

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA
UNIVERSIDADE DO MINHO

Modelos Determinísticos de Investigação Operacional

December 9, 2018

José Pedro Pinto - A80741
Pedro Lima - A80785
Alexandre Pacheco - A80760
Mariana Fernandes - A81728

Índice

Introdução	2
Questão 1	4
a Modelo Primal	4
a.1 Código	6
b Modelo Dual	7
b.1 Código	9
c Resolução do Modelo Primal	9
d Resolução do Modelo Dual	11
Questão 2	13
a Proteção de uma Célula com Recursos Limitados	13
b Resolução do Modelo	15
b.1 Código	15
b.2 Resultados	16
c Variação dos Resultados do Modelo segundo os Parâmetros	17
Questão 3	19
a Minimização da Área Ardida	19
b Resolução do Modelo	21
b.1 Código	21
b.2 Resultados	22
c Previsão da Área Ardida em Função do Tempo	23
Literatura Consultada	24

Introdução

O trabalho proposto aos alunos envolvia a modelação de problemas de investigação operacional, bem como a otimização de soluções para os mesmos. No seguimento desta tarefa, este relatório explica os pensamentos que nos levaram às soluções dos diferentes problemas e a implementação dos mesmos nas ferramentas aconselhadas pelos docentes (IBM ILOG CPLEX Optimization Studio, ou OPL).

Contextualizando as perguntas que se seguem, a grelha de nodos e arcos que o nosso grupo utilizou para responder às perguntas propostas no trabalho foi a seguinte.

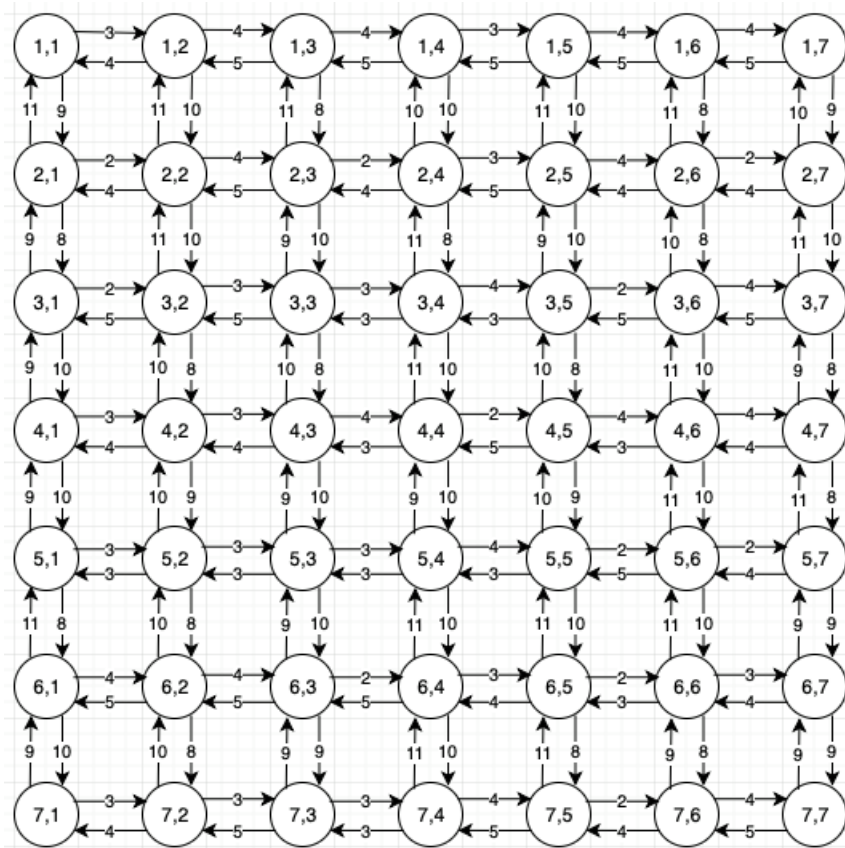


Figure 1.1: Mapa de nodos, arcos e respetivos pesos

Quanto às outras constantes, são as seguintes:

- $\Delta = 8$; (Constante de Propagação)
- $\beta = 8$; (Limite de Recursos)
- $g = 13$; (Constante de Proteção)

Questão 1

a Modelo Primal

Modelo PL que permite determinar o instante de chegada do fogo a cada nodo, assumindo que o nodo de ignição é o nodo 1, para uma rede com $n \times n$ nodos.

Parâmetros:

n - Número de linhas/colunas.

$Custos_{(i,j)(k,l)}$ - Custo do arco com origem no nodo (i,j) e destino no nodo (k,l) .

$i_inicio = 1$

$j_inicio = 1$

Variável de Decisão:

$X_{(i,j)(k,l)}$ - Número de caminhos em que o arco $(i,j)(k,l)$ é usado.

Função Objetivo:

$$\text{Min } Z = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n Custos_{(i,j)(k,l)} \times X_{(i,j)(k,l)}$$

Como o tempo de ignição de um nodo é calculado a partir da soma dos custos dos arcos que fazem parte do caminho mais curto que liga esse nodo à origem, então minimizar a soma das multiplicações todos os arcos usados pelos seus respectivos custos garante que os arcos "escolhidos" pelo motor de otimização são aqueles que representam uma árvore de caminhos mais curtos nesta rede.

