

## Simplex

O algoritmo simplex implementado se baseia no algoritmo descrito por Arenales et al. (2007). Foi implementado o método de duas fases. Não foi considerada a degeneração por parte de variáveis artificiais.

## Requisitos

A implementação foi feita utilizando Python3. É necessário ter instalado o pacote numpy.

## Formato da entrada

### Forma Padrão

Inicialmente o algoritmo considera o formato padrão:

$$\min f(x) = c^T x$$

s. a.:

$$Ax = b$$

$$x \geq 0$$

Um algoritmo de inserção de variáveis de folga foi criado para que restrições de desigualdade ( $\leq$  ou  $\geq$ ) fossem transformadas em restrições de igualdade. Portanto o formato aceito passou a ser:

$$\min f(x) = c^T x$$

s. a.:

$$A_1 x = b$$

$$A_2 x \leq b$$

$$A_3 x \geq b$$

$$x \geq 0$$

Sendo  $A_1$ ,  $A_2$  e  $A_3$  partições de  $A$ , tal que  $A_1 \cup A_2 \cup A_3 = A$ .

### Arquivo de Entrada

O arquivo de entrada deve conter os seguintes dados:

$T$ : se a função é de minimização ou maximização (*string*)  $n$ : número de variáveis (inteiro positivo)

$m$ : número de restrições (inteiro positivo)

$c$ : vetor de custos

$A$ : matriz  $A$  para o simplex (matrix  $m \times n$  de números reais)

$b$ : vetor de recursos (vetor com  $m$  números reais)

$O$ : vetor de operadores, tal que  $O_i$  é o operador da linha  $i$  (vetor com  $m$  *strings*)

Valores reais devem estar separados por ponto, não por vírgula. Os valores possíveis para os operadores são qualquer *string*. Se o valor for " $\leq$ " ou " $<$ ", a restrição será considerada de menor igual. Analogamente, se o valor for " $\geq$ " ou " $>$ ", a restrição será

considerada de maior igual. Qualquer outro valor será considerado como igualdade. Por fim, o valor de  $T$  deve ser igual a  $max$  se o problema for de maximização. Caso contrário será considerado de minimização. A capitalização da palavra  $max$  não é considerada, ou seja, as *strings*  $Max$  ou  $max$  também seria considerada como maximização.

As linhas do arquivo é estruturado da seguinte forma

```

1      T
2      n
3      m
4      c
5      A1
6      A2
      .
      .
      .
i+4    Ai
      .
      .
      .
m+4    Am
m+5    O m+6    b

```

Cada linha da matriz  $A$  contém  $n$  valores que devem estar separados por espaço. O mesmo é válido para os vetores  $O$  e  $b$ . É importante notar que todas as posições de todos os vetores e matrizes devem ser passados.

**Este arquivo** apresenta uma instância exemplo com a explicação de cada linha. Nele está apresentado o a seguinte instância do problema da dieta com mínimo e máximo de nutrientes:

$$\min f(x) = 0.56x_1 + 0.81x_2 + 0.46x_3$$

s. a. :

$$0.2x_1 + 0.5x_2 + 0.4x_3 \geq 0.3$$

$$0.6x_1 + 0.4x_2 + 0.4x_3 \geq 0.5$$

$$0.2x_1 + 0.5x_2 + 0.4x_3 \leq 0.8$$

$$0.6x_1 + 0.4x_2 + 0.4x_3 \leq 0.7$$

$$x_1 + x_2 + x_3 = 1$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$$

Note que os operadores " $\leq$ " foi representado tanto por " $\leq$ " quanto por " $<=$ ". O mesmo ocorre com " $\geq$ ", representador por " $\geq$ " ou " $>=$ ". No caso do operador "=", pois qualquer outra *string* diferente de " $<$ ", " $\leq$ ", " $>$ " e " $\geq$ " é considerada o operador "=".

Por fim, o domínio das variáveis não é necessário. Assume-se que toda variável deve ser maior ou igual a zero.

## Execução

Para executar a implementação do Simplex basta utilizar o seguinte comando na pasta do projeto:

```
python main.py <arquivo-de-entrada>
```

O parâmetro <arquivo-de-entrada> é o arquivo contendo o modelo no formato descrito pela Seção [Arquivo de Entrada](#). Foram criados 5 exemplos para ilustrar o funcionamento do algoritmo. Eles se encontram na [pasta exemplos](#).

Se desejar executar todos os exemplos de uma única vez, basta executar a seguinte linha de comando:

```
python run_examples.py
```

## Refêrencias

ARENALES, M.; ARMENTANO, V. A.; MORABITO, R.; YANASSE, H. H. Pesquisa operacional. Rio de Janeiro: Campus/elsevier, 2007. 523 p. ISBN 10-85-352-145-1454-2.