Universidade do Minho
Licenciatura em Engenharia Informática

Investigação Operacional

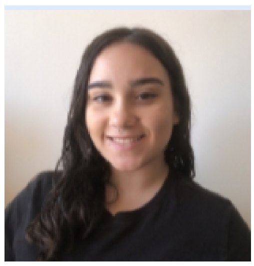
Trabalho Prático 2

ESCALONAMENTO DE EQUIPAS

(com tempos de serviço fixos)


Henrique Pereira
a100831


Luís Caetano
a100893


Mariana Gonçalves
a100662


Maya Gomes
a100822

13 de maio de 2023

Conteúdos

1 Introdução

No presente relatório são abordados os elementos-chave do desenvolvimento do segundo trabalho prático, no contexto da unidade curricular de Investigação Operacional. O enunciado propõe a formulação de um modelo de fluxos em rede para resolver um problema relacionado com o escalonamento de equipas de trabalho, considerando tempos de serviço fixos.

Em detalhes, trata-se de um problema de transporte que busca minimizar o custo total de uma operação envolvendo diversos custos, como deslocamento e utilização de veículos. O trabalho aborda a situação em que é necessário atribuir serviços específicos a clientes geograficamente distribuídos entre equipas de trabalho, com o objetivo de minimizar os custos gerais da operação.

O objetivo principal é determinar o número ótimo de equipas para realizar os serviços de forma pontual, cumprindo os horários de trabalho da sede e dos clientes, e levando em consideração a minimização dos custos de deslocamento (incluindo portagens, combustível, etc.). Para além disso, determinar as melhores rotas possíveis. As deslocações ocorrem entre a sede e os clientes, e entre diferentes clientes, bem como do cliente de volta à sede após o serviço.

O problema utiliza informações disponíveis em uma tabela que descreve os clientes e os horários em que os serviços devem ser prestados. Cada parâmetro é explicado em detalhes na secção de Descrição do Problema, fornecendo informações introdutórias relevantes. Em seguida, são apresentadas matrizes com os tempos e custos de deslocamento entre os clientes e entre os clientes e a sede em Keleirós. É importante observar que cada cliente é representado por um identificador específico na tabela.

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: Indique o valor de ABCDE, e o grafo de compatibilidades (pode ser um desenho feito à mão e colado como imagem no relatório).

2.1 Remoção de clientes e tempos de deslocação dependentes de ABCDE

Segundo o enunciado temos que : 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 for par,remover o cliente D;
- se E for par,remover o cliente E;

Desta forma, sendo que os números de inscrição dos estudantes têm a seguinte relação: 100662 < 100822 < 100831 < 100893, iremos utilizar:

- xABCDE = 100893
- sendo que:
  - a1 = 0+1 = 1;
  - a8 = 8+1 = 9;
  - D é ímpar, pelo que, não removemos o cliente D, isto é, o Diogo;
  - E é ímpar, pelo que, não removemos o cliente E, isto é, o Eduardo.

Sendo que a junção dos critérios de remoção ao nosso número de aluno não removam nenhum cliente, permanecemos com a mesma matriz e seguimos com as seguinte tabela:

j	cliente	a_j (¼hora)	a_j (hora do serviço)
1	Ana	1	09:15
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	9	11:15
9	Inês	2	09:30
10	José	5	10:15

Figura 3: Informação alusiva ao serviço a prestar a cada cliente.

2.2 Grafo de Compatibilidades Resultante

Para obter o grafo de compatibilidades resultante, foi necessário realizar cálculos para determinar se era possível chegar a tempo ao próximo cliente, seguindo a condição estabelecida no enunciado. Foram verificados quais clientes (ou o ponto de origem, se aplicável) era possível alcançar a partir de cada cliente, considerando o momento em que uma equipa sai de um cliente (hora do serviço ou, no caso do vértice K, a hora de início), somando o tempo de deslocamento, e comparando com a hora de início do próximo serviço de destino (próximo cliente).

Na tabela abaixo, foi verificado se a propriedade $a_j, i.e., a_i + d_i + t_{ij} \leq aj$, era cumprida para cada par de vértices, determinando assim os emparelhamentos a serem estabelecidos no grafo de compatibilidades.

Pares com K	Incluir (S/N)		Pares com A	Incluir (S/N)		Pares com D	Incluir (S/N)
KA	0+1<=1 (S)		AD	1+1+2<=2(N)		DI	2+1+2<=2(N)
KD	0+1<=2(S)		AI	1+1+0<=2(S)		DC	2+1+3<=4(N)
KI	0+2<=2 (S)		AC	1+1+1<=4(S)		DJ	2+1+3<=5(N)
KC	0+2<=4(S)		AJ	1+1+3<=5(S)		DF	2+1+3<=6(S)
KJ	0+4<=5(S)		AF	1+1+3<=6(S)		DB	2+1+5<=7(N)
KF	0+4<=6 (S)		AB	1+1+4<=7(S)		DG	2+1+3<=9(S)
KB	0+5<=7(S)		AG	1+1+2<=9(S)		DH	2+1+3<=9(S)
KG	0+3<=9(S)		AH	1+1+1<=9(S)		DE	2+1+1<=10(S)
KH	0+1<=9(S)		AE	1+1+2<=10(S)			
KE	0+2<=10(S)						
Pares com I	Incluir (S/N)		Pares com C	Incluir (S/N)		Pares com J	Incluir (S/N)
IC	2+1+1<=4(S)		CJ	4+1+1<=5(N)		JF	5+1+3<=6(N)
IJ	2+1+3<=5(N)		CF	4+1+3<=6(N)		JB	5+1+2<=7(N)
IF	2+1+3<=6(S)		CB	4+1+3<=7(N)		JG	5+1+2<=9(S)
IB	2+1+4<=7(S)		CG	4+1+2<=9(S)		JH	5+1+1<=9(S)
IG	2+1+2<=9(S)		CH	4+1+0<=9(S)		JE	5+1+2<=10(S)
IH	2+1+1<=9(S)		CE	4+1+2<=10(S)			
IE	2+1+2<=10(S)						
Pares com F	Incluir (S/N)		Pares com B	Incluir (S/N)		Pares com G	Incluir (S/N)
FB	6+1+3<=7(N)		BG	7+1+2<=9(N)		GH	9+1+2<=9(N)
FG	6+1+2<=9(S)		BH	7+1+3<=9(N)		GE	9+1+1<=10(N)
FH	6+1+3<=9(N)		BE	7+1+3<=10(N)			
FE	6+1+2<=10(S)						
Pares com H	Incluir (S/N)		Pares com E	Incluir (S/N)			
HE	9+1+2<=10(N)		É O ÚLTIMO CLIENTE				

Tabela 1: Tabela com os emparelhamentos e respetivos custos.