Sujeito a:

$$\sum_{i=1}^n \sum_{j=1}^n X_{(i_inicio,j_inicio)(i,j)} - X_{(i,j)(i_inicio,j_inicio)} = n^2 - 1$$

$$\sum_{k=1}^n \sum_{l=1}^n X_{(i,j)(k,l)} - X_{(k,l)(i,j)} = -1, \forall i, j \in \{1..n\} : (i,j) \neq (i_{inicial}, j_{inicial})$$

$$X_{(i,j)(k,l)} \geq 0, \forall i, j \in \{1..n\}$$

A primeira restrição implica que do nodo origem do fogo saem $n^2 - 1$ caminhos (uma vez que não se usam arcos com destino no nodo de origem, porque não faria sentido numa árvore de caminhos mais curtos). O que é verdade, visto que há n^2 nodos, sendo a própria origem um deles (e não há um caminho que comece e acabe na origem).

A segunda restrição obriga a que o número de caminhos que "saem" de um nodo (ou seja, que utilizam arcos com origem nele) são menos um do que os que "entram" nesse mesmo nodo (ou seja, que utilizam arcos com destino nele). Isto porque um desses caminhos, naturalmente, é o caminho mais curto para esse nodo, e, portanto, termina nele.

A terceira condição obriga a que não haja valores negativos nas variáveis de decisão (dado que o motor de otimização preencheria as variáveis de decisão com valores infinitamente negativos na tentativa de minimizar a função objetivo, proporcional às mesmas).

a.1 Código

```
//Parâmetros sobre a dimensão do problema
int n = 7;
int a[i in 1..n] = i;

//Início do fogo
int i_inicio = 1;
int j_inicio = 1;

//Parâmetros sobre os custos dos arcos
int Custos[1..n][1..n][1..n][1..n];

//Variáveis de Decisão
dvar float+ X[1..n][1..n][1..n][1..n];                                     //Número de caminhos
                                                                                   //em que o arco ijkl
                                                                                   //é utilizado

//Função Objetivo
minimize sum(i in 1..n, j in 1..n, k in 1..n, l in 1..n) Custos[i][j][k][l]
* X[i][j][k][l];

//Restrições
subject to{
    sum(i in 1..n, j in 1..n)
        (X[i_inicio][j_inicio][i][j] - X[i][j][i_inicio][j_inicio]) == n^2 -1;
                                                                                   //Do nodo 1,1 saem
                                                                                   //48 caminhos

    forall(i in 1..n, j in 1..n: i != i_inicio || j != j_inicio)
        (sum(k in 1..n, l in 1..n)(X[i][j][k][l] - X[k][l][i][j])) == -1);
                                                                                   //De um nodo sai menos
                                                                                   //um caminho do que a
                                                                                   //ele chegam (visto que
                                                                                   //o caminho mais curto
                                                                                   //para esse nodo
                                                                                   //termina nele)
}
```

b Modelo Dual

Modelo PL que determina o instante de chegada do fogo a cada nodo, assumindo que o nodo de ignição é o nodo 1, para uma rede com $n \times n$ nodos.

Este modelo é dual do da alínea anterior, e passamos a explicar o raciocínio para o obter. Primeiro olhámos para as restrições do primal. Se olharmos para o quantificador da segunda restrição, podemos perceber que na verdade esta corresponde a $n \times n - 1$ restrições. Com a primeira, são então n^2 restrições. Para cada uma delas, consideremos uma variável $y_{i,j}$. Consideremos também o número de parcelas no somatório da função objetivo do primal. Tem n^4 parcelas, pelo que o dual terá esse número de restrições.

y_{i_inicio, j_inicio} corresponde à primeira restrição (tendo em conta o quantificador da segunda). E, pensando na parcela que na função objetivo do primal corresponde a $k = i_inicio$ & $j = j_inicio$ (que é 0), chegamos à restrição $y_{i_inicio, j_inicio} = 0$.

Pensando no resto dos $y_{i,j}$, sabemos que a parcela correspondente no somatório da função objetivo do primal representa o custo de um arco. Olhando novamente para a segunda restrição do modelo, podemos reparar que devido à inversão dos índices na subtração $X_{(i,j)(k,l)} - X_{(k,l)(i,j)}$, cada variável de decisão apresenta 2 coeficientes, pelo que as restrições do dual terão presentes duas variáveis de decisão, que podem ser $y_{i,j}$ e $y_{k,l}$. Assim podemos chegar à restrição $y_{k,l} - y_{i,j} \leq Custos_{(i,j)(k,l)}$. Para evidenciar que a $y_{i,j}$ corresponde ao tempo de ignição do nodo i, j , escrevamos esta restrição na forma $y_{k,l} \leq y_{i,j} + Custos_{(i,j)(k,l)}$.

O dual tem de ser um modelo de maximização, de onde resulta a função objetivo $max Z = \sum_{i=1}^n \sum_{j=1}^n y_{(i,j)}$. Torna-se evidente que estas variáveis de decisão correspondem aos tempos de ignição, pelo que se encontra abaixo descrito mais formalmente o modelo dual.

Parâmetros:

n - Número de linhas/colunas.

$Custos_{(i,j)(k,l)}$ - Custo do arco com origem no nodo (i,j) e destino no nodo (k,l) .

$i_inicio = 1$

$j_inicio = 1$

Variável de Decisão:

$T_{(i,j)}$ - Tempo necessário para o fogo atingir o nodo (i,j) .

Função Objetivo:

$$\text{Max } Z = \sum_{i=1}^n \sum_{j=1}^n T_{(i,j)}$$

Como as restrições já obrigam (devido à sua conjunção) a que o único caminho possível seja o mais curto, temos de maximizar (contra intuitivamente) a soma dos tempos de ignição, de modo que o motor de otimização do OPL faz com que as variáveis de decisão convirjam para esses valores.

Sujeito a:

$$T_{(i_inicio,j_inicio)} = 0$$

$$T_{(k,l)} \leq T_{(i,j)} + Custos_{(i,j),(k,l)}, \forall i,j,k,l \in \{1..n\} : (k,l) \neq (i_inicio,j_inicio)$$

A simplicidade destas restrições tornam a sua interpretação trivial. O tempo de ignição da origem do fogo é 0. O tempo de ignição do nodo (k,j) é menor ou igual a **todas** (e, portanto, apenas ao menor, devido à conjunção das condições respeitantes a este nodo) as somas dos custos dos arcos com origem nos outros nodos todos e os respetivos tempos de ignição (dos nodos de origem desses arcos).

b.1 Código

```
//Parâmetros sobre a dimensão do problema
int n = 7;
int a[i in 1..n] = i;
int Custos[1..n][1..n][1..n][1..n];                                     //Matriz contendo todos
                                                                                   //os arcos

//Início ou fim do fogo
int i_inicio = 1;
int j_inicio = 1;
//Variáveis de Decisão
dvar float+ T[1..n][1..n];                                             //Tempo de ignição
                                                                                   //de um nodo ij

//Função Objetivo
maximize sum(i in 1..n, j in 1..n) T[i][j];

//Restrições
subject to{
    T[i_inicio][j_inicio] == 0;
    forall(i in 1..n, j in 1..n, k in 1..n, l in 1..n: k != i_inicio || l !=
j_inicio)
        T[k][l] <= T[i][j] + Custos[i][j][k][l];
}
```

c Resolução do Modelo Primal

Utilizando o IBM ILOG CPLEX Optimization Studio, podemos resolver o modelo primal e obter os valores da matriz $X_{(i,j)(k,l)}$. Como essa matriz tem 7^4 entradas, por simplicidade apresentamos apenas os valores diferentes de 0 e uma representação gráfica da árvore de caminhos mais curtos determinada.

i	j	k	l	X(i,j)(k,l)					
1	1	1	2	1	32				
2	1	3	1	30					
3	1	3	2	25					
1	1	1	2	16					
1	2	1	3	15					
3	2	4	2	14					
4	2	5	2	13					
5	2	6	2	10					
3	2	3	3	10					
1	3	2	3	10					
3	3	4	3	9					
2	3	2	4	9					
4	3	4	4	8					
4	4	4	5	7					
6	2	6	3	6					
6	3	6	4	5					
4	5	5	5	5					
2	4	3	4	5					
6	4	6	5	4					
5	5	5	6	4					
3	4	3	5	4					
3	1	4	1	4					
1	3	1	4	4					
6	2	7	2	3					
5	6	5	7	3					
4	1	5	1	3					
3	5	3	6	3					
2	4	2	5	3					
1	4	1	5	3					
7	2	7	3	2					
6	5	6	6	2					
5	7	6	7	2					
5	2	5	3	2					
5	1	6	1	2					
3	6	3	7	2					
2	5	2	6	2					
1	5	1	6	2					
7	3	7	4	1					
6	7	7	7	1					
6	6	7	6	1					
6	5	7	5	1					
6	1	7	1	1					
5	3	5	4	1					
4	5	4	6	1					
3	7	4	7	1					
2	6	2	7	1					
2	1	2	2	1					
1	6	1	7	1					

Nesta tabela, o valor da coluna à direita representa o número de caminhos em que o arco $X(i,j)(k,l)$ é utilizado. Ou seja, as duas primeiras colunas correspondem às coordenadas do nodo origem do arco e as 3ª e 4ª colunas representam as coordenadas do nodo destino do arco.

Figure 1.2: Número de caminhos em que cada arco é utilizado

A figura seguinte ilustra esse resultado recorrendo ao esquema já apresentado.

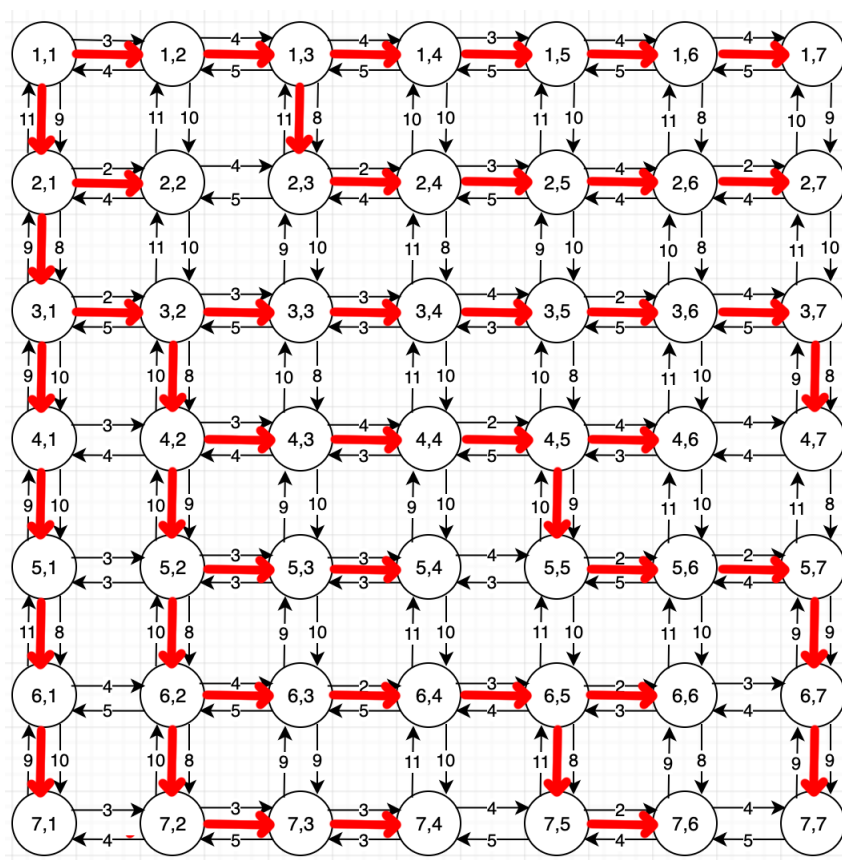


Figure 1.3: Caminhos mais curtos para cada célula

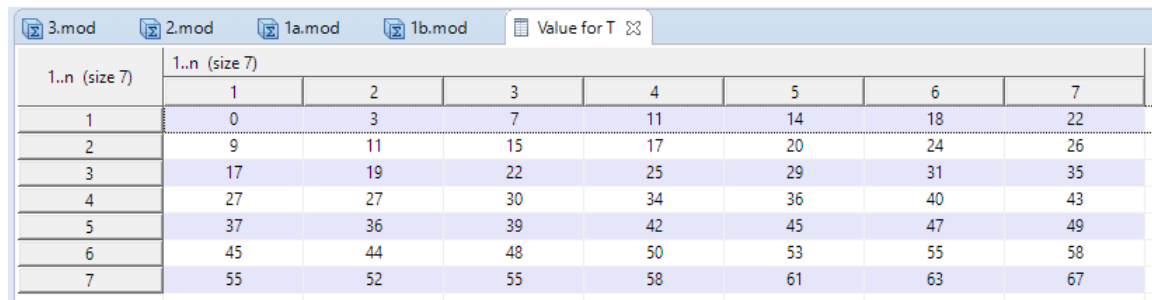
Por fim, podemos interpretar estes valores (ou utiliza-los automaticamente no OPL, gerando uma matriz auxiliar) e podemos obter os tempos de ignição de cada célula.

i,j	1	2	3	4	5	6	7
1	0	3	7	11	14	18	22
2	9	11	15	17	20	24	26
3	17	19	22	25	29	31	35
4	27	27	30	34	36	40	43
5	37	36	39	42	45	47	49
6	45	44	48	50	53	55	58
7	55	52	55	58	61	63	67

Figure 1.4: Tempos de chegada do fogo

d Resolução do Modelo Dual

Novamente, recorrendo ao OPL, resolvemos o modelo dual. Como as suas variáveis de decisão são os próprios tempos de ignição, obtemos logo a seguinte tabela.



1..n (size 7)	1	2	3	4	5	6	7
1	0	3	7	11	14	18	22
2	9	11	15	17	20	24	26
3	17	19	22	25	29	31	35
4	27	27	30	34	36	40	43
5	37	36	39	42	45	47	49
6	45	44	48	50	53	55	58
7	55	52	55	58	61	63	67

Figure 1.5: Resultado do modelo dual

Como podemos verificar, estes valores são coerentes com os que obtivemos a partir do modelo primal.

Questão 2

a Proteção de uma Célula com Recursos Limitados

Para esta pergunta, o desafio era, mantendo toda a contextualização anterior, otimizar a alocação de recursos em células de modo a proteger uma determinada célula. Optámos por modificar o modelo dual da questão 1 para atingir este objetivo.

Parâmetros:

n - Número de linhas/colunas.

$Custos_{(i,j)(k,l)}$ - Custo do arco com origem no nodo (i,j) e destino no nodo (k,l) .

i_{inicio} - Linha da célula de ignição.

j_{inicio} - Coluna da célula de ignição.

i_{fim} - Linha da célula a proteger.

j_{fim} - Coluna da célula a proteger.

Variáveis de Decisão:

$R_{(i,j)}$ - Recurso utilizado no nodo (i,j) para retardar a propagação do fogo.

$T_{(i,j)}$ - Tempo necessário para o fogo atingir o nodo (i,j) que se deseja proteger.

Função Objetivo:

$$\text{Max } Z = T_{(i_{fim}, j_{fim})}$$

Aqui, a função objetivo é maximizar o tempo de ignição da célula a proteger. Esta função objetivo funciona porque como agora o tempo de ignição de uma célula é dependente do facto de se ter utilizado ou não um recurso na célula anterior, podemos maximizar o tempo de ignição da célula a proteger de modo a chegar à solução ótima de alocação de recursos. O facto de não maximizarmos a soma dos tempos de ignição de todos os nodos (como no

dual da questão 1) não influencia esta solução, dado que isso está a ser feito indiretamente quando maximizamos o tempo de ignição da célula a proteger, uma vez que este depende de todos os arcos que podem constituir o caminho mais curto desde a origem do fogo até esta célula.

Sujeito a:

$$T_{(i_inicio, j_inicio)} = 0$$

$$T_{(k,l)} \leq T_{(i,j)} + Custos_{(i,j),(k,l)} + R_{(i,j)} \times \Delta, \forall i,j,k,l \in \{1..n\} : (k,l) \neq (i_inicio, j_inicio)$$

$$\sum_{i=1}^n \sum_{j=1}^n R_{(i,j)} \leq \beta$$

A primeira restrição estabelece que o tempo de ignição da origem do fogo é 0.

A segunda restrição é a evolução óbvia de uma das restrições do modelo dual da questão 1, face à implementação da existência dos recursos de combate aos incêndios neste problema. Ora, como descrito no enunciado, modelamos que o tempo de ignição de uma célula será menor ou igual a **todos** (novamente, e portanto só ao menor, devido à conjunção das condições referentes a esse mesmo nodo) os tempos de ignição de todos os nodos mais o custo do respetivo nodo e a constante de retardamento (Δ) multiplicada pelos recursos (1 ou 0) existentes nos respetivos nodos de origem (dos arcos).

Há que notar que, desta maneira, se um recurso for colocado numa célula que fazia parte do caminho mais curto para a célula a proteger, o fogo pode muito bem chegar mais cedo que o esperado, se outro caminho passar a ser o mais curto. Aqui se nota a complexidade deste problema e a importância de saber modelar estes problemas e recorrer a ferramentas como o OPL para os resolver.

A terceira restrição limita o número total de recursos que podem ser utilizados nas células.

b Resolução do Modelo

b.1 Código

```
//Parâmetros sobre a dimensão do problema
int n = 7;
int Custos[1..n][1..n][1..n][1..n];           //Matriz contendo
                                                //todos os arcos

//Início ou fim do fogo
int i_inicio = 1;
int j_inicio = 1;
int i_fim = 7;
int j_fim = 7;

//Número de recursos b
int b = 8;
//Constante de retardamento do fogo por recurso (delta)
int delta = 8;

//Variáveis de Decisão
dvar float+ T[1..n][1..n];                   //Tempo de ignição
                                                //de um nodo ij
dvar boolean R[1..n][1..n];                  //Recursos utilizados
                                                //no nodo ij

//Função Objetivo
maximize T[i_fim][j_fim];

//Restrições
subject to{
    T[i_inicio][j_inicio] == 0;
    forall(i in 1..n, j in 1..n, k in 1..n, l in 1..n: k != i_inicio || l !=
j_inicio)
        T[k][l] <= T[i][j] + Custos[i][j][k][l] + R[i][j]*delta;
    sum(i in 1..n, j in 1..n) R[i][j] <= b;    //Só há b recursos
                                                //disponíveis
}
```


b.2 Resultados

Recorrendo ao OPL e resolvendo o modelo explicado acima, obtemos as seguintes tabelas para $T_{i,j}$ e para $R_{i,j}$, respetivamente.

i,j	1	2	3	4	5	6	7
1	0	11	23	32	35	39	38
2	17	27	39	41	44	47	47
3	33	43	46	49	53	55	57
4	51	51	54	58	60	61	65
5	61	60	62	65	69	71	73
6	64	68	72	74	77	79	82
7	73	76	79	82	85	87	99

Figure 2.1: Tempos de chegada do fogo

i,j	1	2	3	4	5	6	7
1	1	1	1	0	0	0	0
2	1	1	0	0	0	0	0
3	1	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	1
7	0	0	0	0	0	1	0

Figure 2.2: Células Tratadas

Como podemos ver na primeira tabela, os tempos de ignição de todas as células, tirando a origem do incêndio, aumentaram. Isto deveu-se ao facto de logo na origem se ter aplicado recursos de combate ao incêndio. A tabela com os tempos revela valores coerentes entre si, no entanto é na segunda tabela que podemos fazer as observações mais interessantes sobre a solução do problema.

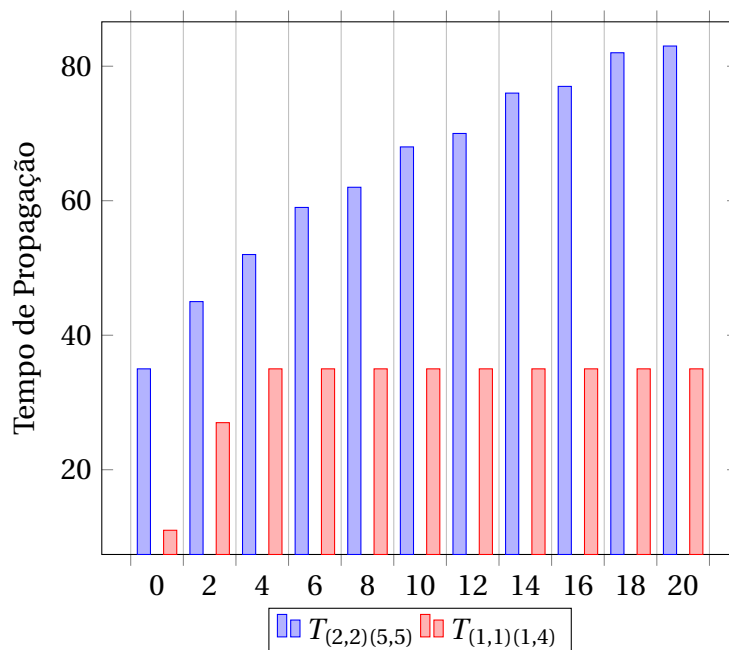
A resolução deste modelo resultou numa distribuição que concentra os recursos de combate ao incêndio em duas zonas: a da origem do incêndio e a da célula a proteger. Naturalmente, isto acontece porque inevitavelmente o incêndio terá de ser propagado nas zonas mais próximas da sua origem, e para chegar à área a proteger tem de passar pelas células que a rodeiam.

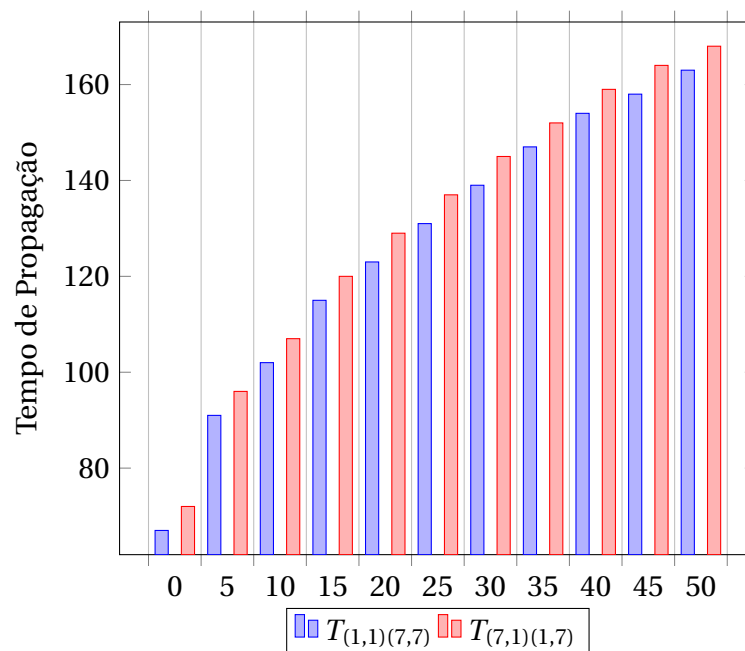
Na verdade este é um cenário em que a aplicação de recursos consegue ser mais eficaz, uma vez que tanto a origem do fogo como a célula a proteger se encontram em cantos do

mapa, e portanto apenas é necessário retardar o avanço do incêndio em duas frentes.

Há que notar que, resolvendo estes problemas no OPL, quando o motor de otimização encontra a solução para o modelo, este não efetua mais cálculos. Posto isto, as soluções, mesmo sendo corretas, não obrigam a que todas as variáveis de decisão, por exemplo, assumam os valores dos caminhos mais curtos, ou dos tempos de ignição corretos, pelo que há que ter atenção ao interpretar os resultados devolvidos pelo software.

c Variação dos Resultados do Modelo segundo os Parâmetros





Nos gráficos de barras abaixo representados observa-se a relação entre recursos disponibilizados ao retardamento dos incêndios e o tempo de propagação do fogo até ao nodo que se pretende proteger.

No primeiro está representado a propagação nas duas diagonais de células de canto, de (1, 1) a (7, 7) e de (7, 1) a (1, 7). Nas extremidades horizontais do gráfico pode-se observar o tempo mínimo de propagação de uma célula à outra (inexistência de recursos) e o tempo máximo (recursos "ilimitados" originando assim o maior tempo possível, mais recursos não faria diferença a nível de tempo). Por fim, torna-se possível observar a existência de uma proporção quase direta entre a disponibilidade de recursos e o aumento de tempo.

O segundo permite uma análise mais interessante sobre os resultados obtidos. Este representa a propagação entre os nodos (2, 2) a (5, 5) e de (1, 1) a (1, 4). Rapidamente, torna-se evidente que a partir de um reduzido número de recursos, o tempo de propagação estabiliza mesmo que se disponibilize uma maior quantidade de recursos à proteção. Tal acontece, uma vez que são nodos relativamente próximos, logo mantém-se como caminho mais eficiente entre estes o uso de nodos com recursos do que os contornar (o tempo para tal é, por base, maior e no caso de haver mais recursos seria também atrasado).

Podemos assim concluir, inevitavelmente, que o número de recursos necessários para atingir o tempo máximo entre dois nodos, é dependente da distância entre estes, não com pensando assim utilizar cegamente em todos as células.

Questão 3

Nesta questão, era pretendido, a partir das probabilidades de um incêndio ter origem num certo nodo, chegar à distribuição ótima *a priori* de recursos de combate aos incêndios de maneira a minimizar a área ardida.

a Minimização da Área Ardida

Para construir este modelo, baseámo-nos no da questão 2, dado que a definição que pretendemos para os tempos de ignição é semelhante. No entanto, há uma grande mudança a fazer relativamente às variáveis de decisão. Primeiro, tivemos a necessidade de transformar a variável $T_{(i,j)}$ numa variável $T_{(i,j)(k,l)}$, uma vez que agora temos de saber quanto tempo leva o fogo a chegar de uma célula a outra (não há uma origem do fogo definida).

Ora, sabemos que se o fogo começar numa célula (i, j) , $T_{(i,j)(k,l)}$ é o tempo de ignição da célula (k, l) . Daqui podemos retirar que se $T_{(i,j)(k,l)}$ for maior que um determinado tempo, então a célula não arderá nesse incêndio (12, no caso especificado pelo enunciado). Assim, utilizámos uma variável de decisão $Arde_{(i,j)(k,l)}$ que toma o valor 1 quando a célula (k, j) arde num incêndio começado na célula (i, j) , e toma o valor 0 caso contrário.

A variável de decisão $R_{i,j}$ (recursos aplicados na célula (i, j)) mantém-se, e influencia as outras duas de maneira que é (outra vez) precisamente a distribuição ótima dos seus valores que constitui a resposta ao problema.

O objetivo do problema é minimizar a área ardida, em função das probabilidades de um incêndio começar nas diferentes células. Assim, pensámos na seguinte função objetivo:

$$\text{Min } Z = \sum_{i=1}^n \sum_{j=1}^n \text{Probs}_{i,j} * Arde_{(i,j)(k,l)}$$

Como não sabemos onde o fogo começa, mas sim a probabilidade de cada célula ser a origem do fogo, estimamos a área ardida somando (para todos os nodos) a probabilidade do nodo originar o fogo e o número de células que ardem nessa situação. Este raciocínio originou o modelo a seguir descrito.

Parâmetros:

n - Número de linhas/colunas.

$Custos_{(i,j)(k,l)}$ - Custo do arco com origem no nodo (i,j) e destino no nodo (k,l) .

$Probs_{(i,j)}$ - Probabilidade de o fogo começar no nodo (i,j) .

$time$ - Instante para além do qual as células estão protegidas do incêndio.

Variáveis de Decisão:

$R_{(i,j)}$ - Recurso utilizado no nodo (i,j) para retardar a propagação do fogo.

$T_{(i,j)(k,l)}$ - Tempo necessário para o fogo com origem no nodo i,j atingir o nodo k,l .

$Arde_{(i,j)(k,l)}$ - Variável que toma o valor 1 se o fogo originado em (i,j) atingir o nodo (k,l) , e toma o valor 0 caso contrário.

Função Objetivo:

$$\text{Min } Z = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n Probs_{(i,j)} * Arde_{(i,j)(k,l)}$$

Como já referido, este somatório representa uma estimativa da área que arde tendo em conta as probabilidades de um incêndio originar em cada célula.

Sujeito a:

$$T_{(i,j)(i,j)} = 0, \forall i,j \in 1..n$$

$$T_{(i,j)(k,l)} \leq T_{(i,j)(m,n)} + Custos_{(m,n)(k,l)} + R_{(m,n)} \times \Delta, \forall i,j,k,l,m,n \in \{1..n\}$$

$$Arde_{(i,j)(k,l)} \leq (time - T_{(i,j)(k,l)}) / time, \forall i,j,k,l,m,n \in \{1..n\}$$

$$\sum_{i=1}^n \sum_{j=1}^n R_{(i,j)} \leq \beta$$

A primeira restrição estabelece que o tempo de ignição de uma célula caso o fogo tenha origem nela é 0.

A segunda restrição é parecida com uma utilizada nos anteriores modelos, no entanto em vez de representar tempos de ignição respeitantes a um incêndio, representa os tempos de ignição para as diferentes origens possíveis do incêndio (todos os nodos).

A terceira restrição obriga a que a variável de decisão $Arde_{(i,j)(k,l)}$ assuma o valor 1 se uma célula (k,l) arde num incêndio iniciado em (i,j) , e o valor 0 caso contrário.

A quarta restrição, mais uma vez, limita os recursos disponíveis.

b Resolução do Modelo

b.1 Código

```
//Parâmetros sobre a dimensão do problema
int n = 7;
int Custos[1..n][1..n][1..n][1..n];
float Probs[i in 1..n][j in 1..n] = (14 - i - j)/14;
//Número de recursos b
int b = 8;
//Constante de retardamento do fogo por recurso (delta)
int delta = 8;
//Intervalo de tempo onde minimizar área ardida
float time = 12;

//Variáveis de Decisão
dvar float+ T[1..n][1..n][1..n][1..n];      //Tempo de um ij
dvar boolean Arde[1..n][1..n][1..n][1..n]; //para um kl          dvar
boolean R[1..n][1..n];                      //Recursos utilizados
                                              //no nodo ij

//Função Objetivo
minimize sum(i in 1..n, j in 1..n) P[i][j];

//Restrições
subject to{
  forall(i in 1..n, j in 1..n) T[i][j][i][j] == 0;
  forall(i in 1..n, j in 1..n, k in 1..n, l in 1..n, m in 1..n, n in 1..n)
    T[i][j][k][l] <= T[i][j][m][n] + Custos[m][n][k][l] + R[m][n]*delta;
  forall(i in 1..n, j in 1..n, k in 1..n, l in 1..n)
    Arde[i][j][k][l] <= (time - T[i][j][k][l])/time;
  sum(i in 1..n, j in 1..n) R[i][j] <= b;
}
```

b.2 Resultados

Para as circunstâncias referidas no enunciado, ou seja, sendo 12 a constante de tempo a partir do qual as células se encontram protegidas do fogo, recorreremos ao OPL para determinar a solução ótima do problema.

A seguinte tabela é a resposta ao problema, correspondendo à alocação ótima de recursos de combate aos incêndios.

i,j	1	2	3	4	5	6	7
1	0	0	1	0	0	0	0
2	1	0	1	0	0	0	0
3	0	1	0	1	0	0	0
4	0	0	1	0	0	0	0
5	0	0	1	0	0	0	0
6	0	0	1	0	0	0	0
7	0	0	0	0	0	0	0

Figure 3.1: Distribuição de recursos pelas células

Para interpretar estes resultados é mais útil perceber de que maneira influenciam a área ardida. Para isto, nós recorreremos a variáveis auxiliares no OPL que nos permitiram registar o número de áreas afetadas nos incêndios originados nas diferentes células.

i,j	1	2	3	4	5	6	7
1	4	4 •1		6	6	6	4
2 •2		5 •2		7	7	6	5
3	4 •2		5 •2		6	6	5
4	6	5 •1		8	8	7	5
5	6	6 •3		8	7	8	5
6	5	6 •2		8	8	7	6
7	5	6	7	7	7	5	4
			<= 2 células atingidas				
			<= 4 células atingidas				
			<= 6 células atingidas				
			<= 8 células atingidas				
		•	Recurso utilizado				

Figure 3.2: Número de células afetadas nos incêndios com diferentes origens

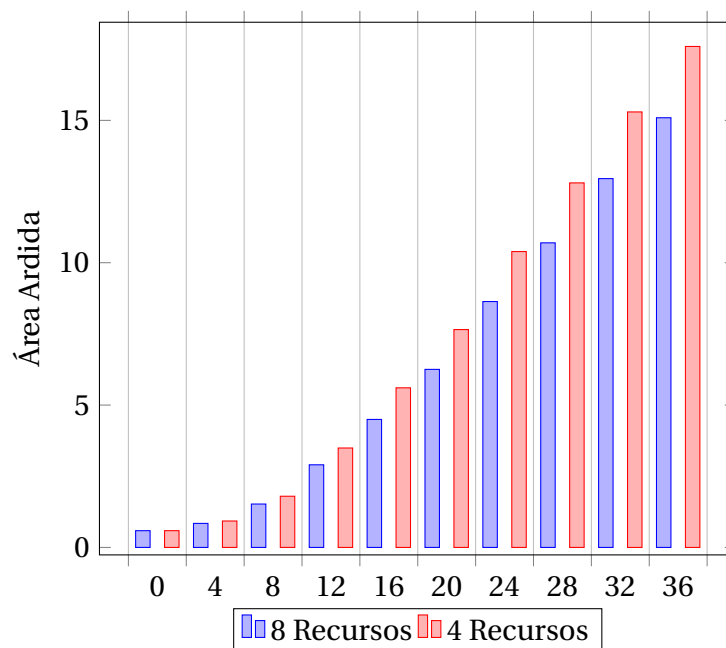
Uma conclusão que logo se destaca é o facto de o número de células afetadas nas zonas onde foram usados recursos ser de maneira geral bastante menor. Uma observação (apesar de não tão direta) mais interessante é o facto de a concentração dos recursos se situar nas zonas em que a probabilidade de se iniciarem fogos é maior. Isto faz com que, como pretendido, a área ardida seja minimizada uma vez que os fogos mais frequentes se alastrarão

menos, enquanto os fogos em zonas desprotegidas são os menos frequentes.

Novamente, como referimos na questão **2 b)** há que ter atenção ao interpretar as soluções devolvidas pelo software, uma vez que este (neste caso) apenas otimiza os valores de $R_{(i,j)}$ (devido ao peso destes na função objetivo, e uma vez que são a resposta propriamente dita ao problema).

c Previsão da Área Ardida em Função do Tempo

Para prever a área ardida em função do tempo para o qual uma célula se encontra protegida, utilizámos uma variável auxiliar no OPL de modo a registar a função objetivo na situação ótima. Os resultados que obtivemos foram os que passamos a apresentar.



Como esperado, a área ardida para um "tempo de proteção" maior (ou seja, instante a partir do qual as células não ardem) aumenta. É interessante a curva que obtivemos (exponencial) a partir da variação desse parâmetro, mas compreende-se devido ao facto de cada célula ardida poder incendiar mais células vizinhas e assim sucessivamente.

Por curiosidade, também fizemos o mesmo teste para um limite de recursos menor, e o resultado também foi o expectável. A área ardida esperada é maior e o comportamento exponencial com a variação do parâmetro *time* mantém-se.

Literatura Consultada

Hof, J., Omi, P. N., Bevers, M., & Laven, R. D. (2000). *A timing-oriented approach to spatial allocation of fire management effort*. Forest Science, 46(3), 442-451.

Wei, Y. (2012). *Optimize landscape fuel treatment locations to create control opportunities for future fires*. Canadian Journal of Forest Research, 42(6), 1002-1014.

Ravi, R. (2005) *Approximations Algorithms - Primal-Dual Algorithms* , 15-854