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

Modelos Determinísticos de Investigação Operacional

Relatório sobre o Trabalho 1

Grupos 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

16 de Outubro de 2019

Conteúdo

1 Introdução **2**

2 Problema **2**

3 Sentido das ruas BCDE **4**

4 Formulação do problema como um Modelo de Programação Linear **4**

5 Ficheiro de input **7**

6 Ficheiro de output **8**

7 Solução ótima **9**

8 Validação do modelo **10**

9 Conclusão **11**

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 objetivo 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 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 que 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/trajeto 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 expor que existe solução temos que descobrir um caminho de qualquer vértice para qualquer outro vértice do grafo.

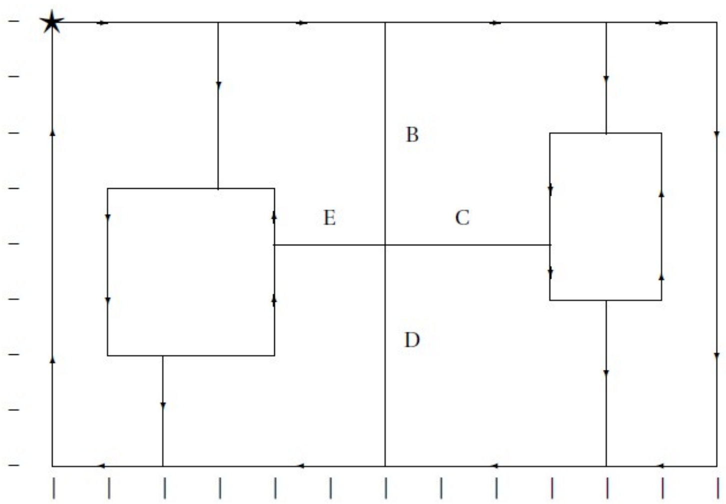


Figura 1: Grafo que se pretende determinar o circuito em que todos os seus arcos são percorridos.

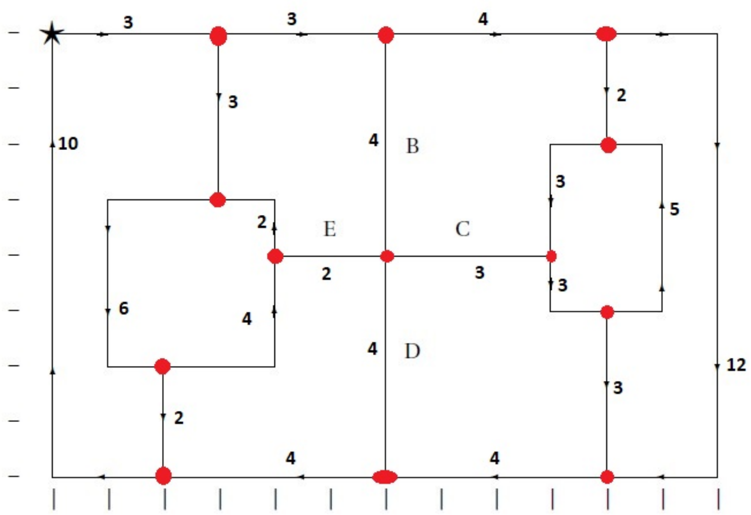


Figura 2: A distância associada a cada arco.

3 Sentido das ruas BCDE

Conforme a figura 1 o sentido dos arcos já está definido, exceto 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.

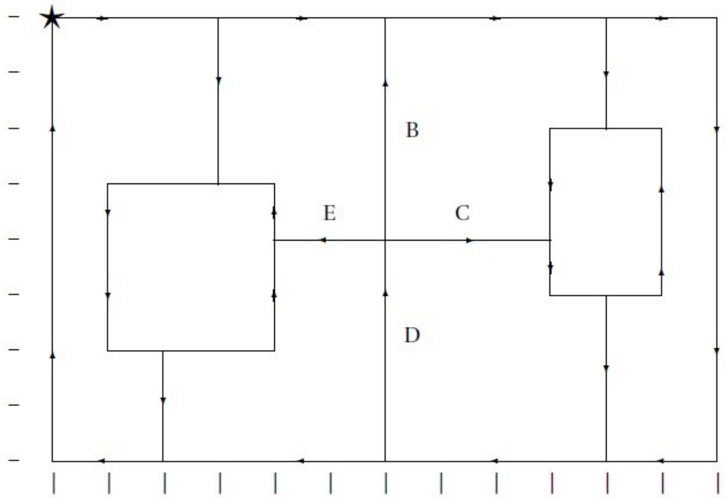


Figura 3: Definição dos sentidos dos arcos BCDE do grafo apresentado anteriormente.