Desta forma, os arcos [AD], [DI], [DC], [DJ], [DB], [IJ], [CJ], [CF], [CB], [JF], [JB], [FB], [FH], [BG], [BH], [BE], [GH], [GE] e [HE] não estarão representado no grafo de compatibilidades.

Além disso, apenas os emparelhamentos entre nós cujo tempo de início de serviço seja igual ou posterior serão considerados. Os arcos para o nó Kf, que representam o retorno à sede, também não serão representados na tabela.

Procedemos então com o seguinte grafo:

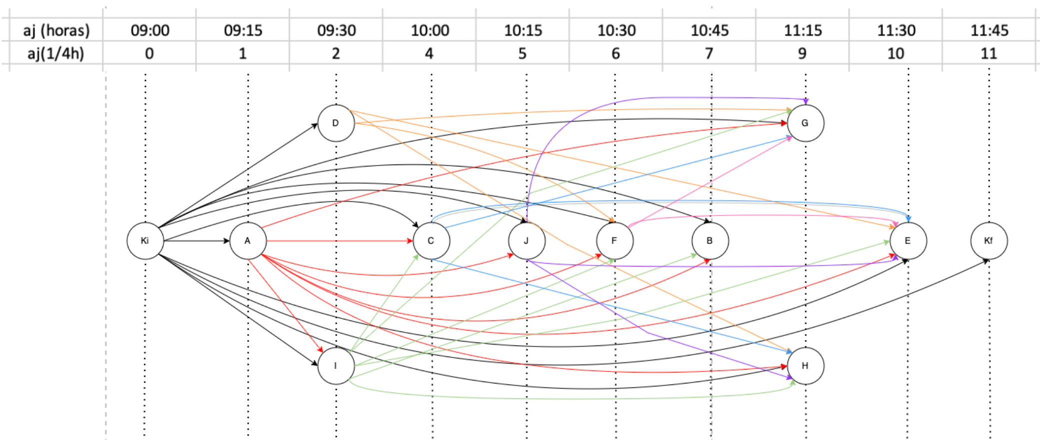


Figura 4: Grafo de compatibilidades com vértices enumerados.

2.3 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 numeração de 2 a 11;
- Para a fonte, a sede (Ki), assumiu-se o vértice 1, para o destino (Kf), usou-se o vértice 12:
- 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 vértice 2, então A' toma o valor 12, como vértice, e assim sucessivamente. Há apenas a exceção dos vértices 10 e 11 que tomam os valores de 20 e 21 respetivamente (e não de 110 e 111).

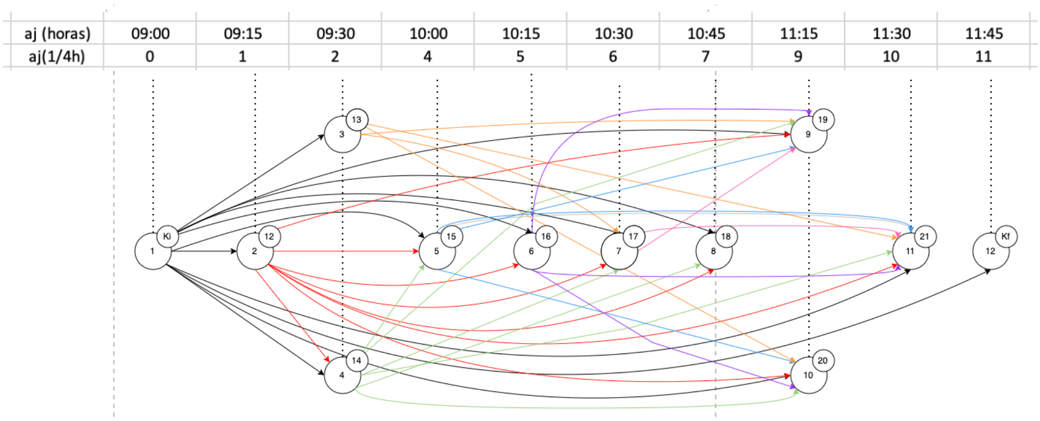


Figura 5: Grafo de compatibilidades com vértices enumerados.

Uma vez que a numeração é agora outra, segue uma nova a matriz com os respetivos custos de deslocação atualizados, para uma melhor análise.

		2	3	4	5	6	7	8	9	10	11	22
		A	D	I	C	J	F	B	G	H	E	K
1	K	1	4	9	2	10	11	15	9	9	6	0
2	A		6	0	5	7	10	13	7	5	5	1
3	D			6	8	11	8	14	8	8	4	4
4	I				5	7	10	13	7	5	5	9
5	C					6	10	11	6	0	6	2
6	J						8	4	5	6	7	10
7	F							8	5	10	6	11
8	B								6	11	10	15
9	G									10	4	9
10	H										6	9
11	E											6
						custos de deslocação						

Figura 6: Matriz de custos atualizada com a numeração atribuída a cada cliente.

3 Questão 1: Apresente a formulação deste problema e o modelo de problema de fluxo sem rede.

3.1 Formulação do Problema

A formulação do problema é uma das partes mais importantes da programação linear, pois ajuda-nos a compreendê-lo de forma mais detalhada, reunindo os dados necessários, identificando as variáveis de decisão relevantes e as restrições envolvidas.

Embora algumas informações já tenham sido abordadas na introdução, é importante incluí-las novamente na formulação para uma melhor compreensão do problema.[3]

3.2 Descrição do problema

Ao propor um problema específico, é possível identificar regras gerais de funcionamento e recursos disponíveis. Neste caso, o objetivo é distribuir equipas para prestar serviços a clientes que têm horários específicos. As equipas partem da sede às 9h00 e devem atender todos os seus clientes, levando em conta os custos envolvidos, antes de retornarem à mesma.

