

Unidade Curricular de Otimização Heurística

Trabalho Individual 2



Elaborado por:

Sebastião Manuel Inácio Rosalino, n.º 98437

Licenciatura de Ciência de Dados - 2º ano - Turna CDB1

Ano Letivo 2021/2022 - 2º Semestre

Docente:

Professora Anabela Costa

Índice

1 – Enquadramento	2
2 – Resolução	2
a) Determinação de uma solução admissível para o problema recorrendo à Heurística do Vizinho mais próximo	2
b) Estrutura de Vizinhança	5
b1) Determinação de uma solução vizinha da solução apresentada em a)	5
b2) Definição de uma Lista Tabu	7
b3) Realização de uma iteração do Algoritmo de Pesquisa Tabu	7
Referências bibliográficas	11

1 – Enquadramento

O presente trabalho tem subjacente a aplicação do “Problema do Caixeiro Viajante”, em inglês designado por “*Travelling Salesman Problem*” (TSP).

Resumidamente, é um problema que tenta determinar a menor rota possível para percorrer um conjunto de cidades (visitando uma única vez cada uma delas), retornando no final à cidade de origem. Ou seja, dado um conjunto de n cidades e as respectivas distâncias entre as mesmas, pretende-se determinar o percurso de menor distância que, partindo de uma determinada cidade, visita todas as outras cidades uma e uma só vez e regressa sempre à cidade de partida.

O Problema do Caixeiro Viajante tem aplicação em inúmeros problemas da vida real, designadamente ao nível logístico, industrial e, sobretudo, no domínio da distribuição de bens e serviços.

No caso em estudo não nos é dada informação sobre a realidade subjacente à aplicação proposta, sendo apenas indicado um conjunto de cidades (8) e a distância entre elas (em unidades de distância não especificadas).

O problema apresentado é de natureza simétrica, o que significa que a distância da cidade i para a cidade j é sempre igual à distância da cidade j para a cidade i .

Neste quadro, é solicitada a determinação de uma solução admissível a partir dos elementos fornecidos; a procura de uma solução vizinha daquela (trocando 2 vértices/cidades) e, por fim, a aplicação do Algoritmo Pesquisa Tabu.

2 – Resolução

Na resolução das várias alíneas do trabalho, assume-se que todos os percursos foram realizados seguindo o paradigma de colocar como última cidade visitada a cidade inicial do percurso.

a) Determinação de uma solução admissível para o problema recorrendo à Heurística do Vizinho mais próximo

Neste problema, determinar uma solução admissível consiste em encontrar um percurso válido com uma distância percorrida associada.

Considerou-se, de acordo com o enunciado, a seguinte instância (dados) de um Problema do Caixeiro Viajante, simétrico, com $n = 8$ cidades:

idades	1	2	3	4	5	6	7	8
1		14	15	-	-	-	-	17
2			13	14	20	-	-	21
3				11	21	17	9	9
4					11	10	8	20
5						15	18	-
6							9	-
7								13

Etapa 1:

Construção do ficheiro “Distancias.xlsx” com a instância acima transcrita, com o objetivo de utilização nos algoritmos usados na resolução do problema (*disponibilizado no ficheiro .zip*).

Etapa 2:

Leitura e visualização em código *python* do ficheiro criado na etapa 1, conforme imagem abaixo:

Código:

```
mapa_caixeiro = pd.read_excel('Distancias.xlsx')
mapa_caixeiro.index += 1
mapa_caixeiro
mapa_caixeiro.fillna('-')
```

Output:

	1	2	3	4	5	6	7	8
1	-	14.0	15.0	-	-	-	-	17.0
2	14.0	-	13.0	14.0	20.0	-	-	21.0
3	15.0	13.0	-	11.0	21.0	17.0	9.0	9.0
4	-	14.0	11.0	-	11.0	10.0	8.0	20.0
5	-	20.0	21.0	11.0	-	15.0	18.0	-
6	-	-	17.0	10.0	15.0	-	9.0	-
7	-	-	9.0	8.0	18.0	9.0	-	13.0
8	17.0	21.0	9.0	20.0	-	-	13.0	-

Etapa 3:

Devido à falta de ligações entre algumas cidades, consoante a cidade inicial pode ou não ser possível completar o percurso.

