



PROJETO FINAL

Descrição do problema

Uma transportadora chamada *Entrega Tudo* entrou em contato com o seu grupo a fim de negociar uma consultoria. Essa empresa opera em toda a Paraíba e realiza um alto volume de entregas de produtos de grandes empresas do setor de *e-commerce*, como a *Amazônia S.A.* e *Magazine Maria*. Em uma primeira reunião, lhes foi explicado que a *Entrega Tudo* recebe diariamente demandas de diversas empresas e deve se planejar para realizar todas as entregas dentro do prazo estipulado. Visando elevar o nível de qualidade de serviço, todas as entregas devem ser realizadas no dia seguinte ao recebimento da demanda. Sendo assim, apesar de possuírem uma frota própria de veículos, existe a possibilidade de terceirizar parte das entregas para transportadoras menores de forma a garantir que toda a demanda será satisfeita.

Atualmente, as rotas de entrega são criadas manualmente por um funcionário da empresa. Entretanto, devido ao rápido crescimento no volume de entregas, essa tarefa está ficando cada vez mais complexa e custosa. Sendo assim, a *Entrega Tudo* gostaria que vocês desenvolvessem um algoritmo capaz de gerar um plano de roteamento otimizado que vise reduzir ao máximo os custos para a empresa.

Após a reunião, seu grupo se juntou para elaborar um esboço do projeto e formalizar o problema de otimização, de forma a descrever claramente todos os requisitos e objetivos. Vocês chegaram à conclusão que, diariamente, a empresa deve resolver o seguinte problema:

“ Seja $G = (V, A)$ um grafo orientado onde, $V = \{0, 1, \dots, n\}$ é o conjunto de vértices, o vértice 0 representa o depósito, n é o número de clientes a serem atendidos, e A é o conjunto de arcos do grafo. Um arco $(i, j) \in A$ representa o caminho que um veículo deve percorrer para ir do ponto i ao j , e o custo associado a tal arco é dado por c_{ij} . Cada cliente i possui uma demanda d_i de pacotes a serem entregues. A empresa possui uma frota com k veículos, todos do mesmo modelo e com capacidade Q . Todos os veículos devem iniciar e terminar suas rotas no depósito, a capacidade máxima deve ser respeitada e cada veículo utilizado incorre em um custo r para a empresa. Visto que existe a opção de terceirizar entregas, caso um cliente i não seja atendido por nenhum veículo, deve ser pago um valor p_i para que a entrega seja realizada por outra empresa. A fim de garantir uma utilização mínima de sua frota e de seus funcionários, a empresa estabelece que ao menos L entregas devem ser realizadas sem terceirização. O objetivo do problema é encontrar o conjunto de rotas que minimize a soma do custo de roteamento (custo dos arcos), do custo associado à utilização dos veículos e do total pago com a terceirização de entregas. ”

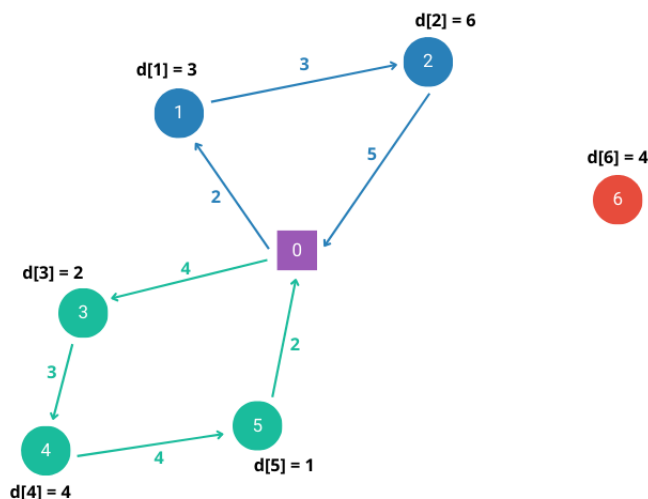


Exemplo de instância e solução

Para exemplificar o problema, considere uma instância (cenário) com um total de $n = 6$ entregas e uma frota com $k = 2$ veículos, cada um com capacidade $Q = 10$. As demandas dos clientes são dadas por $d = \{3, 6, 2, 4, 1, 4\}$, os custos de terceirização por $p = \{5, 2, 4, 4, 5, 3\}$, $L = 3$ e $r = 5$. Por fim, a matriz c contendo o custo de cada arco é como a seguir:

c_{ij}	0	1	2	3	4	5	6
0	-	2	3	4	1	9	6
1	2	-	3	2	3	2	9
2	5	5	-	5	13	7	5
3	7	3	1	-	3	4	11
4	6	5	3	1	-	4	15
5	2	8	11	2	3	-	8
6	7	10	7	10	15	8	-

