Modelos Determinísticos de

Investigação Operacional

Uma imagem com ClipArt

Descrição gerada automaticamente

Trabalho prático nº2

29 de outubro de 2019

**Grupo nº 11**

Filipa Alves dos Santos (A83631)

Hugo André Coelho Cardoso (A85006)

João da Cunha e Costa (A84775)

Válter Ferreira Picas Carvalho (A84464)

Uma imagem com pessoa, parede, interior, jovem

Descrição gerada automaticamenteUma imagem com pessoa, parede, homem, interior

Descrição gerada automaticamenteUma imagem com pessoa, céu, homem, parede

Descrição gerada automaticamenteUma imagem com pessoa, parede, vestuário, sorriso

Descrição gerada automaticamente

Mestrado Integrado em Engenharia Informática

Universidade do Minho

**Índice de conteúdos**

**1. Introdução 3**

**2. Desenvolvimento 3**

2.1. Parte I 3

2.1.1. Modelo de Programação Linear Original 3

2.1.2. Modelo de Programação Linear Novo 4

2.1.3. Rede de Transporte Nova 5

2.1.4. Ficheiro Input (Relax4) 6

2.1.5. Ficheiro Output (Relax4) 7

2.1.6. Interpretação da Solução 8

2.1.7. Validação do Modelo 8

2.2. Parte II 9

2.2.1. Grafo Bipartido do Problema de Transporte 9

2.2.2. Balanceamento do Problema de Transporte 10

2.2.3. Caminhos Mais Curtos entre os Vértices Relevantes 10

2.2.4. Ficheiro Input (Relax4) 11

2.2.5. Ficheiro Output (Relax4) 12

2.2.6. Interpretação da Solução 13

2.2.7. Validação do Modelo 13

**3. Conclusão 14**

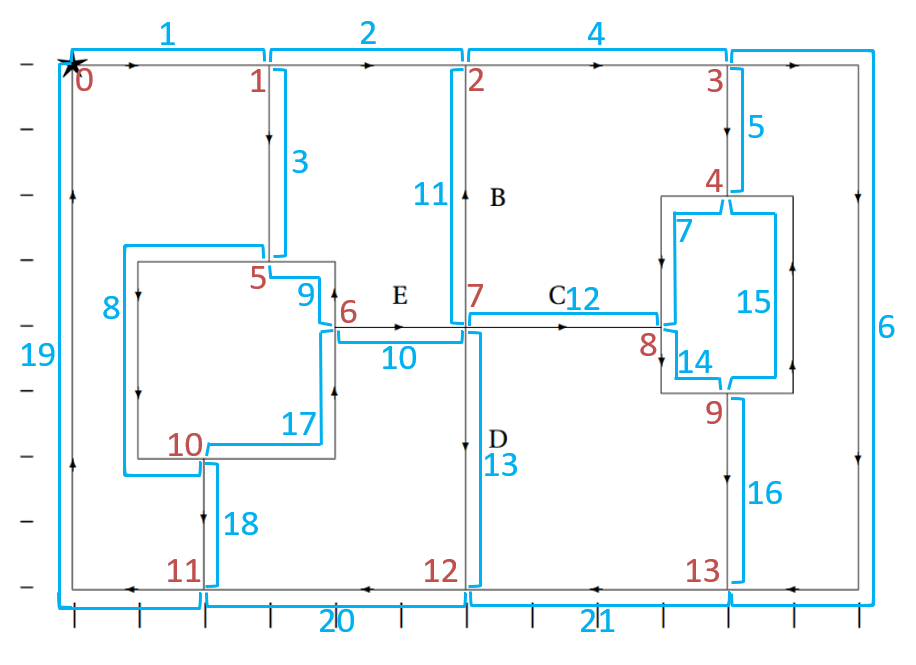
1. **Introdução**

Neste trabalho prático, era pretendido analisar o problema inicial do Trabalho Prático 1 numa outra ótica. Inicialmente, era analisado o fluxo em cada vértice do grafo orientado (rede de ruas) fornecido, injetando uma unidade no vértice inicial e seguindo o seu percurso, de modo a percorrer as várias ruas no mínimo uma vez e em cada vértice o número de unidades que saem é igual ao número de unidades que entram no mesmo. Tendo isto em conta, na primeira parte do trabalho, o que fizemos foi escolher uma nova variável de decisão, que era resultado de decrementar uma unidade à variável de decisão do primeiro projeto, ou seja, garantimos a existência de um limite inferior de fluxos em cada arco e, portanto, passamos a analisar o problema como um problema de transportes numa rede geral, utilizando o Relax como software auxiliar de resolução. Obtivemos, portanto, o número de vezes adicionais que cada arco é percorrido ao fazer os caminhos de reposicionamento determinados na