Universidade do Minho
Licenciatura em Engenharia Informática

Investigação Operacional

Trabalho Prático 2
Escalonamento de Equipas




Lídia Sousa
a93205

Diogo Rebelo
a93278

Bohdan Malanka
a93300

Henrique Alvelos
a93316

20 de julho de 2022

Conteúdo

Lista de Figuras

1 Introdução

O presente relatório incorpora o conjunto de itens primordiais do desenvolvimento do segundo trabalho prático, no âmbito da Unidade Curricular de Investigação Operacional. O respetivo enunciado propõe a formulação de um modelo de fluxos em rede que permita responder a um problema que envolve o escalonamento de equipas, com tempos de serviço fixos.

De um modo mais detalhado, trata-se de um problema de transportes, que consiste em minimizar o custo total de uma operação que envolve vários custos, sejam eles de deslocação ou de utilização de veículos. Analogamente, o trabalho trata a situação em que se pretende atribuir um determinado serviço a clientes que são distribuídos geograficamente a equipas, com o objetivo de minimizar o custo de toda a operação requirida. É, então, bem explícito o objetivo principal: determinar o número de equipas a utilizar na prestação destes serviços, de modo a que os serviços se realizem a todos os clientes, atempadamente, ou seja, cumprindo o horário de trabalho da sede e as horas de início do serviço para cada cliente e tendo sempre em conta a minimização dos custos gerais associados às deslocações (que incluem portagens, despesas de combustível, etc.). Pretende-se efetuar deslocações da sede para os clientes, entre clientes e destes para a sede (no final do serviço).

O problema em questão socorre-se das informações que constam na tabela que surge de seguida, com cada cliente e as horas a que o serviço deve ser prestado. Cada parâmetro surge explicado na secção de Descrição do Problema, servindo, aqui, apenas de informação introdutória. Imediatamente a seguir, são apresentadas as matrizes com os tempos e custos de deslocação entre clientes e entre estes e a sede em Keleirós. Note-se que cada cliente é representado pela inicial do seu nome nas matrizes abaixo e que a cada um está associado um número que o identifica.

Para a resolução do problema supracitado, socorre-se à utilização do software de programação linear Relax4, muito conhecido neste âmbito de programação com recurso à otimização de redes.

j	cliente	a_j (¼hora)	a_j (hora do serviço)
1	Ana	a_1	depende de ABCDE
2	Beatriz	7	10:45
3	Carlos	4	10:00
4	Diogo	2	09:30
5	Eduardo	10	11:30
6	Francisca	6	10:30
7	Gonçalo	9	11:15
8	Helena	a_8	depende de ABCDE
9	Inês	2	09:30
10	José	5	10:15

Figura 1: Informação alusiva ao serviço a prestar a cada cliente.

	B	C	D	E	F	G	H	I	J	K
A	4	1	2	2	3	2	1	0	3	1
B		3	5	3	3	2	3	4	2	5
C			3	2	3	2	0	1	1	2
D				1	3	3	3	2	3	1
E					2	1	2	2	2	2
F						2	3	3	3	4
G							2	2	2	3
H								1	1	1
I									3	2
J										4

tempos de deslocação

	B	C	D	E	F	G	H	I	J	K
A	13	5	6	5	10	7	5	0	7	1
B		11	14	10	8	6	11	13	4	15
C			8	6	10	6	0	5	6	2
D				4	8	8	8	6	11	4
E					6	4	6	5	7	6
F						5	10	10	8	11
G							10	7	5	9
H								5	6	9
I									7	9
J										10

custos de deslocação

Figura 2: Informação alusiva ao custos e tempos de deslocação.

2 Questão 0

Remoção de clientes e tempos de deslocação dependentes de BCDE

Nesta questão, pretendia-se proceder à estratégia de remoção de clientes a servir e dos respetivos tempos de deslocação, de acordo com as indicações que se apresentam de seguida:

Seja ABCDE o número de inscrição do estudante do grupo com maior número de inscrição.

- fazer a1 = B + 1;
- fazer a8 = C + 1;
- se D par, remover o cliente D;
- se E par, remover o cliente E;

Tendo em conta a observação inicial e que os números mecanográficos dos estudantes têm a seguinte relação: $93205 < 93278 < 93300 < 93316$, ABCDE corresponde ao número 93316. É assim possível calcular o valor de a_1 e de a_8:

- a1 = 3 + 1 = 4 (quatro quartos de hora - hora inicial do serviço: 10h00);
- a8 = 3 + 1 = 4 (quatro quartos de hora - hora inicial do serviço: 10h00);

Como o único dígito par é o dígito E, o grupo tem de remover apenas o cliente com j = E (j = 6) do conjunto de clientes inicial. Efetuando as alterações requisitadas, a tabela (com os respetivos serviços a prestar) e as matrizes (com os tempos e custos de deslocação), após remoção de clientes, são:

j	cliente	a_j (¼hora)	a_j (hora do serviço)
1	Ana	4	10:00
2	Beatriz	7	10:45
3	Carlos	4	10:00
4	Diogo	2	09:30
5	Eduardo	10	11:30
7	Gonçalo	9	11:15
8	Helena	4	10:00
9	Inês	2	09:30
10	José	5	10:15

Figura 3: Informação alusiva ao serviço a prestar a cada cliente, após remoção de clientes.

	B	C	D	E	G	H	I	J	K
A	4	1	2	2	2	1	0	3	1
B		3	5	3	2	3	4	2	5
C			3	2	2	0	1	1	2
D				1	3	3	2	3	1
E					1	2	2	2	2
G						2	2	2	3
H							1	1	1
I								3	2
J									4

tempos de deslocação

	B	C	D	E	G	H	I	J	K
A	13	5	6	5	7	5	0	7	1
B		11	14	10	6	11	13	4	15
C			8	6	6	0	5	6	2
D				4	8	8	6	11	4
E					4	6	5	7	6
G						10	7	5	9
H							5	6	9
I								7	9
J									10

custos de deslocação

Figura 4: Informação alusiva ao custos e tempos de deslocação, após remoção de clientes.

Grafo de Compatibilidades Resultante

De modo a obter o grafo de compatibilidades resultante, foi necessária a realização de alguns cálculos, para que fosse possível saber se, saindo da sede, passando por um certo cliente, era possível chegar a horas ao próximo cliente, e assim sucessivamente.

Então, tendo em conta a condição estabelecida no enunciado, verificamos com que clientes (ou com o ponto de origem, se aplicável) seria possível chegar a cada cliente, observando se o instante em que uma equipa sai de um cliente (hora do serviço ou, no caso do vértice K, a hora de início) mais o tempo de deslocação seria menor ou igual à hora de início do próximo serviço de destino (próximo cliente).

Na tabela abaixo, para cada par de vértices verificamos se a propriedade $a_i + t_{ij} \leq a_j$ se cumpria, determinando os emparelhamentos a estabelecer no grafo de compatibilidades.

