



PROJETO FINAL

Descrição do problema

Logo após terminar a faculdade, Ana decidiu ser empreendedora e se juntou a sua amiga Paula para abrir uma startup que presta consultoria na área de Pesquisa Operacional. A empresa foca principalmente em resolver problemas de otimização combinatória para empresas de diversos setores e utiliza tanto algoritmos exatos quanto heurísticos.

Após três anos difíceis e desafiadores, a empresa está passando por uma ótima fase e vem crescendo bastante nos últimos meses com a entrada de clientes importantes. O modelo de negócios deles foca principalmente no licenciamento de soluções web desenvolvidas pela própria empresa, de forma que o cliente paga um valor inicial bem menor do que o normal, somente para financiar parte do desenvolvimento e customização, e depois pode utilizar livremente a solução pagando uma assinatura anual. Ana atribui muito do sucesso da empresa a esse modelo. Entretanto, isso gera algumas complicações. Em particular, a execução dos algoritmos de otimização propriamente ditos é o que consome a maior parte dos recursos computacionais da empresa. Todas as vezes que um cliente acessa a aplicação web e requisita uma simulação, é gerado um *job* para a execução do respectivo algoritmo. Tal *job* deve ser alocado e executado em um dos servidores da empresa.

Até o momento, todas as aplicações em funcionamento rodam em servidores locais, adquiridos e mantidos pela própria empresa. Porém, durante a última reunião de planejamento estratégico, foi repassado a Ana e Paula que, para acompanhar a demanda, seria necessário um grande investimento para atualizar e escalar a infra-estrutura. Uma alternativa apresentada foi a manter localmente a hospedagem das aplicações, mas terceirizar o processamento dos *jobs* em servidores na nuvem. Nesse caso a empresa paga somente pelo tempo de processamento consumido no servidor contratado.

O funcionamento do novo sistema foi planejado da seguinte forma. Quando um cliente submete uma requisição para a execução de uma simulação, um *job* é criado e adicionado a uma fila. Algumas vezes por dia, o sistema da empresa coleta todos os *jobs* presentes na fila e os distribui entre servidores na nuvem. Como a empresa de Ana e Paula é especialista em Otimização, eles decidiram desenvolver um algoritmo para, dado um conjunto de *jobs*, decidir a melhor alocação nos servidores disponíveis. Para compreender melhor o problema e seus requisitos, considere a seguinte definição formal:

“ Seja n o número de *jobs* e m o número de servidores na nuvem disponíveis no portfólio da empresa. Cada *job* $j \in \{1, \dots, n\}$ requer um tempo de processamento igual a t_{sj} para ser processado no servidor $s \in \{1, \dots, m\}$ e gera um custo c_{sj} . Além disso, cada servidor s , possui uma capacidade b_s , que especifica a quantidade máxima de tempo que pode ser contratada. Para cada *job* não alocado em um servidor na nuvem, vamos assumir que este será rodado localmente com um custo fixo p . Sendo assim, o objetivo do problema é o de alocar cada *job*



em um dos servidores, sem violar a capacidade dos mesmos, e de forma a minimizar o custo total para a empresa. ”

Exemplo de instância e solução

Para exemplificar o problema, considere uma instância (cenário) onde $n = 5$, $m = 2$ e $p = 1000$. O vetor de capacidades dos servidores é dado por $b = \{220, 350\}$, e as matrizes t e c são como a seguir:

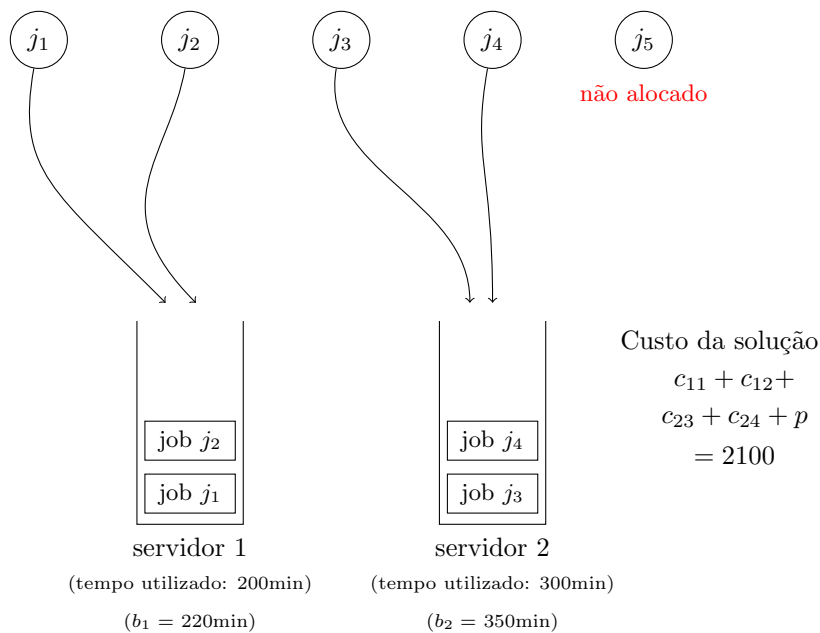
Table 1: Matriz t

	j_1	j_2	j_3	j_4	j_5
servidor 1	120	80	180	95	35
servidor 2	145	70	230	70	40

Table 2: Matriz c

	j_1	j_2	j_3	j_4	j_5
servidor 1	350	50	540	245	145
servidor 2	410	80	500	200	100

Nesse cenário, uma possível solução seria como a seguir:





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 n
2 m
3 p
4
5 array b
6
7 matriz t (servidores x jobs)
8
9 matriz c (servidores x jobs)
```

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

```
1 5
2 2
3 1000
4
5 220 350
6
7 120 80 180 95 35
8 145 70 230 70 40
9
10 350 50 540 245 145
11 410 80 500 200 100
```

Etapas e prazos

Esse 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 prazo de entrega do projeto é até às **23:59 do dia 16 de Junho de 2022**. 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.

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