Universidade do Minho
Mestrado Integrado em Engenharia Informática
3ºano - 1º Semestre

Modelos Determinísticos de Investigação Operacional

Relatório sobre o Trabalho 2

Grupo de trabalho prático experimental 34

   

a83732 – Gonçalo Rodrigues Pinto
a84197 – João Pedro Araújo Parente
a84829 – José Nuno Martins da Costa
a85059 – Diogo Paulo Lopes de Vasconcelos

29 de Novembro de 2019

Conteúdo

Lista de Figuras

1 Introdução

No 1º semestre do 3º ano do Curso de Engenharia Informática da Universidade do Minho, existe uma Unidade Curricular denominada por Modelos Determinísticos de Investigação Operacional, que tem como objectivo ajudar os estudantes a desenvolver a sua capacidade de resolução de problemas com ênfase em problemas de engenharia de sistemas, dar a conhecer as técnicas e os métodos de Investigação Operacional, e consequentemente permitir que os alunos sejam capazes de aplicar estas técnicas e métodos na resolução de instâncias de problemas de pequena dimensão. Tem também como meta desenvolver a capacidade dos estudantes na análise de sistemas complexos, de criar modelos para os descrever, de obter soluções para esses modelos utilizando programas computacionais adequados, de validar os modelos obtidos, de interpretar as soluções obtidas, de elaborar recomendações para o sistema em análise e por fim, permitir que os alunos sejam capazes de compreender a importância da avaliação das soluções e de realizar análises de sensibilidade.

O presente trabalho pretende implementar em duas fases, um problema típico de Investigação Operacional que consiste em determinar o circuito ou conjunto de circuitos em que todos os arcos de um grafo são percorridos, pelo menos uma vez, minimizando a distância total percorrida. Um arco se necessário pode ser atravessado mais que uma vez. Para solucionar este problema aplicamos técnicas e métodos que aprendemos nesta unidade curricular tentando cumprir e seguindo os objetivos da mesma. Neste relatório vamos descrever as diferentes etapas, soluções e estratégias implementadas para solucionar o problema apresentado.

2 Problema

O problema a solucionar, tal como foi dito anteriormente, é determinar o circuito ou conjunto de circuitos em que todos os arcos de um grafo são percorridos, pelo menos uma vez, minimizando a distância total percorrida. Na figura 1 é apresentado o grafo que se pretende determinar o circuito. Na prática este problema pode ser traduzido na forma de minimizar a distância/trajecto que um veículo de recolha de lixo tem de percorrer determinadas ruas, ruas essas, que têm a particularidade de terem um único sentido, para recolher os sacos existentes ao longo da rua, ou quando há um sistema automático de recolha de contentores, todos localizados no mesmo lado da rua, cuja recolha obriga a percorrer a rua num determinado sentido. O veículo parte do depósito localizado no ponto assinalado com uma estrela e regressa ao mesmo ponto. Considerámos que as ruas têm um comprimento inteiro, proporcional à dimensão do seu traço em centímetros. Todas as ruas têm apenas um sentido, indicado pelas setas. O problema poderá não ter solução, para tal, basta haver um vértice em que só haja arcos a entrar. Por outro lado, para mostrar que existe solução temos que descobrir um caminho de qualquer vértice para qualquer outro vértice do grafo.

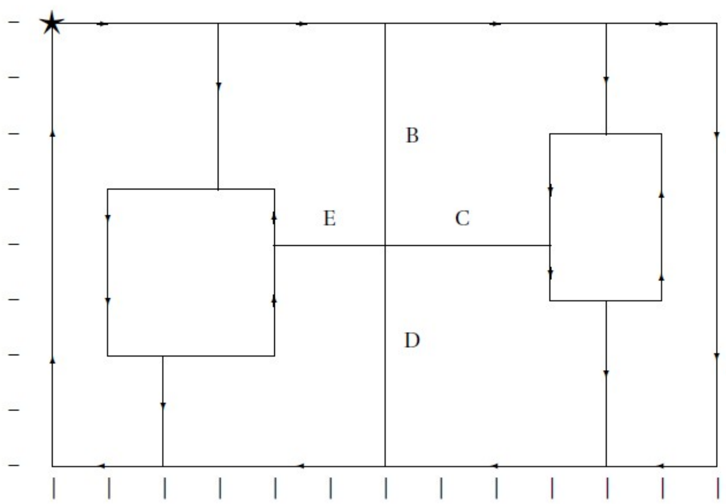

Figura 1: Grafo que se pretende determinar o circuito em que todos os seus arcos são percorridos.