Para resolver esse problema, é necessário usar uma rede que represente o fluxo de informações e recursos. Portanto, este pode ser considerado como um problema de otimização de fluxos em rede. Para resolver isto, foi necessário construir um grafo auxiliar a partir do grafo de compatibilidades, que desdobra os vértices e ajuda a encontrar a solução.[1] [2]

3.3 Grafo de Desdobramentos

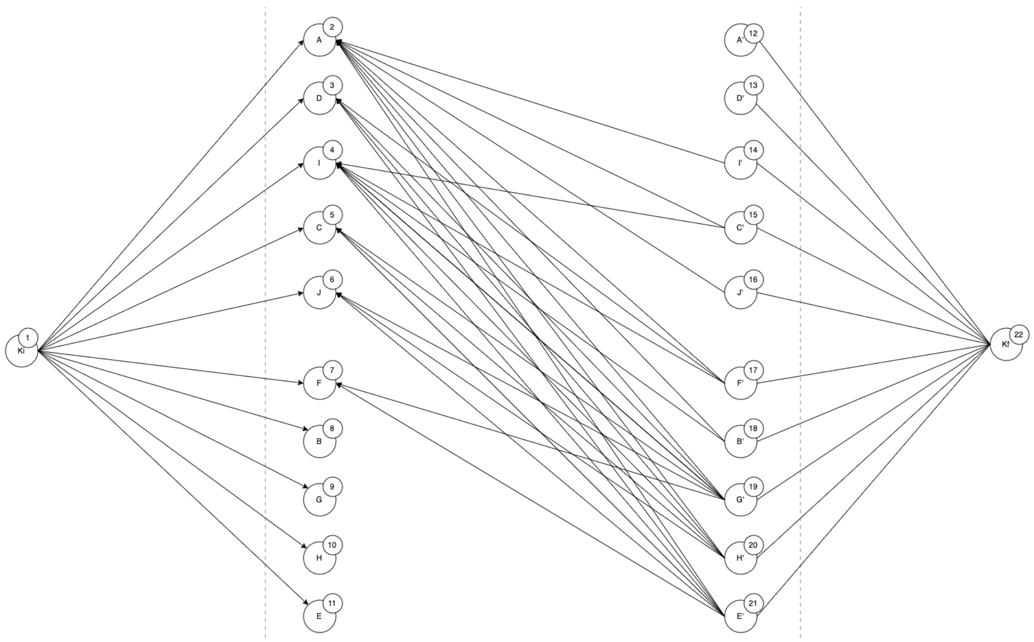


Figura 7: Grafo de desdobramento utilizado no Relax4.

Para criar um grafo bipartido com 2n vértices (no caso, n = 10), onde cada vértice representa um cliente, foi necessário dividir cada vértice do grafo inicial em dois vértices. Para garantir que cada serviço seja executado apenas por uma equipa, foi necessário assegurar que cada arco pertence a apenas um caminho e, portanto, a capacidade de todos os arcos de A' a J' para A a J seja unitária. No arquivo do Relax4, os arcos entre os vértices divididos (por exemplo, A'->A, B'->B, etc.) são representados com um custo de 0. Embora esses arcos não sejam representados na figura, eles são importantes para resolver o problema de distribuição de equipas e serviços aos clientes.

3.3.1 Objetivo Principal

Como referido no enunciado o objetivo do trabalho é "atribuir serviços a efetuar 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 ". Ou seja, o devemos procurar reduzir os custos envolvidos na distribuição das equipas aos clientes, ao saber quantas equipas devem sair da sede e ser distribuídas. A solução ótima é encontrar o conjunto mais eficiente de caminhos que as equipas devem seguir, considerando os requisitos de tempo e garantindo que a operação inteira seja executada com o menor custo possível.

3.3.2 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 12 é o nosso terminal, representando também a sede, mas, neste caso, o regresso à mesma. Os restantes vértices representam clientes a servir (decomposição).

Os arcos servem para mostrar as ligações que existem entre diferentes entidades. Por exemplo, podemos ir diretamente da sede para qualquer cliente, ou de um cliente para outro cuja hora de serviço seja igual ou maior que a do cliente anterior mais o tempo de deslocamento necessário. Também é possível ir diretamente de um cliente para a sede, retornando assim ao ponto inicial(sede).

Decisões a implementar no sistema real

A solução ótima deve refletir os caminhos percorridos pelas equipas ao atenderem os clientes. Cada caminho só pode ser percorrido por uma equipa, exceto pelo caminho fictício que liga a equipa de partida à equipa de chegada. Esse caminho foi criado para identificar quais equipas não foram utilizadas, ou seja, quais não saíram da sede. Assim, foi definido que o número máximo de equipas seria 10, já que no pior cenário, uma equipa teria que atender cada cliente.

Significado dos custos e capacidades

A matriz de "custos de deslocação"contém os custos necessários para se deslocar entre diferentes locais. Esses custos são usados para criar a rede de conexões entre os diferentes pontos, mas os custos associados aos caminhos fictícios (A' -> A; B' -> B; ...) são considerados como sendo zero. O caminho fictício Ki -> Kf é também considerado como tendo custo zero, pois não há custo associado à permanência na sede. Em relação à capacidade dos caminhos, todos os caminhos possuem capacidade máxima igual a um, pois só é permitido passar por eles uma única vez. A exceção é o caminho Ki -> Kf, que tem capacidade máxima igual ao número total de clientes (10), pois como referido anteriormente, na pior das hipóteses, uma equipa seria necessária para cada cliente.

Significado das ofertas e consumos

Quanto à oferta e demanda, o nodo de origem tem uma oferta máxima de 10, representando o fluxo máximo de saída daquele ponto. O nodo de destino tem uma demanda máxima de -10, representando o fluxo máximo de entrada para aquele ponto. Os vértices resultantes da decomposição (A a J) têm um valor de -1, representando a entrada de uma equipa em cada vértice. Os vértices correspondentes à decomposição (A' a J') possuem capacidade máxima igual a 1, representando a saída de uma equipa de cada vértice. Esses valores de capacidade limitam o fluxo de entrada e saída em cada vértice.

