
TRABALHO INDIVIDUAL II



TRABALHO REALIZADO POR:

- DIOGO FREITAS Nº104841 CDB1



Índice

Enunciado.....	2
Pergunta a)	3
Pergunta b)	3
Pergunta c)	3
Pergunta d) & e)	4
Pergunta f) & g)	5
Extra	5
Pergunta i)	6
Algoritmo Tabu I.....	8
Algoritmo Tabu II	9
Algoritmo Tabu III.....	10
Conclusão.....	10
Anexos.....	11



Enunciado

A companhia farmacêutica **Lusa_Med** detém os direitos sobre dez projetos I&D, P1, ..., P10, e pretende iniciá-los na tentativa de desenvolver novos medicamentos para o tratamento de dez tipos específicos de doenças. Cada projeto necessita de um coordenador distinto para o liderar e, atendendo à exigência dos projetos, cada coordenador só poderá liderar um só projeto.

A **Lusa_Med** já selecionou dez cientistas seniores, C1, ..., C10, e pretende saber como deve alocar os cientistas aos projetos. Para o efeito, a companhia elaborou uma tabela com a aptidão de cada cientista para liderar cada um dos projetos (medida na escala 0-100):

Cientistas	Projetos I&D									
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
C1	70	65	55	50	90	67	80	62	100	85
C2	74	83	54	60	100	75	50	76	82	45
C3	71	87	66	58	74	81	48	52	100	64
C4	50	89	78	63	51	40	48	100	68	75
C5	100	66	83	77	54	58	93	89	53	48
C6	80	55	70	65	94	47	60	88	73	60
C7	87	63	90	79	47	77	90	76	85	90
C8	67	95	100	40	70	54	70	100	65	57
C9	90	45	88	48	65	68	80	46	71	67
C10	67	77	50	60	100	70	80	60	65	70

A **Lusa_Med** pretende determinar a alocação dos cientistas aos projetos que maximiza a aptidão total. Na tentativa de atingir este objetivo, a companhia irá definir e implementar um algoritmo de Pesquisa Tabu.

Passaremos então à resposta de cada uma das perguntas.



Pergunta a)

Descreva por palavras em que consiste uma solução admissível para o problema.

Na otimização heurística, uma solução admissível para um problema consiste em uma solução que satisfaça todas as restrições e requisitos deste, atingindo um nível aceitável de desempenho, mesmo que não seja necessariamente a solução ótima ou a melhor solução possível. No contexto deste problema, seria arranjar uma solução admissível para que houvesse uma boa associação entre os cientistas e os projetos, dando uma soma de aptidões minimamente boa.

O objetivo da otimização heurística é encontrar soluções admissíveis eficientes e próximas do ótimo, dentro das limitações de recursos e tempo disponíveis. Essas soluções admissíveis podem ser encontradas usando métodos heurísticos, que são técnicas que exploram regras práticas, conhecimento especializado ou abordagens aproximadas para encontrar soluções promissoras.

Pergunta b)

Defina uma heurística que lhe permita determinar uma solução admissível para o problema.

Uma heurística que permite determinar uma solução admissível para o problema seria:

- Começamos por descobrir a aptidão média de cada um dos cientistas. De seguida, começando pelos cientistas com pior média de aptidão, atribuímos o projeto que este possui melhor aptidão.
- Existe várias outras soluções, como por exemplo, poderíamos, simplesmente, começar pelo projeto 1 e ir associando este ao cientista com melhor aptidão (como esta não foi a heurística escolhida, não será mais aprofundada, mas fica aqui referida como outra possibilidade).

Pergunta c)

Tendo em conta a alínea b), determine uma solução admissível para o problema.

Podemos então por começar por visualizar a respetiva média de aptidão de cada um dos cientistas:

Cientistas	Média de Aptidão
C_4	66.2
C_9	66.8
C_6	69.2
C_2	69.9
C_{10}	69.9
C_3	70.1
C_8	71.8
C_5	72.1
C_1	72.4
C_7	78.4

Visualizando a tabela acima, é notório a existência de cientistas que apresentam uma maior aptidão que outros. Ao organizar os cientistas desta forma, existe uma maior segurança em relação à aptidão final, pois, o melhor cientista vai ficar com o projeto mais difícil, não escolhido pelos colegas, enquanto o pior cientista vai ficar com o seu melhor projeto.

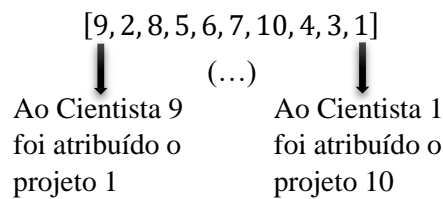


Após esta breve explicação e visualização do que se teve em base para a heurística, podemos então passar para a visualização da solução obtida

Projetos I&D	Cientistas	Pontos de Aptidão
P1	C_9	90
P2	C_2	83
P3	C_8	100
P4	C_5	77
P5	C_6	94
P6	C_7	77
P7	C_{10}	80
P8	C_4	100
P9	C_3	100
P10	C_1	85
Total		886

De seguida, será mostrada qual a lógica que foi usada para organizar as soluções obtidas no *Python*.