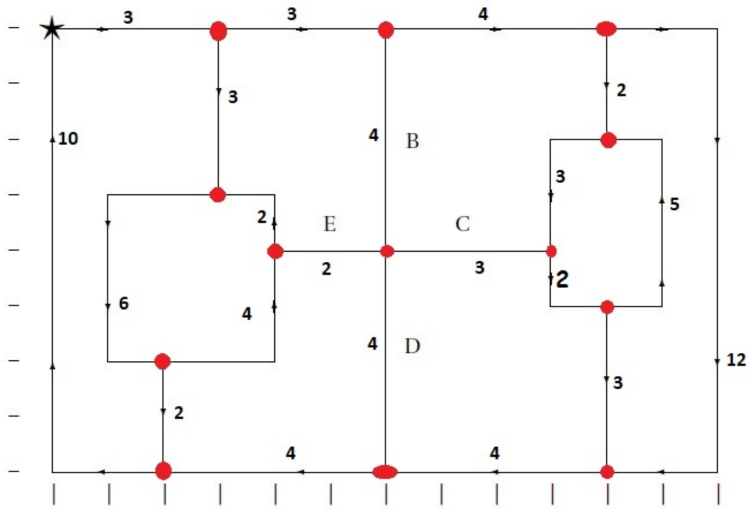

Figura 2: A distância associada a cada arco.

3 Parte I

3.1 Sentido das ruas BCDE

Conforme a figura 1 o sentido dos arcos já está definido, excepto os arcos BCDE que são determinados pelo número de inscrição do aluno do grupo com maior número. No nosso caso, o elemento com o maior número de inscrição é o Diogo Vasconcelos com o número 85059. Pelas regras apresentadas do enunciado, B é a subir porque 5 é ímpar, C é para a direita porque 0 é par, D é a subir porque 5 é ímpar e por fim E é para esquerda porque 9 é ímpar, assim podemos concluir desde já que não existem vértices em que haja apenas arcos a entrar ou a sair como se apresenta na figura 3.

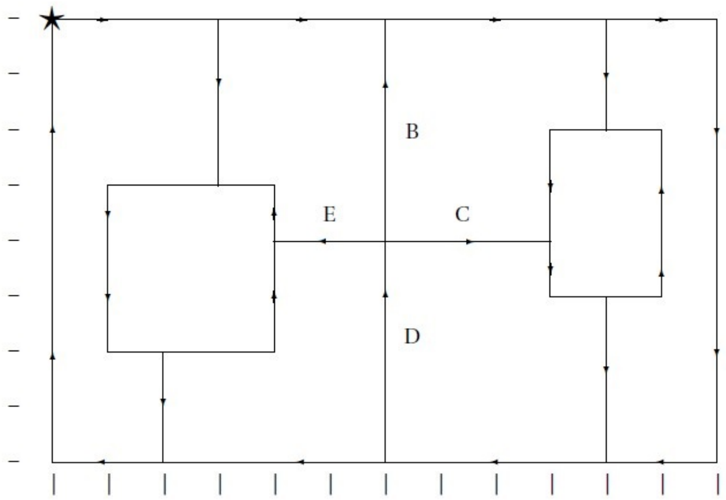

Figura 3: Definição dos sentidos dos arcos BCDE do grafo apresentado anteriormente.

3.2 Formulação como um problema de transporte

Dado o problema apresentado pretendemos determinar o caminho que percorre todos os arcos e que seja ao mesmo tempo o mais curto, para atingir esse fim recorremos à formulação como um problema de transporte numa rede geral transmitido e sugerido para a realização do problema. De modo a facilitar a compreensão do problema, decidimos a cada intersecção de arcos atribuir uma letra, sendo que estas variam entre A' e 'M', cada arco identificado pela junção de duas intersecções, por exemplo, o arco entre as intersecções 'A' e 'B' vai ser tratado de 'AB'. A origem representada por uma estrela, vai ser denominado de 'O', tem um estatuto especial pois é exigido que o caminho comece e termine neste ponto.

