

Relatório do Trabalho Final

Marlene Oliveira, N° 11327

Universidade de Évora - Escola de Ciências e Tecnologia
Mestrado em Engenharia Informática
Sistemas Computacionais de Apoio à Decisão

1 Introdução

No âmbito da disciplina de Sistemas Computacionais de Apoio à Decisão foi-nos solicitada a formulação e consequente resolução de três problemas relacionados com as temáticas abordadas nas aulas, nomeadamente, *Hidden Markov Models*, *Markov Decision Processes* e *Game Theory*.

2 Hidden Markov Models

2.1 O Problema

Um robot move-se seguindo uma linha que faz parte de uma pista pintada no chão de uma sala rectangular, no entanto este não conhece a sua posição na referida pista da sala. A pista é semelhante à da figura 2. Porém, o robot possui um sensor que mede a distância entre o ponto médio de um troço da pista e a parede mais próxima, mas que não é totalmente fiável (as medidas registadas não estão isentas de erro). Com o sensor é possível medir a distância a uma parede que se encontre entre dez centímetros a um metro do robot na direcção em que o mesmo está orientado. Pretende-se, portanto, saber em que local da pista se encontra o robot num determinado instante no tempo.

2.2 Formulação e Resolução do Problema

Considera-se que a pista está dividida em segmentos, tal como na figura 2. Estes segmentos correspondem ao local onde o robot se encontra num determinado instante, isto é, correspondem aos estados do problema. Assim, pode dizer-se que o conjunto de estados será: $\mathcal{X} = \{A, B, C, D, E, F, G, H, I\}$.

O robot mantém-se no mesmo segmento com uma probabilidade $p = 0.1$ e muda de segmento com uma probabilidade $1 - p$, ou seja, muda de segmento com probabilidade $p = 0.9$. Pode então usar-se uma Cadeia de Markov semelhante à da figura 1 para representar o movimento do robot na pista.

A partir da Cadeia de Markov é possível obter o modelo de transição de estado:

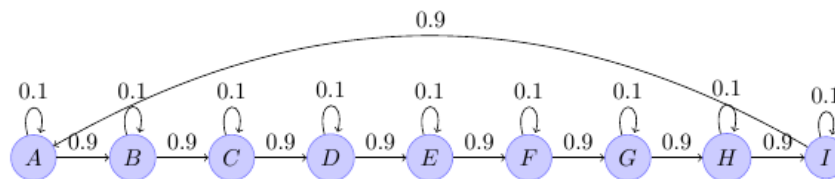


Fig. 1. Cadeia de Markov que representa o movimento do robot na pista.

$$p(x_{t+1}|x_t) = \begin{bmatrix} 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9 \\ 0.9 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.9 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.9 & 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.9 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.9 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.9 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.9 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9 & 0.1 \end{bmatrix}$$

As medições efectuadas pelo sensor de distância corresponderão às observações. Sendo assim, o conjunto das observações será: $\mathcal{Y} = \{a, b, c, d, e, f, g, h, i\}$, com cada um destes elementos a corresponder à distância entre o ponto médio de um segmento e a parede mais próxima. O modelo das observações, caso o sensor fosse perfeito, seria dado por:

$$p(y_t|x_t) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Como o sensor não é perfeito, as medições não estão isentas de erro. Assim, utiliza-se a distribuição de erro (1). O modelo das observações, tendo em conta o erro, obtém-se calculando a convolução entre o modelo das observações no caso do sensor ser perfeito e a distribuição correspondente ao erro do sensor, ou seja, $p(y_t|x_t) = p(y_t|x_t) * p(erro)$.

$$p(erro) = [0.0039, 0.0078, 0.0233, 0.155, 0.6202, 0.155, 0.0233, 0.0078, 0.0039] \quad (1)$$

Dado que não existe informação sobre o estado anterior, considera-se que a distribuição inicial do estado corresponderá a:

$$p(x_0) = [1/9 \ 1/9 \ 1/9 \ 1/9 \ 1/9 \ 1/9 \ 1/9 \ 1/9 \ 1/9]$$

A sequência observada foi: a, b, c, d, e .

