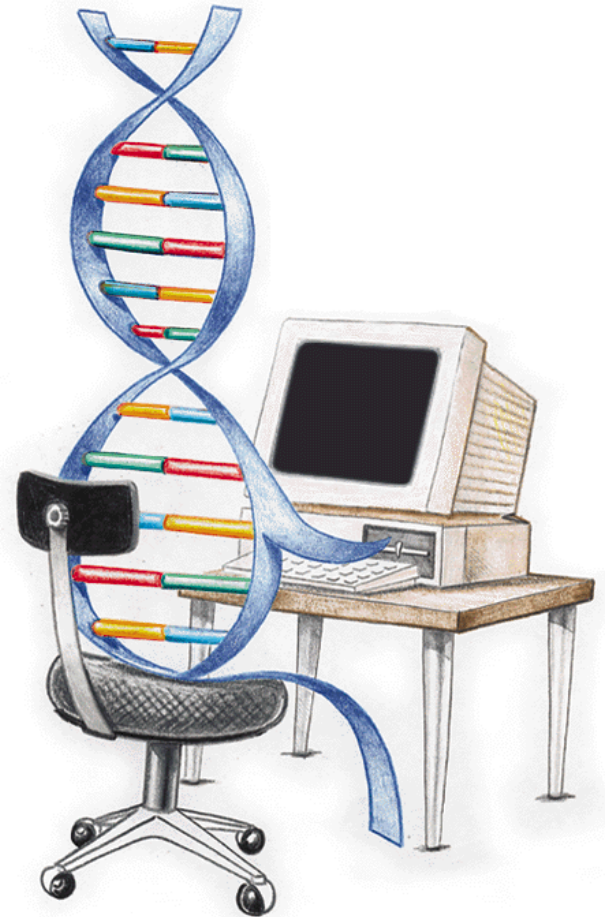


Algoritmos Genéticos Aplicados em Otimização

Prof. Ademir A. Constantino



Algoritmos Genéticos

- Algoritmo probabilístico baseado na analogia entre o processo de **evolução natural** e a seleção de soluções de um problema de otimização combinatória.
- Mantém uma população representada pelos seus **cromossomos**. Novos cromossomos são gerados a partir da população corrente e são incluídos na população, ao mesmo tempo que outros são excluídos.

Aplicações

- Grande variedade de aplicações

- * Otimização

- Indústria, solução de problemas: máquinas x processos, alocação de recursos, rota de veículos.

- * Busca

- Mineração de Dados, descoberta de conhecimento em bases de dados, indução de classificadores (características x doenças, estrutura de proteínas)

- * Aprendizado e adaptação

Algoritmos Genéticos

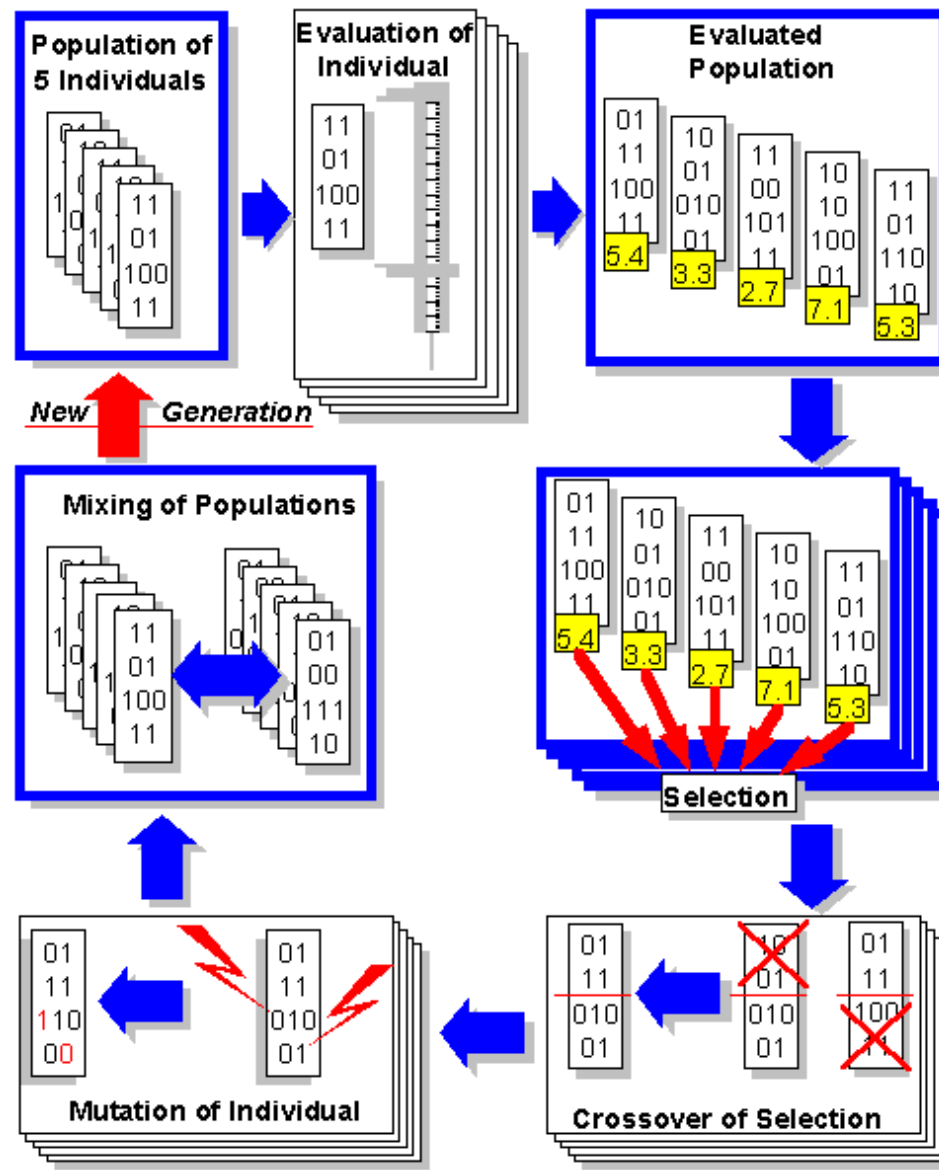
Elementos básicos:

- Cromossomo : um indivíduo (solução)
- Gene: um elemento do cromossomo
- População: um conjunto de cromossomos
- Operadores genéticos:
 - * Seleção: consiste em selecionar os cromossomos (pais) para reprodução;
 - * Reprodução (*crossover*): executar uma combinação (cruzamento) dos cromossomo pais para gerar novos cromossomos filhos.
 - * Mutação: é uma modificação arbitrária de uma parte (pequena) do cromossomo.

Visão Geral de um Algoritmo Genético

- A seguir veremos um exemplo clássico da aplicação de algoritmos genéticos:
 - * Codificação de cada indivíduo é binária;
 - * População com 5 indivíduos;
 - * O cruzamento (*crossslover*) é feito a partir de 2 indivíduos;
 - * A mutação modifica alguns bits do indivíduo.

Visão Geral de um Algoritmo Genético



Algoritmos Genéticos

- Algoritmo Básico

Gerar população inicial

while critério-de-parada **do**

Avaliação: Calcula a qualidade dos cromossomos filhos

Seleção: Escolher cromossomos reprodutores

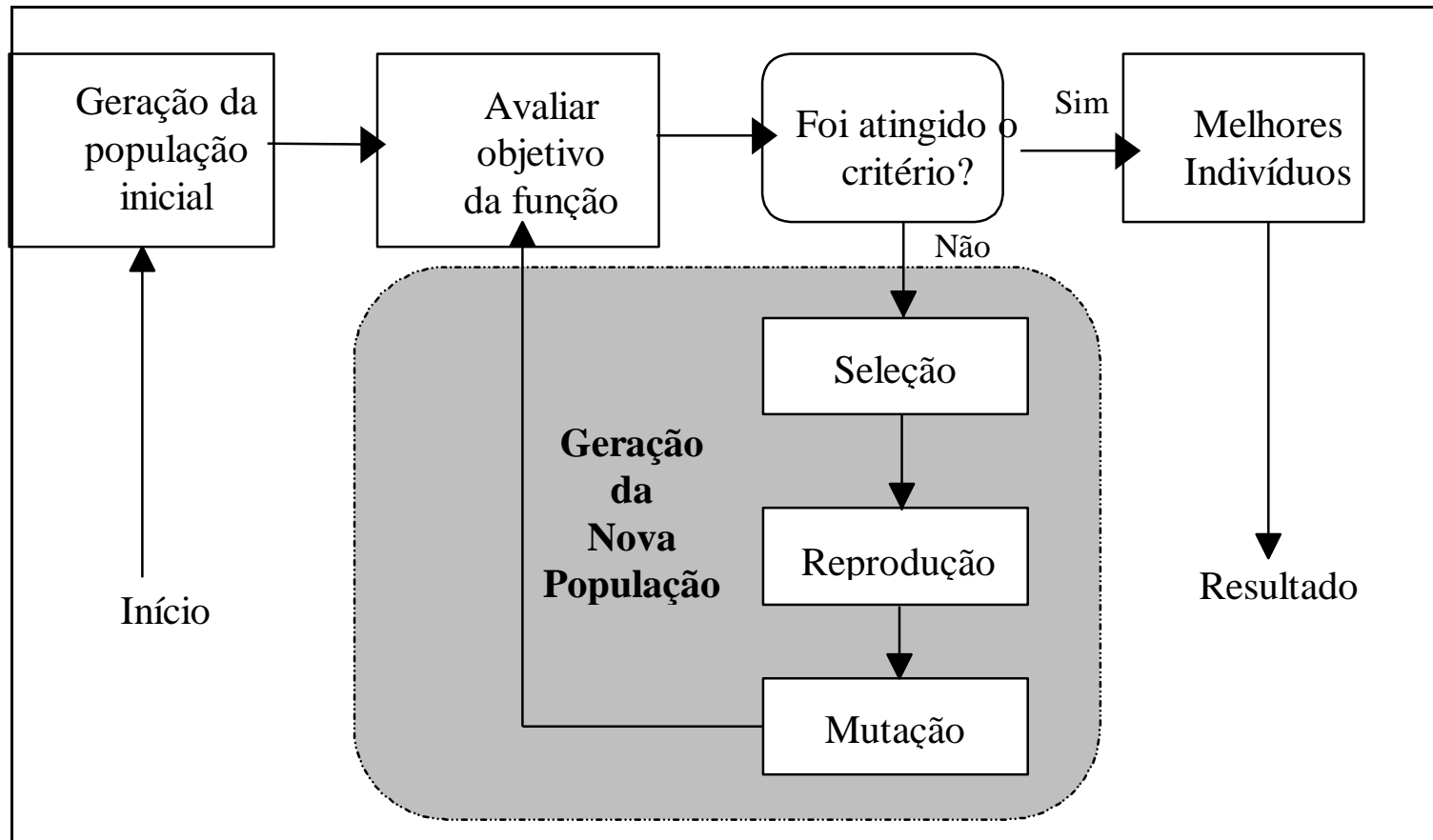
Cruzamento: Fazer o cruzamento dos reprodutores

