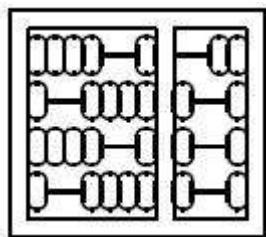


Universidade Estadual de Campinas – UNICAMP



Instituto de Computação – IC

MC548 – Projeto e Análise de Algoritmos II

Professor Zanoni – Turma B



Trabalho Prático: Relatório

Marcelo da Silva Madeira – RA:016747

1º de Julho de 2005

1 Problema do Equicorte Mínimo

1.1 Descrição das Variáveis do Modelo

As variáveis usadas foram:

- *pertence*: Um vetor booleano de vértices, que indica se um certo vértice *pertence(i)* está (valor 1) ou não (valor 0) no conjunto U^* . Ou seja, indica quais vértices fazem parte do corte;
- *equicorte*: Uma matriz booleana, que indica se o aresta *equicorte(i,j)* faz parte (valor 1) ou não (valor 0) do corte.

1.2 Descrição das Restrições

$$\left\lfloor \binom{n}{2} \right\rfloor = \sum_{i:1}^n \text{pertence}(i)$$

-De acordo com o enunciado, garanta que o conjunto U^* tenha cerca de metade do número de vértices do grafo G original.

$$\text{equicorte}(i, j) \leq |\text{pertence}(i) - \text{pertence}(j)|, \forall i, j, \text{ tais que arestas}(i, j) = 1$$

-Garanta que cada aresta do corte possuirá um vértice em cada conjunto. Inclui checagem de existência da aresta.

1.3 Descrição da Função Objetivo

$$\text{Objetivo} = \sum_{i:1}^n \sum_{j:1}^n \text{equicorte}(i, j)$$

Devemos minimizar esta função. Este somatório duplo nos dará o número de arestas presentes no equicorte mínimo.

2 Problema Anual da Coleta de Lixo

2.1 Descrição das Variáveis do Modelo

As variáveis usadas foram:

- *cobre*: Matriz de valores booleanos que mostra qual setor é atendido por qual depósito. Ex: *cobre*(2,3)=0 indica que o setor 2 não é atendido pelo depósito 3.
- *deposito*: Vetor de valores booleanos indicando se o depósito foi instalado. Ex: *deposito*(4)=1 indica que o depósito 4 foi instalado.

2.2 Descrição das Restrições

$$\sum_{i:1}^n (cobre(i, j) * req(i)) \leq capacidade(j), \forall j, j \in \{1, \dots, n\}$$

-Garante que a capacidade do depósito seja no máximo igual à requisitada *req*(*i*).

$$\sum_{j:1}^m cobre(i, j) = 1, \forall i, i \in \{1, \dots, n\}$$

-Garante que a cada setor seja alocado apenas um depósito

$$deposito(j) \geq cobre(i, j), \forall i, j, i \in \{1, \dots, n\} \text{ e } j \in \{1, \dots, m\}, \text{ tal que } \exists distancia(i, j)$$

-Faz com que haja alocação de um setor a um depósito, apenas se este depósito já estiver instalado. Checa também se é possível fazer tal alocação.

2.3 Descrição da Função Objetivo

$$Objetivo = \sum_{i:1}^n \sum_{j:1}^m (req(i) * cobre(i, j) * distancia(i, j) * custo_{km} * 1000) + \sum_{j:1}^m (deposito(j) * custo_{fixo}(j))$$

Devemos calcular a fórmula acima. Ela é um somatório do custo total de operação. Inclui custo de instalação de um depósito e custo do transporte do lixo entre os setores e depósitos aos quais estão alocados. A multiplicação por 1000 na fórmula se deve ao fato da capacidade anual de produção de lixo estar em milhares de metros cúbicos ao passo que o custo por quilômetro está por metro cúbico de lixo transportado.

3 Problema da Cadeia de Restaurantes

3.1 Estratégia 1

3.1.1 Descrição das Variáveis do Modelo

As variáveis usadas foram:

- *cadeias*: É uma matriz booleana que indica se há um restaurante da cadeia j no local i . Ex: *cadeias*(3,1) indica que há um restaurante da cadeia 1 no local 3.

3.1.2 Descrição das Restrições

$$cadeias(i,1) + cadeias(j,1) + Menos5(i,j) \leq 2, \forall i, j, i \in \{1, \dots, n\} \text{ e } j \in \{1, \dots, m\}$$

-Garante que não tenha dois restaurantes da cadeia 1 em uma distância inferior a 5km.

$$cadeias(i,2) + cadeias(j,2) + Menos5(i,j) \leq 2, \forall i, j, i \in \{1, \dots, n\} \text{ e } j \in \{1, \dots, m\}$$

-Garante que não tenha dois restaurantes da cadeia 2 em uma distância inferior a 5km.

$$cadeias(i,2) + cadeias(i,1) \leq 1, \forall i, j, i \in \{1, \dots, n\} \text{ e } j \in \{1, \dots, m\}$$

-Garante que haja apenas um restaurante em cada local.

