

TRABALHO INDIVIDUAL 2

OTIMIZAÇÃO HEURÍSTICA
2023/2024



MARIA FERREIRA
Nº 110986 CDB2

Índice

Introdução	3
Enunciado.....	4
Resolução	6
Alínea a).....	7
Alínea b).....	7
Alínea c).....	8
Alínea d).....	9
Alínea e).....	10
Alínea f)	10
Pseudocódigo para o algoritmo <i>Simulated Annealing</i> – Maximização da Rentabilidade	
Esperada	11
Alínea g).....	11
Implementação do Algoritmo <i>Simulated Annealing</i> – Maximização da Rentabilidade	
Esperada	12
Output.....	14
Output .txt	15

Introdução

O objetivo deste trabalho é implementar um algoritmo de *Simulated Annealing* para resolver o problema de seleção de quatro zonas florestais para corte, maximizando a rentabilidade esperada e respeitando as restrições de impacto ambiental e adjacência entre zonas.

Enunciado

Uma herdade possui uma área florestal dividida nas zonas A1, ..., A30, conforme pode ser visto no mapa da Figura 1.

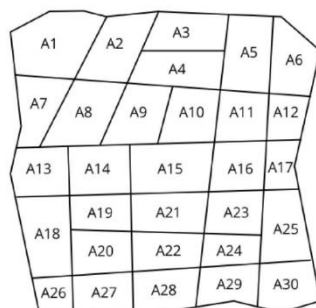


Figura 1: Mapa da área florestal

O dono da herdade pretende escolher **quatro zonas para fazer o corte de árvores** para vender a uma empresa produtora de mobiliário. Um estudo foi feito à maturidade das árvores, tendo sido concluído que não existe nenhuma zona em toda a área florestal na qual as árvores ainda não satisfaçam as exigências da empresa de mobiliário. Tendo em conta a quantidade e a qualidade das árvores, para cada zona foi feita uma estimativa do rendimento que poderá ser obtido com o seu corte. Para além disso, cada área foi avaliada através de um índice de impacto ambiental do corte, que varia entre 1 (menor impacto ambiental) e 4 (maior impacto ambiental). A informação é apresentada na tabela seguinte:

Zona	Rentabilidade Esperada	Impacto Ambiental	Zona	Rentabilidade Esperada	Impacto Ambiental
A1	2600	2	A16	2900	1
A2	3400	4	A17	1300	4
A3	1900	3	A18	3300	1
A4	2100	4	A19	3000	1
A5	3000	2	A20	4200	1
A6	3200	2	A21	4200	4
A7	3400	1	A22	2400	1
A8	4300	2	A23	4600	3
A9	4700	4	A24	1800	1
A10	1100	3	A25	4500	4
A11	4600	4	A26	2000	2
A12	4700	4	A27	1700	1
A13	1700	1	A28	1700	3
A14	5000	4	A29	1500	2
A15	1900	1	A30	1700	3

Para garantir a conservação do habitat existente nesta área florestal, as zonas escolhidas não podem ser adjacentes umas às outras. Por simplicidade, vamos assumir que duas zonas são adjacentes apenas se tiverem uma fronteira em comum. Por exemplo, a zona A15 é adjacente à zona A9, mas não é adjacente à zona A19. Assim, o dono da herdade pretende escolher as quatro zonas que maximizam a rentabilidade esperada, sem que a soma dos seus índices de impacto ambiental ultrapasse o valor 8. Na tentativa de atingir este objetivo, a companhia irá definir e implementar um algoritmo de *Simulated Annealing*.

Resolução

No enunciado, foram sombreadas as informações importantes para a resolução do problema.

Antes de se iniciar qualquer resolução, de modo a facilitar todo o processo, ir-se-á organizar numa tabela as informações de rendimento esperado, impacto ambiental e zonas que são adjacentes à respetiva zona.

