



# Algoritmos Genéticos

---

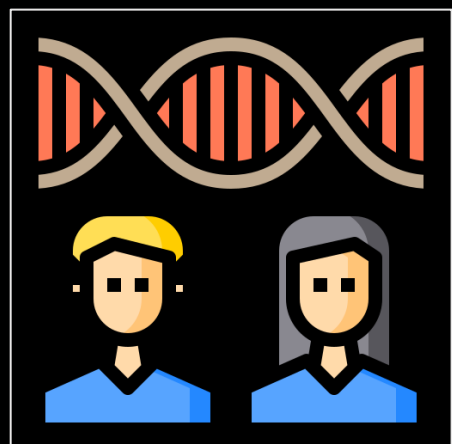
Trabalho de Grupo – Otimização Heurística

iscte

INSTITUTO  
UNIVERSITÁRIO  
DE LISBOA

Trabalho realizado por:

- Allan Kardec N°103380 CDB1
- André Plancha N°105289 CDB2
- Diogo Freitas N°104841 CDB1
- Ricardo Ângelo N°104826 CDB1
- Rui Chaves N°104914 CDB





## Contents

Enunciado.....	2
Alínea a).....	3
Tarefas Prioritárias .....	4
Alínea b).....	5
Alínea c).....	7
Alínea d).....	7
Alínea e).....	8
Alínea f).....	9
Alínea g).....	10
População inicial .....	11
Nova população.....	11
Função aptidão .....	12
Elitismo .....	12
Critérios de paragem .....	13
Algoritmo .....	13
Alínea h).....	15
Anexos.....	17
População Inicial .....	17
Criação das Gerações .....	18
Solução Final.....	19



## Enunciado

A *Química\_PT* é uma indústria do setor químico e está a planear a execução de dez tarefas independentes,  $T_1, T_2, \dots, T_{10}$ , em três máquinas idênticas,  $M_1, M_2, M_3$ , que funcionam em paralelo e sem restrições de disponibilidade.

A antiguidade das máquinas levou o gestor de produção da *Química\_PT* a definir o número de tarefas a alocar a cada uma das máquinas: 4, 3 e 3 tarefas, respetivamente, para as máquinas  $M_1, M_2$  e  $M_3$ .

Cada tarefa  $T_i, i=1, \dots, 10$ , deve ser processada uma única vez por uma das três máquinas existentes e o respetivo tempo de processamento,  $p_i$ , em horas, é apresentado na tabela seguinte:

Tabela 1 - Enunciado

Tarefas	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
Tempo de processamento, em horas, da Tarefa $T_i$ ( $p_i$ )	38	33	36	20	32	29	46	38	34	40

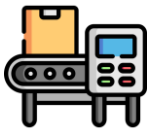
Dada a compatibilidade das tarefas, não será necessário contabilizar tempo entre a execução consecutiva de tarefas, quer seja para limpeza quer seja para mudança de tarefa.

Atendendo aos compromissos assumidos, a *Química\_PT* irá iniciar o processamento das tarefas nas três máquinas no dia 1 de junho, às 8h, e tem como obrigatoriedade a conclusão das tarefas  $T_3, T_4$  e  $T_5$  até ao dia 5 de junho, às 8h.

A *Química\_PT* pretende:

- atribuir a execução das dez tarefas às três máquinas;
- determinar a sequência de processamento das tarefas em cada máquina;

que respeite a data de entrega acordada para as tarefas  $T_3, T_4$  e  $T_5$  e minimize o tempo que decorre entre a programação da primeira tarefa apresentada e a conclusão da última tarefa.



## Alínea a)

Descreva, por palavras, uma solução admissível para o problema da *Química\_PT*

Neste problema são apresentadas 10 tarefas que, para serem realizadas, têm de ser processadas em 3 máquinas diferentes. Estas máquinas funcionam em paralelo, sem restrições de disponibilidade. É importante reter que a máquina **M<sub>1</sub>** realizará 4 tarefas, enquanto as outras 2 apenas realizarão 3 tarefas.

Analisando os dados com atenção, é notório que existem vários pontos/restrições a serem respeitadas pela solução final. Após a leitura do enunciado, é importante perceber que, para que uma solução seja admissível, ela tem de respeitar as seguintes restrições:

- Todas as tarefas têm de ser processadas, independentemente da ordem;
- Cada tarefa só pode ser processada uma (1) única vez;
- As tarefas **T<sub>3</sub>**, **T<sub>4</sub>** e **T<sub>5</sub>** têm de ser realizadas antes do dia 5 de junho, às 08:00.

Como o processamento das tarefas começam no dia 1 de junho, às 08:00 também, a última restrição é equivalente a dizer que as tarefas têm de estar processadas 96 horas depois de começar o processo.

De seguida encontra-se um gráfico de barras com os tempos de execução das tarefas, notado em azul as tarefas **T<sub>3</sub>**, **T<sub>4</sub>** e **T<sub>5</sub>**, pois estas são as sujeitas a uma restrição especial extra.

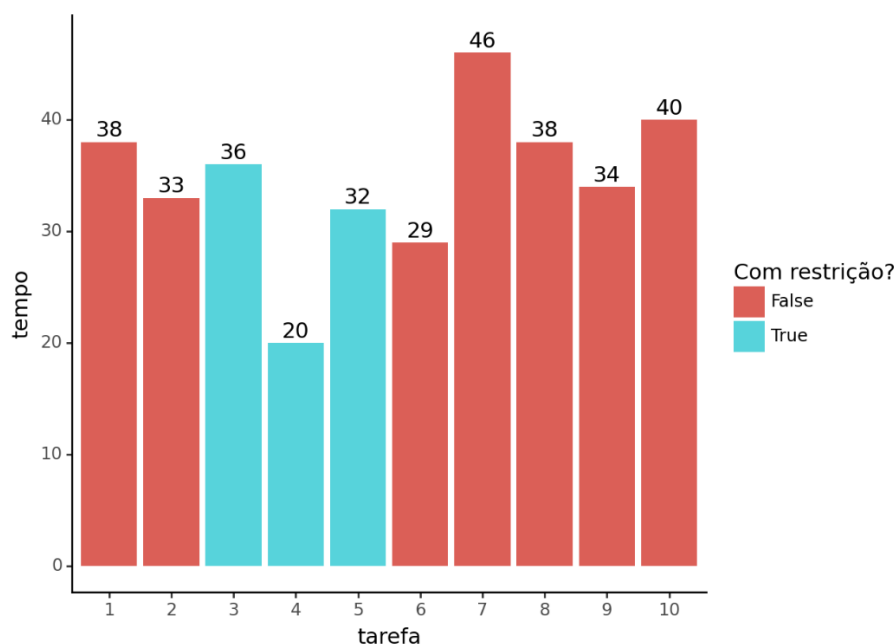


Figura 1: Tempos de cada uma das tarefas, com ou sem restrição

É notório que a tarefa com maior tempo de execução é a **tarefa T<sub>7</sub>**, possuindo um tempo de execução de 46 horas; enquanto a tarefa com menos horas de execução é a **tarefa T<sub>4</sub>**, sendo esta uma tarefa com a tal restrição especial, que demora 20 horas a ser executada.