Optou-se, assim, por iniciar o percurso na cidade 3, ou seja: $K1 = 3$, com o objetivo de percorrer todas as cidades.

Etapa 4:

Procedeu-se, de seguida, à aplicação da Heurística Construtiva do “Vizinho mais Próximo”, cuja abordagem passa por visitar sempre a cidade (vértice) mais próxima da mais recente, desde que não tenha sido visitada previamente, de forma a procurar uma rota de percurso admissível.

Foi usado para o efeito o seguinte código (*in notebook*):

Heurística do “Vizinho mais Próximo” – Código em Python

```
# Leitura do ficheiro excel que tem a tabela das distâncias entre as cidades consideradas na instância do Problema
# do Caixeiro Viajante

distance_matrix = mapa_caixeiro

nodes = len(distance_matrix) # A função len() indica o tamanho do objeto

# Indicação da cidade onde começa o circuito (X0)
# Exemplo: se X0=3 então o circuito é iniciado na cidade 3

x0 = 3
if not x0:
    x0 = random.randint(1, nodes)

path = [x0] # Lista que tem a sequência de cidades que o caixeiro Viajante irá visitar

distance = 0

finish = True
while finish:
    closest_neighbour = distance_matrix.loc[path[-1], ~distance_matrix.index.isin(path)].idxmin()
    distance += distance_matrix.loc[path[-1], closest_neighbour]
    path.append(closest_neighbour)
    if len(path) == nodes:
        distance += distance_matrix.loc[path[-1], x0]
        finish = False
path.append(x0) # Adição do caminho de volta à cidade inicial

print('Distância total do Circuito Gerado:', distance, 'unidades') # Distância do circuito do Caixeiro Viajante
print()
print('Circuito gerado para o Caixeiro Viajante:')
print(path) # Circuito gerado para o Caixeiro Viajante
```

Etapa 5:

Aplicado o código, obteve-se como solução admissível o seguinte percurso:

Para $K1 = 3$, temos:

$$A = \{(3,7), (7,4), (4,6), (6,5), (5,2), (2,1), (1,8), (8,3)\}$$

Circuito: [3, 7, 4, 6, 5, 2, 1, 8, 3]

Sendo A a solução obtida.

A distância total deste circuito é igual a $102 = 9 + 8 + 10 + 15 + 20 + 14 + 17 + 9$

Comentário à solução obtida:

No circuito obtido, as distâncias aumentam com algum significado após os três primeiros percursos (a partir da cidade 6), embora se volte a reduzir na última ligação (8,3). No entanto, não é possível, apenas com a informação disponível, formular uma avaliação sobre o nível de eficiência desta solução. Para tal seria necessário avaliar soluções alternativas. Para tentar obter uma solução diferente (eventualmente com menor distância), poderia repetir-se a aplicação da heurística, mas iniciando o circuito a partir de uma outra cidade.

b) Estrutura de Vizinhaça

Uma Estrutura de Vizinhaça especifica quais as soluções que são vizinhas (alcançáveis numa única iteração) de qualquer solução atual.

Uma solução vizinha da solução atual é aquela que é alcançada através de uma operação 2-optimal (i.e., pela troca de posição, no circuito, de dois vértices/cidades, levando à obtenção de um novo circuito).

Pelo enunciado: a solução **S1** é vizinha da solução **S** se é obtida a partir de **S** através da troca de posição, no circuito, de dois vértices/cidades.

b1) Determinação de uma solução vizinha da solução apresentada em a)

Pode considerar-se esta questão no domínio das Heurísticas Melhorativas, como sendo as que partem duma solução e, a cada passo, introduzem pequenas alterações à solução de forma a tentar obter uma solução melhor.