Zona	Rendimento Esperado	Impacto Ambiental	Zonas Adjacentes
A1	2600	2	A2, A7
A2	3400	4	A1, A3, A4, A8
A3	1900	3	A2, A4, A5
A4	2100	4	A2, A3, A5, A9, A10
A5	3000	2	A3, A4, A6, A11, A12
A6	3200	2	A5, A12
A7	3400	1	A1, A8, A13
A8	4300	2	A2, A7, A9, A13, A14
A9	4700	4	A4, A8, A10, A14, A15
A10	1100	3	A4, A9, A11, A15
A11	4600	4	A5, A10, A12, A16
A12	4700	4	A6, A11, A17
A13	1700	1	A7, A8, A14, A18
A14	5000	4	A8, A9, A13, A15, A19
A15	1900	1	A9, A10, A14, A16, A21
A16	2900	1	A11, A15, A17, A23
A17	1300	4	A12, A16, A25
A18	3300	1	A13, A19, A20, A26
A19	3000	1	A14, A18, A20, A21
A20	4200	1	A18, A19, A22, A27
A21	4200	4	A15, A19, A22, A23
A22	2400	1	A20, A21, A24, A28
A23	4600	3	A16, A21, A24, A25
A24	1800	1	A22, A23, A25, A29
A25	4500	4	A17, A23, A24, A30
A26	2000	2	A18, A27
A27	1700	1	A20, A26, A28
A28	1700	3	A22, A27, A29
A29	1500	2	A24, A28, A30
A30	1700	3	A25, A29

Foi também criado um ficheiro Excel para que a resolução do problema seja mais prática, passe-se então à primeira alínea.

Alínea a)

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

Uma solução admissível para o problema tem de obedecer às seguintes condições:

- As 4 zonas escolhidas não podem ser adjacentes entre si (ou seja, não podem estar contidas na coluna “*Zonas Adjacentes*” de outra coluna escolhida);
- A soma dos impactos ambientais das áreas escolhidas não pode superar o valor 8 (ou seja, a soma dos valores da coluna “*Impacto Ambiental*” tem de ser menor ou igual a 8).

Alínea b)

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

O problema consiste em escolher quatro das 30 zonas florestais disponíveis, maximizando a rentabilidade e garantindo as condições referidas anteriormente. Assim, para o desenvolvimento da heurística serão seguidas as seguintes etapas:

1. Percorrer para cada zona na coluna Rendimento Esperado, ordenada de forma decrescente, e verificar se a zona pode ser adicionada à solução atual (se cumpre as condições referidas anteriormente).
2. Caso a zona satisfaça essas condições e possa ser adicionada à solução atual, continua-se a percorrer a coluna através do mesmo processo até que se tenha adicionado as 4 zonas necessárias.

Com este método, obtém-se uma solução admissível a partir da heurística, apesar de não garantir um rendimento esperado total ótimo, uma vez que, para isso, seria essencial calcular todas as combinações possíveis que respeitassem as condições e fosse calculada a soma dos rendimentos esperados para cada uma. Assim, procurar-se-ia a solução que definisse o maior valor para essa soma.

Alínea c)

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

Como referido na alínea anterior, é necessário reorganizar a tabela, tem-se então:

Zona	Rendimento Esperado	Impacto Ambiental	Zonas Adjacentes
A14	5000	4	A8, A9, A13, A15, A19
A9	4700	4	A4, A8, A10, A14, A15
A12	4700	4	A6, A11, A17
A23	4600	3	A16, A21, A24, A25
A11	4600	4	A5, A10, A12, A16
A25	4500	4	A17, A23, A24, A30
A8	4300	2	A2, A7, A9, A13, A14
A21	4200	4	A15, A19, A22, A23
A20	4200	1	A18, A19, A22, A27
A7	3400	1	A1, A8, A13
A2	3400	4	A1, A3, A4, A8
A18	3300	1	A13, A19, A20, A26
A6	3200	2	A5, A12
A5	3000	2	A3, A4, A6, A11, A12
A19	3000	1	A14, A18, A20, A21
A16	2900	1	A11, A15, A17, A23
A1	2600	2	A2, A7
A22	2400	1	A20, A21, A24, A28
A4	2100	4	A2, A3, A5, A9, A10
A26	2000	2	A18, A27
A15	1900	1	A9, A10, A14, A16, A21
A3	1900	3	A2, A4, A5
A24	1800	1	A22, A23, A25, A29
A13	1700	1	A7, A8, A14, A18
A27	1700	1	A20, A26, A28
A28	1700	3	A22, A27, A29
A30	1700	3	A25, A29
A29	1500	2	A24, A28, A30
A17	1300	4	A12, A16, A25
A10	1100	3	A4, A9, A11, A15

Iniciando-se a contagem do impacto ambiental (IA) a 0, a solução vazia $S = []$ e soma dos rendimentos esperados (RE) igualmente zerada. Inicie-se o processo definido na heurística:

1. A primeira zona que se encontra na tabela é a A14 (RE=5000, IA=4), então atualizam-se os parâmetros: IA total = 4, $S = [A14]$, RE = 5000. Prossegue-se com o pensamento;
2. A segunda zona a ser encontrada é a A9 (RE=4700, IA=4), feitos os cálculos percebe-se que caso se adicione esta zona à solução IA passa a ser 8, o que não é compatível com o objetivo pois não seria possível adicionar mais áreas sem ultrapassar o valor pedido, por isso, ignoram-se essas zonas. A próxima zona que se adiciona à solução é a A20 (RE=4200, IA=1), então atualizam-se os parâmetros: IA total = 5, $S = [A14, A20]$, RE = 9200, continua-se o processo;
3. A terceira zona a ser selecionada é a A7 (RE=3400, IA=1), então atualizam-se os parâmetros: IA total = 6, $S = [A14, A20, A7]$, RE = 12600, continua-se o processo;
4. A quarta, e última, zona a ser selecionada é a A6 (RE=3200, IA=2), então atualizam-se os parâmetros: IA total = 8, $S = [A14, A20, A7, A6]$, RE = 15800 e dá-se por terminada a pesquisa.

Tem-se então uma solução admissível onde as zonas escolhidas são A14, A20, A7 e A6, o impacto ambiental é igual a 8 e o rendimento esperado é de 15 800 unidades monetárias. Não sendo possível determinar se é ou não a solução ótima.

Alínea d)

Defina a estrutura de vizinhança de uma solução.

Uma estrutura de Vizinhança é uma função $N: S \rightarrow 2^S$ que atribui a cada solução $s \in S$, um conjunto de soluções vizinhas, $N(s)$, contido ou igual a S . O conjunto $N(s)$ designa-se por Vizinhança da Solução s .

Neste caso, s_2 é uma solução vizinha de s_1 , caso seja obtida através da troca de uma zona. Dentro das quatro selecionadas, trocar uma por outra diferente que cumpra com os objetivos (não ser adjacente a nenhuma das restantes e que a soma do impacto ambiental continue menor ou igual a 8).

Alínea e)

Tendo em conta a alínea d), determine uma solução vizinha da solução que apresentou na alínea c).

Na alínea c) tinha-se encontrado a seguinte solução:

- $IA_{total} = 8$, $S = [A14, A20, A7, A6]$, $RE = 15800$;

Por isso, para se encontrar uma solução vizinha opta-se por alterar a última zona (A6). A próxima zona é a A5 ($RE=3000$, $IA=2$) logo tem-se:

- $IA_{total} = 8$, $S = [A14, A20, A7, A5]$, $RE = 15600$.

Conclui-se que a solução vizinha encontrada não é melhor que a original, uma vez que para o mesmo impacto ambiental (8), é ultrapassada por 200 unidades monetárias no rendimento esperado pela solução original.

Alínea f)

Na tentativa de determinar uma solução, o dono da herdade irá definir e implementar um algoritmo de Simulated Annealing. Tendo em conta a estrutura de vizinhança definida na alínea d), apresente o pseudocódigo para a aplicação do algoritmo tomando como parametrização:

- $T = (t_0, t_1, t_2, t_3, t_4)$, em que $t_0 = 0.2 \times f(\text{Solução Inicial})$; $t_k = 0.5^k t_0$, $k = 1, 2, 3, 4$;
- $m_k = 5$, $k = 0, 1, 2, 3, 4$.

Pseudocódigo para o algoritmo *Simulated Annealing* – Maximização da Rentabilidade Esperada

Input:

solução inicial: s_0 (conjunto de quatro zonas iniciais, não adjacentes, e com impacto ambiental ≤ 8)
 estrutura de vizinhança: \aleph (conjunto de vizinhos gerados trocando duas zonas da solução atual)
 função de avaliação: f (rentabilidade esperada de uma solução)
 processo de arrefecimento: $T = [t_0, t_1, t_2, t_3, t_4]$ (temperaturas)
 número de iterações para cada temperatura: $M = [m_0, m_1, m_2, m_3, m_4]$ (iteraões)

Inicialização:

Solução_atual $\leftarrow s_0$
 Solução_melhor $\leftarrow s_0$
 $k \leftarrow 0$

Iteração:

Enquanto ($k < \text{tamanho}(T)$):
 $m \leftarrow 0$
 Enquanto ($m < M[k]$):
 $s \leftarrow$ solução aleatoriamente escolhida na $\aleph(\text{Solução_atual})$
 Se (soma dos índices de impacto ambiental de $s \leq 8$ e s não contém zonas adjacentes):
 $\Delta \leftarrow f(s) - f(\text{Solução_atual})$
 Se ($\Delta > 0$):
 Solução_atual $\leftarrow s$
 Se ($f(\text{Solução_atual}) > f(\text{Solução_melhor})$):
 Solução_melhor \leftarrow Solução_atual
 Caso contrário:
 $\epsilon \leftarrow$ número aleatório com distribuição uniforme em $[0, 1]$
 Se ($\epsilon < e^{(-\Delta/T[k])}$):
 Solução_atual $\leftarrow s$
 $m \leftarrow m + 1$
 $k \leftarrow k + 1$

Output:

Retornar Solução_melhor

Alínea g)

Implemente o procedimento definido na alínea f).

Implementação do Algoritmo *Simulated Annealing* – Maximização da Rentabilidade Esperada

```
import random
import math
import pandas as pd

# Função para verificar se duas zonas são adjacentes
def sao_adjacentes(zona1, zona2, df):
    adjacentes_zona1 = df.loc[df['Zona'] == zona1, 'Zonas Adjacentes'].iloc[0]
    return zona2 in adjacentes_zona1.split(', ')

# Função para verificar se um conjunto de zonas tem adjacentes
def tem_adjacentes(zonas, df):
    for i in range(len(zonas)):
        for j in range(i + 1, len(zonas)):
            if sao_adjacentes(zonas[i]["Zona"], zonas[j]["Zona"], df):
                return True
    return False

# Função para calcular a rentabilidade total de uma solução
def rentabilidade_total(solucao):
    return sum(zona["Rendimento Esperado"] for zona in solucao)

# Função para calcular o impacto ambiental total de uma solução
def impacto_ambiental_total(solucao):
    return sum(zona["Impacto Ambiental"] for zona in solucao)

# Função para gerar uma solução inicial aleatória
def gerar_solucao_inicial(df):
    while True:
        solucao = []
        while len(solucao) < 4:
            zona = random.choice(df.to_dict('records'))
            if zona not in solucao:
                solucao.append(zona)
            if tem_adjacentes(solucao, df) or impacto_ambiental_total(solucao) > 8:
                solucao = []
                break
        if len(solucao) == 4:
            return solucao

# Função para gerar uma solução vizinha trocando uma zona
def gerar_vizinho(solucao, df):
    while True:
        nova_solucao = solucao[:]
        idx = random.randint(0, 3)
        nova_zona = random.choice(df.to_dict('records'))
        if nova_zona not in nova_solucao:
            nova_solucao[idx] = nova_zona
            if not tem_adjacentes(nova_solucao, df) and impacto_ambiental_total(nova_solucao) <= 8:
                return nova_solucao
```

Implementação do algoritmo de Simulated Annealing

```
def simulated_annealing(df):
```

Definição das temperaturas e iterações

```
    solucao_inicial = gerar_solucao_inicial(df)
    t0 = 0.2 * rentabilidade_total(solucao_inicial)
    temperaturas = [t0 * (0.5 ** k) for k in range(5)]
    iteracoes_por_temperatura = [5, 5, 5, 5, 5]
```

```
    solucao_atual = solucao_inicial
    solucao_melhor = solucao_atual[:]
```

Lista para armazenar todas as soluções

```
    solucoes = pd.DataFrame(columns=['iter', 'temp', 'zonas', 'rent', 'imp', 'prob_acceptance',
'acceptance'])
    contador = 0
    solucoes.loc[contador] = {'iter': 0, 'temp': '-', 'zonas': solucao_inicial, 'rent':
rentabilidade_total(solucao_inicial), 'imp': impacto_ambiental_total(solucao_inicial),
'prob_acceptance': '-', 'acceptance': '-'}
    contador += 1
```

```
    for k in range(len(temperaturas)):
```

```
        temperature = temperaturas[k]
        iterationsk = 0
```

```
        while iterationsk < iteracoes_por_temperatura[k]:
```

```
            vizinho = gerar_vizinho(solucao_atual, df)
            delta = rentabilidade_total(vizinho) - rentabilidade_total(solucao_atual)
```

```
            accepted = False
            prob_aceitacao = 1.0
```

```
            if delta > 0:
                accepted = True
            else:
```

```
                prob_aceitacao = math.exp(delta / temperature)
                if random.random() < prob_aceitacao:
                    accepted = True
```

```
            solucoes.loc[contador] = {'iter': iterationsk, 'temp': temperature, 'zonas': vizinho,
'rent': rentabilidade_total(vizinho), 'imp': impacto_ambiental_total(vizinho),
'prob_acceptance': prob_aceitacao, 'acceptance': accepted}
            contador += 1
```

```
            if accepted:
                solucao_atual = vizinho
                iterationsk += 1
```

```
            if rentabilidade_total(solucao_atual) > rentabilidade_total(solucao_melhor):
                solucao_melhor = solucao_atual[:]
```

```
with open("output.txt", "w") as f:
```

```
    print('Solução inicial:', file=f)
    print(solucoes.head(1).iloc[:, 2:5], file=f)
    print('Resumo da aplicação do Simulated Annealing:', file=f)
    print(solucoes, file=f)
    print('Melhor solução:', file=f)
    print('Zonas:', [zona["Zona"] for zona in solucao_melhor], file=f)
    print('Impacto Ambiental:', impacto_ambiental_total(solucao_melhor), file=f)
    print('Rentabilidade Esperada:', rentabilidade_total(solucao_melhor), file=f)
```