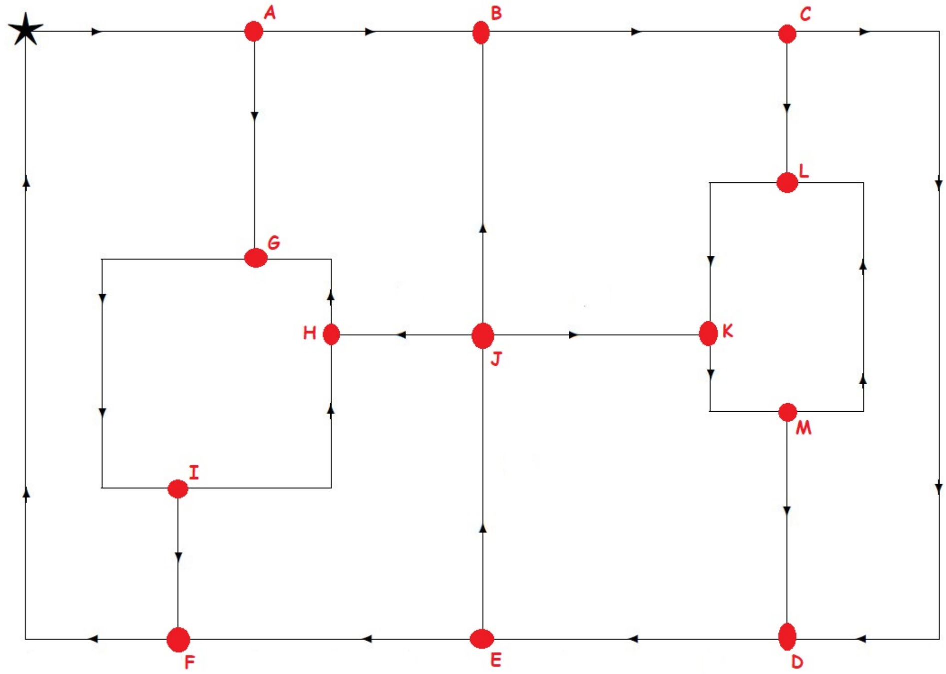

Figura 4: Identificação dos vértices do grafo em questão.

Considerando variáveis de decisão xij que representam o fluxo no arco $(i, j), \forall(i, j) \in A$, isto é, o número de vezes que o arco *(i,j)* é percorrido. Por exemplo, xAB vai corresponder ao número de vezes que o veículo vai percorrer o arco no sentido de A para B.

De forma a cumprir os requisitos necessários para a resolução de um problema de minimização de custo de transporte em rede é necessário formular a função objectivo que vai minimizar o percurso percorrido, esta função vai relacionar o número de vezes que passamos num arco com a sua distância.

Função Objectivo: min : 3 xOA + 3 xAB + 4 xBC + 12 xCD +4 xDE + 4 xEF + 10 xFO + 3 xAG + 6 xGI + 4 xIH + 2 xHG + 2 xJH + 4 xEJ + 4 xJB + 3 xJK + 3 xLK + 2 xKM + 5 xML +2 xCL +3 xMD + 2 xIF ;

Para completar o modelo é necessário definir as restrições que a função objetivo está sujeita, portanto uma das restrições é que fluxo em qualquer arco deve ser, pelo menos, uma unidade, para visitar todos os arcos.; outra restrição é o fluxo que entra num vértice do grafo deve ser igual ao que sai, para o percurso ser fechado, por exemplo, se ele entrar na intersecção 'B', pelo arco 'AB', não pode ficar a meio do caminho, sendo que neste exemplo é obrigado a seguir pelo arco 'BC', para definir esta restrição teríamos de igualar 'xAB' a 'xBC' mas como o tráfego no arco 'BC' não é proveniente apenas de 'AB' mas também de 'JB', esta restrição para o exemplo passa a ser: 'xBC = xAB + xJB'. Repetindo de forma análoga este raciocínio para todas as intersecções obtemos as restrições a seguir apresentadas.

xOA = xFO;
xOA = xAB + xAG;
xBC = xAB + xJB;
xBC = xCD + xCL;
xDE = xCD + xMD;
xDE = xEF + xEJ;
xFO = xEF + xIF;
xGI = xAG + xHG;
xGI = xIF + xIH;
xHG = xJH + xIH;
xKM = xLK + xJK;
xKM = xML + xMD;
xLK = xCL + xML;

xOA>=1;
xAB>=1;
xAG>=1;
xBC>=1;
xCD>=1;
xCL>=1;
xDE>=1;
xEF>=1;
xEJ>=1;
xFO>=1;
xGI>=1;
xHG>=1;
xIF>=1;
xIH>=1;
xJB>=1;
xJH>=1;
xJK>=1;
xKM>=1;
xLK>=1;
xMD>=1;
xML>=1;

3.3 Modelo de programação linear com uma mudança de variável em que os limites inferiores são todos iguais a zero

Nas restrições acima apresentadas, é imposto um limite inferior ao valor das variáveis. O software de resolução de problemas de transportes em rede normalmente apenas aceita arcos com limite superior. Usando a mudança de variável $yij = xij - lij, \forall(i,j) \in A$, em que li j é o limite inferior de fluxo no arco (i,j), pode obter-se uma nova instância em que os limites inferiores são todos iguais a zero. Desta forma com $lij = 1$ obtemos $xij = yij + 1$. De seguida apresentamos o modelo de programação linear da nova instância.