De forma resumida, as soluções foram guardadas em listas, onde foram organizadas pelo projeto (correspondendo ao índice da lista mais um), e cada elemento da lista representa o respetivo cientista que possui o projeto. Para uma melhor visualização, em baixo é possível visualizar um exemplo do que foi explicado:



Pergunta d) & e)

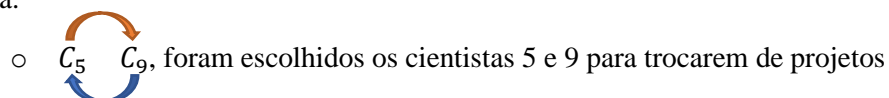
Defina a estrutura de vizinhança de uma solução e determine uma solução vizinha.

Primeiro, é necessário entender o que é ser uma estrutura de vizinhança. Podemos então começar por dizer que, em um algoritmo de pesquisa tabu, a estrutura de vizinhança define as modificações que podem ser aplicadas a uma solução inicial para explorar soluções próximas.

Neste problema em questão, será realizada uma modificação relativamente às trocas dos cientistas (Vizinhança por troca) tendo cada um destes um projeto a ele associado, que impacta a aptidão total. É importante relembrar que o ponto de partida será a solução obtida anteriormente na alínea c).

Para então realizar esta troca, serão escolhidos 2 cientistas aleatoriamente, não tendo sido aplicado nenhuma regra específica, nem tendo havido a preocupação de melhorar a aptidão total da solução.

Neste passo, iremos então realizar a nova troca. Em baixo podemos visualizar a troca que foi realizada:



Esta troca trouxe consequências na soma das aptidões dos cientistas, pois houve atualizações relativamente às associações dos projetos. Relativamente às mudanças que ocorreram com esta troca, podemos começar pela visualização da solução vizinha, não



esquecendo que esta corresponde a uma troca de 2 cientistas da solução inicial: [5, 2, 8, 9, 6, 7, 10, 4, 3, 1]

De seguida, podemos visualizar a tabela com as respetivas aptidões de cada um dos cientistas em cada um dos projetos a eles associado, com a respetiva soma destes:

Projetos I&D	Cientistas	Pontos de Aptidão
P1	C_5	100
P2	C_2	83
P3	C_8	100
P4	C_9	48
P5	C_6	94
P6	C_7	77
P7	C_{10}	80
P8	C_4	100
P9	C_3	100
P10	C_1	85
Total		867

Se analisarmos atentamente, podemos verificar que esta solução vizinha possui uma aptidão total inferior ao da solução inicial onde foi aplicado a heurística. Realizando as contas, podemos concluir que a aptidão total diminuiu $886 - 867 = 19$.

Pergunta f) & g)

Relativamente à lista Tabu, primeiro é necessário entender o que é isto. Uma lista tabu é uma estrutura utilizada em algoritmos de busca e otimização para evitar que soluções já exploradas sejam revisitadas em iterações subsequentes. Ela é usada em algoritmos de busca local, como o algoritmo tabu, tema principal deste trabalho.

Como foi referido anteriormente, a lista tabu existe para não permitir que soluções visitadas sejam revisitadas (não permitindo que o algoritmo retorne a soluções anteriores, incentivando a exploração de diferentes regiões do espaço de busca). Consequentemente, é necessário de adicionar à lista tabu a mudança que foi realizada no exercício e), mudança esta que se intitula de movimento tabu (por outras palavras, o movimento tabu é uma mudança específica em uma solução que é temporariamente proibida por um determinado período).

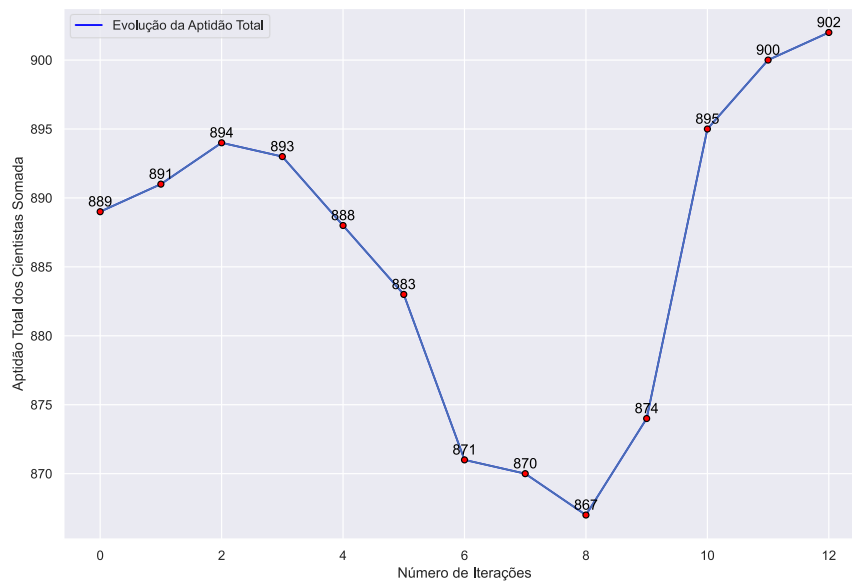
Podemos então visualizar a atualização da lista tabu:

Lista Tabu: [(5,9)]

Apesar de estar apenas a aplicar a parte teórica, fica a questão: Será que esta troca terá sido útil? Na tabela é possível visualizar que a soma das aptidões diminuiu, o que comprova que a solução é pior que a inicial. Sem visualização de um gráfico, pode ficar um pouco difícil de conseguir compreender o algoritmo tabu e, devido a isso, colocarei um tópico **extra**, a fim de conseguir dar um melhor entendimento a este algoritmo e como o mesmo funciona.