Pares com K	**Incluir (S/N)**	**Pares com A**	**Incluir (S/N)**	**Pares com H**	**Incluir (S/N)**
KD	0+1 <= 2 (S)	AJ	**4+3 <= 5 (N)**	HJ	4+1 <= 5 (S)
KI	0+2 <= 2 (S)	AB	**4+4 <= 7 (N)**	HB	4+3 <= 7 (S)
KA	0+1 <= 4 (S)	AG	4+2 <= 9 (S)	HG	4+2 <= 9 (S)
KC	0+2 <= 4 (S)	AE	4+2 <= 10 (S)	HE	4+2 <= 10 (S)
KH	0+1 <= 4 (S)	AC	**4+1 <=4 (N)**	HC	4+0 <= 4 (S)
KJ	0+4 <= 5 (S)	AD	**4+2 <= 2 (N)**		
KB	0+5 <= 7 (S)	AH	**4+1 <= 4 (N)**		
KG	0+3 <= 9 (S)	AI	**4+0 <= 2 (N)**		
KE	0+2 <= 10 (S)				
Pares com D	**Incluir (S/N)**	**Pares com C**	**Incluir (S/N)**	**Pares com B**	**Incluir (S/N)**
DA	2+2 <= 4 (S)	CJ	4+1 <= 5 (S)	BG	7+2 <= 9 (S)
DC	**2+3 <= 4 (N)**	CB	4+3 <= 7 (S)	BE	7+3 <= 10 (S)
DH	**2+3 <= 4 (N)**	CG	4+2 <= 9 (S)	GE	9+1 <= 10 (S)
DJ	2+3 <= 5 (S)	CE	4+2 <= 10 (S)		
DB	2+5 <= 7 (S)	CH	4+0 <= 4 (S)		
DG	2+3 <= 9 (S)	CA	**4+5 <= 4 (N)**		
DE	2+1 <= 10 (S)				
DI	**2+2 <= 2 (N)**				
Pares com I	**Incluir (S/N)**	**Pares com J**	**Incluir (S/N)**	**Pares com E**	**Incluir (S/N)**
IA	2+0 <= 4 (S)	JB	5+2 <= 7 (S)	É já o último cliente	
IC	2+1 <= 4 (S)	JG	5+2 <= 9 (S)		
IH	2+1 <= 4 (S)	JE	5+2 <= 10 (S)		
IJ	2+3 <= 5 (S)				
IB	2+4 <= 7 (S)				
IG	2+2 <= 9 (S)				
IE	2+2 <= 10 (S)				
ID	**2+2 <= 2 (N)**				

Tabela 1: Tabela com os emparelhamentos e respetivos custos.

Então, no grafo de compatibilidades não constarão os arcos [DC], [DH], [DI], [AJ], [AB], [AC], [AD], [AH] e [AI] e também é sabido que só existem emparelhamentos para nodos em que o início do serviço é igual ou posterior. Para representar a sede, usa-se Ki e Kf. Os arcos para o nodo Kf também não estão representados na tabela, já que servem apenas para indicar o regresso à sede. Reparamos, também, que o arco [CH] e [HC] seriam emparelhamentos possíveis, já que o tempo de inicio de serviço de ambos é igual e o tempo de deslocação entre esses vértices é de 0, então, tecnicamente, quando visitamos um deles, podemos considerar que visitamos o outro também. Para evitar prblemas de circularidade no grafo, decidimos escolher um dos arcos, acabando por considerar apenas o arco [CH].

Temos, então, o seguinte grafo:

aj (horas)	9:00	9:30	10:00	10:15	10:45	11:15	11:30	12:00
aj (1/4h)	0	2	4	5	7	9	10	12

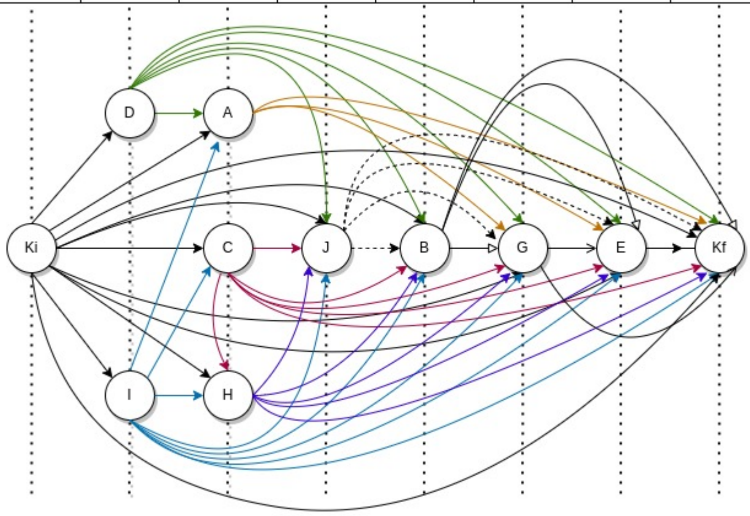

Figura 5: Grafo de compatibilidades.

Grafo de Compatibilidades Transformado

Este grafo tem em conta o grafo de desdobramentos que surge na descrição do problema. No sentido de introduzir a informação no Relax4, o grafo de compatibilidades teve de ser alterado, já que o software aceita apenas números inteiros superiores ou iguais a 1 para os respetivos vértices da rede. Sendo assim, fizeram-se as seguintes alterações:

- Assumiu-se para os vértices A a J, dos clientes, a númeração de 2 a 10;
- Para a fonte, a sede (Ki), assumiu-se o vértice 1, para o destino (Kf), usou-se o vértice 11:
- Para os vértices desdobrados, de A' a J', assumiu-se a representação dos vértices, adicionando um dígito "1" ao mesmo: ou seja, por exemplo, seja A o véritce 4, então A' toma o valor 14, como vértice, e assim sucessivamente. Há apenas a exceção do vértice 10 que toma o valor de 20 (e não de 110).

aj (horas)	9:00	9:30	10:00	10:15	10:45	11:15	11:30	12:00
aj (1/4h)	0	2	4	5	7	9	10	12

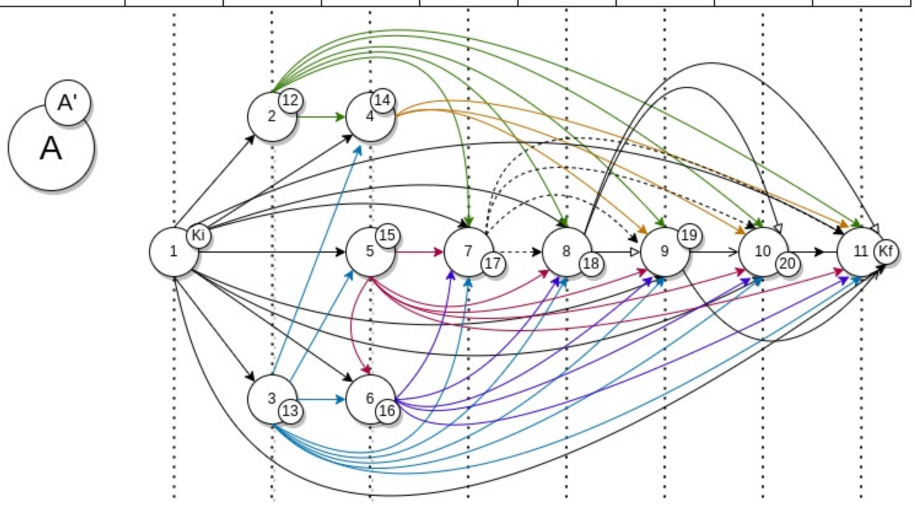

Figura 6: Grafo de compatibilidades com vértices enumerados.

Uma vez que a numeração é agora outra, o grupo refez a matriz com os respetivos custos de deslocação, para melhor análise.

