

CI1238 - Otimização

Segundo Trabalho

2 de fevereiro de 2023

1 Introdução

O trabalho consiste em modelar e implementar, por *Branch & Bound*, o problema *Envio perigoso*, descrito na Seção 2.

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 em pdf. Além disso, deve ser feita uma análise com duas funções limitantes (*bounds*) diferentes, sendo uma dada pelo professor (Seção 3) e outra escolhida pelos alunos (que deve ser melhor que a dada). A função limitante dos alunos deve ser a default na implementação e a dada pelos professores é escolhida usando uma opção da linha de comando (-a).

O texto deve conter o nome dos autores (alunos), uma introdução com o problema, a modelagem e sua explicação (de por que essa modelagem resolve o problema), detalhes da implementação (com exemplos de uso), e uma análise do uso das funções limitantes. Nesta análise devem ser feitas e comparadas contagens de número de nós da árvore e tempo de execução (com relatório gerado pelo programa). Outras métricas também podem ser usadas.

Todas as referências que forem usadas devem estar citadas corretamente no texto.

Para facilitar a análise, o seu programa deve ter a opção de fazer ou não os cortes por viabilidade e otimalidade. Ou seja, com a opção de linha do comando -f os cortes por viabilidade não devem estar ativos; e com a opção de linha do comando -o os cortes por otimalidade são desligados.

Você pode usar bibliotecas para estruturas de dados (como listas, conjuntos etc), mas não para o algoritmo de resolução principal do problema. O seu programa deve compilar e executar nas servidoras do DINF.

O trabalho deve ser entregue com um **makefile** de forma que ao digitar o comando **make** o executável **envio** seja construído.

Resumindo, o texto deve ter:

- identificação;
- explicação do problema;
- modelagem;
- análise das funções limitantes;
- detalhes da implementação.

A implementação:

- deve ter executável de nome **envio**;
- na opção default todos os cortes (viabilidade e otimalidade) estão ativos e a função limitante é a criada pelos autores;
- com a opção **-f** na linha de comando deve desligar os cortes de viabilidade;
- com a opção **-o** na linha de comando deve desligar os cortes de otimalidade;
- com a opção **-a** na linha de comando deve usar a função limitante dada pelos professores;
- deve gerar relatório (na saída de erro padrão, **stderr**) com número de nós da árvore e tempo gasto (sem contar o tempo de entrada e saída).

Você deve entregar um arquivo compactado (**tar.gz**) com os seguintes arquivos no diretório corrente:

- texto (em pdf);
- os fontes (podem estar em subdiretórios);
- makefile;
- exemplos usados na análise (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 2” (exatamente).

2 O problema

Envio perigoso

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 é uma fábrica de produtos químicos e informou que certos pares de produtos não podem viajar juntos. O cliente entregou a empresa um conjunto de pares de itens indicando que estes itens não podem estar juntos no caminhão. Ou seja, se o cliente incluiu o par (a, b) , isso significa que $v(a) \neq v(b)$.

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 seja respeitado, e que minimize o número de viagens (k).

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 padrão (`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 os valores de n (número de itens), p (número de restrições) e C (capacidade do caminhão) na primeira linha (separados por espaço). Em seguida temos uma linha com os n valores dos pesos w_i dos itens, para $1 \leq i \leq n$. Depois temos p linhas, cada uma com dois números, a_j e b_j , com $1 \leq j \leq p$, sendo que $1 \leq a_j, b_j \leq n$.

Saída: Os números das viagens de cada item. Ou seja, os valores de $v(1)$ até $v(n)$, em uma mesma linha, separados por espaço (simples) e sem espaço no começo nem no fim da linha. Na linha seguinte deve estar o número total de viagens.

2.2 Exemplos

Exemplos de entrada e saída.

2.2.1 Exemplo simples com $n = 3$, $p = 1$ e $C = 10$

Entrada:

3 1 10
3 4 2
1 3

Saída:

1 1 2
2

2.2.2 Exemplo simples com $n = 5$, $p = 2$ e $C = 100$

Entrada:

5 2 100
10 20 30 40 50
1 2
2 5

Saída:

1 2 2 1 1

3 Função limitante dada

Dados o conjunto de itens com viagens já escolhidas (E), sabendo que $v(i)$ é a viagem do item i (já escolhida), considerando que $k = \max\{v(i) \mid i \in E\}$, podemos definir a função $B_{dada}(E)$ por:

$$B_{dada}(E) = \max\{k, \sum_{i=1}^n w_i/C\}.$$

Ou seja, $B_{dada}(E)$ é o máximo entre as viagens já usadas e a soma de todos os pesos dividida pela capacidade do caminhão.