

Trabalho de Grupo

Algoritmo Genético



UC Otimização Heurística

Licenciatura Ciência de Dados

Grupo 2, CDB1 e CDB2

Docente

Anabela Costa

André Silvestre N°104532

Diogo Catarino N°104745

Francisco Gomes N°104944

Maria Margarida Pereira N°105877

Rita Matos N°104936

Índice

Introdução	3
Algoritmo Genético	3
Resolução	4
a) Solução Admissível	4
b) Heurística e Aplicação	4
c) Definição do Cromossoma	6
d) Operador de <i>Crossover</i>	7
e) Operador de Mutação	9
f) Casos com Soluções Não Admissíveis	10
g) Implementação do Algoritmo Genético	12
h) Resultados Obtidos	15
Conclusão	16
Bibliografia	16

Introdução

Algoritmo Genético

A otimização de problemas complexos tem sido um desafio constante em diversas áreas de estudo. Neste contexto, surgiram as *Metaheurísticas* como técnicas eficazes que procuram soluções admissíveis, sem compromisso de ser a solução ótima, e são especialmente úteis em problemas que não têm uma solução analítica ou que são muito complexos para serem resolvidos com técnicas tradicionais de otimização.

Uma das Metaheurísticas amplamente utilizadas é o **Algoritmo Genético (AG)**. O **AG** é inspirado pelo processo de evolução biológica e é baseado em princípios da *Teoria da Seleção Natural de Darwin*. Foi proposto por *John Holland* na década de 1970 e desde então tem sido aplicado com sucesso em diversas áreas. [1]

O Algoritmo Genético é um método de procura populacional que simula a evolução de uma população de possíveis soluções ao longo de várias gerações. Cada solução, também conhecida como indivíduo, é representada por um cromossoma que codifica uma possível solução para o problema em questão. Os cromossomas são compostos por genes que representam características individuais das soluções.

Este algoritmo utiliza ainda operadores genéticos, como seleção, *crossover* (recombinação genética) e mutação, para explorar melhores soluções ao longo das iterações. Durante o processo evolutivo, os indivíduos mais aptos, ou seja, aqueles com maior qualidade de solução (que será dada pelo objetivo do problema), têm maior probabilidade de serem selecionados para reprodução, promovendo a propagação de características desejáveis na população.

O **Algoritmo Genético** destaca-se como uma poderosa técnica de otimização devido à sua notável flexibilidade e adaptabilidade, permitindo enfrentar desafios numa ampla variedade de problemas.

Resolução

a) Solução Admissível

Neste problema, determinar uma solução admissível consiste em:

- Determinar a atribuição das 10 tarefas (T_1, \dots, T_{10}) a uma e uma só máquina das 3 máquinas disponíveis (M_1, M_2 e M_3);
- Determinar a sequência de processamento das tarefas em cada máquina.

Minimizando o tempo de processamento p_i que decorre entre a programação da primeira tarefa e a conclusão da última, atendendo à condição de iniciar o processo no dia 1 de junho, às 8h, e ter obrigatoriedade de concluir as tarefas T_3, T_4 e T_5 até ao dia 5 de junho, às 8h.

b) Heurística e Aplicação

Pode ser obtida uma solução admissível para o problema da *Química_PT* através da seguinte heurística:

1º Passo - Ordenar as tarefas que têm de estar concluídas até dia 5 de junho, às 8h (T_3, T_4 e T_5) por ordem crescente de tempo de processamento.

Tarefas Prioritárias		
Por Ordem Crescente		
T4	T5	T3
20h	32h	36h

2º Passo - Atribuir à máquina que realiza mais tarefas (M_1) aquela que tem menor tempo de processamento. Assim, esta máquina, que está responsável por processar uma tarefa adicional, é adequada para a execução das tarefas que requerem menos tempo.

Aplicando este passo ao problema, escolheríamos a tarefa T_4 .

3º Passo - Atribuir a próxima tarefa com menor tempo de processamento a uma das máquinas restantes que não têm tarefas atribuídas (**M2** ou **M3**) aleatoriamente.

Ou seja, atribuiremos a tarefa **T5**, por exemplo, à máquina **M2**.

4º Passo - Atribuir a última *Tarefa Prioritária* (isto é, dentro do conjunto das tarefas que têm obrigatoriedade de conclusão anterior às 96h iniciais do processamento), à máquina que ainda não tem nenhuma tarefa atribuída.

Uma vez que resta a tarefa **T3**, e no **Passo 3** afetamos à máquina **M2** a tarefa **T5**, fica a restar a afetação à máquina **M3**, pelo que ficará esta última com a tarefa **T3**.

5º Passo - Organizar as restantes tarefas por ordem crescente de tempo de processamento.

Tarefas Não Prioritárias						
Por Ordem Crescente						
T6	T2	T9	T1	T8	T10	T7
29h	33h	34h	38h	38h	40h	46h

6º Passo - Atribuir à máquina que tem de realizar mais tarefas (**M1**) as tarefas que têm o menor tempo de processamento.

Assim, afetamos as tarefas **T6**, **T2** e **T9** à máquina **M1**.

7º Passo - Atribuir às máquinas disponíveis (**M2** e **M3**) as restantes tarefas de forma a equilibrar as tarefas com maior e menor tempo de processamento.

No caso de ambas as tarefas terem tempos de processamento iguais, escolheremos aleatoriamente a sua afetação.

Aplicando este passo, afetamos as tarefas **T5**, **T8** e **T7** à máquina **M2** e **T3**, **T1** e **T10** à máquina **M3**.

Após a execução destes passos, obtemos a **Solução Admissível**:

- **M1**: T4 → T6 → T2 → T9 (Tempo Total da **Máquina 1** = 116h)
- **M2**: T5 → T8 → T7 (Tempo Total da **Máquina 2** = 116h)
- **M3**: T3 → T1 → T10 (Tempo Total da **Máquina 3** = 114h)

C) Definição do Cromossoma

De modo a resolver o problema proposto, definimos o cromossoma com dimensão 10, em que cada gene assume a posição em que uma das três máquinas realizará uma determinada tarefa e o valor do gene indica qual a tarefa que lhe ficou afeta.

Podemos representar o nosso cromossoma da seguinte forma (Fig.1):

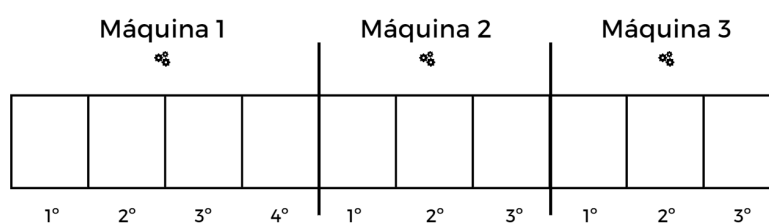


Figura 1 | Ilustração gráfica do cromossoma.

Exemplificando um cromossoma admissível para o problema, podemos representar a solução admissível obtida em **b)**, sendo a codificação:

T4	T6	T2	T9	T5	T8	T7	T3	T1	T10
----	----	----	----	----	----	----	----	----	-----

Este cromossoma representa a solução:

- Tarefas da máquina **M1**: T4 → T6 → T2 → T9
(Tempo Total da **Máquina 1** = 116h)
- Tarefas da máquina **M2**: T5 → T8 → T7
(Tempo Total da **Máquina 2** = 116h)
- Tarefas da máquina **M3**: T3 → T1 → T10
(Tempo Total da **Máquina 3** = 114h)

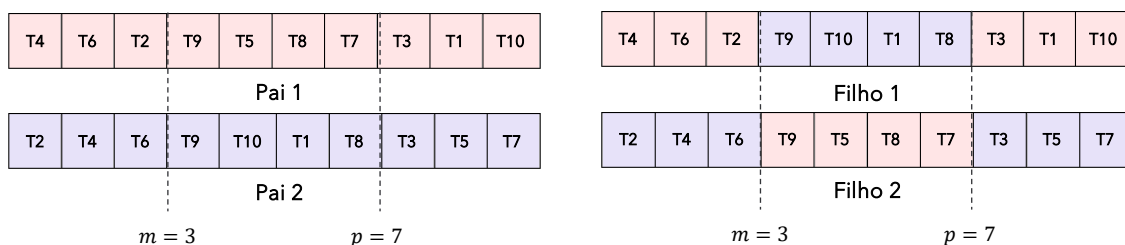
d) Operador de *Crossover*

O operador de crossover proposto para implementação do algoritmo genético será o *Crossover k=2 pontos*, tendo este a probabilidade de ocorrência igual a 1.

Este operador consiste em:

1. Gerar dois números aleatórios, m e p , entre 1 e a dimensão do cromossoma (10), que serão os dois pontos de corte aleatório nos cromossomas Pai.
2. Cada cromossoma Pai é dividido em 3 conjuntos:
 - "Cauda" da esquerda com os genes $1:m$;
 - Parte central com os genes $m:p$;
 - "Cauda" da direita com os genes $p+1:N^o \text{ de Tarefas}$.
3. Copiar a sequência de ambas as "caudas" do cromossoma **Pai 1** e a parte central do cromossoma **Pai 2** para o **Filho 1**.
4. Copiar a sequência de ambas as "caudas" do cromossoma **Pai 2** e a parte central do cromossoma **Pai 1** para o **Filho 2**.

Segue-se um exemplo da realização do *Crossover a k= 2 ponto* em que $m = 3$ e $p = 7$, a partir de dois cromossomas pais admissíveis:



Após o *crossover* dos cromossomas, podemos verificar a sua aptidão e respetiva admissibilidade das soluções. Para tal, apresenta-se de seguida a tabela com os respetivos valores calculados.

	Aptidão				Admissibilidade das Tarefas Prioritárias		
	M1	M2	M3	t_{max}	T3	T4	T5
Pai 1	116h (T4+T6+T2+T9)	116h (T5+T8+T7)	114h (T3+T1+T10)	116h	36h (T3)	20h (T4)	32h (T5)
Pai 2	116h (T2+T4+T6+T9)	116h (T10+T1+T8)	114h (T3+T5+T7)	116h	36h (T3)	53h (T2+T4)	68h (T3+T5)
Filho 1	116h (T4+T6+T2+T9)	116h (T10+T1+T8)	114h (T3+T1+T10)	X	A solução é não admissível porque existem tarefas repetidas e em falta		
Filho 2	116h (T2+T4+T6+T9)	116h (T5+T8+T7)	114h (T3+T5+T7)	X	A solução é não admissível porque existem tarefas repetidas e em falta		

Analisando a tabela anterior verificamos que as soluções **Filho 1** e **Filho 2**, descendentes de 2 pais admissíveis, passam a ser não admissíveis dado que contêm tarefas repetidas e outras em falta.

Assim, teremos de resolver casos como estes numa etapa posterior.

Note-se ainda que poderia ocorrer não admissibilidade no *crossover* com o não cumprimento da data acordada para a conclusão das tarefas **T3**, **T4** e **T5**, apesar de no caso exemplificado não ter ocorrido tal situação.

e) Operador de Mutação

O operador de mutação escolhido é *Mutação por Troca* que terá probabilidade de ocorrer igual a **0.1**. Este valor foi escolhido por ser considerado um valor moderado, permitindo introduzir diversidade na população sem causar grandes perturbações.

Este operador consiste em:

1. Selecionar aleatoriamente um número com distribuição Uniforme em $[0,1]$: u .
2. Caso $u < 0.1$ então selecionam-se aleatoriamente dois genes e troca-se o valor dos seus alelos.

Segue-se um exemplo da realização do operador de *Mutação por Troca* num indivíduo admissível, em que $u = 0.05$ e por isso, selecionam-se aleatoriamente dois genes (por exemplo, 3 e 7) e troca-se o valor dos seus alelos:



Após aplicar o operador de *mutação* no cromossoma, podemos verificar a sua aptidão e respetiva admissibilidade. Para tal, apresenta-se de seguida a tabela com os respetivos valores calculados antes e após mutação.

	Aptidão				Admissibilidade das Tarefas Prioritárias		
	M1	M2	M3	t_{max}	T3	T4	T5
Sem Mutação	136h (T4+T7+T2+T9)	99h (T5+T8+T6)	114h (T3+T1+T10)	136h	36h (T3)	20h (T4)	32h (T5)
Com Mutação	133h (T4+T7+T3+T9)	99h (T5+T8+T6)	114h (T2+T1+T10)	133h	102h (T7+T4+T3)	20h (T4)	32h (T5)

Através da análise das soluções antes e após mutação, podemos verificar que a troca dos alelos dos genes 3 e 8 a solução torna-se não admissível, pois o tempo de conclusão da tarefa $T3 = 102h > 96h$ acordadas.

f) Casos com Soluções Não Admissíveis

Os operadores propostos em **d)** e **e)** não garantem a obtenção de soluções admissíveis para o problema.

No caso do operador de *Crossover $k=2$ pontos*, é necessário garantir que as tarefas são alocadas uma e uma só vez no cromossoma, não faltando nenhuma; e se ocorre violação à restrição do tempo das tarefas com obrigatoriedade de conclusão no prazo definido (T3, T4 e T5).

No caso do operador de *Mutação por Troca*, é importante verificar se ocorre violação à restrição do tempo das tarefas com obrigatoriedade de conclusão no prazo definido (T3, T4 e T5).

Todavia, é possível optar por 2 abordagens distintas para resolver o problema.

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

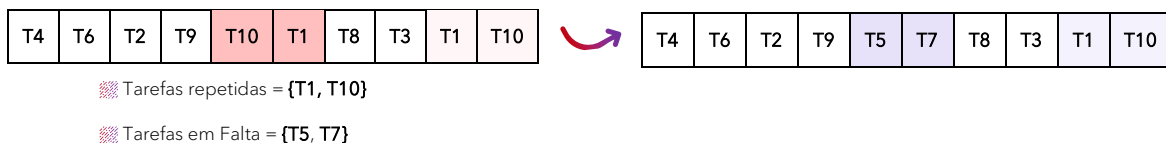
Um exemplo que pode ser posto em prática será:

- Inicialmente, em cada cromossoma, identificam-se as tarefas que estão repetidas e, em seguida, determina-se as tarefas em falta.

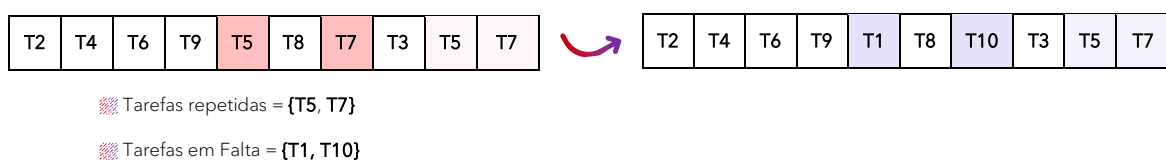
Após identificadas as tarefas que fazem a solução não admissível, procura-se o 1º gene no cromossoma que possui um alelo correspondente a uma tarefa repetida e substitui esse gene pelo valor da tarefa em falta, corrigindo assim a repetição.

O processo é repetido até que todas as tarefas repetidas sejam substituídas pelas tarefas em falta.

Aplicação no Filho 1:



Aplicação no Filho 2:

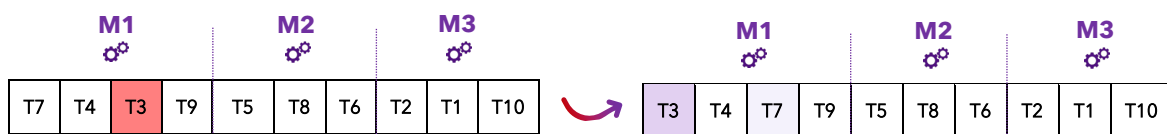


- No caso de as tarefas prioritárias não serem cumpridas, isto é, o tempo de conclusão das tarefas **T3**, **T4** e/ou **T5** seja superior às 96h, divide-se as tarefas por máquina e verifica-se em qual é realizada. De seguida existem dois cenários possíveis.

No primeiro cenário, em que há apenas uma tarefa prioritária na máquina, o alelo da tarefa em questão recua uma posição no cromossoma, a fim de corrigir a inadmissibilidade.

No segundo cenário, em que existem duas tarefas prioritárias não cumpridas, o alelo da tarefa é movido duas posições para trás, garantindo que não ocorra novamente uma situação de inadmissibilidade.

Aplicação no **Cromossoma da Mutação** (alínea **e**):



Neste cromossoma ocorre o 2º cenário dado que as tarefas **T3** e **T4** são processadas na mesma máquina (**M1**), e daí para a solução não ficar inadmissível, a tarefa **T3** recua 2 índices.

➤ **Abordagem 2:** Considerar filhos não admissíveis, mas com a inclusão de uma penalização no respetivo valor da aptidão.

Tomaremos como resposta para as soluções não admissíveis a **Abordagem 1**.

g) Implementação do Algoritmo Genético

Para implementar o **Algoritmo Genético** seguimos as seguintes etapas:

Etapa 1 - População Inicial:

Definimos, aleatoriamente, uma população inicial de **100 indivíduos** utilizando a função **random.shuffle** num cromossoma com as 10 tarefas a realizar.

Para esta seleção, gerou-se todas as permutações possíveis (**10!**) de soluções e, posteriormente, reparou-se a sua admissibilidade, a fim de garantir que “pais” não admissíveis não pudessem ser selecionados em etapas subsequentes.

Etapa 2 - Cálculo dos Valores de Aptidão:

O valor de aptidão de uma solução é tido como o máximo de tempo de processamento por máquina, uma vez que este problema tem como objetivo minimizar o tempo que decorre entre a programação da primeira tarefa e a conclusão da última.

Uma solução será tanto melhor, quanto menor for o valor de aptidão.

Etapa 3 - Método de Seleção dos Pais:

Para seleccionar os pais utilizámos o método de Seleção **Proporcional à Aptidão** no qual cada indivíduo pode tornar-se um pai com uma probabilidade que é proporcional à sua aptidão que foi obtida na **Etapa 2**.

Após a escolha dos dois cromossomas pais, é aplicado o **Operador de Crossover k=2 pontos**, visando originar dois cromossomas filhos.

De seguida, para manter e introduzir diversidade na população, é aplicado o **Operador de Mutação por Troca** com probabilidade igual a **0.1**.

Dada a complexidade do problema, ambos os operadores podem produzir “filhos” inadmissíveis, pelo que, resolvemos esses casos, tornando-os admissíveis.

Etapa 4 - Método de Substituição da População:

Como método de substituição da população, optámos por substituir toda a população "pais" pela população "filhos" (*Método 1*), tendo em conta o **Elitismo**, garantindo que ao longo das várias gerações o cromossoma mais apto é sempre mantido como um dos elementos da população.

Após substituir a população, de modo a aplicar o **Elitismo** averiguamos se na nova população está o cromossoma mais apto:

- ⦿ Se o cromossoma mais apto estiver na nova população, então o algoritmo prossegue para a próxima iteração;
- ⦿ Se o cromossoma mais apto não estiver na nova população, então é necessário retirar da nova população um cromossoma (optámos por retirar o menos apto) e incluir o cromossoma mais apto.

Etapa 5 - Critérios de Paragem do Algoritmo:

Adotou-se como **Critério de Paragem** para o algoritmo, um número de 1000 iterações efetuadas (equivale ao número de gerações da população).

Todo este processo está esquematizado no seguinte fluxograma (Fig. 2):