segunda parte do projeto. Já na segunda parte do trabalho, consideramos o grafo como um grafo bipartido, dividido em vértices de excesso (nos quais entram mais caminho do que saem) e vértices de defeito (vice-versa). Com esta estratégia, e sabendo que os caminhos de reposicionamento do veículo começam num vértice por excesso e acabam num vértice por defeito, torna-se possível descobrir estes mesmos caminhos de modo a minimizar os custos associados.

1. **Desenvolvimento**

**2.1. Parte I**

**2.1.1. Modelo de Programação Linear Original**

 Considerando que 85006 é o maior número de aluno dos membros do nosso grupo, determinou-se que o sentido da rua B é a subir, o da D é a descer e o das C e E é para a direita. Abaixo apresentamos a rede da cidade atualizada, com estes sentidos já indicados:

(Este modelo está apresentado no relatório do trabalho prático anterior.)

**2.1.2. Modelo de Programação Linear Novo**

Para efetuar a mudança de variável pedida, yij = xij – lij, determinamos que lij, o limite inferior do fluxo do arco, é 1 dado que pretendemos que cada arco seja percorrido no mínimo uma vez. Assim, temos que substituir todos xij nas restrições e na função objetivo do modelo anterior por yij + 1. Por exemplo, na restrição correspondente ao vértice 1 tínhamos a expressão x1 – x2 – x3 = 0 e após efetuarmos a substituição, temos (y1+1) – (y2+1) – (y3+1) = 0, que equivale a y1 – y2 – y3 – 1 = 0.

Assim, o novo modelo de Programação Linear é o seguinte:

Declaração das variáveis:

int y1, y2, y3, y4, y5, y6, y7, y8, y9, y10, y11, y12, y13, y14, y15, y16, y17, y18, y19, y20, y21;

y1, y2, y3, y4, y5, y6, y7, y8, y9, y10, y11, y12, y13, y14, y15, y16, y17, y18, y19, y20, y21 >= 0;

Função Objetivo:

min: 3 y1 + 3 y2 + 3 y3 + 4 y4 + 2 y5 + 12 y6 + 3 y7 + 6 y8 + 2 y9 + 2 y10 + 4 y11 + 3 y12 + 4 y13 + 2 y14 + 5 y15 + 3 y16 + 4 y17 + 2 y18 + 10 y19 + 4 y20 + 4 y21 + 85;

Restrições de fluxo:

* Vértice 0: y19 - y1 = 0;
* Vértice 1: y1 - y2 - y3 - 1 = 0;
* Vértice 2: y2 + y11 - y4 + 1 = 0;
* Vértice 3: y4 - y5 - y6 - 1 = 0;
* Vértice 4: y5 + y15 - y7 + 1 = 0;
* Vértice 5: y3 + y9 - y8 + 1 = 0;
* Vértice 6: y17 - y9 - y10 - 1 = 0;
* Vértice 7: y10 - y11 - y12 - y13 - 2 = 0;
* Vértice 8: y7 + y12 - y14 + 1 = 0;
* Vértice 9: y14 - y15 - y16 - 1 = 0;
* Vértice 10: y8 - y17 - y18 - 1 = 0;
* Vértice 11: y18 + y20 - y19 + 1 = 0;
* Vértice 12: y13 + y21 - y20 + 1 = 0;
* Vértice 13: y6 + y16 - y21 + 1 = 0;