4 Formulação do problema como um Modelo de Programação Linear

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 ao Modelo de Programação Linear transmitido e sugerido para a realização do problema, pois este modelo é usado para resolver problemas que exijam otimização. Nos problemas de Programação Linear temos que definir uma função objetivo e as suas respetivas restrições, onde estas são lineares. De modo a facilitar a compreensão do problema, decidimos a cada interseção de arcos atribuir uma letra, sendo que estas variam entre A' e 'M', cada arco identificado pela junção de duas interseções, por exemplo, o arco entre as interseções '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.

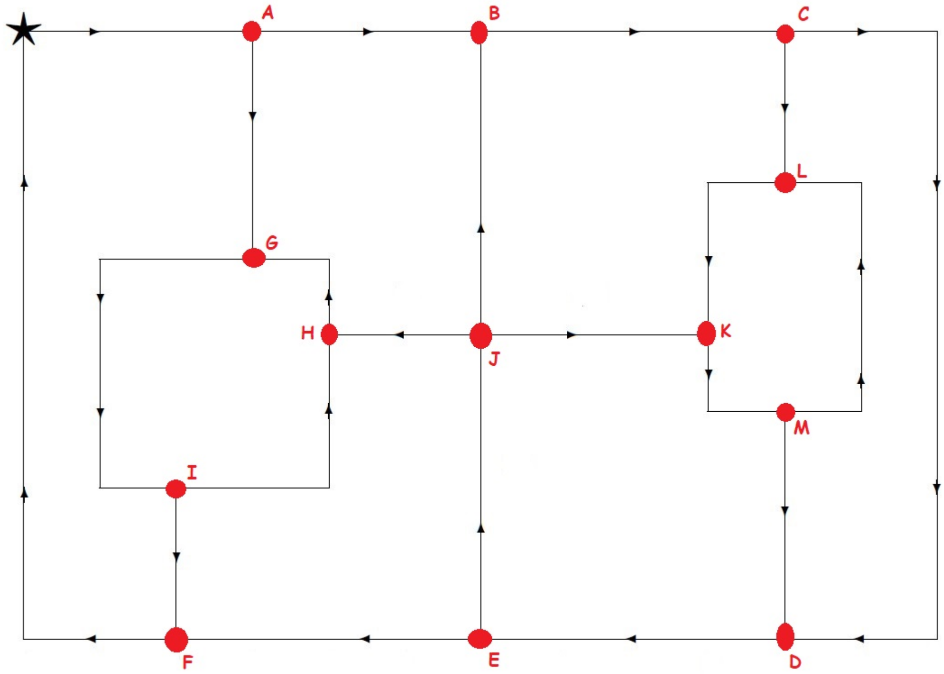


Figura 4: Identificação dos vértices do grafo em questão.

Como foi apresentado anteriormente é exigido que se passe pelo menos uma vez em cada arco, desta forma temos o poder de decidir quantas vezes se pode passar em cada arco. Assim a incógnita que associamos às nossas decisões admissíveis denominada variável de decisão vai equivaler ao número de vezes que um determinado arco 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 Programação Linear é necessário formular a função objetivo que vai minimizar o percurso percorrido, esta função vai relacionar o número de vezes que passamos num arco com a sua distância.

