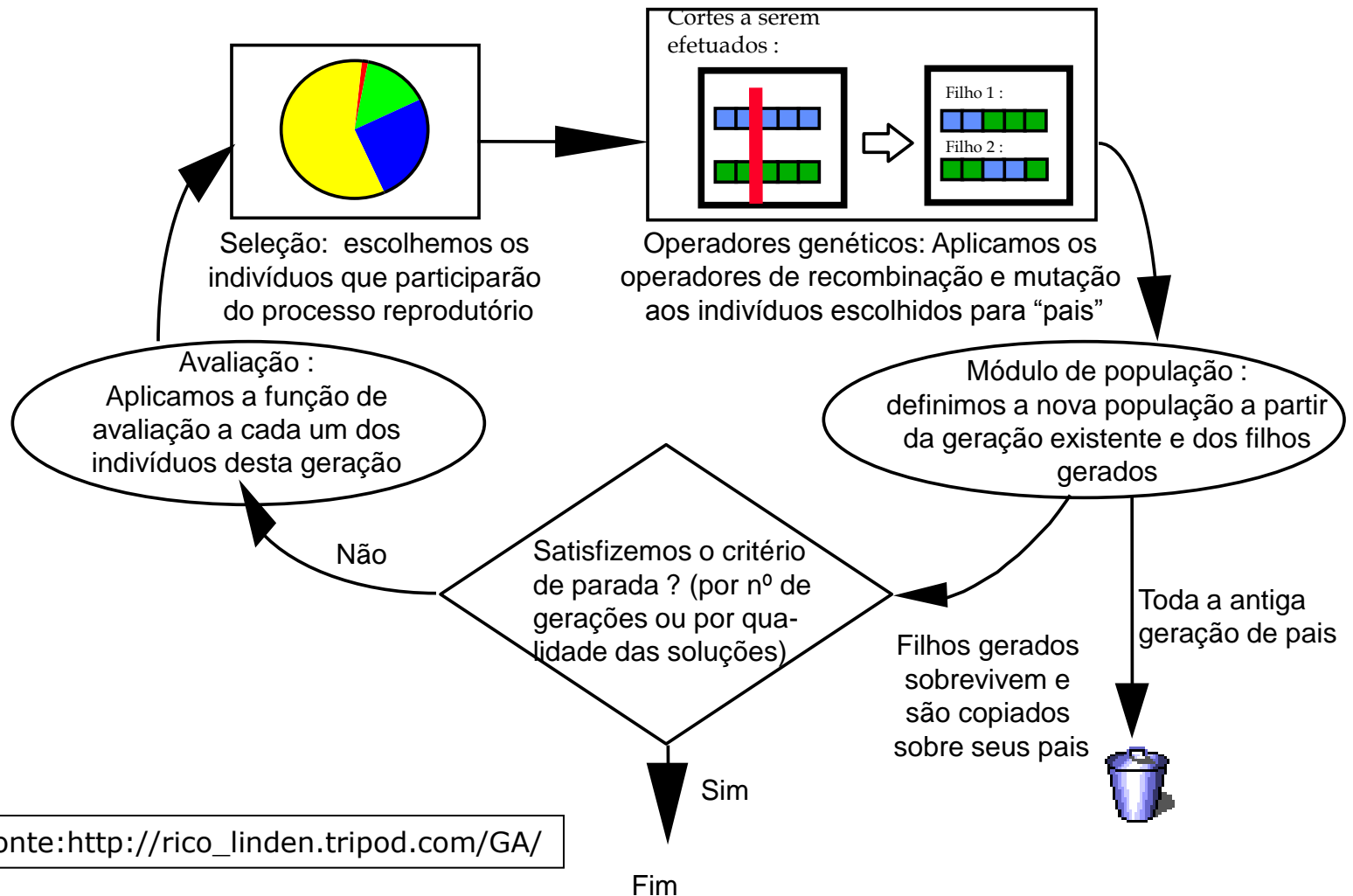
Em Busca de Soluções

Algoritmos Genéticos

□ Funcionamento:

- Inicia com um conjunto de k estados gerados aleatoriamente, chamado **população**; cada estado (ou **indivíduo**) é representado como uma cadeia sobre um alfabeto finito;
- Indivíduos são avaliados por uma **função de fitness** (função de avaliação em AG);
- Indivíduos selecionados geram novos indivíduos por meio de **cruzamentos e mutações**;
- Repete avaliação/seleção/cruzamento-mutação até que um indivíduo seja avaliado como adequado para solução.

Algoritmos Genéticos





Algoritmos Genéticos

Questões importantes

- ❑ Como criar cromossomos e qual tipo de codificação usar?
 - É a primeira pergunta que deve ser feita ao resolver um problema com AG.
 - A codificação dependerá fortemente do problema.