Dado que se pretende saber onde se encontra o robot em cada instante no tempo, isto é, pretende-se saber o estado atual dadas as observações até ao momento, podemos concluir que este problema será resolvido com recurso à filtragem. Para tal, utiliza-se o algoritmo **forward()**, que permite actualizar o conhecimento para a distribuição 2 com propagação de informação "para a frente no tempo".

$$p(x_t|y_{1:t}) = \frac{1}{p(y_t)} \sum_{x_t} p(x_{t+1}|x_t) p(x_t|y_{1:t}) \quad (2)$$

Aplicando o algoritmo **forward()** obtém-se a sequência de distribuições:

$$\begin{bmatrix} 0.7656 & 0.0248 & 0.0001 & 0 & 0 \\ 0.1914 & 0.917 & 0.0327 & 0.0001 & 0 \\ 0.0287 & 0.0567 & 0.9525 & 0.0349 & 0.0002 \\ 0.0096 & 0.0013 & 0.0147 & 0.9613 & 0.0355 \\ 0.0048 & 0.0001 & 0.0001 & 0.0037 & 0.9634 \\ 0 & 0 & 0 & 0 & 0.0009 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

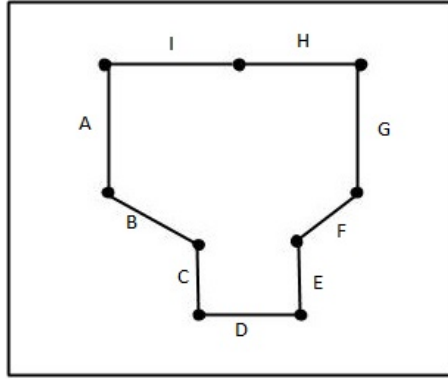


Fig. 2. Exemplo de uma pista desenhada no chão de uma sala.

em que cada distribuição do estado em cada instante de tempo corresponde a uma coluna.

Assim, quando se observa a é mais provável que o robot se encontre no segmento A , quando é observado b é mais provável que o robot esteja no segmento B , se foi observado c o robot estará no segmento C com maior probabilidade, caso seja observado d o robot tem maior probabilidade de se encontrar no segmento D e se for observado e o robot estará, com grande probabilidade, no segmento E da pista.

3 Markov Decision Processes

3.1 O Problema

Uma empresa pretende reduzir custos, poupando nos custos gerados pelos seus servidores, que respondem a vários tipos de requests(web, bases de dados, etc). Os servidores podem falhar enquanto estão online e podem falhar durante o boot. Para tal, a empresa precisa de ter um modelo de gestão que permita responder aos pedidos que os servidores recebem e reduzir o custo da energia gasta por estes mesmos servidores. Considera-se que os servidores têm uma probabilidade de falha de 0.1 quando estão ligados e uma probabilidade de falha de 0.01 quando estão a ser inicializados.

3.2 Formulação e Resolução do Problema

O problema pode ser visto como um Markov Decision Process, dado que a gestão dos custos com os servidores pode ser vista como uma sequência de tomada de decisões que envolve incerteza, dado que a o volume de pedidos não é constante ao longo do tempo.

O conjunto de instantes de decisão deste problema é finito e enumerável. Assim, o horizonte será $\mathcal{T} = \{1, 2, \dots, n\}$.

Os estados do problema consistem no número de servidores ligados em cada instante de decisão. Este conjunto é dado por: $\mathcal{X} = \{1, 2, \dots, n\}$.

As acções que podem ser tomadas consistem no número de servidores que se pretende ligar num dado instante de decisão. O conjunto das acções será: $\mathcal{A} = \{1, 2, \dots, n\}$.

Para calcular as probabilidades de transição de estado é necessário considerar dois casos distintos: o caso em que o número de servidores necessários para fazer face aos pedidos é superior ao número de servidores que se encontram ligados e o caso em que o número de servidores que são necessários para fazer face aos pedidos é inferior ao número de servidores ligados.

Consideremos o caso em que o número de servidores necessários para atender a todos os pedidos é superior ao número de servidores que se encontram ligados ($a_t \geq x_t$). A figura 3 fornece uma representação visual para este caso, em que x_t representa o número de servidores que já se encontram ligados e $a_t - x_t$ representa o número de servidores adicionais que terão de ser ligados.