Coerência Global do Modelo a Construir

Nesta subdivisão, explica-se alguns detalhes sobre a relação da rede com o modelo como um todo.

O objetivo principal é minimizar o custo total da operação, e não necessariamente o número de equipas a serem utilizadas. A decomposição dos vértices é necessária para permitir a seleção do número de equipas e para garantir que, ao entrar em um vértice, haja um vértice correspondente para saída. Para evitar confusão na rede, optamos por não incluir o custo e a capacidade em cada arco na forma (custo, capacidade), mas essas informações podem ser encontradas na matriz de custos fornecida. A capacidade do arco fictício é de 10 (representando o número máximo de equipas), enquanto os outros vértices têm capacidade máxima de 1.

3.3.3 Dados

A configuração da rede do problema são facilmente extraídos do próprio enunciado e não podem ser alterados. 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) e respeitam o critério $a_i + d_i + t_{ij} \leq aj$);
- 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).

3.3.4 Variáveis de Decisão

As variáveis de decisão são os valores que determinam as escolhas a serem feitas no sistema real. Para resolver o problema descrito, é importante identificar quais são os arcos da rede que serão selecionados para minimizar o custo total da operação e determinar o número de equipas que serão usadas no percurso. Essas variáveis de decisão devem ser inteiras e maiores ou iguais a zero e serão usadas para indicar se um arco foi selecionado para o percurso ou não. Se o valor for 1, significa que uma equipa passará pelo arco, e se for 0, nenhuma equipa passará. As variáveis associadas aos arcos serão binárias, com apenas duas possibilidades (r e t) e serão usadas para determinar o número de equipas que percorrem cada arco e o número de equipas que permanecem na sede e não são utilizadas no percurso. As variáveis de decisão também refletirão o custo fixo total (no caso de r e t).

3.3.5 Restrições

As restrições são regras importantes para garantir que as soluções encontradas pelo modelo sejam viáveis. No nosso caso, as restrições estão relacionadas com a conservação de fluxo e capacidade, ao longo dos arcos. Isso significa que precisamos garantir que o fluxo de entrada em cada vértice é igual ao fluxo de saída e que cada cliente é atendido apenas uma vez. Essas restrições estão implícitas na estrutura da rede, mas necessitamos garantir que sejam atendidas por meio das seguintes condições, onde x_{ij} representa o arco que liga i a j:

Para $i \equiv$ nó qualquer do conjunto $\{1, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22\}$ e $j \equiv$ nó 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 11) e para o conjunto de clientes de A' a J', deve verificar- se:)

$$\sum_i x_{i,j} = 1 \tag{1}$$

Seja r o número de arcos de entrada nos nodos do tipo $j \in \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$ (A a J) e seja t o número de arcos que saem dos nodos do tipo $i \in \{12, 13, 14, 15, 16, 17, 18, 19, 20, 21\}$ (A' a J'), deve verificar-se:

$$r = t(\leq 10) \tag{2}$$

Desta forma:

(1) Para cada nó nos conjuntos 2 a 11 e 12 a 21, escolhemos um único arco que entra ou sai, respetivamente.

(2) Ao fazer isso, estamos garantindo que o número de arcos selecionados que entram nos nós de 2 a 11 é igual ao número de arcos selecionados que saem dos nós de 12 a 21. Em outras palavras, se selecionarmos x arcos que entram nos nós de 2 a 11, também teremos x arcos que saem dos nós de 12 a 21. Isso garante que sempre que entrarmos num nó, também sairemos dele, mantendo um fluxo unitário.

3.4 Problema do Escalonamento de Equipas - Modelo

3.4.1 Variáveis de Decisão

Para representar cada arco, utilizamos x_i_j , com i e j pertencentes a um domínio inteiro e finito, pois esses representam os vértices de partida e chegada, respetivamente, do arco em questão.

- x_{ij} : i representa o vértice de origem e j representa o vértice de destino. Permitindo assim representar um arco ij.

$$\begin{aligned} i &\in \{1,13,14,15,16,17,18,19,20,21,22\} \\ j &\in \{2,3,4,5,6,7,8,9,10,11\} \end{aligned} \tag{3}$$

- r: representa o número de arcos (que entram nos vértices de 2 a 11) a selecionar;
- t: representa o número de arcos (que saem dos vértices de 12 a 21) a selecionar.

Essas restrições estabelecem que as variáveis do tipo x_{ij} só podem assumir dois valores possíveis, 0 ou 1. Se uma variável x_{ij} for atribuída o valor 1, isso significa que o percurso que liga o nó i ao nó j foi selecionado. Caso contrário, se x_{ij} for atribuída o valor 0, o percurso não foi selecionado.

Já as variáveis r e t indicam o número de arcos de entrada e saída selecionados, respetivamente. Elas refletem o número de equipas utilizadas no percurso, ou seja, quantas equipas estão percorrendo os arcos de entrada e saída selecionados. Por exemplo, se r for igual a 3, isso significa que três equipas estão utilizando os arcos de entrada selecionados.

$$\begin{gathered} x_{ij} \in \{0,1\} \\ 1 \leq r,t \leq 10 \end{gathered} \tag{4}$$

3.4.2 Restrições

GRUPO I

Conjunto de restrições identificado na formulação como (2).

```
vertice2: x12 + x142 + x152 + x162 +x172 + x182 + x192 + x202 + x212 = 1
vertice3: x13 + x173 +x193 + x203 + x213 = 1
vertice4: x14 +1 x154 + x174 + x184 + 194 + x204 + x214 = 1
vertice5: x15 + x195 + x205 + x215 = 1
vertice6: x16 + x196 + x206 + x216 = 1
vertice7: x17 + x197 + x217 = 1
vertice8: x18 = 1
vertice9: x19 = 1
vertice10: x110 = 1
vertice11: x111 = 1
vertice12: x1222=1
vertice13: x1322 = 1
vertice14: x142 + x1422 = 1
vertice15: x152 + x154 + x1522 = 1
vertice16: x162 + x1622 = 1
vertice17: x172 + x173 + x174 + x1722 = 1
vertice18: x182 + x184 + x1822 = 1
vertice19: x192 + x194 + x195 + x196 + x197 + x1922 = 1
```

```
vertice20: x202 + x203 + x204 + x205 +x206 + x2022 = 1
vertice21: x212 + x213 + x214 + x215 + x216 + x217 + x2122 = 1
```

GRUPO II

Conjunto de restrições identificado na formulação como (1).

```
i = x12 + x13 + x14 + x15 + x16 + x17 + x18 + x19 + x110 + x111;
f = x1222 + x1322 + x1422 + x1522 + x1622
+ x1722 + x1822 + x1922 + x2022 + x2122;
i = f;
```

3.5 Função Objetivo

Não esquecendo o objetivo do problema, torna-se crucial que na função objetivo estejam os custos de deslocação associados a cada arco, isto é, o conjunto dos custos multiplicados pela sua respetiva variável binária.

