1 Problema

2 Problema

3 Problema

O problema colocado é descrito na literatura como Problema do Carteiro Chines Direcionado. O método que vamos apresentar para solucionar problemas desse tipo foi descrito por Thimbleby (2003) baseado no problema clásssico de transportes apresentado por Beltrami & Bodin (1972).

Seja um Digrafo $D = (V, A)$ com $V = \{v_1, v_2, v_3, ..., v_i\}$ o conjunto de vértices e $A = \{a_1, a_2, a_3, ...a_n\}$ o conjunto dos arcos, vamos dar algumas definições:

Grau de saída de v - o número total de vértices saindo de v, denotaremos por $d^+(v)$

Grau de entrada de v - o número total de vértices entrando em v, denotaremos por $d^-(v)$

Vértice balanceado - quando o grau de entrada é igual ao grau de saída. Um vértice é balanceado se $\delta(v) = d^(v) - d^-(v) = 0$ e desbalanceado nos casos que $\delta \neq 0$.

Variáveis de decisão - denotadas por $x_{i,j}$ a quantidade de arcos a serem acrescentados no caminho $i \rightarrow j$

Custo - denotado por $c_{i,j}$ se refere ao peso do caminho $i \rightarrow j$

Denotaremos por $D^+ = \{v|\delta(v) > 0\}$ o conjunto de vértices desbalanceados com excesso de arcos de saída.

Denotaremos por $D^- = \{v|\delta(v) < 0\}$ o conjunto de vértices desbalanceados com excesso de arcos de entrada.

3.1 Algoritmo

1. Determinar $\delta(v)$ para cada vertice, D^+, D^- e $c_{i,j}$.

2. Encontrar a função objetivo Z a ser minimizada. Ela é encontrada resolvendo o seguinte Problema de Programação Linear (PPL):

$$min \sum c_{i,j} x_{i,j}$$

sujeito as seguintes restrições:

$$\sum_{j \in D^+} x_{i,j} = -\delta(i)$$

$$\sum_{i \in D^-} x_{i,j} = \delta(j)$$

$x_{i,j}$ são inteiros positivos.

3. Construir o ciclo euleriano com base no algoritmo de Hierholzer.

Na figura 1 mostramos um fluxograma para a solução de problemas deste tipo.

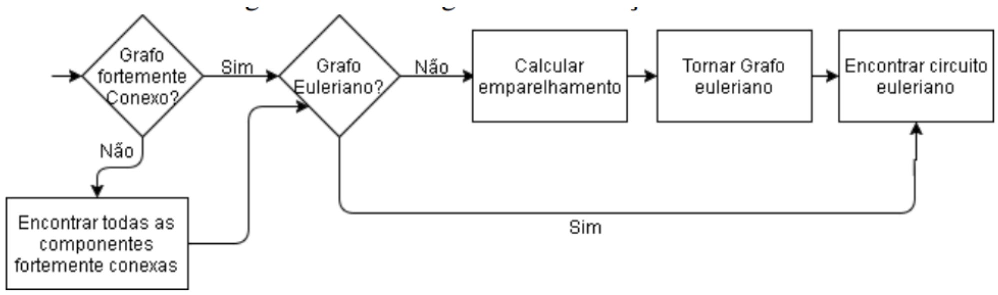

Figure 1: Fluxograma para o Problema do Carteiro Chinês Direcionado

O Problema do Carteiro Chinês direcionado pode ser facilmente solucionado se todos vértices do dígrafo forem balanceados. Neste trabalho iremos apenas focar em dígrafos que apresentam vértices desbalanceados.

3.2 Exemplo

Seja um dígrafo D orientado e valorado apresentado na figura 2. Vamos encontrar nele um ciclo euleriano.

