

Inteligência Artificial

ACH2016

Aula 7: Algoritmos Genéticos

Profa. Karina Valdivia Delgado
EACH-USP

Slides baseados em:

Slides do Prof. Edirlei Soares de Lima

Slides da Profa. Sarajane Marques Peres

RUSSEL, S.; NORVIG, P. Artificial Intelligence: A modern approach. Third Edition, 2010. Capítulo 4.

Métodos de busca

- **Busca sem informação ou cega ou exaustiva ou força bruta ou sistemática :**
 - Não sabe qual o melhor nó da fronteira a ser expandido. Apenas distingue o estado objetivo dos não objetivos.
- **Busca informada ou heurística:**
 - Estima qual o melhor nó da fronteira a ser expandido com base em funções heurísticas. Sabem se um estado não objetivo é “mais promissor”.
- **Busca local:**
 - Operam em um único estado e movem-se para a vizinhança deste estado.

Algoritmos genéticos

- Método de **busca local**.
- Abordagem para lidar com **espaços de busca muito grandes**.
 - procura por soluções que talvez não fossem encontradas por uma busca convencional mesmo que ela durasse centenas de anos.
- Baseado na **evolução dos seres biológicos**.

Teoria da Evolução

- Na natureza todos os indivíduos dentro de um ecossistema **competem** entre si por recursos limitados (comida, água...)
- Os **indivíduos mais fracos** de uma mesma espécie tendem a não se proliferarem.
- A descendência reduzida faz com que a probabilidade de ter seus **genes propagados** ao longo de sucessivas gerações seja menor.
- A combinação entre os genes dos indivíduos que sobrevivem pode produzir um novo **indivíduo muito melhor adaptado** às características de seu meio ambiente ao combinar possivelmente características positivas de cada um dos seus pais.

Relembrando

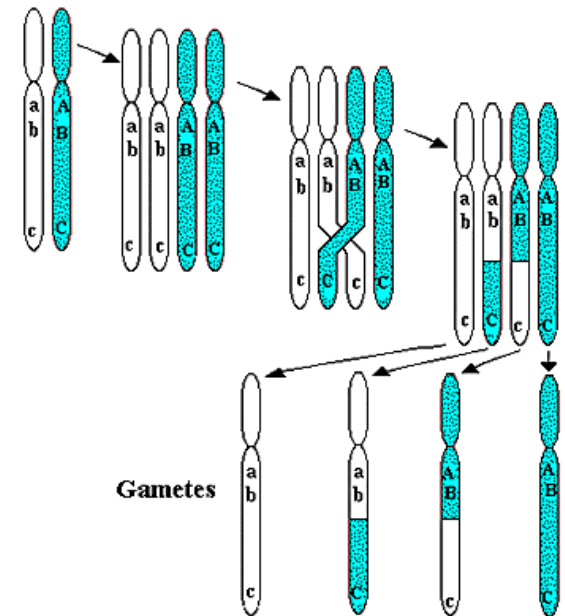
- Todo indivíduo biológico é formado por uma ou mais **células**.
- Dentro de cada célula existe um conjunto de **cromossomos**.
 - Os seres humanos têm 23 pares de cromossomos por célula.
- Um cromossomo possui vários **genes** (blocos de sequências de DNA, as quais codificam toda a informação necessária para o desenvolvimento e funcionamento de organismos vivos).
- A qualidade de um indivíduo (**fitness**) é medida pelo seu sucesso (sobrevivência)

Relembrando

- Na natureza existem dois tipos de **reprodução**:
 - **Assexuada**: típica de organismos inferiores, como bactérias.
 - **Sexuada**: exige a presença de dois organismos que trocam material genético.
- **Reprodução assexuada** é base para o algoritmo de busca local **Beam Search**.
- **Reprodução sexuada** é a base dos **algoritmos genéticos**.

Relembrando

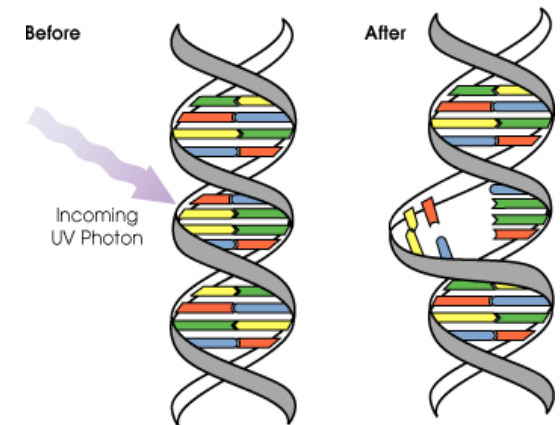
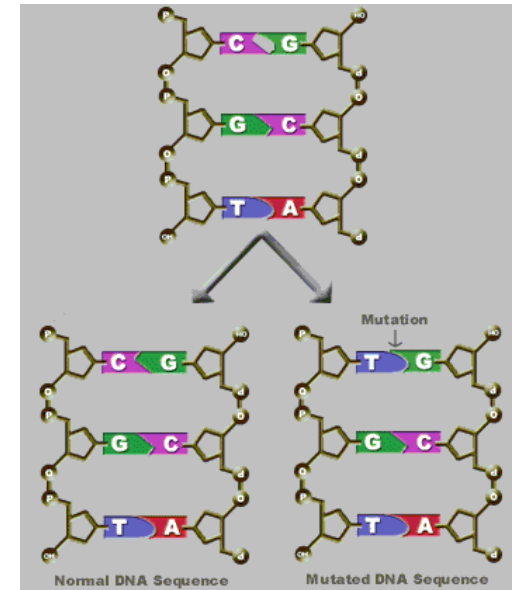
- Na **reprodução sexuada** ocorre a formação de um **novo indivíduo** através da combinação de duas células.
- Na formação dos novos indivíduos, ocorre o processo de **recombinação genética** (**crossing-over**).



Crossing-over and recombination during meiosis

Relembrando

- O processo de replicação do DNA é extremamente complexo.
- Pequenos **erros** podem ocorrer ao longo do tempo, gerando **mutações** dentro do código genético.
- Estas mutações podem ser boas, ruins ou neutras.
- Alguns **fatores externos** (ex. radiação ultravioleta) também podem causar pequenas interrupções no código genético.



Relembrando

- Indivíduos com uma melhor adequação ao meio ambiente (**melhor fitness**) se reproduzem mais.
- Dessa forma têm mais chances de passar seus genes para a **próxima geração**.
- Entretanto, graças aos **operadores genéticos** (**recombinação e mutação**) os cromossomos dos filhos não são exatamente iguais aos dos pais.
- Assim, eles podem **evoluir** e se **adaptar** cada vez mais ao meio ambiente.

Computação evolutiva

- A computação evolutiva se inspira em princípios da teoria da evolução e seleção natural e utiliza modelos destes processos naturais para a solução de problemas.
- Principais ramos:
 - Algoritmos Genéticos
 - Programação Genética
 - Evolução Diferencial

Computação evolutiva

- São aplicados **operadores genéticos**, como:
recombinação
mutação
- Cada indivíduo recebe uma **avaliação (fitness)** que é uma quantificação da sua **qualidade** como solução do problema.
- Baseados nesta avaliação são aplicados operadores genéticos de forma a simular a **sobrevivência do mais apto**.
- **busca** (dentro da atual população) aquelas **soluções** que possuem as **melhores características** e tenta combiná-las de forma a gerar soluções ainda melhores.
- O processo é repetido até que tenha se passado tempo suficiente ou que tenhamos obtido uma solução satisfatória para nosso problema.

