

## CRUZAMENTO DE GENES NO PROBLEMA DO CAIXEIRO VIAJANTE

Sejam dois indivíduos ou cromossomos (respostas) de uma população contendo cada um uma rota distinta. Vamos fazer um cruzamento dos genes da posição 4 até a posição 6 (**Fig. 1**).

Posicao	1	2	3	4	5	6	7	8	9	10
<i>genesPai1</i>	9	8	4	5	7	6	1	3	2	10
<i>genesPai2</i>	8	7	1	2	3	10	9	5	4	6

**Fig. 1 - Dois vetores de indivíduos (rotas) pais que se cruzarão da posição 4 a 6.**

Temos um mapeamento de  $5 \Leftrightarrow 2$ ,  $7 \Leftrightarrow 3$ , e  $6 \Leftrightarrow 10$ . Os novos filhos gerados trocarão os genes da coluna 4 até 6.

Somente trocar não resulta em uma rota correta. Seja, por exemplo, um Filho Doente (geneticamente) que recebesse os genes de cor verde do segundo pai, mantendo as demais características do primeiro pai (**Fig. 2**).

Filho Doente	9	8	4	2	3	10	1	3	2	10
							ou	5	4	6

Os genes 3, 2 e 10 (cidades) se repetem.

**Fig. 2 - Os genes 3,2 e 10 são repetidos**

A solução é trocar os genes repetidos pelo seu gene associado. Na **Fig. 1**, vemos que existe o mapeamento  $5 \Leftrightarrow 2$ ,  $7 \Leftrightarrow 3$ , e  $6 \Leftrightarrow 10$ , vamos trocar 2 por 5, 3 por 7 e 10 por 6, consertando o problema.

Do mesmo modo, no segundo filho, os genes 5, 7 e 6 se repetirão. A correção é semelhante.

Logo, temos então, os novos indivíduos (rotas) *genesFilho1* e *genesFilho2* (**Fig. 3**).

<i>genesFilho1</i>	9	8	4	2	3	10	1	7	5	6
<i>genesFilho2</i>	8	3	1	5	7	6	9	2	4	10

**Fig. 3 - Filhos resultantes do cruzamento**

O ideal é ter um vetor que aponte para a posição onde a cidade está para que se possa localizar rapidamente a cidade no vetor de rotas. Assim, conforme a **Fig. 1**, no *genesPai1* podemos ter um vetor  $PosGene1[1]=7$ ,  $PosGene1[2]=9$ , isto é, a cidade 1 está na 7a posição, a cidade 2 está na 9a posição. Isso ajuda na localização rápida da cidade.

Assim,  $PosGene1[genesPai1[i]] = i$ , onde geralmente  $genesPai[i] \neq i$ . Exemplo,  $genesPai1[2]=8$ , o 2o elemento de *genesPai1* é 8, e  $PosGene1[8]=2$ , o elemento 8 está na 2a posição.

Algoritmicamente:

para  $i$  de 1 até TamanhoIndividuo

$PosGene[genesPai[i]] \leftarrow i$

O processo de geração de um novo filho pode ser visto como o algoritmo a seguir onde temos os índices  $j$  e  $k$  para especificar a sequência de troca.

Por simplicidade é mostrado um filho, a geração dos dois filhos são semelhantes.

```
Para p de j até k
  seqTroca[i]=genesPai[p]
```

```
para p de 1 ate j-1
  genesFilho[p]=genesPai[p]
```

```
i=1
```

```
Para p de j até k
  genesFilho[p] = seqTroca[i++]
```

```
Para p de k+1 até tamanho(genesPai)
  genesFilho[p]=genesPai[p]
```

Para o *genesFilho1* doente podemos ver o seguinte algoritmo. A *seqTroca2* fara parte de *genesFilho1* e vice versa. Existe um mapeamento *seqTroca1* com *seqTroca2*, conforme comentado mais acima. O procedimento é recíproco para *genesFilho2*.

```
Para i de 1 ate tamanho(seqTroca2)
  g=seqTroca2[i]
  gMap = seqTroca1[i]
  se posGene1[g] < j ou posGene1[g] > k
    genesFilho1[posGene1[g]] =gMap //troca o valor repetido com o pareado
```

Outro cuidado a ser tomado, é que a sequência 8,9,7,6,5,2,3,1,4 é o mesmo que 6,5,2,3,1,4,8,9,7, já que o problema do caixeiro implica em sair de uma origem e voltar ao destino. É bom verificar estes casos, onde uma simples operação de rotação em uma lista resolveria.

Esta proposta não cobre tudo. Mais problemas podem surgir, que aqui não foram previstos.