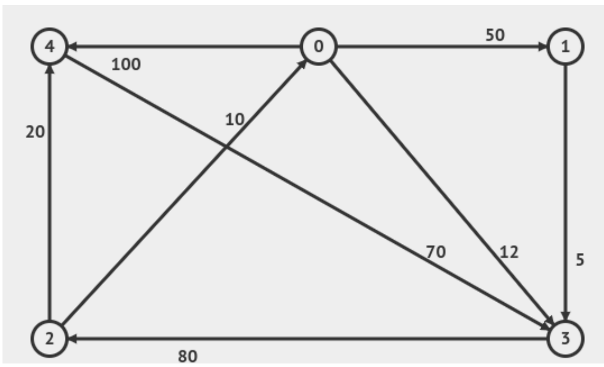

Figure 2: Dígrafo D.

Vamos determinar $\delta(v)$.

$\delta(0) = 2$, $\delta(1) = 0$, $\delta(2) = 1$, $\delta(3) = -2$ e $\delta(4) = -1$

Os vértices 0, 2, 3 e 4 estão desbalanceados. Apenas o vértice 1 está balanceado.

Assim podemos criar dois conjuntos $D^- = \{3, 4\}$ e $D^+ = \{0, 2\}$. Juntos formam um subgrafo D', bipartido, contendo os vértices 0, 2, 3 e 4 Como ilustra figura 3

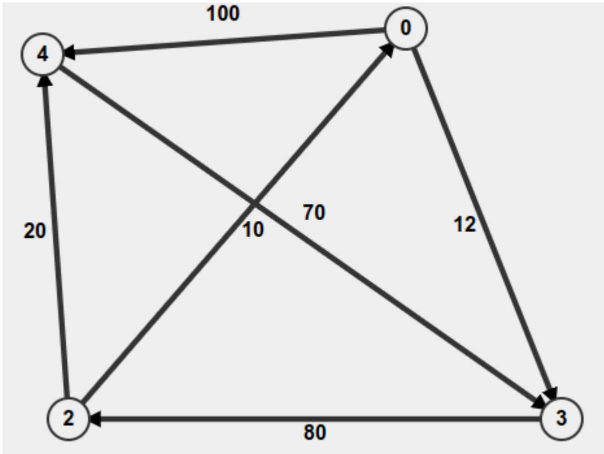

Figure 3: Subgrafo de G.

Para determinar os custos $c_{i,j}$ vamos calcular as distâncias mínimas entre os vértices usando o algoritmo de Dijkstra. Em alternativa pode ser também usado o algoritmo de Floyd-Warshall.

Origem	**Destino**	**Distância**	**Caminho**
0	2	92	0-3-2
0	3	12	0-3
0	4	100	0-4
2	0	10	2-0
2	3	22	2-0-3
2	4	20	2-4
3	0	90	3-2-0
3	2	80	3-2
3	4	100	3-2-4
4	0	160	4-3-2-0
4	2	150	4-3-2
4	3	70	4-3

Table 1: Distância mínima entre os vértices usando o Algoritmo de Dijkstra.

Podemos agora representar os arcos no dígrafo D', de modo que cada arco tenha distância mínima calculada anteriormente. Isso transforma D' em um K_{12}. A figura 4 ilustra isso:

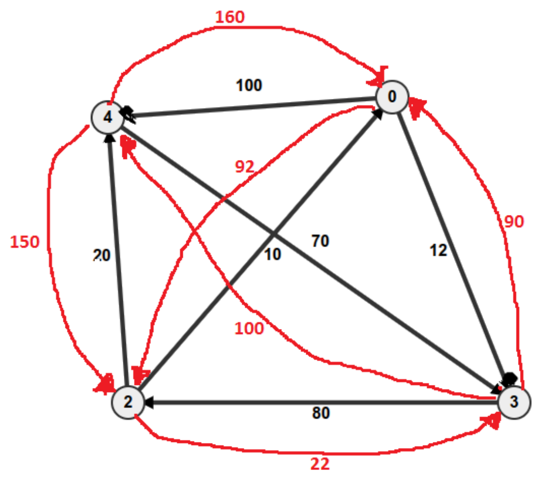

Figure 4: Subgrafo completo de D com as distâncias mínimas inseridas.

O problema agora passa por solucionar quais dos arcos (na cor vermelha) serão mantidos, removidos ou duplicados para que D' tenha um ciclo euleriano com custo mínimo.

Para isso podemos usar o modelo de PPL descrito no passo 2 do algoritmo.