Computação evolutiva: esqueleto

- Gere aleatoriamente uma população de soluções candidatas.
- Enquanto o critério de parada não for satisfeito, faça:
 1. recombine alguns indivíduos da população
 2. mute alguns indivíduos da população
 3. avalie todo o repertório de soluções candidatas
 4. selecione segundo algum critério quais soluções irão para a próxima geração

Computação evolutiva

- São extremamente dependentes de **fatores estocásticos** (probabilísticos), tanto na fase de **inicialização** da população quanto na fase de **evolução**.
- Isto faz com que os seus **resultados sejam raramente reprodutíveis de maneira perfeita**.
- Não garantem a obtenção do melhor resultado possível em todas as suas execuções.

Computação evolutiva

Sempre dê prioridade aos algoritmos exatos.
Porém, em diversas aplicações, ainda não foram concebidos:

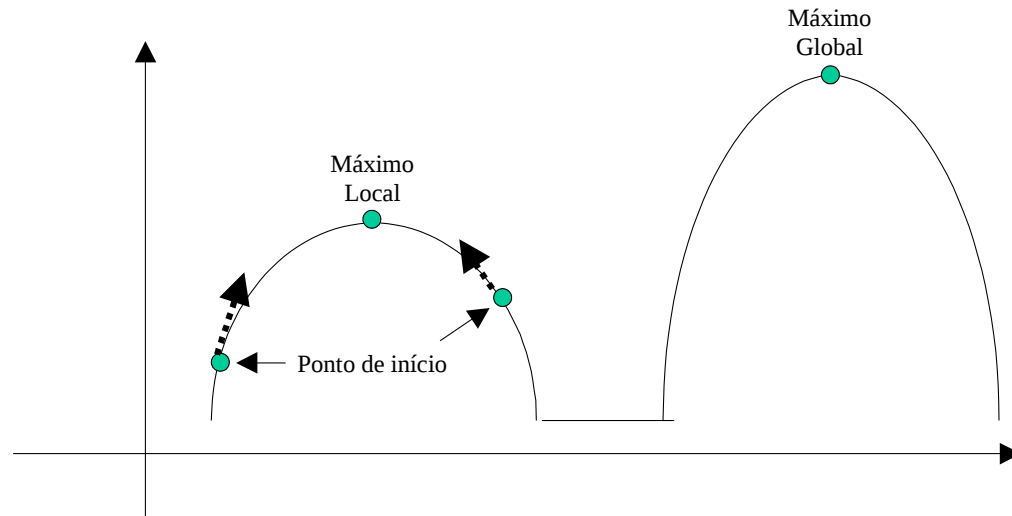
- algoritmos exatos de solução ou
- mesmo heurísticas específicas.

Ou pode ser que os algoritmos exatos existem, porém são extremamente lentos ou incapazes de obter uma solução.

Nestas situações, os algoritmos evolutivos se tornam candidatos interessantes.

• Algoritmos genéticos

- São algoritmos evolucionários. Logo, são uma metáfora para a evolução natural.
- São técnicas heurísticas de **otimização global**. Com isto, raramente ficam presos em **máximos locais**.



Algoritmos genéticos

- As populações de indivíduos são criadas e submetidas a operadores genéticos:
 - Seleção.
 - Recombinação.
 - Mutação.
- É utilizada uma caracterização da **qualidade de cada indivíduo** como solução do problema em questão chamada de avaliação do indivíduo (**fitness**).
- É gerado um processo de **evolução natural** destes indivíduos.

Algoritmos genéticos: processo

- Gere aleatoriamente uma população de soluções candidatas (inicialize a população).
- Avalie cada indivíduo na população
- Enquanto o critério de parada não for satisfeito (ex: o tempo acabou ou o melhor indivíduo satisfaz os requerimentos da solução do problema), faça:
 1. Selecione os melhores pais para gerar novos indivíduos.
 2. Aplique os operadores de recombinação e mutação de forma a gerar os indivíduos da nova geração.
 3. Atualize a população, segundo algum critério (quais soluções irão para a próxima geração)
 4. Avalie cada indivíduo na população.

Algoritmos genéticos: processo

Seja $S(t)$ a população de cromossomos na geração t .

$t \leftarrow 0$

inicializar $S(t)$

avaliar $S(t)$

enquanto o critério de parada não for satisfeito **faça**

$t \leftarrow t + 1$

 selecionar $S(t)$ a partir de $S(t-1)$

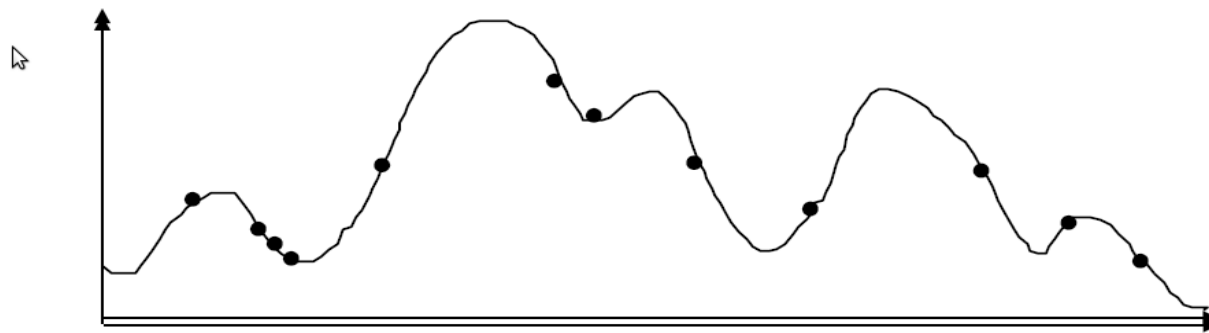
 aplicar *crossover* sobre $S(t)$

 aplicar mutação sobre $S(t)$

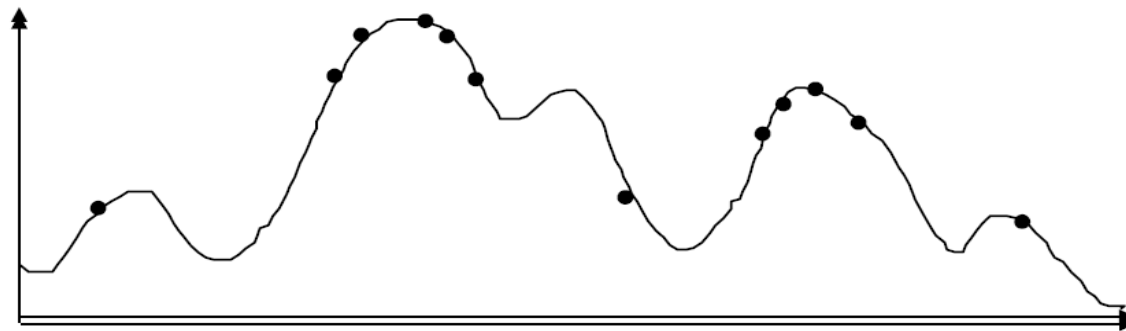
 avaliar $S(t)$

fim enquanto

Algoritmos genéticos



Distribuição dos indivíduos na Geração 0



Distribuição dos indivíduos na Geração N

Algoritmos genéticos

Definição de um problema em algoritmos genéticos:

- É necessário definir uma maneira de **codificar os indivíduos**.
- Definir os **operadores genéticos** que serão utilizados:
 - Recombinação
 - Mutação
- Definir uma **função de avaliação** para medir a capacidade de sobrevivência de cada indivíduo.
- Definir um método de **seleção dos pais**
- Definir um método de **atualização** da população.

