

2º Trabalho de Laboratório:

Simulação de Monte Carlo do jogo do Monopólio



Curso: Licenciatura em Engenharia Electrotécnica e de Computadores

UC: Modelação e Simulação

1º Semestre 2021/2022 (P2)

Docente: Alberto Vale

Autores: Grupo nº 20

- (96170) Daniel Alexandre Mendes Pacheco daniel.pacheco@tecnico.ulisboa.pt
- (96258) José Miguel Valério Antunes jose.m.valerio.antunes@tecnico.ulisboa.pt
- (96265) Luís Das Neves Alves Maçorano luis.macorano@tecnico.ulisboa.pt
- (96317) Rui Pedro Canário Daniel ruipcdaniel@tecnico.ulisboa.pt

Compromisso de Ética de Originalidade

O grupo de alunos acima identificado garante que o texto deste relatório e todo o software e resultados entregues foram inteiramente realizados pelos elementos do grupo, com uma participação significativa de todos eles, e que nenhuma parte do trabalho ou do software e resultados apresentados foi obtida a partir de outras pessoas ou fontes.

Data: 21/01/2022

Índice

| | |
|--|----|
| Introdução | 2 |
| P1. | 3 |
| a) Aplicação do método de Monte Carlo | 3 |
| b) Frequências relativas dos diferentes estados | 3 |
| c) Rendas médias em regime estocástico estacionário | 5 |
| P2. | 5 |
| i) Coerência entre a sequência de estados e a sequência de eventos | 5 |
| ii) Convergência da distribuição de probabilidades dos estados com o número de <i>runs</i> | 6 |
| iii) Estudo ao valor de <i>Ndiscard</i> | 7 |
| iv) Validação do programa e dos resultados | 8 |
| P3. | 9 |
| P4. | 10 |
| Conclusão | 10 |
| Bibliografia | 10 |

Introdução

Temos como objetivo simular um sistema estocástico de Markov discreto usando o método de Monte Carlo.

Para tal, realiza-se uma simulação do jogo de Monopólio simplificado, representado na figura 1, recorrendo ao método de Monte Carlo, para determinar a distribuição de probabilidade de equilíbrio de entrar nas várias “casas”.

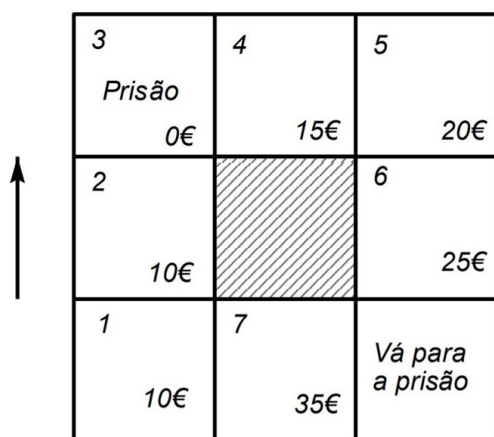


Figura 1: Tabuleiro do jogo de Monopólio simplificado

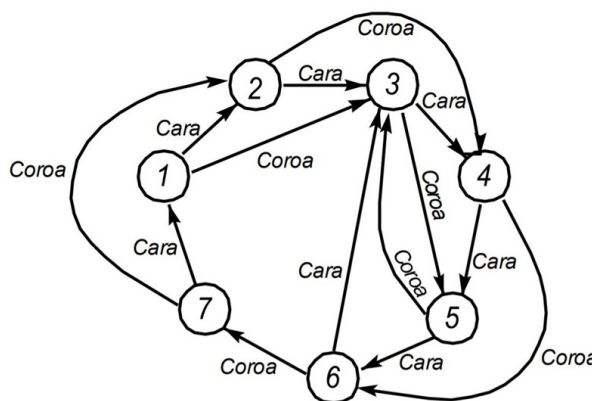


Figura 2: Diagrama de transição de estado

Partindo da casa 0, o jogador atira uma moeda ao ar. Se sair “cara”, a sua marca avança uma casa, e se sair “coroa” avança duas casas, devendo pagar uma renda correspondente à casa em que entrou. Caso entre em “Vá para a prisão”, é imediatamente transferido para a casa 3. Este jogo pode então ser modelado pelo diagrama de transição de estados da figura 2.

Trata-se de um sistema de acontecimentos discretos estocástico porque depende do lançamento da moeda, que é um acontecimento definido de forma probabilística por:

$$P(\text{“Sai cara”}) = P(\text{“Sai coroa”}) = 0,5 \quad (1)$$

Este constitui uma cadeia de Markov dado que a transição a partir de um dado estado depende apenas do estado atual e do valor do evento que leva à transição.