Considerando essa instância, uma possível solução (não necessariamente a ótima) seria a seguinte:



Na solução acima, ambos os veículos foram utilizados e nenhum teve sua capacidade excedida. No caso da rota azul, o veículo partiu do depósito com um total de $d_1 + d_2 = 3 + 6 = 9$ pacotes, enquanto que, na verde, o veículo iniciou a rota com $d_3 + d_4 + d_5 = 2 + 4 + 1 = 7$ pacotes. Além disso, note que, como foram feitas um total de 5 entregas, o limite mínimo L também foi respeitado. Somente o cliente 6 não foi visitado, o que significa que essa entrega foi terceirizada a um custo $p_6 = 3$. Sendo assim, o custo total dessa solução é dado por:



$$\underbrace{2+3+5}_{\text{Custo da rota azul}} + \underbrace{4+3+4+2}_{\text{Custo da rota verde}} + \underbrace{5+5+3}_{\text{Custo da utilização de ambos os veículos}} + \underbrace{\text{Terceirização da entrega do cliente 6}}_{\text{Terceirização da entrega do cliente 6}} = \underline{\underline{36}}$$

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 do problema e utilizar tais dados como entrada para o algoritmo. O formato de arquivo a ser utilizado é o seguinte:

```
1 n
2 k
3 Q
4 L
5 r
6
7 array d
8
9 array p
10
11 matriz c
```

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

```
1 6
2 2
3 10
4 3
5 5
6
7 3 6 2 4 1 4
8
9 5 2 4 4 5 3
10
11 0 2 3 4 1 9 6
12 2 0 3 2 3 2 9
13 5 5 0 5 13 7 5
14 7 3 1 0 3 4 11
15 6 5 3 1 0 4 15
16 2 8 11 2 3 0 8
17 7 10 7 10 15 8 0
```



Ao final da execução, seu código deve produzir um arquivo de saída, no seguinte formato, contendo a melhor solução encontrada:

```
1 <valor total da solucao>
2 <custo de roteamento>
3 <custo associado a utilizacao dos veiculos>
4 <custo de terceirizacao>
5
6 <lista de clientes terceirizados>
7
8 <numero de rotas>
9 <rota 1>
10 <rota 2>
11 ...
```

Por exemplo, a solução mostrada na seção anterior geraria o seguinte arquivo de saída:

```
1 36
2 23
3 10
4 3
5
6 6
7
8 2
9 1 2
10 3 4 5
```

Etapas e prazos

Este projeto contém os seguintes entregáveis:

1. Implementação de **ao menos uma heurística de construção**, que nada mais é do que um **algoritmo guloso** para a geração de uma solução viável.
2. Implementação de **pelo menos 3 (três) estruturas de vizinhança**, sendo que:
 - A primeira deve realizar movimentos envolvendo uma única rota;
 - A segunda deve realizar movimentos envolvendo múltiplas rotas;
 - A terceira deve ser capaz de lidar com entregas terceirizadas.

Observação: Todas as estruturas devem realizar uma busca exaustiva na vizinhança, o que significa que devem verificar todas as possíveis combinações.

3. Implementação do algoritmo de busca local **VND** (Variable Neighborhood Descent).
4. Implementação de uma **meta-heurística** (OPCIONAL). Sugestões: GRASP ou ILS.



5. 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:

- Melhor solução encontrada;
- Média do tempo gasto pelo respectivo algoritmo;
- GAP para a solução ótima.
- Caso exista algum fator aleatório no algoritmo, incluir a média do valor da solução, do tempo de execução e da métrica GAP de no **mínimo 10 execuções** para cada instância.

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

6. Todas as implementações devem vir acompanhadas de um arquivo *makefile* para a compilação. Tal arquivo deve ser preparado de forma a funcionar em sistemas UNIX.

7. Criar uma pasta contendo os arquivos de saída gerados durante os testes com cada instância. Favor incluir somente os resultados dos testes finais, com a versão a ser entregue.

IMPORTANTE: O projeto deve ser entregue até às **23:59 do dia 29 de outubro de 2023** (Não haverá adiamento do prazo de entrega). Devem ser enviados via SIGAA, em um arquivo compactado, o **código-fonte** do projeto, o arquivo **makefile**, os arquivos de saída contendo as soluções encontradas, 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 uma instância qualquer. 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:

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



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



Note que o valor do gap é dado em porcentagem (%) 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 $valor_{heurística}$ pelo valor encontrado pelo 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