LifeInspiration

Projecto de Sistemas de Informação

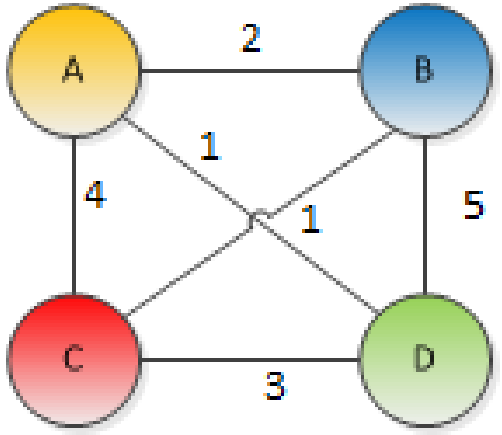
Life Inspiration V0.3

Caixeiro Viajante

- Problema:
  - Procura-se encontrar um ou mais caminhos, para percorrer todas as cidades e voltar ao ponto de origem, sem nunca passar numa cidade duas vezes e obter o caminho com menos custo.
- Esquema:
  - Tamanho do gene igual ao número de cidades;
  - As cidades são indexadas para que as suas representações sejam números inteiros;
  - A ordem com que os alelos aparecem no gene é a mesma ordem pela qual passamos nas cidades.

Caixeiro Viajante

- Representação de Caminhos e custos



	A	B	C	D
A	0	2	4	1
B	2	0	1	5
C	4	1	0	3
D	1	5	3	0

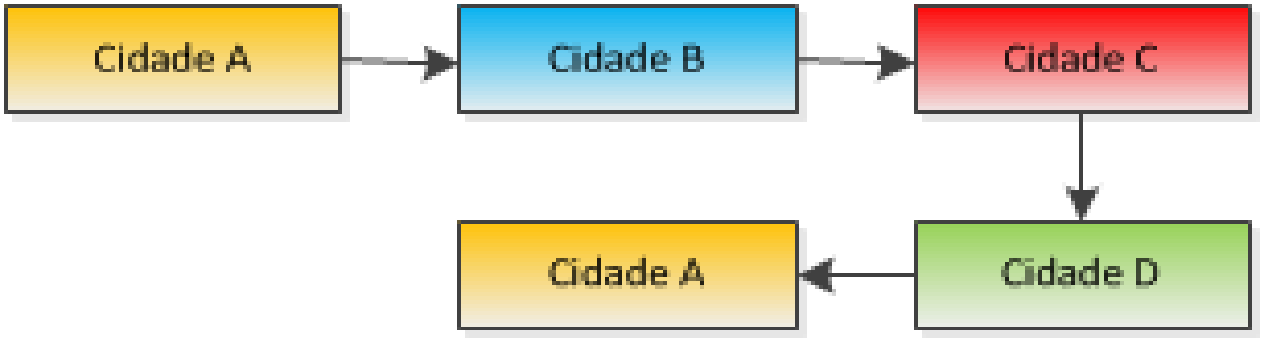
- Representação de Genes

0	1	2	3

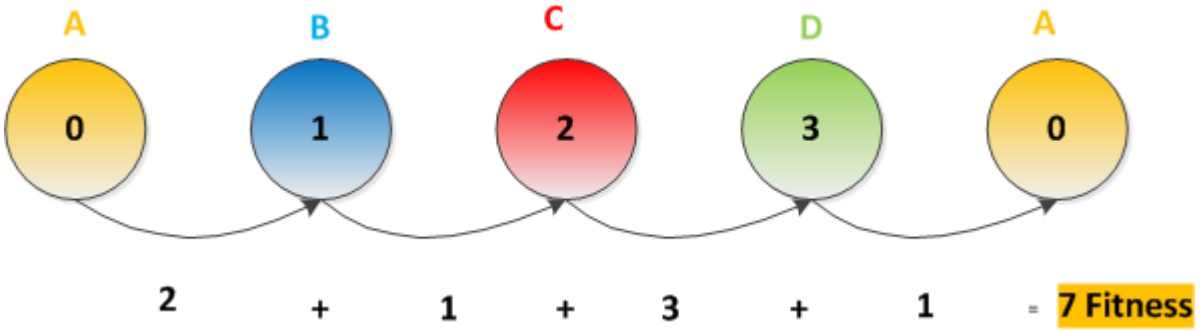
Cidade	A	B	C	D
Índex	0	1	2	3

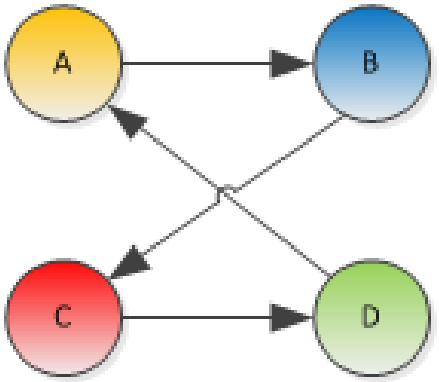
Caixeiro Viajante

- Ordem do caminho



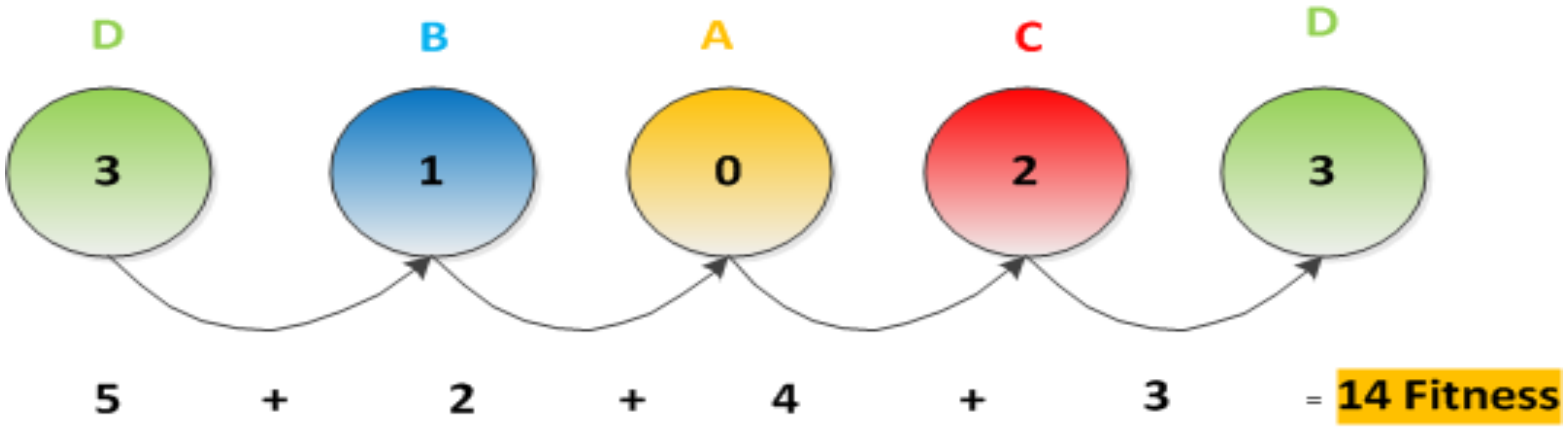
- Cálculo de Fitness
  - Exemplo1:

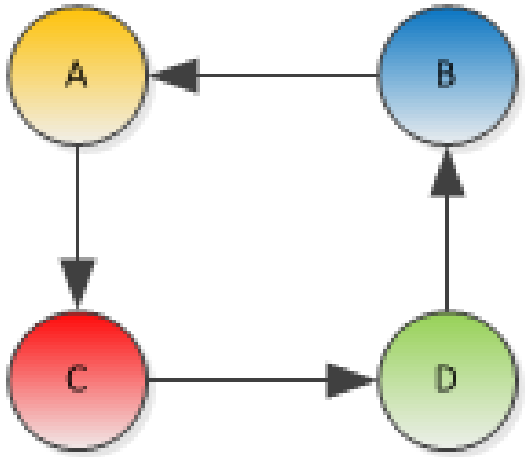




Caixeiro Viajante

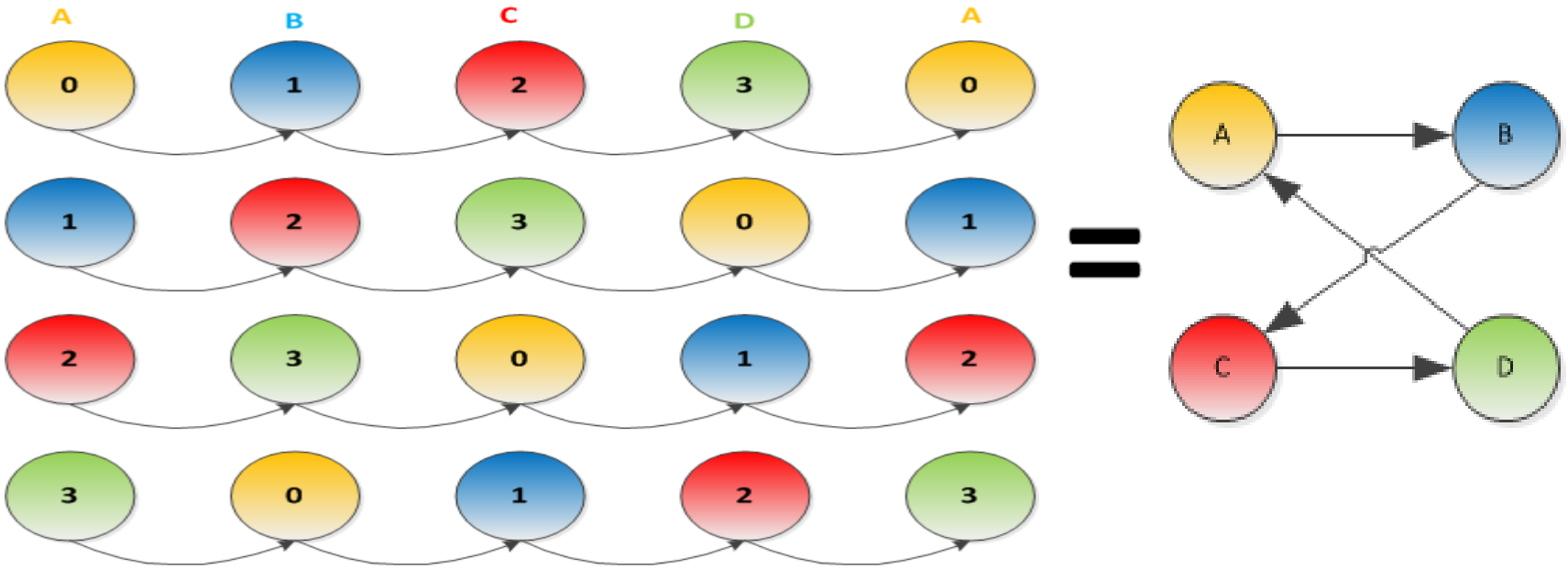
- Exemplo2:





Caixeiro Viajante

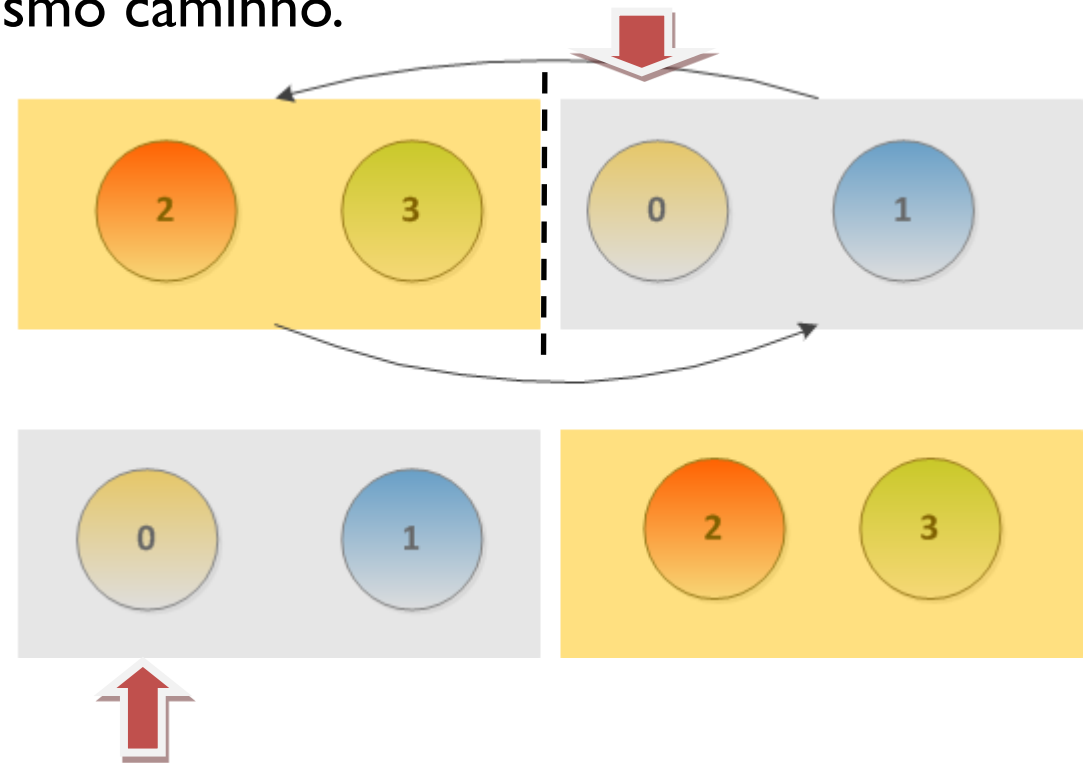
- Atenção:



  - São sempre o mesmo caminho, a única diferença é o ponto de partida.