Claramente, teremos como objetivo minimizar o tempo de processamento das tarefas e, para tal, será necessário distribuir as tarefas pelas máquinas de forma que não haja de execução excessivas, enquanto outras finalizam as tarefas muito rapidamente, sendo necessário arranjar um equilíbrio entre estas, pois estas 3 máquinas irão trabalhar ao mesmo tempo.

### Tarefas Prioritárias

A soma do tempo de execução das 3 tarefas com restrição, não ultrapassam as 96 horas ( $36 + 20 + 32 = 88 < 96$ ). Com esta ideia em mente, podemos retirar várias conclusões caso estas 3 tarefas fiquem numa única máquina:

1. Caso estas três tarefas estejam numa máquina que executa apenas 3 tarefas, independentemente da ordem destas, a restrição será sempre respeitada, pois o resultado será sempre inferior a 96, devido à propriedade comutativa da soma.
2. Caso estas três tarefas sejam colocadas na máquina  $M_1$ , a máquina escolhida para executar 4 tarefas, poderá ocorrer problemas, pois caso estas 3 tarefas sejam executadas depois de uma primeira, a restrição poderá não ser respeitada uma solução para isto seria trocar a ordem de execução destas que não influenciaria o tempo de execução total, a partir da técnica introduzida de seguida.

Imaginemos agora uma situação em que as tarefas com restrição não se encontram na mesma máquina, surgindo o problema de que estas não são finalizadas pela máquina em menos de 96 horas, não respeitando a restrição. Este problema pode ser resolvido, simplesmente mudando a ordem de execução das tarefas na máquina onde essas se encontram, obtendo uma solução adjacente com o mesmo o valor de objetivo, mas garantindo a sua admissibilidade.

Para demonstrar, consideremos a letra  $M_i$  como a máquina  $i$ , e  $T_j$  como a tarefa  $j$ . Na  $M_1$  temos associadas as tarefas  $T_1, T_2, T_3$ , e  $T_4$  (na ordem apresentada); na  $M_2$  as tarefas  $T_5, T_6$ , e  $T_7$ ; e na  $M_3$  com  $T_8, T_9$ , e  $T_{10}$ . Esta solução é inadmissível porque as tarefas  $T_3$  e  $T_4$  não são processadas antes das 96

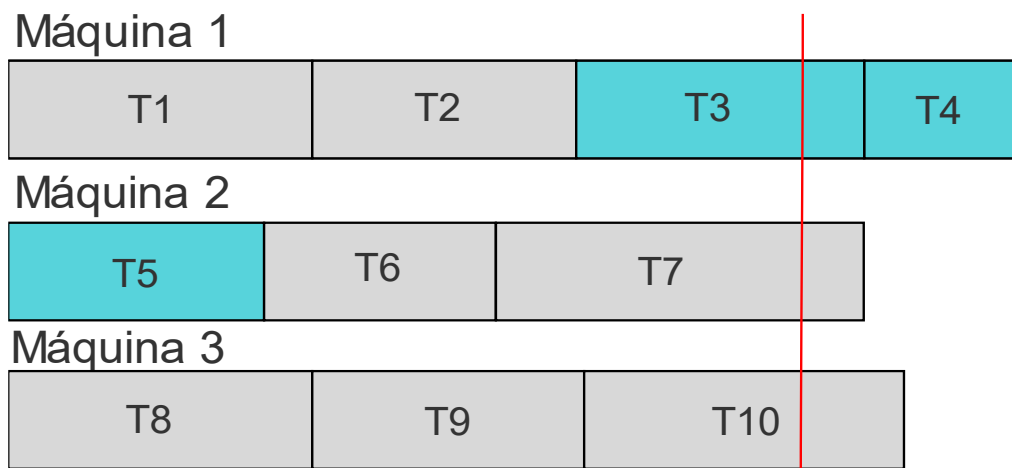


Figura 2: Exemplo de uma solução inadmissível



horas. Esta solução encontra-se representada na Figura 2, encontrando-se em azul as tarefas com tal restrição especial, alcunhadas de tarefas prioritárias. Na máquina  $M_1$  é notório que as tarefas prioritárias se encontram a ser executadas por último, sendo a ordem destas a seguinte:  $[T_1, T_2, T_3, T_4]$ .

Vamos então deslocar as tarefas com restrição para o início da máquina  $M_1$ , que por consequência reorganiza a ordem das tarefas para:  $[T_1, T_2, T_3, T_4] \rightarrow [T_3, T_4, T_1, T_2]$ . Esta nova solução é admissível, tendo o mesmo valor de execução total quando comparada com a sua ordem original/inicial, pois continua a possuir as mesmas tarefas, mas agora em ordens diferentes, respeitando, finalmente, a restrição inicial.

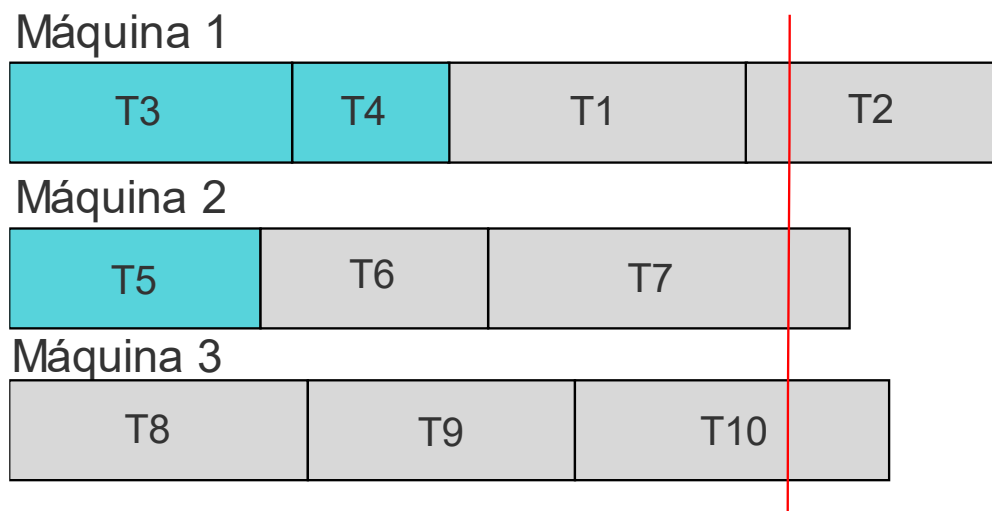


Figura 3: Solução corrigida

Este conceito será importante mais à frente, pois esta técnica será usada para corrigir soluções inadmissíveis.