Função Objectivo : min : 3yOA + 3 + 3yAB + 3 + 4 yBC + 4 + 12yCD + 12 + 4yDE + 4 + 4yEF + 4 + 10yFO + 10 + 3 yAG + 3 + 6yGI + 6 + 4yIH + 4 + 2yHG + 2 + 2yJH + 2 + 4yEJ + 4 + 4yJB + 4 + 3yJK + 3 + 3yLK + 3 + 2yKM + 2 + 5yML + 5 + 2yCL + 2 + 3yMD + 3 + 2yIF + 2 ;

yOA + 1 = yFO + 1;
yOA + 1 = yAB + 1 + yAG + 1;
yBC + 1 = yAB + 1 + yJB + 1;
yBC + 1 = yCD + 1 + yCL + 1;
yDE + 1 = yCD + 1 + yMD +1;
yDE + 1 = yEF + 1 + yEJ + 1;
yFO + 1 = yEF + 1 + yIF + 1;
yGI + 1 = yAG + 1 + yHG + 1;
yGI + 1 = yIF + 1 + yIH + 1;
yHG + 1 = yJH + 1 + yIH + 1;
yKM + 1 = yLK + 1 + yJK + 1;
yKM + 1 = yML + 1 + yMD + 1;
yLK + 1 = yCL + 1 + yML + 1;

yOA + 1 >=1;
yAB + 1 >=1;
yAG + 1 >=1;
yBC + 1 >=1;
yCD + 1 >=1;
yCL + 1 >=1;
yDE + 1 >=1;
yEF + 1 >=1;
yEJ + 1 >=1;
yFO + 1 >=1;
yGI + 1>=1;
yHG + 1>=1;
yIF + 1 >=1;
yIH + 1 >=1;
yJB + 1 >=1;
yJH + 1 >=1;
yJK + 1>=1;
yKM + 1>=1;
yLK + 1>=1;
yMD + 1>=1;
yML + 1>=1;

3.4 Rede que resulta da mudança de variável

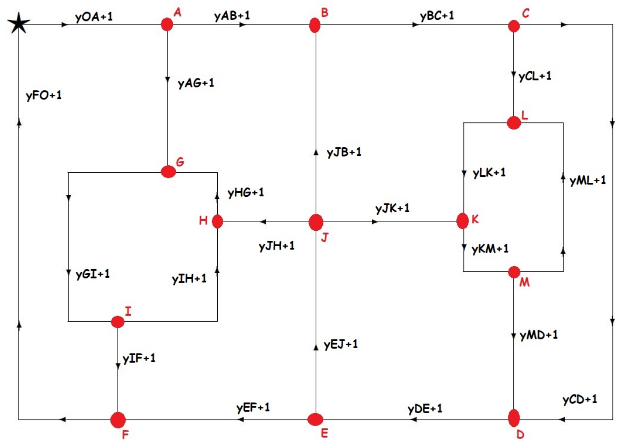

Figura 5: A rede do problema de transporte que resulta da mudança de variável.

A partir da figura acima apresentada podemos concluir quais os valores de oferta (para os vértices de excesso, o número de ruas que entram é maior do que o número de ruas que saem) e de procura (para os vértices de defeito, o número de ruas que entram é menor do que o número de ruas que saem) associados a cada vértice do grafo. Oferta de um vértice é a quantidade de arcos que entram e a procura é a quantidade de arcos que saem.