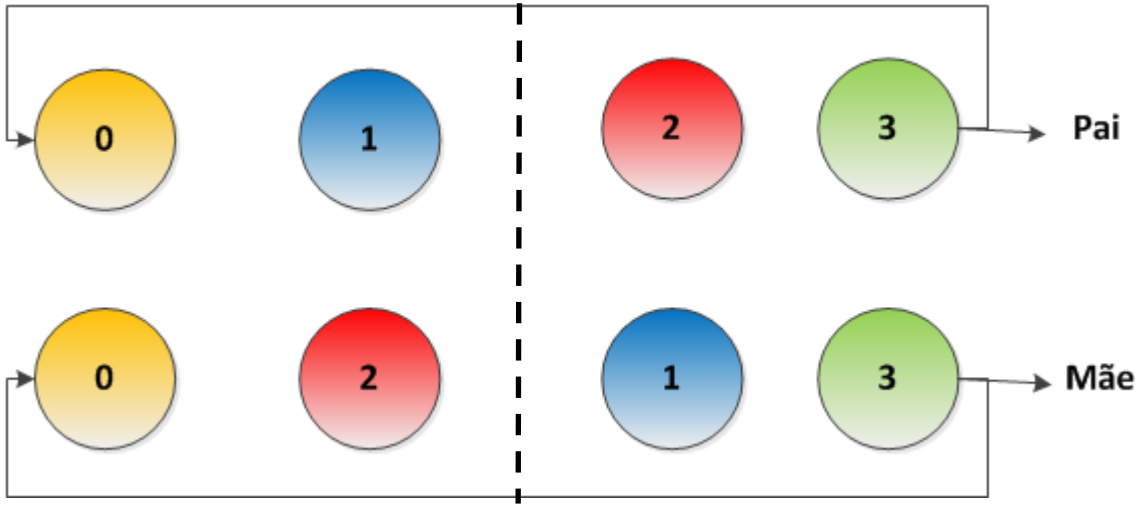
Caixeiro Viajante

- Reparar o problema anterior:
  - Pegar na cidade A e colocá-la como primeira cidade através de uma rotação, eliminando assim diferentes representações para o mesmo caminho.



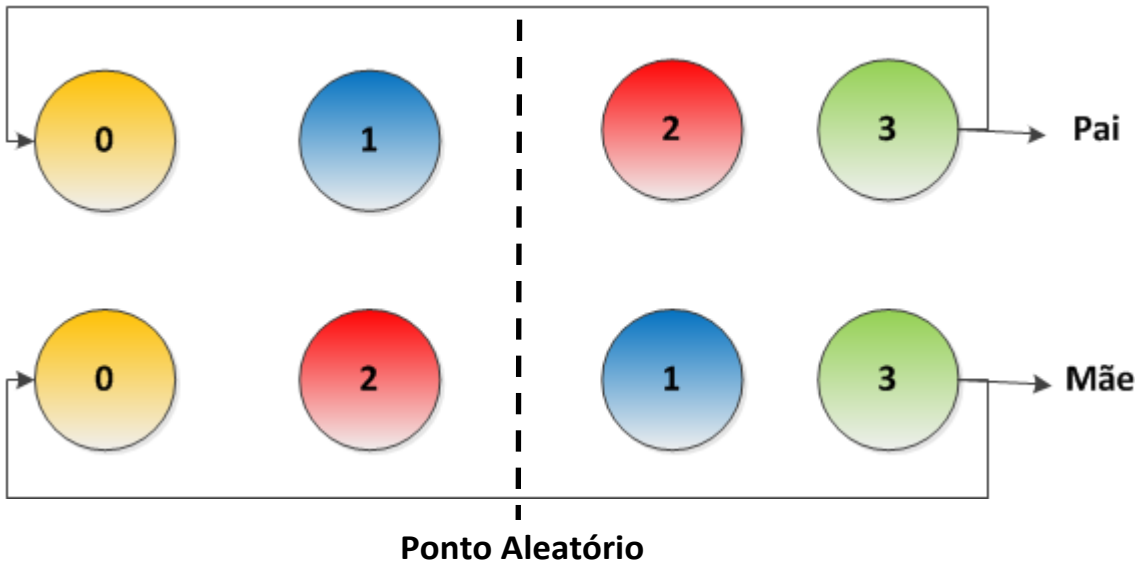
Operador: Cycle Crossover

- Descrição:
  - Uma parte do pai é mapeada para uma porção da mãe. A partir da porção substituída, o resto dos genes são preenchidos mas omitindo os genes já presentes e respeitando a ordem com que eles se encontram.
- Esquema:

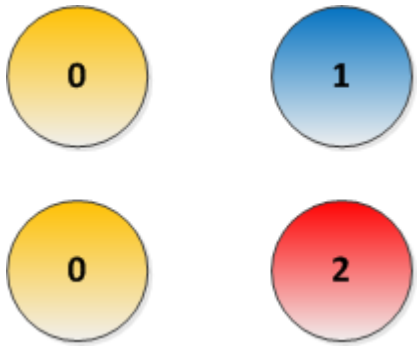


Operador: Cycle Crossover

- Esquema:



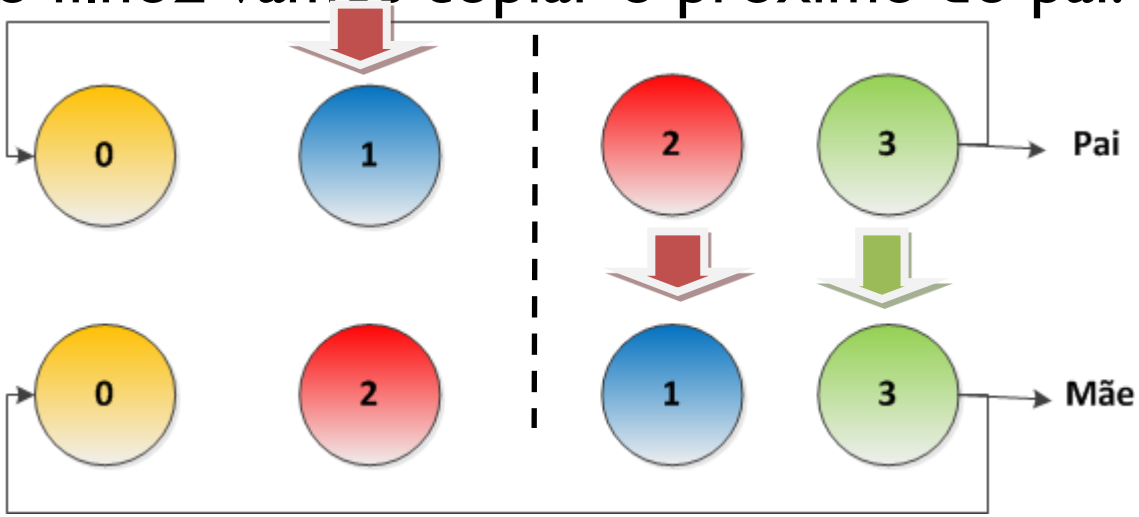
- **Filhos**

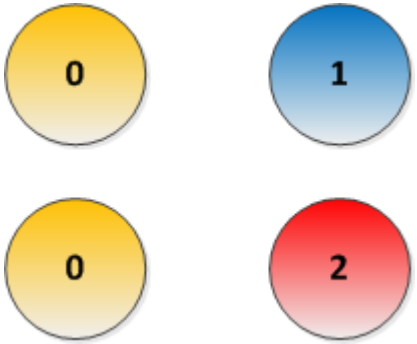


Os genes do lado esquerdo são iguais aos dos pais.

Operador: Cycle Crossover

- Para o filho1 vamos copiar o próximo gene da mãe, e para o filho2 vamos copiar o próximo do pai.

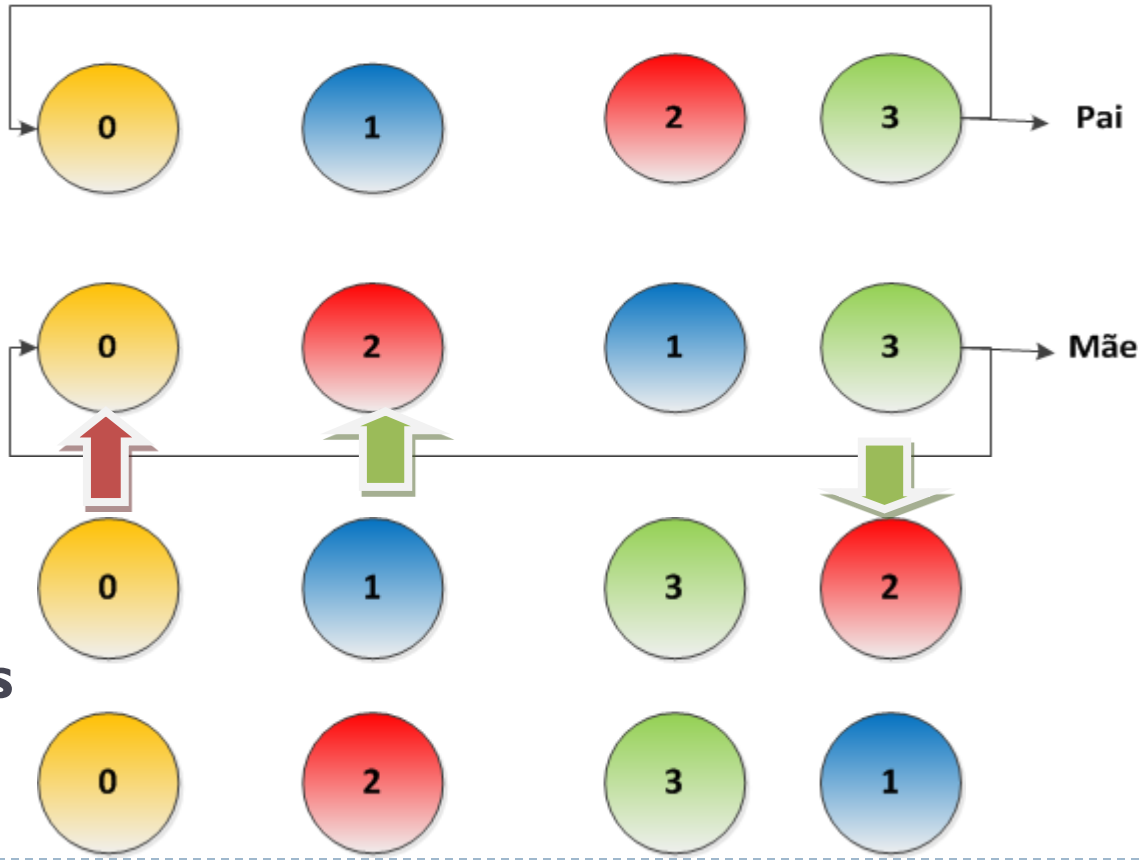




Como o filho1 já possuía o gene 1, não pode ficar com esse gene repetido, por isso coloca-se o próximo gene da mãe que não seja repetido.

Operador: Cycle Crossover

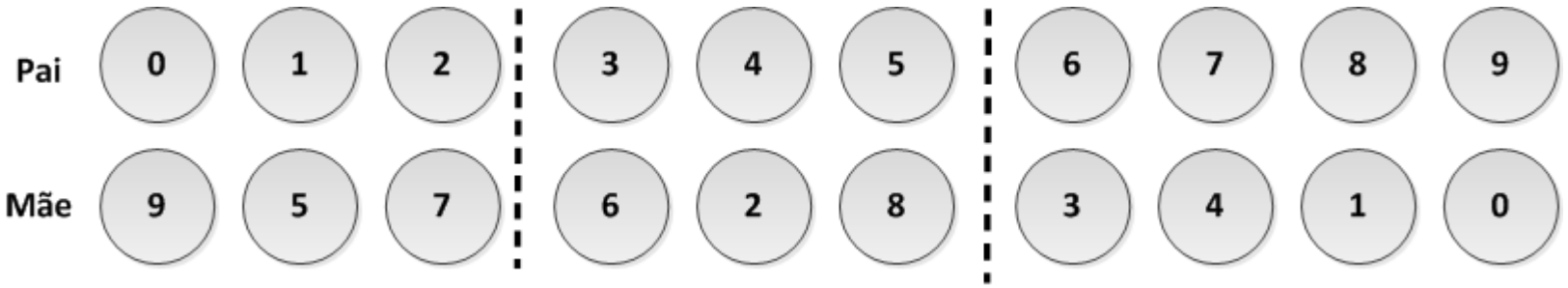
- Quando se chega ao fim do gene e ainda não colocámos todos os genes no filho, começamos no inicio do gene do pai e repetimos o processo até os filhos terem todos os genes.