Mutação: Gerar mutações da população

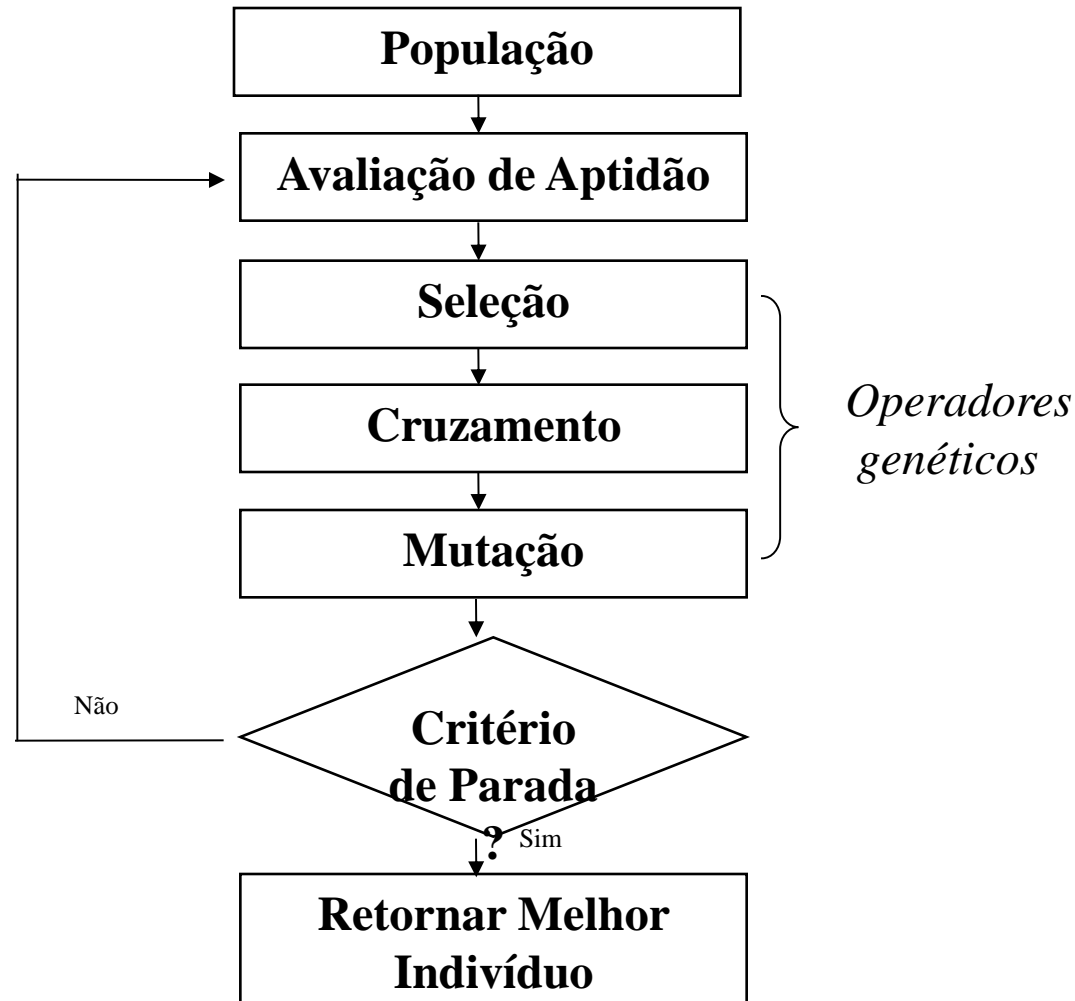
Atualização: Atualizar a população

end-while

Estrutura de um Algoritmo Genético



Estrutura de um Algoritmo Genético



Algoritmos Genéticos

- Parâmetros mais comuns:
 - * tamanho da população
 - * critério de seleção
 - * critério de sobrevivência dos cromossomos
 - * taxa de mutação
 - * critério de parada

Representação do Cromossomo

Existem várias formas de representação.
Veremos alguns exemplos.

- Codificação Binária - *String* de Bits

- Cromossomo A 101100101100101011100101
- Cromossomo B 111111100000110000011111

- * Aplicações:

- Forma clássica para representação para achar raiz de funções;
- Problema da Mochila

Representação do Cromossomo

- Codificação por Permutação

- Cromossomo A 1 5 3 2 6 4 7 9 8

- Cromossomo B 8 5 6 7 2 3 1 4 9

- * Aplicação:

- Problema do Caixeiro Viajante

- Codificação por Valor

- Cromossomo A 4,5 3,8 8.1

- * Aplicação:

- Otimização de funções não lineares com solução n-dimensional.

População

- Conjunto de indivíduos - possíveis soluções de um problema
 - * Populações pequenas têm grandes chances de perder a diversidade necessária (exploração de todo o espaço de soluções)
 - * Populações grandes perderá grande parte de sua eficiência pela demora em avaliar a função de aptidão (*fitness*).

Avaliação - Função de Aptidão

- Função de Avaliação (aptidão ou *fitness*)