		1	2	3	4	5	6	7	8	9	10	11
		K	D	I	A	C	H	J	B	G	E	K
1	K	0	4	9	1	2	9	10	15	9	6	0
2	D		0	8	6	5	5	7	14	8	4	4
3	I			0	0	5	5	7	13	7	5	9
4	A				0	5	1	3	4	2	2	1
5	C					0	0	1	3	2	2	2
6	H						0	1	3	2	2	9
7	J							0	2	2	2	10
8	B								0	2	3	15
9	G									0	1	9
10	E										0	6
11	K											0

Figura 7: Matriz dde custos atualizada com a numeração atribuída a cada cliente.

3 Questão 1

3.1 Formulação do Problema

Uma das etapas mais importantes num problema de programação linear é a formulação do mesmo, já que permite perceber o problema de um modo mais profundo, nomeadamente, reunir os dados, definir as variáveis de decisão relevantes e o conjunto das restrições. Parte da secção de Descrição do Problema já surgiu inevitavelmente na introdução, todavia, é relevante incluí-la nesta formulação.

3.1.1 Descrição do Problema

Através do problema proposto, facilmente se identificam regras gerais de funcionamento e recursos disponíveis. Pretende-se, assim, distribuir equipas para prestar um serviço a um conjunto de clientes, onde cada cliente possui uma hora de serviço especifica. Estas equipas iniciam a sua saída da sede às 9h e devem servir os seus clientes, com os seus custos associados, regressando à sede no final dos serviços. Então, temos como princípio na resolução do problema a utilização de uma rede que traduza um conjunto de fluxos. Este problema pode então ser descrito como um problema de otimização de fluxos em rede.

Neste sentido, através do grafo de compatibilidades, teve de ser construído o grafo auxiliar, que efetua o desdobramento dos vértices.

Grafo de Desdobramentos

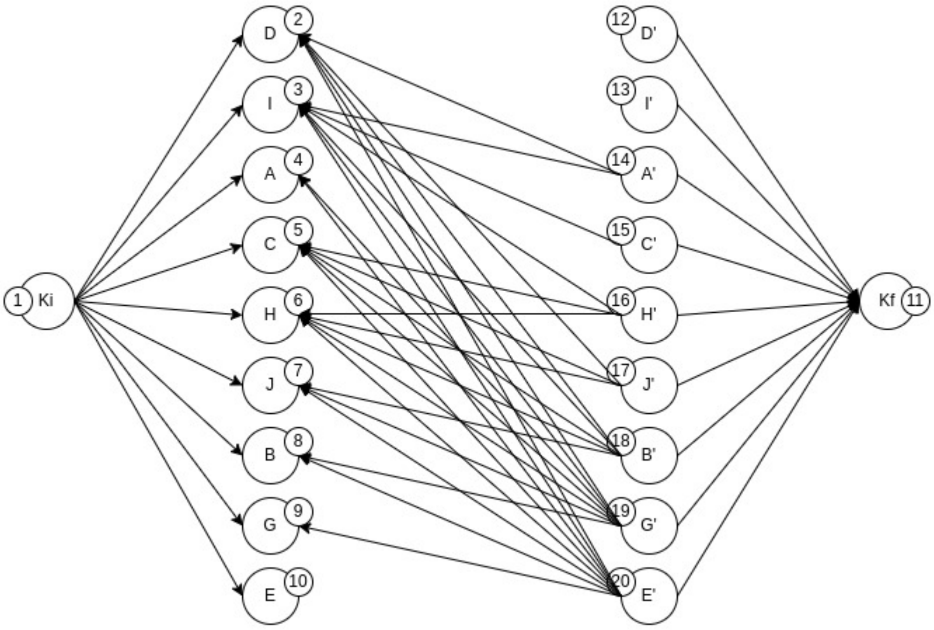

Figura 8: Grafo de Desdobramentos utilizado no Relax.

Sendo *n* o número de vértices (clientes) inicial, basicamente, criou-se este grafo bipartido com 2n vértices (n = 9), sendo cada vértice do grafo inicial dividido em dois vértices. Sendo necessário garantir que cada arco pertence apenas a um caminho, que corresponde a que cada serviço seja apenas executado por uma equipa, a capacidade de todos os arcos de A' a J' para A a J é unitária. No ficheiro do Relax4, são representados os arcos entre vértices decompostos (A'->A, B'->B,...), com custo 0. Optamos por não os representar na figura acima.

3.1.2 Objetivo Principal

O objetivo surge bem explícito no enunciado: "atribuir serviços a efectuar a clientes distribuídos geograficamente a equipas, de modo a minimizar o custo total da operação, que inclui custos de deslocação e custos fixos de utilização de veículos". Tal permite afirmar que o principal objetivo para a resolução deste problema é minimizar os custos associados à operação de distribuição de equipas aos clientes, sabendo o número de equipas que devem sair da sede e ser distribuídas. É necessário encontrar a sua solução ótima, que é o conjunto de caminhos que a(s) equipa(s) deve(m) fazer, tendo em conta os requisitos de tempo e de modo a garantir o mínimo custo de toda a operação.

3.1.3 Rede

A rede utilizada é, então, o grafo bipartido apresentado anteriormente, tendo as seguintes particularidades:

Significado dos vértices e dos arcos

- Cada um dos vértices representa uma entidade diferente. O vértice 1 é a nossa fonte, representando a sede. O vértice 11 é o nosso terminal, representando também a sede, mas, neste caso, o regresso à mesma. Os restantes vértices representam clientes a servir (decomposição). Já os arcos indicam as ligações possíveis entre estas entidades: da sede podemos ir diretamente para qualquer cliente; de um cliente podemos ir para outro cliente cuja hora do serviço seja superior ou igual à hora do serviço do cliente anterior mais o tempo dessa deslocação; de um cliente podemos ir diretamente para a sede, regressando;

Decisões a implmentar no sistema real

- A solução ótima deve traduzir o conjunto de arcos que as equipas percorreram ao servir os clientes, sendo que, em cada arco (à exceção do arco fictício Ki->Kf), pode apenas passar uma equipa. O arco fictício foi criado para saber que equipas não saem da sede, ou seja, que equipas não foram utilizadas das 9 equipas totais. Selecionou-se como número máximo de equipas, o valor 9, já que, no pior caso, iria uma equipa a cada cliente.

Significado dos custos e capacidades

- O conjunto dos custos surge na matriz de "custos de deslocação" disponibilizada. Então, usam-se esses custos para o conjunto dos arcos da rede, contudo, os arcos resultantes da decomposição têm um custo igual a 0 (A'->A; B'->B;...). Assume-se que o custo do arco fictício Ki->Kf é 0 (pois, não se saindo da sede, não há custo associado). Em relação às capacidades, todos os arcos têm capacidade igual a 1, já que, por estes, é permitido passar-se apenas uma vez, havendo exceção no arco Ki->Kf, em que a sua capacidade é igual ao número total de clientes (9), já que na pior da hipóteses teríamos uma equipa por cliente;

Significado das ofertas e consumos

- Em relação à oferta/consumo, o vérice inicial tem valor 9, já que esse é o fluxo máximo de saída (oferta) desse nodo; O vértice final tem valor de -9, já que esse é o fluxo máximo de entrada (procura) nesse nodo. Os vértices resultantes da decomposição (A a J) têm valor igual a -1 - entrada de uma equipa em cada vértice - e (A' a J') têm capacidade igual a 1 - saída de uma equipa de cada vértice. Estes valores de capacidade limitam o fluxo em cada vértice.

