

CI1238/CI7056

Otimização/Tópicos em Algoritmos

Primeiro Trabalho

13 de outubro de 2021

(Atualizado em 14 de outubro de 2021)

1 Introdução

O trabalho consiste em modelar e implementar, por programação linear, uma solução para o problema de escolha de capacidades de transmissão.

A resolução do problema, ou seja, a descrição do problema, da modelagem e da implementação, deve estar em um texto claro em formato de um artigo e em pdf. Deve conter o nome do autor (aluno), uma introdução com o problema, a modelagem e sua explicação (de por que essa modelagem resolve o problema). Todas as referências que forem usadas devem estar citadas corretamente no texto.

Não espero a implementação do método do simplex. Você deve gerar uma saída para ser usada pelo resolvidor `lp_solve`. Seu programa pode ser feito em qualquer linguagem, mas deve compilar e executar nas servidoras do DINF. A implementação deve estar descrita em um texto com exemplos de uso (pode ser o mesmo texto da resolução).

O trabalho deve ser entregue com um `makefile` de forma que ao digitar o comando `make` o executável `escolha` seja construído no diretório corrente.

Você deve entregar um arquivo compactado (no formato `tar.gz`) com seu nome (ou login) com os seguintes arquivos no diretório corrente:

- texto (em pdf);
- os fontes (podem estar em subdiretórios);
- `makefile`;
- exemplos usados no texto (podem estar em subdiretórios).

A entrega deve ser feita por e-mail para `andre@inf.ufpr.br`, em um arquivo compactado com todos os arquivos do trabalho, com assunto “Otimização-trabalho 1” (exatamente).

O trabalho pode ser feito em dupla para os alunos da graduação. Os alunos da pós-graduação devem fazer individualmente.

2 O problema

Escolha de Capacidades de Transmissão

Uma empresa de tecnologia com várias sedes em uma mesma cidade precisa reestruturar sua rede de comunicação. Esta rede é formada de conexões ponto-a-ponto entre as sedes. Existe uma demanda de transmissão entre um certo par de sedes e também um custo de instalação das conexões. A empresa precisa definir qual a capacidade de transmissão de cada conexão para minimizar os custos e atender a demanda. Estes custos são proporcionais à capacidade de transmissão de cada conexão instalada.

2.1 Formato de entrada e saída

Os formatos de entrada e saída, são descritos a seguir e devem ser usados a entrada e a saída padrões (STDIN e STDOUT).

A entrada é formada de um conjunto de números inteiros. Os números podem estar separados por 1 ou mais espaços, tabs ou fim de linha.

Entrada: Inicia com dois números, n e c representando, respectivamente, o número de sedes e o número de conexões que podem ser instaladas. Assuma que as sedes são numeradas de 1 a n . Em seguida temos uma linha com três números representando a sede de origem (o), a sede de destino (d), e a demanda de transmissão de o para d . As próximas c linhas, representando os custos, são formadas por três números representando as duas sedes e o custo de instalação desta conexões por unidade de capacidade de transmissão (nas duas direções).

Saída: um arquivo no formato de entrada do `lp_solve` com a descrição do programa linear que resolve o problema para a instância dada. O formato de entrada do `lp_solve` está descrito na URL abaixo:

<http://lpsolve.sourceforge.net/5.5/lp-format.htm>

2.2 Exemplo de entrada

Se a empresa tem 3 sedes, com demanda de 10 (não importa a unidade) da sede 1 para a sede 3, e com custos de 2 entre 1 e 2, 3 entre 2 e 3, e 5 entre 1 e 3 (veja a Figura 1).

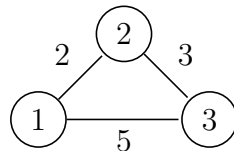


Figura 1: Desenho do exemplo. Em cada conexão temos os custos nos rótulos. A demanda é 10 da sede 1 para a sede 3.

O arquivo de entrada seria como abaixo.

```
3 3
1 3 10
1 2 2
2 3 3
1 3 5
```

Para este exemplo o melhor é instalar as conexões entre 1 e 2, e entre 2 e 3, ambas com capacidade 10, e não instalar a conexão direta entre 1 e 3. Neste caso o custo total seria 50.

2.3 Exemplo de entrada do lp_solve

Um exemplo, tirado de outro problema, pode ser visto abaixo.

```
min : 100x31 + 100x32;
```

```
x11 + x21 + x31 = 10;
```

```
x12 + x22 + x32 = 20;
```

```
x11 + x12 <= 5;
```

```
x21 + x22 <= 10;
```

```
x31 + x32 <= 50;
```

```
x21 = 0;
```