Tendo em vista a obtenção da distribuição de probabilidade de equilíbrio da ocorrência de cada estado, é utilizado o método de Monte Carlo. Este consiste na repetição de uma experiência aleatória simulada em computador, utilizando para isso um gerador de números aleatórios. Assim, através destes resultados, é possível obter uma boa estimativa das quantidades que se pretendem calcular.

De seguida apresentamos o código, realizado em MATLAB, que simula o sistema:

```
function [z, y, avancador, zfreq, lucro, zfreqfinal] = monopoly(Njogadas, NMC, Ndiscard, Ncasas)
    z = zeros(1, Ncasas); % Número de vezes que se caiu em cada estado
    zfinal = zeros(1, Ncasas); % Número de vezes que o jogo acabou em cada estado
    zfreq = zeros(1, Ncasas); % Frequências relativas de cada estado
    zfreqfinal = zeros(1, Ncasas); % Probabilidade de o jogo terminar em cada estado
    y = zeros(1, Njogadas); % Casa em que se caiu em cada jogada
    avancador = zeros(1, Njogadas); % Vetor que armazena os sucessivos acontecimentos
    Aluguer = [10, 10, 0, 15, 20, 25, 35]; % Valor do aluguer de cada casa
    x = 0; % Estado atual
    for i = 1 : NMC
        x = 0;
        for j = 1 : Njogadas
            avanca = randi(2,1); % Número de casas a avançar no tabuleiro,
            % dado pelo valor obtido no lançamento da moeda: 1 = Cara, 2 = Coroa
            if x == 7 % Se o jogador estiver no estado 7 é equivalente que este avance
                x = x - 7; % do estado 7 ou do estado (imaginário) 0
            end
            x = x + avanca;
            if x == 7 % Prisão: jogador estava no estado 5 e avançou 2 casas (5 + 2 = 7)
                x = 3; % ou jogador estava no estado 6 e avançou 1 casa (6 + 1 = 7)
            end
            if x == 8 % Caso em que avançamos do estado 6 para o estado 7
                x = x - 1; % (avança 2 casas, mas apenas aumenta em 1 estado)
            end
            y(j) = x;
            avancador(j) = avanca;
            if j > Ndiscard
                z(x) = z(x) + 1;
            end
        end
        zfinal(x)=zfinal(x) + 1;
    end
    zfreqfinal = zfinal./NMC;
    zfreq = z ./ ((Njogadas - Ndiscard) * NMC);
    lucro = zfreq .* Aluguer; % Rendas médias espectáveis de cada estado
end
```

Figura 3: Código que simula um jogo de Monopólio simplificado

Na figura 3 encontra-se o código desenvolvido após as modificações efetuadas na alínea P3, pelo que apenas nesta iremos explicar detalhadamente as variáveis *zfinal*, *zfreqfinal*, bem como o seu respetivo processamento.

P1.

a) Aplicação do método de Monte Carlo

Aplicámos o método de Monte Carlo para um jogo de Monopólio simulado com apenas um *run* e constituído por 20 jogadas, sem descartar um transitório inicial. Observámos os estados percorridos em função da jogada realizada, e os valores simulados para o resultado do lançamento da moeda nas sucessivas jogadas desse *run* (sendo que 1 corresponde ao acontecimento “Sai cara” e 2 corresponde ao acontecimento “Sai coroa”).

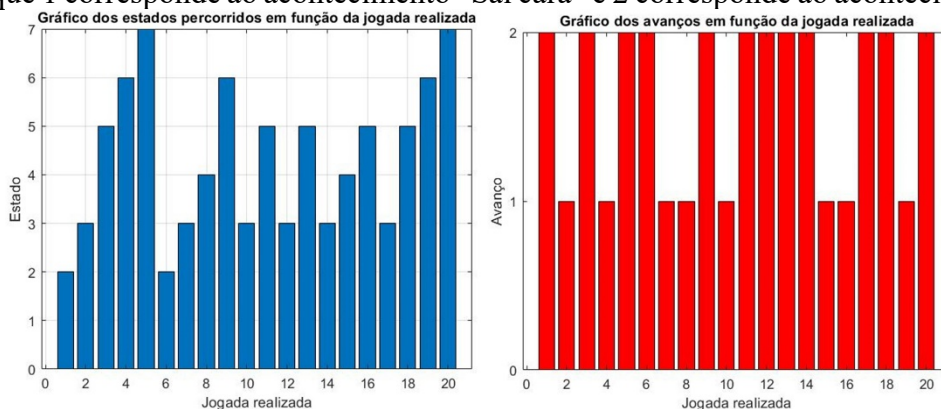


Figura 4: Gráfico dos estados percorridos em função da jogada realizada (esquerda) e dos valores simulados para o resultado do lançamento da moeda nas sucessivas jogadas (direita)

Podemos analisar na figura 4 o funcionamento esperado do sistema. Esta análise será efetuada na alínea P2-i).

b) Frequências relativas dos diferentes estados

Inicialmente verificámos a partir de que jogada, de um único *run*, a probabilidade de o jogador estar num determinado estado atinge uma probabilidade de equilíbrio. Isto é feito através de simulações com um *run* na qual se varia o número de jogadas, de modo a analisar a frequência relativa dos estados à medida que o número de jogadas aumenta (por simplicidade apresentamos apenas os resultados para os estados 1 e 5, sendo que para os restantes estados os resultados apresentam comportamentos semelhantes).