Neste caso é necessário considerar duas probabilidades distintas:

- A probabilidade dos servidores que se encontram ligados se manterem ligados, ou seja, não existir falha enquanto estes estão online ($p(Z|x_t)$);
- A probabilidade dos servidores que têm de ser ligados o fazerem correctamente, isto é, não existirem falhas de boot ($p(W|x_t, a_t)$);

Uma falha enquanto os servidores estão online verifica-se quando um servidor deixa de responder. Este tipo de falhas pode afectar uma ou várias unidades, sendo que a probabilidade de falha quando os servidores se encontram online pode variar de estado para estado. Assim, podem existir vários cenários em que existem falhas, sendo exemplos: falhar um servidor e todos os outros se manterem ligados, falharem todos menos um, falharem metade dos servidores necessários e a outra metade continuar online, etc.

A tabela 1 representa a probabilidade dos servidores se manterem ligados mesmo quando existem falhas, $p(Z|x_t)$, e considera alguns destes casos.

| Z | $p(Z x_t)$ |
|-----------|---|
| x_t | $(0.99)^{x_t}(0.01)^0$ |
| $x_t - 1$ | $(0.99)^{x_t-1}(0.01)^1$ |
| $x_t - 2$ | $(0.99)^{x_t-2}(0.01)^2 \binom{x_t}{2}$ |
| ... | ... |
| 1 | $(0.99)(0.01)^{x_t-1} \binom{x_t}{x_t-1}$ |
| 0 | $(0.99)^0(0.01)^{x_t}$ |

Table 1. Probabilidade dos servidores se manterem ligados em cada estado.

Analisando a tabela, podemos deduzir que a expressão geral para calcular esta probabilidade será:

$$p(Z|x_t) = (0.99)^{x_t-n}(0.01)^n \binom{x_t}{n} \quad (3)$$

com n = número de servidores ligados num dado estado. Também quando os servidores estão a ser inicializados podem ocorrer falhas. Pode dizer-se que um servidor falha no boot quando este não é inicializado ou deixa de responder durante este processo.

Dado que no caso em que o número de servidores necessários para responder a todos os pedidos é superior ao número de servidores ligados será necessário ligar mais servidores, podem verificar-se falhas aquando do arranque dos mesmos. A probabilidade destas falhas ocorrerem varia consoante o estado e o número de servidores que terão de ser ligados. Podem existir vários cenários em que se verificam falhas, por exemplo, quando existe falha no arranque de dois servidores que têm de ser ligados, quando falham os arranques de todos os servidores, quando falham os arranques de metade dos servidores, etc.

A tabela 2 representa a probabilidade dos servidores arrancarem correctamente, $p(W|x_t, a_t)$, e considera alguns destes casos.



Fig. 3. Diagrama que representa o caso em que é necessário ligar mais servidores para fazer face ao volume dos pedidos.

| | |
|-----------------|---|
| W | $p(Z x_t, a_t)$ |
| $a_t - x_t$ | $(0.99)^{a_t - x_t} (0.01)^0$ |
| $a_t - x_t - 1$ | $(0.99)^{a_t - x_t - 1} (0.01)^1$ |
| $a_t - x_t - 2$ | $(0.99)^{a_t - x_t - 2} (0.01)^2 \binom{a_t - x_t}{2}$ |
| \dots | \dots |
| 1 | $(0.99)(0.01)^{a_t - x_t - 1} \binom{x_t}{a_t - x_t - 1}$ |
| 0 | $(0.99)^0 (0.01)^{a_t - x_t}$ |

Table 2. Probabilidade dos servidores serem inicializados correctamente.

Analisando a tabela, podemos deduzir que a expressão geral para calcular esta distribuição de probabilidade será:

$$p(W|x_t, a_t) = (0.99)^{a_t - x_t - n} (0.01)^n \binom{a_t - x_t}{a_t - x_t - n} \quad (4)$$

com n = número de servidores ligados num dado estado.

Consideremos agora o caso em que o número de servidores necessários para atender aos pedidos é inferior ao número de servidores ligados ($a_t < x_t$). O diagrama da figura 4 ilustra esta situação.

