

# Projeto computacional: Implementação do algoritmo simplex

122830 Alcides Goldoni Junior  
148585 Guilherme de Freitas Laranja  
150946 Isabela Marton  
MS 428 - Programação Linear

26 de outubro de 2017

## 1 Introdução

O método simplex é uma ferramenta utilizada para melhorar a solução da função objetivo de um problema de otimização linear. O método tenta melhorar o valor da solução a cada passo de interação até que se tenha a solução ótima.

Esse método viabiliza a solução de muitos problemas de programação linear e é muito popular.

Ele permite que se encontre valores ideais em situações em que condições necessitam ser respeitadas.

## 2 Funcionamento

Para o funcionamento da biblioteca simplex desenvolvida é preciso ter as bibliotecas gsl (GNU Scientific Library) instalada. Essa biblioteca ajuda na resolução dos sistemas lineares presentes no algoritmo do Simplex.

Para a instalação Linux (Debian Like):

```
# apt-get install libgsl-dev liblapack-dev
```

Para a instalação MacOS:

```
# brew install gsl
```

Para compilar o programa, utilize os arquivos compile.bash ou compileLinux.bash (para MacOS e Linux, respectivamente).

Linux:

```
# ./compileLinux.bash
```

MacOS:

```
# ./compile.bash
```

Será gerado um binário de nome "simplexExec" responsável pela execução do programa.

A melhor forma de executar o programa é editar o arquivo teste.in onde cada

linha representa uma entrada:

- Linha 1: Número de restrições (linhas) e número de variáveis (colunas),
- Linha 2: Vetor de custos da função objetivo,
- Linha 3: Vetor de recursos
- As próximas linhas representam a matriz dos coeficientes de restrição.

Dessa forma, o arquivo teste.in ficará da seguinte forma:

```
2 5
1 2 3 4 5
2 2
7 5 3 1 0
6 4 2 0 1
```

Para a execução:

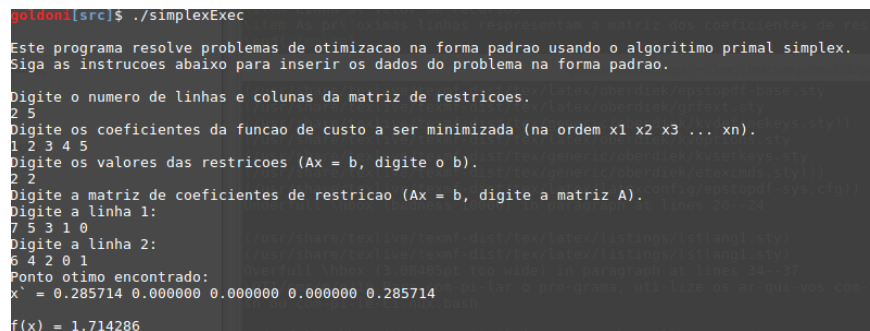
```
# ./simplexExec < teste.in
```

Caso não queira editar o arquivo, pode-se digitar as entradas baseadas na perguntas que o próprio programa pede. Neste caso, para a execução do programa fica da seguinte forma:

```
# ./simplexExec
```

A imagem a seguir, ilustra a execução e as entradas para o programa:

Figura 1: Exemplo para entradas do Simplex



```
goldoni[src]$ ./simplexExec
Este programa resolve problemas de otimizacao na forma padrao usando o algoritmo primal simplex.
Siga as instrucoes abaixo para inserir os dados do problema na forma padrao.

Digite o numero de linhas e colunas da matriz de restricoes.
2 5
Digite os coeficientes da funcao de custo a ser minimizada (na ordem x1 x2 x3 ... xn).
1 2 3 4 5
Digite os valores das restricoes (Ax = b, digite o b).
2 2
Digite a matriz de coeficientes de restricao (Ax = b, digite a matriz A).
Digite a linha 1:
7 5 3 1 0
Digite a linha 2:
6 4 2 0 1
Ponto otimo encontrado:
x' = 0.285714 0.000000 0.000000 0.000000 0.285714
f(x) = 1.714286
```

Como saída, o programa retorna:

- Caso encontre a solução, o programa retorna o ponto encontrado e o valor da função,
- Caso o problema não encontre solução factível, o programa retorna a seguinte frase: "Problema infactível!!! Ainda existem variaveis artificiais diferentes de zero na solucao encontrada com BigM"
- Caso o problema tenha infinitas soluções, o programa retorna a seguinte frase: "Problema nao tem solucao finita!!"

### 3 Testes

As possíveis soluções para o Simplex são: Factível, infactível e ilimitado. Utilizamos os seguintes testes para cada uma das soluções:

- Factível:  
O seguinte exemplo possui solução factível:  
Linhas: 2  
Colunas: 4  
Função:  $-x_1 - 3x_2 + 0x_3 + 0x_4$   
Vetor  $b = [6 \ 3]$   
Matriz de restrição:  
1 1 1 0  
0 1 0 1  
A saída do programa será a seguinte:

Figura 2: Exemplo Simplex: Problema factível

```
goldoni[src]$ ./simplexExec
Este programa resolve problemas de otimizacao na forma padrao usando o algoritmo primal simplex.
Siga as instrucoes abaixo para inserir os dados do problema na forma padrao.

Digite o numero de linhas e colunas da matriz de restricoes.
2 4
Digite os coeficientes da funcao de custo a ser minimizada (na ordem x1 x2 x3 ... xn).
-1 -3 0 0
Digite os valores das restricoes (Ax = b, digite o b).
6 3
Digite a matriz de coeficientes de restricao (Ax = b, digite a matriz A).
Digite a linha 1:
1 1 1 0
Digite a linha 2:
0 1 0 1
Ponto otimo encontrado:
x* = 3.000000 3.000000 0.000000 0.000000
f(x) = -12.000000
```

- Infactível  
O seguinte exemplo possui solução infactível:  
Linhas: 2  
Colunas: 4  
Função:  $2x_1 + 2x_2 + 0x_3 + 0x_4$   
Vetor  $b = [1 \ -1]$   
Matriz de restrição:  
-1 1 -1 0  
1 1 0 1  
A saída do programa será a seguinte:

### 4 Conclusão

Figura 3: Exemplo Simplex: Problema factível

```
goldoni[src]$ ./simplexExec
Este programa resolve problemas de otimizacao na forma padrao usando o algoritmo primal simplex.
Siga as instrucoes abaixo para inserir os dados do problema na forma padrao.

Digite o numero de linhas e colunas da matriz de restricoes.
2 4
Digite os coeficientes da funcao de custo a ser minimizada (na ordem x1 x2 x3 ... xn).
2 2 0 0
Digite os valores das restricoes (Ax = b, digite o b).
1 -1
Digite a matriz de coeficientes de restricao (Ax = b, digite a matriz A).
Digite a linha 1:
-1 1 -1 0
Digite a linha 2:
1 1 0 1

Problema infactivel!!!
Ainda existem variaveis artificiais diferentes de zero na solucao encontrada com BigM
```