Coerência Global do Modelo a Construir

- E porque a rede se relaciona com a globalidade do modelo, achamos por bem, nesta secção, explicar alguns detalhes. O objetivo é minimizar o custo total da operação e não minimizar o número de equipas;
- A Decomposição dos vértices é necessária para que seja possível a seleção do número de equipas a utilizar e também para que, ao entrarmos num vértice, tenhamos um outro vértice (que representa esse mesmo vértice por onde se entrou), por onde se efetua a saída;
- Por uma questão de organização e para a rede não ficar muito confusa, optou-se por não colocar em cada arco o custo e capacidade, na forma, (custo, capacidade). Contudo, na rede da análise da solução, essa forma de representação já é utilizada. As informações desses custos constam na matriz de custos (reoganizada) fornecida. A capacidade do arco fictício é 9 (número máximo de equipas) e a dos restantes vértices é igual a 1.

3.1.4 Dados

Os dados do problema são facilmente extraídos do próprio enunciado e não podem ser alterados. Como dados está subentendida toda a configuração da rede. Então, os dados deste modelo serão, mais propriamente:

- O conjunto de vértices (clientes e sede);
- O conjunto de arcos (respeitam a tabela de atividades, figura 3, respeitam o critério $a_i + t_{ij} \leq a_j$)

- O sentido dos arcos (respeitam o critério acima);
- O conjunto dos custos associados aos arcos (matriz de custos e custo fixo de 1 U.M)
- O conjunto dos tempos de deslocação entre vértices, associados a cada arco (matriz de tempos)

3.1.5 Variáveis de Decisão

Estas variáveis de decisão são o conjunto de valores que se traduzem em decisões a implementar no sistema real. Como referido na descrição do problema, importa saber o conjunto de arcos a selecionar, de modo a que se tenha uma rede onde constem apenas caminhos que minimizem o custo total de operações, tendo-se um número de equipas utilizadas neste percurso.

Então, estas variáveis de decisão terão de ser variáveis inteiras (superiores ou iguais a 0) que definirão o estado de um arco, ou seja, se tiver valor 1 significa que o arco foi selecionado para o percurso, passando uma equipa nesse arco. Caso tenha valor 0, não foi selecionado para o percurso, não passando nenhuma equipa nesse arco. Então, imagine-se o arco A -> B com valor 1, quer dizer que uma equipa vai do cliente A ao B. O conjunto destas informações, em "série", indica qual o caminho que a equipa, ou equipas, devem percorrer. Apesar de termos dito que as variáveis são inteiras e superiores a 0, conseguimos afirmar, mais propriamente, que as variáveis associadas aos arcos serão binárias, tendo-se apenas duas variáveis inteiras (r e t), explicadas mais abaixo.

Reunindo estas informações e tendo em conta o que se pretende representar, as variáveis de decisão terão de:

- Definir se um arco foi ou não selecionado para o percurso;
- Estar associadas ao número de equipas que percorreram determinado arco;
- Estar associadas ao número de equipas que não saíram da sede (não sendo utilizadas no percurso).
- Refletir o custo fixo total (no caso de r e t).

3.1.6 Restrições

As restrições permitirão delimitar o espaço de soluções admissíveis. No nosso caso concreto, as restrições do modelo estarão relacionadas com a conservação do fluxo e capacidade, ao longo dos arcos. Desta forma, reparamos que terão de ser obedecidas as seguintes restrições, com $x_{ij} \equiv$ arco que vai de i para j. A ideia é garantir que, em cada vértice, o fluxo de entrada é igual ao fluxo de sáida. De modo a garantir que em cada vértice se passa apenas uma vez, já que se pretende servir um cliente apenas uma vez. A conservação do fluxo, assim como as restrições de capacidade, estão implícitas na própria configuração da rede, sendo, então, apenas necessário garantir que:

Para i $\equiv$ nodo qualquer do conjunto $\{1, 12, 13, 14, 15, 16, 17, 18, 19, 20\}$ e j $\equiv$ nodo qualquer do conjunto $\{2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$, considerando os arcos respetivos entre estes tipos de nodos, tem-se:

(Para o conjunto de clientes de A a J (vértices de 2 a 10) e para o conjunto de clientes de A' a J', deve verificar-se:)

$$\sum x_{ij} = 1 \tag{1}$$

Seja r o número de arcos de entrada nos nodos do tipo j $\in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$ (A a J) e seja t o número de arcos que saem dos nodos do tipo i $\in \{12, 13, 14, 15, 16, 17, 18, 19, 20\}$ (A' a J'), deve verificar-se:

$$r = t \ \ (\leq 9) \tag{2}$$

(1) Então, para cada vértice do conjunto 2 a 10, garantimos que escolhemos um arco que nele entra. Do mesmo modo, para cada vértice do conjunto 12 a 20, garantimos que escolhemos um arco que dele sai;

(2) Sendo assim, estamos no fundo a garantir que o número de arcos selecionados que entram nos nodos de 2 a 10 é igual ao número de arcos selecionados que saem dos nodos de 12 a 20 (se considerarmos que se selcionam x arcos que entram nos nodos 2 a 10, terão necessariamente de existir x arcos que saem dos nodos 12 a 20). Deste modo, garante-se que sempre que se entra num determinado vértice, também saímos dele, garantindo um fluxo unitário.

3.2 Problema do Escalonamento de Equipas - Modelo

Variáveis de Decisão

Para representar cada arco, utilizamos x_{ij}, com i e j pertencentes a um domínio inteiro e finito, já que representam os vértices de partida e chegada, respetivamente, do arco em questão.

$\boldsymbol{x_{ij}}$: i representa o vértice de origem, j representa o vértice de destino. Permite representar um arco ij.

$$i \in \{1, 12, 13, 14, 15, 16, 17, 18, 19, 20\}$$
$$j \in \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$$

$\boldsymbol{r}$: representa o número de arcos (que entram nos vértices de 2 a 10) a selecionar.
$\boldsymbol{t}$: representa o número de arcos (que saem dos vértices de 12 a 20) a selecionar.

O conjunto de restrições definidas de seguida tem implícita a condição de que as variáveis do tipo x_{ij} tomarão apenas um dos valores: 0 ou 1. Terão o valor 1 se forem selecionados para o percurso e o valor 0 caso contrário. Já as variáveis r e t, como são o número de arcos de entrada/saída selecionados, relfetirão o número de equipas utilizadas no percurso.

$$x_{ij} \in \{0, 1\}$$
$$1 \leq r, t \leq 9$$

Restrições

Em relação ao conjunto de restrições, é coerente e relevante afirmar que se dividem em dois grupos distintos.