De modo a reduzir custos, neste caso considera-se a probabilidade dos servidores que estão

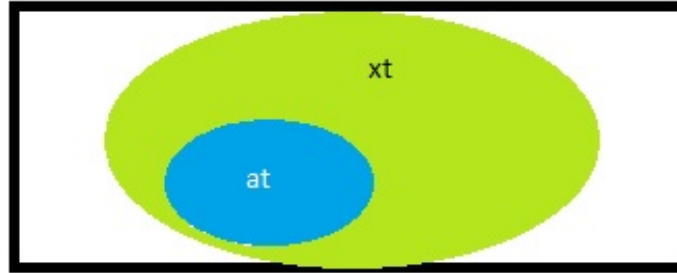


Fig. 4. Diagrama que representa o caso em que o número de servidores ligados x_t é superior ao número de servidores necessários a_t .

ligados sem estarem em uso se desligarem de certeza ($p(Y|x_t, a_t)$). A acção de desligar os servidores que não estão a ser utilizados também está sujeita à ocorrência de falhas. Pode dizer-se que existe uma falha quando um servidor que deve desligar-se não o faz. Assim, podem existir vários cenários em que se verificam falhas, como por exemplo: quando um servidor que devia desligar-se se mantém ligado, quando apenas metade dos servidores que deviam desligar-se o faz, etc. A tabela 3 representa a probabilidade dos servidores que devem desligar-se o fazerem correctamente, $p(Y|a_t)$, e considera alguns destes casos.

Analisando a tabela, podemos deduzir que a expressão geral para calcular esta distribuição de probabilidade será:

$$p(Y|a_t) = (0.99)^{a_t - n} (0.01)^n \binom{a_t}{a_t - n} \quad (5)$$

com n = número de servidores ligados num dado estado.

As probabilidades de transição obtêm-se calculando a convolução das três probabilidades obtidas anteriormente. Assim, temos que:

$$p(x_{t+1}|x_t, a_t) = p(Z|x_t) * p(W|x_t, a_t) * p(Y|a_t) \quad (6)$$

| Y | $p(Y a_t)$ |
|-----------|---|
| a_t | $(0.99)^{a_t}(0.01)^0$ |
| $a_t - 1$ | $(0.99)^{a_t-1}(0.01)^1$ |
| $a_t - 2$ | $(0.99)^{a_t-2}(0.01)^2 \binom{a_t}{2}$ |
| ... | ... |
| 1 | $(0.99)(0.01)^{a_t-1} \binom{a_t}{a_t-1}$ |
| 0 | $(0.99)^0(0.01)^{a_t}$ |

Table 3. Probabilidade dos servidores se manterem ligados em cada estado.

Outro elemento importante para a resolução deste problema é a recompensa, dado que esta representa o prémio que se ganha num instante por se tomar uma determinada decisão. A função recompensa tem em conta dois casos: o caso em que a procura supera a oferta, isto é, o número de pedidos supera a capacidade dos servidores e o caso em que a procura não supera a oferta, isto é, o número de pedidos não supera a oferta.

Assim, no caso em que a procura não supera a oferta, temos que a recompensa será dada pela diferença entre o número de pedidos (h) e o custo total da energia (xE , em que x corresponde ao número de servidores activos e E corresponde ao custo da energia). No caso em que a procura supera a oferta, a recompensa terá em conta o custo das falhas. Assim, a recompensa é dada pela diferença entre o que se ganha pelos pedidos a que é possível responder com sucesso ($x(c - E)$, com c correspondente à capacidade de um servidor e E correspondente ao custo da energia), já descontando o custo da energia, e aquilo que se perde por não ser possível dar resposta aos pedidos que sobram ($k(h - xc)$, com k correspondente ao custo por falha, x correspondente ao número de servidores disponíveis e c correspondente à capacidade de um servidor). A expressão da função recompensa será (7).

$$r(x, h) = \begin{cases} h - xE & h \leq xc \\ x(c - E) - b(h - xc) & h > xc \end{cases} \quad (7)$$

Com esta informação é então possível calcular a utilidade para cada estado utilizando a expressão (8). Como as acções óptimas são obtidas andando para trás no tempo, a utilidade também se propaga nesse sentido e o problema é formulado num intervalo de tempo finito, inicializa-se a propagação no instante de tempo terminal, sendo $U_N(x_N) = r_N(x_N)$.

$$U_t(x_t) = r_t(x_t) + \max_{a_t} \sum_{x_{t+1}} p(x_{t+1}|x_t, a_t) U_{t+1}(x_t + 1) \quad (8)$$