$minZ$

$Z = 92x_{0,2} + 12x_{0,3} + 100x_{0,4} + 10x_{2,0} + 22x_{2,3} + 20x_{2,4} + 90x_{3,0} + 80x_{3,2} + 100x_{3,4} + 160x_{4,0} + 150x_{4,2} + 70x_{4,3}$

sujeito à:

- Restrição para o vértice **0**: $x_{3,0} + x_{3,2} = 2$
- Restrição para o vértice **2**: $x_{4,0} + x_{4,2} = 1$
- Restrição para o vértice **3**: $x_{3,0} + x_{4,0} = 2$
- Restrição para o vértice **4**: $x_{2,3} + x_{2,4} = 1$

Os resultados para a execução do modelo de PPL anterior estão descritos na tabela 2. Este resultado nos informa que o grafo euleriano será obtido duplicando os arcos no caminho mínimo entre os vértices 3 e 0, entre os vértices 3 e 2 e entre os vértices 4 e 0.

Variável	$x_{0,4}$	$x_{3,0}$	$x_{0,3}$	$x_{0,2}$	$x_{3,4}$	$x_{4,2}$	$x_{3,2}$	$x_{2,3}$	$x_{4,0}$	$x_{2,4}$	$x_{2,0}$	$x_{4,3}$
Valor	0	1	0	0	0	0	1	0	1	0	0	0

Table 2: Resultado da execução do modelo de PPL.

De acordo com a tabela 1 que nos mostra a distancia mínima, no caminho 3 e 0 temos 2 arcos, no caminho 3 e 2 temos 1 arcos e no caminho 4 e 0 temos 3 arcos. Ao todo temos 6 arcos.

A figura mostra o dígrafo após inserção dos novos arcos para formar o ciclo euleriano.

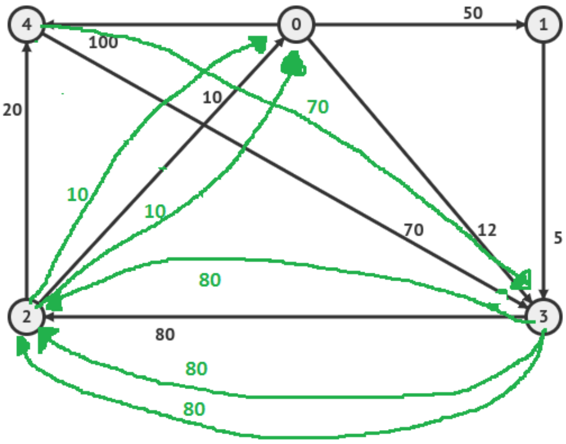

Figure 5: Dígrafo D com ciclo euleriano.

O algoritmo de Hierholzer (que é uma variação do algoritmo de Fleury para grafos orientados) pode ser usado para encontrar o ciclo euleriano. O ciclo euleriano para este exemplo será: 0-3-2-0-4-3-2-0-1-3-2-4-3-2-0. O custo mínimo é obtido somando o peso de todos arcos percorridos, no caso 677.

3.3 Complexidade

A complexidade do método descrito acima depende do algoritmo usado para solucionar o modelo de PPL. O algoritmo descrito acima quando empregado com o método simplex pode oferecer uma complexidade exponencial dependendo do número de vértices do subgrafo D e do número de restrições. Porém, outros métodos como o método de pontos interiores são mais eficientes e quando empregados torna esse algoritmo eficiente entregando uma complexidade $O(mn^2)$.

3.4 Código Java

Este código está disponível no github Clique aqui

Ou pelo link: https://github.com/feliperasan/Eulerian-Graph

4 Referências

Beltrami, E.J. & Bodin, L.D.(1972). *Networks and vehicle routing for municipal waste collection.* Report No. UPS 72-18, State University of New York. Stony Brook, NY.

Thimbleby, H. (2003). *The directed Chinese Postman Problem.* Softw., Pract. Exper.. 33. 1081-1096. 10.1002/spe.540.

Verberk, L. (2019). *The Chinese postman problem in undirected and directed graphs.* Eindhoven University of Technology.