 - * utilizada para **quantificar** a qualidade genética dos cromossomos, correspondendo à **função de custo** em problemas de otimização combinatória.
 - * em otimização o mais comum é utilizar a própria **função objetivo** do problema para avaliar
- Mede a adaptação do indivíduo ou quão boa é a solução dada por este indivíduo (cromossomo).
- Representativa do problema: diferencie uma solução boa de uma má.

Seleção dos Indivíduos

- Operador escolhe quais indivíduos participarão na criação da próxima geração.
- Técnicas:
 - * **Roleta** (proporcional à aptidão de cada indivíduo);
 - * Por **Classificação** (proporcional a sua posição no *ranking*);
 - * **Torneio** (leva em consideração a aptidão, porém, mais rápido que os anteriores).

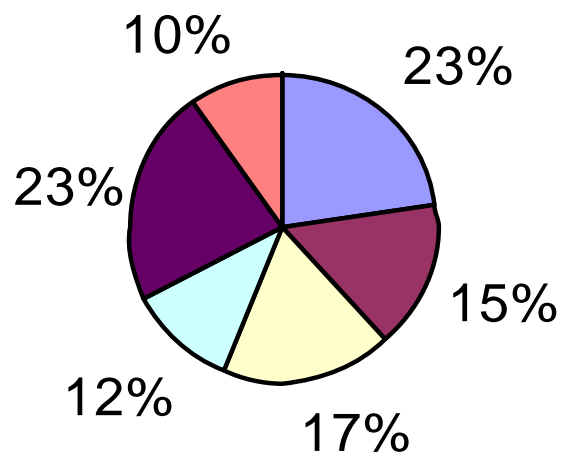
Seleção pela Roleta

- Exemplo:

Indivíduos	Fitness	% Fitness
10101010110101010111	12	23,08
00001001010101110010	8	15,38
00001100001011011101	9	17,31
00000110010010000010	6	11,54
11100011100010011111	12	23,08
00010101001000010000	5	9,62
Total	52	100,00

Exemplo

Roleta



Seleção por Classificação

- O Seleção por Classificação primeiramente ordena a população com base valor de aptidão de cada cromossomo, depois cada cromossomo recebe um valor de acordo com esta classificação. O pior terá valor 1, o segundo pior terá valor 2 e assim sucessivamente. O melhor terá valor N igual ao número de indivíduos da população.
- **Um possível problema:** Este método pode apresentar convergência muito lenta porque os melhores cromossomos não diferem muito dos outros.

