

2º Ano da Licenciatura de Ciência de Dados, 2023-2024

Unidade Curricular de Optimização Heurística

Docente: Mafalda de Ponte

iscte

INSTITUTO UNIVERSITÁRIO DE LISBOA

Afonso Lourenço | Nº111487 | CDB2

Afonso Pereira | Nº111134 | CD-PL-B1

Afonso Santos | Nº111431 | CDB2

Diogo Santos | 111386 | CDB2

Guilherme Simões | Nº111604 | CDB2

Índice

a)	3
b)	3
c)	4
d)	5
e)	6
f)	6
g)	7
h)	10
i)	10

a)

Uma solução admissível para o problema consiste em alocar os enfermeiros aos procedimentos de forma que todos os procedimentos complexos sejam realizados por enfermeiros das categorias 2 ou 3, enquanto os procedimentos regulares podem ser realizados por qualquer enfermeiro. Cada procedimento requer exatamente três enfermeiros. Para além disto, é necessário garantir que nenhum enfermeiro participe em mais de cinco procedimentos durante o dia e que nenhum participe em dois procedimentos no mesmo período. Para minimizar a duração total do dia de trabalho, a alocação deve garantir que a soma das durações dos sete períodos seja a menor possível, considerando que a duração de cada período é determinada pelo procedimento mais longo realizado no mesmo.

b)

Com base na heurística desenvolvida, apresente uma solução admissível para o problema da clínica.

Heurística:

- 1) Começar com uma alocação vazia de enfermeiros para procedimentos.
- 2) Para cada período:
 - a) Selecionar os enfermeiros disponíveis para cada procedimento com base nas categorias permitidas e no número máximo de participações.
 - b) Para cada procedimento:
 - i) Encontrar todas as combinações válidas de enfermeiros que respeitem as restrições da categoria e participação.
 - ii) Se houver combinações válidas, escolher uma aleatoriamente e adicionar à alocação.
 - iii) Caso contrário, reiniciar a alocação do período.
- 3) Repetir o processo até encontrar uma alocação onde todos os procedimentos sejam realizados por enfermeiros válidos e nenhum enfermeiro participe em dois procedimentos no mesmo período.
- 4) Retornar a solução admissível.

De seguida, foi desenvolvido um código seguindo esta heurística e deu a seguinte solução admissível:

P1: ['E2', 'E3', 'E9']
P2: ['E1', 'E6', 'E7']
P3: ['E6', 'E7', 'E10']
P4: ['E1', 'E2', 'E8']
P5: ['E1', 'E3', 'E5']
P6: ['E8', 'E9', 'E10']
P7: ['E5', 'E6', 'E10']
P8: ['E7', 'E8', 'E9']
P9: ['E3', 'E5', 'E10']
P10: ['E6', 'E7', 'E8']
P11: ['E7', 'E8', 'E9']
P12: ['E5', 'E6', 'E10']
P13: ['E1', 'E2', 'E4']
P14: ['E3', 'E5', 'E9']

Levando em conta a heurística, esta é uma solução admissível porque respeita todos os critérios de admissibilidade, tendo uma duração total do dia de trabalho de 555 minutos.

c)

Para a codificação da afetação dos enfermeiros aos procedimentos, foi escolhido um cromossoma estruturado como um vetor com 42 genes. Cada gene do vetor representa um procedimento específico, determinado pela divisão inteira do índice do gene por 3. Por exemplo, o gene na posição 14, ao ser dividido por 3, resulta em 4, indicando que este gene está associado ao procedimento 4.

Além disso, o alelo de cada gene corresponde ao número do enfermeiro designado para o procedimento respetivo. Dessa forma, cada gene no cromossoma não apenas identifica o procedimento, mas também especifica qual enfermeiro será responsável por ele, permitindo uma representação clara e organizada da afetação dos enfermeiros aos procedimentos.

d)

Tendo em conta o cromossoma definido, composto por uma sequência de genes que representam a alocação de enfermeiros para diferentes turnos ao longo de uma semana, o operador crossover escolhido foi o crossover a um ponto.

Para a seleção dos pais para a operação crossover, foi utilizado o método de Seleção por Torneio. Neste processo, inicialmente selecionamos aleatoriamente k cromossomas da população. O Pai1 foi identificado como o cromossoma de melhor aptidão (ou seja, com o menor custo) entre os k cromossomas selecionados. Em seguida, realizamos outro torneio para selecionar o Pai2, garantindo que ele também seja o cromossoma de melhor aptidão entre os novos k cromossomas, mas assegurando que ele seja diferente do Pai1. Este método promove a diversidade genética e evita a convergência prematura, aumentando as chances de encontrar soluções ótimas ao longo das gerações.

O operador crossover escolhido, crossover a um ponto, consiste em selecionar aleatoriamente um ponto de divisão no cromossoma dos pais. Nesse ponto de divisão, a sequência é separada em duas partes: uma "cauda" esquerda, contendo os genes do início até o ponto de divisão, e uma "cauda" direita, com os genes do ponto de divisão até o final. Cada filho resultante receberá uma dessas "caudas" dos cromossomas dos pais, combinando características de ambos. Este método é útil para chegar a novas afetações, com o objetivo de chegar a uma solução ideal, que no caso da clínica é um dia de trabalho com tempo inferior a 8 horas.

Um exemplo deste crossover pode ser os seguintes cromossomas pais com o ponto de crossover em 4:

Pai1 = [7, 2, 4, 1, 6, 3, 10, 8, 5, 1, 2, 3, 4, 6, 9, 1, 2, 7, 9, 10, 5, 8, 7, 6, 1, 4, 3, 5, 9, 10, 3, 9, 4, 5, 10, 7, 1, 9, 6, 4, 10, 7]

Pai2 = [1, 7, 8, 2, 5, 3, 5, 7, 10, 9, 1, 6, 6, 3, 9, 4, 2, 1, 9, 8, 7, 6, 10, 5, 4, 10, 3, 7, 9, 5, 4, 8, 1, 10, 6, 9, 4, 10, 8, 1, 5, 6]

Filho1 = [7, 2, 4, 1, 5, 3, 5, 7, 10, 9, 1, 6, 6, 3, 9, 4, 2, 1, 9, 8, 7, 6, 10, 5, 4, 10, 3, 7, 9, 5, 4, 8, 1, 10, 6, 9, 4, 10, 8, 1, 5, 6]

Filho2 = [1, 7, 8, 2, 6, 3, 10, 8, 5, 1, 2, 3, 4, 6, 9, 1, 2, 7, 9, 10, 5, 8, 7, 6, 1, 4, 3, 5, 9, 10, 3, 9, 4, 5, 10, 7, 1, 9, 6, 4, 10, 7]

e)

Tendo em conta o cromossoma definido em c), uma mutação genética poderá ser a troca entre 2 enfermeiros. Serão escolhidos 2 genes aleatórios que trocam de posição no cromossoma um com o outro, criando assim um pequeno ajuste, de forma a obter uma nova solução. Sendo a mutação um processo raro, para que se faça a troca calcular-se-á um valor aleatoriamente entre 0 e 1 (que siga uma distribuição normal) e se este for menor que 0,1 (10%), proceder-se-á à respetiva mudança.

Um exemplo de mutação, obtido através de código, foi com o valor 0.052363598850944326, obtido aleatoriamente entre 0 e 1, que vai gerar uma mutação entre os genes 6 e 27:

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

Filho após Mutação = [6, 1, 8, 4, 10, 2, 10, 7, 9, 2, 4, 3, 6, 3, 1, 4, 5, 7, 7, 5, 9, 10, 6, 8, 3, 8, 9, 9, 5, 10, 1, 4, 2, 10, 8, 5, 7, 8, 5, 6, 3, 2]

f)

Os operadores propostos podem gerar soluções inadmissíveis que violam as seguintes restrições:

- Enfermeiro participar em mais de 5 procedimentos. Após crossover ou mutação, um enfermeiro pode ser atribuído a mais de 5 procedimentos.
- Enfermeiro participar em dois procedimentos no mesmo período. Após crossover ou mutação, um enfermeiro pode estar alocado em dois procedimentos que ocorrem simultaneamente.
- Procedimentos complexos realizados por enfermeiros não qualificados: O crossover pode alocar enfermeiros de categoria 1 a procedimentos complexos.

A forma de solucionar estas inadmissibilidades é por exemplo, após cada operação de crossover ou mutação, verificar as soluções para essas restrições. Se um enfermeiro for alocado a mais de 5 procedimentos, reatribuir os procedimentos excedentes a outros enfermeiros disponíveis da mesma categoria. Se um enfermeiro estiver alocado em dois procedimentos simultâneos, trocar com outro enfermeiro não alocado nesse período.

É preciso garantir que qualquer alteração nos procedimentos complexos (crossover ou mutação) mantém os enfermeiros das categorias 2 ou 3.

Foi implementada uma função que após o crossover, itera sobre os procedimentos e verifica se as restrições são satisfeitas. Se não estiverem satisfeitas, vão ser corrigidas. Foi ainda implementada uma função que verifica a alocação dos enfermeiros após a mutação e vai corrigir se não satisfazer as condições necessárias.

g)

Em seguida é apresentado o pseudocódigo para a aplicação do algoritmo.

Enquanto o tempo total da melhor_solucao > 480:

 # Selecionar pai1 e pai2 aleatoriamente através da seleção por torneio de k elementos da população

 pai_list1 = SelecionarAleatoriamente(k elementos de df_permutations)

 pai1 = pai_list1.elemento_com_menor_tempo_total

 pai_list2 = SelecionarAleatoriamente(k elementos de df_permutations)

 Enquanto pai1 for igual a pai2:

 pai_list2 = SelecionarAleatoriamente (k elementos de df_permutations)

 pai2 = pai_list2.elemento_com_menor_tempo_total

 # Realizar crossover a um ponto m entre pai1 e pai2 para gerar dois filhos

 m = Aleatório (1, num_genes - 1)

 filho1 = Crossover (pai1, pai2, m)

 filho2 = Crossover (pai2, pai1, m)

```

# Aplicar mutação com probabilidade de 0.1 em dois genes aleatórios
Se Aleatório () <= 0.1:
    MutarDoisGenes(filho1)
Se Aleatório () <= 0.1:
    MutarDoisGenes(filho2)

# Verificar e corrigir admissibilidade dos filhos
filho1 = Função que verifica a admissibilidade e corrige se o filho1 for não
admissível
filho2 = Função que verifica a admissibilidade e corrige se o filho2 for não
admissível

# Calcular tempo total para cada filho
tempo_total_filho1 = Função Calcular Tempo Total (filho1, tempos)
tempo_total_filho2 = Função Calcular Tempo Total (filho2, tempos)

# Atualizar melhor_solucao se necessário
Atualizar Melhor Solucao(filho1, tempo_total_filho1)
Atualizar Melhor Solucao(filho2, tempo_total_filho2)

# Substituir os filhos pelos piores cromossomas da população na próxima
geração
df_nextgen = df_permutations.copy()
df_nextgen [pior_tempo1] = filho1
df_nextgen [pior_tempo2] = filho2

# Iterar mais 100 vezes para procurar uma solução ligeiramente melhor se existir
Para i de 1 a 100:
    #Repete o processo acima descrito
    df_nextgen = gerar_nova_geracao(df_permutations, tempos)
    df_permutations = df_nextgen

```

Explicação do pseudocódigo:

O algoritmo genético funciona em ciclos contínuos até que a melhor solução encontrada tenha um tempo total inferior ou igual a 480. A cada iteração do ciclo, o algoritmo segue uma série de passos estruturados para evoluir as soluções e tentar encontrar a melhor possível.

Inicialmente, dois pais são selecionados aleatoriamente através de um método chamado seleção por torneio. Este método envolve a escolha aleatória de k elementos do conjunto de permutações ($df_permutations$) e a seleção do elemento com o menor tempo total entre esses k elementos. Este processo é realizado duas vezes para selecionar *pai1* e *pai2*. É importante garantir que *pai1* e *pai2* sejam diferentes, por isso, se a seleção inicial resultar no mesmo pai para ambos, uma nova seleção é realizada para *pai2*.

