

Florence Alyssa Sakuma Shibata, Shayenne da Luz Moura

Método Simplex

Fase 2

São Paulo

2015

Sumário

	Introdução	2
1	DESCRIÇÃO DO ALGORITMO	3
2	FUNCIONAMENTO DO ALGORITMO	4
2.1	Solução ótima	4
2.2	Custo ótimo ilimitado	8
3	RESULTADOS	9
	Conclusão	10
	Referências	11

Introdução

O método simplex é baseado em encontrar uma solução viável ótima para um problema de programação linear e realiza esta busca movendo-se de uma solução viável básica para outra, percorrendo os lados do poliedro que define a região viável, sempre numa direção onde o custo se reduz. Enfim, uma solução viável básica é alcançada quando nenhuma das direções viáveis reduzem o custo; então a solução viável básica é ótima e o algoritmo termina.

1 Descrição do algoritmo

O método simplex utilizado como base para o desenvolvimento do algoritmo está descrito em [Bertsimas e Tsitsiklis \(1997, pág. 90-91\)](#).

Dadas uma matriz $A \in \mathbb{R}^{m \times n}$, uma solução viável básica $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, o vetor de custos $c \in \mathbb{R}^n$, o algoritmo realiza os seguintes passos:

- 1 Gera a matriz básica B associada a x .
- 2 Calcula o vetor de custos reduzidos para toda variável não básica. Se nenhuma componente é negativa, então a solução viável básica atual é ótima, acabou.
- 3 Caso contrário, armazena o índice da variável cujo custo reduzido foi menor. Calcula $u = B^{-1}A_j$. Se nenhum componente de u é positivo, então o custo ótimo é $-\infty$, acabou;
- 4 Caso contrário, toma $\theta^* = \min_{(i=1, \dots, m | u_i > 0)} \left\{ \frac{x_{B(i)}}{u_i} \right\}$. Seja l o índice onde o mínimo foi encontrado. Forma uma nova base substituindo $A_{B(l)}$ por A_j . Sendo y é a nova solução viável básica, os valores das novas variáveis básicas são $y_j = \theta^*$, $y_{B(i)} = \theta^* u_i$, $i \neq l$. Volta ao passo 1.

É necessário que o problema possua pelo menos uma solução viável básica e que todas as soluções viáveis básicas sejam não degeneradas.

2 Funcionamento do algoritmo

A seguir está a descrição do funcionamento do algoritmo quando o problema de programação linear dado possui pelo menos uma solução ótima ou quando o custo ótimo é ilimitado.

2.1 Solução ótima

Existem duas possibilidades para este caso:

- A solução ótima é dada;
- A solução ótima é encontrada ao percorrer as direções viáveis que reduzem o custo.

Ao aplicar o algoritmo com os dados descritos a seguir, pode-se verificar seu funcionamento.

```
> A = [1 0 1 0 0; 0 1 0 1 0; 3 2 0 0 1];  
> x = [0;0;4;6;18];  
> b = [4;16;18];  
> m = 3;  
> n = 5;  
> c = [3;5;0;0;0];
```

```
> [ind, v] = simplex(A, b, c, m, n, x);
```

```
#=====
```

```
Iteracao: 0
```

```
#=====
```

Variaveis basicas:

```
3: 4.000000  
4: 6.000000  
5: 18.000000
```

Valor funcao objetivo: 0.000000

O algoritmo calcula quais são as variáveis básicas a partir da solução x dada e o custo associado a essa solução $c^T x$.

Custos reduzidos:

1: 3.000000

2: 5.000000

Como o vetor de custos reduzidos das variáveis não básicas é positivo não existe direção viável que reduza o custo. Logo, o custo ótimo é alcançado em x . Assim, x é solução ótima.

Solucao otima com custo 0.000000:

1 0.000000

2 0.000000

3 4.000000

4 6.000000

5 18.000000

O algoritmo então devolve o valor da solução ótima que é igual a solução viável básica x dada.

Ao realizar outro teste, mudando apenas a função de custo, obtemos a resposta a seguir.

```
> c = [-3;-5;0;0;0];
```

```
> [ind, v] = simplex(A, b, c, m, n, x);
```

```
#=====
```

```
Iteracao: 0
```

```
#=====
```

Variaveis basicas:

3: 4.000000

4: 6.000000

5: 18.000000

Valor funcao objetivo: 0.000000

Custos reduzidos:

1: -3.000000

2: -5.000000

O algoritmo calcula quais são as variáveis básicas da solução viável básica x , o custo associado a ela e os custos reduzidos das variáveis não básicas. Quando encontra

um valor negativo sabe-se que existe uma solução viável básica, diferente de x , cujo custo associado é menor.

Entra na base: 1

Direcao

3: 1.000000

4: 0.000000

5: 3.000000

Theta*: 4.000000

Sai da base: 3

É calculada a direção viável que reduz o custo, além do θ máximo que se pode andar para estar sobre uma nova solução viável básica. A primeira variável encontrada que possuir custo reduzido negativo entra na base e uma das variáveis básicas torna-se não básica. Neste caso, aquela que torna o θ máximo. Uma nova iteração se inicia.

#=====

Iteracao: 1

#=====

Variaveis basicas:

1: 4.000000

4: 6.000000

5: 6.000000

Valor funcao objetivo: -12.000000

Custos reduzidos:

3: 3.000000

2: -5.000000

Entra na base: 2

Direcao

1: 0.000000

4: 1.000000

5: 2.000000

Theta*: 3.000000

Sai da base: 5

Nesta iteração novamente se calcula as variáveis básicas, agora da solução viável básica encontrada na iteração anterior, e seu custo associado. Ao calcular o vetor de custos reduzidos encontra uma componente negativa. Outra vez, sabe-se que a solução em que está não é ótima. Calcula a direção viável que reduz o custo associado, colocando na base a variável que possui custo reduzido negativo, e saindo a que torna o θ máximo. Outra iteração se inicia.

#=====

Iteracao: 2

#=====

Variaveis basicas:

1: 4.000000

4: 3.000000

2: 3.000000

Valor funcao objetivo: -27.000000

Custos reduzidos:

3: -4.500000

5: 2.500000

Entra na base: 3

Direcao

1: 1.000000

4: 1.500000

2: -1.500000

Theta*: 2.000000

Sai da base: 4

Idêntica a iteração anterior. Novamente sabe-se que há uma solução melhor que a dada no início da iteração pois o vetor de custos reduzidos não é positivo. Segue os mesmos passos já descritos e encontra-se uma nova solução viável básica com custo associado menor. Mais uma iteração se inicia.


```
#=====
```

```
Iteracao: 3
```

```
#=====
```

```
Variaveis basicas:
```

```
1:  2.000000
```

```
3:  2.000000
```

```
2:  6.000000
```

```
Valor funcao objetivo: -36.000000
```

```
Custos reduzidos:
```

```
4:  3.000000
```

```
5:  1.000000
```

```
Solucao otima com custo -36.000000:
```

```
1  2.000000
```

```
2  6.000000
```

```
3  2.000000
```

```
4  0.000000
```

```
5  0.000000
```

Esta possui o vetor de custos reduzidos positivo. Logo não existe direção viável que reduza o custo. O algoritmo devolve a solução viável básica em que está como solução ótima, pois esta possui menor custo associado.

2.2 Custo ótimo ilimitado

Como exemplo do funcionamento do algoritmo para um problema de programação linear que contém solução ilimitada segue os dados de entrada.

```
> A = [1,-1,-1,0;-1/2, 1, 0, 1],
```

```
> b = [-1;2]
```

```
> c = [-1; -1/4;0;0]
```

```
> m = 2
```

```
> n = 4
```

```
> x = [2; 3; 0; 0]
```

A execução do algoritmo devolve as iterações a seguir.

```
> [ind, v ]= simplex(A, b, c, m, n, x);
```

```
#=====
```

Iteracao: 0

```
#=====
```

Variaveis basicas:

1: 2.000000

2: 3.000000

Valor funcao objetivo: -2.750000

Custos reduzidos:

3: -2.250000

4: 2.500000

Ao andar na direção da variável básica cujo custo reduzido é menor que zero encontra-se o vetor u com elementos não positivos, isso quer dizer que a direção encontrada leva a solução ilimitada, com custo ótimo $-\infty$.

O valor da funcao objetivo vai para $-\text{Inf}$

Entra na base: 3

Direcao

1: -2.000000

2: 0.000000

Theta*: Inf

Sai da base: 0

A direcao que leva o custo a $-\text{Inf}$:

Direcao

1: -2.000000

2: 0.000000

0: 0.000000

0: 0.000000

O algoritmo devolve então a direção viável que possui solução ilimitada.

3 Resultados

O desenvolvimento do algoritmo em linguagem Octave baseado no método simplex descrito em [Bertsimas e Tsitsiklis \(1997\)](#).

Além da análise do funcionamento do algoritmo, conforme descrito na seção anterior, que demonstrou a corretude do mesmo, desde que fossem garantidas as hipóteses sobre os parâmetros de inicialização.

Conclusão

O algoritmo simplex implementado para resolver problemas de programação linear com soluções viáveis básicas não degeneradas e com pelo menos uma ótima é correto.

Mantendo as hipóteses em todos os casos, obtêm-se os resultados esperados em problemas cuja solução ótima é dada, quando o custo ótimo é ilimitado, ou seja necessário encontrar uma direção viável cujo custo seja menor, encontrando a solução ótima.

Referências

BERTSIMAS, D.; TSITSIKLIS, J. *Introduction to Linear Optimization*. [S.l.]: Athena Scientific, 1997. (Athena Scientific series in optimization and neural computation). ISBN 9781886529199. Citado 2 vezes nas páginas [3](#) e [9](#).