Seleção por Torneio

- Seleciona k indivíduos aleatoriamente e escolhe o mais adaptado (maior *fitness* – para problema de maximização).
- No caso de necessitar selecionar mais de um indivíduo, o processo é repetido.
- O valor de k normalmente é pequeno, por exemplo, 2, 3 ou 4.

Cruzamento

- Cruzamento: cria novos indivíduos misturando características de dois ou mais indivíduos pais (*crossover*)
- Cópia de segmentos entre os pais
- Tipos clássicos: um ponto, dois pontos, multi-ponto, uniforme e inversão

Cruzamento

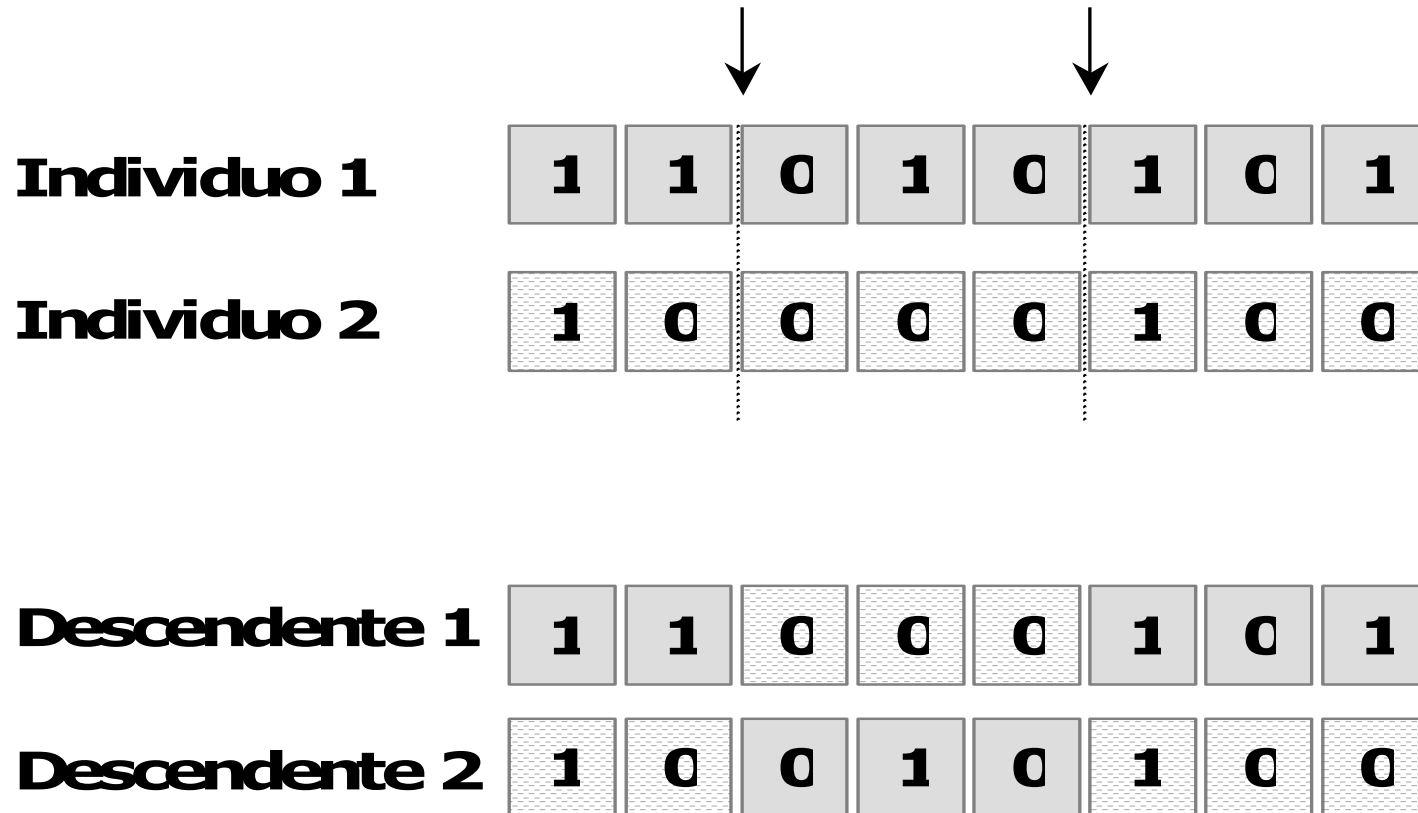
- Pai 1: 10101010110101010111
- Pai 2: 00001001010101110010

Cruzamento em **um ponto**

* 10101010110101110010, 00001001010101010111

Cruzamento **uniforme**: os filhos são formados a partir dos bits dos pais (sorteado)

Cruzamento em Dois Pontos



Operadores de Cruzamento para o PCV

Operadores encontrados na literatura

Operador	Abreviação	Descrição
<i>Partially-Mapped Crossover</i>	PMX	Dois pontos de corte são selecionados aleatoriamente em ambas estruturas dos cromossomos pais, em seguida a sequência de genes delimitada em um cromossomo pai é mapeando para o outro cromossomo formando um cromossomo descendente (subseção 3.2.1).
<i>Cycle Crossover</i>	CX	Os genes são subdivididos em ciclos, e os filhos são gerados pela seleção destes ciclos nos cromossomos pais (subseção 3.2.2).
<i>Order Crossover 1</i>	OX1	Um novo cromossomo é criado selecionando uma porção de genes de um cromossomo pai com o complemento dos genes do outro cromossomo pai, preservando relativamente a ordem de ocorrência dos genes nos pais (subseção 3.2.3).
<i>Order based Crossover</i>	OX2	Seleciona-se posições aleatoriamente dentro da estrutura de um cromossomo pai, e copie os genes presentes nestas posições para cromossomo filho preservando a ordem de precedência do outro cromossomo pai (subseção 3.2.4).
<i>Position Based Crossover</i>	POS	Seleciona-se aleatoriamente vários genes na estrutura do cromossomo pai copiando os genes para o cromossomo filho, mateando os genes no filho na mesma posição do cromossomo pai original (subseção 3.2.5).
<i>Genetic edge recombination crossover</i>	ER	Construa um cromossomo filho, selecionando arestas dos cromossomos pais com base em informações extraídas da tabela de adjacência de arestas dos pais (subseção 3.3.1).
<i>Edge Assembly Crossover</i>	EAX	Construa cromossomos filhos por meio da remoção de arestas de um cromossomo pai definidas pela criação de subciclos com arestas dos dois cromossomos pais e conectando os subciclos formados por meio de método de busca local(subseção 3.3.2).

Exemplo - Prob. Caixeiro Viajante

Operador **OX1** (Order crossover):

OX1 – constrói os filhos escolhendo subsequência de um pai e preservando a ordem relativa do segundo.

Exemplo:

$$p_1 = (1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9)\text{ e}$$

$$p_2 = (4\ 5\ 2\ 1\ 8\ 7\ 6\ 9\ 3)$$

Primeiro, os segmentos entre os cortes são copiados para os filhos:

$$o_1 = (x\ x\ x\ 4\ 5\ 6\ 7\ x\ x)\text{ e}$$

$$o_2 = (x\ x\ x\ 1\ 8\ 7\ 6\ x\ x).$$

Exemplo - Prob. Caixeiro Viajante

Depois disso, inicia-se a partir do segundo ponto de corte de um pai copiando os vértices do outro pai na mesma ordem.

A sequência de vértices no segundo pai é:

9 – 3 – 4 – 5 – 2 – 1 – 8 – 7 – 6

Retire os vértices já presentes no primeiro filho, então resulta a seguinte sequência:

9 – 3 – 2 – 1 – 8

Esta sequência é substituído no primeiro filho:

$o_1 = (2 \ 1 \ 8 \ 4 \ 5 \ 6 \ 7 \ 9 \ 3)$, e de forma similar para o segundo.

$o_2 = (3 \ 4 \ 5 \ 1 \ 8 \ 7 \ 6 \ 9 \ 2)$

Exemplo - Prob. Caixeiro Viajante

Operador OX2:

Pai1 :[6, 3, 5, 1, 2, 4, 8, 7] Pai2:[8, 5, 2, 4, 3, 1, 7, 6]

Sorteio de posições em Pai2: 1, 3 e 8

Copia o Pai1 excluindo os vértices correspondentes as posições sorteadas, ou seja, vértices 8, 2 e 6

Filho1:[, 3, 5, 1, , 4, , 7]

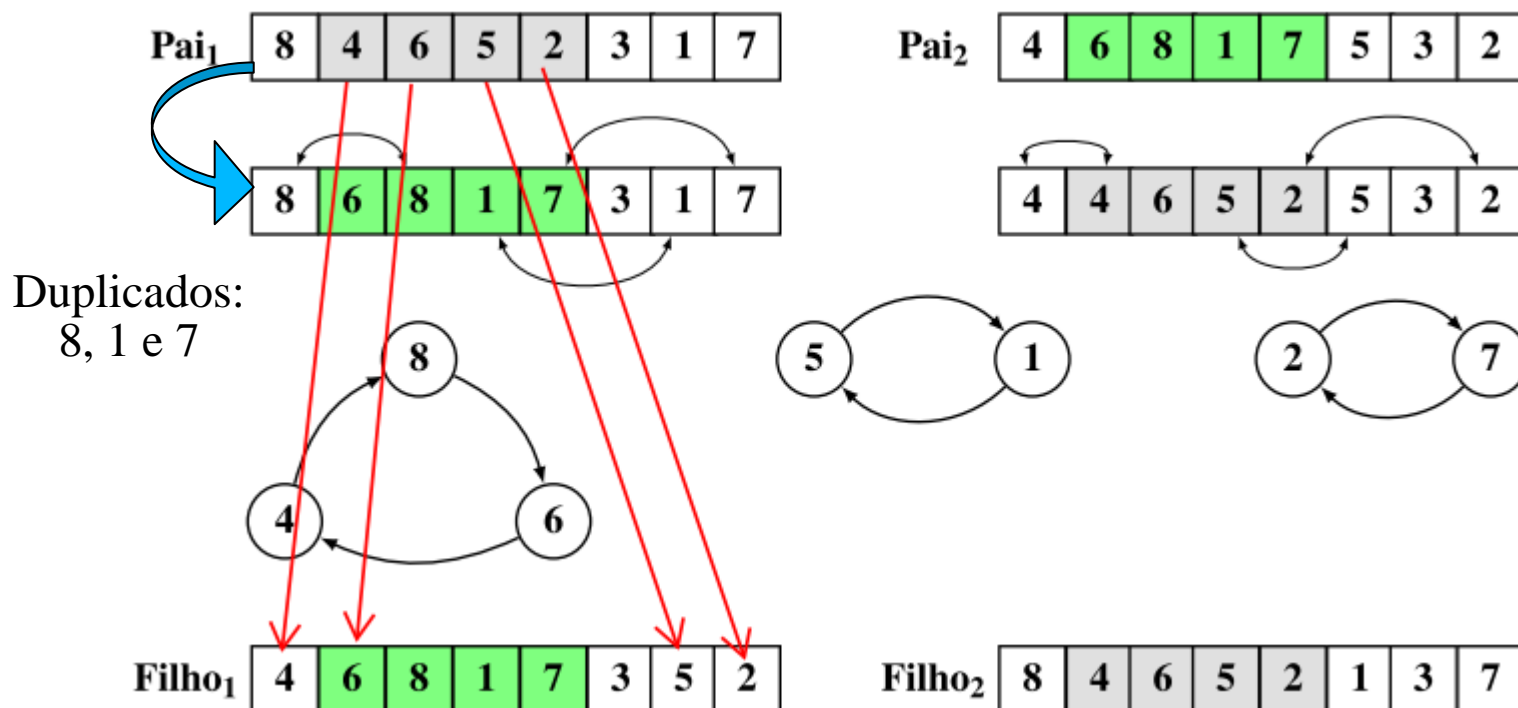
Filho1:[8, 3, 5, 1, 2, 4, 6, 7]



Exemplo - Prob. Caixeiro Viajante

Operador PMX (Partially Mapped Crossover):