No caso em análise, dada uma solução **S** para o TSP, uma solução **S1** pertence à vizinhaça de **S** se for obtida a partir de **S**, **trocando as posições de 2 cidades/vértices no percurso (o equivalente a remover 2 arestas não consecutivas e inserir 2 novas arestas que garantam a obtenção de um novo circuito).**

Regra:

Sejam (i, j) e (k, l) , com $j \neq k$, as arestas removidas.

Então, as arestas a serem inseridas são (i, k) e (j, l) .

A solução S (solução atual) é a seguinte:

{(3,7), (7,4), (4,6), (6,5), (5,2), (2,1), (1,8), (8,3)}

Circuito: [3, 7, 4, 6, 5, 2, 1, 8, 3] = distância 102

Considerando o conceito de vizinhança acima referido e de acordo com o solicitado, proceder-se-á à determinação de uma solução vizinha manualmente.

Para o efeito, decidiu-se retirar as arestas (7,4) e (6,5) e inserir as arestas (7,6) e (4,5), o que significa a troca de posição no circuito da cidade 4 com a 6.

Sendo um TSP simétrico, a ligação entre as cidades 4 e 6 será invertida no novo circuito.

A nova solução S1 (vizinha de S) será a seguinte:

{(3,7), (7,6), (6,4), (4,5), (5,2), (2,1), (1,8), (8,3)}

Circuito: [3, 7, 6, 4, 5, 2, 1, 8, 3] = distância 99

Ligações Removidas	Novas Ligações	Cidades Trocadas	Circuito	Distância da Solução
(7,4), (6,5)	(7,6), (4,5)	4 e 6	[3, 7, 6, 4, 5, 2, 1, 8, 3]	99 unidades

Explicações adicionais:

- Esta solução vizinha só é admissível porque existe ligação entre a cidade 7 e 6, e da cidade 4 para a 5.
- À distância total remove-se a ligação entre as cidades 7 e 4 (distância **-8**) e a ligação entre as cidades 6 e 5 (distância **-15**).
- Adiciona-se à distância total a ligação entre as cidades 7 e 6 (distância **+9**) e a ligação entre as cidades 4 e 5 (distância **+11**).
- A ligação (4,6) é invertida para (6,4), o que não altera a dimensão do percurso porque estamos em presença de um problema de natureza simétrica.
- Assim, a solução S1 encontrada é definida pelo percurso [3, 7, 6, 4, 5, 2, 1, 8, 3] com distância percorrida de 99 unidades, o que representa uma melhoria face à solução inicial S.

b2) Definição de uma Lista Tabu

Usando a metodologia acima descrita, existe a possibilidade de se selecionar soluções vizinhas já analisadas. Para precaver esta situação, utiliza-se um mecanismo, que é designado por Lista Tabu, que evita o retorno a soluções anteriores.

Por definição, em cada iteração do algoritmo adiciona-se à Lista Tabu o par de cidades trocadas que dá origem à solução vizinha obtida (movimentos tabu).

Para o caso em análise definir-se-á a Lista Tabu como uma lista de dimensão 1 contendo o par de cidades trocadas na obtenção da solução vizinha.

Se esta definição de **Lista Tabu** tivesse sido aplicada da passagem da solução inicial S para a solução vizinha S1 obtida em b1), ter-lhe-ia sido adicionado o par de **cidades 4 e 6**.

Lista tabu: cidades [4, 6]

b3) Realização de uma iteração do Algoritmo de Pesquisa Tabu

O Algoritmo de Pesquisa Tabu pode ser definido como um método de procura de uma nova solução (não necessariamente melhor do que a anterior) na vizinhança da Solução Atual – de modo iterativo – com vista a obter uma solução global do problema (ótimo global).

Este método inclui um Procedimento de Pesquisa Local aplicado dentro de um determinado espaço mais alargado de procura, que permite determinar a melhor solução vizinha da solução atual. Contudo, na Pesquisa Tabu a melhor solução vizinha pode ser uma solução pior do que a que lhe deu origem.

