

# CI1238 - Otimização

## Segundo Trabalho

15 de agosto de 2022

(Atualizado em 4 de setembro de 2022)

## 1 Introdução

O trabalho consiste em modelar e implementar, por *Branch & Bound*, o problema *Elenco Representativo*, 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 pelos professores (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 `elenco` 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 `elenco`;
- 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` (turma BCC1) ou `guilherme@inf.ufpr.br` (turma BCC2), em um arquivo compactado com todos os arquivos do trabalho, com assunto “Otimização-trabalho 2” (exatamente).

## 2 O problema

### Elenco Representativo

Uma produtora de filmes quer fazer um filme e está a procura do elenco ideal. Tentando evitar polêmicas de representatividade do elenco em relação a grupos da sociedade, a produtora quer que todos os grupos (previamente elencados) da sociedade sejam representados. Um ator pode fazer parte de mais de um destes grupos, portanto podemos ter menos atores que grupos. Obviamente, é preciso que cada personagem do filme tenha um ator associado e cada ator cobra um valor para fazer o filme

Dados um conjunto  $S$  de grupos, um conjunto  $A$  de atores, um conjunto  $P$  de personagens, e, para cada ator  $a \in A$ , um conjunto,  $S_a \subseteq S$  indicando os grupos dos quais  $a$  faz parte, devemos encontrar um elenco que tenha um ator para cada personagem (todos os atores podem fazer todas as personagens) e todos os grupos tenham um representante. Além disso, também temos um valor,  $v_a$ , associado com cada ator  $a \in A$ , e queremos que o custo do elenco seja mínimo.

Ou seja, devemos encontrar um subconjunto  $X \subseteq A$  tal que:

- $|X| = |P|$ ;
- $\bigcup_{a \in X} S_a = S$ ; e
- $\sum_{a \in X} v_a$  seja mínimo.

### 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  $\ell = |S|$ ,  $m = |A|$  e  $n = |P|$  na primeira linha (separados por espaço). Considere que  $S = \{1, \dots, \ell\}$  e  $A = \{1, \dots, m\}$ . Em seguida temos  $m$  blocos (um para cada ator). Um bloco de ator começa com dois valores,  $v$  e  $s$ , indicando o valor cobrado pelo ator e o número de grupos da sociedade que ele faz parte. Em seguida temos  $s$  números, que são os índices dos grupos.

**Saída:** Os números do conjunto  $X$ , em uma mesma linha, separados por espaço (simples) e sem espaço no começo nem no fim da linha, ordenados em ordem crescente. Na linha seguinte deve estar o valor total de contratação do elenco.

Caso a instância não tenha solução viável, informe isso escrevendo “Inviável” na saída.

## 2.2 Exemplos

Exemplos de entrada e saída.

Note que o número de nós e o tempo apresentados no relatório de cada exemplo são apenas exemplos, já que podem variar com a modelagem e a função limitante escolhida.

### 2.2.1 Exemplo simples com $\ell = 2$ , $m = 3$ e $n = 2$

**Entrada:**

```
2 3 2
10 2
1
2
20 1
2
5 2
1
2
```

**Saída:**

```
1 3
15
```

### 2.2.2 Exemplo inviável

**Entrada:**

```
2 3 2
10 1
1
20 1
1
5 1
1
```

Saída:

Inviável

### 2.2.3 Exemplo onde um ator está em mais de um grupo na solução

Entrada:

3 3 2  
10 2  
1  
2  
20 2  
2  
3  
5 1  
3

Saída:

1 3  
15

## 3 Função limitante dada

Dados os atores já escolhidos ( $E$ ) e o conjunto de atores ainda por escolher ( $F$ ), sabendo que o número de personagens é  $n$ , podemos definir a função  $B_{dada}(E, F)$  por:

$$B_{dada}(E, F) = \sum_{a \in E} v_a + (n - |E|) \min\{v_a \mid a \in F\}.$$

Ou seja,  $B_{dada}(E, F)$  é a soma dos valores cobrados pelos atores já escolhidos mais o valor mais baixo cobrado pelos atores que ainda faltam decidir multiplicado pelo número de personagens ainda sem ator.