Extra

Em baixo, será apresentado um gráfico que demonstra a evolução da aptidão total dos cientistas ao aplicar o algoritmo tabu, é importante ter em conta que foi usado como solução inicial, a solução vizinha obtida na alínea e):



Analisando o gráfico, o 902 corresponde ao máximo global e, para chegar a este resultado, é crucial passar por outras soluções piores. De forma breve, podemos verificar que o 894 é um máximo local e, se não “piorássemos” a solução, ficaríamos presos no pico do máximo local, nunca atingindo o máximo global.

Consequentemente, a solução vizinha obtida na alínea e) não foi um passo negativo e sim um passo necessário para atingir o máximo global. É importante lembrar que este modelo é controlado pelo número de iterações e pelo valor da aptidão total, possuindo uma lista tabu que não permite o modelo voltar a soluções já visitadas, mais à frente, este tópico será explicado de forma mais detalhada.

Em anexo, podemos então visualizar a respetiva tabela do gráfico com as atualizações das listas tabus (lembrar que foi tido como solução inicial a solução: [5, 2, 8, 9, 6, 7, 10, 4, 3, 1]).

Pergunta i)

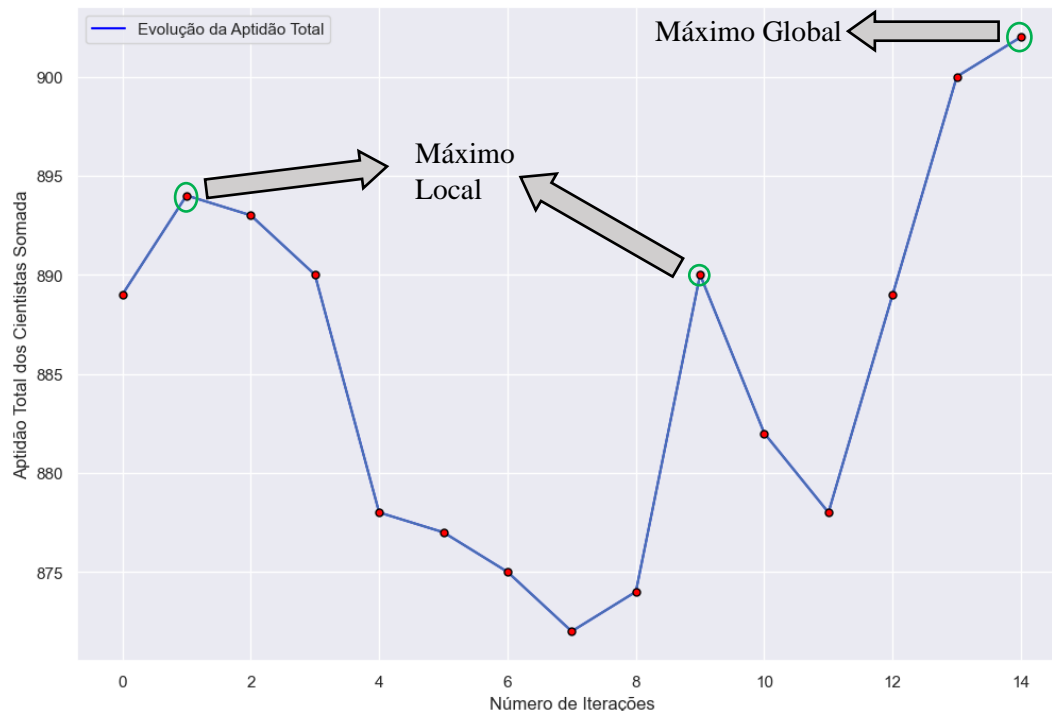
Tendo em conta as opções tomadas para responder às alíneas anteriores, implemente o algoritmo de Pesquisa Tabu, tomando como critérios de paragem um número máximo de 100 iterações ou a obtenção de uma solução admissível cujo valor de aptidão total seja pelo menos 850.

Antes de responder à pergunta, é importante referir que foram feitas algumas alterações relativamente aos valores do enunciado. Todos estas alterações serão referidas no momento que forem aplicadas e, não esquecendo, estas alterações apenas foram realizadas com o objetivo de deixar os resultados mais interessantes.

Para dar uma explicação mais simples sobre o algoritmo Tabu, vamos usufruir do gráfico abaixo que irá exemplificar visualmente as iterações deste algoritmo. Primeiramente, começamos por arranjar uma solução inicial, quer esta seja aleatória ou por uma heurística, e, de seguida, começamos por criar soluções vizinhas (pergunta d & e). Estas soluções vizinhas podem ser piores ou melhores que a solução original, mas atualizamos a solução para a melhor da vizinhança, tendo em conta a existência das listas tabus que não permite que se repita trocas. Por permitirmos que a solução piore, a solução não irá ficar presa num pico (visualizar na



imagem abaixo), podendo se mover livremente até chegar ao máximo global (se tivermos sorte e manipulando o tamanho da lista tabu e do número de iterações)



De seguida podemos então visualizar um pseudocódigo do algoritmo Tabu que foi aplicado na realização deste projeto, ter em atenção que é uma versão simplificada do mesmo.

```
definir AlgoritmoTabu() :

    // Inicialização
    SolucaoAtual = GerarSolucaoInicial()
    MelhorSolucao = SolucaoAtual
    ListaTabu = InicializarListaTabu()

    // Critério de parada
    Enquanto não atingir um número de iterações máximo ou um valor de aptidão:
        MelhorVizinhanca = None

        // Gerar vizinhança da solução atual
        Vizinhanca = GerarVizinhanca(SolucaoAtual)

        Para cada Movimento na Vizinhanca:
            Se Movimento é permitido e não está na ListaTabu:
                NovaSolucao = AplicarMovimento(SolucaoAtual, Movimento)
                AvaliarAptidao(NovaSolucao)

                Se Aptidao(NovaSolucao) > Aptidao(MelhorVizinhanca):
                    MelhorVizinhanca = Movimento
                    MelhorSolucaoVizinhanca = NovaSolucao

        // Atualizar solução atual
        SolucaoAtual = MelhorSolucaoVizinhanca

        Se Aptidao(SolucaoAtual) > Aptidao(MelhorSolucao):
            MelhorSolucao = SolucaoAtual

        // Atualizar ListaTabu
        AdicionarMovimento(ListaTabu, MelhorVizinhanca)
        AtualizarListaTabu(ListaTabu)

    Retornar MelhorSolucao
```




Algoritmo Tabu I

Após esta breve explicação, passemos então à resposta da pergunta. Começaremos por aplicar este algoritmo tabu com as seguintes restrições e predefinições:

- Solução inicial: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
- Tamanho da Lista Tabu: 5
- Número de Iterações Máximo: 100
- Valor mínimo da solução admissível: 900

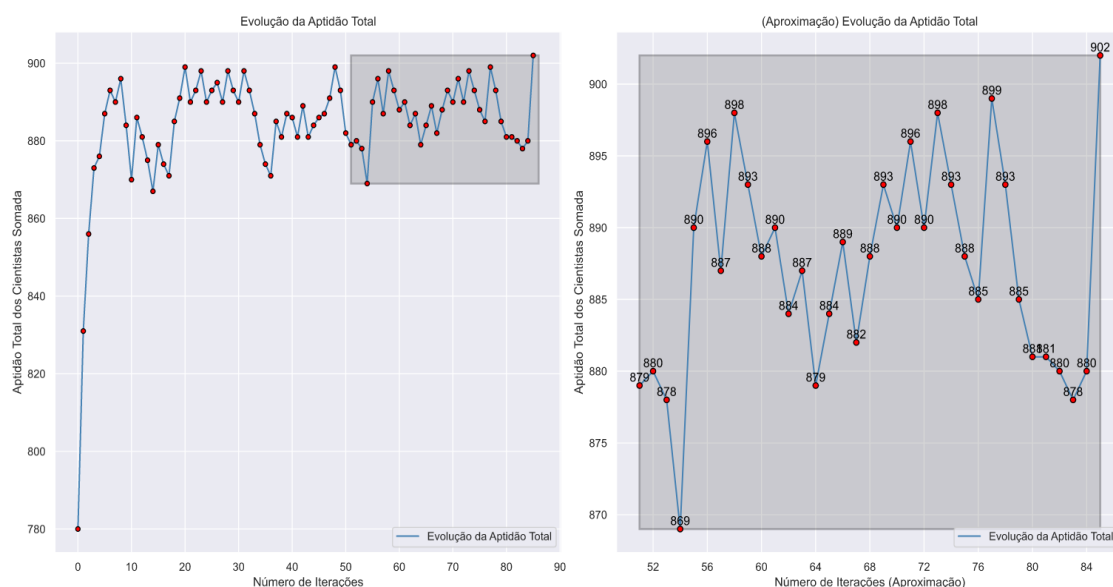
Após aplicar o algoritmo com estas restrições, obtive a seguinte solução final:

Projetos I&D	Cientistas	Pontos de Aptidão
P1	C ₉	90
P2	C ₂	83
P3	C ₈	100
P4	C ₆	65
P5	C ₁₀	100
P6	C ₃	81
P7	C ₅	93
P8	C ₄	100
P9	C ₁	100
P10	C ₇	90
Total		902

Visualizando a tabela acima, podemos ver que obtivemos como solução final foi:

- [9, 2, 8, 6, 10, 3, 5, 4, 1, 7], com uma aptidão total de 902

Relativamente às soluções iterações, de seguida poderemos visualizar um gráfico com a respetiva evolução da solução final ao longo das iterações.



Visualizando o gráfico com atenção, é possível verificar que este algoritmo tabu foi até 86 iterações, atingindo então o valor, da soma das aptidões, de 902, um valor superior a 900. O gráfico da direita é uma aproximação da parte final do gráfico da esquerda e, analisando com atenção, podemos reparar nos valores das somas das aptidões em cada bola vermelha.



Para uma melhor análise da lista Tabu, nos anexos poderemos encontrar, não só a respetiva tabela com os valores do gráfico, como também uma matriz com as combinações Cientista/Projeto com os respetivos valores de aptidão.

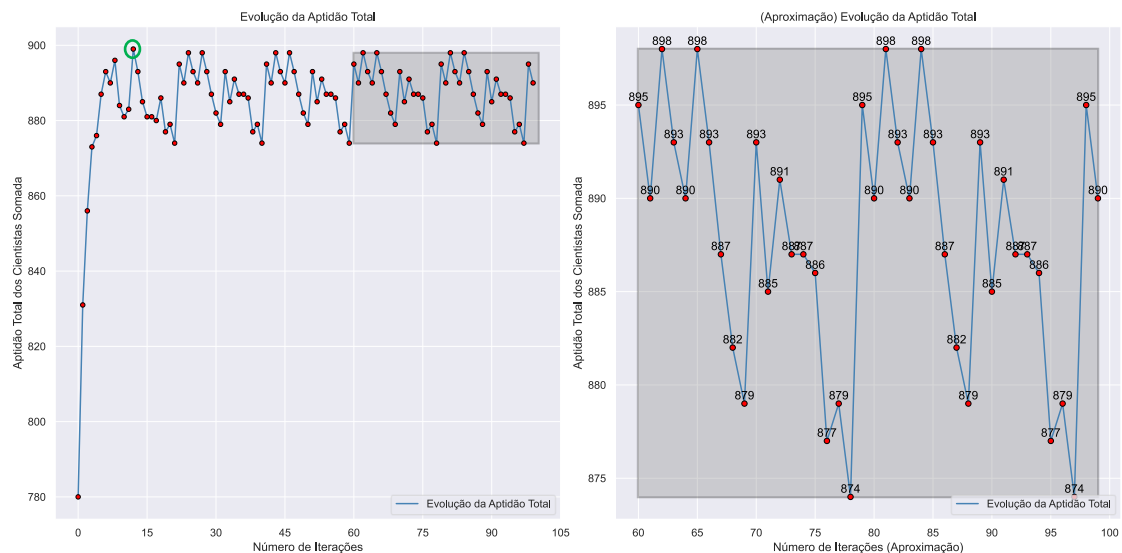
Algoritmo Tabu II

Agora no segundo teste, será realizado o mesmo algoritmo, mas com o tamanho da lista tabu diferente, alterando o tamanho da lista para 4.

Em baixo podemos então visualizar a tabela com a respetiva solução final obtida neste algoritmo com:

Projetos I&D	Cientistas	Pontos de Aptidão
P1	C ₅	100
P2	C ₈	95
P3	C ₉	88
P4	C ₆	65
P5	C ₂	100
P6	C ₃	81
P7	C ₁₀	80
P8	C ₄	100
P9	C ₁	100
P10	C ₇	90
Total		899

A seguir, podemos então visualizar o gráfico com a respetiva evolução da soma das aptidões a cada iteração:



Neste caso, podemos então visualizar que o algoritmo atingiu as 100 iterações, pois, não conseguiu alcançar uma soma das aptidões superior a 900. Este algoritmo conseguiu como melhor valor da soma das aptidões 899 (está marcado a verde no gráfico).

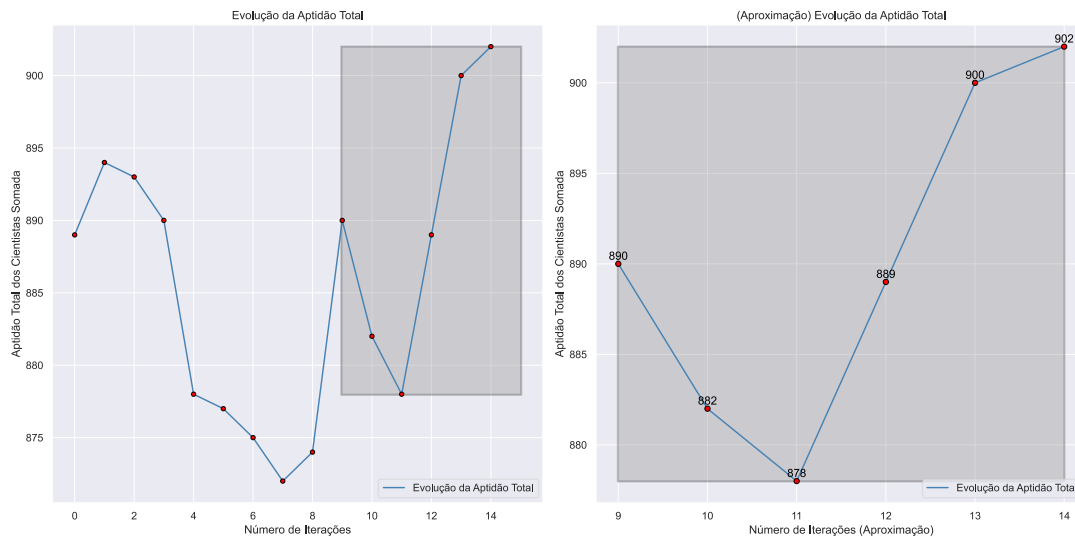
O gráfico à direita é uma aproximação do final do gráfico da esquerda e, visualizando com atenção, é notório que este ficou preso numa espécie de loop. Podemos então concluir que, por termos diminuído o tamanho máximo da lista tabu, o algoritmo não conseguiu atingir um valor da aptidão superior a 900, pois, está a realizar uma troca que prejudica a obtenção da melhor solução possível.



Para uma melhor análise da lista Tabu, nos anexos poderemos encontrar, não só a respectiva tabela com os valores do gráfico, como também uma matriz com as combinações Cientista/Projeto com os respectivos valores de aptidão.

Algoritmo Tabu III

Neste último algoritmo tabu, foi colocado o tamanho da lista tabu igual a 5 novamente, mas foi usado outra solução inicial, tendo sido escolhido a solução obtida da heurística inicial, portanto atualizaremos a solução inicial para a seguinte solução: [9, 2, 8, 5, 6, 7, 10, 4, 3, 1]. Em baixo, podemos visualizar o gráfico com a evolução da soma das aptidões.



Este algoritmo conseguiu atingir a solução máxima global (podemos visualizar a tabela do algoritmo tabu I que possui a mesma solução final e a mesma aptidão).

Neste caso, em menos iterações, conseguiu chegar à solução ótima global, tendo atingido, apenas 14 iterações.

Conclusão

Este trabalho permitiu-me não só ter um maior senso crítico, como também me permitiu treinar a minha criatividade e capacidade de pensar. Desde a criação de uma heurística, como a aplicação do algoritmo tabu, que me obrigaram a inovar e a programar.

Conseguir obter uma maior compreensão de como o algoritmo tabu funciona, tendo programado o algoritmo do zero, também é importante referir que a criação de gráficos e de tabelas facilitaram a compreensão do mesmo.

Concluo assim que, o algoritmo tabu é muito útil para vários problemas, sendo necessário realizar vários testes com o número de iterações, o tamanho da lista tabu e a solução inicial nele implementado.



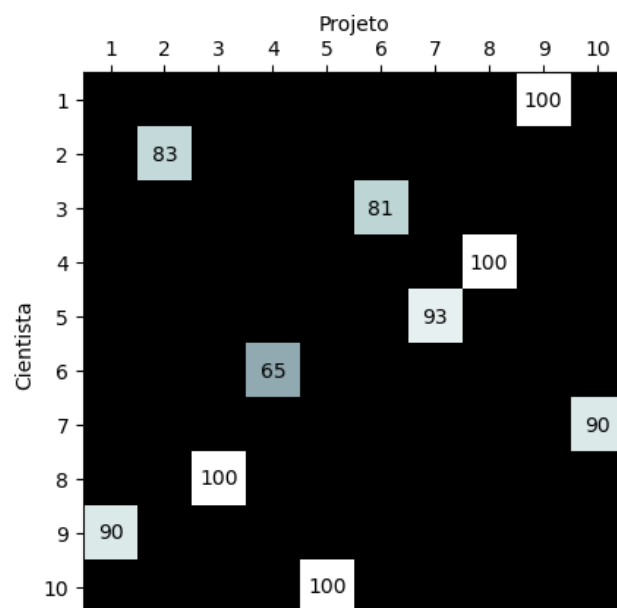
Anexos

Aqui estão colocados os outputs do código, com breves explicações/conclusões dos resultados obtidos. Todas as imagens aqui presentes encontram-se no ficheiro do *jupyter notebook* enviado.

Pergunta f) & g) - Extras

Algoritmo Tabu				
	Iterações	Combinações vizinhas	Aptidão Total	ListaTabu
1	1	[5, 2, 8, 7, 6, 9, 10, 4, 3, 1]	889	[(7, 9)]
2	2	[5, 2, 8, 7, 6, 10, 9, 4, 3, 1]	891	[(7, 9), (10, 9)]
3	3	[9, 2, 8, 7, 6, 10, 5, 4, 3, 1]	894	[(7, 9), (10, 9), (9, 5)]
4	4	[9, 10, 8, 7, 6, 2, 5, 4, 3, 1]	893	[(7, 9), (10, 9), (9, 5), (10, 2)]
5	5	[9, 10, 8, 5, 6, 2, 7, 4, 3, 1]	888	[(10, 9), (9, 5), (10, 2), (5, 7)]
6	6	[9, 10, 8, 5, 6, 2, 1, 4, 3, 7]	883	[(9, 5), (10, 2), (5, 7), (1, 7)]
7	7	[9, 10, 8, 5, 6, 3, 1, 4, 2, 7]	871	[(10, 2), (5, 7), (1, 7), (3, 2)]
8	8	[9, 3, 8, 5, 6, 10, 1, 4, 2, 7]	870	[(5, 7), (1, 7), (3, 2), (3, 10)]
9	9	[9, 3, 8, 5, 2, 10, 1, 4, 6, 7]	867	[(1, 7), (3, 2), (3, 10), (2, 6)]
10	10	[9, 3, 8, 5, 2, 10, 6, 4, 1, 7]	874	[(3, 2), (3, 10), (2, 6), (6, 1)]
11	11	[9, 3, 8, 6, 2, 10, 5, 4, 1, 7]	895	[(3, 10), (2, 6), (6, 1), (6, 5)]
12	12	[9, 3, 8, 6, 10, 2, 5, 4, 1, 7]	900	[(2, 6), (6, 1), (6, 5), (2, 10)]
13	13	[9, 2, 8, 6, 10, 3, 5, 4, 1, 7]	902	[(6, 1), (6, 5), (2, 10), (2, 3)]

É possível visualizar que a lista tabu possui um tamanho máximo de 4 e, quando adicionamos um novo movimento tabu, é eliminado o movimento mais antigo. É notório que o movimento tabu mais à direita da lista tabu é o movimento realizado na iteração. Neste algoritmo, com apenas 13 iterações, conseguiu atingir a melhor solução global.



Ao lado podemos então visualizar a respetiva matriz com os valores de combinação Cientista/Projeto. Para então termos o valor da aptidão total, basta somar todos os valores que se encontram presentes na matriz:

$$\begin{aligned} \text{Aptidão Total} &= 100 + 83 + 81 + 100 \\ &\quad + 93 + 65 + 90 + 100 \\ &\quad + 90 + 100 = 902 \end{aligned}$$

Após vários testes, posso confirmar que 902 é a melhor aptidão que é possível chegar, podendo concluir que o algoritmo Tabu conseguiu atingir a solução ótima global.

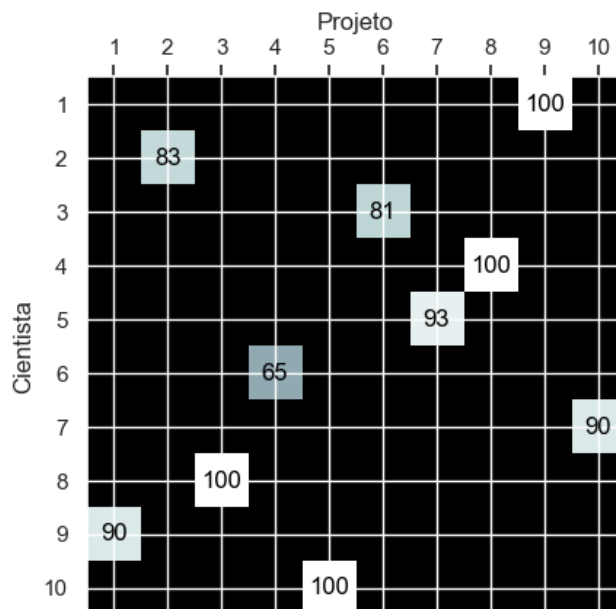


Pergunta i)

Algoritmo Tabu I

Algoritmo Tabu				
	Iterações	Combinações vizinhas	Aptidão Total	ListaTabu
1	1	[5, 2, 3, 4, 1, 6, 7, 8, 9, 10]	780	[(5, 1)]
2	2	[5, 2, 9, 4, 1, 6, 7, 8, 3, 10]	831	[(5, 1), (9, 3)]
3	3	[5, 2, 9, 4, 10, 6, 7, 8, 3, 1]	856	[(5, 1), (9, 3), (10, 1)]
4	4	[5, 2, 9, 4, 6, 10, 7, 8, 3, 1]	873	[(5, 1), (9, 3), (10, 1), (6, 10)]
5	5	[5, 4, 9, 2, 6, 10, 7, 8, 3, 1]	876	[(5, 1), (9, 3), (10, 1), (6, 10), (4, 2)]
...
82	82	[6, 4, 9, 2, 10, 3, 5, 8, 1, 7]	881	[(3, 1), (4, 8), (6, 5), (10, 5), (2, 10)]
83	83	[2, 4, 9, 6, 10, 3, 5, 8, 1, 7]	880	[(4, 8), (6, 5), (10, 5), (2, 10), (2, 6)]
84	84	[6, 2, 9, 4, 10, 3, 5, 8, 1, 7]	878	[(6, 5), (10, 5), (2, 10), (2, 6), (4, 2)]
85	85	[9, 2, 4, 6, 10, 3, 5, 8, 1, 7]	880	[(10, 5), (2, 10), (2, 6), (4, 2), (9, 4)]
86	86	[9, 2, 8, 6, 10, 3, 5, 4, 1, 7]	902	[(2, 10), (2, 6), (4, 2), (9, 4), (8, 4)]

É possível visualizar que a lista tabu possui um tamanho máximo de 5 e, quando adicionamos um novo movimento tabu, é eliminado o movimento mais antigo. É notório que o movimento tabu mais à direita da lista tabu é o movimento realizado na iteração. Em 86 iterações, conseguiu atingir a melhor solução global.



que a solução anterior.

Ao lado podemos então visualizar a respetiva matriz com os valores de combinação Cientista/Projeto. Para então termos o valor da aptidão total, basta somar todos os valores que se encontram presentes na matriz:

$$\begin{aligned} \text{Aptidão Total} &= 100 + 83 + 81 + 100 \\ &\quad + 93 + 65 + 90 + 100 \\ &\quad + 90 + 100 = 902 \end{aligned}$$

Após vários testes, posso confirmar que 902 é a melhor aptidão que é possível chegar, podendo concluir que o algoritmo Tabu conseguiu atingir a solução ótima global. É de acrescentar que foram necessárias mais iterações para atingir a melhor combinação global