Vértice	Oferta	Procura	Tipo
O	1	1	—
A	1	2	Defeito
B	2	1	Excesso
C	1	2	Defeito
D	2	1	Excesso
E	1	2	Defeito
F	2	1	Excesso
G	2	1	Excesso
H	2	1	Excesso
I	1	2	Defeito
J	1	3	Defeito
K	2	1	Excesso
L	2	1	Excesso
M	1	2	Defeito

3.5 Ficheiro de Input submetido ao software de optimização em rede

```
13
20
1 2 3 100
1 7 3 100
2 3 4 100
3 4 12 100
3 12 2 100
4 5 4 100
5 6 4 100
5 10 4 100
6 1 13 100
7 9 6 100
8 7 2 100
9 6 2 100
9 8 4 100
10 2 4 100
10 8 2 100
10 11 3 100
11 13 2 100
12 11 3 100
13 4 3 100
13 12 5 100
-1
1
-1
1
-1
1
1
1
-1
-2
1
1
-1
```

Figura 6: Ficheiro de Input - parte 1.

3.6 Ficheiro de output produzido pelo programa

```
\RELAX4 2013>relax4 <parte1.txt >con
 END OF READING
 NUMBER OF NODES = 13, NUMBER OF ARCS = 20
 CONSTRUCT LINKED LISTS FOR THE PROBLEM
 CALLING RELAX4 TO SOLVE THE PROBLEM
 ***********************************
 TOTAL SOLUTION TIME =  0. SECS.
 TIME IN INITIALIZATION =  0. SECS.
   1 2  1.
   2 3  2.
   3 12  1.
   4 5  3.
   5 10  2.
   6 1  2.
   7 9  2.
   8 7  1.
   9 6  1.
   11 13  3.
   12 11  2.
   13 4  2.
 OPTIMAL COST =   93.
 NUMBER OF AUCTION/SHORTEST PATH ITERATIONS = 75
 NUMBER OF ITERATIONS =  23
 NUMBER OF MULTINODE ITERATIONS =  3
 NUMBER OF MULTINODE ASCENT STEPS =  4
 NUMBER OF REGULAR AUGMENTATIONS =  3
 ***********************************
```

Figura 7: Ficheiro de Output - parte 1.

3.7 Solução Óptima

Definindo a dimensão da rede no cabeçalho (13 vértices e 21 arcos), os arcos da rede (onde a letra corresponde a posição do abecedário, por exemplo G é representado como o vértice número 7) e as Ofertas e as Procuras em cada vértice da Rede num ficheiro .txt apresentado na figura 6.

Com o auxílio do software Relax4 introduzindo o ficheiro criado foi possível obter o caminho mais curto que passe por todos os vértices, no output deste ficheiro é exibido quantas vezes deve passar-se num arco de forma a atingir o valor óptimo que é 93. Por exemplo, a linha no output "1 2 1"diz-nos que do vértice 1 para o 2 passa uma vez.

É de realçar que nem todos os arcos são apresentados pois efectuou-se uma mudança de variável anteriormente, ou seja, os arcos são percorridos mais uma vez do que aquelas que são apresentadas no output, sendo assim o custo total 93+85 (custo de uma passagem em todos os arcos) ou seja 178.

De forma a descobrir o circuito em que todos os arcos são percorridos, partindo do vértice A(1), definimos de forma manual esse circuito. A obtenção da solução óptima foi desenvolvida em duas fases a primeira, figura 8, onde é representado o output obtido do Relax4 no grafo. Posteriormente, adicionamos uma passagem por todos os arcos de forma a ligar o circuito apresentado na figura anterior onde consta o circuito que liga todos os vértices, cumprindo os sentidos estipulados e passando o número de vezes determinado pelo software, figura 9.

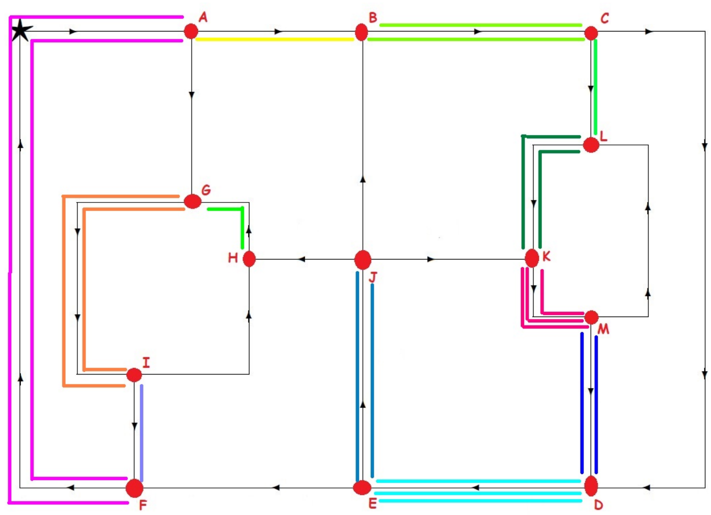

Figura 8: Representação do output no grafo - parte 1.