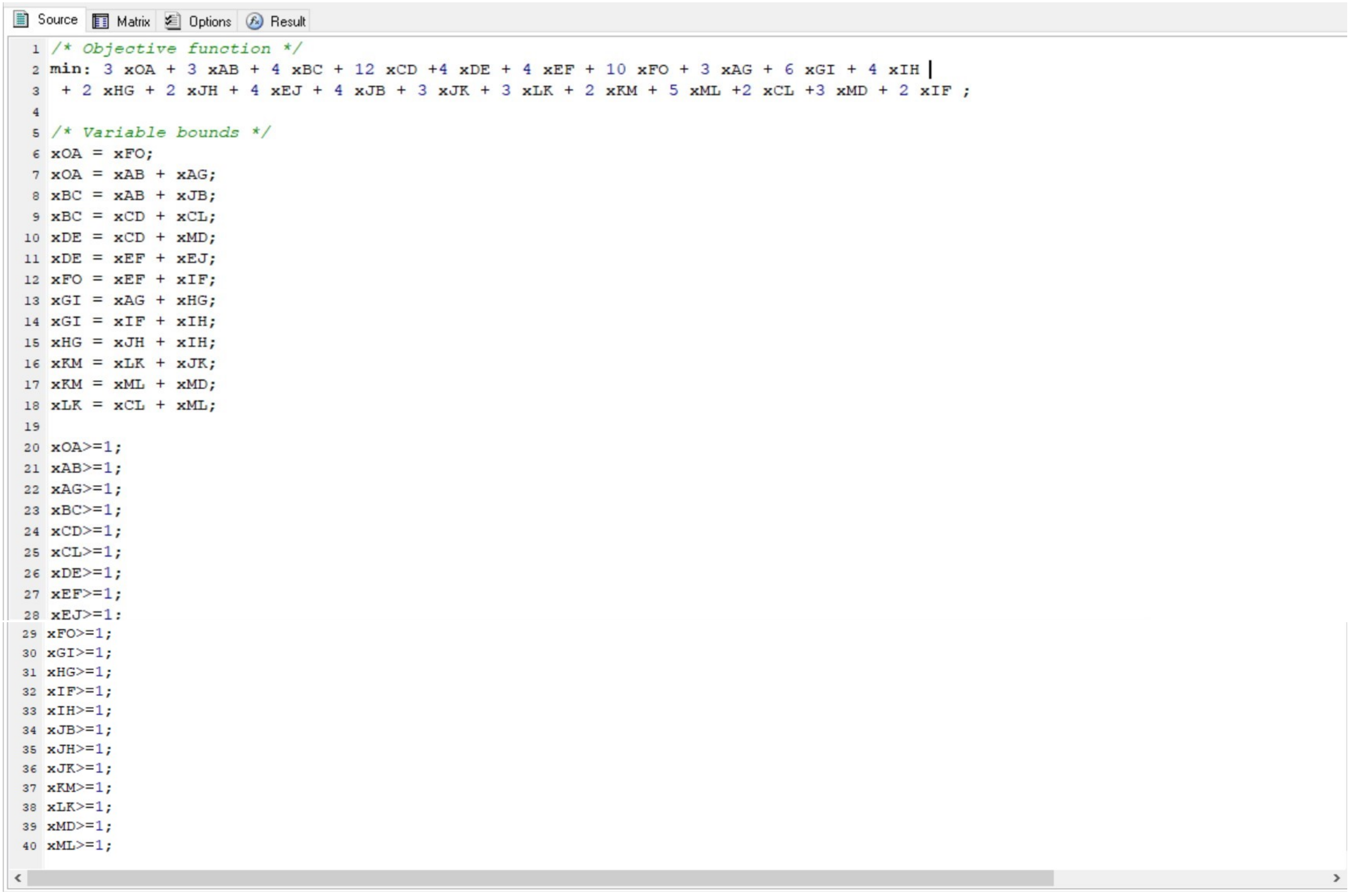
Função Objetivo : min : 3 xOA + 3 xAB + 4 xBC + 12 xCD +4 xDE + 4 xEF + 10 xFO + 3 xAG + 6 xGI + 4 xIH + 2 xHG + 2 xJH + 4 xEJ + 4 xJB + 3 xJK + 3 xLK + 2 xKM + 5 xML +2 xCL +3 xMD + 2 xIF ;

Para completar o modelo é necessário definir as restrições que a função objetivo está sujeita, portanto uma das restrições é que um arco tem de ser percorrido uma vez, ou seja, todas as nossas variáveis de decisão que representam o número de vezes que passou num arco devem ser maiores ou iguais a 1 de forma a garantir a passagem; outra restrição é o veículo não poder ficar a meio do percurso, por exemplo, se ele entrar na interseção '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 interseções 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;

5 Ficheiro de input



Source Matrix Options Result

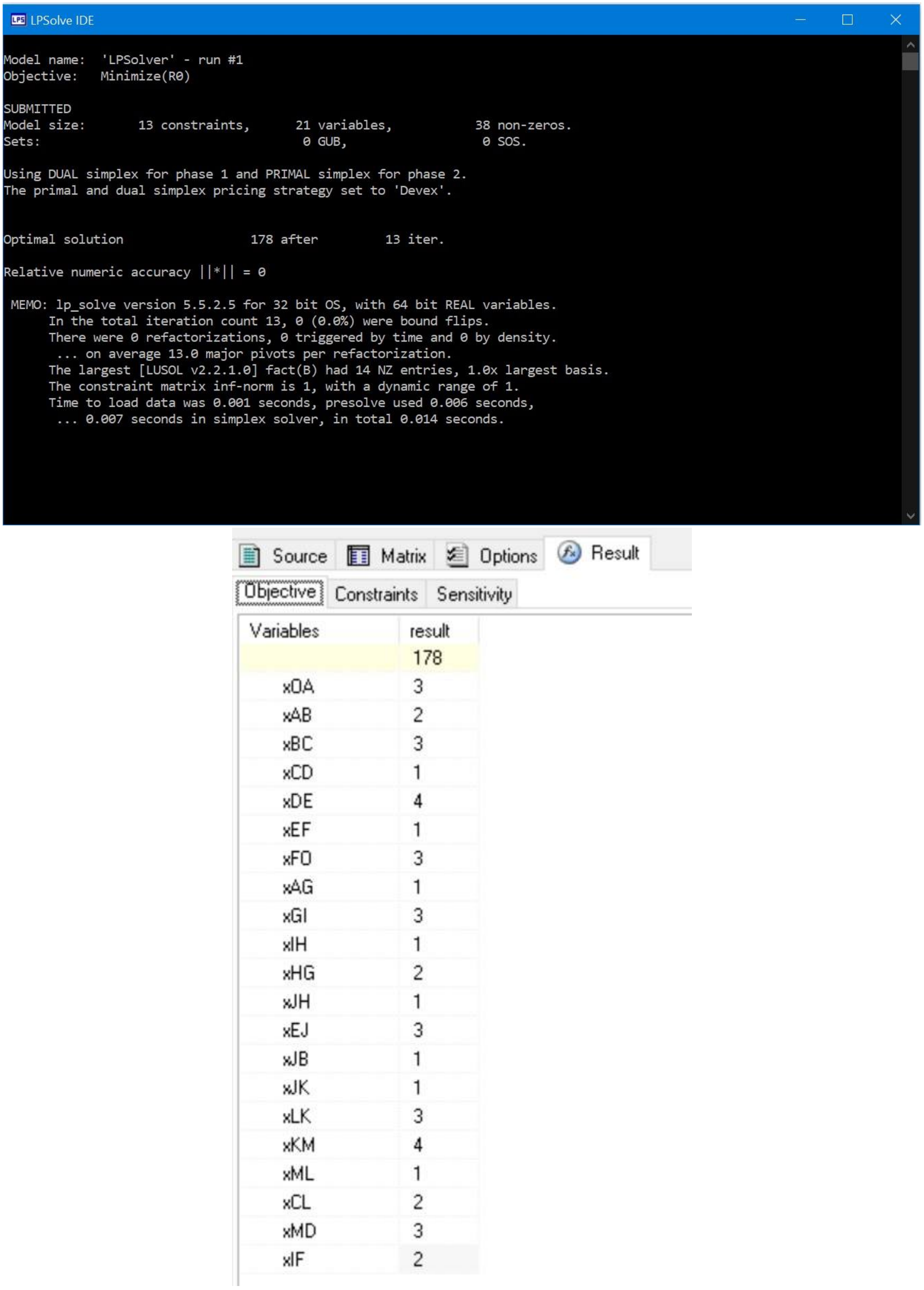
```
/* Objective function */
min: 3 xOA + 3 xAB + 4 xBC + 12 xCD +4 xDE + 4 xEF + 10 xFO + 3 xAG + 6 xGI + 4 xIH
 + 2 xHG + 2 xJH + 4 xEJ + 4 xJB + 3 xJK + 3 xLK + 2 xKM + 5 xML +2 xCL +3 xMD + 2 xIF ;

/* Variable bounds */
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;
```

Figura 5: Função objetivo e respetivas restrições.