**2.1.3. Rede de Transporte Nova**

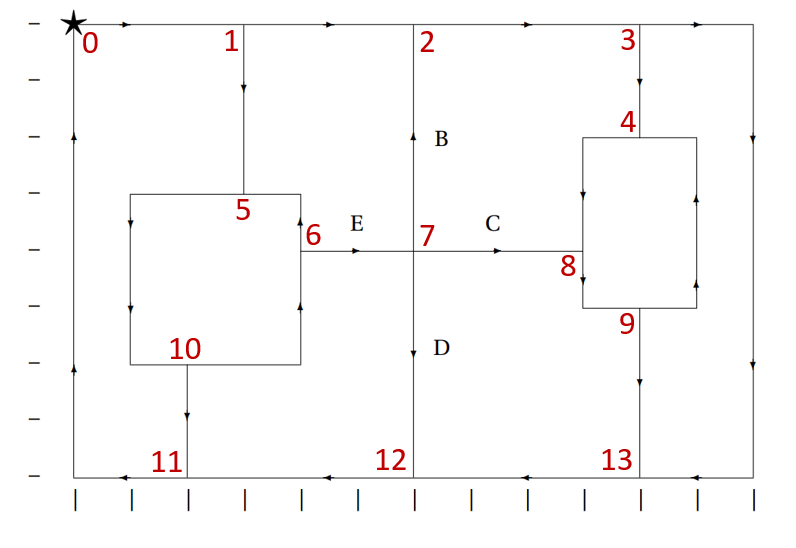
Os vértices com mais arcos a entrar neles do que a sair são chamados de vértices de procura / excesso enquanto vértices com mais a sair do que a entrar são vértices de oferta / defeito. Como definimos no trabalho anterior, os arcos a entrar num certo vértice são considerados um aumento no fluxo e os sair, uma diminuição deste. Estes valores do fluxo podem ser observados nas restrições do modelo de programação facilmente sendo que equivalente ao membro independente da equação - por exemplo, no vértice 1 o fluxo é -1 (y1 – y2 – y3 – 1 = 0).

Na figura seguinte da rede, estão representados os fluxos de cada vértice a azul:

+11

-1

-1



+1

+1

+1

-1

-1

+1

-2

+1

-1

+1

**2.1.4. Ficheiro Input (Relax4)**

14

21

1 2 3 100

1 5 3 100

2 3 4 100

3 4 2 100

3 13 12 100

4 8 3 100

5 10 6 100

6 5 2 100

6 7 2 100

7 2 4 100

7 8 3 100

7 12 4 100

8 9 2 100

9 4 5 100

9 13 3 100

10 6 4 100

10 11 2 100

11 14 10 100

12 11 4 100

13 12 4 100

14 1 3 100

-1

1

-1

1

1

-1

-2

1

-1

-1

1

1

1

0

**2.1.5. Ficheiro Output (Relax4)**

END OF READING

NUMBER OF NODES = 14, NUMBER OF ARCS = 21

CONSTRUCT LINKED LISTS FOR THE PROBLEM

CALLING RELAX4 TO SOLVE THE PROBLEM

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TOTAL SOLUTION TIME = 0. SECS.

TIME IN INITIALIZATION = 0. SECS.

1 5 3.

2 3 1.

4 8 1.

5 10 4.

6 7 2.

8 9 2.

9 13 1.

10 6 3.

11 14 4.

12 11 3.

13 12 2.

14 1 4.

OPTIMAL COST = 135.

NUMBER OF AUCTION/SHORTEST PATH ITERATIONS = 62

NUMBER OF ITERATIONS = 18

NUMBER OF MULTINODE ITERATIONS = 2

NUMBER OF MULTINODE ASCENT STEPS = 4

NUMBER OF REGULAR AUGMENTATIONS = 3

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**2.1.6. Interpretação da Solução**

Através do ficheiro output do Relax4, obtivemos o resultado *135,* que corresponde ao custo de percorrer os arcos, apenas os que são percorridos mais que uma vez, cujo vértice de origem é de excesso e o de destino é de defeito. Este arcos correspondem aos caminhos de reposicionamento do veículo. Se somarmos o custo de percorrer uma só vez cada arco (*85*), ao resultado obtido no Relax4 (*135*) chegamos ao valor *220*, que é o valor da solução ótima do 1º trabalho prático.

