

Trabalho prático 1

Implementação do método Simplex

Data de entrega: 30/09/2018

Valor: 10 pontos (+1 extra)

O objetivo deste trabalho é resolver algumas PLs, a serem fornecidas e cujo formato será incluído abaixo, transformando-as em FPI e implementando o método Simplex.

- (i) A linguagem de programação é livre, mas deve ser algo que eu consiga rodar em Linux, na linha de comando. É permitido o uso de estruturas de dados de bibliotecas de álgebra linear e tratamento de frações (por exemplo, numpy e scipy se escolher Python).
- (ii) Tome cuidado com o condicionamento da matriz. Sugiro transformar números pequenos o suficiente em 0 ou implementar frações.
- (iii) Como sugestão, implemente o pivoteamento como uma função separada que recebe como entrada uma matriz e um par de índices (linha e coluna), e devolve uma matriz com aquela entrada pivoteada.
- (iv) Seu programa deve receber dois argumentos em linha de comando, um arquivo de entrada com a PL e um arquivo de saída com a solução.
- (v) Para o formato do arquivo texto de entrada, considere o seguinte exemplo:

$$\begin{array}{ll}\max & x_1 + x_2 \\ \text{sujeito a} & x_1 - x_2 \leq 2 \\ & x_1 + x_2 \geq 1 \\ & x_1 \geq 0\end{array}$$

Ela será representada assim no arquivo de entrada:

```
2
2
1 0
1 1
1 -1 <= 2
1 1 >= 1
```

A primeira linha representa o número de variáveis e a segunda representa o número de restrições (excluindo as restrições de não negatividade). A terceira linha indica se as variáveis são não-negativas ou livres:

- 1 para variáveis não negativas,

- 0 para variáveis livres.

A quarta linha representa os coeficientes na função objetivo e as demais linhas representam as restrições, incluindo coeficientes, sinal (\leq , \geq ou $=$) e o lado direito.

(vi) O seu programa deve:

- (a) Ler a entrada
- (b) Transforma-la em FPI
- (c) Rodar a PL auxiliar para encontrar uma base e verificar se o problema é viável (este passo pode ser pulado caso você identifique uma base óbvia).
- (d) Se o problema for viável, rodar o Simplex e ou encontrar a solução ótima, ou verificar que o problema é ilimitado.
- (e) Ao final, deve escrever um arquivo de saída. Será dado 1 ponto extra para quem incluir certificados no arquivo de saída.

(vii) O arquivo de saída, caso o problema seja inviável, deve ter o seguinte formato:

```
Status: inviavel
Certificado:
a1 a2 a3 ...
```

(viii) Caso o problema seja ilimitado, deve ter o seguinte formato:

```
Status: ilimitado
Certificado:
a1 a2 a3 ...
```

(ix) Caso possua solução ótima:

```
Status: otimo
Objetivo: z
Solucao:
v1 v2 v3...
Certificado:
a1 a2 a3 ...
```

(x) Em todos os casos, deixe uma linha em branco caso opte por não produzir os certificados. Obviamente que **z** deve ser o valor da função objetivo, **v1 v2 v3 ...** os valores das variáveis e **a1 a2 a3 ...** os elementos dos vetores certificado.

(xi) O trabalho deve ser submetido via Moodle. Submeta um arquivo zip contendo apenas os arquivos de código, um makefile (caso utilize) e um arquivo texto chamado **instrucoes.txt** com instruções de como rodar seu programa em linha de comando. Por exemplo, diga se é em Python3, Python2, se for em C ou C++ dê o comando de compilação. Lembre-se que vou compilar em Linux (Ubuntu) e por isso não utilize bibliotecas que somente funcionam no Windows. **Não inclua executável ou os arquivos de teste contendo PLs.**