

PUC-Rio  
Departamento de Informática  
Profs. Marcus Vinicius S. Poggi de Aragão  
Período: 2006.1  
20 de junho de 2006  
Horário: 2as-feiras e 4as-feiras de 11 às 13 horas

## PROJETO E ANÁLISE DE ALGORITMOS (INF 2926)

### 2º Trabalho Prático (T2)

- O objetivo do 2º Trabalho é a implementação de um algoritmo de *branch-and-bound* para um problema NP-difícil. Isto é, um problema de otimização cuja versão de decisão é um problema NP-completo.

Este trabalho deve ser feito em grupos de no MÁXIMO 4(QUATRO) alunos. Caso contrário não será considerado.

- O problema sobre o qual deve-se aplicar o algoritmo de *branch-and-bound* segue:
  - Problema de Roteamento de Veículos com Restrição de Capacidade (CVRP): Dado um grafo  $G = (V \cup \{r\}, E)$ , custos  $c(i, j) \geq 0$  associados às arestas  $(i, j)$  em  $E$ , demandas  $q(v)$  associadas aos vértices  $v$  em  $V$ , uma capacidade  $Q$  e um número de veículos  $K$ . Deseja-se encontrar  $K$  ciclos (ou rotas) que iniciam e terminam no vértice  $r$  onde a soma das demandas dos vértices na rota não excedam  $Q$ , cujo custo total de suas arestas (nas  $K$  rotas) seja **mínimo**.

As instâncias estão nos links CVRP-1 e CVRP-2 da página de PAA 06.1.

A apresentação do seu algoritmo deve conter uma descrição de cada um dos elementos em um *branch-and-bound*, são eles:

- Critério de particionamento do espaço de soluções. Conforme apresentado em aula o critério a ser utilizado é o de particionar em dois conjuntos: um onde uma das rotas contém obrigatoriamente uma dada aresta; e outro onde esta aresta não está em nenhuma das rotas.
- Uma relaxação do problema, isto é, uma função que obtém um valor garantidamente menor ou igual (minimização) ao da solução ótima do subconjunto considerado do espaço de soluções (o subconjunto pode ser, e é no início, o conjunto de todas as soluções possíveis). Neste item, mostre como a sua relaxação é modificada quando se considera apenas um subconjunto do espaço de soluções (ou seja, quando um elemento tem seu valor fixado, isto é, uma aresta faz obrigatoriamente parte da solução ou não faz parte da solução).
- Este item **NÃO É OBRIGATÓRIO**. Um método para obter uma “boa” solução viável (que seria a melhor solução conhecida antes de iniciar o *branch-and-bound*). Esta solução pode ser obtida por um método guloso, por exemplo.

- Critério de seleção do particionamento a ser feito em cada nó da árvore de busca (ou seja, a critério de escolha da aresta a ser fixada).
- Critério de percorrimento da árvore de busca (sugere-se que seja busca em profundidade, permitindo uma implementação recursiva e mais simples de ser feita).

Apresente sempre a sua melhor solução (lista das arestas com seus respectivos valores) e o seu valor total. Caso o tempo de CPU ultrapasse 1 hora, faça com que seu algoritmo termine imprima a melhor solução encontrada até então.

Deverá ser apresentado um relatório sobre as experiências computacionais comentando os resultados obtidos. Este relatório deverá conter:

- Complexidade do Problema CVRP (prova de que é NP-difícil);
- Uma tabela com o valor da melhor solução obtida, indicando se é ótima (provado pelo seu algoritmo) ou não, o tempo de cpu total utilizado na resolução, o valor do limite inferior obtido no nó raiz, e o valor da solução ótima (ou melhor conhecida) que serão fornecidos. A tabela deverá ter uma linha para cada uma das 40 instâncias contidas nos arquivos nos links;
- Uma análise dos resultados com relação à complexidade assintótica do algoritmo implementado. Mostrar o crescimento do tempo é exponencial em função do tamanho da instância;
- Uma análise separada das diferentes etapas do algoritmo.

Os códigos (comentados) devem ser entregues eletronicamente apenas. Um roteiro para o documento a ser entregue segue:

- Descrever os algoritmos informalmente.
- Demonstrar o entendimento do algoritmo explicando, em detalhe, o resultado que o algoritmo deve obter e justificá-lo.
- Explicar a fundamentação do algoritmo e justificar a sua corretude. Apresentar e explicar a complexidade teórica esperada para cada algoritmo.
- Documente o arquivo contendo o código fonte de modo que cada passo do algoritmo esteja devidamente identificado e deixe claro como este passo é executado.

A corretude código será testada sobre um conjunto de instâncias que será distribuído. O trabalho entregue deve conter:

- Um documento contendo o roteiro de desenvolvimento dos algoritmos (e dos códigos), os itens pedidos acima, comentários e análises sobre a implementação e os testes realizados (papel).
- Envie um e-mail contendo um arquivo . zip com os códigos fonte e os executáveis correspondentes (mudar a extensão de .zip para .zxx) para **marcus.poggi@gmail.com** com o ASSUNTO (ou SUBJECT) PAA061T2.