

Trabalho de Programação Linear

Luiz Fernando Bossa

15 de julho de 2016

1 Simplex Primal

1.1 Motivação

Criei uma implementação do método simplex em Python por se tratar de código livre e que pode ser rodado em qualquer computador. Usando a biblioteca `numpy`, temos basicamente as mesmas funcionalidades que o tão difundido MATLAB®.

Além disso, usei classes para permitir uma melhor portabilidade no código, que está disponível em

github.com/LFBossa/simplex

1.2 Modo de Usar

Dado o problema de minimização na forma padrão

$$\begin{array}{ll} \min c^T x & \\ \text{s.a. } Ax = b & \text{(P)} \\ x \geq 0 & \end{array}$$

tudo o que temos que fazer é escrever a matriz A como um array bidimensional, e b e c como arrays unidimensionais. Em seguida, criamos uma instância da classe `SimplexPrimal` e usamos seu método `resolver()`, da seguinte forma.

```
>>> P = SimplexPrimal(A,b,c)
>>> P.resolver()
```

O pivotamento é mostrado na tela, e ao final, se a região viável não for vazia e o problema tiver uma solução ótima limitada, ela é exibida na tela.

1.3 Funcionamento

Quando chamamos o método `resolver()`, ele verifica se existe uma base igual a matriz identidade entre as colunas da matriz A . Se sim, ele chama o método `jatembase()`. Se não, ele chama os métodos `fase1()` e, em seguida, `fase2()`.

O método `jatembase()` cria o tableau, e chama o método `run()`.

A `fase1()` verifica quais colunas faltam para termos uma base igual a identidade, adiciona as variáveis artificiais correspondentes, monta o tableau e chama o método `run()`. Após isso, ele verifica se existe alguma variável artificial que continuou na base. Se sim e o valor dessa variável for zero, tenta removê-la (ou apaga a linha correspondente no tableau se isso não for possível). Se alguma variável artificial que está na base tiver valor diferente de zero, ele retorna o erro `SemSolucoesViaveis`.

A `fase2()` recebe o tableau da `fase1()`, elimina as colunas correspondentes as variáveis artificiais, coloca o vetor dos custos no tableau e chama o método `run()`.

O método `run()` por sua vez é a implementação do pivotamento. Ele decide quem vai entrar na base, e chama o método `quem_sai_da_base()`. Este último, por sua vez, usa a validação lexicográfica para decidir qual variável sairá da base, e se nenhuma variável puder sair da base, ele retorna o erro `ProblemaIlimitado`.