A acção que maximiza a probabilidade esperada é a acção óptima. Neste caso, corresponderá ao número de servidores que devem estar ligados para se obter maior recompensa, isto é, responder a mais pedidos, enquanto se poupa o máximo de energia possível.

$$a_t^* = \operatorname{argmax}_{a_t} \sum_{x_{t+1}} p(x_{t+1}|x_t, a_t) U_{t+1}(x_t + 1) \quad (9)$$

A política óptima, π , corresponderá ao conjunto das regras de decisão que devem ser utilizadas em cada instante de tempo e que optimizam as recompensas que podem vir a ser obtidas caso seja seguida uma dada política. Como o horizonte é finito, estas políticas não são estacionárias.

A resolução deste problema irá permitir obter um gráfico onde é possível ver qual a acção óptima escolhida ao longo do horizonte de acordo com a variação do número de pedidos.

Considere-se agora uma versão simplificada deste problema em que existem apenas dois servidores e o horizonte corresponde a um dia (24 horas). Foi esta versão do problema que foi resolvida com sucesso.

Para o problema simplificado, o horizonte será $\mathcal{T} = \{0, 1, 2, \dots, 23\}$. Os estados corresponderão

ao número de servidores ligados, isto é, $\mathcal{X} = \{0, 1, 2\}$. As acções possíveis serão: não ligar nenhum servidor, ligar um servidor e ligar os dois servidores, ou seja, o conjunto das acções será $\mathcal{A} = \{0, 1, 2\}$. Os servidores têm uma probabilidade de não arracarem correctamente correspondente a 0.1 e uma probabilidade de avaria em funcionamento de 0.01. Assim, as probabilidades de transição serão:

$$p(x_{t+1}|x_t, a_0) = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$p(x_{t+1}|x_t, a_1) = \begin{bmatrix} 0.1 & 0.9 & 0 \\ 0.01 & 0.99 & 0 \\ 0.01 & 0.99 & 0 \end{bmatrix}$$

$$p(x_{t+1}|x_t, a_2) = \begin{bmatrix} 0.01 & 0.18 & 0.81 \\ 0.001 & 0.108 & 0.891 \\ 0.0001 & 0.0198 & 0.9801 \end{bmatrix}$$

A função recompensa continua a ser a mesma de anteriormente, ou seja, continuará a ser 7.

Aplicando o algoritmo que permite obter as acções óptimas para cada instante do horizonte, obtém-se o gráfico 5. Neste gráfico o sinusóide representa a evolução do número de pedidos ao longo do dia. Além disso, existem três linhas que correspondem às acções (não ligar nenhum servidor (a_0), ligar um servidor (a_1) e ligar os dois servidores (a_2)). As linhas correspondentes a ligar um servidor ou ligar dois servidores encontram-se sobrepostas. Analisando o gráfico, verifica-se que