Assim, a função objetivo é:

```
min:
 1 x12 + 0 x142 + 5 x152 + 7 x162 +10 x172
 + 13 x182 + 7 x192 + 5 x202 + 5 x212
 + 4 x13 + 8 x173 +8 x193 + 8 x203
 + 4 x213 + 9 x14 +5 x154 + 10 x174
 + 13 x184 + 7 x194 + 5 x204 + 5 x214
 + 2 x15 + 6 x195 + 0 x205 + 6 x215 +
 10 x16 + 5 x196 + 6 x206 + 7 x216
 + 11 x17 + 5 x197 + 6 x217 + 15 x18
 + 9 x19 + 9 x110 + 6 x111 + 1 x1222
 + 4 x1322 + 9 x1422+ 2 x1522+ 10 x1622
 + 11 x1722 + 15 x1822 + 9  x1922
 + 9 x2022 + 6 x2122 +i;
```

Em relação ao custo fixo no regresso à sede, tem-se: custo fixo = 1, para cada caminho/equipa. Logo, o custo fixo total = r (com r igual ao número de equipas utilizadas no percurso). Este r é, acima, identificado pela variável i. Poderíamos também ter colocado a variável f, do mesmo modo.

4 Questão 2: Apresente o ficheiro de input submetido ao software de optimização em rede (por exemplo, o Relax4) (cut-and-paste).

Tendo em conta o grafo que obtivemos mediante a decomposição dos vértices, usamos o Relax4, onde introduzimos a informação, seguindo a seguinte metodologia:

- O número de vértices corresponde **à primeira linha**;
- O número de arcos corresponde **à segunda linha**;
- De seguida, as 57 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 22 linhas representam as ofertas e as procuras.

```
22
57

1 2 1 1
1 3 4 1
1 4 9 1
1 5 2 1
1 6 10 1
1 7 11 1
1 8 15 1
1 9 9 1
1 10 9 1
1 11 6 1
1 22 0 10

12 4 0 1
12 5 5 1
12 6 7 1
12 7 10 1
12 8 13 1
12 9 7 1
12 10 5 1
12 11 1 1

13 7 8 1
13 9 8 1
13 10 8 1
13 11 8 1

14 5 5 1
14 7 10 1
14 8 13 1
14 9 7 1
14 10 5 1
14 11 5 1

15 9 6 1
15 10 0 1
15 11 6 1

16 9 5 1
16 10 6 1
```

```
16 11 7 1

17 9 5 1
17 11 6 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
11 21 0 1

12 22 1 1
13 22 4 1
14 22 9 1
15 22 2 1
16 22 10 1
17 22 11 1
18 22 15 1
19 22 9 1
20 22 9 1
21 22 6 1

10
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
1
1
1
1
1
1
1
1
1
1
-10
```

5 Questão 3: Apresente o ficheiro de output produzido pelo programa (cut-and-paste).

Output no Relax4

Foi-nos apresentado o seguinte *output* após introduzir o ficheiro no relax4:

```
NUMBER OF NODES = 22, NUMBER OF ARCS = 57
UNKNOWN OR UNSPECIFIED INITIALIZATION MODE;USING DEFAULT INITIALIZATION
***********************************
Total algorithm solution time =   0.00321698189 sec.
OPTIMAL COST =   88.
NUMBER OF ITERATIONS =  25
NUMBER OF MULTINODE ITERATIONS =  5
NUMBER OF MULTINODE ASCENT STEPS =  0
NUMBER OF REGULAR AUGMENTATIONS =  5
***********************************
```

```
s 88.
f 1 2 1
f 1 3 1
f 1 4 0
f 1 5 1
f 1 6 1
f 1 7 0
f 1 8 0
f 1 9 0
f 1 10 0
f 1 11 0
f 1 22 6
f 12 4 1
f 12 5 0
f 12 6 0
f 12 7 0
f 12 8 0
f 12 9 0
f 12 10 0
f 12 11 0
f 13 7 1
f 13 9 0
f 13 10 0
f 13 11 0
f 14 5 0
f 14 7 0
f 14 8 1
f 14 9 0
f 14 10 0
f 14 11 0
f 15 9 0
f 15 10 1
f 15 11 0
f 16 9 1
f 16 10 0
f 16 11 0
f 17 9 0
f 17 11 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 11 21 0
f 12 22 0
f 13 22 0
f 14 22 0
f 15 22 0
f 16 22 0
f 17 22 0
f 18 22 1
f 19 22 1
f 20 22 1
f 21 22 1
```

6 Questão 4: Interprete a solução ótima dada pelo software, e traduza essa solução num plano global, em que, para cada equipa, se listam os serviços que lhe foram atribuídos, com a explicitação dos tempos associados aos serviços e às deslocações; indique também o custo de operação de cada equipa e o custo total.

