

# Pesquisa Operacional - Problema da Corrente de Equilíbrio

Eduardo César<sup>1</sup>   Manassés Ferreira<sup>1</sup>   Marzo Júnior<sup>1</sup>  
Thiago Linke<sup>1</sup>

<sup>1</sup>Universidade Federal de Minas Gerais, Brasil

Pesquisa Operacional, 2013

# Resumo

1

## Modelagem

- O Problema
- O Modelo

2

## Resultados

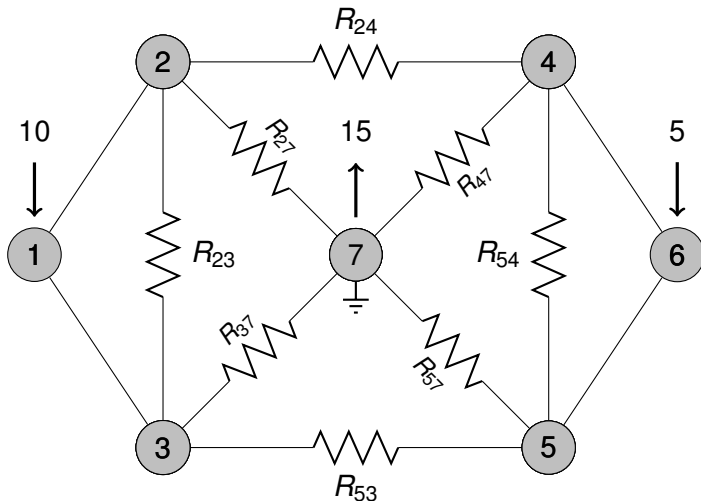
- Rede Exemplo
- Outras redes

3

## Conclusões

## Determinação da corrente de equilíbrio

Deseja-se determinar a corrente de equilíbrio que flui em um circuito elétrico como por exemplo:



# Modelos

Existe mais de uma forma de se resolver o este problema. Três dessas formas serão mostradas aqui:

- 1 Modelagem como sistema de equações lineares.
- 2 Modelagem com sistema quadrático
- 3 Modelagem como fluxo em redes com custo convexo.

## Modelagem como sistema linear

- Problema é resolvido introduzindo-se uma variável  $x_{ij}$  representando o fluxo de corrente no arco  $(i, j)$  do circuito elétrico e montar um sistema de equações de equilíbrio para estes fluxos. A solução para este sistema fornece a intensidade  $x_{ij}$  para cada arco respectivo.
- Baseia-se nos princípios físicos:
  - Lei de Ohm  $\rightarrow V = R \times I$
  - 1ª Lei de Kirchhof  $\rightarrow I_1 = I_2 + I_3$

Sistema  $Ax=b$ , com 'número de nós' equações, em que a matriz  $A$  representa a condutividade e o vetor independente  $b$ , o saldo de corrente.

## Modelagem como fluxo em redes com custo convexo

Esta formulação se utiliza de um comportamento conhecido de que as correntes de equilíbrio nos resistores são os fluxos para qual os resistores dissipam a menor potência total suprida pelas fonte de tensão (ou seja, a corrente elétrica segue o caminho de menor resistência.)

$$\text{Minimizar } \sum_{(i,j) \in A} r_{ij} x_{ij}^2 \quad \text{sujeito a}$$

$$\sum_{j:(i,j) \in A} x_{ij} - \sum_{i:(j,i) \in A} x_{ji} = b_i \quad \text{para cada nodo } i \in N,$$

$$x_{ij} \geq 0 \quad \text{para cada arco } (i,j) \in A.$$

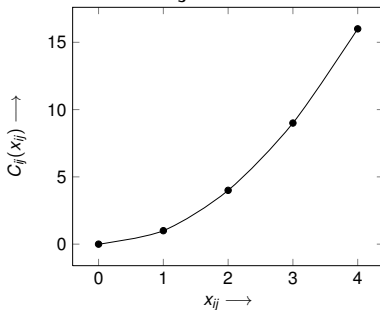
## Linearização do custo convexo

Aproximação por segmentação:

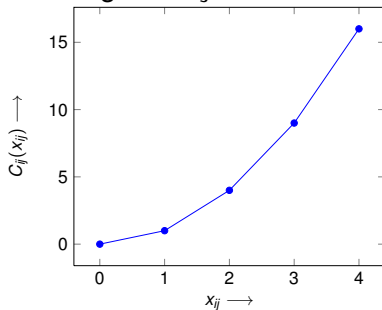
- Cada custo de arco  $C_{ij}(x_{ij})$  possui  $p$  segmentos lineares:  $0 = d_{ij}^0 < d_{ij}^1 < d_{ij}^2 < d_{ij} < \dots$ , que denotam os pontos onde a função “quebra”.
- custo varia linearmente no intervalo  $[d_{ij}^{k-1}, d_{ij}^k]$ . Denotamos  $c_{ij}^k$  como o coeficiente de custo linear no intervalo  $[d_{ij}^{k-1}, d_{ij}^k]$ .
- Sendo assim, para especificar o a função aproximada, precisamos especificar os segmentos e a inclinação da função nesses segmentos.

# Linearização do custo convexo

## Função $x^2$



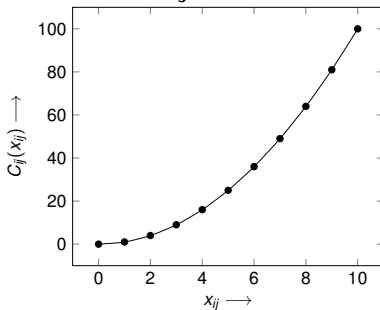
## Segmentação linear



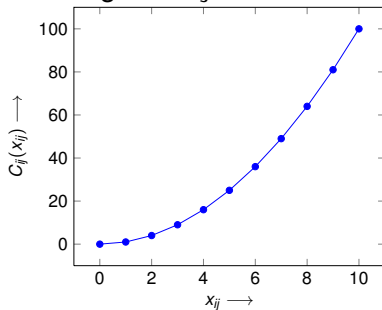


# Linearização do custo convexo

## Função $x^2$

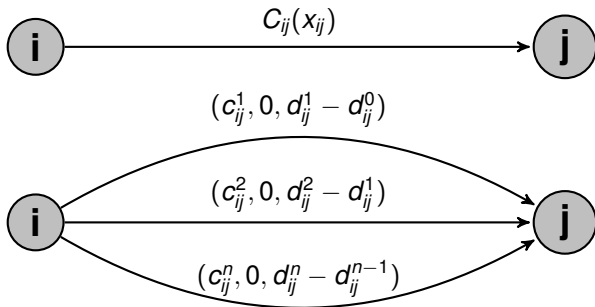


## Segmentação linear

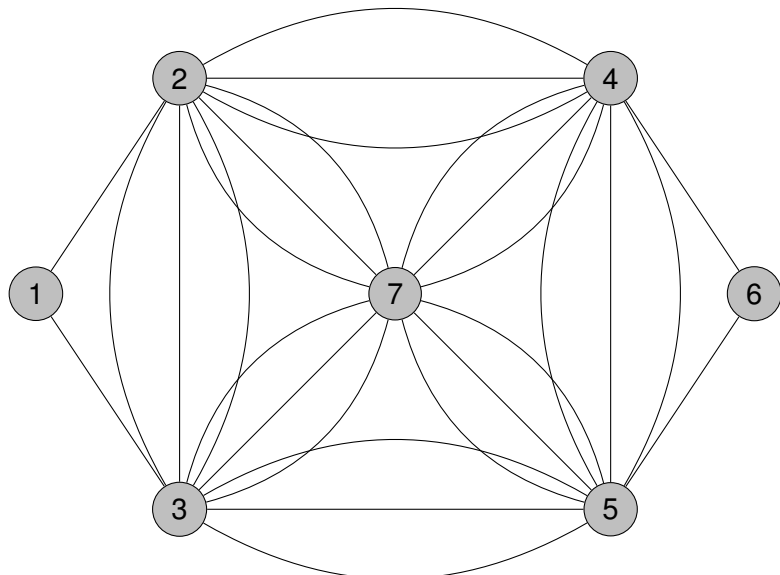


## Linearização do custo convexo

Arco original e arcos correspondentes na nova rede:



# Linearização do custo convexo



## Linearização do custo convexo

- Após a linearização pode-se usar o GLPK

## Linearização do custo convexo

- Após a linearização pode-se usar o GLPK
- Resolve-se o problema de custo mínimo

## Linearização do custo convexo

- Após a linearização pode-se usar o GLPK
- Resolve-se o problema de custo mínimo
- Há duas opções: out-of-kilter (okalg) e relaxation4 (relax4)

- melhor primal-dual. 1960 Minty. 1961 Fulkerson. 1962 Ford-Fulkerson.

- melhor primal-dual. 1960 Minty. 1961 Fulkerson. 1962 Ford-Fulkerson.
- ideia geral: similar ao caminho mínimo kilter number: mínimo aumento ou decremento para satisfazer restrições de fluxo e de viabilidade do dual iterativo até que todos os arcos estejam em 'ordem' (in kilter)



- melhor primal-dual. 1960 Minty. 1961 Fulkerson. 1962 Ford-Fulkerson.
- ideia geral: similar ao caminho mínimo kilter number: mínimo aumento ou decremento para satisfazer restrições de fluxo e de viabilidade do dual iterativo até que todos os arcos estejam em 'ordem' (in kilter)
- pior caso: cubico com numero de arcos

fonte:<http://www.derekroconnor.net/home/PAPERS/MMS-91.pdf>

- um dos mais eficientes para otimização em redes. 1985 Bertsekas. 1988 Bertsekas and Tseng.

- um dos mais eficientes para otimização em redes. 1985 Bertsekas. 1988 Bertsekas and Tseng.
- ideia geral: pseudoFlow, iteração baseada em aumento do fluxo e ajuste de 'preço' do dual.

- um dos mais eficientes para otimização em redes. 1985 Bertsekas. 1988 Bertsekas and Tseng.
- ideia geral: pseudoFlow, iteração baseada em aumento do fluxo e ajuste de 'preço' do dual.
- pior caso: pseudoPolinomial com numero de arcos

fonte: <http://www.dereкроconnor.net/home/PAPERS/MMS-91.pdf>

## Sistema quadrático

- Problema 1: GLPK não resolve funções objetivo que não sejam lineares

## Sistema quadrático

- Problema 1: GLPK não resolve funções objetivo que não sejam lineares
- Solução: O GNU Octave tem um solver para programação linear quadrática

## Sistema quadrático

- Problema 1: GLPK não resolve funções objetivo que não sejam lineares
- Solução: O GNU Octave tem um solver para programação linear quadrática
- Problema 2: GNU Octave não aceita GMPL

## Sistema quadrático

- Problema 1: GLPK não resolve funções objetivo que não sejam lineares
- Solução: O GNU Octave tem um solver para programação linear quadrática
- Problema 2: GNU Octave não aceita GMPL
- Solução: Programa simples converte grafo de entrada para entrada do GNU Octave



## Sistema quadrático

- Para tornar as matrizes mais compactas, vamos usar uma representação baseada em arestas presentes no grafo

## Sistema quadrático

- Para tornar as matrizes mais compactas, vamos usar uma representação baseada em arestas presentes no grafo
- Para tal, vamos definir um mapa  $F : A \longrightarrow I \subset \mathbb{N}$  que mapeia arestas em números

## Sistema quadrático

- Para tornar as matrizes mais compactas, vamos usar uma representação baseada em arestas presentes no grafo
- Para tal, vamos definir um mapa  $F : A \longrightarrow I \subset \mathbb{N}$  que mapeia arestas em números
- Definindo uma ordem total nas arestas e preservando a ordem por  $F$  garante conversão e inversão fáceis

## Sistema quadrático

- Em termos de  $F[(i, j)]$  com  $(i, j) \in A$ , temos:

$$r_{ij} \longrightarrow H_{ab}$$

Tal que

$$H_{ab} = 0 \qquad a \neq b$$

$$H_{F[(i, j)], F[(i, j)]} = r_{ij} \qquad \text{caso contrário}$$

# Sistema quadrático

- Em termos de  $F[(i, j)]$  com  $(i, j) \in A$ , temos:

$$r_{ij} \longrightarrow H_{ab}$$

Tal que

$$H_{ab} = 0 \quad a \neq b$$

$$H_{F[(i,j)], F[(i,j)]} = r_{ij} \quad \text{caso contrário}$$



$$A_{i, F[(i,j)]} = 1 \quad \forall i, j : (i, j) \in A$$

$$A_{i, F[(i,j)]} = -1 \quad \forall i, j : (j, i) \in A$$

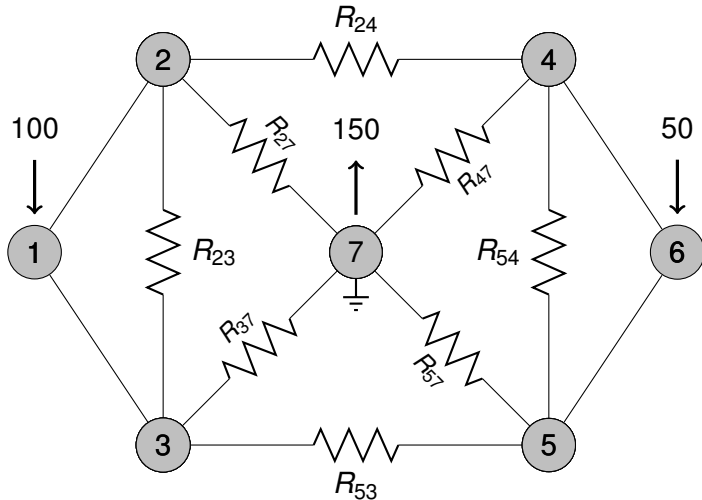
$$A_{a,b} = 0 \quad \text{Caso contrário}$$

## Sistema quadrático

Tendo uma conversão organizada e codificada em um programa, podemos usar o GNU Octave para resolver o sistema quadrático.

## Problema Original levemente Modificado

Assumindo resistências centenárias.



Rede Exemplo

# Sistema Linear - Numerical Recipes

1->2	2->7	2->4	2->3	3->5	3->2	4->6	5->6	6->4	7->5	tempo (ms)
50.000	50.000	8.333	41.667	8.333	41.667	25.000	33.333	25.000	33.333	2



Rede Exemplo

# Minimização Quadrática - Octave

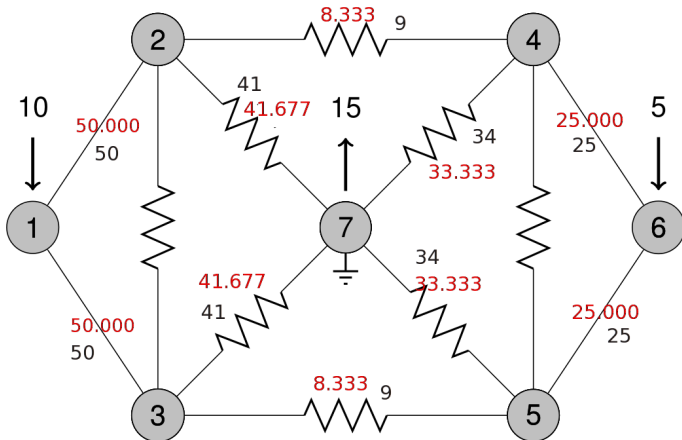
1->2	2->7	2->4	2->3	3->5	3->2	4->6	5->6	6->4	7->5	tempo (ms)
50.000	50.000	8.333	41.667	8.333	41.667	25.000	33.333	25.000	33.333	245

# Fluxo em Redes - Custo Mínimo - Out-of-Kilter - GLPK

## Tabela de segmentação dos fluxos

Segmento	1->2	2->7	2->4	2->3	3->5	3->2	4->6	5->6	6->4	7->5	tempo (ms)
1	0	100	100	0	0	0	50	0	0	50	3.355
2	25	75	75	0	25	0	50	0	0	50	3.948
5	40	60	60	0	40	0	30	20	20	30	4.687
10	45	55	45	10	45	0	30	30	30	20	5.627
15	50	50	50	0	40	10	30	30	20	30	8.428
20	48	52	44	8	40	8	34	32	24	26	5.398
30	50	50	40	10	40	10	35	35	25	25	9.488
50	49	51	42	9	42	7	33	33	26	24	15.303
100	50	50	42	8	42	8	34	32	24	26	23.834
200	50	50	42	8	42	8	34	32	24	26	38.798
300	50	50	41	9	41	9	34	34	25	25	42.686

## Resumindo



## Usando o net generator Klingman

amostra	octave	nr	okalg	relax4	nós	arcos
book	245	2	42	1	7	24
151	86000	4	1491	36	50	125**
101	-*	15min	28472	4548	5000	25000**
108	-	-	13516	187	5000	50000***
109	-	-	19479	243	5000	75000***

\* Octave - arquivo de entrada excedeu um tamanho razoável.

\*\* okalg e relax4 - arcos com segmentação 100

\*\*\* okalg e relax4 - arcos com segmentação 1

Tempo em (ms)

## Conclusões

- O equilíbrio está associado à condição de optimalidade do problema linear associado.

## Conclusões

- O equilíbrio está associado à condição de optimalidade do problema linear associado.
- O algoritmo em redes é mais eficiente que a resolução em sistemas de equações lineares.

## Conclusões

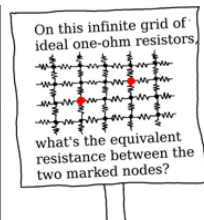
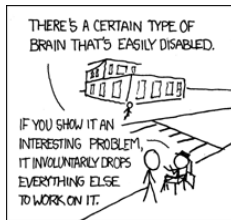
- O equilíbrio está associado à condição de optimalidade do problema linear associado.
- O algoritmo em redes é mais eficiente que a resolução em sistemas de equações lineares.
- Uma vez encontrado um fator de escala, o resultado inteiro é uma desvantagem aparente do método de custo mínimo. Recompensada pelo baixo custo computacional.

## Conclusões

- O equilíbrio está associado à condição de optimalidade do problema linear associado.
- O algoritmo em redes é mais eficiente que a resolução em sistemas de equações lineares.
- Uma vez encontrado um fator de escala, o resultado inteiro é uma desvantagem aparente do método de custo mínimo. Recompensada pelo baixo custo computacional.
- Octave e nr não escalam. okalg é muito bom, no entanto o relax4 é o algoritmo atualmente a ser batido.



# Dúvidas



Apresentação produzida usando

L<sup>A</sup>T<sub>E</sub>X

disponível em [goo.gl/1DtLf](https://goo.gl/1DtLf)