

# CI1238/CI7056

## Otimização/Tópicos em Algoritmos

Trabalho Prático

18 de fevereiro de 2022

### 1 Introdução

O trabalho consiste em modelar e implementar uma solução para o problema de envio ordenado.

O trabalho deve ser feito em duas etapas:

1. Algoritmo aproximado via programação linear relaxada, e
2. Resolução por Branch & Bound.

Na primeira etapa, não espero a implementação do método do simplex. Você pode usar um resolvidor em forma de biblioteca (ou executar um comando externamente). Uma sugestão é usar o resolvidor `lp_solve`.

Na segunda etapa é esperado a implementação da estrutura do algoritmo de Branch & Bound, entretanto, para estruturas auxiliares, como listas, filas etc podem ser usadas bibliotecas externas.

Nas duas etapas é esperado um relatório com a resolução do problema, ou seja, a descrição do problema, da modelagem e da implementação, devem estar em um texto claro em formato de um artigo e em pdf. Este texto 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.

Seus programas podem ser feitos em qualquer linguagem, mas devem compilar e executar nas servidoras do DINF. As implementações devem estar descritas nos relatórios de cada etapa, com exemplos de uso.

Em cada etapa, o trabalho deve ser entregue com um **makefile** de forma que ao digitar o comando **make** os executáveis sejam construídos 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).

As entregas devem ser feitas por e-mail para **andre@inf.ufpr.br**, em arquivos compactados com todos os arquivos de cada etapa, com assunto “Otimização-trabalho - etapa 1” ou “Otimização-trabalho - etapa 2”.

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

## 2 O problema

### Envio ordenado

Um empresa de transporte recebeu uma carga para transportar. Esta carga tem  $n$  itens,  $I_1, I_2, \dots, I_n$ , cada um com seu peso,  $w_i$ , para  $1 \leq i \leq n$ , em kg. A empresa tem somente um caminhão com capacidade de carga de  $C$ kg. A empresa (possivelmente) terá que fazer mais de uma viagem. Nossa tarefa é decidir que itens vão em que viagem, de forma a minimizar o número de viagens.

Considere que as viagens recebem números de 1 a  $k$ , na ordem em que ocorrem, onde  $k$  é o número de viagens feitas. Assim, podemos assumir que existe uma função  $v$  tal que  $v(i)$  representa o índice da viagem em que o item  $i$  foi transportado.

O cliente que contratou a empresa tem algumas restrições adicionais. O cliente entregou a empresa um conjunto de pares ordenados de itens indicando que o primeiro item do par deve chegar ao destino antes do segundo item do par. Ou seja, se o cliente incluiu o par  $(a, b)$ , isso significa que  $v(a) < v(b)$ .

Você tem que encontrar uma atribuição de itens em viagens tal que cada viagem tenha a soma de pesos dentro do limite de carga ( $C$ ), que cada par da restrição de ordem seja respeitado, e que minimize o número de viagens ( $k$ ).

### 3 Etapa 1 - Algoritmo aproximado via programação linear relaxada

Nesta etapa, temos que definir o conceito de solução parcial. Como uma solução pode ser descrita com uma função  $v$ , que é uma atribuição de itens às viagens, uma solução parcial é uma função  $v$  parcialmente definida. Ou seja, para um conjunto  $P$  de itens, já sabemos o valor de  $v$ . Para os demais, ainda não.

Assim, o objetivo desta etapa é construir uma função (de nome `parcial`) que recebe uma solução parcial e devolve um valor de  $k$  menor ou igual que o número mínimo de viagens necessários assumindo a solução parcial. Esta função deve ser construída usando programação linear relaxada.

Você deve fazer um programa que recebe os dados do problema (número de itens, capacidade do caminhão, pesos dos itens e restrições de ordem) e o conjunto  $P$  com os valores de  $v$ , chama a função descrita acima e escreve os valores de  $v$  para todos os itens e o valor de  $k$  (ver Seção 5 para os formatos de entrada e saída).

O programa construído deve ter o nome `parcial-relaxada`.

### 4 Etapa 2 - Resolução por Branch & Bound

Nesta etapa, um programa (de nome `envio`) deve ser construído. Tal programa deve receber os dados do problema (número de itens, capacidade do caminhão, pesos dos itens e restrições de ordem) e devolver (escrever) os valores de  $v$  para todos os itens e o valor de  $k$  da solução ótima (inteira) (ver Seção 5 para os formatos de entrada e saída).

### 5 Formatos 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 três números,  $n$ ,  $m$  e  $C$  representando, respectivamente, o número de itens, o número de pares ordenados (restrições de ordem) e a capacidade do caminhão (em kg). Assuma que os itens são numerados de 1 a  $n$ . Em seguida temos uma linha com  $n$  números representando os pesos dos itens. As próximas  $m$  linhas são formadas

por dois números representando os dois itens de cada par ordenado das restrições de ordem.

Para a primeira etapa, a entrada tem ainda a solução parcial. A solução parcial é dada por um número ( $\ell$ ) representando o tamanho do conjunto  $P$ , seguido de  $\ell$  linhas com dois números representando respectivamente o índice  $i$  de cada item de  $P$  e o valor de  $v(i)$ .

**Saída:** É formada por diversas linhas com um número por linha. Na primeira etapa a saída é composta apenas do número de viagens casas decimais. Na segunda a saída é formada de números inteiros da seguinte forma: na primeira linha está o valor de  $k$  e nas  $n$  linhas seguintes (um valor por linha) estão os valores da função  $v$  para cada item.

## 5.1 Exemplo de entrada

Se a carga tem 5 itens com pesos 5, 6, 4, 8, 5, a capacidade do caminhão é 10, temos 2 pares ordenados, (2, 3) e (5, 1). O arquivo de entrada seria como abaixo.

```
5 2 10
5 6 4 8 5
2 3
5 1
```

Para este exemplo precisamos de 4 viagens, item 2 na primeira, itens 3 e 5 na segunda, depois uma viagem para o item 1 e uma para o item 4. Ou seja,  $v(1) = 3$ ,  $v(2) = 1$ ,  $v(3) = 2$ ,  $v(4) = 4$  e  $v(5) = 2$ .

Em uma solução parcial onde  $v(3) = 2$  e  $v(5) = 3$ , usando programação linear relaxada, temos um total de 2.8 viagens, onde a viagem 1 tem o item 1 e 0,8333 do item 2 (5 kg + 4,99 kg); a viagem 2 tem 0,167 do item 2, o item 3 e 0,625 do item 4 (1 kg + 4 kg + 5 kg); a viagem 3 tem 0,375 do item 4 e o item 5 (3 kg + 5 kg).

## 5.2 Exemplo de saída

Uma saída possível para a primeira etapa para o exemplo de entrada acima com a solução parcial dada seria como abaixo.

2.8

A saída da segunda etapa para o exemplo de entrada acima seria como abaixo.

4  
3  
1  
2  
4  
2