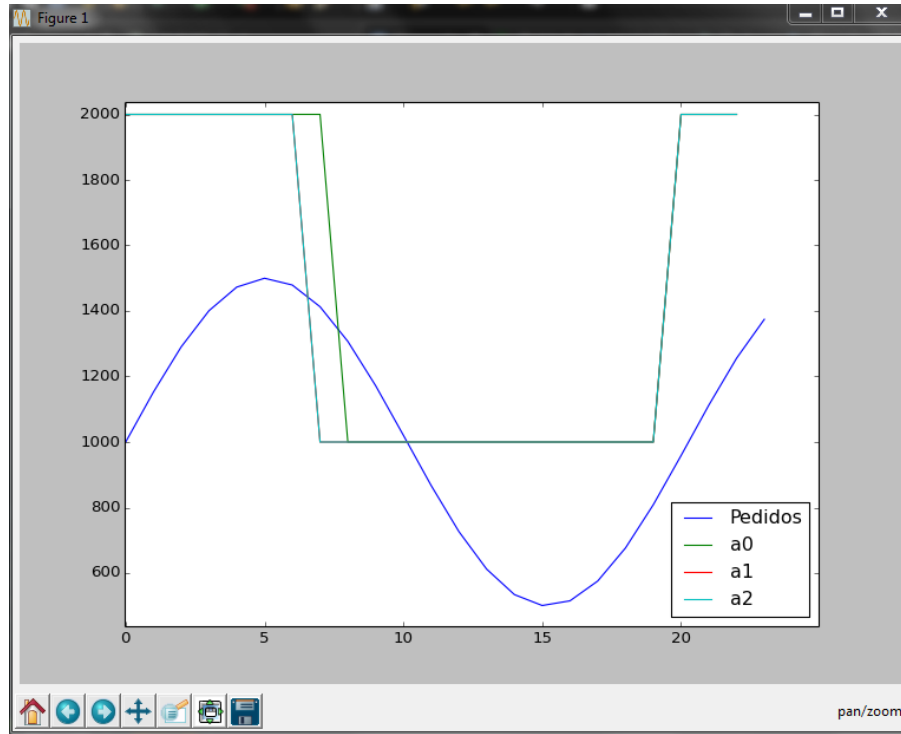


Fig. 5. Representação gráfica da evolução das acções óptimas ao longo do horizonte.

no caso em que a acção tomada corresponde a não ligar nenhum servidor a melhor decisão será manter dois servidores ligados até que o número de pedidos comece a descer e depois desligar um deles para poupar energia durante as horas em que há menor número de pedidos. Idealmente, este

servidor permanecerá desligado até que o número de pedidos volte a aumentar consideravelmente, altura em que será ligado novamente. Os dois servidores permanecerão então ligados. Nos casos em que as acções tomadas correspondem a ligar um ou dois servidores, verifica-se que um servidor é desligado assim que o número de pedidos começa a diminuir ligeiramente e não quando já se regista uma queda mais acentuada, como acontecia anteriormente. Contudo, também nestes casos o servidor permanecerá desligado, de modo a economizar energia, até que se volte a verificar uma subida acentuada do número de pedidos.

Sendo assim, pode concluir-se que a política óptima consistirá em manter os dois servidores ligados durante as horas em que existe maior número de pedidos e desligar um dos servidores nas horas em que o número de pedidos é mais reduzido.

4 Game Theory

4.1 O Problema

Duas placas de rede partilham o mesmo meio e comunicam entre si, sendo que apenas uma das placas pode transmitir informação em cada momento. Para que apenas uma destas placas transmita informação num dado intervalo de tempo, o tempo foi dividido em slots. No caso de existirem colisões, as máquinas esperam e voltam a tentar transmitir informação no intervalo de tempo seguinte. A retransmissão de informação ocorre com uma probabilidade p e a espera ocorre com uma probabilidade $1 - p$.

4.2 Formulação e Resolução do Problema

Este problema pode ser resolvido de dois modos: como um jogo em que as placas enviam informação com a mesma probabilidade ou como um jogo em que cada placa envia informação com uma probabilidade distinta.

| Atraso | $p(\text{atraso})$ |
|--------|--------------------|
| 1 | p |
| 2 | $(1 - p)p$ |
| 3 | $(1 - p)^2 p$ |
| ... | ... |
| n | $(1 - p)^{n-1} p$ |

Table 4. Probabilidade de atraso.

Dado que quando uma placa envia informação a outra tem de esperar, considera-se que o benefício de uma prejudica a outra. Assim, estratégias puras tendem a não ser as mais proveitosas dado que, quando as probabilidades de envio das placas não são iguais, uma placa irá ser mais beneficiada que a outra. As melhores estratégias serão mistas, isto é, cada placa enviará informação aleatoriamente.

Considere-se o caso em que as placas enviam informação com probabilidades de envio diferentes.

Suponha-se então que a placa 1 decide enviar ou esperar com probabilidades $\sigma_1 = (p_1, 1 - p_1)$ e a placa 2 decide enviar ou esperar com probabilidades $\sigma_2 = (p_2, 1 - p_2)$.

Outro dado importante para a resolução deste problema é a penalização esperada, ou seja, é a penalização que existe quando uma placa não envia informação num slot de tempo. O primeiro passo para obter a penalização esperada consiste em calcular a probabilidade de ocorrer um atraso.

Verifica-se que existe um atraso quando uma placa não envia informação num slot, mas envia no slot seguinte. O atraso é proporcional ao número de slots de tempo em que a placa não enviou informação, isto é, quanto mais tempo a placa estiver sem enviar informação, maior será o atraso.