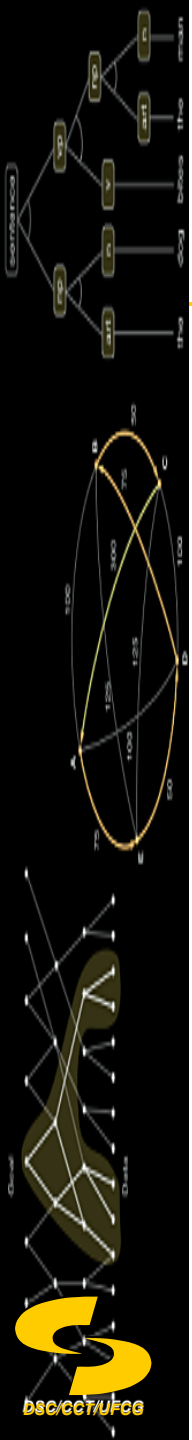
- ❑ Como escolher os pais para a realização do *crossover*?

- ❑ A geração de uma população a partir de duas soluções pode causar a perda da melhor solução. O que fazer?

Algoritmos Genéticos

Seleção

- O princípio básico do funcionamento dos AG é que um critério de seleção vai fazer com que, depois de muitas gerações, o conjunto inicial de indivíduos gere indivíduos mais aptos.





Algoritmos Genéticos

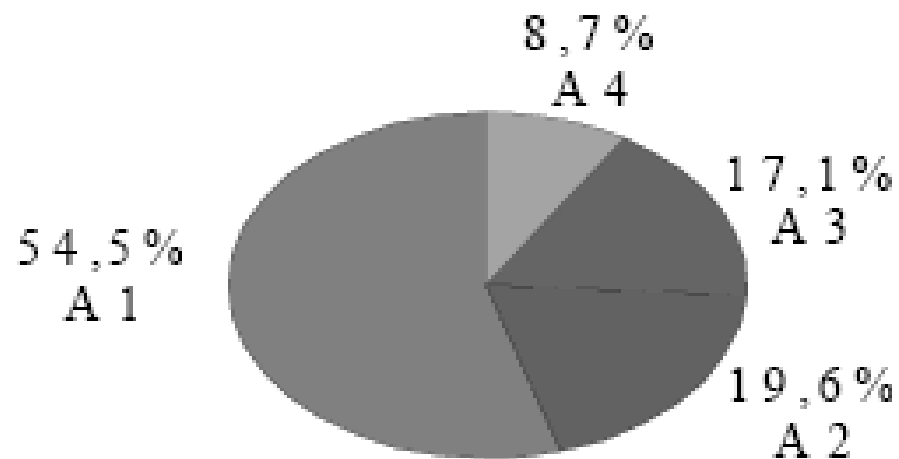
Seleção

- Uso de função objetivo como avaliação de aptidão.
- A aptidão pode ser vista como uma nota que mede o quão boa é a solução codificada por um indivíduo.
- Baseada, normalmente, no valor da função-objetivo, específica para cada problema
- **Métodos de Seleção**
 - Roleta
 - Torneio
 - Amostragem Universal Estocástica

Algoritmos Genéticos

Métodos de Seleção - Roleta

- Aptidão usada para definir fatia
- Valor aleatório para selecionar cromossomo
- Processo repetido até gerar os n indivíduos necessários



Algoritmos Genéticos

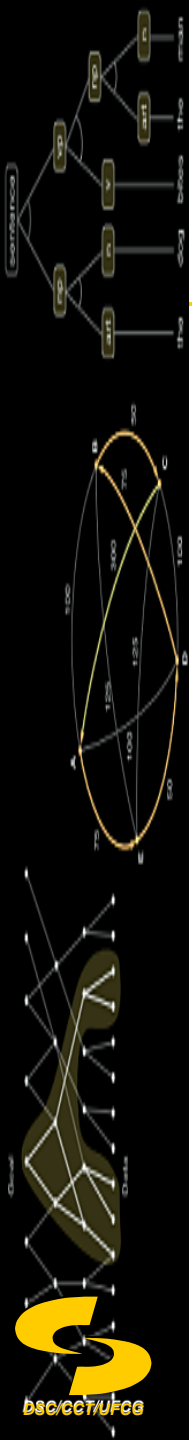
Métodos de Seleção - Torneio

- Escolha aleatória de m indivíduos
- Uso de função de aptidão para escolher o melhor
- Processo repetido até gerar os n indivíduos necessários

Algoritmos Genéticos

Métodos de Seleção - Amostragem

- ❑ Método da roleta com n agulhas igualmente espaçadas
- ❑ Roleta é girada uma única vez