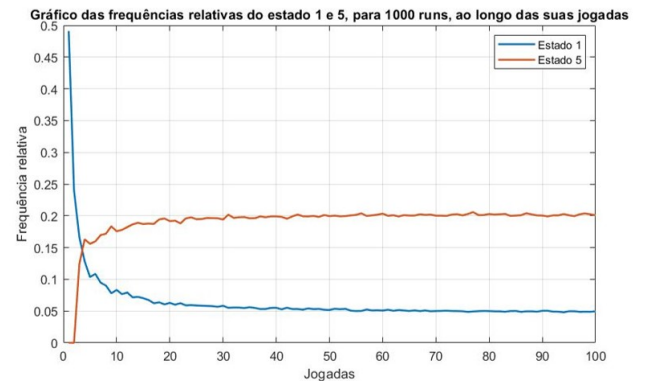
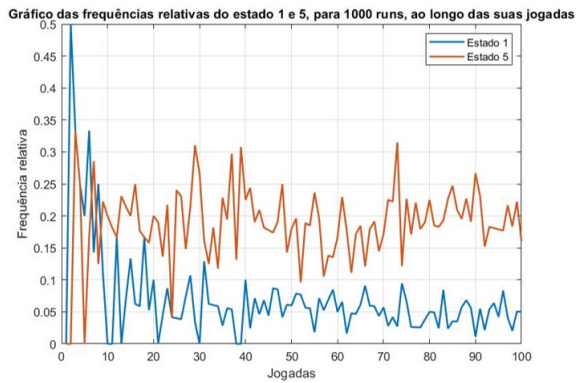


Figura 5: Gráfico das frequências relativas do estado 1 e 5 ao longo das suas jogadas, para 1 run (esquerda) e para 1000 runs (direita)

Podemos então verificar pela parte esquerda da figura 5 que, dado a utilizarmos um único *run*, possuímos enormes oscilações, pois o número de amostras é muito reduzido.

Deste modo, procedemos à mesma simulação, mas com 1000 *runs* (valor que utilizamos também na simulação seguinte), de modo a diminuir as oscilações dos resultados experimentais.

Como podemos observar pela parte direita da figura 5, como seria de esperar, o aumento do número de *runs* permitiu uma muito menor oscilação dos valores das frequências relativas, observando-se que estes tendem para um equilíbrio. Também se verifica que para um número de jogadas muito baixo, apesar de o número de *runs* ter aumentado, os valores continuam a estar muito afastados da probabilidade de equilíbrio, pelo que para a determinação das mesmas deve ser descartado um transitório inicial.

Com o objetivo de estudar a distribuição de probabilidades de equilíbrio dos estados do Monopólio simplificado, procedemos à aplicação do método de Monte Carlo a uma simulação com 1000 *runs*, cada um constituído por 100000 jogadas. Este processo tem em vista a produção de um gráfico de barras com as frequências relativas dos diferentes estados quando se realizam muitas jogadas, descartando um transitório inicial de 6 jogadas. O número de jogadas descartadas no transitório inicial será explicado na pergunta P2-iii).

Obtivemos a frequência relativa de cada estado através do quociente entre o número de vezes que se caiu nesse estado e entre o número de jogadas contabilizadas na totalidade dos *runs*:

$$\text{Frequência relativa} = \frac{z}{(N_{\text{jogadas}} - N_{\text{discard}}) \cdot NMC} \quad (2)$$

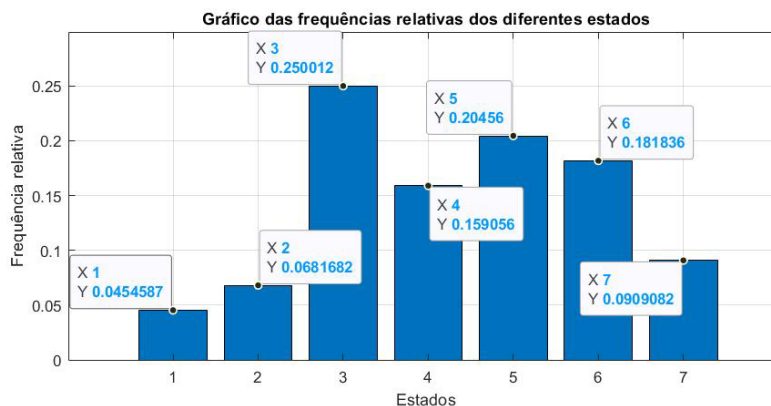


Figura 6: Gráfico das frequências relativas dos diferentes estados

Deste modo, é possível estimar a probabilidade de o jogador estar em cada estado, sendo esta igual à frequência relativa desse mesmo estado. O número de jogadas consideradas foi elevado, tendo em vista melhorar estas estimativas, cujos resultados são (em que P_j é a probabilidade associada ao estado j):