- **Filhos**

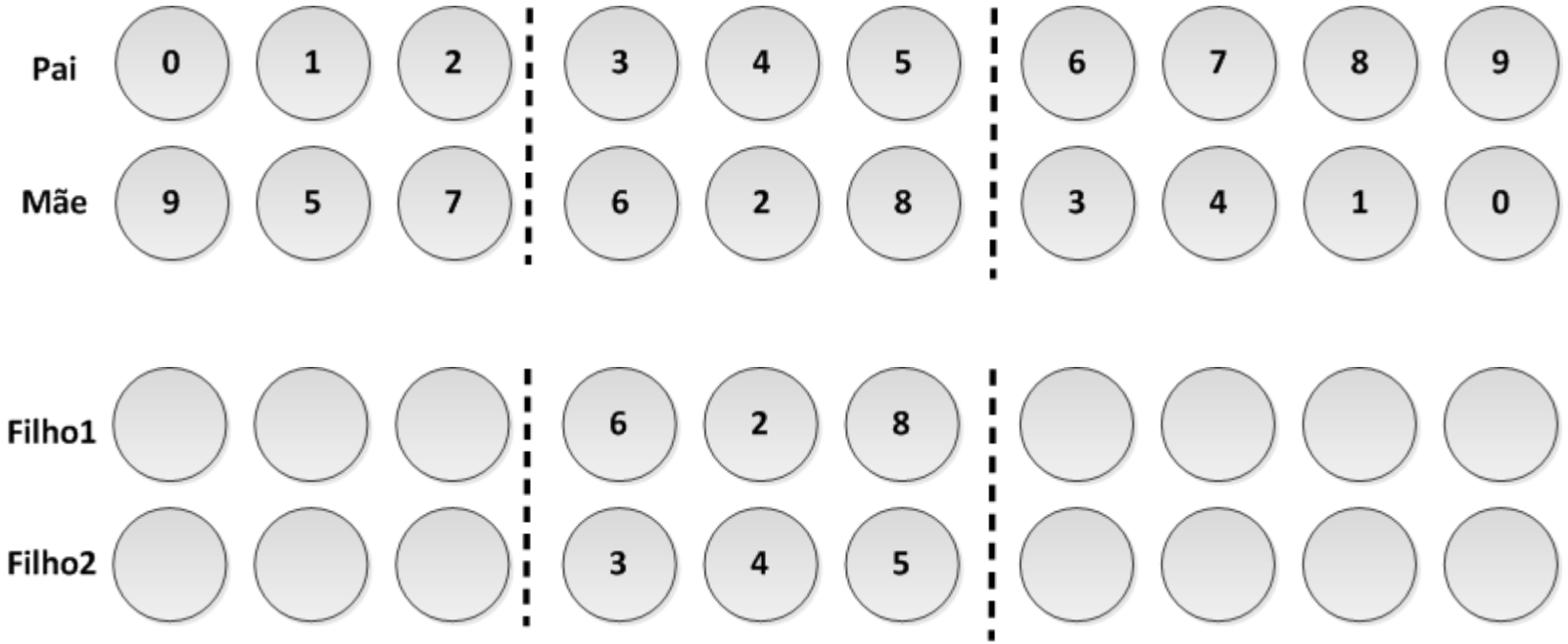
Operador: PMX - Crossover

- São gerados dois pontos de corte aleatórios e apartir desses pontos são mapeados os genes entre os pontos de corte do pai para a mãe e vice versa.
  - Passo 1: Gerar dois pontos aleatórios



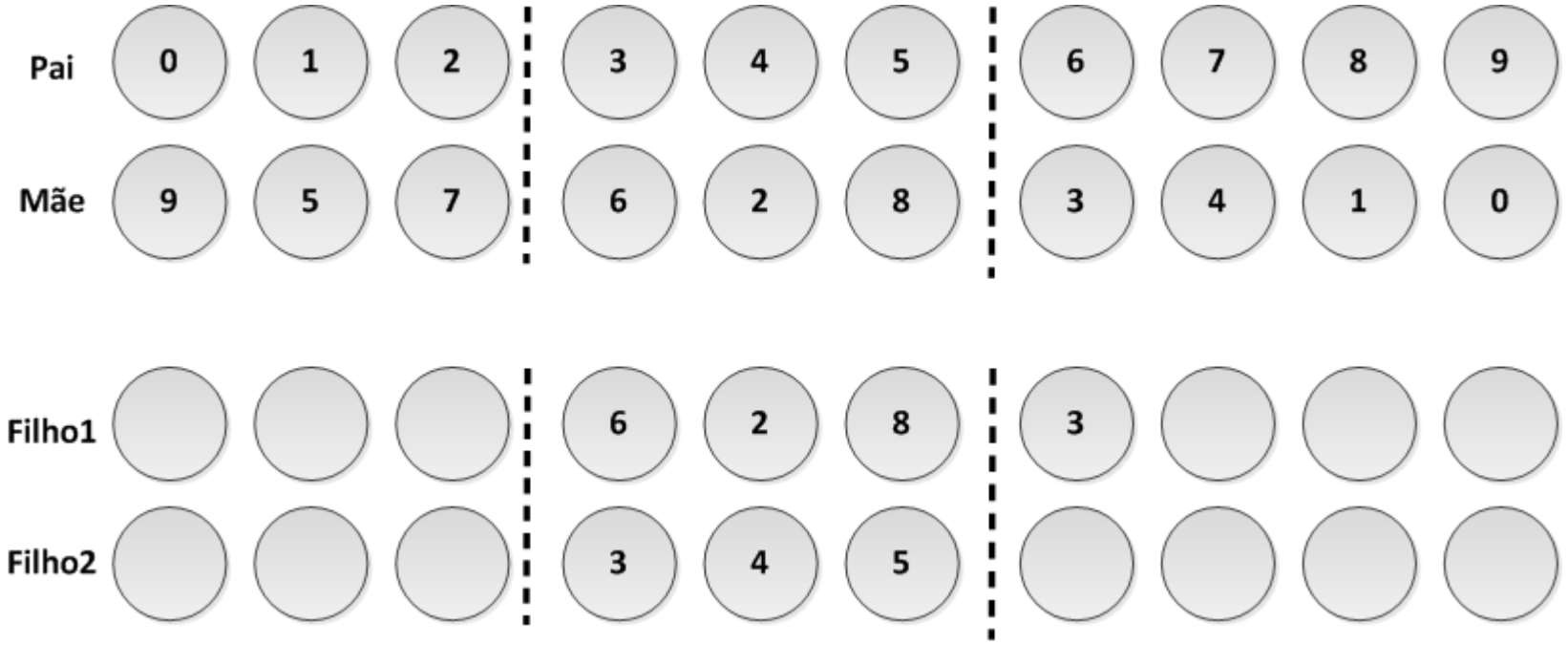
Operador: PMX - Crossover

- Passo 2: Os genes do pai entre os pontos ficam no filho2 e os da mãe no filho1.