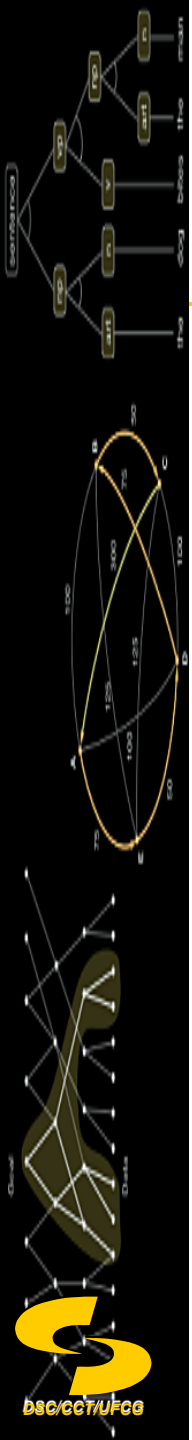
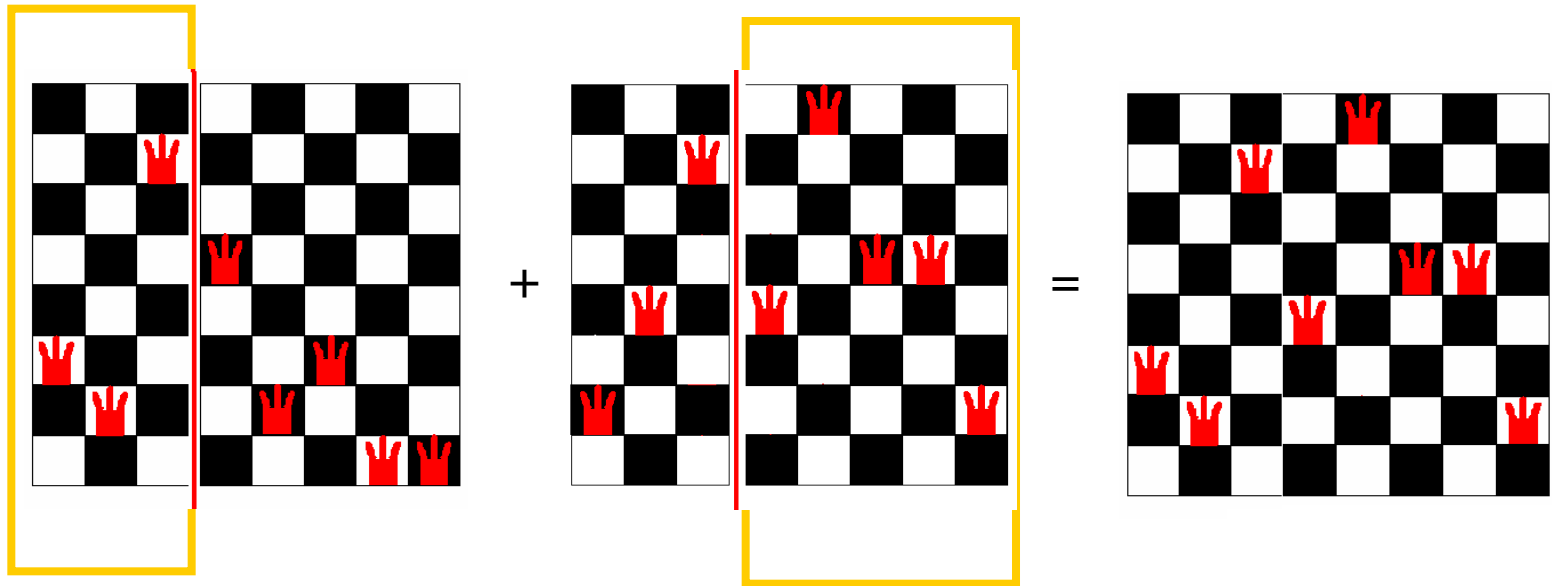
Em Busca de Soluções

Operadores Genéticos

- Recombinação (cruzamento)
 - *Merge* entre dois ou mais indivíduos (n:1):
 $r(i_1, i_2, \dots) = i_x$
 - A maneira com que é feito depende da representação dos indivíduos:
 - Binária
 - Inteira
 - Ponto flutuante
 - Objetos Compostos
 - Acrescenta indivíduos à população

Em Busca de Soluções

Recombinação (exemplo):





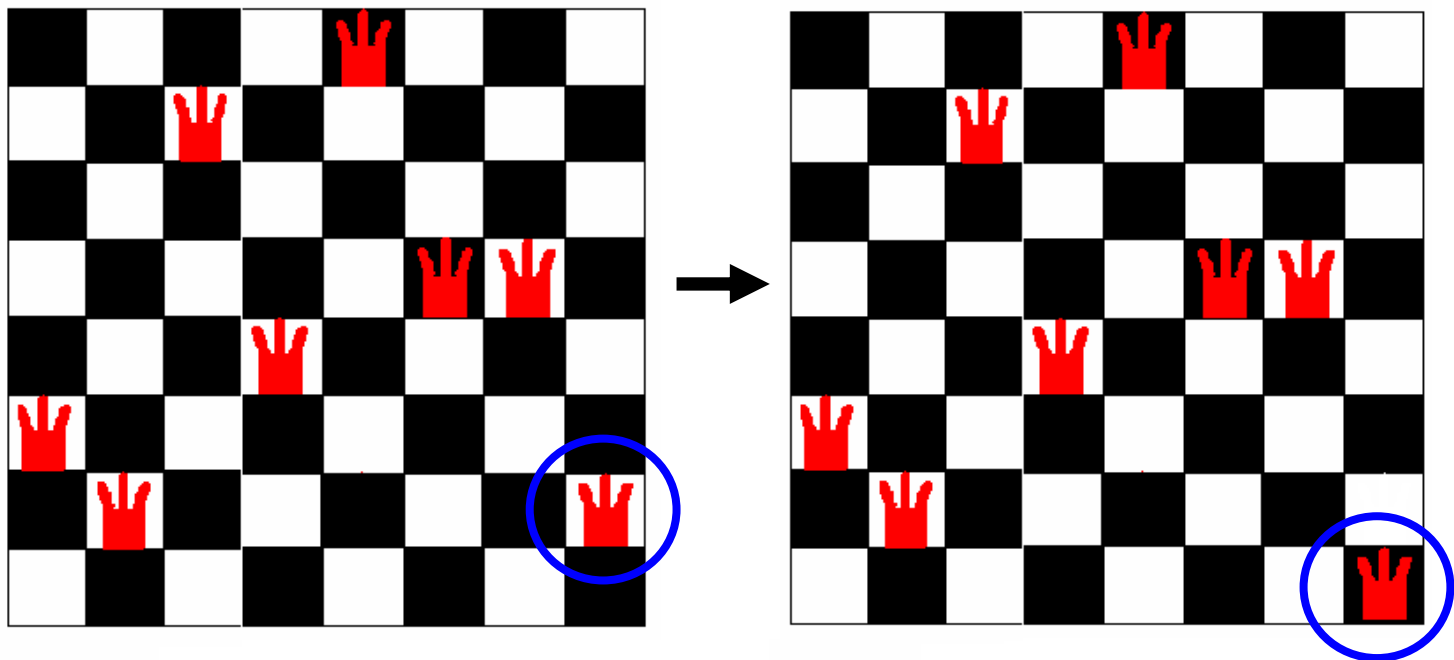
Em Busca de Soluções

Operadores Genéticos

- Mutaç o
 - Ocorre na rela o de 1:1
 - $m(i_1) = i_x$
 - A maneira com que   feito depende da representa o dos indiv duos:
 - Bin ria
 - Inteira
 - Ponto flutuante
 - Objetos Compostos
 - N o afeta o tamanho da popula o

Em Busca de Soluções

Mutação (exemplo):



Algoritmos Genéticos

Operadores Genéticos

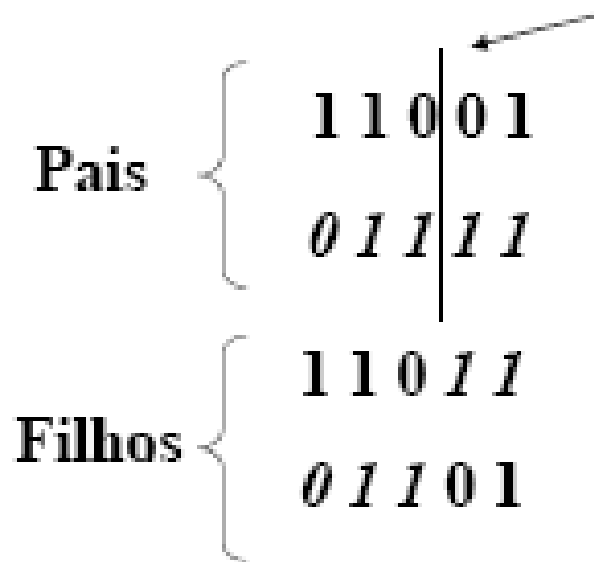
- Cruzamento
 - Cruzamento de pais para gerar dois filhos
- Taxa de *crossover*
- Tipos
 - Ponto Único
 - Dois Pontos
 - Multiponto
- Mutação

Algoritmos Genéticos

Operadores Genéticos



- Cruzamento – Ponto Único



Algoritmos Genéticos

Operadores Genéticos

- Cruzamento – Dois Pontos

<i>pai</i> ₁	0 1 0	0 1 1 0 0 0	1 0 1 0 1 1
<i>pai</i> ₂	0 0 1	0 0 1 1 1 0	0 0 1 1 0 1
<i>filho</i> ₁	0 1 0	0 0 1 1 1 0	1 0 1 0 1 1
<i>filho</i> ₂	0 0 1	0 1 1 0 0 0	0 0 1 1 0 1

Algoritmos Genéticos

Operadores Genéticos

- Cruzamento – Multipontos

<i>pai₁</i>	1 0 1	0 1 0 0 1 0	0 1 0 1 0	0 1	0 0 1
<i>pai₂</i>	0 0 1	0 0 1 1 1 0	0 0 1 1 0	1 1	1 0 0
<i>filho₁</i>	1 0 1	0 0 1 1 1 0	0 1 0 1 0	1 1	0 0 1
<i>filho₂</i>	0 0 1	0 1 0 0 1 0	0 0 1 1 0	0 1	1 0 0

Algoritmos Genéticos

Operadores Genéticos

- Mutaç o
 - Mudana aleat ria de alelo
 - Taxa de mutao
 - Significativamente inferior a de cruzamento

Antes da
mutao 0 1 1 0 1

Depois 0 0 1 0 1

Algoritmos Genéticos

É importante lembrar: Codificação Binária

- ❑ É a mais comum devido a sua simplicidade
- ❑ Cada cromossomo é uma *string* de bits – 0 ou 1
 - Crom: A = 1 0 1 1 0 0 1 0 1 1
 - Crom: B = 1 1 1 1 1 1 0 0 0 0
- ❑ **Exemplo de uso:** problema da mochila
- ❑ **O problema:** É dada uma lista de coisas com preços e tamanhos. É fornecido o valor da capacidade da mochila. Escolha as coisas de forma a maximizar o valor daquilo que cabe dentro da mochila, sem ultrapassar sua capacidade.
- ❑ **Codificação:** Cada bit é usado para dizer se a coisa correspondente está ou não na mochila.



Algoritmos Genéticos

É importante lembrar: Codificação por permutação

- Mais usado em problemas de ordenação
- Cada cromossomo é uma *string* de números que representa uma posição numa seqüência
 - Crom A: 1 5 3 2 6 4 7 9 8
 - Crom B: 8 5 6 7 2 3 1 4 9
- **Exemplo de uso:** problema do caixeiro viajante
- **O problema:** São dadas cidades e as distâncias entre elas. O caixeiro viajante tem que visitar todas elas, sem viajar mais do que o necessário. A solução do problema consiste em encontrar a seqüência de cidades em que as viagens devem ser feitas de forma que a distância percorrida seja a mínima possível.
- **Codificação:** os cromossomos descrevem a ordem em que o caixeiro irá visitar as cidades.



Algoritmos Genéticos

É importante lembrar: Codificação por valor

- ❑ Usado em problemas nos quais valores mais complicados são necessários
- ❑ Cada cromossomo é uma seqüência de valores
 - Crom A: 1.2324 5.3243 0.4556 2.3293 2.4545
 - Crom B: ABDJEIFJDHDIERJFDLDFLFEGT
 - Crom C: (back), (back), (right), (forward), (left)
- ❑ **Exemplo de uso:** dada uma estrutura, encontrar pesos para uma rede neural.
- ❑ **O problema:** É dada uma rede neural com arquitetura definida. Encontre os pesos entre os neurônios da rede de forma a obter a resposta desejada da rede.
- ❑ **Codificação:** Valores reais num cromossomo representam pesos em uma rede neural.



Algoritmos Genéticos

É importante lembrar: Função de Avaliação

- ❑ É a maneira utilizada pelos AG para determinar a qualidade de um indivíduo como solução do problema em questão.
- ❑ É uma nota dada ao indivíduo na resolução do problema.
- ❑ Dada a generalidade dos AG, a função de avaliação, em muitos casos, é a única ligação verdadeira do programa com o problema real.

Algoritmos Genéticos

Parâmetros Genéticos

- Tamanho da população
- Taxa de cruzamento
- Taxa de mutação
- Intervalo de geração
 - Percentual de renovação da população

Algoritmos Genéticos

Parâmetros Genéticos

- Critério de parada
 - Número de gerações
 - Convergência da função de aptidão na população
 - Não melhoria da aptidão do melhor indivíduo após um número de gerações

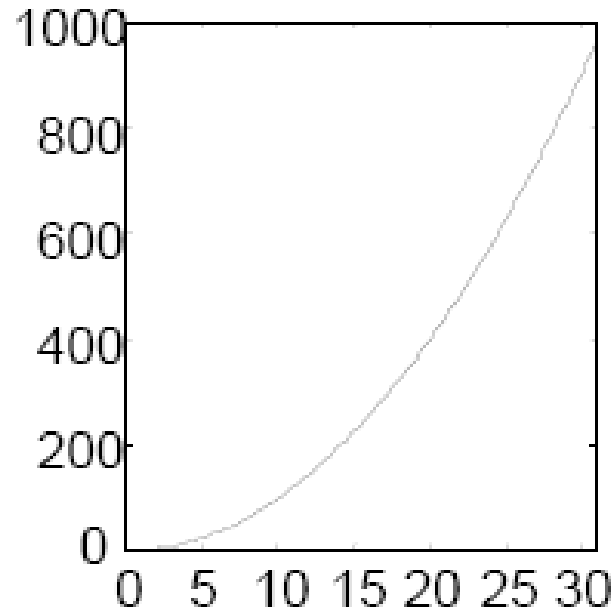
Algoritmos Genéticos

- Exemplo: Use um AG para encontrar o ponto máximo da função $f(x) = x^2$

- com x sujeito às seguintes restrições:

$$0 \leq x \leq 31$$

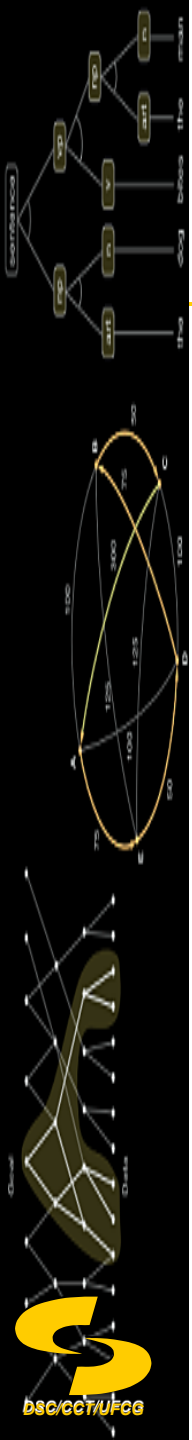
x é inteiro



Algoritmos Genéticos

- Cromossomos binários com 5 bits
 - $0 = 00000$
 - $31 = 11111$

- Aptidão
 - Neste problema a aptidão pode ser a própria função objetivo
 - Exemplo: $\text{Aptidão}(00011) = f(3) = 9$



Algoritmos Genéticos

Pop.
inicial

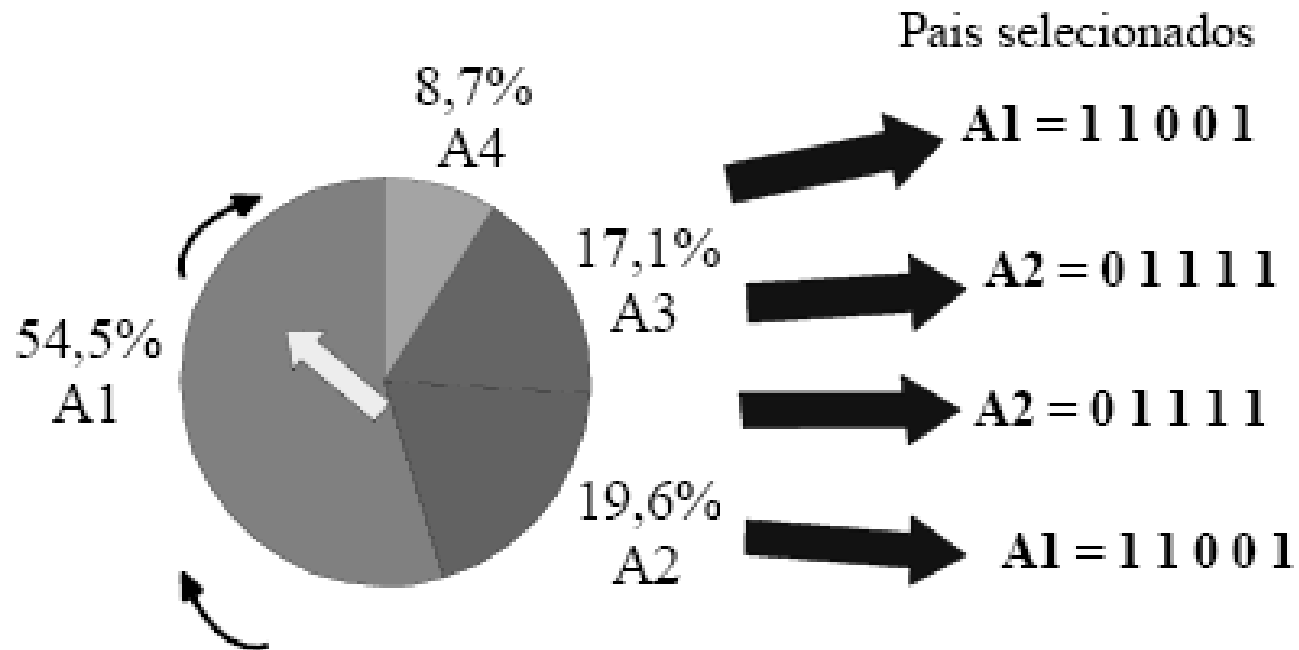
cromossomos	χ	$f(x)$	Prob. de seleção
$A_1 = 1\ 1\ 0\ 0\ 1$	25	625	54,5%
$A_2 = 0\ 1\ 1\ 1\ 1$	15	225	19,6%
$A_3 = 0\ 1\ 1\ 1\ 0$	14	196	17,1%
$A_4 = 0\ 1\ 0\ 1\ 0$	10	100	8,7%