Construindo-se assim o conjunto de caminhos por onde as equipas passaram, podemos ter uma ideia do conjunto de emparelhamentos que o nosso grafo terá, ou seja, as ligações estabelecidas entre as equipas, através da análise da solução ótima. Observa-se o grafo criado com este conjunto:

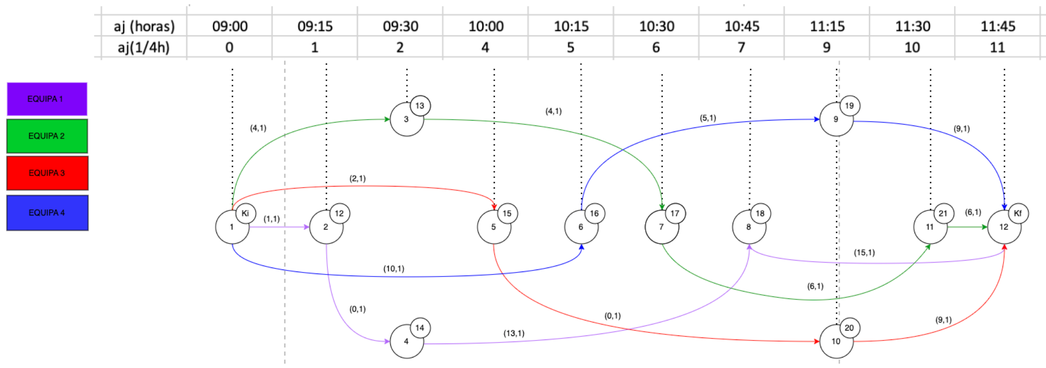


Figura 8: Grafo de caminhos finais percorridos pelas equipas.

Seguem-se agora as observações relevantes:

1. Através da linha do ficheiro Relax4 que simboliza o arco fictício entre Ki e Kf, f 1 22 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 4 (10 - 6) das 10 existentes. É este o número ótimo de equipas, que minimiza os custos totais da operação;

2. Através das linhas, observamos que todos os clientes são servidos;

3. O custo ótimo, fornecido pelo software, é de 88, 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: 1 + 0 + 13 + 15 + 1 U.M (custo fixo) = 30 U.M
(b) Equipa 2: 4 + 8 + 6 + 6 + 1 U.M (custo fixo) = 25 U.M
(c) Equipa 3: 2 + 0 + 9 + 1 U.M (custo fixo) = 12 U.M
(d) Equipa 4: 10 + 5 + 9 + 1 U.M (custo fixo) = 25 U.M
(e) Tendo-se, então, um custo ótimo de 88 U.M (29 + 24 + 11 + 24),ou seja, com o custo fixo, de 92 U.M(30 + 25 + 12 + 25).

4. 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.

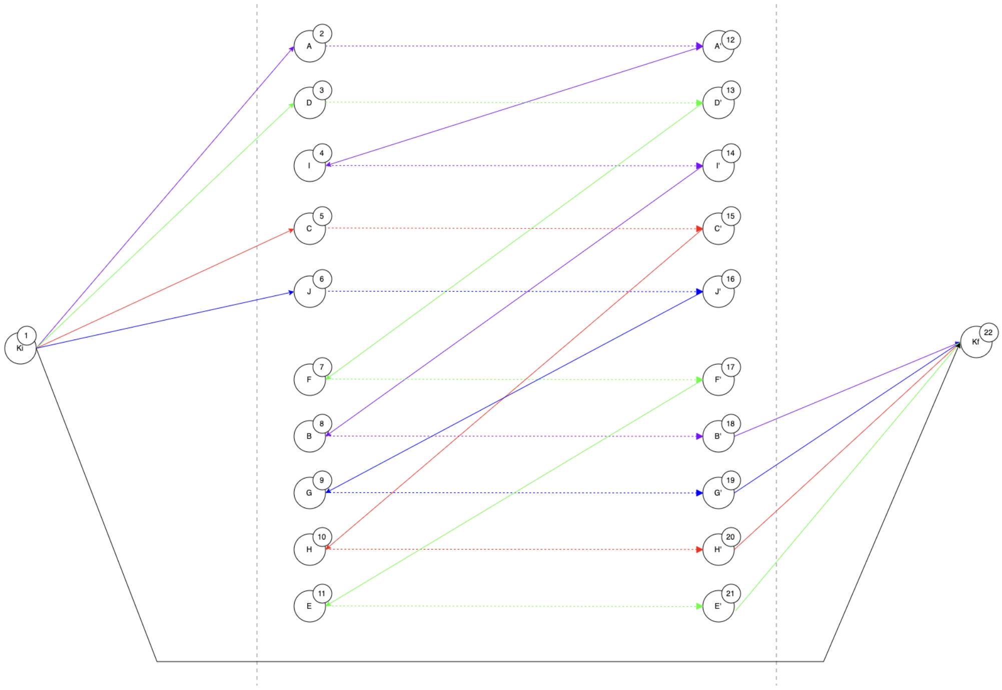


Figura 9: Grafo com os arcos fornecidos pelo *Relax4*.

Neste grafo, os arcos a cheio foram os fornecidos pelo software. Através desses arcos, respeitando as restrições temporais, que a numeração dos vértices já considera e sabendo que teríamos de passar num cliente apenas uma vez conseguimos desenhar os arcos a tracejado, formando um caminho para cada equipa. Uma vez que como ao vértice 1 se ligam 4 arcos, temos 4 equipas diferentes a operar (sem considerar as restantes que foram diretamente para o vértice 22). 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 5: Descreva os procedimentos usados para validar o modelo.

7.1 Validação do Modelo no LPSolve.

Para efetuar a validação do modelo, decidiu-se 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.

7.1.1 Input

```
//função objetivo
min: 1 x12 + 0 x142 + 5 x152 + 7 x162 + 10 x172 + 13 x182 + 7 x192 + 5 x202 + 5 x212 +
4 x13 + 8 x173 + 8 x193 + 8 x203 + 4 x213 +
9 x14 + 5 x154 + 10 x174 + 13 x184 + 7 x194 + 5 x204 + 5 x214 +
2 x15 + 6 x195 + 0 x205 + 6 x215 +
10 x16 + 5 x196 + 6 x206 + 7 x216 +
11 x17 + 5 x197 + 6 x217 +
15 x18 +
9 x19 +
9 x110 +
6 x111 +
1 x1222 + 4 x1322 + 9 x1422+ 2 x1522+ 10 x1622 + 11 x1722 +
15 x1822 + 9  x1922 + 9 x2022 + 6 x2122 + i;

i = x12 + x13 + x14 + x15 + x16 + x17 + x18 + x19 + x110 + x111;
f = x1222 + x1322 + x1422 + x1522 + x1622 + x1722 + x1822 + x1922 + x2022 + x2122;
i = f;

