Disciplina MC658
Projeto e Análise de Algoritmos III
2.o Semestre de 2015 - Prof.: Flávio Keidi Miyazawa
Instituto de Computação - UNICAMP

Marcelo A. G. dos Santos RA:106140

---

**Laboratório 3 - Laboratório 4: Transmissão multipla de dados**

## Introdução

A empresa Skype precisa conectar seus clientes pela internet. Cada conexão tem um custo por bytes transmitidos, uma latência de transmissão e uma capacidade. Assim, a empresa deseja conectar todos os pares de clientes, sem haver quebra de dados por diversos caminhos, com custo minimo, respeitando o limite máximo de latência para cada par e sem ultrapassar a capacidade de cada conexão

## Atividades

Para avaliarmos a solução proposta, geramos diversas entradas com 100 vertices e variando o número de pares a serem conectados até que o limite de 10s de processamento fosse atingido. Para garantir que as entradas tenham solução, foi adicionada uma aresta entre os pares com capacidade 10000 e custo igual a 99999/q[i], onde q[i] é a quantidade de dados a serem transportados pelo par i, de forma que essas arestas só irão fazer parte da solução caso o arquivo de teste gerado não tenha solução.

## Solução Proposta

Seja V os vértices do grafo que não são os pares a serem conectados, E as arestas do grafo e N os vértices a serem conectados do grafo. Sejam $c_j, l_j, k_j, q_i, T_j$ o custo da aresta j, a latência da aresta j, a capacidade da aresa j, a quantidade de dados a serem transportados do par i e o tempo máximo para transporta esses dados entre esses pares. Para solucionar o problema, foram criadas duas matrizes M1 ( NxV ) e M2 ( NxE ). Assim, uma posição M1[i][j] representa se o vértice j está presente na conexão do par i, assim como M2[i][j] representa se a aresta j está na conexão do par i. Assim, formulação de programação linear inteira fica, com xij representando as arestas e zij representando os vertices das soluções:

$$minimizar \sum_{i=0}^{|N|} \sum_{j=0}^{|E|} q_j c_j x_{ij}$$

1) $\sum_{i=0}^{|N|} \sum_{j=0}^{|E|} l_j x_{ij} \leq T_i$

2) $\sum_{j=0}^{|E|} \sum_{i=0}^{|N|} q_i x_{ij} \leq k_j$

3) $\sum_{i=0}^{|N|} \sum_{j \in E \in N} x_{ij} = 1$

4) $\sum_{i=0}^{|N|} \sum_{j \in E \in N} x_{ij} = 2 * z_{ij}$

Explicando melhor cada restrição temos que:

1) A somatoria das arestas utilizadas para conectar cada par i vezes sua latência deve ser menor que o tempo máximo estipulado para cada par
2) A somatoria dos dados que passam por uma aresta j não deve ultrapassar a capacidade kj.
3) A somatoria das arestas que pertencem aos vértices que devem ser conectados deve ser igual a um para toda linha da matriz M2
4) A somatoria das arestas dos outros vértices deve ser igual a zero ou igual a dois, garantindo que o fluxo dos dados não irá parar em um vértice intermediario.

## Experimentos e Discussões

Na tabela abaixo encontra-e o resultados dos testes obtidos para 100 vértices e incrementando os pares a serem conectados:

| **Número de Arestas** | **Número de Pares** | **Tempo de Solução (segundos)** | **Custo Total** |
|---|---|---|---|
| 288 | 3 | 0,05 | 299.997 |
| 290 | 5 | 0,14 | 300.671 |
| 292 | 7 | 0,22 | 301.140 |
| 294 | 9 | 0,71 | 501.138 |
| 296 | 11 | 0,85 | 601.377 |
| 298 | 13 | 1,81 | 701.639 |
| 300 | 15 | 2.49 | 702.017 |
| 302 | 17 | 6.92 | 802.205 |
| 304 | 19 | 9.32 | 902.405 |

## Conclusão

Podemos perceber que conforme aumentavamos o números de pares a serem conectados o tempo de solução aumentava signficativamente, uma vez que a matriz de variaveis aumentava o seu número de linhas. Podemos perceber que sem a adição das arestas entre os pares todos os problemas não teriam solução, uma vez que todas as entradas geradas utilizaram pelo menos uma das arestas que foram criadas.