

## Exemplo 2 (documento de apoio: 107-110) – implementação de um Algoritmo Genético

### 1. Definição do Cromossoma

**Cromossoma** de tamanho 4, em que cada gene representa um Trabalhador e o valor do gene indica qual a Tarefa que lhe ficou afeta.

Exemplo:

4	3	1	2
---	---	---	---

Este cromossoma representa a seguinte solução:

- Trabalhador 1 realiza a Tarefa 4
- Trabalhador 2 realiza a Tarefa 3
- Trabalhador 3 realiza a Tarefa 1
- Trabalhador 4 realiza a Tarefa 2

O custo desta solução é:  $17 + 24 + 17 + 21 = 79$ .

#### Nota

Em alternativa, o cromossoma podia ser definido por:

- cada gene representa uma Tarefa e o valor do gene indica qual o Trabalhador que a realiza.

Considerando a solução:

- Trabalhador 1 realiza a Tarefa 4
- Trabalhador 2 realiza a Tarefa 3
- Trabalhador 3 realiza a Tarefa 1
- Trabalhador 4 realiza a Tarefa 2

a codificação será: 

3	4	2	1
---	---	---	---

### 2. Como gerar uma solução admissível inicial?

Através da geração de uma permutação de n°s inteiros entre 1 e 4.

Em python, a função **permutations** da biblioteca **itertools**, permite gerar todas as permutações de uma lista de números.

#### Código

```
from itertools import permutations
import pandas as pd
import random
trabalhadores = list(range(1, 5))
permutation_list = list(permutations(trabalhadores))
print(permutation_list)
```

## Exemplo 2 (documento de apoio: 107-110) – implementação de um Algoritmo Genético

### OUTPUT

```
[ (1, 2, 3, 4), (1, 2, 4, 3), (1, 3, 2, 4), (1, 3, 4, 2), (1, 4, 2, 3),  
(1, 4, 3, 2), (2, 1, 3, 4), (2, 1, 4, 3), (2, 3, 1, 4), (2, 3, 4, 1),  
(2, 4, 1, 3), (2, 4, 3, 1), (3, 1, 2, 4), (3, 1, 4, 2), (3, 2, 1, 4),  
(3, 2, 4, 1), (3, 4, 1, 2), (3, 4, 2, 1), (4, 1, 2, 3), (4, 1, 3, 2),  
(4, 2, 1, 3), (4, 2, 3, 1), (4, 3, 1, 2), (4, 3, 2, 1) ]
```

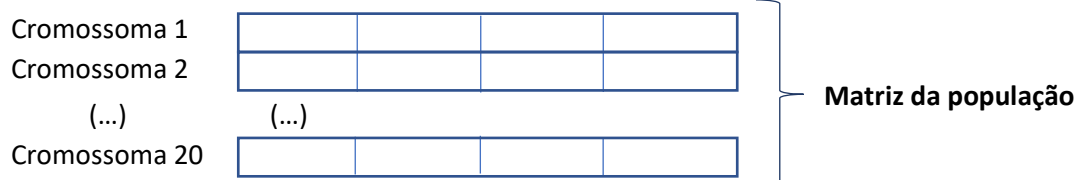
### 3. Geração da População Inicial

Qual deve ser a dimensão da população?

- A dimensão da população deve ser igual em todas as iterações do Algoritmo Genético. A dimensão da população deve assegurar a sua diversidade. Não existe uma regra para definir o valor da dimensão da população. Devem ser testados vários valores para a dimensão da população. Como este problema tem  $4 \times 3 \times 2 \times 1 = 4! = 24$  soluções admissíveis, podemos, **por exemplo**, considerar que a população é constituída por 20 cromossomas/ soluções.

Em que estrutura se deve guardar a população?

- A população pode ser guardada numa matriz em que cada linha corresponde a um cromossoma/ solução:



### Código

```
df_permutations = pd.DataFrame(permutation_list, columns=trabalhadores)
print(df_permutations)
```

### OUTPUT

	1	2	3	4
0	1	2	3	4
1	1	2	4	3
2	1	3	2	4
3	1	3	4	2
4	1	4	2	3
5	1	4	3	2
6	2	1	3	4
7	2	1	4	3
8	2	3	1	4
9	2	3	4	1
10	2	4	1	3
11	2	4	3	1
12	3	1	2	4
13	3	1	4	2

14	3	2	1	4
15	3	2	4	1
16	3	4	1	2
17	3	4	2	1
18	4	1	2	3
19	4	1	3	2
20	4	2	1	3
21	4	2	3	1
22	4	3	1	2
23	4	3	2	1