6 Ficheiro de output



LPSolve IDE

```
Model name:  'LPSolver' - run #1
Objective:   Minimize(R0)

SUBMITTED
Model size:       13 constraints,      21 variables,           38 non-zeros.
Sets:                                   0 GUB,                   0 SOS.

Using DUAL simplex for phase 1 and PRIMAL simplex for phase 2.
The primal and dual simplex pricing strategy set to 'Devex'.


Optimal solution                 178 after         13 iter.

Relative numeric accuracy ||*|| = 0

 MEMO: lp_solve version 5.5.2.5 for 32 bit OS, with 64 bit REAL variables.
      In the total iteration count 13, 0 (0.0%) were bound flips.
      There were 0 refactorizations, 0 triggered by time and 0 by density.
       ... on average 13.0 major pivots per refactorization.
      The largest [LUSOL v2.2.1.0] fact(B) had 14 NZ entries, 1.0x largest basis.
      The constraint matrix inf-norm is 1, with a dynamic range of 1.
      Time to load data was 0.001 seconds, presolve used 0.006 seconds,
       ... 0.007 seconds in simplex solver, in total 0.014 seconds.
```

Source Matrix Options Result

Objective Constraints Sensitivity

Variables	result
	178
xOA	3
xAB	2
xBC	3
xCD	1
xDE	4
xEF	1
xFO	3
xAG	1
xGI	3
xIH	1
xHG	2
xJH	1
xEJ	3
xJB	1
xJK	1
xLK	3
xKM	4
xML	1
xCL	2
xMD	3
xIF	2

Figura 6: Resultado obtido.

7 Solução ótima

Com auxílio do software LPSolve IDE introduzimos a nossa função objetivo e as restrições definidas previamente, figura 4, e obtivemos a tabela apresentada na secção anterior, figura 5, onde são exibidas as variáveis de decisão definidas anteriormente com um valor associado, valor esse que representa como foi dito anteriormente o número de vezes que um determinado arco vai ser percorrido. Nesta tabela também foi determinada a distância total percorrida, assim obtivemos 178 como o custo mais reduzido de forma a percorrer todos os arcos, e cumprir os sentidos impostos, este resultado equivale à solução ótima.

De forma a descobrir o circuito em que todos os arcos são percorridos, partindo da origem representada por uma estrela na figura 1, definimos de forma manual esse circuito. De seguida apresentamos a figura 6 onde consta o circuito que liga todos os vértices, cumprindo os sentidos estipulados e passando o número de vezes determinado pelo software. Dividimos o circuito em diferentes cores conforme passe na origem e no caso explicito da cor verde como se pode observar efetua várias voltas ao grafo de forma a facilitar a visualização encontra-se apresentado por vários tons.