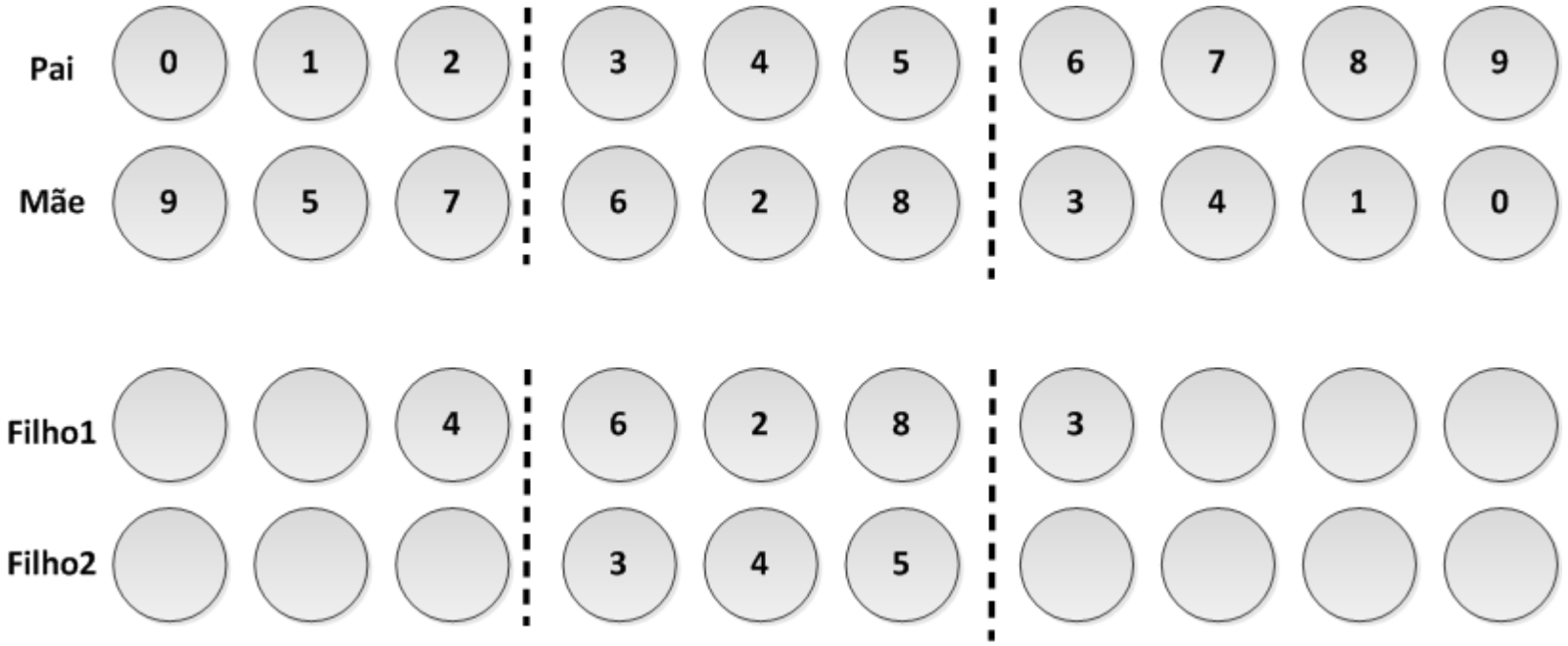
Operador: PMX - Crossover

- Passo 3: Para os genes que foram trocados coloca-se o gene pelo qual foi trocado, no sitio onde ele se encontrava anteriormente.
- Exemplo para o filho1: Troca o 6 pelo 3;



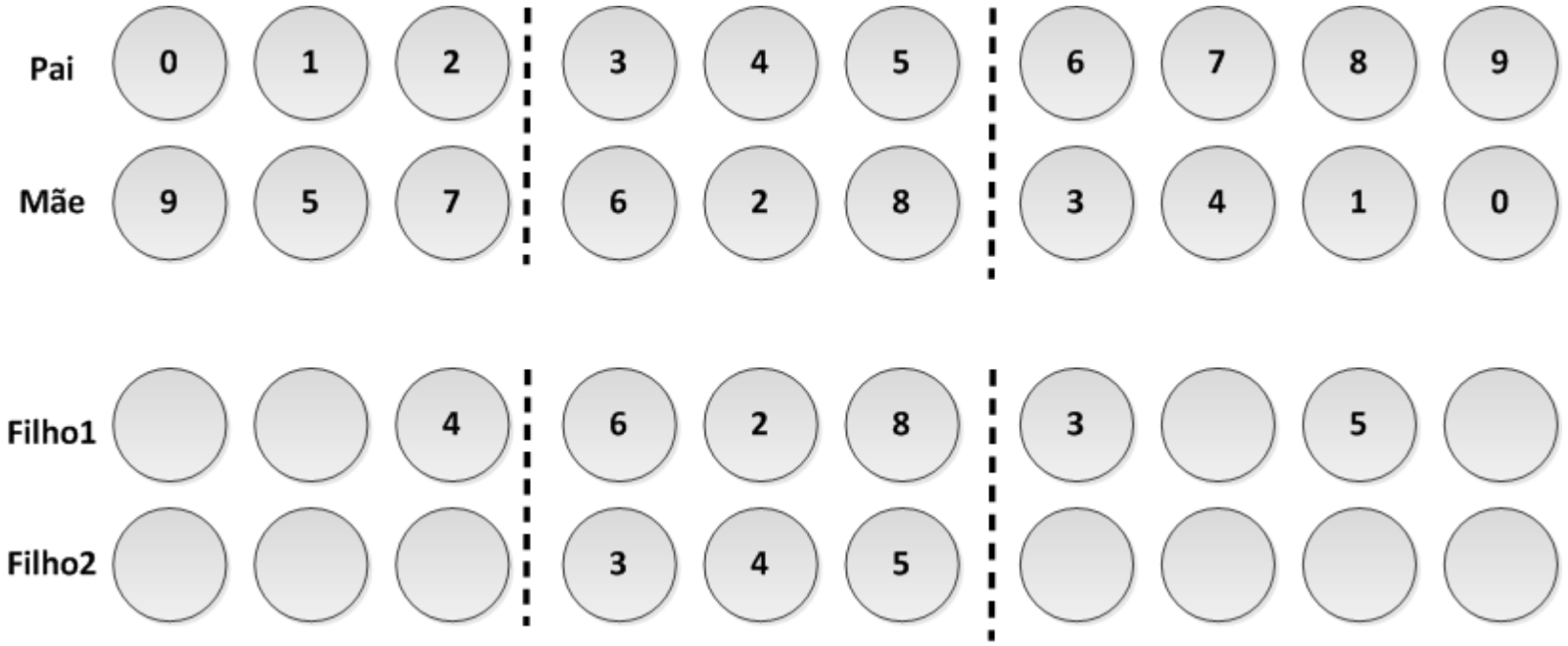
Operador: PMX - Crossover

- Troca o 2 pelo 4;