```

```
/* Entry  Edge */
vertice2: x12 + x142 + x152 + x162 +x172 + x182 + x192 + x202 + x212 = 1;
vertice3: x13 + x173 +x193 + x203 + x213 = 1;
vertice4: x14 + x154 + x174 + x184 + x194 + x204 + x214 = 1;
vertice5: x15 + x195 + x205 + x215 = 1;
vertice6: x16 + x196 + x206 + x216 = 1;
vertice7: x17 + x197 + x217 = 1;
vertice8: x18 = 1;
vertice9: x19 = 1;
vertice10: x110 = 1;
vertice11: x111 = 1;
```

```
/* Exit Edge*/
vertice12: x1222 = 1;
vertice13: x1322 = 1;
vertice14: x142 + x1422 = 1;
vertice15: x152 + x154 + x1522 = 1;
vertice16: x162 + x1622 = 1;
vertice17: x172 + x173 + x174 + x1722 = 1;
vertice18: x182 + x184 + x1822 = 1;
vertice19: x192 + x193 + x194 + x195 + x196 + x197 + x1922 = 1;
vertice20: x202 + x203 + x204 + x205 + x206 + x2022 = 1;
vertice21: x212 + x213 + x214 + x215 + x216 + x217 + x2122 = 1;
```

Figura 10: Ficheiro input LPSolve.

7.1.2 Output

```
mayagomes@Air-de-Maya aula4 % lp_solve io2.lp

Value of objective function: 92.00000000

Actual values of the variables:
x12                             0
x142                            1
x152                            0
x162                            0
x172                            0
x182                            0
x192                            0
x202                            0
x212                            0
x13                             0
x173                            1
x193                            0
x203                            0
x213                            0
x14                             0
x154                            0
x174                            0
x184                            1
x194                            0
x204                            0
x214                            0
x15                             0
x195                            0
x205                            1
x215                            0
x16                             0
x196                            1
x206                            0
x216                            0
x17                             0
x197                            0
x217                            1
x18                             1
x19                             1
x110                            1
x111                            1
x1222                           1
x1322                           1
x1422                           0
x1522                           1
x1622                           1
x1722                           0
x1822                           0
x1922                           0
x2022                           0
x2122                           0
i                               4
f                               4
```

Figura 11: Output LPSolve.

7.1.3 Análise

Conseguimos observar os seguintes percursos:

- 1 -> 8 -> 18 -> 4 -> 14 -> 2 -> 12 -> 22

- 1 -> 11 -> 21 -> 7 -> 17 -> 3 -> 13 -> 22
- 1 -> 110 -> 20 -> 5 -> 15 -> 22
- 1 -> 9 -> 19 -> 6 -> 16 -> 22

No entanto, estes percurso, à primeira vista, podem ser enganosos, pois fazem pensar que esse foi o caminho que a equipa fez, contudo, é o caminho reverso. Assim, com base no primeiro caminho temos que se a equipa visitou o vértice 8, significa que teve de sair do nodo 18; se saiu do 18, teve de ser visitado o vértice 4 (pois há o arco 18->4). 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 2 (pois há o arco 14->2). Tendo visitado o vértice 2, e tendo-se o arco 12->22, significa que a equipa, depois de visitar o vértice 2, regressou para a sede, tendo-se o caminho real seguinte:

- 1 -> 2 -> 4 -> 8 -> 22

Aplicando este raciocínio nos caminhos das 4 equipas obtemos os caminhos reias seguintes:

- 1 -> 2 -> 4 -> 8 -> 22
- 1 -> 3 -> 7 -> 11 -> 22
- 1 -> 5 -> 10 -> 22
- 1 -> 6 -> 9 -> 22

Estes caminhos foram, inclusive, os caminhos que o Relax4 forneceu, como se consegue visualizar no Grafo de caminhos finais percorridos pelas equipas feito após a análise do Relax4.

Por fim, repare-se que o parâmetro i e f refletem o número de equipas utilizadas (4) e que o custo ótimo fornecido é igual ao que foi calculado pelo output do Relax4 (88+4=92).

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 X e Kf como vértice X):

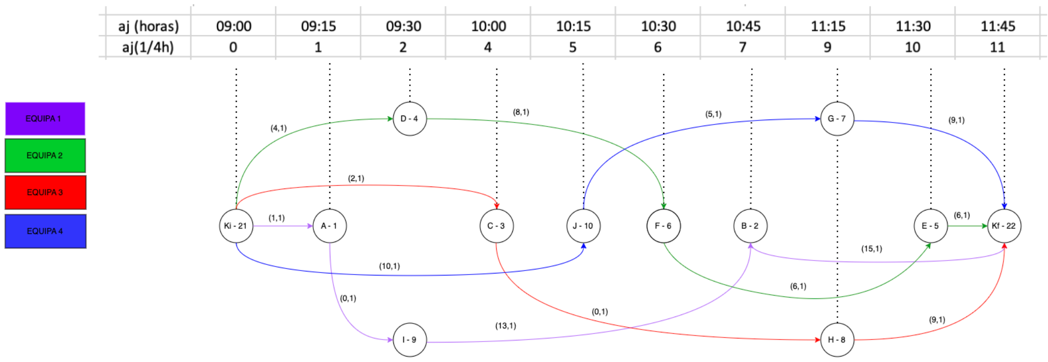


Figura 12: Grafo final com os respetivos percursos (vértices numerados de acordo com o enunciado).

7.2 Validação Intuitiva do Modelo

A validação do modelo pode ainda ser mais intuitiva. Analisando ambos os outputs de cada software e confrontando-os com aquilo que se pretende na vida real e verificando-se as propriedades abaixo, conseguimos validar o nosso modelo.

- Todos os caminhos do resultado são possíveis, isto é, o conjunto de caminhos fornecido é um conjunto de caminhos possível, tendo em conta as condições do enunciado;
- Esses caminhos têm em conta o tempo de deslocação, ou seja, 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 quatro equipas, não há vértices iguais e, dentro de uma mesma equipa, não existem vértices repetidos.

