

MC 548 - Análise de Algoritmos II

Trabalho Final

PLI

Alexandre C. Monteiro RA007946
Carlos R. Camolesi RA 008283

1 Problema do Equicorte Mínimo

1.1 Descrição das variáveis do modelo

Utilizamos uma variável U^* que mapeia os vértices que pertencem a U^* . Sendo 1 para os vértices incluídos e 0 para os excluídos.

E utilizamos uma variável δU que representa as arestas (i,j) que pertencem a $\delta(U^*)$. Seguindo a mesma idéia da primeira variável, 1 incluído 0 excluído.

1.2 Descrição das restrições

O módulo de $U^* = n/2$, assim temos que $\sum_{i=0}^n U^* = n/2$.

As restrições que foram aplicadas em $\delta(U^*)$, U^*_i e U^*_j estão na tabela:

U^*_i	U^*_j	δU^*	Restrição
0	0	0	$\delta U^* \leq U_i + U_j$
0	1	1	$\delta U^* \geq U_j + U_i$
1	0	1	$\delta U^* \geq U_i - U_j$
1	1	0	$\delta U^* \leq 2 - U_i - U_j$

1.3 Descrição da função objetivo

É a soma das arestas que pertencem a $\delta(U^*)$.

$$\sum_{i=0}^n \sum_{j=0}^n \delta U(i,j)$$

2 Problema da Coleta de Lixo

2.1 Descrição das variáveis do modelo

As variáveis utilizadas no modelo, além das de entrada, são uma string $alocou(n \times m)$ e uma lista $alocado(m)$. A primeira retorna 1 se o setor i alocou o depósito j , enquanto a segunda retorna 1 se o depósito j foi alocado.

2.2 Descrição das restrições

A primeira restrição é que, como cada setor deve estar alocado a um único depósito, a soma das linhas da matriz $Alocou(n,m)$ deve ser 1.

$$\text{forall } (i \text{ in } 1..n) \\ \text{sum}(j \text{ in } 1..m) Alocou(i,j) = 1$$

A segunda restrição diz que a capacidade de armazenamento de um depósito não pode ser menor que a capacidade de produção dos setores alocados ao mesmo.

$$\text{forall } (j \text{ in } 1..m) \\ \text{capacidade}(j) \geq \text{sum}(i \text{ in } 1..n) Alocou(i,j) * req(i)$$

A terceira restrição diz que um depósito j só atende algum setor se ele estiver instalado em j .

$$\text{forall}(i \text{ in } 1..n, j \text{ in } 1..m) \\ Alocou(i,j) \leq Alocado(j)$$

2.3 Descrição da função objetivo

O objetivo do nosso programa é minimizar os custos, logo o custo final será o custo fixo de instalação dos depósitos mais o custo de transporte. Devemos perceber que o custo por km está em metros cúbicos, enquanto o $req(i)$ retorna milhares de metros cúbicos, logo, devemos multiplicar por 1000.

Podemos separar o custo em 2 variáveis:

```
custo_1:=sum(j in 1..m) Alocado(j)* custo_fixo(j)
custo_2:= sum(i in 1..n,j in 1..m) Alocou(i,j)* req(i) * distancia(i,j) *
custo_km * 1000
```

O objetivo do nosso modelo é minimizar $custo_1 + custo_2$.

3 Problema da Cadeia de Restaurantes

3.1 Descrição das variáveis do modelo

A variável utilizada no modelo é a string Instalado ($n \times 2$), cujo elemento será 1 se no local i houver sido instalado um restaurante da cadeia j .

3.2 Descrição das restrições

A primeira restrição diz que em um mesmo local deve haver no máximo um restaurante de cada cadeia

```
forall (i in 1..n)
    sum(j in 1..2) Instalado(i,j) <= 1
```

A segunda restrição diz que não se deve instalar restaurantes da mesma cadeia em locais distantes menos de 5 km (restaurantes de cadeias diferentes são permitidos)

```
forall (i in 1..n, j in 1..n) do
    Instalado(j,1) <= 1 - (Instalado(i,1)*Menos5(i,j))
    Instalado(j,2) <= 1 - (Instalado(i,2)*Menos5(i,j))
end-do
```

A terceira restrição só é usada no exercício 3-b. Ela diz que se instalarmos um restaurante de uma cadeia num local então obrigatoriamente irá instalar um restaurante da outra cadeia a uma distância não superior a 5 Km.

```
forall(i in 1..n) do
    Instalado(i,2) <= sum(j in 1..n ) Instalado(j,1) * Menos5(i,j)
    Instalado(i,1) <= sum(j in 1..n ) Instalado(j,2) * Menos5(i,j)
end-do
```

3.3 Descrição da função objetivo

Agora comece nossa função objetivo. O Objetivo do nosso programa é maximizar os lucros do restaurante, com essas restrições dadas acima.

$$\text{Max}(\text{sum}(i \text{ in } 1..n, j \text{ in } 1..2) \text{ Instalado}(i,j) * \text{lucro}(i,j))$$

4 Problema do tanque de combustíveis

4.1 Descrição das variáveis do modelo

As variáveis utilizadas no nosso sistema, além das variáveis de entrada, são uma string Usado (nxm) e uma string Quant(nxm). Enquanto um elemento da primeira string é 1 é quando o tanque n está sendo utilizado com o combustível m, um elemento da string 2 representa qual a quantidade de combustível m contida no tanque n.

4.2 Descrição das restrições

A primeira restrição diz que em um compartimento do tanque só pode haver um tipo de combustível:

```
forall (i in 1..n)
    sum(j in 1..m) Usado(i,j) = 1
```

A segunda restrição diz que a quantidade de combustível em todos os compartimentos que contém um tipo de combustível m deve ser menor que a demanda por esse mesmo combustível;

$$\text{forall}(j \text{ in } 1..m) \\ \text{sum}(i \text{ in } 1..n) \text{Quant}(i,j) \leq \text{demanda}(j)$$

A terceira restrição diz que quantidade de combustível levada em cada compartimento do tanque não pode ser maior que a capacidade do mesmo compartimento:

$$\text{forall } (i \text{ in } 1..n, j \text{ in } 1..m) \\ \text{capacidade}(i) * \text{Usado}(i,j) \geq \text{Quant}(i,j)$$

4.3 Descrição da função objetivo

O Objetivo do nosso programa é minimizar a quantidade de combustível total que deixa de ser transportado.

$$\text{Min}(\text{sum } (j \text{ in } 1..m) \text{demanda}(j) - \text{sum}(i \text{ in } 1..n, j \text{ in } 1..m) \text{Quant}(i,j))$$