Exibir apenas as zonas, impacto e rentabilidade da primeira solução no DataFrame

```

primeira_solucao = solucoes.iloc[0]
primeira_solucao_zonas = [zona["Zona"] for zona in primeira_solucao['zonas']]
primeira_solucao_impacto = primeira_solucao['imp']
primeira_solucao_rentabilidade = primeira_solucao['rent']
print("-----Solução inicial-----")
print("A primeira solução:", primeira_solucao_zonas)
print("O impacto ambiental é:", primeira_solucao_impacto)
print("O rendimento esperado é:", primeira_solucao_rentabilidade)

return solucao_melhor

```

```

melhor_solucao = simulated_annealing(df)

```

```

print("\n-----Melhor solução-----")

```

Exibir a melhor solução encontrada de forma organizada

```

zonas_selecionadas = [zona["Zona"] for zona in melhor_solucao]
impacto_total = impacto_ambiental_total(melhor_solucao)
rendimento_total = rentabilidade_total(melhor_solucao)

print("A melhor solução encontrada:", zonas_selecionadas)
print("O impacto ambiental é:", impacto_total)
print("O rendimento esperado é:", rendimento_total)

```

Output:

```

-----Solução inicial-----
A primeira solução: ['A15', 'A24', 'A30', 'A28']
O impacto ambiental é: 8
O rendimento esperado é: 7100

-----Melhor solução-----
A melhor solução encontrada: ['A23', 'A20', 'A6', 'A8']
O impacto ambiental é: 8
O rendimento esperado é: 16300

```

Output .txt:

```

Solução inicial:
                                zonas  rent  imp
0  [{'Zona': 'A15', 'Rendimento Esperado': 1900, ... 7100    8
Resumo da aplicação do Simulated Annealing:
      iter    temp                                zonas  rent  \
0         0      -  [{'Zona': 'A15', 'Rendimento Esperado': 1900, ... 7100
1         0 1420.0  [{'Zona': 'A15', 'Rendimento Esperado': 1900, ... 8600
2         1 1420.0  [{'Zona': 'A15', 'Rendimento Esperado': 1900, ... 7100
3         1 1420.0  [{'Zona': 'A16', 'Rendimento Esperado': 2900, ... 9600
4         2 1420.0  [{'Zona': 'A16', 'Rendimento Esperado': 2900, ... 11200
..      ...      ...                                ...      ...
889      3   88.75  [{'Zona': 'A23', 'Rendimento Esperado': 4600, ... 16100
890      4   88.75  [{'Zona': 'A23', 'Rendimento Esperado': 4600, ... 13800
891      4   88.75  [{'Zona': 'A24', 'Rendimento Esperado': 1800, ... 13300
892      4   88.75  [{'Zona': 'A23', 'Rendimento Esperado': 4600, ... 13700
893      4   88.75  [{'Zona': 'A23', 'Rendimento Esperado': 4600, ... 16300

      imp  prob_acceptance  acceptance
0         8              -           -
1         7              1.0         True
2         6          0.347727        False
3         7              1.0         True
4         5              1.0         True
..      ...              ...         ...
889      8          0.105029         True
890      8              0.0        False
891      6              0.0        False
892      7              0.0        False
893      8              1.0         True

[894 rows x 7 columns]
Melhor solução:
Zonas: ['A23', 'A20', 'A6', 'A8']
Impacto Ambiental: 8
Rentabilidade Esperada: 16300

```