7.3 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:
 1 x12 + 0 x142 + 5 x152 + 7 x162 +10 x172
 + 13 x182 + 7 x192 + 5 x202 + 5 x212
 + 4 x13 + 8 x173 +8 x193 + 8 x203 + 4 x213 +
 9 x14 +5 x154 + 10 x174 + 13 x184 + 7 x194 + 5 x204 + 5 x214
 + 2 x15 + 6 x195 + 0 x205 + 6 x215 +
 10 x16 + 5 x196 + 6 x206 + 7 x216 +
 11 x17 + 5 x197 + 6 x217 +
 15 x18 +
 9 x19 +
 9 x110 +
 6 x111 +
 1 x1222 + 4 x1322 + 9 x1422+ 2 x1522+ 10 x1622 + 11 x1722 +
 15 x1822 + 9  x1922 + 9 x2022 + 6 x2122 +i;
```

Efetuando a substituição dos valores dados pelos LPSolver, vem:

0 * x142 + 8 * x173 + 13 * x184 + 0 * x205 + 5 * x196 + 6 * x217 + 15 * x18 + 9 *x19 + 9 * x110 + 6 * x111 + 1 * x1222 + 4 * x1322 + 2 * x1522 + 10 * x1622 + i = 8 + 13 + 5 + 6 + 15 + 9 + 9 + 6 + 1 + 4 + 2 + 10 + 4 = 92

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

Confrontando com as respetivas restrições, temos:

$i = x12 + x13 + x14 + x15 + x16 + x17 + x18 + x19 + x110 + x111 \equiv$
$i = 0 + 0 + 0 + 0 + 0 + 0 + 1 + 1 + 1 + 1 \equiv$
$i = 4$

$f = x1222 + x1322 + x1422 + x1522 + x1622 + x1722 + x1822 + x1922 + x2022 + x2122 \equiv$
$f = 1 + 1 + 0 + 1 + 1 + 0 + 0 + 0 + 0 \equiv$
$f = 4$

$i = f = 4 \equiv$ TRUE

vertice2: $x12 + x142 + x152 + x162 + x172 + x182 + x192 + x202 + x212 = 1 \equiv$
$0 + 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 1 \equiv$ TRUE
vertice3: $x13 + x173 + x193 + x203 + x213 = 1 \equiv$
$0 + 1 + 0 + 0 + 0 = 1 \equiv$TRUE

vertice4: $x14 + x154 + x174 + x184 + x194 + x204 + x214 = 1 \equiv$
$0 + 0 + 0 + 1 + 0 + 0 + 0 = 1 \equiv$ TRUE
vertice5: $x15 + x195 + x205 + x215 = 1 \equiv$
$0 + 0 + 1 + 0 = 1 \equiv$ TRUE
vertice6: $x16 + x196 + x206 + x216 = 1 \equiv$
$0 + 1 + 0 + 0 = 1 \equiv$ TRUE
vertice7: $x17 + x197 + x217 = 1 \equiv$
$0 + 0 + 1 = 1 \equiv$ TRUE
vertice8: $x18 = 1 \equiv$
$1 = 1 \equiv$ TRUE
vertice9: $x19 = 1 \equiv$
$1 = 1 \equiv$ TRUE
vertice10: $x110 = 1 \equiv$
$1 = 1 \equiv$ TRUE
vertice11: $x111 = 1 \equiv$
$1 = 1 \equiv$ TRUE
vertice12: $x1222 = 1 \equiv$
$1 = 1 \equiv$ TRUE
vertice13: $x1322 = 1 \equiv$
$1 = 1 \equiv$ TRUE
vertice14: $x142 + x1422 = 1 \equiv$
$1 + 0 = 1 \equiv$ TRUE
vertice15: $x152 + x154 + x1522 = 1 \equiv$
$0 + 0 + 1 = 1 \equiv$TRUE
vertice16: $x162 + x1622 = 1 \equiv$
$0 + 1 = 1 \equiv$ TRUE
vertice17: $x172 + x173 + x174 + x1722 = 1 \equiv$
$0 + 1 + 0 + 0 = 1 \equiv$ TRUE
vertice18: $x182 + x184 + x1822 = 1 \equiv$
$0 + 1 + 0 = 1 \equiv$ TRUE
vertice19: $x192 + x193 + x194 + x195 + x196 + x197 + x1922 = 1 \equiv$
$0 + 0 + 0 + 0 + 1 + 0 + 0 = 1 \equiv$TRUE
vertice20: $x202 + x203 + x204 + x205 + x206 + x2022 = 1 \equiv$
$0 + 0 + 0 + 1 + 0 + 0 = 1 \equiv$ TRUE
vertice21: $x212 + x213 + x214 + x215 + x216 + x217 + x2122 = 1 \equiv$
$0 + 0 + 0 + 0 + 0 + 1 + 0 = 1 \equiv$TRUE

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

8 Aplicação na realidade

Após a resolução, iremos efetuar com pensamento crítico a aplicação da solução que encontramos na realidade.

Alguns aspetos importantes que vemos concretizados nesta resolução são:

- Todos os clientes serem atendidos;
- Não haver mais que uma intervenção no mesmo cliente;
- Não haver equipas que ficam a meio do percurso, isto é, todas as equipas que saem da sede regressam à mesma;

Considerando a configuração do grafo inicial, conseguimos perceber que existem no máximo dois clientes por horário e que existe um total de 10 clientes. Desta forma, a solução mínima é de 2 equipas e a solução máxima seria de 10 equipas. Assim sendo, tendo em conta que a nossa solução ótima é igual a 4 equipas, trata-se de uma solução intermédia, o que parece correto conforme o cenário real.

Referências

[1] Brandão, F., Pedroso, J.P., Pedroso, P.: Solving bin packing related problems using an arc flow formulation, `http://www.dcc.fc.up.pt/Pubs/`

[2] Gradišar, D., Glavan, M.: Material requirements planning using variable-sized bin-packing problem formulation with due date and grouping constraints. Processes **8**, 1–16 (10 2020). https://doi.org/10.3390/pr8101246

[3] Val, J.M., Carvalho, E.D.: Lp models for bin packing and cutting stock problems, `www.elsevier.com/locate/dsw`