Codificação da População

- A **representação dos cromossomos** é fundamental para a codificação.
- Consiste em uma maneira de traduzir a informação do problema em uma maneira viável de ser tratada pelo computador.
- Cada pedaço indivisível desta representação é chamado de um **gene**, por analogia aos genes que compõem um cromossomo biológico.

Codificação da População

- A representação computacional dos cromossomos é **completamente livre**
- **Cromossomos podem ser:**
 - Strings de bits (0101 ... 1100)
 - Números reais (43.2 -33.1 ... 0.0 89.2)
 - Listas de regras (R1 R2 R3 ... R22 R23)
 - Qualquer estrutura de dados!

Exemplo - Codificação da População

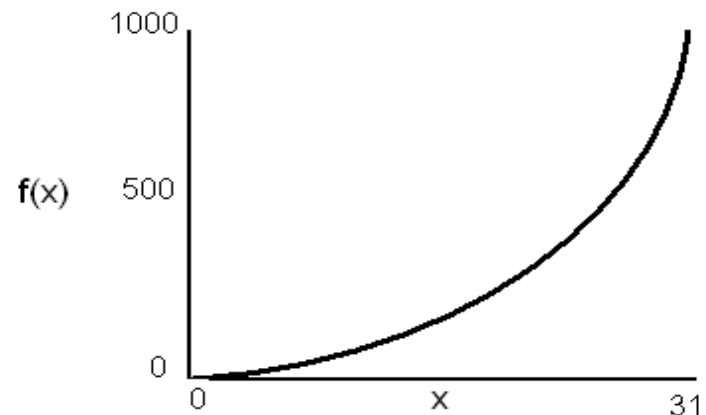
- **Objetivo:** Encontrar o máximo da função $f(x)=x^2$ no intervalo $[0,31]$.
- Os indivíduos da população precisam armazenar o valor de uma **variável inteira**.
- Podemos **codificar cada indivíduo da população** como uma sequência de **5 bits**

- | | | | | |
|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 |
|---|---|---|---|---|

 $x=20$

- | | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 |
|---|---|---|---|---|

 $x=3$



Exemplo - Codificação da População

- **Objetivo:** Encontrar o máximo da função $f(x)=x^2$ no intervalo $[0,31]$.
- Os indivíduos da população precisam armazenar o valor de uma **variável inteira**.
- Podemos **codificar cada indivíduo da população** como uma sequência de **5 bits**

•

1	0	1	0	0
---	---	---	---	---

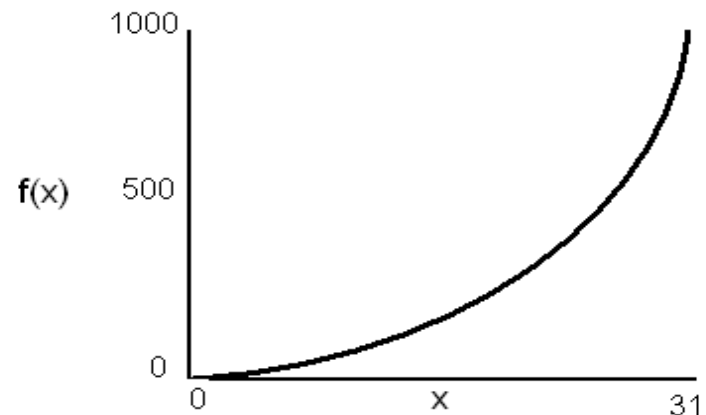
 $x=20$

•

0	0	0	1	1
---	---	---	---	---

 $x=3$

cromossomo



Exemplo - Codificação da População

- **Objetivo:** Encontrar o máximo da função $f(x)=x^2$ no intervalo $[0,31]$.
- Os indivíduos da população precisam armazenar o valor de uma **variável inteira**.

- Podemos codificar a população como uma sequência binária.

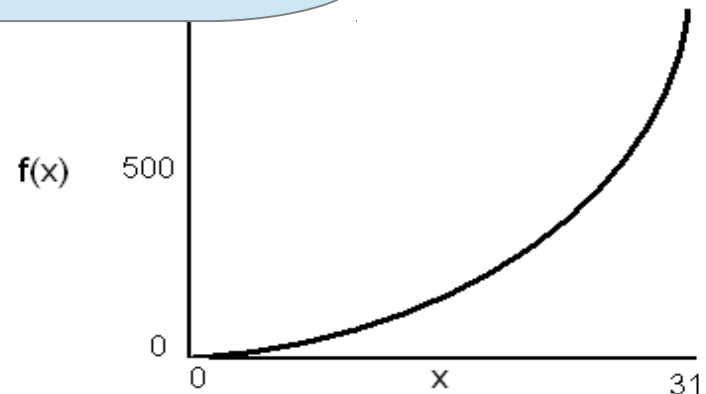
Em um indivíduo com codificação binária, cada bit é um gene.

- | | | | | |
|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 |
|---|---|---|---|---|

 $x=20$

- | | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 |
|---|---|---|---|---|

 $x=3$



Função de Avaliação

- A **função de avaliação** é a maneira utilizada pelos algoritmos genéticos para determinar a **qualidade de um indivíduo** como solução do problema em questão.
- **A função de avaliação deve ser escolhida cuidadosamente.** Ela deve embutir todo o conhecimento que se possui sobre o problema a ser resolvido.

Exemplo: Função de Avaliação

- **Objetivo:** Encontrar o máximo da função $f(x)=x^2$ no intervalo $[0,31]$.
- A função de avaliação para este caso consiste simplesmente em converter o número de binário para inteiro e depois elevá-lo ao quadrado.
- Indivíduos que tiverem maiores valores na função de avaliação são os mais aptos.

Seleção dos Pais

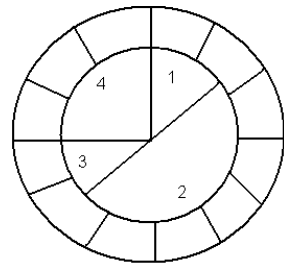
- O método de **seleção de pais** deve tentar simular o **mecanismo de seleção natural** que atua sobre as espécies biológicas:
 - Os pais mais capazes geram mais filhos, mas os menos aptos também podem gerar descendentes.
- Temos que privilegiar os indivíduos com função de avaliação alta, sem desprezar completamente aqueles indivíduos com função de avaliação extremamente baixa.
- Isto ocorre pois até indivíduos com péssima avaliação podem ter características genéticas que sejam favoráveis à criação de um "**super indivíduo**".

Seleção dos Pais

- Métodos:
 - Roleta
 - Torneio

Seleção dos Pais: método da roleta

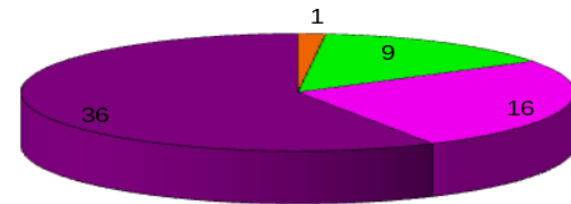
- Método mais comum de seleção de pais: **Roleta.**
- Cria-se uma roleta (virtual) na qual cada cromossomo recebe um pedaço proporcional à sua avaliação.
- Roda-se a roleta para sortear os indivíduos que serão pais de um novo indivíduo.