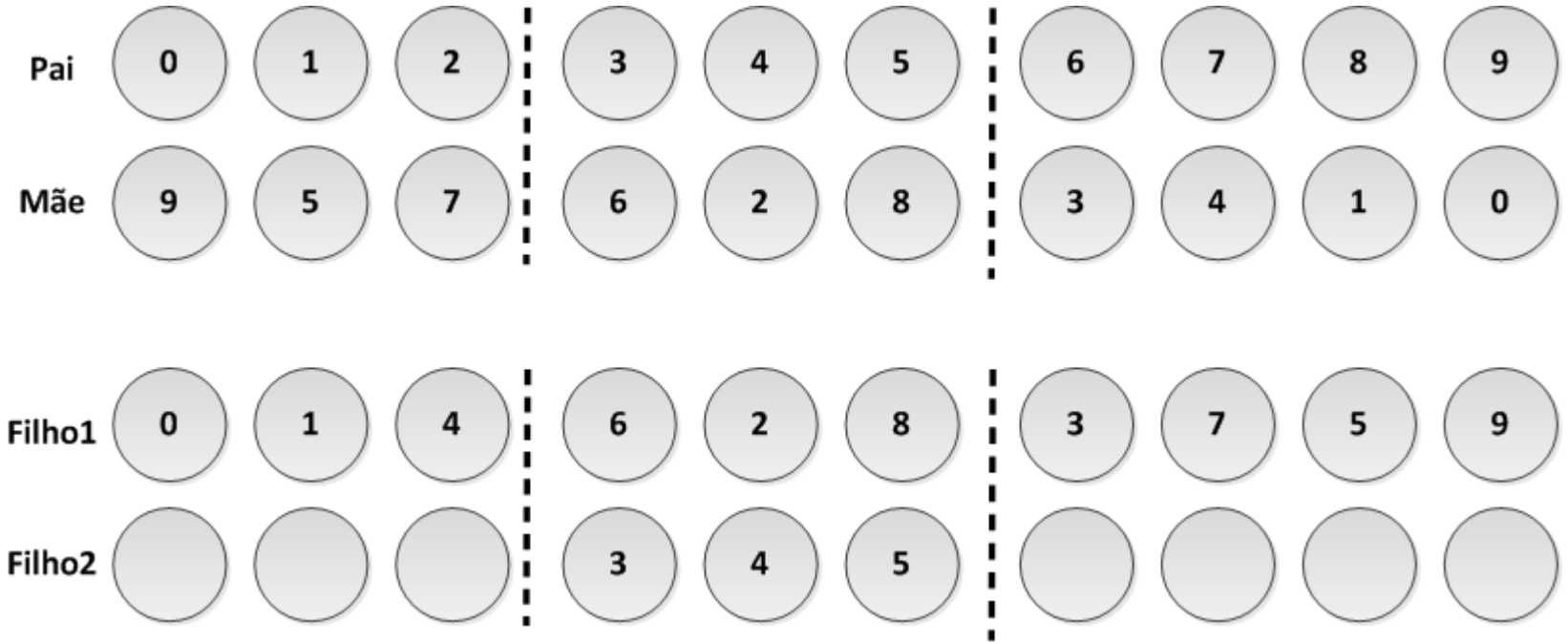
Operador: PMX - Crossover

- Troca o 8 pelo 5;



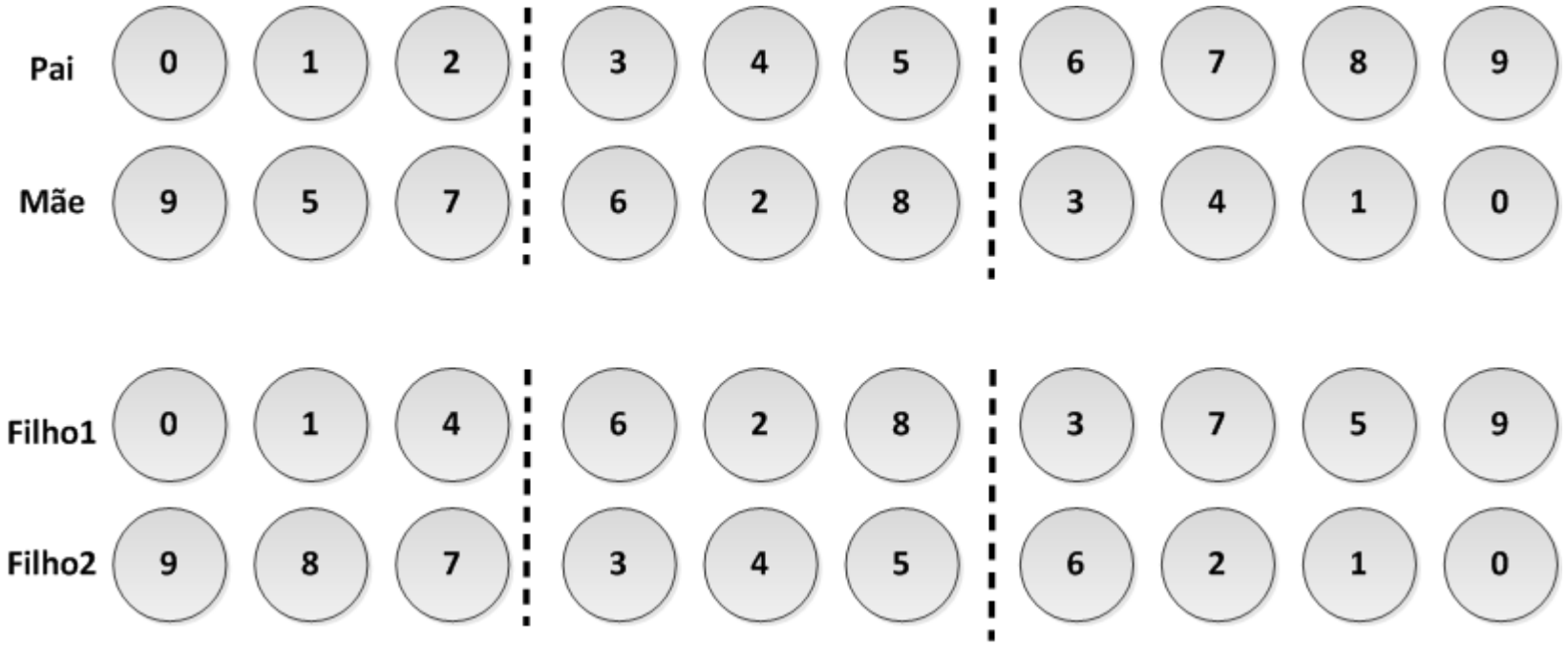
Operador: PMX - Crossover

- Os restantes genes são iguais aos do Pai.