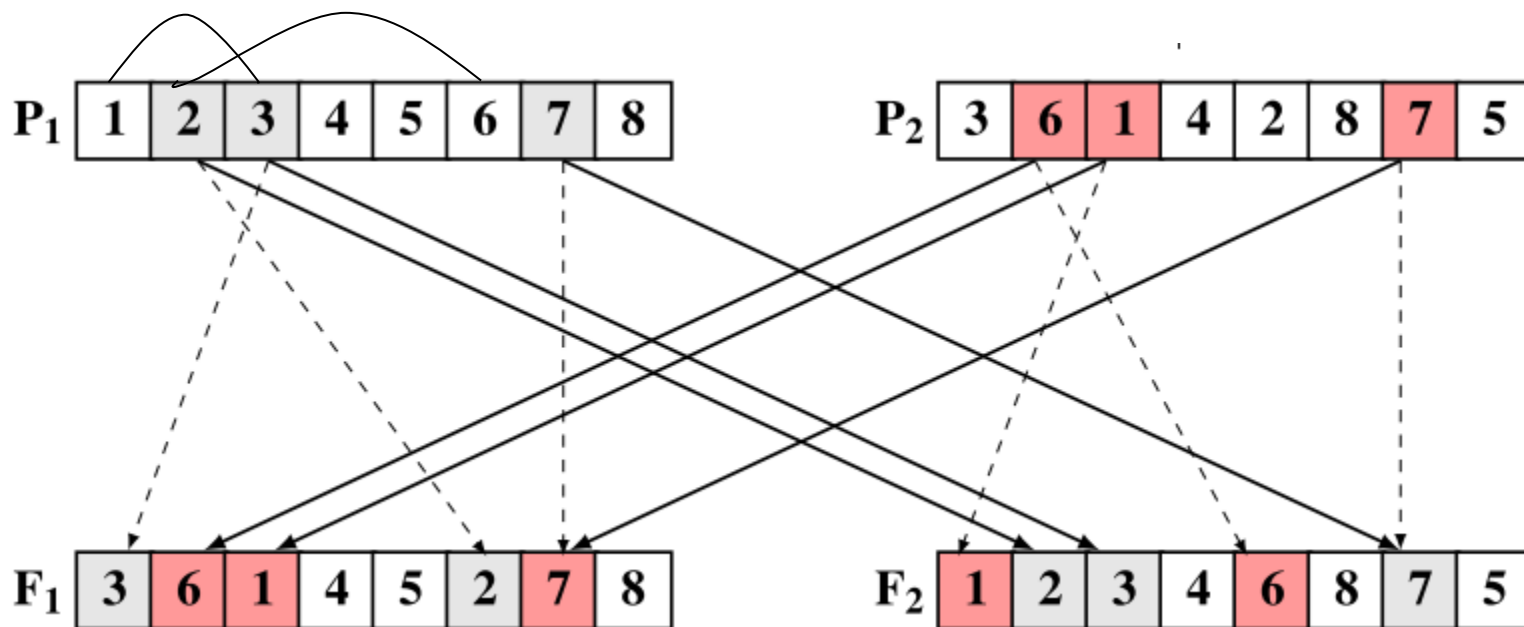
Sorteio de dois pontos de cortes:



Exemplo - Prob. Caixeiro Viajante

Operador POS (*Position based Crossover*)

Posições sorteadas: 2,3 e 7



Mutação

- Esta operação inverte aleatoriamente alguma característica do indivíduo
- Cria novas características que não existiam
- Mantém diversidade na população

Manutenção da população

- Algumas técnicas clássicas para manter a população durante a evolução do algoritmo.
- Técnicas clássicas:
 - * **Populacional**: cada iteração do algoritmo genético, uma nova população é gerada substituindo a anterior;
 - * **Elitismo**: semelhante a técnicas populacional, porém, parte da população anterior é mantida, ou seja, os melhores indivíduos (elite) da população anterior são mantidos.
 - * ***Steady Stated***: cada vez que um novo indivíduo é gerado, ele entra na população atual se tiver uma aptidão melhor que o pior indivíduo, assim, a população é atualizada a cada iteração do algoritmo.

Critérios de Parada

- Regras mais comuns:
 - * Um número pré-determinado de iterações (ou gerações).
 - * Um solução satisfatória foi alcançada.
 - * Não houve melhoria na qualidade da solução por um número pré-determinado de iterações (ou gerações).
 - * Estagnação das soluções.

Algoritmos Genéticos

- RAMIREZ, J.: Los Algoritmos Genéticos. Un enfoque introductorio. Qué son, de donde vienen, a donde van.
<http://www.vilspa.esa.es/~jrz/introesp.html>
- BEASLY, DAVID : An Overview of Genetic Algorithms, 1993.
<ftp://ftp.aic.nrl.navy.mil/pub/galist/README>
- <http://www.genetic-programming.com/gpanimatedtutorial.html>
- <http://www.geneticprogramming.com/Tutorial/tutorial.html>
- Carlos Eduardo Botelho Lourenço
<http://black.rc.unesp.br/ccomp/algoritmo/>

Applets

- <http://www.ads.tuwien.ac.at/raidl/tspga/TSPGA.html>
- <http://cs.felk.cvut.cz/~xobitko/ga/tspexample.html>
- <http://www.heatonresearch.com/articles/65/page1.html>
- <http://www.rennard.org/alife/english/gavgb.html>

Algoritmos Genéticos

- Algoritmo Básico

Gerar população inicial

while critério-de-parada **do**

Avaliação: Calcula a qualidade dos cromossomos filhos

Seleção: Escolher cromossomos reprodutores

Cruzamento: Fazer o cruzamento dos reprodutores

Mutação: Gerar mutações da população

Atualização: Atualizar a população

end-while

Algoritmos Genéticos

- Algoritmo Básico

Gerar população inicial

while critério-de-parada **do**

Avaliação: Calcula a qualidade dos cromossomos filhos

Seleção: Escolher cromossomos reprodutores

Cruzamento: Fazer o cruzamento dos reprodutores

Mutação: Gerar mutações da população

BuscaLocal: Realiza uma busca na vizinhança

Atualização: Atualizar a população

end-while