3.1.3 Descrição da Função Objetivo

$$Objetivo1 = \sum_{j:1}^n (lucro(j,1) * cadeias(j,1))$$

$$Objetivo = Objetivo1 + \sum_{j:1}^n (lucro(j,2) * cadeias(j,2))$$

A função Objetivo, que deve ser maximizada, foi aqui dividida em 2 partes para melhor entendimento. A primeira (*Objetivo1*) calcula o lucro obtido por restaurantes da cadeia 1. A segunda calcula o lucro da cadeia 2 e soma com o valor de *Objetivo1*, dando o valor total lucrado pela rede no mês.

3.2 Estratégia 2

3.2.1 Descrição das Variáveis do Modelo

As variáveis usadas foram:

- *cadeias*: É uma matriz booleana que indica se há um restaurante da cadeia j no local i . Ex: *cadeias*(3,1) indica que há um restaurante da cadeia 1 no local 3.

3.2.2 Descrição das Restrições

$$cadeias(i,1) + cadeias(j,1) + Menos5(i,j) \leq 2, \forall i, j, i \in \{1, \dots, n\} e j \in \{1, \dots, m\}$$

-Garante que não tenha dois restaurantes da cadeia 1 em uma distância inferior a 5km.

$$cadeias(i,2) + cadeias(j,2) + Menos5(i,j) \leq 2, \forall i, j, i \in \{1, \dots, n\} e j \in \{1, \dots, m\}$$

-Garante que não tenha dois restaurantes da cadeia 2 em uma distância inferior a 5km.

$$cadeias(i,2) + cadeias(i,1) \leq 1, \forall i, j, i \in \{1, \dots, n\} e j \in \{1, \dots, m\}$$

-Garante que haja apenas um restaurante em cada local.

$$cadeias(i,1) - \sum (cadeias(1,2) * Menos5(i,1)) \leq 0, \forall i, i \in \{1, \dots, n\}$$

$$cadeias(i,2) - \sum (cadeias(1,1) * Menos5(i,1)) \leq 0, \forall i, i \in \{1, \dots, n\}$$

-Garante que caso haja instalado um restaurante de uma cadeia, um da outra cadeia será instalado em uma distância não-superior à 5km. *Obs: Esta é a única restrição diferente em relação à estratégia 1.

3.2.3 Descrição da Função Objetivo

$$Objetivo1 = \sum_{j:1}^n (lucro(j,1) * cadeias(j,1))$$

$$Objetivo = Objetivo1 + \sum_{j:1}^n (lucro(j,2) * cadeias(j,2))$$

A função Objetivo, que deve ser maximizada, foi aqui dividida em 2 partes para melhor entendimento. A primeira (*Objetivo1*) calcula o lucro obtido por restaurantes da cadeia 1. A segunda calcula o lucro da cadeia 2 e soma com o valor de *Objetivo1*, dando o valor total lucrado pela rede no mês.

4 Problema da Distribuição de Combustíveis

4.1 Descrição das Variáveis do Modelo

As variáveis usadas foram:

- *carrega*: Matriz nXm de valores inteiros que mostra quanto do tipo *m* de combustível está sendo carregado no compartimento *n*.
- *qual*: Matriz nXm de valores booleanos que mostra se o combustível *m* está no compartimento *n*.

4.2 Descrição das Restrições

$$carrega(i,j) \leq capacidade(i) * qual(i,j), \forall i, j, i \in \{1, \dots, n\} e j \in \{1, \dots, m\}$$

-Garante que um tanque (compartimento) não carregue mais do que sua capacidade.

$$\sum_{i:1}^n qual(i, j) = 1, \forall i, i \in \{1, \dots, n\}$$

-Garante que só haverá um tipo de combustível em cada compartimento, evitando a adulteração do mesmo.

$$\sum_{i:1}^n carga(i, j) \leq demanda(i), \forall j, j \in \{1, \dots, m\}$$

-Garante que a quantidade transportada será no máximo igual a demanda

4.3 Descrição da Função Objetivo

$$Objetivo = \sum_{i:1}^m demanda(i) - \sum_{i:1}^n \sum_{j:1}^m carga(i, j)$$

A função Objetivo nos informa a quantidade total de combustível que deixou de ser transportada. É a diferença da soma da demanda dos combustíveis pela soma de tudo o que pôde ser transportado.