Operador: PMX - Crossover

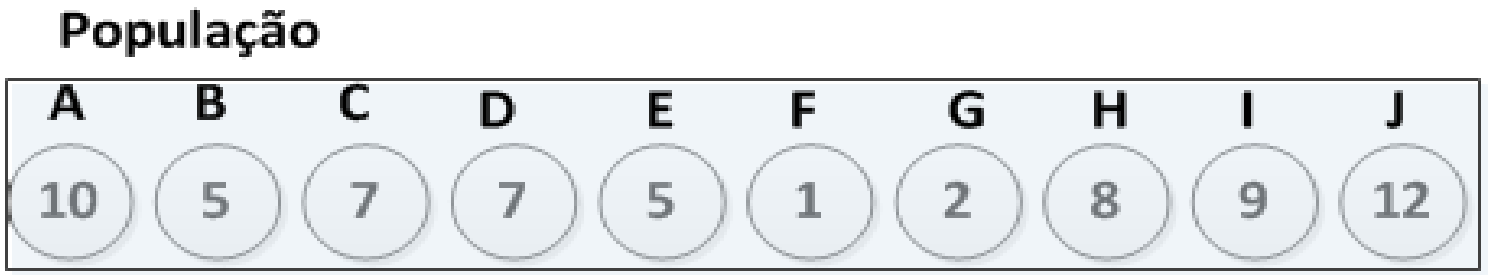
- Para o filho2:



Operador: SUS - Minimização

- Problema:
  - O operador SUS seleciona o fitness mais alto, por isso é preciso adaptá-lo para passar a selecionar o mais baixo e poder assim utilizá-lo no caixeiro viajante.

- Esquema:



  - Max(População)=12
  - Min(População)=1

Operador: SUS - Minimização

- PseudoCodigo : Calcular novo fitness

```
entra(Populaçao individuos[])

maxFitness := maiorFitness(individuos[])+1

para i := 0 ate i < tamanho(individuos[])
        individuos[i].fitness := maxFitness-individuos[i].fitness
        i := i+1

sai(Populaçao individuos[])


maiorFitness(individuos[])
        max := 0
        para i := 0 ate i < tamanho(individuos[])
                se (max < individuos[i].fitness) entao
                        max := individuos[i].fitness
                i := i+1

        retorna max
```

Operador: SUS - Minimização

- Novos valores de fitness:

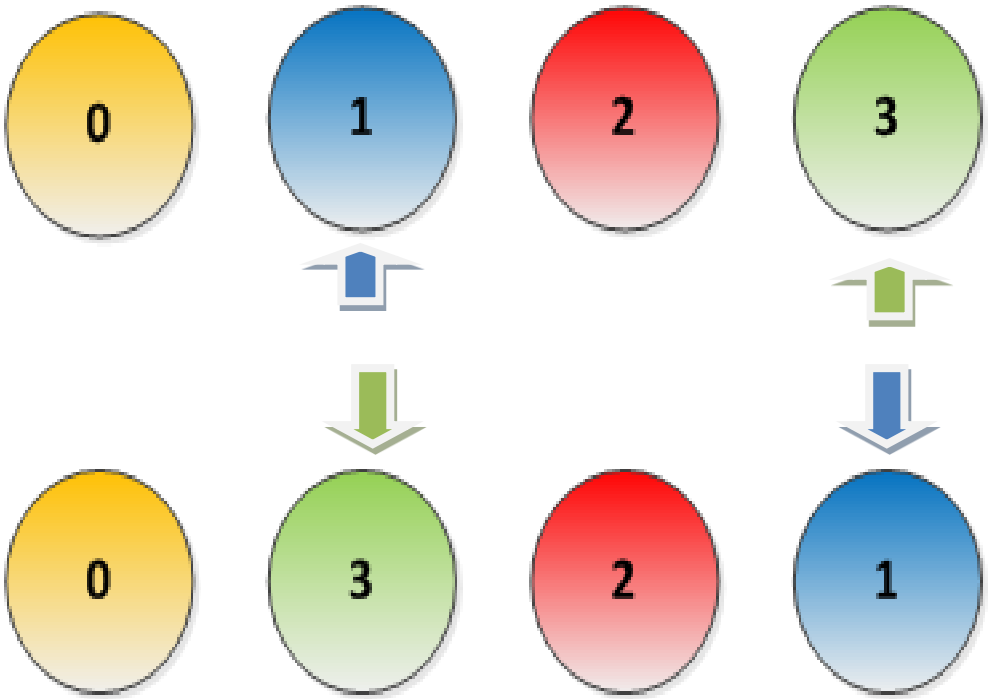
	A	B	C	D	E	F	G	H	I	J
Fitness	10	5	7	7	5	1	2	8	9	12
Novo Fitness	3	8	6	6	8	12	11	5	4	1

  - Novo Fitness = (12 + 1) - Fitness
  - Novo Fitness A = 13 – 10 = 3

- Fazendo esta conversão do fitness, o fitness com valor mais alto passa a ter o valor mais baixo, o que torna possível a utilização do operador SUS.

Operador: Swap Genes - Mutação

- Descrição:
  - Escolher dois alelos aleatórios de um indivíduo, e trocá-los de sítio.
- Esquema:



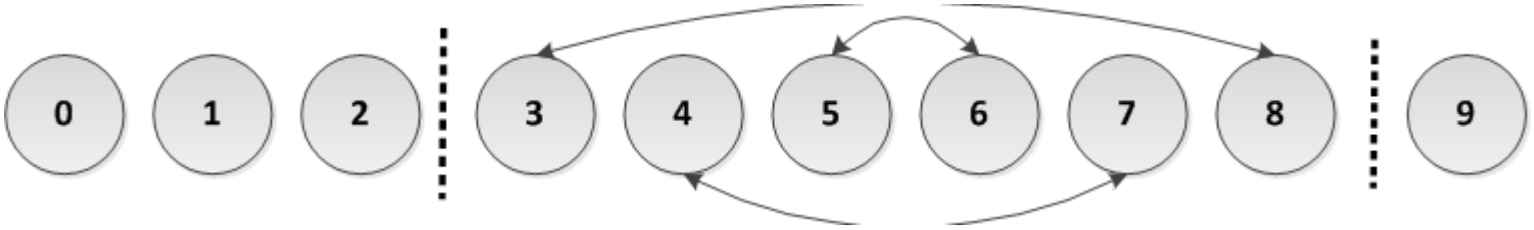
Operador: Inversion - Mutação

- Descrição: São gerados 2 pontos de corte aleatórios, e os genes entre os cortes são invertidos na sua ordem, e os restantes mantêm-se.

- Passo 1: Gerar 2 pontos aleatórios;

Operador: Inversion - Mutação

- Passo 2: Os genes dentro dos pontos de corte são invertidos;



Operador: Inversion - Mutação

- Resultado: