



PROJETO FINAL

Descrição do problema

A empresa *Móveis Tudo* produz diversos tipos de móveis, desde mesas de escritório até cadeiras e sofás. Eles estão atualmente construindo uma nova fábrica dedicada à produção de cadeiras de escritório e desejam que essa nova unidade tenha à disposição não somente os melhores e mais eficientes equipamentos, mas também os melhores processos. No caso da nova fábrica, ela possuirá uma série de linhas de produção idênticas, cada uma capaz de produzir qualquer tipo de cadeira do catálogo da empresa. O plano de produção diário é gerado por um software da empresa que fornece a lista de produtos que devem ser produzidos a cada dia de forma a atender os pedidos em aberto de seus clientes e manter um estoque mínimo dos produtos mais populares. Entretanto, um dos maiores problemas da empresa está justamente em como distribuir entre as linhas de produção os produtos que devem ser fabricados em um mesmo dia, de forma a não só concluir toda a produção planejada, mas também terminar o mais rápido possível. Pensando em otimizar esse processo em sua nova fábrica, a *Móveis Tudo*, contratou o seu grupo para criar um algoritmo que receba como entrada a lista de produtos que devem ser produzidos em um certo dia e determine a sequência de produção em cada linha de forma que a produção termine o mais cedo possível. Formalmente, o problema pode ser descrito da seguinte forma:

“Seja n o número de produtos que devem ser fabricados em um determinado dia e m o número de linhas de produção disponíveis, todas idênticas. Cada produto $i \in \{1, \dots, n\}$ requer um tempo t_i (em minutos) para ser produzido. Entretanto, entre a produção de dois produtos na mesma linha pode ser necessário a limpeza ou a realização de alguma manutenção. Sendo assim, consideramos que s_{ij} , onde $i, j \in \{1, \dots, n\}$ e $i \neq j$, especifica o tempo (em minutos) necessário para preparar uma linha de produção para fabricar o produto j logo após ter fabricado o produto i . Nesse contexto, deseja-se obter a sequência de produtos a serem fabricados em cada linha de produção de forma a minimizar o tempo de operação da fábrica, que é dado pela linha de produção que termina sua produção por último. ”

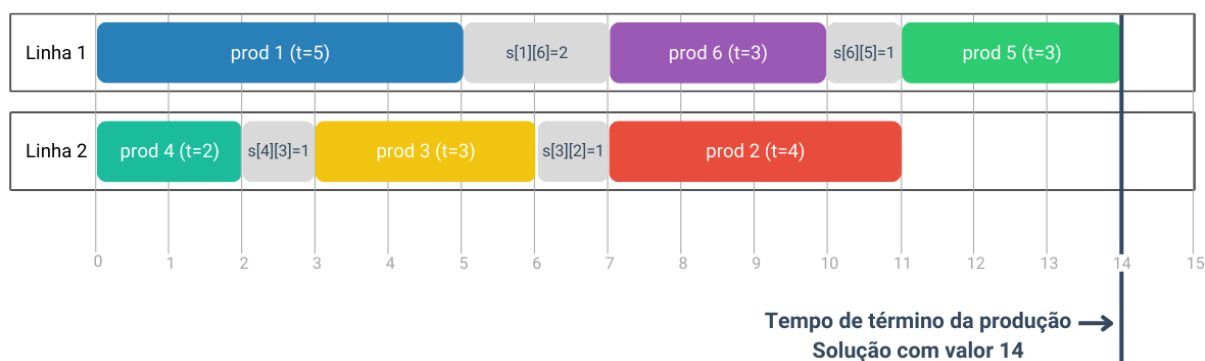
Exemplo de instância e solução

Para exemplificar o problema, considere uma instância (cenário) onde existam 2 linhas de produção e 6 produtos, com os respectivos tempos de produção dados por $t = [5, 4, 3, 2, 3, 3]$. A matriz s contendo os tempos de preparação é como a seguir:



s_{ij}	1	2	3	4	5	6
1	-	1	2	3	2	2
2	10	-	5	13	7	12
3	7	1	-	10	4	4
4	1	3	1	-	3	2
5	9	11	2	3	-	8
6	1	7	3	5	1	-

Considerando essa instância, o seguinte plano de produção é uma solução viável (não necessariamente a ótima) para o problema.