**Exemplo 2 (documento de apoio: 107-110) – implementação de um Algoritmo Genético**

**4. Como seleccionar os cromossomas pais?**

Existem quatro processos de selecção:

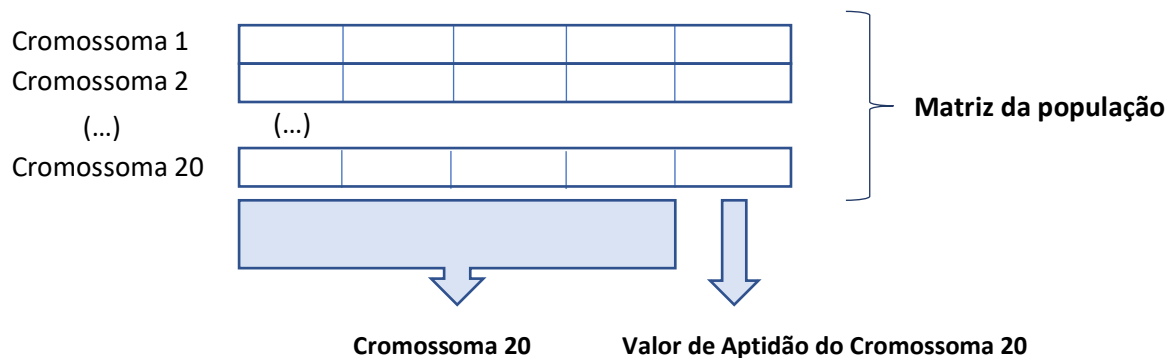
- i. Selecção Proporcional à Aptidão;
- ii. Selecção por Torneio;
- iii. Selecção por Ordenação;
- iv. Selecção Aleatória.

**Por exemplo**, podemos optar pela **Selecção por Torneio**:

- Atribui-se um valor inteiro a **k** ( $k < \text{dimensão da população}$ ). **Por exemplo**,  $k=3$ .
- Seleccionam-se aleatoriamente **k** cromossomas e o **Pai1** é o cromossoma com melhor valor de aptidão;
- Seleccionam-se aleatoriamente **k** cromossomas e o **Pai2** é o cromossoma com melhor valor de aptidão;

Nota

Atendendo à necessidade de conhecer o Valor de Aptidão de cada cromossoma da população, faz sentido considerar mais uma coluna na matriz da população, onde é registado o Valor de Aptidão de cada cromossoma:



**5. Tipo de Crossover a realizar?**

Os operadores de Crossover mais utilizados são:

- i. Crossover a um ponto;
- ii. Crossover a  $k=2$  pontos;
- iii. Crossover uniforme.

**Por exemplo**, podemos optar pelo **Crossover a um ponto**:

- Considera-se que a probabilidade de crossover é igual a 1;
- Gera-se um  $n^\circ$  aleatório, **m**, entre 1 e 3 ( $N^\circ \text{ de Tarefas} - 1 = 3$ );
- Cada cromossoma Pai é dividido em duas caudas:
  - Cauda esquerda com os genes 1: **m**;
  - Cauda direita com os genes **m+1**:  $N^\circ \text{ de Tarefas}$
- Cada filho irá ficar com uma das caudas dos cromossomas Pai.

**Exemplo 2 (documento de apoio: 107-110) – implementação de um Algoritmo Genético**

Exemplo de Crossover a um ponto

Pai1:	3	1	2	4
Pai2:	4	2	1	3

Considerando  $m = 3$ , geram-se os seguintes filhos:

Filho1:	3	1	2	3
Filho2:	4	2	1	4

Os cromossomas filho são não admissíveis! O Filho1 corresponde a uma solução em que a tarefa 3 é realizada pelos trabalhadores 1 e 4 e em que a tarefa 4 não é realizada. Enquanto na solução associada ao Filho2, a tarefa 4 é realizada pelos trabalhadores 1 e 4 e a tarefa 3 não é realizada.

**6. Como lidar com os cromossomas filho não admissíveis?**

Se os cromossomas filho são não admissíveis podemos optar por uma de duas abordagens:

- **Abordagem 1:** Desenvolver um procedimento que torne o cromossoma filho admissível.

Podemos definir um procedimento simples para tornar um cromossoma filho admissível:

- Identificar a tarefa que **não está a ser realizada**;
- Identificar a tarefa que **está a ser realizada por dois trabalhadores**;
- No primeiro gene com o valor da tarefa que é realizada por dois trabalhadores, altera-se o valor para a tarefa que não estava a ser realizada.