Esta possibilidade de selecionar soluções vizinhas piores, cria um problema: a possibilidade de haver retorno a uma solução anterior. Para evitar esta situação, a Pesquisa Tabu utiliza um mecanismo, que é designado por Lista Tabu, que evita o retorno a soluções anteriores.

Para a resolução desta alínea, assume-se que:

- A Solução Atual corresponde à solução vizinha gerada na alínea b1)
- A Lista Tabu está vazia.

Aplicação do Algoritmo de Pesquisa Tabu

- Inicialização**

Solução Inicial: {(3,7), (7,6), (6,4), (4,5), (5,2), (2,1), (1,8), (8,3)}

Circuito: [3, 7, 6, 4, 5, 2, 1, 8, 3]

Distância = 99 unidades

Lista Tabu: Vazia

Melhor Solução: {(3,7), (7,6), (6,4), (4,5), (5,2), (2,1), (1,8), (8,3)}

Solução Atual \leftarrow Solução Inicial

- Iteração 1**

Para aplicação da primeira iteração aplicou-se o código que consta no *notebook*.

i) Lista de todas as soluções vizinhas da solução atual

Output: determinação da vizinhança 2 - Optimal

	Ligações removidas	Ligações adicionadas	Cidades trocadas	Novo Circuito	Distância Percorrida
1	(3, 7) (6, 4)	(3, 6) (7, 4)	7 e 6	[3, 6, 7, 4, 5, 2, 1, 8, 3]	105.0
2	(3, 7) (4, 5)	(3, 4) (7, 5)	7 e 4	[3, 4, 6, 7, 5, 2, 1, 8, 3]	108.0
3	(3, 7) (5, 2)	(3, 5) (7, 2)	7 e 5	[3, 5, 6, 4, 7, 2, 1, 8, 3]	Não admissível
4	(3, 7) (2, 1)	(3, 2) (7, 1)	7 e 2	[3, 2, 6, 4, 5, 7, 1, 8, 3]	Não admissível
5	(3, 7) (1, 8)	(3, 1) (7, 8)	7 e 1	[3, 1, 6, 4, 5, 2, 7, 8, 3]	101.0
6	(7, 6) (4, 5)	(7, 4) (6, 5)	6 e 4	[3, 7, 4, 6, 5, 2, 1, 8, 3]	102.0
7	(7, 6) (5, 2)	(7, 5) (6, 2)	6 e 5	[3, 7, 5, 4, 6, 2, 1, 8, 3]	Não admissível
8	(7, 6) (2, 1)	(7, 2) (6, 1)	6 e 2	[3, 7, 2, 4, 5, 6, 1, 8, 3]	Não admissível
9	(7, 6) (1, 8)	(7, 1) (6, 8)	6 e 1	[3, 7, 1, 4, 5, 2, 6, 8, 3]	Não admissível
10	(7, 6) (8, 3)	(7, 8) (6, 3)	6 e 8	[3, 7, 8, 4, 5, 2, 1, 6, 3]	111.0
11	(6, 4) (5, 2)	(6, 5) (4, 2)	4 e 5	[3, 7, 6, 5, 4, 2, 1, 8, 3]	98.0
12	(6, 4) (2, 1)	(6, 2) (4, 1)	4 e 2	[3, 7, 6, 2, 5, 4, 1, 8, 3]	Não admissível
13	(6, 4) (1, 8)	(6, 1) (4, 8)	4 e 1	[3, 7, 6, 1, 5, 2, 4, 8, 3]	Não admissível
14	(6, 4) (8, 3)	(6, 8) (4, 3)	4 e 8	[3, 7, 6, 8, 5, 2, 1, 4, 3]	Não admissível
15	(4, 5) (2, 1)	(4, 2) (5, 1)	5 e 2	[3, 7, 6, 4, 2, 5, 1, 8, 3]	Não admissível
16	(4, 5) (1, 8)	(4, 1) (5, 8)	5 e 1	[3, 7, 6, 4, 1, 2, 5, 8, 3]	Não admissível
17	(4, 5) (8, 3)	(4, 8) (5, 3)	5 e 8	[3, 7, 6, 4, 8, 2, 1, 5, 3]	120.0
18	(5, 2) (1, 8)	(5, 1) (2, 8)	2 e 1	[3, 7, 6, 4, 5, 1, 2, 8, 3]	Não admissível
19	(5, 2) (8, 3)	(5, 8) (2, 3)	2 e 8	[3, 7, 6, 4, 5, 8, 1, 2, 3]	Não admissível
20	(2, 1) (8, 3)	(2, 8) (1, 3)	1 e 8	[3, 7, 6, 4, 5, 2, 8, 1, 3]	112.0