O conjunto ótimo de arcos de reposicionamento pode ser retirado pelo output apresentado anteriormente. Isto é, os arcos cuja origem é a primeira coluna do ficheiro e o destino é segunda. Por ordem, temos os arcos: 3, 4, 7, 8, 10, 14, 16, 17, 19, 20, 21 e 1. Podemos ver a correspondência gráfica na figura da secção 2.1.1.

**2.1.7. Validação do Modelo**

Para validarmos o nosso modelo, temos que substituir as restrições com os valores obtidos no ficheiro output do Relax4. Por exemplo, para a restrição do vértice 1 (y1 – y2 – y3 – 1 = 0), vamos usar os valores obtidos correspondentes aos arcos 1 (do vértice 0 - representado como 14 no Relax4 - para o 1), 2 (do vértice 1 para o 2) e 3 (do vértice 1 para o 5), que se encontram 3ª coluna do ficheiro output. Quando estes arcos não estão representados no output, significa que não há reposicionamento nesses arcos, como é o caso do arco 2 nesta restrição. Substituindo, obtemos 4 – 0 – 3 – 1= 0 que é equivalente a 0 = 0. Para além de fazer este raciocínio com todos as outras restrições de fluxo, também confirmamos que todos os valores de yij são >= 0.

**2.2. Parte II**

**2.2.1. Grafo Bipartido do Problema de Transporte**

De seguida, podemos observar o grafo bipartido correspondente ao problema. À esquerda vemos os vértices de excesso e respetivos valores de procura, e à direita os vértices de defeito e respetivos valores de oferta. É possível observar os custos unitários de forma clara na tabela apresentada na alínea seguinte.

**1**

**1**

**1**

**1**

**1**

**1**

**1**

**-1**

**-1**

**-1**

**-1**

**-2**

**-1**

**2.2.2. Balanceamento do Problema de Transporte**

Seja o grafo da figura representado por , isto é, grafo cujo conjunto de vértices é partido em E (representam vértices de excesso) e D (representa vértices de defeito) e em que todos os arcos ligam uma origem a um destino . Para o grafo ser balanceado, o número total de caminhos que saem dos vértices de excesso é igual ao número de caminhos que chegam aos vértices de defeito.

Para o conjunto E, temos que o número total de caminhos acima referidos é dado por:

Para o conjunto D, temos que o número total de caminhos também acima referidos é dado por:

Como os dois somatórios são iguais, verifica-se a condição acima imposta e conclui-se que o grafo é, de facto, balanceado.

**2.2.3. Caminhos Mais Curtos entre os Vértices Relevantes**

Os valores na tabela representam os custos unitários de transporte dos caminhos mais curtos de um vértice de excesso i para um vértice de defeito j, calculados por inspeção do mapa fornecido no enunciado.

Conjunto dos vértices de excesso: i = {2,4,5,8,11,12,13} – coluna da tabela

Conjunto dos vértices de defeito: j = {1,3,6,7,9,10} – linha da tabela

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 3 | 6 | 7 | 9 | 10 |
| 2 | 35 | 4 | 48 | 50 | 11 | 44 |
| 4 | 29 | 36 | 42 | 44 | 5 | 38 |
| 5 | 21 | 28 | 10 | 12 | 17 | 6 |
| 8 | 26 | 33 | 39 | 41 | 2 | 35 |
| 11 | 13 | 20 | 26 | 28 | 27 | 22 |
| 12 | 17 | 24 | 30 | 32 | 31 | 26 |
| 13 | 21 | 28 | 34 | 36 | 35 | 31 |