Grupo 1
Conjunto de restrições identificado na formulação como (2)

```
vertice2: x12 + x142 + x172 + x182 + x192 + x202 = 1;
vertice3: x13 + x143 + x153 + x163 + x173 + x183 + x193 + x203 = 1;
vertice4: x14 + x194 + x204 = 1 ;
vertice5: x15 + x165 + x175 + x185 + x195 + x205 = 1;
vertice6: x16 + x176 + x186 + x196 + x206 = 1;
vertice7: x17 + x187 + x197 + x207 = 1;
vertice8: x18 + x198 + x208 = 1;
vertice9: x19 + x209 = 1;
vertice10: x110 = 1;
vertice12: x1211 = 1;
vertice13: x1311 = 1;
vertice14: x142 + x143 + x1411 = 1;
vertice15: x153 + x1511 = 1;
vertice16: x163 + x165 + x1611 = 1;
vertice17: x172 + x173 + x175 + x176 + x1711 = 1;
vertice18: x182 + x183 + x185 + x186 + x187 + x1811  = 1;
vertice19: x192 + x193 + x194 + x195 + x196 + x197 + x198 + x1911 = 1;
vertice20: x202 + x203 + x204 + x205 + x206 + x207 + x208 + x209 + x2011 = 1;
```

Grupo 2
Conjunto de restrições identificado na formulação como (1)

```
i = x12 + x13 + x14 + x15 + x16 + x17 + x18 + x19 + x110;
f = x1211 + x1311 + x1411 + x1511 + x1611 + x1711 + x1811 + x1911 + x2011;
i = f;
```

Neste caso, a variável r (na formulação) é a variável i (nas retrições acima) e a variável t (na formulação) é a variável f (nas retrições acima).

Função Objetivo

Não esquecendo nunca o objetivo do problema - "atribuir serviços a efectuar a clientes distribuídos geograficamente a equipas, de modo a minimizar o custo total da operação, que inclui custos de deslocação e custos fixos de utilização de veículos"-, torna-se fulcral que na função objetivo estejam os custos de deslocação associados a cada arco, ou seja, o conjunto dos custos multiplicados pela sua respetiva variável binária.

A função objetivo é:

```
min:
4 x12  + 6 x142 + 11 x172 + 14 x182 + 8 x192 + 4 x202 +
9 x13  + 0 x143 + 5 x153  + 5 x163  + 7 x173 + 13 x183 + 7 x193 + 5 x203 +
1 x14  + 7 x194 + 5 x204 +
2 x15  + 0 x165 + 6 x175  + 11 x185 + 6 x195 + 6 x205 +
9 x16  + 6 x176 + 11 x186 + 10 x196 + 6 x206 +
10 x17 + 4 x187 + 5 x197  + 7 x207 +
15 x18 + 6 x198 + 10 x208 +
9 x19  + 4 x209 +
6 x110 +
4 x1211 + 9 x1311 + 1 x1411 + 2 x1511 +
9 x1611 + 10 x1711 + 15 x1811 + 9 x1911 + 6 x2011 + i;
```

Em relação ao custo fixo no regresso à sede, tem-se:

`custo fixo = 1`, para cada caminho/equipa. Logo, custo fixo total = r (com r igual ao número de equipas utilizadas no percurso). Este r é, acima, identificado pela variável i. Poderíamos ter colocado a variável f, do mesmo modo.

4 Questão 2

Tendo em conta o grafo resultante da decomposição dos vértices, introduzimos a informação no Relax4, que segue a seguinte metodologia:

- A primeira linha corresponde ao número de vértices;
- A segunda linha corresponde ao número de arcos;
- De seguida, as 56 linhas correspondem à listagem dos arcos, segundo o padrão `org dst custo cap`, onde `org` representa o vértice de origem, `dst` o vértice de destino, `custo` o respetivo custo unitário e `cap` a capacidade do arco;
- As restantes 20 linhas representam as ofertas e as procuras.

```
20
57
1 2 4 1
1 3 9 1
1 4 1 1
1 5 2 1
1 6 9 1
1 7 10 1
1 8 15 1
1 9 9 1
1 10 6 1
1 11 0 9

12 4 6 1
12 7 11 1
12 8 14 1
12 9 8 1
12 10 4 1

13 4 0 1
13 5 5 1
13 6 5 1
13 7 7 1
13 8 13 1
13 9 7 1
13 10 5 1

14 9 7 1
14 10 5 1

15 6 0 1
15 7 6 1
15 8 11 1
15 9 6 1
15 10 6 1

16 7 6 1
16 8 11 1
16 9 10 1
16 10 6 1

17 8 4 1
17 9 5 1
17 10 7 1
```

```
18 9 6 1
18 10 10 1

19 10 4 1

2 12 0 1
3 13 0 1
4 14 0 1
5 15 0 1
6 16 0 1
7 17 0 1
8 18 0 1
9 19 0 1
10 20 0 1

12 11 4 1
13 11 9 1
14 11 1 1
15 11 2 1
16 11 9 1
17 11 10 1
18 11 15 1
19 11 9 1
20 11 6 1

9
-1
-1
-1
-1
-1
-1
-1
-1
-1
-9
1
1
1
1
1
1
1
1
1
```

5 Questão 3

Output no Relax4

Após introduzido o ficheiro no relax, foi-nos apresentado o seguinte output:

```
************************************************************************
 NUMBER OF NODES = 20, NUMBER OF ARCS = 57
 UNKNOWN OR UNSPECIFIED INITIALIZATION MODE;USING DEFAULT INITIALIZATION
 ***********************************************************************
 Total algorithm solution time =   0.00313806534 sec.
 OPTIMAL COST =   46.
 NUMBER OF ITERATIONS =  18
 NUMBER OF MULTINODE ITERATIONS =  1
 NUMBER OF MULTINODE ASCENT STEPS =  0
 NUMBER OF REGULAR AUGMENTATIONS =  2
************************************************************************
s 46.
f 1 2 1
f 1 3 1
f 1 4 0
f 1 5 1
f 1 6 0
f 1 7 0
f 1 8 0
f 1 9 0
f 1 10 0
f 1 11 6
f 12 4 0
f 12 7 0
f 12 8 0
f 12 9 0
f 12 10 0
f 13 4 1
f 13 5 0
f 13 6 0
f 13 7 0
f 13 8 0
f 13 9 0
f 13 10 0
f 14 9 0
f 14 10 0
f 15 6 1
f 15 7 0
f 15 8 0
f 15 9 0
f 15 10 0
f 16 7 1
f 16 8 0
f 16 9 0
f 16 10 0
f 17 8 1
f 17 9 0
f 17 10 0
f 18 9 1
f 18 10 0
f 19 10 1
f 2 12 0
f 3 13 0
```

```
f 4 14 0
f 5 15 0
f 6 16 0
f 7 17 0
f 8 18 0
f 9 19 0
f 10 20 0
f 12 11 1
f 13 11 0
f 14 11 1
f 15 11 0
f 16 11 0
f 17 11 0
f 18 11 0
f 19 11 0
f 20 11 1
```

6 Questão 3

Análise de Resultados

Através da análise da solução ótima, conseguimos ter uma ideia do conjunto de emparelhamentos que o nosso grafo terá, como o conjunto das ligações que as equipas fizeram, construindo-se assim o conjunto de caminhos por ondem as equipas passaram. Observe-se o grafo criado, com este conjunto:

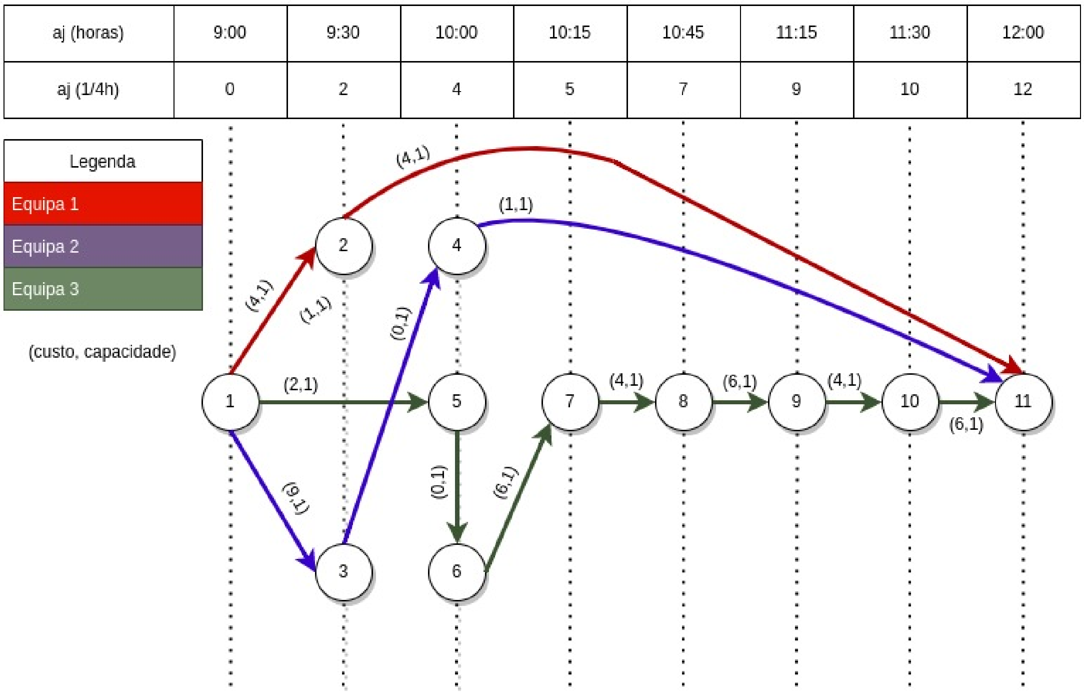

Figura 9: Grafo de caminhos finais percorridos pelas equipas.

Seguem-se agora as observações relevantes:

1. Através da linha do ficheiro Relax4 que simboliza o arco fictício entre Ki e Kf, `f 1 11 6`, conseguimos perceber que 6 equipas percorreram este arco no seu caminho, logo, estas acabam por ser as equipas que não serviram nenhum cliente (não passaram em nenhum vértice de um cliente), sendo as equipas que não saíram da sede. Podemos, então, afirmar que apenas foram utilizadas 3 (9 - 6) das 9 existentes. É este o número ótimo de equipas, que minimiza os custos totais da operação;

2. Através das restantes linhas, observamos que se passa em cada cliente apenas uma vez e que, existem equipas que passam apenas num cliente, regressando de imediato à sede, como é o caso das equipas 1,2 e 3. Há, também, equipas que passam em mais do que um cliente, como é o caso da equipa 4. Observe-se também que todos os clientes são servidos;

3. O custo ótimo, fornecido pelo software, é de 46, que é, se reparamos, a soma de todos os custos dos arcos presentes no grafo acima, ou seja, a soma dos custos dos caminhos percorridos por cada uma das equipas:

   (a) Equipa 1: 4 + 4 = 8 U.M + 1 U.M (custo fixo) = 9 U.M

   (b) Equipa 2: 9 + 0 + 1 = 10 U.M + 1 U.M (custo fixo) = 11 U.M

   (c) Equipa 3: 2 + 0 + 6 + 4 + 6 + 4 + 6 = 36 U.M + 1 U.M (custo fixo) = 37 U.M

   (d) Tendo-se, então, um custo ótimo de 46 U.M, ou seja, com o custo fixo, de 49 U.M

4. Repare-se que, do vértice 5 se vai imediatamente para o 6, pois o custo desse arco é 0. Isto simboliza o que se disse no início do trabalho: Ao visitar o vértice 5 estamos implicitamente a visitar o 6, podendo considerar-se estes dois vértices como um vértice só;

5. Através do grafo de decomposição dos vértices, conseguimos perceber esta relação entre o número de clientes que uma equipa serve. Observe-se a explicação abaixo.

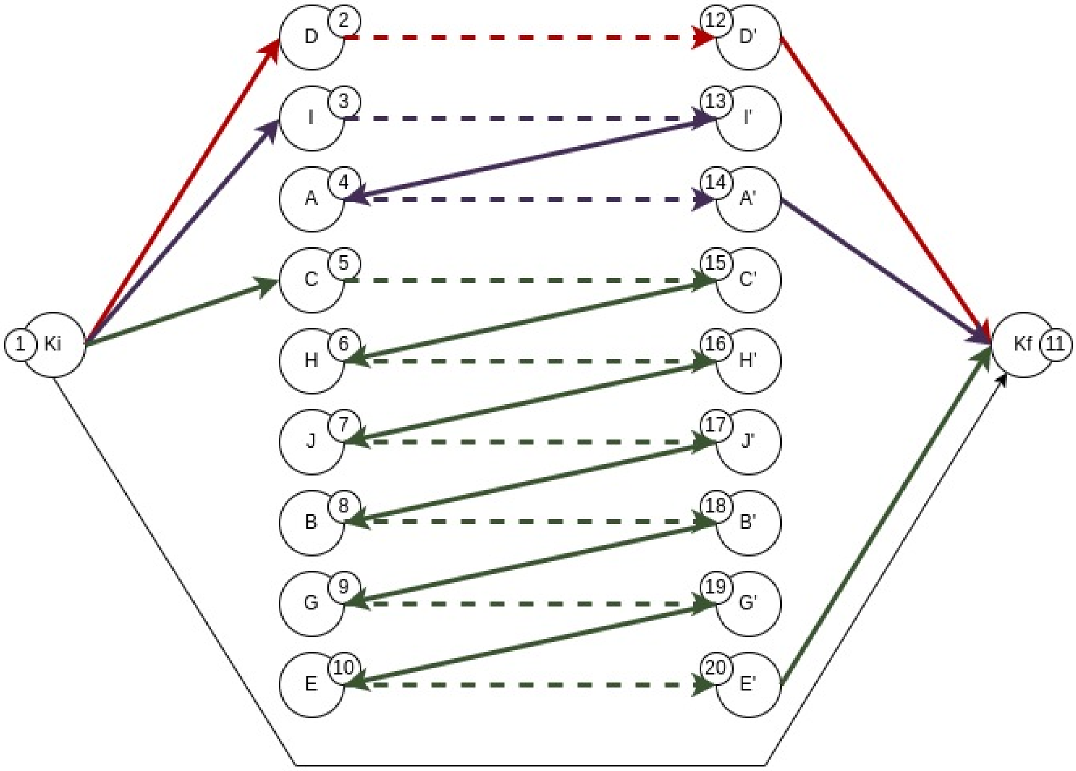

Figura 10: Grafo com os arcos fornecidos pelo Relax4.

Este foi o grafo utilizado para chegar ao que se encontra acima. Os arcos a cheio foram os fornecidos pelo software. Através desses arcos, e sabendo que teríamos de passar num cliente apenas uma vez e que temos de respeitar as restrições temporais, que a numeração dos vértices já considera, conseguimos desenhar os arcos a tracejado, formando um caminho para cada equipa. É fácil de ver que, como ao vértice 1 se ligam 3 arcos, temos 3 equipas diferentes a operar (considerando também as restantes que foram diretamente para o vértice 11). Esta é, então, a solução que minimiza os recursos utilizados, que são, neste caso, os associados à distribuição das equipas pelos clientes (veículos, portagens, deslocações, etc.).