Podemos então concluir que o tempo total de processamento das tarefas (o valor de objetivo/execução das tarefas) é apenas dependente das máquinas em que as tarefas se encontram associadas, e independente da ordem de processamento destas.

## Alínea b)

Desenvolva uma heurística para determinar uma sequência de processamento das tarefas em cada máquina que respeite a data de conclusão acordada para as tarefas  $T_3$ ,  $T_4$  e  $T_5$ . Com base na heurística desenvolvida, apresente uma solução admissível para o problema da *Química\_PT*.

Aqui será comentada a heurística construtiva que foi desenvolvida pelo grupo, onde esta será explicada passo a passo, de forma direta e simples:

1. De forma a respeitar, de imediato, a restrição das tarefas que têm de ser processadas em menos de 96 horas, associámos as respetivas 3 tarefas às primeiras tarefas a serem realizadas pelas



máquinas. Como a máquina  $M_1$  tem maior capacidade que as outras, associamos a tarefa crítica mais rápida, nomeadamente a **Tarefa 4** com 20 horas de execução;

2. De seguida, vinculamos as tarefas mais pequenas sem restrição (**Tarefa  $T_6$**  = 26 horas de execução, **Tarefa  $T_2$**  = 33 horas de execução e a **Tarefa  $T_9$**  = 34 horas de execução) à máquina que irá executar mais tarefas, sendo essa a máquina  $M_1$ ;

3. Por fim, das tarefas restantes atribuímos a com valor mais baixo e mais alto a máquina 2 e as restantes a 3, de forma a alocar uma espécie de equilíbrio nas máquinas.

De seguida será mostrado o resultado obtido e, ao mesmo tempo, será explicado como as tarefas serão agrupadas para os futuros cromossomas:

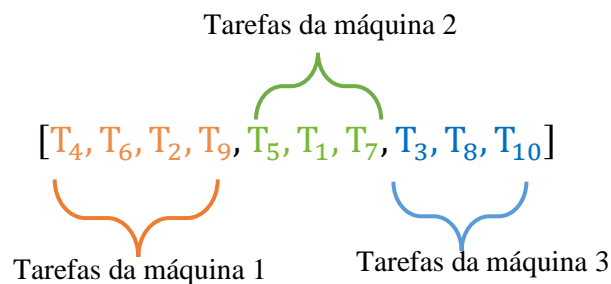


Figura 4: Solução da heurística + explicação do cromossoma

É importante referir que a ordem de processamento das tarefas é da esquerda para a direita. Isto significa que nesta solução, na máquina  $M_1$ , por exemplo, as tarefas a serem processadas serão as tarefas  $T_3, T_5, T_6$ , e  $T_9$ , na ordem referida.

Após criar o cromossoma com a heurística construtiva, o grupo criou um gráfico para uma melhor visualização dos tempos de conclusão de cada tarefa, a fim de poder retirar algumas conclusões (**Figura 5**)

Visualizando o gráfico, podemos então concluir que as tarefas foram concluídas em 116 horas, respeitando a restrição das tarefas  $T_3$  e  $T_4$  e  $T_5$ . Podemos também observar que as máquinas acabam relativamente ao mesmo tempo, demonstrando que esta heurística foi capaz de construir uma solução equilibrada, em termos de tempo de execução, entre as máquinas

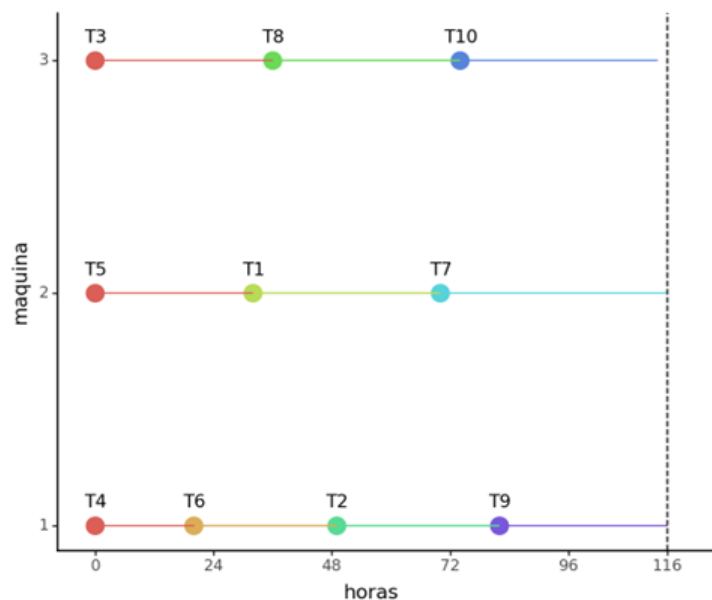
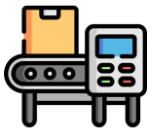


Figura 5 - Tempo de Execução das Tarefas da solução da heurística





## Alínea c)

Defina um cromossoma que permita codificar uma sequência de processamento das tarefas em cada máquina.

A definição do cromossoma foi introduzida na [alínea b\)](#). Os cromossomas irão estar codificados de forma semelhante àquilo que foi referido anteriormente, possuindo uma pequena diferença. Agora será apenas referenciado o respetivo número da tarefa, tornando-se assim uma lista de valores inteiros, como exemplo encontra-se de seguida a solução encontrada anteriormente.

[4, 6, 2, 9, 5, 1, 7, 3, 8, 10]

Caso seja realizada uma visualização rápida e descuidada do cromossoma, este aparenta ser uma representação de valores inteiros ordenadas numa única lista. É importante relembrar que este cromossoma representa 3 permutações, cada um por máquina  $\rightarrow [(4-6-2-9), (5-1-7), (3-8-10)]$ .

Apesar desta interpretação, podemos utilizar as propriedades originais da lista, assumindo que todas as tarefas estão conectadas, desta forma, facilitando o processo de [cruzamento \(Crossover\)](#) e [mutação](#) que serão referidos e explicados mais à frente.