**2.2.4. Ficheiro de input (Relax4)**

13

42

2 1 35 100

2 3 4 100

2 6 48 100

2 7 50 100

2 9 11 100

2 10 44 100

4 1 29 100

4 3 36 100

4 6 42 100

4 7 44 100

4 9 5 100

4 10 38 100

5 1 21 100

5 3 28 100

5 6 10 100

5 7 12 100

5 9 17 100

5 10 6 100

8 1 26 100

8 3 33 100

8 6 39 100

8 7 41 100

8 9 2 100

8 10 35 100

11 1 13 100

11 3 20 100

11 6 26 100

11 7 28 100

11 9 27 100

11 10 22 100

12 1 17 100

12 3 24 100

12 6 30 100

12 7 32 100

12 9 31 100

12 10 26 100

13 1 21 100

13 3 28 100

13 6 34 100

13 7 36 100

13 9 35 100

13 10 31 100

-1

1

-1

1

1

-1

-2

1

-1

-1

1

1

1

**2.2.5. Ficheiro de output (Relax4)**

END OF READING

NUMBER OF NODES = 13, NUMBER OF ARCS = 42

CONSTRUCT LINKED LISTS FOR THE PROBLEM

CALLING RELAX4 TO SOLVE THE PROBLEM

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TOTAL SOLUTION TIME = 0. SECS.

TIME IN INITIALIZATION = 0. SECS.

2 3 1.

4 6 1.

5 10 1.

8 9 1.

11 7 1.

12 1 1.

13 7 1.

OPTIMAL COST = 135.

NUMBER OF AUCTION/SHORTEST PATH ITERATIONS = 24

NUMBER OF ITERATIONS = 21

NUMBER OF MULTINODE ITERATIONS = 3

NUMBER OF MULTINODE ASCENT STEPS = 1

NUMBER OF REGULAR AUGMENTATIONS = 3

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**2.2.6. Interpretação da solução ótima**

Seja C o conjunto ótimo dos caminhos de reposicionamento xi j do veículo entre os vértices de excesso i e os vértices de defeito j. Segundo o Relax4,

C = {x2 3, x4 6, x5 10, x8 9, x11 7, x12 1, x13 7}

Como podemos observar no ficheiro de output, a soma dos custos destes caminhos é 135. Além disso, temos ainda que o custo de percorrer todos os arcos uma vez é 85.

Logo, o custo da solução ótima é 135 + 85 = 220, tal como no primeiro trabalho prático.

**2.2.7. Validação do modelo**

A substituição dos valores da solução ótima, ou seja, o número de vezes que cada arco é atravessado em caminhos de reposicionamento, na função objetivo e restrições permite verificar a validade da resolução:

**Função Objetivo:** 3\*4 + 3\*0 + 3\*3 + 4\*1 + 2\*0 + 12\*0 + 3\*1 + 6\*4 + 2\*0 + 2\*2 + 4\*0 + 3\*0 + 4\*0 + 2\*2 + 5\*0 + 3\*1 + 4\*3 + 2\*0 + 10\*4 + 4\*3 + 4\*2 + 85 = 220

**Restrições:**

* Vértice 0: 4 - 4 = 0;
* Vértice 1: 4 - 0 -3 -1 = 0;
* Vértice 2: 0 + 0 - 1 + 1 = 0;
* Vértice 3: 1 - 0 - 0 - 1 = 0;
* Vértice 4: 0 + 0 - 1 + 1 = 0;
* Vértice 5: 3 + 0 - 4 + 1 = 0;
* Vértice 6: 3 - 0 - 2 - 1 = 0;
* Vértice 7: 2 - 0 - 0 - 0 - 2 = 0;
* Vértice 8: 1 + 0 - 2 + 1 = 0;
* Vértice 9: 2 - 0 - 1 - 1 = 0;
* Vértice 10: 4 - 3 - 0 - 1 = 0;
* Vértice 11: 0 + 3 - 4 + 1 = 0;
* Vértice 12: 0 + 2 - 3 + 1 = 0;
* Vértice 13: 0 + 1 - 2 + 1 = 0;

1. **Conclusão**

Este trabalho incentivou-nos a ter um pensamento lógico de modo a achar as restrições necessárias para resolver o problema e um olhar crítico ao observar os resultados, pois tivemos que perceber se os valores obtidos faziam sentido no contexto do problema e qual o percurso feito de modo a atingir a solução ótima, além de averiguar se a solução correspondia à solução obtida no trabalho prático 1.

Concluimos também que era possível resolver o mesmo problema usando dois métodos completamente distintos. Consideramos que foi um trabalho bem sucedido e pretendemos ter resultados semelhantes em futuros trabalhos.