Probabilidade de seleção
Proporcional à aptidão

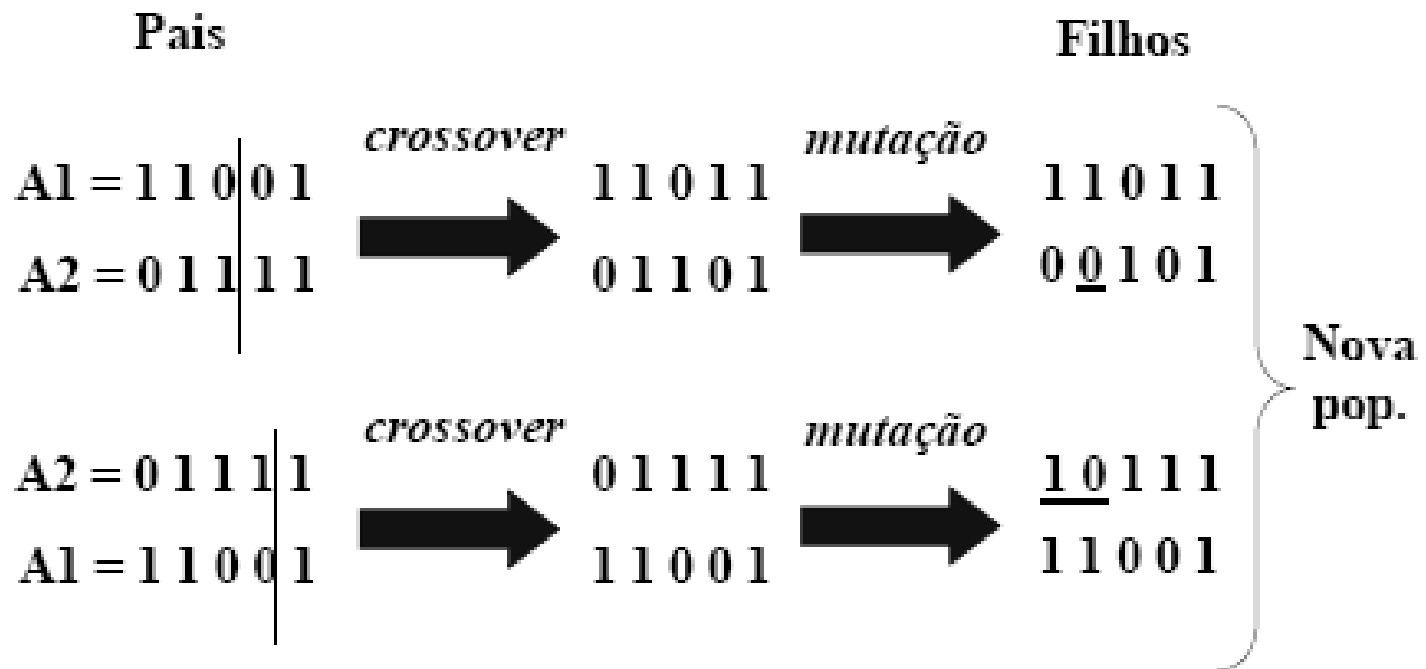
$$p_i = \frac{f(x_i)}{\sum_{k=1}^N f(x_k)}$$

A população inicial é aleatória (mas quando possível, o conhecimento da aplicação pode ser utilizado para definir população inicial).

Algoritmos Genéticos



Algoritmos Genéticos



Algoritmos Genéticos

cromossomos	x	$f(x)$	prob. de seleção	
1	1 1 0 1 1	27	729	29,1%
2	1 1 0 0 1	25	625	24,9%
3	1 1 0 0 1	25	625	24,9%
4	1 0 1 1 1	23	529	21,1%

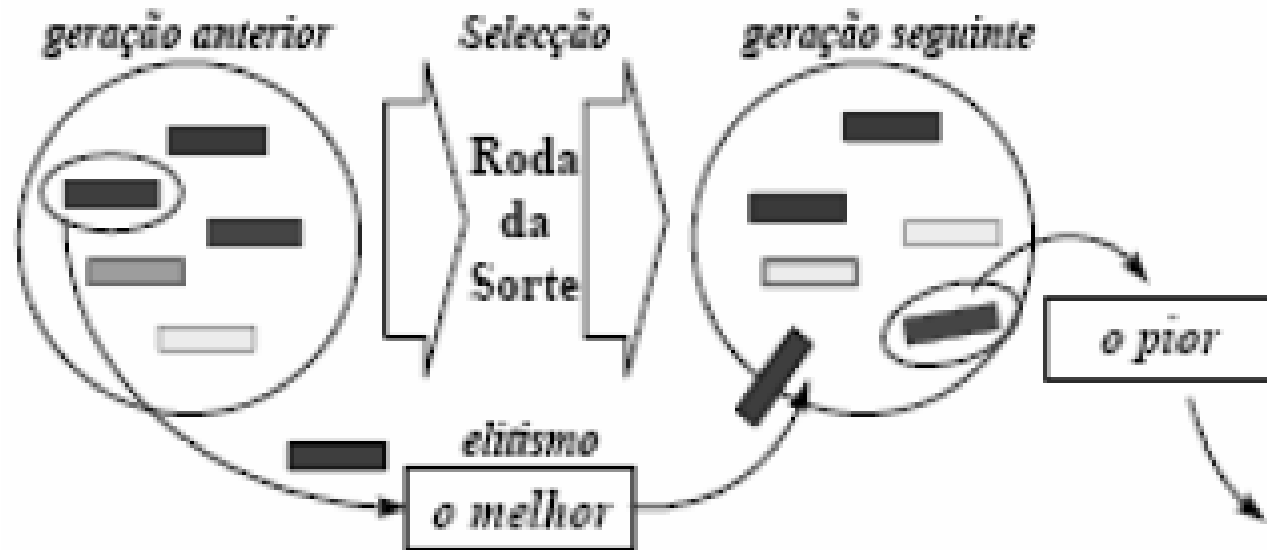


Algoritmos Genéticos

Elitismo

- ❑ Um elemento que tenha maior aptidão do que outro tem também maior probabilidade de ser selecionado.
- ❑ Nada impede que seja selecionado o pior, perdendo-se assim talvez o melhor elemento da população, que poderia levar a uma convergência mais rápida.
- ❑ Para tentar minimizar este possível problema, elitismo pode ser adicionado à seleção.
- ❑ Percentual de indivíduos com melhor aptidão é mantido na nova geração.

Algoritmos Genéticos





Algoritmos Genéticos

Aspectos Práticos

- A implementação prática de um AG requer atenção para várias questões:
 1. Escolha da Função de Avaliação/Aptidão
 2. Problemas de convergência
 3. Escolha da Técnica de Seleção
 4. Lacuna entre gerações (*generation gap*)

Algoritmos Genéticos

- ❑ AG são técnicas probabilísticas, e não técnicas determinísticas.
- ❑ Iniciando um AG com a mesma população inicial e o mesmo conjunto de parâmetros é possível encontrar soluções diferentes a cada vez que se executa o programa.

Em Busca de Soluções

Algoritmos Genéticos

- A recombinação eleva a qualidade da busca, pois oferece uma maior diversidade para a população de indivíduos.

Em Busca de Soluções

- ❑ GA não são métodos de "hill climbing", logo não ficarão estagnados simplesmente pelo fato de terem encontrado um máximo local.
- ❑ Eles se parecem com a evolução natural, que só porque encontrou um indivíduo que é instantaneamente o melhor de um certo grupo não deixa de “procurar” outros indivíduos ainda melhores.
- ❑ Na evolução natural isto também decorre proveniente de circunstâncias que mudam de um momento para outro.



Algoritmos Genéticos

Demonstrações

- ❑ <http://www.blprnt.com/smartrockets/>
- ❑ <http://math.hws.edu/eck/jsdemo/jsGeneticAlgorithm.html>
- ❑ http://rednuht.org/genetic_walkers/

Implementação em Java

- ❑ http://www.univale.com.br/unisite/mundo-j/artigos/59_Algoritmosgeneticos.pdf