

## Laboratório 3 - Programação Linear

Erik Perillo, RA135582

28 de maio de 2017

### 1 Formulação do Problema

O problema consiste em encontrar a melhor alocação possível de recursos entre roteadores e servidores.

Para isso, considere um problema com um conjunto de terminais  $T$  e tamanho  $|T| = n_1$  com requerimentos de banda  $r_t$  para todo  $t \in T$ , um conjunto de roteadores  $R$  e tamanho  $|R| = n_2$  com limites de banda  $l_r$  para todo  $r \in R$  e um conjunto de possíveis conexões  $X$  com tamanho  $|X| = m$  com custo por unidade de banda  $c_x$  para todo  $x \in X$ . Pode-se imaginar a instância como um grafo não direcionado  $G = (V, E)$ , com  $V = T \cup R$  e  $E = X$ .

Queremos encontrar os valores  $x \in X$  de modo a minimizar o custo total das ligações:

$$\min z = \sum_{x \in X} x c_x$$

Sujeito a:

- Cada terminal deve receber pelo menos uma certa quantidade de banda:

$$\sum_{x \in Adj(t)} x \geq r_t, \forall t \in T$$

- Cada roteador pode fornecer no máximo uma certa quantidade de banda:

$$\sum_{x \in Adj(r)} x \leq l_r, \forall r \in R$$

- Os valores são não-negativos:

$$x \geq 0, \forall x \in X$$

### 2 Implementação

A implementação foi feita com o uso do gurobi (versão 7.0.2) e lemon.

### 3 Avaliação do modelo proposto

O modelo implementado foi testado primeiramente em arquivos de entrada fornecidos. Os nomes do arquivo estão no formato `<num_terminais>_<num roteadores>.in`. Nota-se que, para um limite de 15 segundos, o programa chegou à solução ótima para todas as entradas com folga de tempo, como mostra a tabela a seguir.

Tabela 1: Desempenho para entradas dadas.

arquivo	tMax(s)	custo sol	tempo(s)	timeout
1000_900.in	15	93648	7.93	0
100_59.in	15	62516	0.02	0
300_250.in	15	42117	0.48	0
3_2.in	15	391	0.00	0

Foram geradas outras entradas com tamanhos entre 1000 e 2000, os resultados encontram-se na tabela a seguir.

Tabela 2: Desempenho para entradas geradas.

arquivo	tMax(s)	custo sol	tempo(s)	timeout
659_692.in	15	14016	1.05	0
785_662.in	15	30290	0.73	0
828_887.in	15	18472	1.02	0
892_772.in	15	17972	2.03	0
927_856.in	15	20081	1.05	0
942_845.in	15	19472	1.98	0
761_752.in	15	16008	1.14	0
1123_1061.in	15	31899	1.44	0
1412_1126.in	15	29103	3.34	0
1613_1551.in	15	32894	4.67	0
1742_1508.in	15	35983	6.35	0
1893_1391.in	15	42583	4.01	0
1935_1408.in	15	39937	9.81	0
2001_1940.in	15	41242	9.04	0