7 Questão 4

Validação do Modelo

Validação do Modelo no LPSolve

Para efetuar a validação do modelo, o grupo decidiu recorrer ao software de programação linear LPSolver, confrontando os resultados obtidos com os que foram fornecidos pelo Relax4. Segue-se o respetivo input e output, acompanhado da sua análise.

Input

```
/* Objective function */
min:
4 x12  + 6 x142 + 11 x172 + 14 x182 + 8 x192 + 4 x202 +
9 x13  + 0 x143 + 5 x153  + 5 x163  + 7 x173 + 13 x183 + 7 x193 + 5 x203 +
1 x14  + 7 x194 + 5 x204 +
2 x15  + 0 x165 + 6 x175  + 11 x185 + 6 x195 + 6 x205 +
9 x16  + 6 x176 + 11 x186 + 10 x196 + 6 x206 +
10 x17 + 4 x187 + 5 x197  + 7 x207 +
15 x18 + 6 x198 + 10 x208 +
9 x19  + 4 x209 +
6 x110 +
4 x1211 + 9 x1311 + 1 x1411 + 2 x1511 + 9 x1611 +
10 x1711 + 15 x1811 + 9 x1911 + 6 x2011 + i;


i = x12 + x13 + x14 + x15 + x16 + x17 + x18 + x19 + x110;
f = x1211 + x1311 + x1411 + x1511 + x1611 + x1711 + x1811 + x1911 + x2011;
i = f;

/* Variable bounds */
/* Entry Edges */
vertice2: x12 + x142 + x172 + x182 + x192 + x202 = 1;
vertice3: x13 + x143 + x153 + x163 + x173 + x183 + x193 + x203 = 1;
vertice4: x14 + x194 + x204 = 1 ;
vertice5: x15 + x165 + x175 + x185 + x195 + x205 = 1;
vertice6: x16 + x176 + x186 + x196 + x206 = 1;
vertice7: x17 + x187 + x197 + x207 = 1;
vertice8: x18 + x198 + x208 = 1;
vertice9: x19 + x209 = 1;
vertice10: x110 = 1;

/* Exit Edges */
vertice12: x1211 = 1;
vertice13: x1311 = 1;
vertice14: x142 + x143 + x1411 = 1;
vertice15: x153 + x1511 = 1;
vertice16: x163 + x165 + x1611 = 1;
vertice17: x172 + x173 + x175 + x176 + x1711 = 1;
vertice18: x182 + x183 + x185 + x186 + x187 + x1811  = 1;
vertice19: x192 + x193 + x194 + x195 + x196 + x197 + x198 + x1911 = 1;
vertice20: x202 + x203 + x204 + x205 + x206 + x207 + x208 + x209 + x2011 = 1;
```

Output

```
Variables result
           49
x12  1
x142 0
x172 0
x182 0
x192 0
x202 0
x13  0
x143 1
x153 0
x163 0
x173 0
x183 0
x193 0
x203 0
x14  1
x194 0
x204 0
x15  0
x165 1
x175 0
x185 0
x195 0
x205 0
x16  0
x176 1
x186 0
x196 0
x206 0
x17  0
x187 1
x197 0
x207 0
x18  0
x198 1
x208 0
x19  0
x209 1
x110 1
x1211 1
x1311 1
x1411 0
x1511 1
x1611 0
x1711 0
x1811 0
x1911 0
x2011 0
i 3
f 3
```

Através do output gerado pelo software LPSolve, conseguimos identificar o conjunto de caminhos pelos quais cada equipa serviu os seus clientes. Têm o valor 1 os arcos que foram selecionados para o percurso. Então, sabemos que as equipas passaram por esses arcos. Contudo, ao analisar esta solução, temos de ter o pensamento "reverso" àquele que tivemos ao analisar o output do relax. Segue-se o grafo bipartido com os arcos selecionados para mais fácil interpretação.

Conseguimos, assim, observar que uma das equipas foi ao cliente 2, seguindo de imediato para a sede, tendo-se o caminho:

$$(1- > 2- > 12- > 11)$$

Depois, observamos o percurso: (1->4->14->3->13->11). Este percurso, à primeira vista, pode ser enganoso, porque nos faz pensar que esse foi o caminho que a equipa fez, contudo, é o caminho reverso, no sentido em que: se a equipa visitou o vértice 4, significa que teve de sair do nodo 14; se saiu do 14, teve de ser visitado o vértice 3 (pois há o arco 14->3). Tendo visitado o vértice 3, e tendo-se o arco 13->11, significa que a equipa, depois de visitar o vértice 3, regressou para a sede, tendo-se o caminho real seguinte:

$$(1- > 3- > 4- > 11)$$

Para a equipa restante, o raciocínio é o mesmo, tendo-se o caminho:

$$(1- > 5- > 6- > 7- > 8- > 9- > 10- > 11)$$

Estes caminhos foram, inclusive, os caminhos que o Relax4 forneceu, como se consegue visualizar na figura que serve de análise.

Por fim, repare-se que o parâmetro i e f refletem o número de equipas utilizadas (3) e que o custo ótimo fornecido é igual ao que foi calculado pelo output do Relax4.

Uma vez a numeração utilizada difere da do enunciado, decidimos efetuar, agora, a respetiva conversão, tendo em conta a numeração do enunciado, tendo-se os seguintes caminhos (com Ki como vértice 21 e Kf como vértice 22):

- Caminho (1->2->11), após conversão, fica: (21->4->22), ou seja, (Ki->D->Kf);
- Caminho (1->3->4->11), após conversão, fica: (21->9->1->22), ou seja, (Ki->I->A->Kf);
- Caminho (1>5>6>7>8>9>10>11), após conversão, fica: (21->3->8->10->2->7->5->20), ou seja, (Ki->C->H->J->B->G->E->Kf).

Esta solução, traduz-se, então no grafo renumerado seguinte:

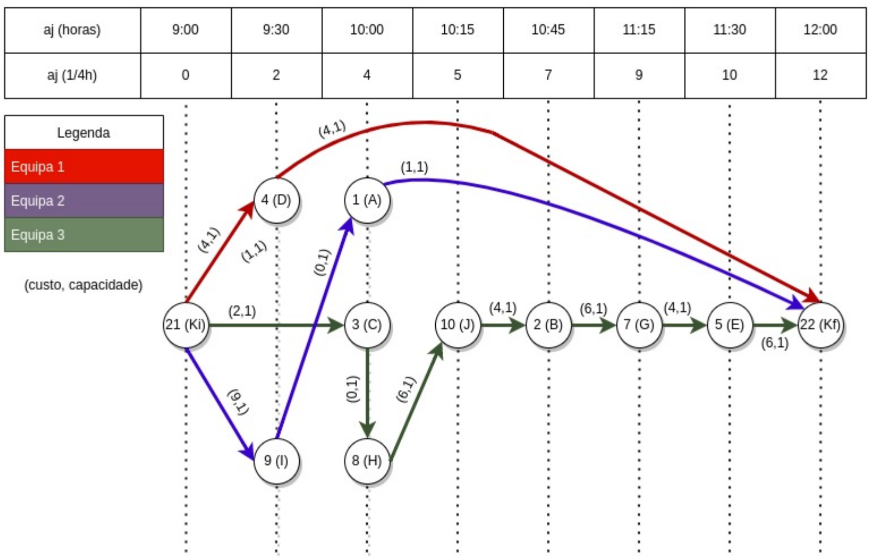

Figura 11: Grafo final com os respetivos percursos (vértices numerados de acordo com o enunciado).