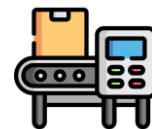
## Alínea d)

Tendo em conta a codificação que sugeriu em c), proponha um operador de crossover e exemplifique-o no contexto do problema da *Química\_PT*

Na escolha do operador de *Crossover*, chegou-se à conclusão de que alguns dos operadores mais usados iriam produzir solução não admissíveis, como o *crossover* uniforme, pois iriam facilmente produzir soluções com tarefas repetidas e, consequentemente, tarefas em falta. Para resolver este problema, o grupo optou por usar um operador dedicado para o problema do caixeiro-viajante (TSP), adaptando-o para este problema, chamado de [Operador de recombinação de arestas](#). O algoritmo, após a seleção de dois cromossomas (pai e mãe) da população, resume-se no seguinte:

1. Selecionar, aleatoriamente, entre o primeiro gene do pai ou o da mãe;
2. Identificar as ligações do gene escolhido (tendo em conta que a cabeça e a cauda estão conectadas);
3. Escolher uma ligação aleatoriamente (entre o último alelo do cromossoma com um alelo que ainda não esteja no cromossoma) e adicionar ao cromossoma, no último gene, o respetivo alelo novo;
4. Repetir a partir do passo 2 com o novo gene adicionado no passo 3;
  - a. Em caso de não haver novas ligações que respeitem os passos anteriores, adicionar aleatoriamente um gene com um dos alelos restantes.





Para uma melhor visualização do operador, iremos usar um exemplo:

- Cromossoma Pai: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
- Cromossoma Mãe: [4, 6, 2, 9, 5, 1, 7, 3, 8, 10]

Tabela 2 - Operador de Crossover

GENES	LIGAÇÕES	SELEÇÃO ALEATÓRIA	CROMOSSOMA FILHO
4	4 – 10, 4 – 3, 4 – 5, 4 – 6	4 – 10	[4, 10]
10	10 – 1, 10 – 9, 10 – 8	10 – 9	[4, 10, 9]
9	9 – 8, 9 – 2, 9 – 5	9 – 8	[4, 10, 9, 8]
8	8 – 3, 8 – 7	8 – 7	[4, 10, 9, 8, 7]
7	7 – 1, 7 – 3, 7 – 6	7 – 3	[4, 10, 9, 8, 7, 3]
3	3 – 2	3 – 2	[4, 10, 9, 8, 7, 3, 2]
2	2 – 6, 2 – 1	2 – 6	[4, 10, 9, 8, 7, 3, 2, 6]
6	6 – 5	6 – 5	[4, 10, 9, 8, 7, 3, 2, 6, 5]
5	5 – 1	5 – 1	[4, 10, 9, 8, 7, 3, 2, 6, 5, 1]

Com este operador de crossover, podemos então passar por vários cromossomas diferentes, a fim de conseguir descobrir várias soluções. É importante lembrar que o primeiro e o último gene do cromossoma não se encontra conectados como seria suposto, originalmente, pelo operador de crossover [Operador de recombinação de arestas](#).

## Alínea e)

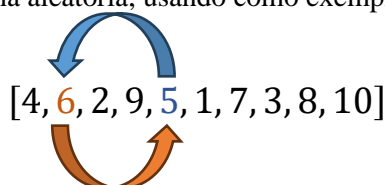
Tendo em conta a codificação que sugeri em c), proponha um operador de mutação e exemplifique-o no contexto do problema da *Química\_PT*.

O grupo propôs um algoritmo mais simples para o operador de mutação, que consiste na **mutação por troca**. Este operador tem como objetivo trocar de posição dois genes aleatórios, baseado numa probabilidade dada *a priori*.

O grupo definiu uma probabilidade de 10%, isto quer dizer que 1 em cada 10 cromossomas terão a posição de 2 genes trocados. Para uma melhor visualização deste operador de mutação, de seguida será mostrado um exemplo:

- Imaginando que o Cromossoma seguinte sofre uma mutação:
  - [4, 6, 2, 9, 5, 1, 7, 3, 8, 10]

Selecionado 2 genes de forma aleatória, usando como exemplo os genes com alelos 6 e 5, ocorreria o seguinte processo:





Realizada a mutação, o cromossomo passaria a ser da seguinte forma:

[4, 5, 2, 9, 6, 1, 7, 3, 8, 10]

### Alínea f)

Os operadores propostos em **d)** e **e)** garantem a obtenção de soluções admissíveis para o problema da *Química\_PT*? Caso não garantam, justifique que tipos de inadmissibilidades os operadores indicados podem gerar e indique como pode ultrapassar esta situação na implementação do algoritmo genético.

Tendo em conta o enunciado do trabalho, existem 2 tipos de soluções inadmissíveis que podem ocorrer relativamente às tarefas:

- **Problema 1:** No cromossoma final, poderão existir tarefas repetidas, fazendo com que nem todas as tarefas sejam realizadas.

Um exemplo de um cromossoma inadmissível, tendo em conta o problema referido anteriormente, poderia ser: [4, 6, 1, 9, 5, 1, 7, 3, 8, 4]. Analisando com atenção, é notório que existem tarefas repetidas ( $T_4, T_1$ ), estando em falta as tarefas ( $T_2, T_{10}$ ). Uma forma de resolver este problema, seria substituir as tarefas repetidas pelas tarefas em falta, de forma aleatória, por exemplo. Devido ao operador de *crossover* escolhido pelo grupo, os cromossomas nunca irão sofrer deste problema, portanto esta correção será desnecessária.

- **Problema 2:** No cromossoma final, pode não ser respeitada a restrição, isto é, as tarefas **T3, T4** e **T5** não são finalizadas em menos de 96 horas.

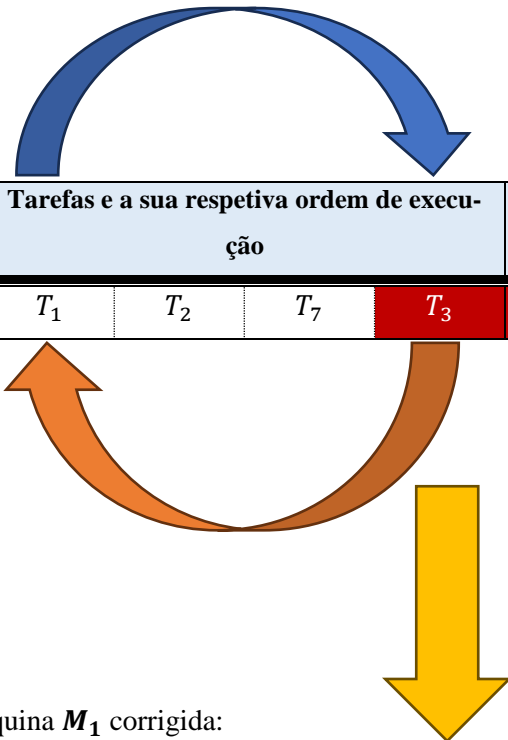
Apesar de os operadores de crossover e de mutação não sofrerem com o **Problema 1**, o algoritmo genético aqui desenvolvido poderá, eventualmente, sofrer do **Problema 2**. Para a resolução deste problema, o grupo irá usar uma [propriedade](#) já referida anteriormente.

