Projeto computacional: Implementação do algorítmo 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 encontrar o conjunto de soluções ótimas para problemas de otimização linear.

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) eliblapack instaladas. Essas bibliotecas ajudam nas operações envolvendo vetores e matrizes, e também na resolução dos sistemas lineares presentes no algorítmo 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, utiliza-se 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 executação 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 respresentam 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:

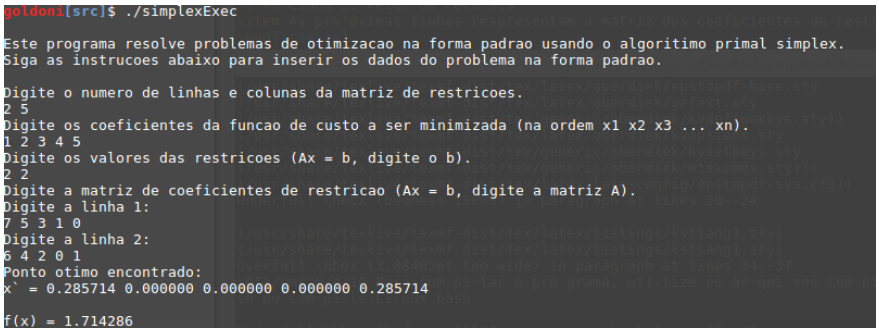
```
# ./simplexExec < teste.in
```

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



Como saída, o programa retorna:

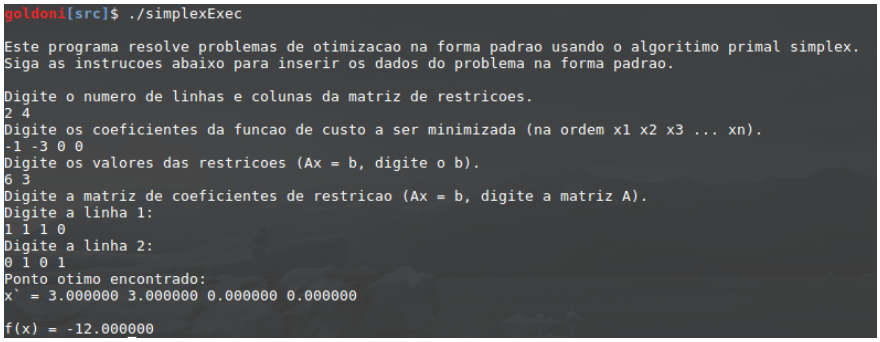
- O ponto encontrado e o valor da função, caso encontre a solução;
- A seguinte frase: "Problema infactivel!!! Ainda existem variaveis artificiais diferentes de zero na solucao encontrada com BigM", caso o problema não encontre solução factível;
- A seguinte frase: "Problema nao tem solucao finita!!", caso o problema tenha infinítas soluções.

3 Testes

Foram realizados três testes para validar o programa (factível, infactível e ilimitado), cobrindo as possíveis saídas descritas anteriormente.

- Factível:
  O seguinte exemplo (Figura 2) 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

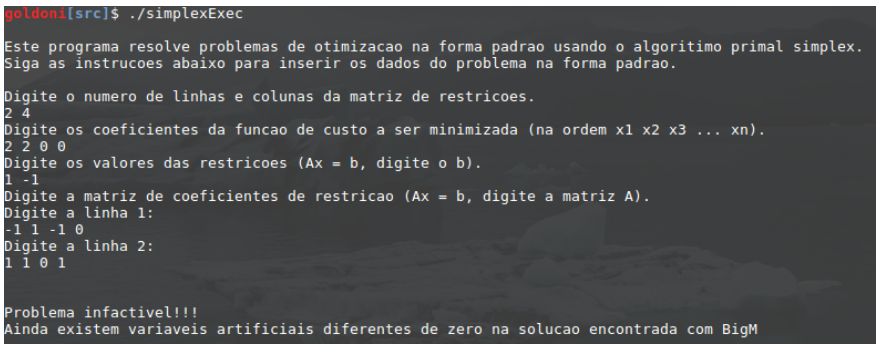
Figura 2: Exemplo Simplex: Problema factível



- Infactível
  O seguinte exemplo (Fingura 3) 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

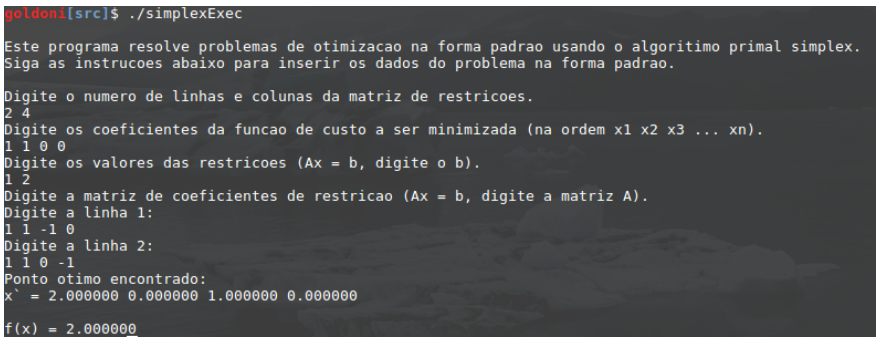
- Ilimitado
  O seguinte exemplo (Fingura 4) possui solução infactível:
  Linhas: 2
  Colunas: 4
  Função: x_1 + x_2 + 0x_3 + 0x_4
  Vetor b = [1 2]
  Matriz de restrição:
  1 1 -1 0

Figura 3: Exemplo Simplex: Problema infactível



1 1 0 -1

Figura 4: Exemplo Simplex: Problema ilimitado



4 Discusão

Podemos ver que o programa se comportou como o esperado para os casos de solução factível e infactível, porém, para os casos de solução ilimitada, ele não retorna o valor esperado. Investigamos porém não encontramos o motívo desse problema.

5 Conclusão

Neste trabalho desenvolvemos o algorítmo Simplex em linguagem C e ultilizamos algumas bibliotecas já prontas e conhecidas para facilitar sua escrita.

Foi importante para consolidar o algoritímo visto em aula.

O programa foi testados para os possíveis casos de uso porém só foi bem sucedido nas situações onde o problema era factível ou infactível.