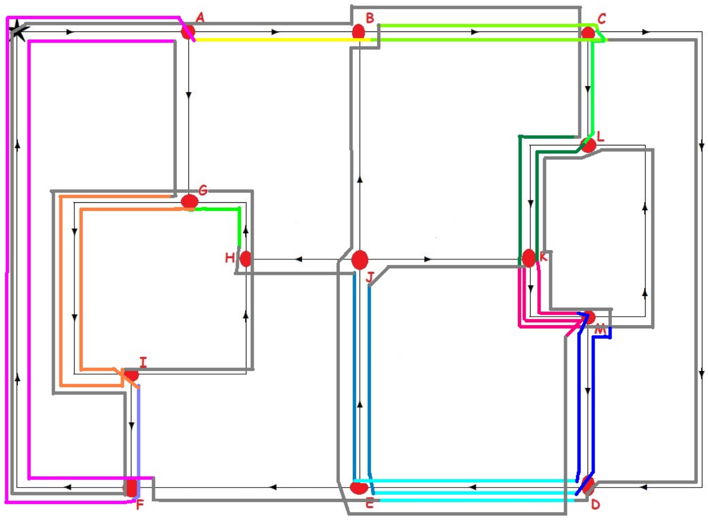

Figura 9: Representação do circuito - parte 1.

3.8 Validação do modelo

Para validar o nosso percurso, tivemos em conta os seguintes aspectos:

1. **Percurso entre vértices**: Provámos que existe um caminho entre todos os vértices, para isso, bastou provar que existe uma solução pois ao termos uma solução ela passa por todos os vértices, desta forma existe um trajecto de um qualquer vértice para outro qualquer vértice;

2. **Todos os arcos foram percorridos**: No percurso obtido todos os arcos foram percorridos pois ao fazermos a mudança de variável de yij = xij - 1 garantimos automaticamente que todos os vértices são percorridos pelo menos uma vez;

3. **Assegurar que o veículo terminou o percurso no ponto de partida**: O percurso do veículo começou na origem (representada por uma estrela na figura 1) e foi necessário confirmar que este também acabou no mesmo ponto. A solução que apresentamos tinha início no depósito e terminava no mesmo, isto deve-se ao resultado obtido ser um circuito, ou seja, qualquer que seja o ponto de partida que se escolha ele acabará no mesmo sítio, a solução apresentada é apenas a interpretação que começa na origem, o que nos é conveniente para o nosso problema.

4 Parte II

Considerando as variáveis de decisão xij, cada uma correspondendo a um caminho entre um vértice de excesso i e um vértice de defeito j.

4.1 Grafo bipartido deste problema de transporte

	A	C	E	I	J	M	
B	35	4	18	32	22	11	1
D	21	16	4	18	8	13	1
F	13	20	34	22	38	27	1
G	21	28	42	6	46	35	1
H	23	30	44	8	48	37	1
K	26	21	9	23	13	2	1
L	29	24	12	26	16	5	1
	1	1	1	1	2	1	

Do lado esquerdo da tabela são apresentados os vértices de excesso (linhas) e em cima são apresentados os vértices por defeito (colunas). Ou seja, uma célula da tabela representa o caminho mais curto do vértice de excesso dessa linha até ao vértice de defeito dessa coluna.

Os valores do caminho mais curtos entre os vértices foi obtido por inspecção, isto é, olhamos para o grafo e calculamos as distâncias entre os diferentes vértices.

O problema é balanceado, isto é, o número total de caminhos que saem dos vértices de excesso (identificados à esquerda da tabela) é igual ao número de caminhos que chegam aos vértices de defeito (identificados em cima da tabela).

4.2 Ficheiro de input submetido ao software de optimização em rede

```
13
42
2 1 35 1000
2 3 4 1000
2 5 18 1000
2 9 32 1000
2 10 22 1000
2 13 11 1000
4 1 21 1000
4 3 16 1000
4 5 4 1000
4 9 18 1000
4 10 8 1000
4 13 13 1000
6 1 13 1000
6 3 20 1000
6 5 34 1000
6 9 22 1000
6 10 38 1000
6 13 27 1000
7 1 21 1000
7 3 28 1000
7 5 42 1000
7 9 6 1000
7 10 46 1000
7 13 35 1000
8 1 23 1000
8 3 30 1000
8 5 44 1000
8 9 8 1000
8 10 48 1000
8 13 37 1000
11 1 26 1000
```

```
11 3 21 1000
11 5 9 1000
11 9 23 1000
11 10 13 1000
11 13 2 1000
12 1 29 1000
12 3 24 1000
12 5 12 1000
12 9 26 1000
12 10 16 1000
12 13 5 1000
-1
1
-1
1
-1
1
1
1
-1
-2
1
```

Figura 10: Ficheiro de Input - parte 2.

4.3 Ficheiro de output produzido pelo programa

```
>relax4 <parte2.txt >con
 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 10  1.
   6 5  1.
   7 9  1.
   8 1  1.
   11 13  1.
   12 10  1.
 OPTIMAL COST =    93.
 NUMBER OF AUCTION/SHORTEST PATH ITERATIONS = 26
 NUMBER OF ITERATIONS =  18
 NUMBER OF MULTINODE ITERATIONS =  1
 NUMBER OF MULTINODE ASCENT STEPS =  1
 NUMBER OF REGULAR AUGMENTATIONS =  1
 ***********************************
```

Figura 11: Ficheiro de Output - parte 2.

4.4 Solução óptima

Definindo a dimensão da rede no cabeçalho (13 vértices e 42 arcos), os arcos da rede (onde a letra corresponde a posição do abecedário, por exemplo G é representado como o vértice número 7) e as Ofertas e as Procuras em cada vértice da Rede num ficheiro .txt apresentado na figura 10.

Com o auxílio do software Relax4 introduzindo o ficheiro criado foi possível obter o caminho mais curto que passe por todos os vértices, no output deste ficheiro é exibido quantas vezes a mais deve passar-se num arco de forma a atingir o valor óptimo apresentado 93. Por exemplo a linha no output "4 10 1"diz-nos que do vértice de excesso 4(D) para o vértice de defeito 10 (J) passa uma vez no caminho mais curto definido acima.

É de realçar que nem todos os arcos são apresentados pois efectuou-se uma mudança de variável anteriormente, ou seja, os arcos são percorridos mais uma vez do que aquelas que são apresentadas no output, sendo assim o custo total 93+85 (custo de uma passagem em todos os arcos), ou seja, 178.

De forma a descobrir o circuito em que todos os arcos são percorridos, partindo do vértice A(1), definimos de forma manual esse circuito. A obtenção da solução óptima foi desenvolvida em em duas fases a primeira, figura 12, onde é representado o output obtido do Relax4 no grafo. Posteriormente, adicionamos uma passagem por todos os arcos de forma a ligar o circuito apresentado na figura anterior onde consta o circuito que liga todos os vértices, cumprindo os sentidos estipulados e passando o número de vezes determinado pelo software, figura 13.

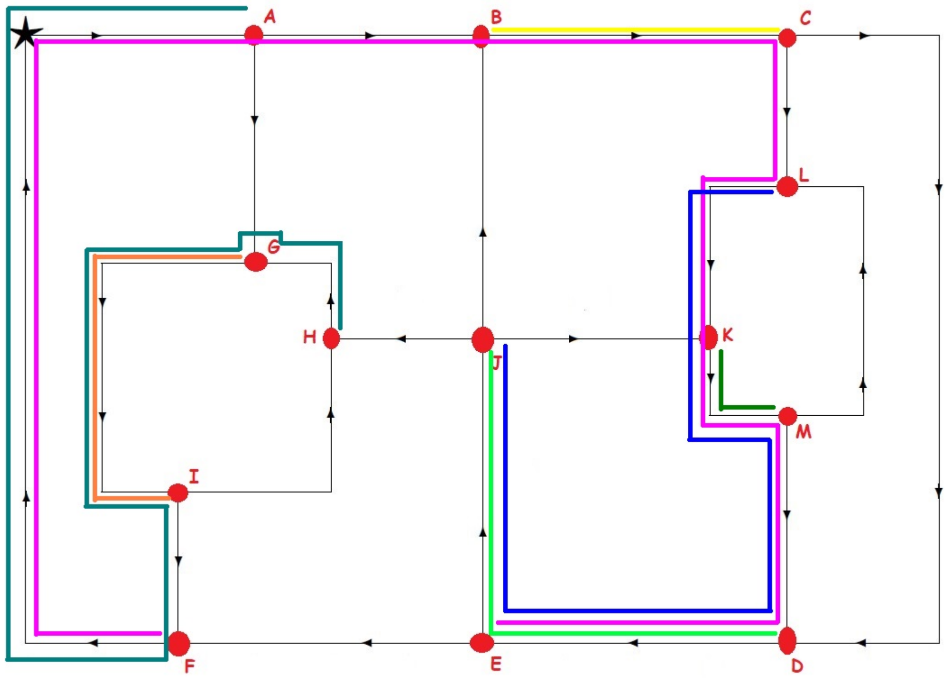

Figura 12: Representação do output no grafo - parte 2.

