

MC548 – Analise de Algoritmos 2

Fábio Maki Yamashiro
Nelson Waldemir Corrocher Filho

RA:008623
RA:009512

01 de Julho de 2005

1 Exercício 1

1.1 Variáveis

Recebemos na entrada a variável inteira n e *matriz aresta*, que indicam respectivamente o número de vértices do grafo, e as arestas existentes.

Criamos duas variáveis: u que indica os vértices do *shore* U da vizinhança, e du , que representa a vizinhança $\delta(U)$. Finalmente, temos a variável objetivo, que nada mais será que o tamanho do corte $\delta(U)$.

1.2 Restrições

Primeiro tornamos a matriz arestas simétrica (grafo não-dirigido). Embora desnecessário, facilita o trabalho sem incorrer em mudanças na solução.

A primeira restrição é muito simples, indica que uma aresta só poderá estar no corte se estiver no grafo original. Para isso, basta que a variável que indica se existe uma aresta na vizinhança seja menor ou igual à respectiva no grafo original:

$$\forall i, j \quad du(i; j) \leq arestas(i; j)$$

Também foi nos dito que o tamanho do conjunto U é a metade do tamanho do grafo. Esta restrição é *straight-forward*:

$$\sum_{i=1}^n u(i) = \left\lfloor \frac{n}{2} \right\rfloor$$

As próximas restrições servem a garantir que, para cada par de vértices, se ambos estão no conjunto U , então não há arestas entre eles no corte $\delta(U)$. Também padroniza que uma aresta $(i; j)$ do corte garante que i está em U e j está em U' .

$$\forall i, j \quad u(i) + u(j) + du(i, j) \leq 2$$

$$\forall i, j \quad du(j) + du(i, j) \leq 1$$

Perceba que isto ainda permite que se i e j estão em U' , exista a aresta $du(i; j)$. A próxima restrição trata disto. Ela garante que a aresta existe no corte se i está em U e j está em U' .

$$\forall i, j \quad arestas(i, j) \cdot (u(i) - u(j)) \leq du(i, j)$$

O que nos garante que $du(i; j)$ não será colocada mesmo que isto não esteja satisfeito é que minimizaremos du . A função objetivo é dada por:

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

2 Exercício 2

2.1 Variáveis

Recebemos na entrada as variáveis de entrada n (número de setores), m (número de locais potenciais), $custo_fixo(j)$ em reais, $distancia(i, j)$ em km., $capacidade(j)$ e $req(i)$ em milhares de metros cúbicos.

Criamos duas variáveis binárias: $x(j)$ igual a 1 se o local j possui um depósito ou zero caso contrario e $y(i, j)$ igual a 1 se o depósito em j atende o setor i ou zero caso contrario.

2.2 Restrições

O enunciado do problema nos dá duas restrições:

A primeira, diz respeito a lei ambiental em cada setor pode somente ser atendida por um único depósito:

$$\forall i \sum_{j=1}^m y(i, j) = 1$$

A segunda é que cada depósito não pode ultrapassar seus respectivos limites de capacidade, podendo ser atendida pela seguinte restrição:

$$\forall j \sum_{i=1}^n (y(i, j) \cdot req(i)) \leq capacidade(j)$$

Uma terceira restrição foi imposta para relacionar as duas variáveis binárias porque um depósito em j só pode atender um setor i se haver um depósito no local j :

$$\forall i, j \quad y(i, j) \leq x(j)$$

2.3 Objetivo

A função Objetivo dá-se pelo somatório dos custos fixos de implementação dos depósitos mais o somatório do custo variável de transporte do lixo dos setores para os depósitos que os atendem:

$$\text{Objetivo} = \sum_{j=1}^m (\text{custo_fixo}(j) \cdot x(j)) + \sum_{i=1}^n \sum_{j=1}^m (1000 \cdot \text{req}(i) \cdot \text{distancia}(i, j) * \text{custo_km} \cdot y(i, j))$$

3 Exercício 3

3.1 Variáveis

Além das variáveis de entrada n (quantidade de pontos), $lucro$ (lucro de cada cadeia em cada ponto) e $Menos5$ (se a distância é menor que 5km ou não), criamos dois vetores: $cadeia1$ e $cadeia2$, binários, que indicam se há um restaurante respectivamente *fast-food* ou *à la carte* em cada ponto.

3.2 Restrições

A primeira restrição é correspondente à primeira lei de Newton: não podem haver dois restaurantes no mesmo lugar:

$$\forall i \quad \text{cadeia1}_i + \text{cadeia2}_i \leq 1$$

A segunda restrição garante que dois restaurantes de uma mesma cadeia não podem estar a menos de 5km. Para isso, multiplicamos o fato de dois pontos estarem ou não a essa distância pela existência de restaurantes de uma mesma cadeia nesses pontos. O resultado não pode ser maior que um, o que implicaria que há dois restaurantes a menos de 5km.

$$\forall i, j \quad (\text{cadeia1}_i) + (\text{cadeia1}_j) \cdot \text{Menos5}_{ij} \leq 1$$

$$\forall i, j \quad (\text{cadeia2}_i) + (\text{cadeia2}_j) \cdot \text{Menos5}_{ij} \leq 1$$

3.3 Objetivo

A função objetivo é dada pela soma dos lucros. Que $_{e}$ a quantidade de lucro de um lugar multiplicada pela existência de um restaurante naquele lugar:

$$\text{Objetivo} = \sum_{i=1}^n (\text{cadeia1}_i \cdot \text{lucro1}_i + \text{cadeia2}_i \cdot \text{lucro2}_i)$$

3.4 Parte b

Para a parte b, só precisamos criar uma nova restrição, a função objetivo e as outras restrições mantêm-se inalteradas. Se existe um restaurante *fast-food* em um ponto, deve haver um *à la carte* a menos de 5km. Ou seja, o número de restaurantes de uma cadeia a 5km de um ponto deve ser maior ou igual à existência de um restaurante da outra cadeia nesse ponto, garantindo que um restaurante sempre tem outro da outra rede a menos de 5km.

$$\forall i \quad \sum_{j=1}^n \text{cadeia2}_j \cdot \text{Menos5}_{ij} \geq \text{cadeia1}_i$$

$$\forall i \sum_{j=1}^n \text{cadeia1} j . \text{Menos5} ij \geq \text{cadeia2} i$$

4 Exercício 4

4.1 Variáveis

Recebemos na entrada as variáveis de entrada n (número de compartimentos do caminhão tanque), m (número de tipos de combustíveis), $\text{capacidade}(j)$ e $\text{demanda}(i)$ em litros.

Criamos duas variáveis binárias: $x(j,i)$ igual a 1 se tanque i carrega o combustível do tipo j ou zero caso contrário e $w(i)$ igual a 1 se o tanque i for utilizado ou zero caso contrário.

Além disso, criamos uma variável inteira $y(i,j)$ que indicará quanto do combustível j irá ser carregado no tanque i ; e uma outra variável do tipo `linctr` $\text{enviado}(j)$ usada somente para indicar a quantidade que foi transportada de cada combustível do tipo j .

4.2 Restrições

A primeira restrição, apesar de não fazer parte totalmente necessária da resolução do problema facilita na apresentação da resposta, indica somente que a quantidade enviada de cada combustível j é a somatória de todos os tanques que possuem o combustível desse tipo:

$$\forall j \quad \text{enviado}(j) = \sum_{i=1}^n y(i, j)$$

A segunda restrição foi imposta para relacionar as duas variáveis binárias porque um tanque i só pode estar carregando o combustível j se o tanque estiver sendo usado:

$$\forall i, j \quad x(i, j) \leq w(i)$$

A terceira restrição impede que os combustíveis seja misturados, pois um compartimento só pode levar um tipo de combustível:

$$\forall i \quad \sum_{j=1}^m x(i, j) \leq 1$$

A quarta restrição impede que a quantidade enviada de cada combustível seja maior que sua demanda:

$$\forall j \quad \text{enviado}(j) \leq \text{demanda}(j)$$

A quinta restrição impede que cada tanque exceda seu limite de capacidade:

$$\forall i, j \quad y(i, j) \leq x(i, j) . \text{capacidade}(j)$$

A sexta restrição garante que a quantidade enviada do combustível j seja menor que a soma de cada tanque que transporta esse combustível:

$$\forall j \quad enviado(j) \leq \sum_{i=1}^n x(i, j).capacidade(i)$$

4.3 Objetivo

A função Objetivo dá-se pela soma da diferença da demanda pelo envio de cada tipo de combustível:

$$\text{Objetivo} = \sum_{j=1}^m (demanda(j) - enviado(j))$$