Exemplo: Seleção dos Pais, método da roleta

- Considerando a seguinte população gerada aleatoriamente para o problema de maximização de $f(x)=x^2$ no intervalo $[0,31]$

Indivíduo	Avaliação	Pedaço da roleta (%)	Pedaço da roleta (°)
00001	1	1.61	5.8
00011	9	14.51	52.2
00100	16	25.81	92.9
00110	36	58.07	209.1
Total	62	100.00	360.0



■ "00001" ■ "00011" ■ "00100" ■ "00110"

Seleção dos Pais: método do torneio

- Escolhemos n cromossomos da população atual, de forma aleatória.
- Dentre tais cromossomos escolhidos, o com maior valor de aptidão é selecionado para compor uma população intermediária.
- Em seguida, os demais cromossomos são recolocados na população e realiza-se o mesmo processo até que a população intermediária esteja completa.

Seleção dos Pais: método do torneio

População	Cromossomos candidatos	Selecionados
0100, 1000, 0001, 0010, 0111, 0000, 0011, 1001, 1011, 0110, 0101, 1010	0000, 0001, 0100	→ 0001
	0010, 0111, 1010	→ 1010
	0011, 1000, 1011	→ 0011
	.	.
	.	.
	.	.
	0110, 0101, 1001	→ 0101

Operadores Genéticos – Recombinação

- Técnicas de recombinação:
 - de um ponto
 - de dois pontos
 - de n pontos
 - uniforme

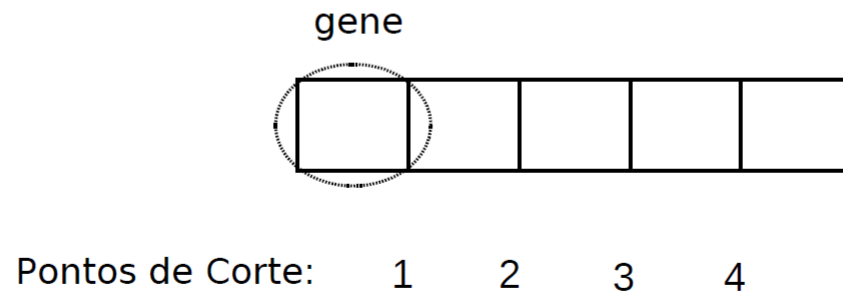
Operadores Genéticos – Recombinação de um ponto

- **Processo:**

- (1) Seleciona-se **dois pais** através processo de seleção de pais.
- (2) Seleciona-se **um ponto de corte** (uma posição entre dois genes de um cromossomo). Este ponto de corte é o ponto de separação entre cada um dos genes que compõem o material genético de cada pai.
- (3) A parte à esquerda do ponto de corte vai para um filho e a parte à direita vai para outro.

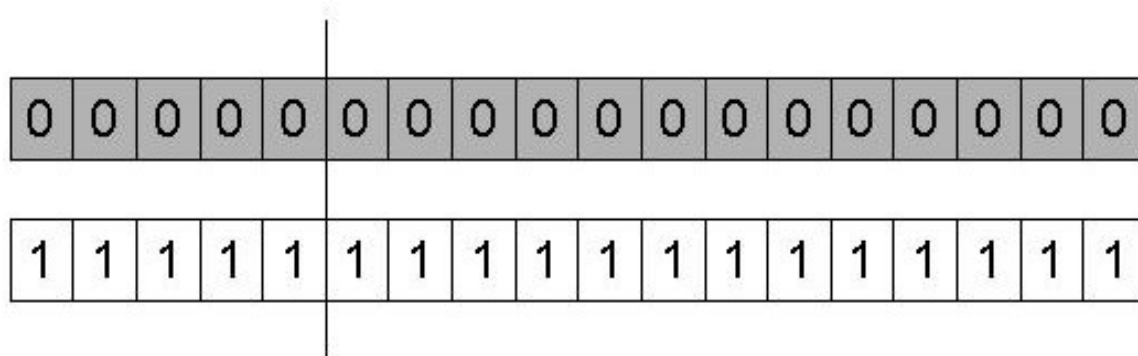
Operadores Genéticos – Recombinação de um ponto

- Cada indivíduo com n genes possui $n-1$ pontos de corte.
- Por exemplo, em um indivíduo com codificação binária, cada bit é um gene.

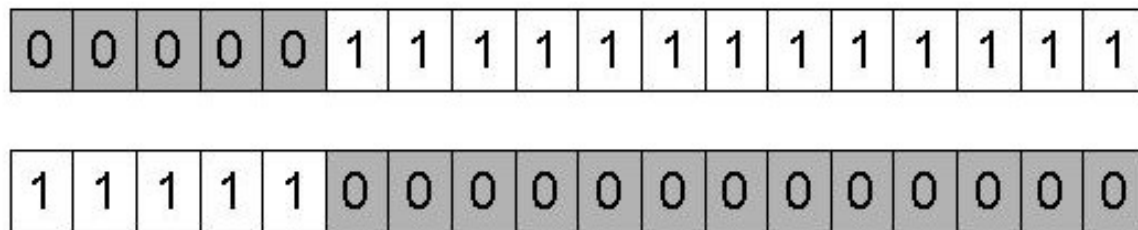


Exemplo: Recombinação de um ponto

parents



children

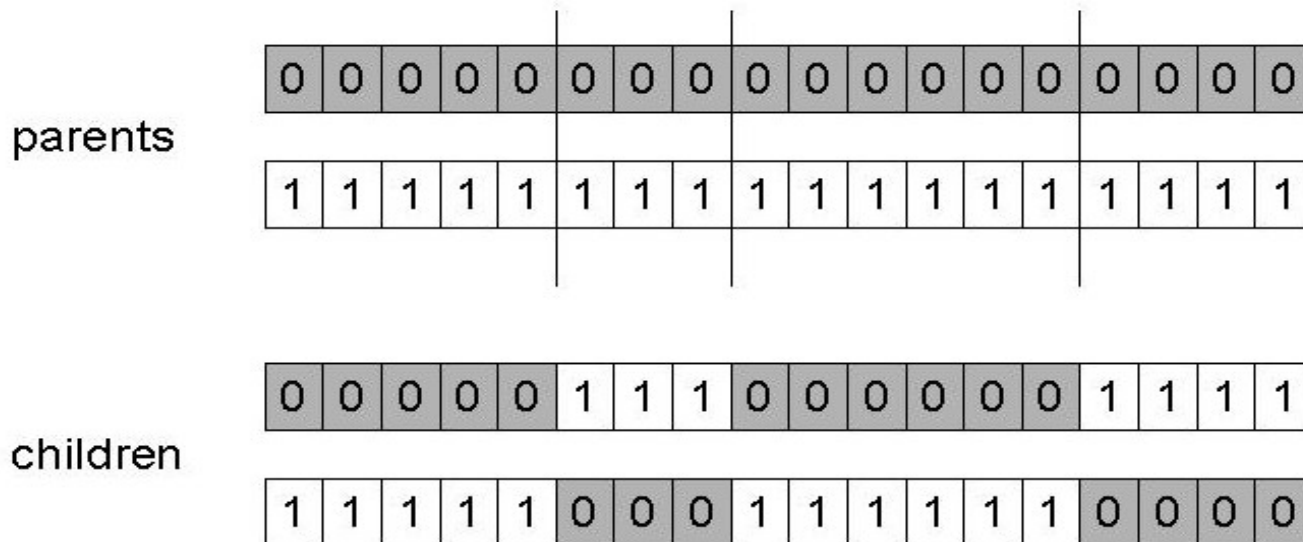


Operadores Genéticos – Recombinação de dois pontos

- Existem indivíduos que não podem ser gerados com a recombinação de somente um ponto. Ex: 1*****1.
- O algoritmo genético fica limitado na sua capacidade de gerar um certo conjunto de cromossomos.
- Para resolver isso é possível introduzir a recombinação de 2 pontos.
- Nele, em vez de sortearmos um só ponto de corte, sortearmos dois.

