

PUC-Rio
Departamento de Informática
Prof. Marcus Vinicius S. Poggi de Aragão
Período: 2008.2
10 de dezembro de 2008
Horário: 2as-feiras de 13 às 16 horas
Entrega: 3a.-feira 16 de dezembro de 2008, 23hs

PROJETO E ANÁLISE DE ALGORITMOS (INF 2926)

2º Trabalho Prático (T2)

- O objetivo do 2º Trabalho é a implementação de algoritmos em grafos. Estes trabalhos devem ser feitos em grupos de no MÁXIMO 5(CINCO) alunos. Caso contrário não será considerado.

- Trabalho 2

Este trabalho consiste em implementar algoritmos para encontrar caminho-mais-curto em grafos e utilizá-los em algoritmos para resolver o problema de fluxo a custo mínimo. O problema de fluxo máximo também pode ser resolvido por um algoritmo que utiliza o problema de caminho-mais-curto, qual ?

O algoritmo que devem ser implementados seguem.

Algoritmo 1: SSP

Passo 1 *Inicialização*

Encontre um pseudo-fluxo dual viável (x, p)

- *Faça $x = 0$, i.e. faça o fluxo ser nulo em todos os arcos;*
- *Entontre uma árvore de caminho-mais-curto a partir de qualquer vértice do grafo, utilizando os custos dos arcos como distâncias;*
- *Faça as variáveis duais p , uma para cada vértice, assumir valores tais que todos os arcos na árvore de caminho-mais-curto encontrada tenham custo reduzido zero, i.e. $\bar{c}_{ij} = c_{ij} - p_i + p_j$, ou seja faça $p_i = d(i)$ para todo $i \in V$, onde $d(i)$ é a distância do caminho-mais-curto do vértice inicial até o vértice i .*

Passo 2 *Iteração*

Enquanto existe um vértice com um excesso positivo, i.e. $S \neq \emptyset$

- *$S = \{i : e(i) > 0\}; T = \{j : e(j) < 0\};$*
- *Escolha um par de vértices s e t , $s \in S$ e $t \in T$, tal que existe um caminho de s a t em $G(x)$;*
- *Determine as distâncias dos caminhos mais curtos, $d(j)$, entre s e os demais vértices em $G(x)$ onde os arcos têm custos $c(p)$;*

- Atualize $p = p - d$;
- Envie o máximo de fluxo no caminho de custo reduzido zero de s a t e atualize x ;
- Atualize a rede residual $G(x)$, atualize também S e T ;

Algoritmo 2: ACM

Passo 1 Inicialização

Encontre um fluxo primal viável x

- Crie um vértice artificial s insira arcos deste vértice para todo vértice com fluxo entrante positivo. A capacidade deste arco deve ser exatamente o valor deste fluxo entrante;
- Crie um vértice artificial t insira arcos de todo vértice com fluxo entrante negativo para este vértice t . A capacidade deste arco deve ser exatamente o simétrico do valor deste fluxo entrante negativo;
- Encontre um fluxo máximo de s para t . Se este fluxo for inferior à soma de todos os fluxos entrantes o problema é inviável. Caso contrário determine x atribuindo a cada variável x_{ij} o valor assumido no fluxo máximo obtido.

Passo 2 Iteração

Enquanto existe um ciclo de valor negativo em $G(x)$

- Determine um ciclo negativo em $G(x)$;
- Encontre o arco de menor capacidade neste ciclo em $G(x)$, seja δ esta capacidade;
- Atualize x . I.e., para cada arco (i, j) no ciclo negativo atualize x_{ij} somando δ caso esteja no ciclo de forma direta, ou subtraindo δ no caso de aparecer no sentido inverso no ciclo;

As experiências com os algoritmos acima devem ser feitas sobre as instâncias disponíveis na página do curso.

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

- Uma análise da complexidade dos algoritmos acima;
- Uma tabela com o valores dos fluxos encontrados (de custo mínimo), seu respectivos tempos de cpu: total e utilizado resolvendo problemas de caminho-mais-curto, e o número de vezes que um problema de caminho-mais-curto (especificando qual) foi resolvido;
- Uma análise dos resultados com relação à complexidade assintótica do algoritmo implementado;
- 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 o conjunto de instâncias. 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 **poggi@inf.puc-rio.br** com o ASSUNTO (ou SUBJECT) PAA082T2.