



PROJETO FINAL

Descrição do problema











Uma empresa que produz uma variedade de sucos entrou em contato para discutir um projeto focado no desenvolvimento de um algoritmo eficiente para otimizar sua produção na fábrica. Atualmente, eles operam com uma única linha de produção capaz de processar qualquer tipo de suco e acomodar diferentes tipos de embalagens. No entanto, devido a um crescimento rápido e inesperado na demanda, a empresa enfrenta dificuldades em cumprir todos os prazos e atender ao volume necessário de produção. O principal desafio é que todos os pedidos possuem prazos máximos de entrega, e o não cumprimento desses prazos resulta em multas descontadas do valor final pago pelos clientes. Após diversas reuniões, seu grupo conseguiu mapear o problema específico que precisa ser resolvido. Formalmente, o problema pode ser descrito da seguinte forma:

“Seja n o número de pedidos que devem ser processados em um determinado dia na linha de produção da fábrica. Cada pedido $i \in \{1, \dots, n\}$ requer um tempo t_i (em minutos) para ser processado e deve ser concluído até o tempo p_i . Entre o processamento de dois pedidos, pode ser necessário a limpeza e/ou adequação do tipo de embalagem. Sendo assim, consideramos que s_{ij} especifica o tempo (em minutos) necessário para preparar a linha de produção para processar o pedido j logo após o pedido i , onde $i, j \in \{1, \dots, n\}$ e $i \neq j$. Note que, caso o pedido j seja o primeiro a ser processado na linha de produção, é necessário considerar o tempo s_{0j} para preparar a linha. Caso um pedido i seja finalizado após o tempo p_i , a empresa paga uma multa por atraso $M_i = w_i * (C_i - p_i)$, onde C_i especifica o tempo de término do processamento do pedido i e w_i determina o valor da multa por minuto de atraso associada ao pedido i . O objetivo do problema é obter a sequência de processamento dos pedidos na linha de produção de forma a garantir que todos os pedidos sejam processados e que a soma das multas por atraso seja mínima.”

Exemplo de instância e solução

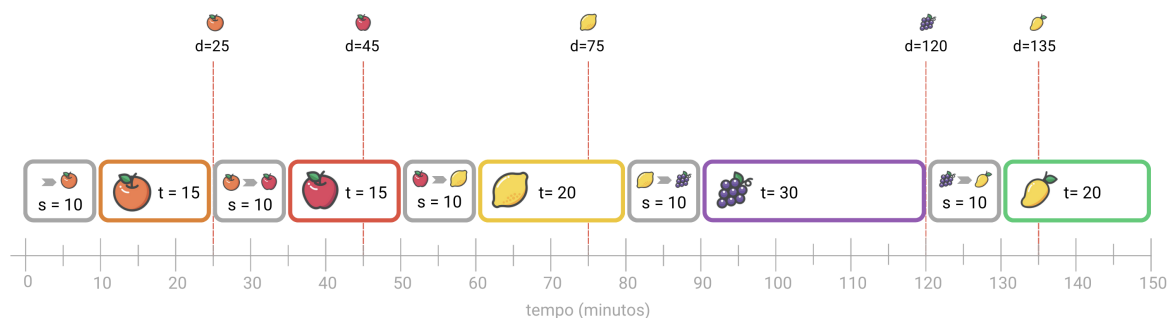
Para exemplificar o problema, considere uma instância (cenário) onde existam 5 pedidos de tipos de sucos diferentes (laranja 🍊, maçã 🍏, limão 🍋, uva 🍇 e manga 🥭), com os respectivos tempos de produção dados por $t = [15, 15, 20, 30, 20]$, prazos $p = [25, 45, 75, 120, 135]$ e valores de multa por minuto de atraso $w = [10, 12, 30, 15, 10]$. A matriz s contendo os tempos de preparação da linha de produção entre um pedido e outro é como a seguir:



s_{ij}					
-	10	15	11	10	17
	-	10	15	8	21
	10	-	10	13	9
	17	9	-	10	14
	11	13	12	-	10
	5	10	15	20	-

Note que, a primeira linha da matriz s_{ij} especifica o tempo necessário para preparar a linha de produção caso o pedido j seja o primeiro.

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



Para calcularmos o valor da solução acima é necessário calcular as possíveis multas por atraso de cada pedido usando a seguinte fórmula:

$$\text{Multa} = \max(0, \text{Valor Multa por Minuto} * (\text{Tempo de Conclusão} - \text{Prazo}))$$

Sendo assim, os valores de multa por pedido são os seguintes:

$$\text{Multa} = \max(0, 10 * (25 - 25)) = 0$$

$$\text{Multa} = \max(0, 12 * (50 - 45)) = 60$$

$$\text{Multa} = \max(0, 30 * (80 - 75)) = 150$$

$$\text{Multa} = \max(0, 15 * (120 - 120)) = 0$$

$$\text{Multa} = \max(0, 10 * (150 - 135)) = 150$$

Portanto, no exemplo, o valor da solução é $0 + 60 + 150 + 0 + 150 = 360$.



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_pedidos
2
3 array t
4 array p
5 array w
6
7 matriz s
```

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

```
1 5
2
3 15 15 20 30 20
4 25 45 75 120 135
5 10 12 30 15 10
6
7 10 15 11 10 17
8 0 10 15 8 21
9 10 0 10 13 9
10 17 9 0 10 14
11 11 13 12 0 10
12 5 10 15 20 0
```

Etapas e prazos

Este projeto contém os seguintes entregáveis:

- Implementação de ao menos um algoritmo guloso para a geração de uma solução viável.
- Implementação de pelo menos 3 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 13 de outubro de 2024**. 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 ou fora do prazo.

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:



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
Disciplina: Análise e Projeto de Algoritmos
Professor: Bruno Bruck



	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