Algoritmo Tabu II

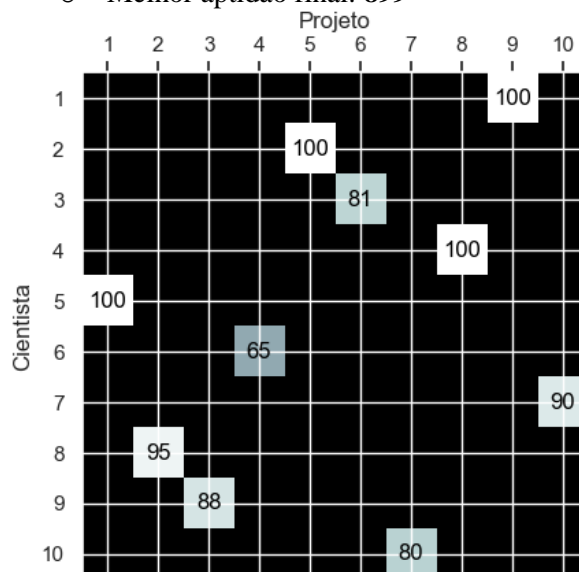
Algoritmo Tabu

	Iterações	Combinações vizinhas	Aptidão Total	ListaTabu
1	1	[5, 2, 3, 4, 1, 6, 7, 8, 9, 10]	780	[(5, 1)]
2	2	[5, 2, 9, 4, 1, 6, 7, 8, 3, 10]	831	[(5, 1), (9, 3)]
3	3	[5, 2, 9, 4, 10, 6, 7, 8, 3, 1]	856	[(5, 1), (9, 3), (10, 1)]
4	4	[5, 2, 9, 4, 6, 10, 7, 8, 3, 1]	873	[(5, 1), (9, 3), (10, 1), (6, 10)]
5	5	[5, 4, 9, 2, 6, 10, 7, 8, 3, 1]	876	[(9, 3), (10, 1), (6, 10), (4, 2)]
...
96	96	[3, 8, 9, 6, 10, 2, 5, 4, 1, 7]	877	[(10, 5), (2, 10), (2, 6), (3, 2)]
97	97	[6, 8, 9, 3, 10, 2, 5, 4, 1, 7]	879	[(2, 10), (2, 6), (3, 2), (6, 3)]
98	98	[6, 8, 9, 7, 10, 2, 5, 4, 1, 3]	874	[(2, 6), (3, 2), (6, 3), (7, 3)]
99	99	[6, 8, 9, 7, 10, 2, 5, 4, 3, 1]	895	[(3, 2), (6, 3), (7, 3), (3, 1)]
100	100	[6, 8, 9, 5, 10, 2, 7, 4, 3, 1]	890	[(6, 3), (7, 3), (3, 1), (5, 7)]

É possível visualizar que a lista tabu possui um tamanho máximo de 4 e, quando adicionamos um novo movimento tabu, é eliminado o movimento mais antigo. É notório que o movimento tabu mais à direita da lista tabu é o movimento realizado na iteração.

Também é possível reparar que a última linha da tabela não corresponde à melhor combinação, pois, nunca atingiu um valor da aptidão total superior a 900, mas atingiu as 100 iterações que era uma das restrições do algoritmo. Os melhores resultados que este algoritmo atingiu foram os seguintes:

- Melhor combinação final: [5, 8, 9, 6, 2, 3, 10, 4, 1, 7]
- Melhor aptidão final: **899**



Vamos então calcular a aptidão total através da matriz presente no lado esquerdo:

$$\text{Soma das aptidões} = 100 + 95 + 88 + 65 + 100 + 81 + 80 + 100 + 100 + 90 = 899$$

Podemos concluir que diminuir a tamanho da lista Tabu prejudicou os resultados, fazendo com que apenas alcançássemos um soma de aptidões máxima de 899.



Algoritmo Tabu III

Algoritmo Tabu				
	Iterações	Combinações vizinhas	Aptidão Total	ListaTabu
1	1	[9, 2, 8, 5, 6, 10, 7, 4, 3, 1]	889	[(10, 7)]
2	2	[9, 2, 8, 7, 6, 10, 5, 4, 3, 1]	894	[(10, 7), (7, 5)]
3	3	[9, 10, 8, 7, 6, 2, 5, 4, 3, 1]	893	[(10, 7), (7, 5), (10, 2)]
4	4	[5, 10, 8, 7, 6, 2, 9, 4, 3, 1]	890	[(10, 7), (7, 5), (10, 2), (5, 9)]
5	5	[5, 10, 8, 7, 6, 3, 9, 4, 2, 1]	878	[(10, 7), (7, 5), (10, 2), (5, 9), (3, 2)]
...
11	11	[6, 3, 8, 5, 2, 9, 10, 4, 1, 7]	882	[(9, 10), (2, 6), (1, 6), (6, 7), (6, 5)]
12	12	[6, 3, 8, 10, 2, 9, 5, 4, 1, 7]	878	[(2, 6), (1, 6), (6, 7), (6, 5), (10, 5)]
13	13	[9, 3, 8, 10, 6, 2, 5, 4, 1, 7]	889	[(1, 6), (6, 7), (6, 5), (10, 5), (9, 2)]
14	14	[9, 3, 8, 6, 10, 2, 5, 4, 1, 7]	900	[(6, 7), (6, 5), (10, 5), (9, 2), (6, 10)]
15	15	[9, 2, 8, 6, 10, 3, 5, 4, 1, 7]	902	[(6, 5), (10, 5), (9, 2), (6, 10), (2, 3)]

É possível visualizar que a lista tabu possui um tamanho máximo de 4 e, quando adicionamos um novo movimento tabu, é eliminado o movimento mais antigo. É notório que o movimento tabu mais à direita da lista tabu é o movimento realizado na iteração.

Este algoritmo em apenas 15 iterações conseguiu atingir a melhor combinação possível, atingindo uma aptidão máxima de 902. A matriz será a mesma que os 2 primeiro algoritmo aqui apresentados.