Aplicação do procedimento ao cromossoma Filho2:

4	2	1	4
---	---	---	---

- Identifica-se a tarefa que não é realizada: 3;
- Identifica-se a tarefa que se repete: 4;
- Atribui-se a tarefa 3 ao 1º gene com a tarefa 4.

Cromossoma admissível gerado (a partir do Filho2):

3	2	1	4
---	---	---	---

com  $custo = 12 + 30 + 17 + 28 = 87$ .

- **Abordagem 2:** Desenvolver uma função de avaliação que contabilize não só o custo da afetação das tarefas aos trabalhadores, mas também uma parcela adicional que penalize a avaliação da solução, por ser inadmissível.

➤ **Função de Penalização** para cada *Tarefa i*, dada por

$$P \times \max c_{ij} \times (n^{\circ} \text{vezes que a tarefa } i \text{ foi realizada} - 1),$$

em que  $P$  é um fator de penalização, previamente fixo.

Exemplo 2 (documento de apoio: 107-110) – implementação de um Algoritmo Genético

Aplicação da Função Penalização ao cromossoma Filho2:

4	2	1	4
---	---	---	---

Função Penalização para a Tarefa 4 (considerando  $P=2$ ):

$$P \times \max c_{4j} \times (n^{\circ} \text{ vezes que a tarefa 4 foi realizada} - 1) = \\ = 2 \times 33 \times (2 - 1) = 66$$

Valor de Aptidão para o Filho2:

$$\text{custo de afetação} + \text{penalização} = 17 + 30 + 17 + 28 + 66 = 158.$$

O cromossoma Filho 2 poderá fazer parte da população, mas terá um valor de aptidão pouco aliciante (158).

## 7. Como é realizada a substituição da População?

Os dois modelos mais utilizados são:

- Modelo Estacionário (em cada iteração, substituir uma percentagem, previamente definida, da população “pais” pela mesma percentagem da população “filhos”).
- Modelo Geracional (em cada iteração, são gerados  $n$  descendentes, em que  $n$  representa o tamanho da população, e toda a população é substituída pelos seus descendentes).

Por exemplo, podemos optar pelo Modelo Estacionário.

## 8. Tipo de Mutação a realizar?

Os operadores de mutação mais utilizados são:

- Mutação Bit Flip;
- Mutação por Troca;
- Mutação por Inversão.

Por exemplo, podemos optar pelo operador **Mutação por Troca**:

- Considera-se a probabilidade de mutação ( $pm$ ) igual a 0.1 (dado que a mutação é um fenómeno raro,  $pm$  deve assumir um valor reduzido, por exemplo,  $pm = 0.1$ );
- Gera-se um  $n^{\circ}$  aleatório com distribuição Uniforme em  $[0,1]$ :  $u$ ;
- Se  $u < 0.1$  então seleccionam-se aleatoriamente dois genes e troca-se o valor dos seus alelos.

**Exemplo 2 (documento de apoio: 107-110) – implementação de um Algoritmo Genético**

**Aplicação do operador Mutação por Troca ao cromossoma**

3	1	2	4
---	---	---	---

- $pm = 0.1$  (por exemplo);
- Geração de um  $n^\circ$  aleatório com distribuição Uniforme em  $[0,1]$ :  $u = 0.02$  (por exemplo);
- Como  $u < 0.1$  então seleccionam-se aleatoriamente dois genes: por exemplo, gene 2 e gene 3;
- Troca-se o valor do gene 2 com o valor do gene 3.
- Cromossoma gerado: 

3	2	1	4
---	---	---	---

**9. Critérios de paragem?**

Os critérios mais usuais são:

- Terminar o algoritmo se a melhor solução conhecida para o problema não é melhorada há 3(ou 4, ...) iterações consecutivas;
- Terminar o algoritmo quando é atingido o  $n^\circ$  máximo de iterações (definido à priori);
- (...)

**10. Desenvolvimento do código**

O Jupyter Notebook [\*\*AG\\_exemplo2\*\*](#), disponível no Moodle, contem o código das seguintes etapas do Algoritmo Genético:

- Geração da população inicial e Cálculo do Valor de Aptidão de cada cromossoma.
- Seleção dos Pais através do Processo de Seleção por Torneio.
- Operação de Crossover a um Ponto e Tratamento dos Cromossomas Filho não admissíveis (abordagens 1).
- Operação de Mutação por Troca.