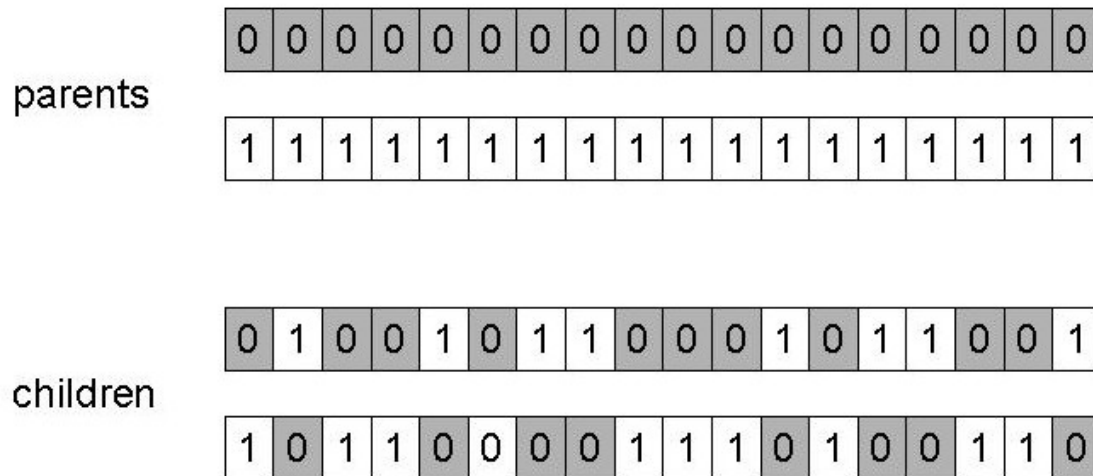
Operadores Genéticos – Recombinação de n pontos

- É possível tornar o operador em uma recombinação de n pontos.



Operadores Genéticos – Recombinação Uniforme

- Para cada gene é sorteado um número zero ou um.
 - Se o sorteado for 1, um filho recebe o gene do primeiro pai e o segundo filho o gene do segundo pai.
 - Se o sorteado for 0, o primeiro filho recebe o gene do segundo pai e o segundo filho recebe o gene do primeiro pai.



Operadores Genéticos - Mutação

- Depois de compostos os filhos, entra em ação o operador de **mutação**.
- Técnicas de mutação:
 - Mutação aleatória
 - Mutação por troca

Operadores Genéticos – Mutaç o aleat ria

- O operador atua com base em uma **probabilidade** extremamente baixa (da ordem de 5%) de **altera o aleat ria** do valor de um gene ou mais genes dos filhos.
- O valor da probabilidade   um dos par metros do algoritmo gen tico que pode alterar o resultado alcan ado pelo algoritmo.

Exemplo: Mutação aleatória

- Altere-se cada gene de forma independente com base em uma probabilidade pm
- pm é denominada taxa de mutação e costuma ser bem baixa.

Antes	$filho_1$	(0010101010010010101100)
	$filho_2$	(0011111011100000111111)
Depois	$filho_1$	(0010 <u>0</u> 010100100101 <u>1</u> 1100)
	$filho_2$	(0011111011 <u>0</u> 00000111111)

Operadores Genéticos – Mutação por troca

- n pares de genes são sorteados
- Após o sorteio os pares trocam valores entre si

Cromossomo antes da mutação:

(3 5 7 2 1 6 4 8)

Cromossomo depois da mutação:

(3 1 7 2 5 6 4 8)

Atualização

- Na atualização a população antiga é substituída por uma nova população.
- As formas mais conhecidas de atualização (critérios de sobrevivência) são:
 - $(x+y)$ chamada de estratégia soma ou elitismo
 - (x,y) chamada de estratégia vírgula

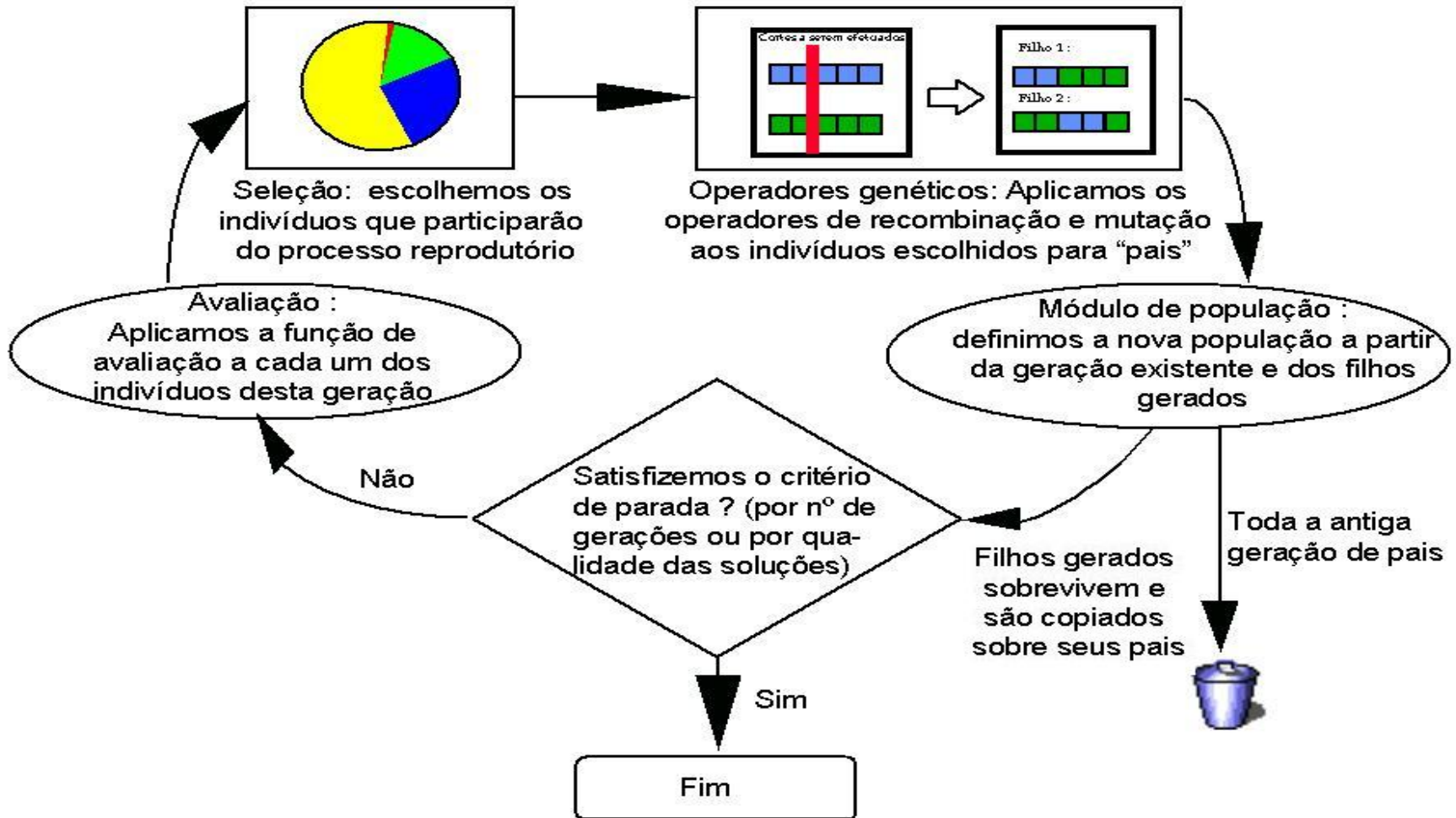
Atualização: estratégia soma

- Indivíduos da população anterior convivem com a população formada por seus filhos.
- Os n melhores indivíduos de cada geração não devem "morrer" junto com a sua geração, mas sim passar para a próxima geração para garantir que seus genomas sejam preservados.
- É uma forma de garantir que o algoritmo nunca regrida.

Atualização: estratégia vírgula

- A população anterior não convive com a próxima população
- Perdendo-se então soluções boas encontradas

Atualização: estratégia vírgula



Algoritmos Genéticos: Exemplo 0

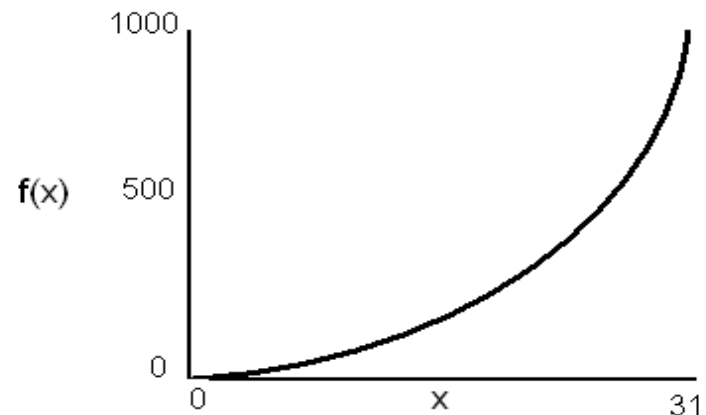
- **Objetivo:** Encontrar o máximo da função $f(x)=x^2$ no intervalo $[0,31]$.
- Podemos **codificar cada indivíduo da população** como uma sequência de **5 bits**

- | | | | | |
|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 |
|---|---|---|---|---|

 $x=20$

- | | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 |
|---|---|---|---|---|

 $x=3$



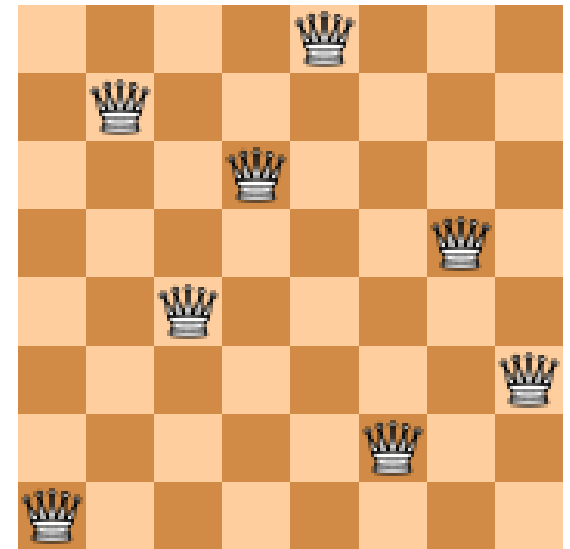
Algoritmos Genéticos: Exemplo 0

String no.	População inicial Gerada randomicamente	Valor de x Inteiro associado	f(x) x^2	%	Seleções
1	0 1 1 0 1	13	169	0,14	1
2	1 1 0 0 0	24	576	0,49	2
3	0 1 0 0 0	8	64	0,06	0
4	1 0 0 1 1	19	361	0,31	1
			1170	1,00	

Pool de reprodução	Seleção de pares	Ponto de crossover	Nova população	Valor de x	f(x)
0 1 1 0 1	2	4	0 1 1 0 0	12	144
1 1 0 0 0	1	4	1 1 0 0 1	25	625
1 1 0 0 0	4	2	1 1 0 1 1	27	729
1 0 0 1 1	3	2	1 0 0 0 0	16	256

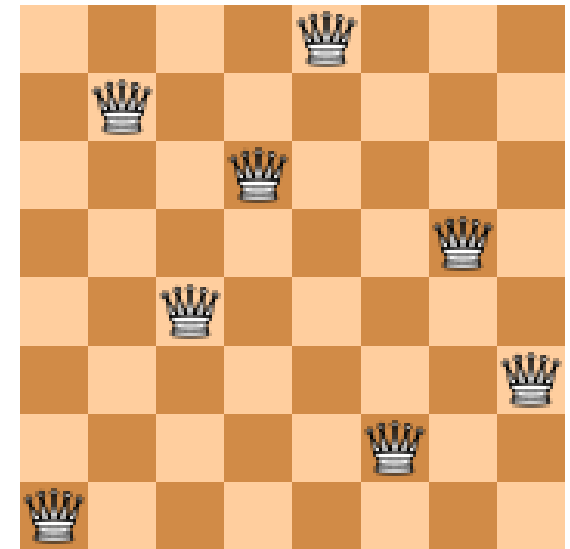
Algoritmos Genéticos - Exemplo 1

- **Problema:** 8 Rainhas
- **Como representar os indivíduos?**



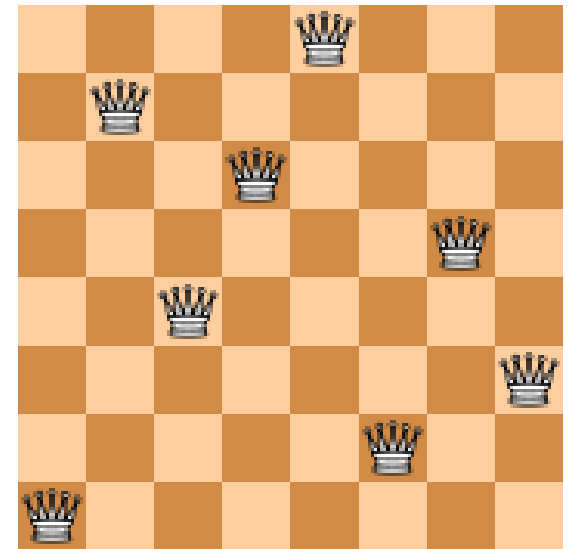
Algoritmos Genéticos - Exemplo 1

- **Problema:** 8 Rainhas
- **Como representar os indivíduos?**
 - 8 dígitos – cada um representando a posição da rainha em sua coluna.
 - **Exemplo:** (1, 7, 4, 6, 8, 2, 5, 3)
- **Qual a função de avaliação?**

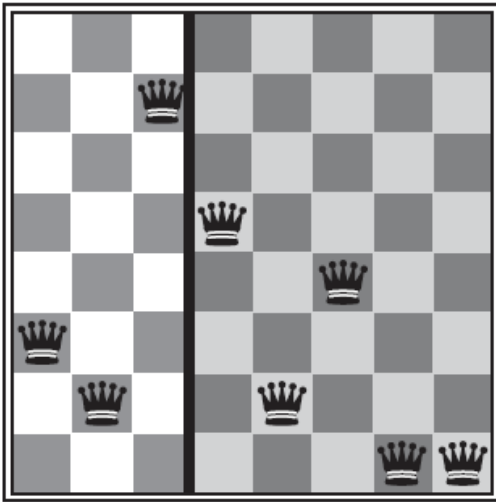


Algoritmos Genéticos - Exemplo 1

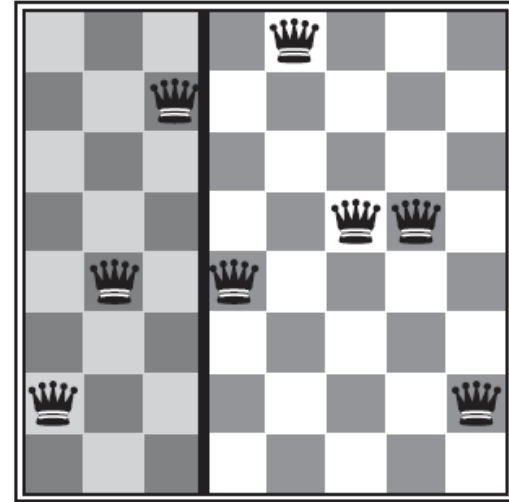
- **Problema:** 8 Rainhas
- **Como representar os indivíduos?**
 - 8 dígitos – cada um representando a posição da rainha em sua coluna.
 - **Exemplo:** (1, 7, 4, 6, 8, 2, 5, 3)
- **Qual a função de avaliação?**
 - Número de pares de rainhas não sendo atacadas.



Algoritmos Genéticos - Exemplo 1

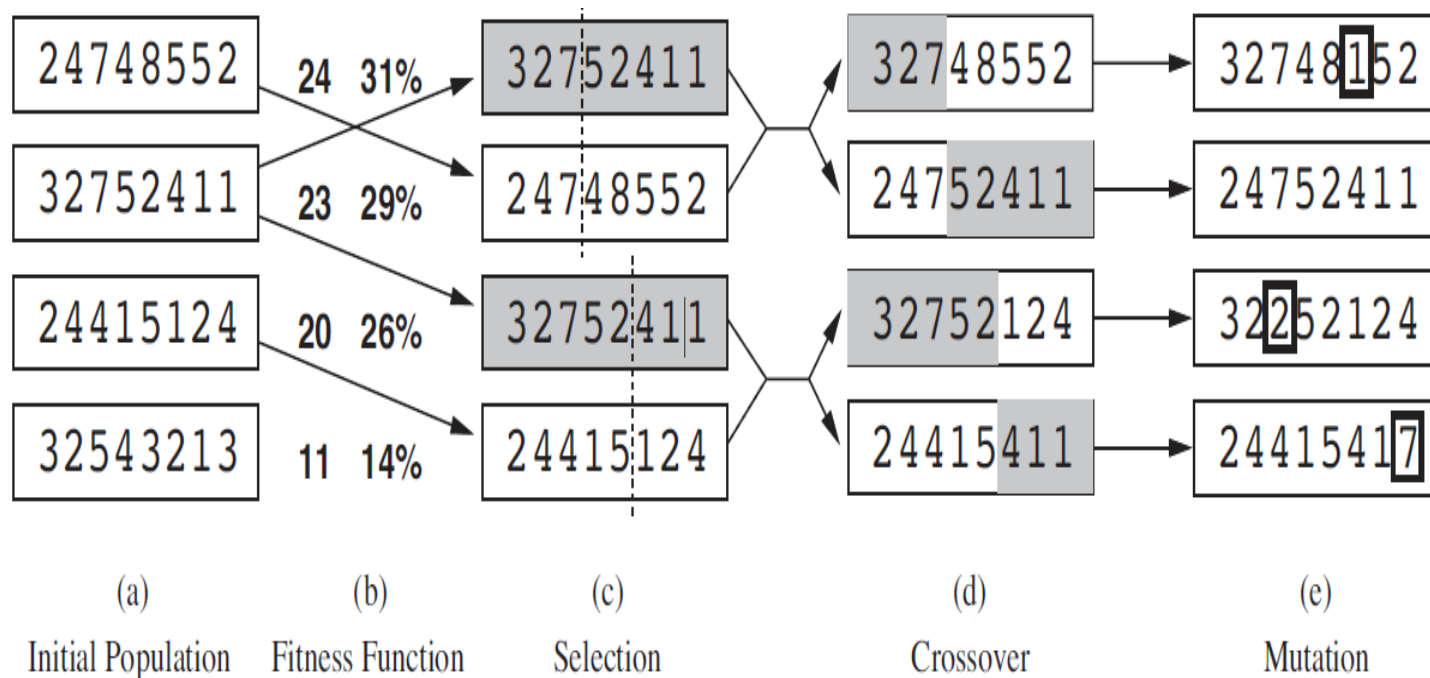


$$f(3, 2, 7, 5, 2, 4, 1, 1) = 23$$



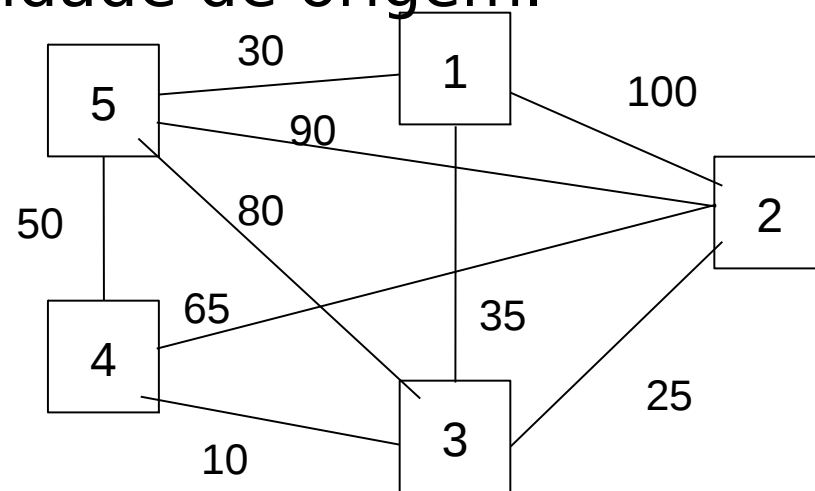
$$f(2, 4, 7, 4, 8, 5, 5, 2) = 24$$

Algoritmos Genéticos - Exemplo 1



Algoritmos Genéticos - Exemplo 2

Problema do caixeiro viajante: Deve-se encontrar o caminho mais curto para percorrer n cidades (visitando cada uma pelo menos uma vez), retornando à cidade de origem.



- **Como representar os indivíduos?**

Algoritmos Genéticos - Exemplo 2

Problema do caixeiro viajante: Deve-se encontrar o caminho mais curto para percorrer n cidades (visitando cada uma pelo menos uma vez), retornando à cidade de origem.

- **Como representar os indivíduos?**

- Cada indivíduo pode ser representador por uma lista **ordenada** de cidades, que indica a ordem em que cada uma será visitada.

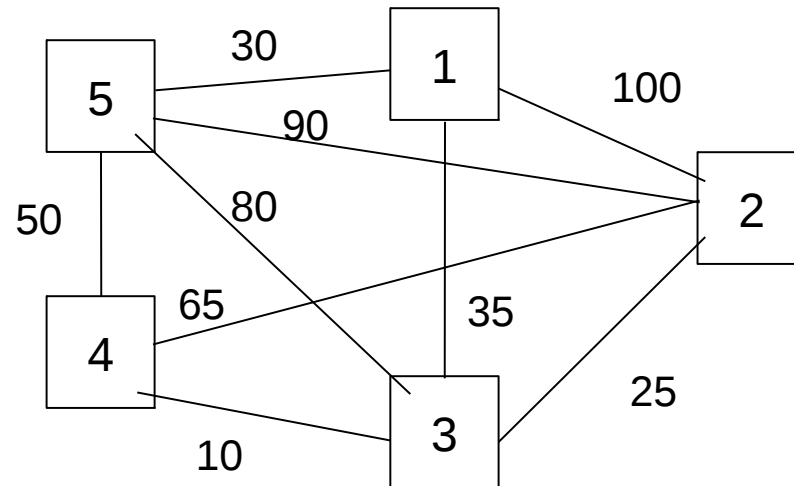
- **Exemplo:** (3 5 7 2 1 6 4 8)

Algoritmos Genéticos - Exemplo 2

- Cada cromossomo tem que conter **todas as cidades** do percurso, **apenas uma vez**.
- **Considerando 8 cidades:**
 - **Cromossomos válidos:**
 - (1 2 3 4 5 6 7 8)
 - (8 7 6 5 4 3 2 1)
 - (1 3 5 7 2 4 6 8)
 - **Cromossomos inválidos:**
 - (1 5 7 8 2 3 6) falta a cidade 4
 - (1 5 7 8 2 3 6 5) falta a cidade 4 e a cidade 5 está representada 2 vezes

Algoritmos Genéticos - Exemplo 2

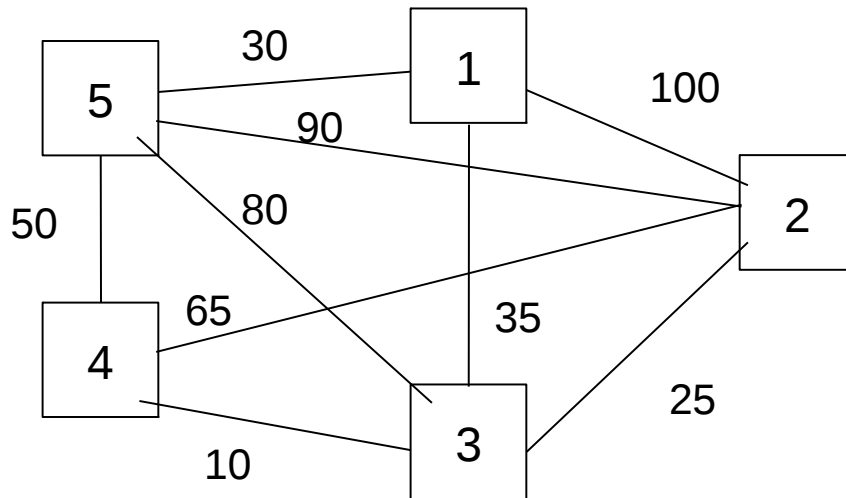
- Qual a função de avaliação?



Algoritmos Genéticos - Exemplo 2

- **Qual a função de avaliação?**

- A **função de avaliação** soma das distâncias entre as cidades



- $f(1\ 3\ 5\ 4\ 2) = 35 + 80 + 50 + 65 + 100 = 330$

Algoritmos Genéticos - Exemplo 2

- **Recombinação (uniforme):**

- Pai1 (3 5 7 2 1 6 4 8)

- Pai2 (2 5 7 6 8 4 3 1)

1) Gera-se uma string de bits aleatória do mesmo tamanho que os pais:

1 0 0 1 0 1 0 1

2) Copia-se para o filho 1 os elementos do pai 1 referentes àquelas posições onde a string de bits possui um 1:

3 _ _ 2 _ 6 _ 8

3) Elementos não copiados do pai1:

5 7 1 4

4) Permuta-se essa lista de forma que os elementos apareçam na mesma ordem que no pai 2 (5 7 4 1) e copia-se eles para dentro do Filho1:

3 5 7 2 4 6 1 8

Algoritmos Genéticos - Exemplo 2

Mutação baseada na troca da ordem de duas cidades:

Cromossomo antes da mutação:

(3 5 7 2 1 6 4 8)

Escolhem-se dois elementos aleatórios dentro do cromossomo e trocam-se as suas posições. Ex

Cromossomo antes da mutação:

(3 5 7 2 1 6 4 8)

Cromossomo depois da mutação:

(3 1 7 2 5 6 4 8)

Algoritmos Genéticos - Exemplo 2

Mutação baseada na troca de posição de uma cidade

Cromossomo antes da mutação:

(3 5 7 2 1 6 4 8)

Escolhem-se um elemento aleatório dentro do cromossomo e uma posição. O elemento é inserido na posição sorteda. Ex:

Cromossomo antes da mutação:

(3 **5** 7 2 1 6 4 - 8)

Cromossomo depois da mutação:

(3 7 2 1 6 4 **5** 8)

Algoritmos Genéticos

- Questões importantes na definição de um problema em algoritmos genéticos:
 - Representação dos indivíduos.
 - Parâmetros do sistema (tamanho da população, taxa de mutação...).
 - Políticas de seleção de indivíduos.
 - Operadores genéticos (recombinação e mutação)
 - Critério de atualização da população (sobrevivência)
 - Critérios de parada.
 - Função de avaliação (a mais importante e mais complicada de ser definida).

Algoritmos Genéticos:

Observações importantes

- ▶ População inicial
 - ▶ Não pode ser excessivamente pequena
 - ▶ Pouca representatividade do espaço de busca
 - ▶ Não pode ser excessivamente grande
 - ▶ Demora na convergência
 - ▶ Para melhorar a representatividade, a população inicial pode possuir indivíduos igualmente espaçados no espaço de busca

Algoritmos Genéticos:

Observações importantes

► Critérios de Parada

- Número máximo de gerações
- Função objetivo com valor ótimo alcançado (quando esse valor é conhecido)
- Convergência na função objetivo (i.e., quando não ocorre melhoria significativa da função)

Algoritmos Genéticos:

Observações importantes

- ▶ Operador Crossover considera características importantes presentes nos pais
 - Podemos usar uma taxa relativamente alta, mas cuidado com efeitos destrutivos
- ▶ Operador Mutação explora novas características nos indivíduos que seriam possivelmente úteis
 - Podemos usar uma taxa relativamente baixa, mas dependendo do problema podemos usar taxas mais altas

Algoritmos Genéticos:

Observações importantes

- ▶ Convergência Prematura
 - Em algumas execuções, o AG pode convergir para soluções iguais
 - Cromossomos com boa aptidão (mas ainda não ótimos) que geram filhos com pouca diversidade
 - Nesses casos, aconselha-se:
 - Aumento da taxa de mutação e crossover
 - Evitar a inserção de filhos duplicados