Relembrado a propriedade, uma máquina pode ter a ordem das suas tarefas alteradas, pois irá sempre concluir as tarefas todas ao mesmo tempo, quando comparado com a ordem original destas. Tendo isto em conta, bastará priorizar as tarefas com restrição e, para tal, bastará mover as respetivas tarefas para o início das diferentes tarefas, fazendo com que estas sejam as primeiras a serem concretizadas. De seguida, será demonstrado um exemplo, exemplo mais simples do que o que foi comentado e explicado anteriormente ([LINK](#)), para dar um melhor entendimento de como resolver este problema.

Imaginemos que as máquinas  $M_2$  e  $M_3$  respeitam a restrição, mas a máquina  $M_1$  não. Em baixo, é possível visualizar uma demonstração da ordem de execução das tarefas da máquina  $M_1$  e, um exemplo, de o que se pode realizar para que a restrição não seja desrespeitada:



- Máquina  $M_1$  que possui a ordem de execução que torna o cromossoma inadmissível:



Máquina	Tarefas e a sua respetiva ordem de execução				Quando é que a tarefa crítica é executada?
$M_1$	$T_1$	$T_2$	$T_7$	$T_3$	$38 + 37 + 46 + 36 = 157 > 96$

- Máquina  $M_1$  corrigida:

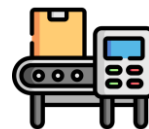
Máquina	Tarefas e a sua respetiva ordem de execução				Quando é que a tarefa crítica é executada?
$M_1$	$T_3$	$T_2$	$T_7$	$T_1$	$36 < 96$

## Alínea g)

Com base nas alíneas anteriores, implemente um algoritmo genético que permita gerar uma solução admissível de qualidade para o problema da *Química\_PT*. Dê uma breve explicação do algoritmo genético que implementou, nomeadamente dimensão da população, forma como é gerada a população inicial, método de seleção, método de substituição da população e critério(s) de paragem

O algoritmo genético escolhido para ser implementado foi o modelo geracional. Neste tipo de algoritmo genético, existe sempre uma população inicial que, com o passar das iterações/gerações, é substituída de forma completa com a criação de filhos (novos cromossomas) a partir destes (relembrando que os pais são escolhidos aleatoriamente, dando prioridade aos melhores crossovers), utilizando operadores de crossover e de mutação, seguindo os princípios da seleção natural. De seguida, podemos visualizar os passos que serão aqui realizados:

1. Inicialização: Geração, de forma aleatória, da população inicial;
2. Seleção dos indivíduos mais aptos, dando uma maior probabilidade de estes serem pais;



- a. Em caso de já não se tratar da população inicial, substituir o pior cromossoma da população/geração pelo melhor global (elitismo);
3. Criação da nova população/ geração através de pais escolhidos aleatoriamente da população/geração anterior, através de:
  - a. Operador de Crossover;
  - b. Operador de Mutação;
  - c. Correção dos inadmissíveis.
4. Se os critérios/condições de paragem não se verificarem (número máximo de iterações; meta do tempo de execução; não existirem melhorias na população durante  $k$  iterações consecutivas), voltar ao passo 2
5. Devolver o melhor cromossoma e o respetivo tempo de execução.

### População inicial

Para a inicialização da população, decidimos fazer uma inicialização aleatória, de forma a ter uma maior diversidade de soluções sem dependência entre elas. Para isso, foi criada uma população composta por  $n$  cromossomas ( $n$  pode ser alterador para qualquer valor superior, ou igual, a 2) com todas as tarefas enunciadas pelo enunciado de forma aleatória em cada um dos cromossomas. De seguida, é possível visualizar o aspeto que uma população inicial teria se tivesse apenas 5 cromossomas

População inicial – 5 cromossomas

Cromossomas Filhos	Aptidão
[1, 2, 9, 3, 8, 7, 5, 4, 10, 6]	141
[10, 3, 7, 6, 4, 5, 9, 8, 2, 1]	151
[1, 10, 6, 4, 3, 7, 5, 8, 9, 2]	127
[4, 10, 1, 2, 3, 7, 5, 6, 9, 8]	131
[8, 7, 6, 9, 5, 3, 10, 2, 1, 4]	147

### Nova população

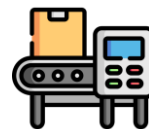
A construção da nova população será composta pelos seguintes passos:

1. Seleção dos pais

A seleção dos pais será feita a partir de uma seleção proporcional à aptidão do cromossoma, ou seja, quanto maior a aptidão do cromossoma na população, maior será a sua probabilidade de ser escolhida, proporcional à sua aptidão. Esta aptidão é calculada através da [função aptidão](#).

2. Cruzamento dos pais

Relativamente ao cruzamento, o operador de crossover já foi explicado na [alínea d\)](#), sendo este uma adaptação do [Operador de recombinação de arestas](#).



### 3. Mutação dos filhos

Por último, a mutação também já foi explicada e comentada, estando já referida na [alínea e](#)), tendo sido escolhida a mutação por troca.

### *Função aptidão*

A função de aptidão é necessária não só para a seleção dos pais, como também para o processo de [elitismo](#) (de seguida, será explicado o que é). A seleção dos pais é feita de forma proporcional à aptidão, ou seja, quanto melhor a aptidão, maior será a probabilidade de ele ser selecionado.

Apesar de teoricamente ser simples, é importante relembrar que o nosso objetivo não é maximizar a aptidão, e sim minimizar o tempo total de processamento das máquinas. A solução encontrada para este problema foi realizar uma transformação, de forma que a aptidão e o tempo total sejam inversamente proporcionais. Para este efeito, a função aptidão terá o seguinte aspeto:

$$aptidao = \frac{1}{tempo\ total}$$

Para uma melhor demonstração do que foi referido, será demonstrado um exemplo: um cromossoma com 120 horas de execução total, terá uma aptidão de  $\frac{1}{120} = 8. (3) \times 10^{-3}$ , enquanto um cromossoma com um tempo de 150 horas terá uma aptidão de  $\frac{1}{150} = 6. (6) \times 10^{-3}$ , fazendo com que a aptidão deste último seja inferior. Isto irá facilitar na seleção dos pais, devido à menor aptidão associada a cada um dos cromossomas.

É importantíssimo referir que estes valores não são a probabilidade de cada um dos pais ser escolhido, sendo apenas uma reinterpretação dos valores, fazendo com que os que tenham um maior tempo de execução se tornem aqueles que possuem menor aptidão, sendo o inverso também verdade. De seguida, com estes novos valores de aptidão, a soma destes corresponderá a 100%, fazendo com que os cromossomas com maiores aptidões tenham um maior peso e, consequentemente, uma maior probabilidade de serem selecionados.