Validação Intuitiva do Modelo

Apesar disto, a validação pode ainda ser mais intuitiva. Esta validação pode ser feita analisando ambos os outputs de cada software e confrontando-os com aquilo que se pretende na vida real, verificando-se as propriedades seguintes:

- Todos os caminhos do resultado são possíveis: o conjunto de caminhos fornecido é um conjunto de caminhos possível, tendo em conta as condições do enunciado, sendo fazível;
- Esses caminhos têm em conta o tempo de deslocação: para todos os caminhos, verifica-se que o primeiro cliente a visitar tem uma hora de serviço anterior ou igual ao próximo;
- Só se passa uma vez em cada vértice: nos caminhos das três equipas, não há vértices iguais e, dentro de uma mesma equipa, não existem vértices repetidos.

Validação Formal do Modelo

Para além das validações anteriores, pode ser feita uma validação mais formal, onde se verifica se a solução ótima, fornecida pelo software LPSolve, é uma solução admissível, substituindo os valores das variáveis nas respetivas restrições e função objetivo, como se segue:

A função objetivo é:

```
    min:  4 x12 + 6 x142 + 11 x172 + 14 x182 + 8 x192 + 4 x202 + 9 x13 + 0 x143
+ 5 x153 + 5 x163 + 7 x173 + 13 x183 + 7 x193 + 5 x203 + 1 x14 + 7 x194 + 5 x204
+ 2 x15 + 0 x165 + 6 x175 + 11 x185 + 6 x195 + 6 x205 + 9 x16 + 6 x176 + 11 x186
+ 10 x196 + 6 x206 + 10 x17 + 4 x187 + 5 x197 + 7 x207 + 15 x18 + 6 x198 + 10
x208 + 9 x19 + 4 x209 + 6 x110 + 4 x1211 + 9 x1311 + 1 x1411 + 2 x1511 + 9 x1611
+ 10 x1711 + 15 x1811 + 9 x1911 + 6 x2011 + i;
```

Efetuando a substituição dos valores dados pelos LPSolver, vem:

```
    min:  4 * x12 + 1 * x14 + 6 * x176 + 4 * x187 + 6 * x198 + 4 * x209 + 6 * x110
+ 4 * x1211 + 9 * x1311 + 2 * x1511 + i = 4 + 1 + 6 + 4 + 6 + 4 + 6 + 4 + 9 +
2 + 3 = 49
```

O custo ótimo apresentado está, então, validado, com a função objetivo.

Confrontando com as respetivas restrições, temos:

```
i = x12 + x13 + x14 + x15 + x16 + x17 + x18 + x19 + x110 ⇔
i = 1 + 0 + 1 + 0 + 0 + 0 + 0 + 0 + 1 ⇔
i = 3

f = x1211 + x1311 + x1411 + x1511 + x1611 + x1711 + x1811 + x1911 + x2011 ⇔
f = 1 + 1 + 0 + 1 + 0 + 0 + 0 + 0 + 0 ⇔
f = 3 = i ⇔ True

vertice2:  x12 + x142 + x172 + x182 + x192 + x202 = 1 ⇔
1 + 0 + 0 + 0 + 0 + 0 = 1 ⇔ True
vertice3:  x13 + x143 + x153 + x163 + x173 + x183 + x193 + x203 = 1 ⇔
0 + 1 + 0 + 0 + 0 + 0 + 0 + 0 = 1 ⇔ True
vertice4:  x14 + x194 + x204 = 1 ⇔
1 + 0 + 0 = 1 ⇔ True
vertice5:  x15 + x165 + x175 + x185 + x195 + x205 = 1 ⇔
0 + 1 + 0 + 0 + 0 + 0 = 1 ⇔ True
vertice6:  x16 + x176 + x186 + x196 + x206 = 1 ⇔
0 + 1 + 0 + 0 + 0 = 1 ⇔ True
vertice7:  x17 + x187 + x197 + x207 = 1 ⇔
```

```
0 + 1 + 0 + 0 = 1 ⇔ True
vertice8:  x18 + x198 + x208 = 1 ⇔
0 + 1 + 0 = 1 ⇔ True
vertice9:  x19 + x209 = 1 ⇔
0 + 1 = 1 ⇔ True
vertice10:  x110 = 1 ⇔
1 = 1 ⇔ True
vertice12:  x1211 = 1 ⇔
1 = 1 ⇔ True
vertice13:  x1311 = 1 ⇔
1 = 1 ⇔ True
vertice14:  x142 + x143 + x1411 = 1; ⇔
0 + 1 + 0 = 1 ⇔ True
vertice15:  x153 + x1511 = 1 ⇔
0 + 1 = 1 ⇔ True
vertice16:  x163 + x165 + x1611 = 1 ⇔
0 + 1 + 0 = 1 ⇔ True
vertice17:  x172 + x173 + x175 + x176 + x1711 = 1 ⇔
0 + 0 + 0 + 1 + 0 = 1 ⇔ True
vertice18:  x182 + x183 + x185 + x186 + x187 + x1811 = 1 ⇔
0 + 0 + 0 + 0 + 1 + 0 = 1 ⇔ True
vertice19:  x192 + x193 + x194 + x195 + x196 + x197 + x198 + x1911 = 1 ⇔
0 + 0 + 0 + 0 + 0 + 0 + 1 + 0 = 1 ⇔ True
vertice20:  x202 + x203 + x204 + x205 + x206 + x207 + x208 + x209 + x2011 = 1
⇔
0 + 0 + 0 + 0 + 0 + 0 + 0 + 1 + 0 = 1 ⇔ True
```

Assim, conseguimos perceber que a solução encontrada é assim uma solução admissível válida, pois respeita o conjunto de restrições estabelecidas, satisfazendo-as.

8 Conclusão e Apreciação crítica

Um problema de transportes tem que se lhe diga, envolvendo uma preparação cuidadosa e exigente, necessária perante o desenvolvimento do problema com as ferramentas solicitadas. Assim, o grupo dedicou algum tempo à reflexão sobre o problema e qual a melhor forma de o resolver, tendo em conta o software a utilizar: Relax4.

Para além disso, enfrentamos algumas dificuldades, nomeadamente em como obter o grafo de compatibilidades, e também aquando da validação, através do LPSolver. A existência de circularidade no grafo (devido aos arcos CH e HC), foi algo que não estávamos à espera de encontrar e, para o resolver, achamos por bem retirar uma das direções. Contudo, nesse sentido, podemos pensar que, quando servimos o cliente C, serivmos também o cliente H, logo, poderíamos alternativamente considerar a existência de apenas um dos nodos. Outra alternativa seria a mudança desse custo nula para um custo diferente de 0, mas achamos por bem não alterar o enunciado.

Apesar disso, de um modo sumário, o trabalho realizado contribuiu para aprofundar o conhecimento alusivo à programação linear com otimização de redes, através da aplicação prática dos conceitos e respetiva utilização e manipulação de software útil nessa mesma área. O conjunto de objetivos inicialmente estabelecidos foi cumprido e o projeto prático, na perspetiva do grupo, foi bem conseguido.

Referências Bibliográficas

I. Carvalho, J. M. V. D. (n.d.). Modelos de Investigação Operacional
(Apêndice disponibilizado pelo docente)