Depois de selecionar os pais, o algoritmo procede ao crossover para gerar dois novos filhos. O crossover é realizado em um ponto m escolhido aleatoriamente entre 1 e o número total de genes menos um ($num_genes - 1$). O crossover a um ponto implica que os genes dos pais são trocados a partir do ponto m , resultando em dois novos cromossomas, *filho1* e *filho2*.

Os filhos gerados podem então sofrer mutações. Cada filho tem uma probabilidade de 10% de sofrer uma mutação, onde dois genes aleatórios são trocados entre si. Este processo de mutação é importante para introduzir variabilidade genética na população, prevenindo a convergência prematura para soluções sub-ótimas.

Após a mutação, é necessário verificar se os filhos gerados são admissíveis. Isto significa que cada filho deve ser verificado e corrigido, se necessário, para garantir que eles satisfazem todas as restrições e requisitos do problema. A função responsável por essa verificação e correção é chamada para cada filho.

Uma vez garantida a admissibilidade, o tempo total de cada filho é calculado utilizando uma função específica que avalia o cromossoma de cada filho em relação aos tempos. Este cálculo é crucial, pois o objetivo do algoritmo é minimizar este tempo total.

Com os tempos totais dos filhos calculados, o algoritmo compara esses valores com o tempo total da melhor solução encontrada até o momento. Se o tempo total de *filho1* ou *filho2* for menor que o da melhor solução atual, a melhor solução é atualizada para refletir o novo cromossoma com o menor tempo total.

Finalmente, os piores cromossomas da população ciclo são substituídos pelos filhos gerados e avaliados. Isto é feito copiando a população atual (*df_permutations*) para uma nova estrutura (*df_nextgen*) e atualizando os cromossomas e tempos totais das piores soluções com os dos filhos. Esta substituição garante que a nova geração está preparada para a próxima iteração do ciclo.

Quanto é finalmente obtida uma solução com tempo abaixo das 8 horas, são ainda criadas 100 gerações para procurar uma solução melhor que esteja ao alcance da população usada.

h)

Implementação do procedimento definido na alínea g)

i)

A primeira melhor solução obtida não cumpria com o objetivo das 8 horas (480 minutos) esperado por parte da clínica. Como se observa, o cromossoma obtido excedia o tempo máximo ambicionado em 84min:

```
Nova melhor solução encontrada: 564 minutos
Cromossoma: [6, 1, 8, 4, 10, 7, 5, 8, 6, 7, 1, 2, 2, 8, 5, 4, 9, 3, 6, 5, 7, 8, 9, 10, 7, 1, 3, 9, 10, 6, 9, 4, 8, 6, 10, 5, 2, 5, 1, 3, 10, 4]
```

Porém, após diversas iterações do algoritmo genético observou-se uma solução válida dado o objetivo apontado pela clínica, com um tempo total de 475 minutos, 7 horas e 55 minutos. Em seguida é apresentado o cromossoma com a divisão dos enfermeiros pelos períodos e procedimentos:

```
Melhor solução final:
P1 [6, 7, 9]
P2 [2, 8, 3]
P3 [5, 9, 10]
P4 [2, 1, 6]
P5 [1, 3, 7]
P6 [5, 9, 10]
P7 [8, 10, 7]
P8 [6, 5, 9]
P9 [7, 4, 10]
P10 [6, 5, 8]
P11 [4, 8, 3]
P12 [6, 7, 10]
P13 [2, 8, 1]
P14 [3, 4, 9]
```

Após observar os resultados obtidos, pode-se concluir que foi possível encontrar uma solução que cumpre todos os requisitos da clínica: nenhum enfermeiro realiza mais de cinco turnos por dia, nenhum enfermeiro participa em dois procedimentos no mesmo período e nenhum enfermeiro realiza um procedimento para o qual não tem habilitações. Solução esta que cumpre também com o objetivo de tempo despendido desejado pela clínica: que o dia de trabalho não ultrapasse as 8 horas (475min).