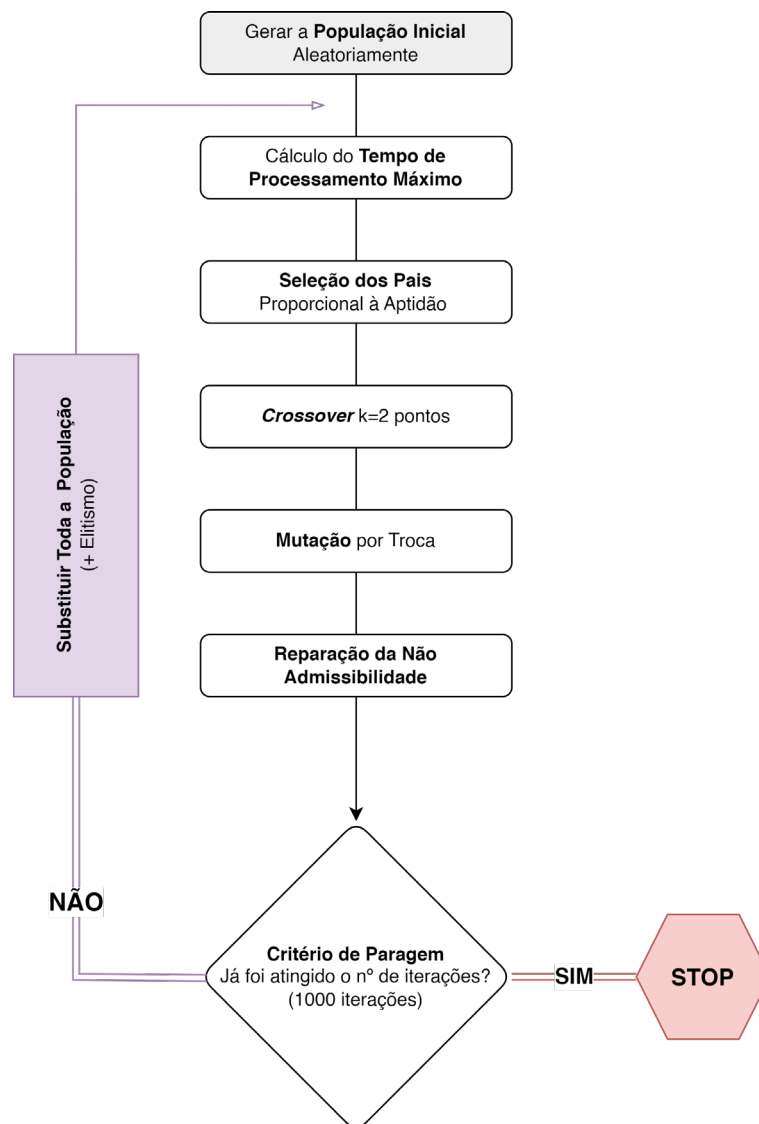


Figura 2 | Fluxograma do Algoritmo Genético.

h) Resultados Obtidos

Após executar o código desenvolvido, obtivemos 4 melhores soluções admissíveis para o problema da *Química_PT*, tal como apresentado no *output* da Figura 3.

```
Melhor Solução Admissível 1 :  
Máquina M1: [4, 5, 6, 2] | Total = 114  
Máquina M2: [1, 10, 8] | Total = 116  
Máquina M3: [9, 3, 7] | Total = 116  
Tempo Máximo: 116
```

```
Melhor Solução Admissível 2 :  
Máquina M1: [2, 4, 6, 9] | Total = 116  
Máquina M2: [8, 10, 1] | Total = 116  
Máquina M3: [5, 3, 7] | Total = 114  
Tempo Máximo: 116
```

```
Melhor Solução Admissível 3 :  
Máquina M1: [5, 4, 2, 6] | Total = 114  
Máquina M2: [1, 10, 8] | Total = 116  
Máquina M3: [9, 3, 7] | Total = 116  
Tempo Máximo: 116
```

```
Melhor Solução Admissível 4 :  
Máquina M1: [2, 9, 4, 6] | Total = 116  
Máquina M2: [8, 10, 1] | Total = 116  
Máquina M3: [5, 3, 7] | Total = 114  
Tempo Máximo: 116
```

Figura 3 | *Output* obtido após executar o algoritmo desenvolvido em *Python*.

Todas as soluções obtidas têm aptidão **116**, sendo este o tempo máximo que decorre entre a programação da primeira tarefa e a conclusão da última tarefa. Um dos fatores na obtenção de várias soluções alternativas deve-se à utilização do **Elitismo** que preserva as melhores soluções ao longo de gerações.

É curioso salientar que coincidentemente as soluções apresentadas têm o mesmo valor de aptidão da obtida na heurística desenvolvida na alínea **b)**.

A obtenção de várias melhores soluções poderá ser de grande importância para o agente de decisão (neste projeto, a *Química_PT*), caso haja alguma impossibilidade em afetar alguma das tarefas na máquina atribuída, e por isso poderá optar pela solução alternativa mais favorável na tomada de decisão.

Conclusão

Sucintamente, o presente trabalho de otimização da afetação de tarefas a máquinas para a indústria do setor químico *Quimica_PT* foi realizado com sucesso, aplicando-se a metaheurística do *Algoritmo Genético*.

Ao utilizar uma técnica de otimização robusta e adaptável, foi possível superar as limitações das abordagens tradicionais e lidar eficazmente com problemas complexos e sem soluções analíticas.

Assim, este trabalho demonstrou uma vez mais a capacidade das metaheurísticas em fornecer soluções promissoras e viáveis, mesmo que não sejam necessariamente soluções ótimas. Estas abordagens podem ser aplicadas em várias áreas da vida quotidiana que desejem utilizar a otimização para auxiliar nas tomadas de decisão, abrindo portas para futuras aplicações.

Em suma, cumprimos com sucesso o objetivo pretendido neste projeto de responder a um problema utilizando a metaheurística do *Algoritmo Genético*.

Bibliografia

[1] Siarry, P. (Ed.) (2016). Metaheuristics, Springer.