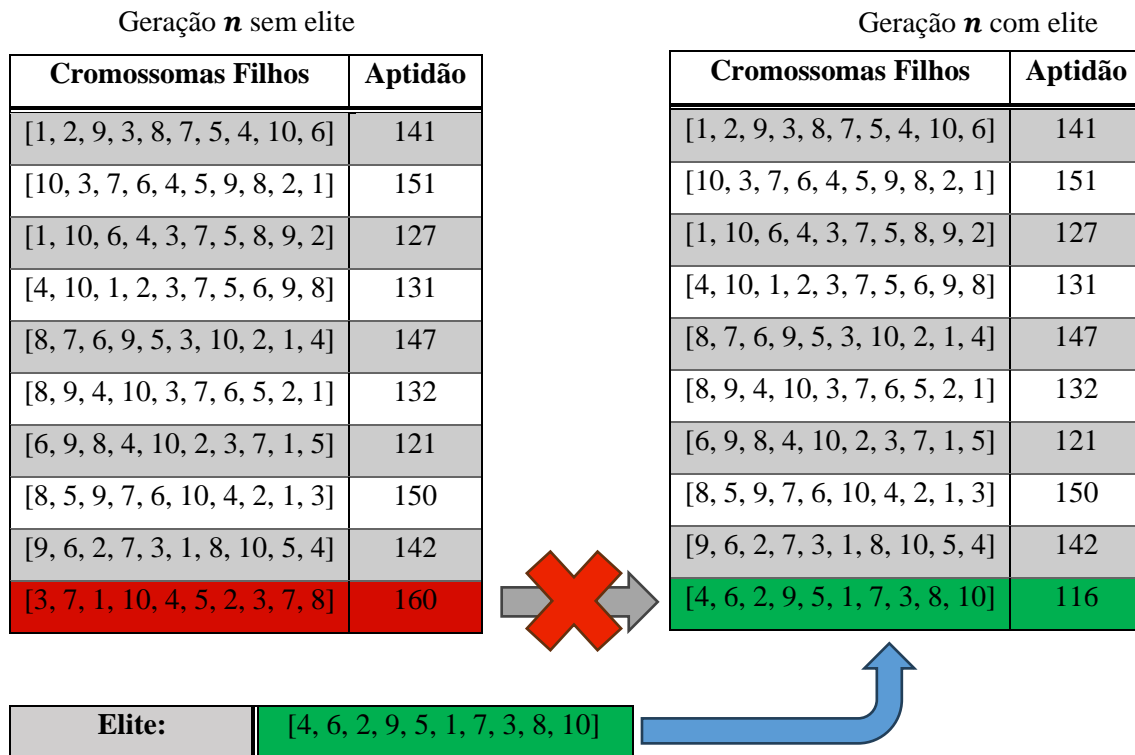
### *Elitismo*

O elitismo é uma técnica utilizada em algoritmos genéticos para preservar os indivíduos mais aptos ao longo das gerações.

O elitismo consiste em selecionar os melhores indivíduos da população atual e garantir a sua reprodução direta para a próxima geração, sem sofrer alterações por meio de operadores genéticos como operadores de crossover e mutação. Desta forma, o elitismo permite manter as características dos indivíduos mais promissores, aumentando a chance de encontrar soluções de melhor qualidade ao longo das iterações do algoritmo genético.



Para uma visualização simples do que é o elitismo, a seguir será apresentado um exemplo simples do mesmo, a fim de ser mais fácil de compreender e de visualizar o mesmo:



### Critérios de paragem

Como o algoritmo genético é uma heurística melhorativa potencialmente sem fim, é necessário criar critérios de paragem. O grupo implementou 3 critérios:

- Um número de iterações máximas que o algoritmo pode fazer, ou seja, caso seja referido  $k$  iterações máximas, o algoritmo não continuará depois da geração  $k$ ;
- Um número de iterações máximas que o algoritmo pode fazer sem melhorias na aptidão, ou seja, seja definido  $n$  como valor deste parâmetro, caso fique  $n$  sem melhorar a aptidão, o algoritmo é interrompido;
- Uma aptidão mínima para o algoritmo parar, por outras palavras, caso se atinga uma aptidão melhor que a referida nos parâmetros (caso seja interpretado com o enunciado, seria uma solução inferior à referida) o algoritmo devolve essa solução e termina.

### Algoritmo

Após esta explicação toda dos passos internos do algoritmo genético aqui implementado, de seguida será apresentado um pseudocódigo do respetivo algoritmo, com o objetivo de dar uma possível forma de visualizar de como o algoritmo num todo funciona:

**Algoritmo** GENETICO**Recebe**

$I(T_p)$  := Função da geração da população inicial  
 $T_e$  := Tamanho da lista de elites  
 $A(c)$  := Função aptidão  
 $C_p(*)$  := Função que devolve se algum critério de paragem é cumprido  
 $T_p$  := Tamanho das populações geradas  
 $S(P)$  := Função seleção que devolve p e m  
 $C_r(p,m)$  := Cruzamento do p e m  
 $M(f,p_m)$  := Mutação do filho, baseado na probabilidade de mutação  
 $p_m$  := Probabilidade de mutação

**Devolve**

$M$  := O cromossoma com maior aptidão encontrado

**Usa**

$P$  := População  
 $E$  := Elites  
 $P'$  := População seguinte que vai substituir a anterior  
 $p$  := Cromossoma pai  
 $m$  := Cromossoma mãe  
 $f$  := Cromossoma filho

**Início:**

$P \leftarrow I(T_p)$   
 Ordenar  $P$  de forma a que os com maior  $A(c)$  estejam primeiro  
 $E \leftarrow$  Os primeiros  $T_e$  cromossomas de  $P$   
**Enquanto não**  $C(*)$ :  
    $P'$  inicializada, vazia  
   **Para**  $i := [1 .. T_p]$  **fazer**:  
      $p, m \leftarrow S(P)$   
      $f \leftarrow C_r(p,m)$   
      $f \leftarrow M(f,p_m)$   
      $P'[i] \leftarrow f$   
   Ordenar  $P'$  de forma a que os com maior  $A(c)$  estejam primeiro  
   **Para**  $i := [1 .. T_e]$  **fazer**:  
      $P'[(T_e - 1 - i)] \leftarrow E[i]$   
   Ordenar  $P'$  de forma a que os com maior  $A(c)$  estejam primeiro  
   **Para**  $i := [1 .. T_e]$  **fazer**:  
      $E[i] \leftarrow P'[i]$   
    $P \leftarrow P'$   
 $M \leftarrow E[1]$

Como este é um problema simples com poucas combinações, nós usámos os seguintes parâmetros:

- Tamanho da População: 20
- Número de iterações máximos: 1000
- Número de iterações máximas sem melhoria: 20
- Tempo Objetivo/Mínimo: 100
- Tamanho da lista de elites: 1
- Probabilidade de mutação: 10%





## Alínea h)

Execute o código desenvolvido e faça uma breve análise à solução admissível obtida para o problema da *Química\_PT*.

Relativamente ao código, em anexo é possível visualizar os respetivos outputs da execução do código:

- Em primeiro, é possível reparar numa possível visualização da [População Inicial](#), onde nos é apresentada os 10 primeiros cromossomas criados, lembrando que estes foram criados aleatoriamente;
- Em seguida, podemos também visualizar um exemplo de uma [Geração](#), onde nos é mostrado as diferentes características de cada um dos cromossomas criados, com os operadores de crossover e mutação referidos anteriormente, como também a atualização do cromossoma de elite.
- Por último, é possível visualizar o output da [Solução Final](#), onde nos é fornecido o melhor cromossoma que já foi criado, com o respetivo valor do tempo total de execuções das tarefas.

Para responder a esta alínea, será analisado o último output, onde nos é mostrado a solução final, sendo esta a seguinte: [4, 5, 9, 6, 3, 2, 7, 8, 1, 10]. Esta solução completou todas as tarefas em 116 horas, respeitando a restrição de finalizar as tarefas  $T_3$ ,  $T_4$  e  $T_5$  em menos de 96 horas.

Interpretando os valores com o enunciado, podemos retirar a seguinte informação:

- Na máquina  $M_1$ , foram realizadas, pela ordem aqui apresentada, as seguintes tarefas:
  - $T_4 \rightarrow T_5 \rightarrow T_9 \rightarrow T_6$
- Na máquina  $M_2$ , foram realizadas, pela ordem aqui apresentada, as seguintes tarefas:
  - $T_3 \rightarrow T_2 \rightarrow T_7$
- Na máquina  $M_3$ , foram realizadas, pela ordem aqui apresentada, as seguintes tarefas:
  - $T_8 \rightarrow T_1 \rightarrow T_{10}$

Relativamente aos valores dos parâmetros, estes foram os escolhidos, pois foram os que permitiram ao grupo atingir a melhor solução global. É de notar que, quando era colocado uma população de 10 cromossomas, a melhor solução global ficava preso no valor 117.



É de notar que o mesmo valor de tempo total deu o mesmo que a heurística construtiva inicial, o que demonstra a utilidade da técnica exposta.

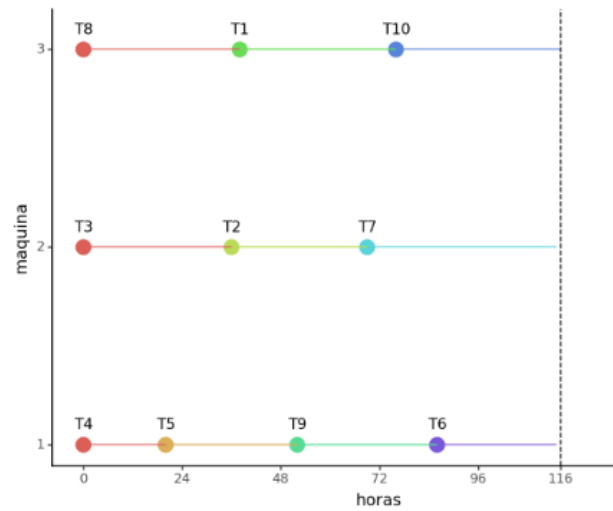


Figura 6: Solução do algoritmo genético

Podemos observar que a solução é semelhante à solução da heurística em termos visuais, mostrando também o tal equilíbrio que a solução mostrava. Curiosamente, a solução alocou as tarefas com menos tempo à máquina 1, semelhante à nossa heurística, mas deixou a máquina 3 sem tarefas prioritárias, diferente da heurística.



## Anexos

### População Inicial

```

|===== População inicial: =====|
Cromossoma 1: [3, 2, 5, 1, 4, 6, 8, 10, 9, 7]
Tempos: [36, 33, 32, 38, 20, 29, 38, 40, 34, 46]
Tempo de execução das Tarefas: 139
Cromossoma 2: [2, 8, 1, 7, 4, 6, 9, 3, 10, 5]
Tempos: [33, 38, 38, 46, 20, 29, 34, 36, 40, 32]
Tempo de execução das Tarefas: 155
Cromossoma 3: [4, 2, 6, 1, 7, 8, 5, 9, 10, 3]
Tempos: [20, 33, 29, 38, 46, 38, 32, 34, 40, 36]
Tempo de execução das Tarefas: 120
Cromossoma 4: [1, 9, 4, 7, 8, 6, 2, 5, 10, 3]
Tempos: [38, 34, 20, 46, 38, 29, 33, 32, 40, 36]
Tempo de execução das Tarefas: 138
Cromossoma 5: [8, 5, 1, 4, 9, 10, 2, 3, 7, 6]
Tempos: [38, 32, 38, 20, 34, 40, 33, 36, 46, 29]
Tempo de execução das Tarefas: 128
Cromossoma 6: [6, 9, 7, 5, 10, 2, 3, 8, 4, 1]
Tempos: [29, 34, 46, 32, 40, 33, 36, 38, 20, 38]
Tempo de execução das Tarefas: 141
Cromossoma 7: [10, 1, 8, 2, 6, 4, 5, 3, 9, 7]
Tempos: [40, 38, 38, 33, 29, 20, 32, 36, 34, 46]
Tempo de execução das Tarefas: 149
Cromossoma 8: [9, 6, 2, 4, 10, 5, 1, 7, 8, 3]
Tempos: [34, 29, 33, 20, 40, 32, 38, 46, 38, 36]
Tempo de execução das Tarefas: 120
Cromossoma 9: [8, 1, 10, 9, 3, 5, 4, 6, 7, 2]
Tempos: [38, 38, 40, 34, 36, 32, 20, 29, 46, 33]
Tempo de execução das Tarefas: 150
Cromossoma 10: [9, 4, 1, 7, 2, 5, 6, 10, 3, 8]
Tempos: [34, 20, 38, 46, 33, 32, 29, 40, 36, 38]
Tempo de execução das Tarefas: 138
Cromossoma 11: [8, 5, 1, 4, 6, 9, 2, 3, 10, 7]
Tempos: [38, 32, 38, 20, 29, 34, 33, 36, 40, 46]
Tempo de execução das Tarefas: 128
Cromossoma 12: [1, 8, 6, 5, 7, 2, 4, 9, 3, 10]
Tempos: [38, 38, 29, 32, 46, 33, 20, 34, 36, 40]
Tempo de execução das Tarefas: 137
Cromossoma 13: [6, 1, 10, 4, 5, 9, 2, 3, 8, 7]
Tempos: [29, 38, 40, 20, 32, 34, 33, 36, 38, 46]
Tempo de execução das Tarefas: 127
Cromossoma 14: [2, 9, 5, 1, 6, 7, 8, 4, 10, 3]
Tempos: [33, 34, 32, 38, 29, 46, 38, 20, 40, 36]
Tempo de execução das Tarefas: 137
Cromossoma 15: [5, 4, 10, 1, 8, 2, 3, 6, 7, 9]
Tempos: [32, 20, 40, 38, 38, 33, 36, 29, 46, 34]
Tempo de execução das Tarefas: 130
Cromossoma 16: [3, 5, 10, 6, 8, 7, 4, 9, 2, 1]
Tempos: [36, 32, 40, 29, 38, 46, 20, 34, 33, 38]
Tempo de execução das Tarefas: 137
Cromossoma 17: [8, 2, 1, 3, 4, 7, 6, 10, 9, 5]
Tempos: [38, 33, 38, 36, 20, 46, 29, 40, 34, 32]
Tempo de execução das Tarefas: 145
Cromossoma 18: [5, 9, 4, 10, 8, 1, 2, 7, 6, 3]
Tempos: [32, 34, 20, 40, 38, 38, 33, 46, 29, 36]
Tempo de execução das Tarefas: 126
Cromossoma 19: [1, 3, 8, 6, 5, 4, 2, 9, 7, 10]
Tempos: [38, 36, 38, 29, 32, 20, 33, 34, 46, 40]
Tempo de execução das Tarefas: 141
Cromossoma 20: [9, 1, 7, 6, 5, 3, 4, 10, 2, 8]
Tempos: [34, 38, 46, 29, 32, 36, 20, 40, 33, 38]
Tempo de execução das Tarefas: 147

```



## Criação das Gerações

```

===== Geração nº 1 =====
○ Candidatos a Elite:
  1: [4, 2, 6, 1, 7, 8, 5, 9, 10, 3]
    - Tempo de execução das Tarefas: 120
    - Aptidão: 0.008333333333333333
  2: [4, 6, 5, 3, 10, 2, 7, 8, 9, 1]
    - Tempo de execução das Tarefas: 119
    - Aptidão: 0.008403361344537815
○ Elite Atualizado ○
  1: [4, 6, 5, 3, 10, 2, 7, 8, 9, 1]
    - Tempo de execução das Tarefas: 119
    - Aptidão: 0.008403361344537815
○ População Inteira:
  - Cromossoma Nº1: [6, 7, 10, 1, 5, 8, 4, 3, 2, 9]
    - Tempo de execução das Tarefas: 153
    - Aptidão: 0.006535947712418301
  - Cromossoma Nº2: [10, 1, 8, 2, 7, 6, 3, 5, 4, 9]
    - Tempo de execução das Tarefas: 149
    - Aptidão: 0.006711409395973154
  - Cromossoma Nº3: [1, 10, 3, 9, 4, 2, 7, 5, 6, 8]
    - Tempo de execução das Tarefas: 148
    - Aptidão: 0.006756756756756757
  - Cromossoma Nº4: [1, 2, 3, 8, 6, 5, 10, 7, 4, 9]
    - Tempo de execução das Tarefas: 145
    - Aptidão: 0.006896551724137931
  - Cromossoma Nº5: [9, 2, 10, 3, 5, 6, 4, 8, 7, 1]
    - Tempo de execução das Tarefas: 143
    - Aptidão: 0.006993006993006993
  - Cromossoma Nº6: [4, 1, 7, 8, 6, 2, 5, 10, 9, 3]
    - Tempo de execução das Tarefas: 142
    - Aptidão: 0.007042253521126761
  - Cromossoma Nº7: [4, 2, 7, 1, 6, 5, 3, 8, 9, 10]
    - Tempo de execução das Tarefas: 137
    - Aptidão: 0.0072992700729927005
  - Cromossoma Nº8: [1, 2, 4, 7, 5, 6, 8, 9, 3, 10]
    - Tempo de execução das Tarefas: 137
    - Aptidão: 0.0072992700729927005
  - Cromossoma Nº9: [1, 2, 7, 4, 9, 10, 3, 8, 5, 6]
    - Tempo de execução das Tarefas: 137
    - Aptidão: 0.0072992700729927005
  - Cromossoma Nº10: [8, 5, 4, 7, 9, 6, 3, 10, 1, 2]
    - Tempo de execução das Tarefas: 136
    - Aptidão: 0.007352941176470588
  - Cromossoma Nº11: [8, 4, 10, 1, 7, 2, 6, 5, 3, 9]
    - Tempo de execução das Tarefas: 136
    - Aptidão: 0.007352941176470588
  - Cromossoma Nº12: [6, 1, 3, 2, 4, 7, 8, 5, 9, 10]
    - Tempo de execução das Tarefas: 136
    - Aptidão: 0.007352941176470588
  - Cromossoma Nº13: [4, 3, 10, 1, 7, 6, 2, 9, 5, 8]
    - Tempo de execução das Tarefas: 134
    - Aptidão: 0.007462686567164179
  - Cromossoma Nº14: [6, 7, 4, 8, 5, 10, 9, 1, 3, 2]
    - Tempo de execução das Tarefas: 133
    - Aptidão: 0.007518796992481203
  - Cromossoma Nº15: [6, 2, 1, 5, 3, 10, 9, 7, 4, 8]
    - Tempo de execução das Tarefas: 132
    - Aptidão: 0.007575757575757576
  - Cromossoma Nº16: [9, 1, 8, 4, 7, 2, 5, 6, 10, 3]
    - Tempo de execução das Tarefas: 130
    - Aptidão: 0.007692307692307693
  - Cromossoma Nº17: [1, 3, 4, 9, 6, 2, 5, 8, 7, 10]
    - Tempo de execução das Tarefas: 128
    - Aptidão: 0.0078125
  - Cromossoma Nº18: [10, 4, 5, 3, 1, 6, 2, 8, 7, 9]
    - Tempo de execução das Tarefas: 128
    - Aptidão: 0.0078125
  - Cromossoma Nº19: [4, 2, 6, 1, 7, 8, 5, 9, 10, 3]
    - Tempo de execução das Tarefas: 120
    - Aptidão: 0.008333333333333333
  - Cromossoma Nº20: [4, 6, 5, 3, 10, 2, 7, 8, 9, 1]
    - Tempo de execução das Tarefas: 119
    - Aptidão: 0.008403361344537815
  
```



## *Solução Final*

- Melhor solução encontrada: [4, 9, 5, 6, 3, 2, 7, 8, 1, 10]
- Tempo de execução das Tarefas: 116



Corrigido: [4, 5, 9, 6, 3, 2, 7, 8, 1, 10]