Note que, na solução acima, a linha de produção 1 termina no tempo 14, enquanto que a linha 2 finaliza no tempo 11. Nesse problema estamos interessados em otimizar o tempo de finalização da produção como um todo, determinado pela máquina que termina por último. Portanto, no exemplo, o valor da solução é 14.

Instruções

O projeto deve ser realizado em grupo de **3 integrantes** e vale 10 pontos, relativos à terceira nota da disciplina. Cada grupo deve desenvolver um algoritmo eficiente de busca local (ou meta-heurística) para o problema de otimização descrito acima. O código-fonte deve ser **obrigatoriamente** escrito na linguagem **C/C++**.

Note que o seu programa deve ser capaz de ler um arquivo contendo os dados de uma instância (cenário) do problema e utilizar tais dados como entrada para o algoritmo. O formato de arquivo a ser utilizado é o seguinte:

```
1 numero_de_linhas
2 numero_de_produtos
3
4 array t
5
6 matriz s
```



A instância utilizada na seção anterior, por exemplo, poderia ser representada pelo seguinte arquivo:

1	2						
2	6						
3							
4	5	4	3	2	3	3	
5							
6	0	1	2	3	2	2	
7	10	0	5	13	7	12	
8	7	1	0	10	4	4	
9	1	3	1	0	3	2	
10	9	11	2	3	0	8	
11	1	7	3	5	1	0	

Etapas e prazos

Este projeto contém os seguintes entregáveis:

- Implementação de **ao menos uma heurística de construção**, que nada mais é do que um **algoritmo guloso** para geração de uma solução viável.
- Implementação de **pelo menos 2 movimentos de vizinhança**.
- Implementação do algoritmo de busca local chamado VND (Variable Neighborhood Descent).
- Implementação de uma meta-heurística (OPCIONAL). Sugestões: GRASP ou ILS.
- Resultados computacionais: **criar uma tabela** que contenha os resultados obtidos pela(s) heurística(s) construtiva(s) e pelo VND, e que compare tais resultados com a solução ótima de cada instância. Essa tabela deverá conter os seguintes dados para cada heurística construtiva e para o VND:
 - Média do valor da solução (em no **mínimo 10 execuções** para cada instância caso exista algum fator aleatório no algoritmo)
 - Melhor solução encontrada
 - Média do tempo gasto pelo respectivo algoritmo
 - GAP para a solução ótima

Caso decida implementar a meta-heurística, é necessário adicionar uma coluna de resultados para ela na tabela.

O projeto deve ser entregue até às **23:59 do dia 28 de maio de 2023**. Devem ser enviados via SIGAA, o código-fonte do projeto e um relatório em *pdf* contendo o nome dos integrantes do grupo e a tabela de resultados computacionais. Note que é necessário somente uma entrega por grupo e não serão aceitos envios por e-mail.



Avaliação

Cada grupo deverá apresentar presencialmente o projeto em data a ser agendada pelo professor. A nota do projeto é individual e leva em consideração diversos critérios, como demonstração de entendimento do código na apresentação, qualidade do código, eficiência dos algoritmos implementados, qualidade dos resultados obtidos, dentre outros. Não apresentar o projeto implica em nota zero.

Dicas

Como calcular o valor da medida GAP: Suponha que desejamos calcular o valor GAP para o resultado da heurística construtiva para a instância chamada *nome_instancia*. Supondo que o valor encontrado pela heurística para essa instância é dado por $valor_{heuristica}$ e o valor ótimo para essa instância é $valor_{otimo}$, o cálculo do GAP é realizado da seguinte forma:

$$gap = \left(\frac{valor_{heuristica} - valor_{otimo}}{valor_{otimo}} \right) \times 100$$

Note que o valor do gap é dado em percentagem (%) e indica a “distância” da solução, no caso, da heurística construtiva para o valor ótimo.

Para calcular o GAP dos resultados obtidos pelo VND basta substituir $valor_{heuristica}$ pelo valor encontrado pela VND.

Exemplo de tabela de resultados:

	ótimo	Heurística construtiva			VND		
		valor solução	tempo	gap	valor solução	tempo	gap
instancia1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
instancia2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
instancia3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
instancia4	0.0	0.0	0.0	0.0	0.0	0.0	0.0