A probabilidade de atraso é calculada tendo em conta o atraso anterior, isto é, a probabilidade da placa envia informação no slot actua e não enviou no(s) slot(s) anterior(es). O cálculo da probabilidade pode ser consultado na tabela 4.

Analisando a tabela 4, verifica-se a expressão geral que permite calcular penalização esperada será dada pela equação (10).

$$\text{Penalização Esperada} = \sum_{i=1}^{+\infty} i(1-p)^{i-1}p \quad (10)$$

A penalização esperada comporta-se como uma progressão geométrica semelhante à da equação (11).

$$\sum_{n=1}^{+\infty} nr^n \quad (11)$$

Desenvolvendo a equação, obtém-se (12).

$$\begin{aligned} \sum_{n=1}^{+\infty} nr^n &= r + \sum_{n=1}^{+\infty} (n+1)r^{n+1} \\ &= r + r \sum_{n=1}^{+\infty} (n+1)r^n \\ &= r + r \sum_{n=1}^{+\infty} nr^n + r \sum_{n=1}^{+\infty} r^n \end{aligned} \quad (12)$$

Sabendo (13), temos que (12) pode ser escrita como (14).

$$r \sum_{n=1}^{+\infty} r^n = \frac{r}{1-r} \quad (13)$$

$$\sum_{n=1}^{+\infty} nr^n = r + r \sum_{n=1}^{+\infty} nr^n + r \left(\frac{r}{1-r} \right) \quad (14)$$

Para resolver (14), consideremos que $\sum_{n=1}^{+\infty} nr^n$ corresponde a X. Assim, a equação (14) poderá ser escrita como $X = r + rX + \frac{r^2}{1-r}$. Resolvendo esta equação em ordem a X, obtém-se (15) e podemos concluir (16).

$$X = \frac{r}{1-r} + \frac{r^2}{(1-r)^2} \quad (15)$$

$$\sum_{n=1}^{+\infty} nr^n = \frac{r}{1-r} + \frac{r^2}{(1-r)^2} \quad (16)$$

Dado que, como foi referido anteriormente, a penalização esperada se comportar como uma progressão geométrica, podemos dizer que a expressão geral da penalização esperada quando ocorre um atraso será dada por (17).

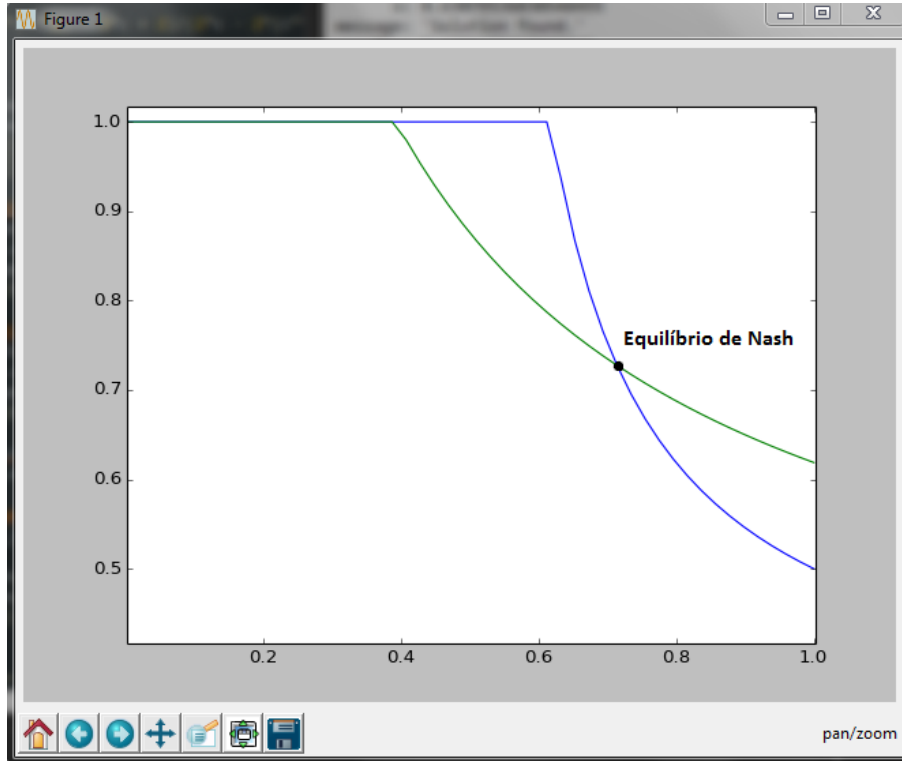


Fig. 6. Representação gráfica das melhores respostas das placas e Equilíbrio de Nash obtido.

$$\begin{aligned}
 \text{Penalização Esperada} &= \sum_{i=1}^{+\infty} i(1-p)^{i-1}p \\
 &= \frac{p}{1-p} \left(\frac{1-p}{p} + \frac{(1-p)^2}{p^2} \right) \\
 &= \frac{1}{1-p} \left((1-p) + \frac{(1-p)^2}{p} \right) \\
 &= \frac{1}{1-p} * \frac{(1-p)}{p} \\
 &= \frac{1}{p}
 \end{aligned} \tag{17}$$

Uma vez que as placas usarão estratégias mistas para conseguirem enviar informação num dado slot de tempo, a recompensa esperada tem em conta o que se ganha quando as placas enviam informação ao mesmo tempo, o que se ganha quando uma placa envia informação e a outra não e o que se ganha quando nenhuma das placas envia. Assim, vejamos a expressão que permite obter a recompensa para a placa 1 num dado slot de tempo, tal como pode ser observado em (18). A recompensa para a placa 2 calcula-se de um modo análogo.

$$r_1(p_1, p_2) = p_1 p_2 (1 + r_1(p_1, p_2)) + \frac{(1-p_1)p_2}{p_1} + (1-p_1)(1-p_2)(1 + r_1(p_1, p_2)) \tag{18}$$

Resolvendo (18) em ordem a r_1 , obtém-se (19).

$$r_1(p_1, p_2) = \frac{p_1^2 p_2 + (1-p_1)p_2 + p_1(1-p_1)(1-p_2)}{p_1^2 + p_1 p_2 + 2p_1^2 p_2} \tag{19}$$

Para obter a melhor resposta para a placa 1, é necessário proceder à minimização desta função recompensa. Assim, a melhor resposta para a placa 1 será dada por (20). A melhor resposta para a placa 2 obtém-se de um modo análogo.

$$m_1(p_2) = \operatorname{argmin}_{p_1} r_1(p_1, p_2) \quad (20)$$

Com toda esta informação é agora possível obter o gráfico 6 que representa as melhores respostas de uma placa em função da estratégia da outra placa. A linha verde representa as melhores respostas para a placa 1 e a linha azul representa as melhores respostas para a placa 2.

O Equilíbrio de Nash permite saber qual a melhor estratégia (aquela que tem uma recompensa esperada maior) para uma placa e que mantém a estratégia da placa adversária fixa. No gráfico 6, a intersecção das curvas corresponde ao equilíbrio de Nash, que corresponde, sensivelmente, ao ponto (0.71, 0.72).

No caso em que o problema é visto como um jogo em que as placas enviam informação com a mesma probabilidade, considera-se que $p_1 = p_2 = p$. Assim, a recompensa esperada será dada pela expressão (21). Minimizando esta expressão, verifica-se que o valor óptimo para p será, sensivelmente, 0.58.

$$\begin{aligned} r(p) &= \frac{p^2 + (1-p) + (1-p)^2}{2p - 2p_2} \\ &= \frac{2p^2 - 3p + 2}{2p - 2p_2} \end{aligned} \quad (21)$$

Tendo em conta os resultados obtidos nos dois métodos, verifica-se que a probabilidade de envio é superior no caso em que as placas enviam informação com probabilidades diferentes. Será, portanto, mais proveitoso utilizar placas com probabilidades de envio diferentes, dado que a probabilidade de envio é maior. Por exemplo, caso se tratem de placas com modelos diferentes e/ou de fabricantes diferentes, é mais proveitoso que estas placas enviem informação deste modo. Porém, se as placas pertencerem ao mesmo fabricante e tiverem o mesmo modelo, será mais proveitoso que ambas enviem com a mesma probabilidade.

References

- Bar13. Miguel Barão. *Sistemas Computacionais de Apoio à Decisão*. 2013.
 Wik. Wikipedia. *Convolution*. <http://en.wikipedia.org/wiki/Convolution>. Consultado em: 2013-12-23.