ii) Análise às soluções obtidas

Após uma iteração do Algoritmo de Pesquisa Tabu, a **Melhor Solução Vizinha** encontrada corresponde à solução de índice 11:

$\{(3,7), (7,6), (6,5), (5,4), (4,2), (2,1), (1,8), (8,3)\}$

Correspondente ao circuito:

$[3, 7, 6, 5, 4, 2, 1, 8, 3]$

Com uma distância associada de: 98 unidades.

A **Melhor Solução Conhecida** neste caso, como a distância de circuito diminuiu (passou de 99 para 98 unidades de distância), é atualizada para a solução correspondente à Melhor Solução Vizinha, ou seja, a solução:

$\{(3,7), (7,6), (6,5), (5,4), (4,2), (2,1), (1,8), (8,3)\}$, associada ao circuito:

$[3, 7, 6, 5, 4, 2, 1, 8, 3]$.

Circuito anterior: $[3, 7, 6, 4, 5, 2, 1, 8, 3]$

Novo circuito: $[3, 7, 6, 5, 4, 2, 1, 8, 3]$

A **Lista Tabu** sofrerá alterações, deixando de estar vazia. Ser-lhe-á acrescentado o par de cidades trocadas na obtenção da Melhor Solução Vizinha, ou seja, o par de cidades: 4 e 5.

Um **Movimento Tabu** é gerado quando um movimento de reposicionamento no circuito é efetuado para um determinado par de cidades. Ou seja, quando duas cidades são escolhidas para serem trocadas de posição no circuito, o movimento tabu a gerar visa proibir que essas cidades possam voltar a ser trocadas entre si.

No caso em análise, o **Movimento Tabu** pode ser definido como o movimento que conduziu à troca de posição das cidades 4 e 5 no percurso e que passou a integrar da lista tabu.

Movimento Tabu: corresponde à troca das cidades 4 e 5.

A **Solução Atual**, numa seguinte iteração, passaria a ser a Melhor Solução Vizinha da iteração efetuada, portanto a solução:

$\{(3,7), (7,6), (6,5), (5,4), (4,2), (2,1), (1,8), (8,3)\}$; correspondente ao circuito:
[3, 7, 6, 5, 4, 2, 1, 8, 3].

Se o algoritmo continuasse, a solução atual seria atualizada em todas as iterações, no entanto a melhor solução conhecida só seria atualizada caso fosse encontrada uma solução com uma distância percorrida associada menor.

Referências bibliográficas

- Documentos de apoio à UC de Otimização Heurística 2021/2022 - ISCTE
- Burke, E. K.; Kendall, G. (Eds.) (2014). Search Methodologies: Introductory Tutorials in Optimization and Decision Support, 2nd edition, Springer.
- Siarry, P. (Ed.) (2016). Metaheuristics, Springer.
- Ehrgott, M. (2005). Multicriteria Optimization, 2nd edition, Springer.
- Ragsdale, C.T. (2017). Spreadsheet Modeling and Decision Analysis: A Practical Introduction to Business Analytics. 8th Ed. Cengage Learning.