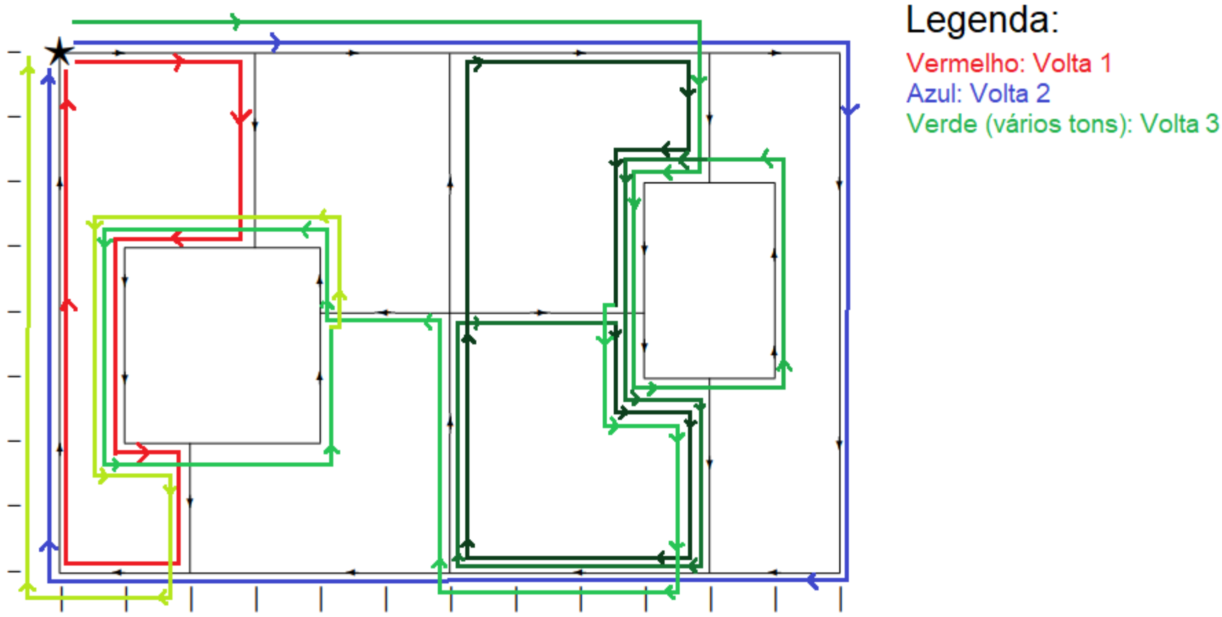


Figura 7: O caminho definido.

Apresentamos por exemplo o processo/raciocínio inicial neste caso o caminho azul, partindo obrigatoriamente da origem só temos a possibilidade de percorrer o arco ‘OA’, cujo valor no software obtido é 3, ou seja, vai ser percorrido 3 vezes desta forma como já se percorreu uma vez ainda falta passar 2 vezes, agora temos duas possibilidades de seguir pelo arco ‘AB’ que vai ser percorrido 2 vezes ou o arco ‘AG’ que é percorrido uma, decidimos seguir pelo arco ‘AB’ dado ser para já ser indiferente ir em frente ou para baixo, dado que o arco que não vai ser percorrido de momento, irá sê-lo mais à frente. Percorrido este arco ficando a faltar passar ainda mais uma vez lá, de momento somos obrigados a seguir pelo arco ‘BC’ porque não existe outra alternativa, ficando a faltar mais duas vezes obrigatoriamente. Encontramos de momento no vértice C onde podemos voltar a escolher para onde pretendemos ir ou pelo arco ‘CD’, que vai ser percorrido apenas 1 vez, ou pelo arco ‘CL’ que vai ser percorrido 2 vezes. Decidimos optar por percorrer ‘CD’ por ser indiferente. Posteriormente, somos obrigados a seguir o arco ‘DE’ que vai ser percorrido 4 vezes, com esta vez fica a faltar 3 vezes. Agora no vértice E podemos subir utilizando o arco ‘EJ’ cujo irá ser percorrido 3 vezes ou o arco ‘EF’ que é apenas percorrido uma vez, decidimos por optar por este pelos mesmos motivos anteriormente referidos, ficando restringidos de nunca mais passar por lá. Em ‘F’ somos obrigados a regressar à origem através do arco ‘FO’, ficando a faltar ainda mais duas vezes dado o valor obtido ter sido 3. Como ainda existem arcos que ainda não foram percorridos e arcos que ainda não foram percorridos o número de vezes que o software indicou, é necessário continuar o processo de decrementação aos valores obtidos. Repetindo o processo analogamente chegamos a um momento em que todas as variáveis foram decrementadas sendo iguais a zero, ou seja, foi possível descobrir um percurso que percorreu todos os arcos.

8 Validação do modelo

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

1. **Interseções inválidas**: Asseguramos que nenhuma das interseções teria todos arcos a entrar nela, se fosse este o caso, o veículo chegava à interseção e não conseguia sair de lá;

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

3. **Todos os arcos foram percorridos**: No percurso obtido todos os arcos foram percorridos porque a solução que obtivemos no software indicou-nos que nenhuma das variáveis de decisão ficou a zero, isto é, todos os caminhos foram percorridos, o que se refletiu com a solução ótima;

4. **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. Para cada interseção existe uma restrição que garante que o número de vezes que entre numa interseção é o mesmo número que sai, o que implica que o veículo termine o percurso na interseção onde ele começou, para isso bastou escolher como ponto de partida o depósito, como era pedido.

9 Conclusão

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

Após a realização deste trabalho, ficamos conscientes das potencialidades que a Programação Linear para resolução de problemas das mais diversas áreas nomeadamente: economia, gestão, física e engenharia otimizando as soluções obtidas.

Consideramos que os principais objetivos foram compridos, no entanto, há que ter conta que o caminho apresentado como solução não é único pois as escolhas feitas de percorrer um arco primeiro que outro afeta o caminho 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.

Sentimos que a realização deste projeto 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.