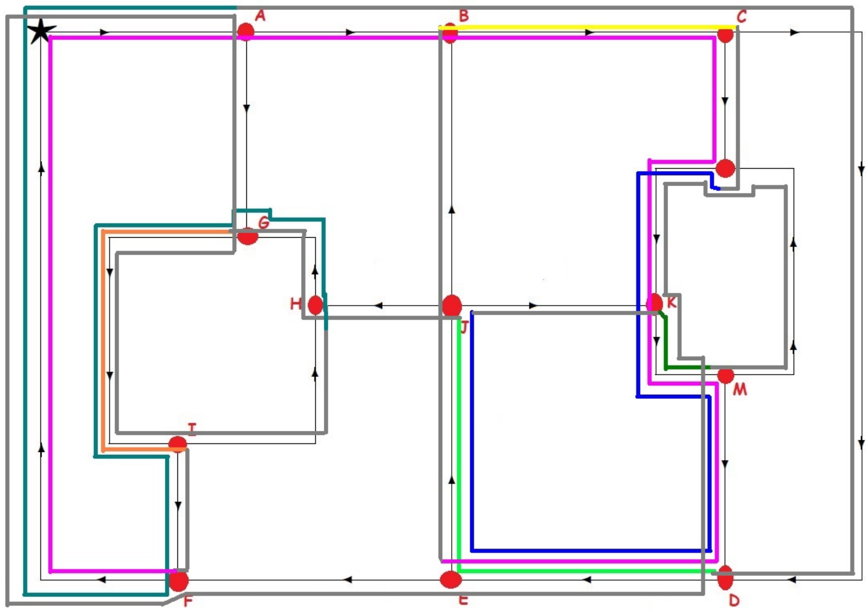

Figura 13: Representação do circuito - parte 2

4.5 Validação do modelo

Para validar o nosso percurso, tivemos em conta os seguintes aspectos:

1. **Percurso entre vértices**: Provámos que existe um caminho entre todos os vértices, para isso, bastou provar que existe uma solução pois ao termos uma solução ela passa por todos os vértices desta forma existe um trajecto de um qualquer vértice para o outro qualquer vértice;

2. **Todos os arcos foram percorridos**: No percurso obtido todos os arcos foram percorridos pois ao fazermos a mudança de variável de yij = xij - 1 garantimos automaticamente que todos os vértices são percorridos pelo menos uma vez;

3. **Assegurar que o veículo terminou o percurso no ponto de partida**: O percurso do veículo começou na origem (representada por uma estrela na figura 1) e foi necessário confirmar que este também acabou no mesmo ponto. A solução que apresentamos tinha início no depósito e terminava no mesmo, isto deve-se ao resultado obtido ser um circuito, ou seja qualquer que seja o ponto de partida que se escolha ele acabará no mesmo sítio, a solução apresentada é apenas a interpretação que começa na origem, o que nos é conveniente para o nosso problema.

5 Conclusão

O presente relatório descreveu, de forma sucinta, a resolução do problema proposto utilizando um Modelo de Programação Linear com diferentes abordagens.

Consideramos que os principais objectivos foram compridos, no entanto, há que ter conta que os caminhos apresentados como soluções não são únicos pois as escolhas feitas de ligação entre os arcos afecta os caminhos na forma geral, gerando um caminho novo, apesar de não ser diferente quanto ao número de vezes passadas num determinado arco, o que leva a um custo igual.

O valor obtido de menor custo foi idêntico nas duas partes o que equivale ao valor obtido na solução do primeiro trabalho apresentado onde o problema é o mesmo, o que demonstra consistência.

Sentimos que a realização deste projecto consolidou os nossos conhecimentos de resolução de problemas através do Modelo de Programação Linear, nomeadamente ajudou-nos a desenvolver a nossa capacidade de analisar sistemas complexos, como também ajudou no processo de criação de modelos de sistemas complexos. Propiciou-nos a experiência de utilizar um programa computacional adequado para obter soluções dos modelos que definimos consequentemente fortaleceu a nossa capacidade de validação, análise e interpretação das soluções obtidas.