$$P_1 \approx 0,045 \quad P_2 \approx 0,068 \quad P_3 \approx 0,250 \quad P_4 \approx 0,159 \quad P_5 \approx 0,205 \quad P_6 \approx 0,182 \quad P_7 \approx 0,091$$

Pela análise da figura 6, podemos observar que o estado 3 (“Prisão”), é o que possui maior frequência relativa com o valor de 0,25. Isto deve-se à existência de 4 estados com probabilidade igual a 0,5 de transitar para o estado 3 (estados 1, 2, 5, 6).

Contabilizando as transições possíveis para cada estado, verifica-se que existe apenas uma transição possível para o estado 1. O mesmo sucede para o estado 7. Desta forma as frequências relativas destes são baixas. Já para o estado 2, existem duas transições possíveis para o mesmo, verificando-se a mesma situação para os estados 4, 5 e 6.

A frequência relativa do estado 2 é baixa, dado que as transições possíveis com destino ao estado 2, partem dos estados 1 e 7, que por sua vez apresentam frequências relativas baixas.

Desta forma, as frequências relativas dos estados 4, 5 e 6 são semelhantes, sendo superiores às dos estados 1 e 7, mas inferiores à do estado 3.

c) Rendas médias em regime estocástico estacionário

De seguida, nas condições da alínea anterior, produzimos um gráfico de barras com as rendas médias expectáveis para cada um dos estados, em €/jogada, sendo esta obtida para cada estado através do produto da probabilidade do jogador se encontrar neste pelo seu valor de aluguer.

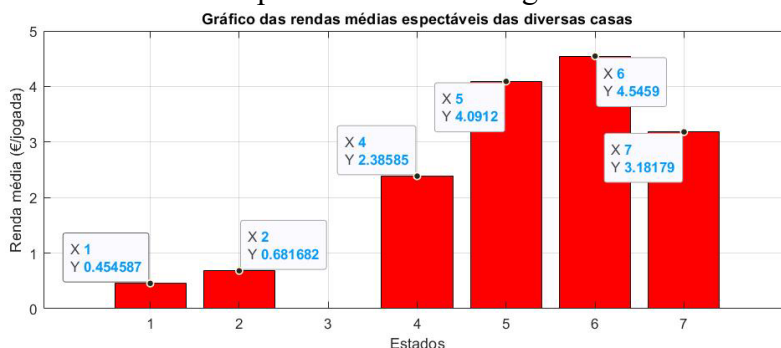


Figura 7: Gráfico das rendas médias expectáveis das diversas casas

Analisando a figura 7, podemos verificar que a renda média dos estados 1 e 2 é bastante baixa, devido não só à baixa probabilidade de cair dos estados 1 e 2, mas também ao facto de estes possuírem um custo menor que o das outras casas. A casa mais lucrativa é a 6, apesar de não ser a de renda mais elevada, enquanto a casa 7 é apenas a terceira casa mais lucrativa, apesar de possuir a renda mais elevada.

A renda média da casa 3 é nula, pois apesar desta ter a maior frequência relativa entre todas, possui custo de aluguer nulo.

Podemos então concluir, que, no contexto do jogo do Monopólio, para determinação das propriedades mais rentáveis é necessário não só ter em conta a sua renda como também a probabilidade de um jogador entrar nestas.

P2.

i) Coerência entre a sequência de estados e a sequência de eventos

Podemos proceder à validação do nosso programa através da verificação da coerência entre a sequência de estados e a sequência de eventos, tendo em conta o diagrama de transição de estado da figura 2. Para tal, utilizámos os resultados obtidos na simulação da alínea P1a) e apresentados na figura 4, dado que o baixo número de jogadas utilizado permite que a verificação seja feita manualmente.

De forma a simplificar a nossa análise, atribuímos, na figura 8, designações a cada uma das transições possíveis.

Passando à análise da figura 4, podemos através do gráfico da esquerda observar o estado atual e o estado seguinte, e através do gráfico da direita observar o resultado do lançamento da moeda, correspondendo 1 a “cara” e 2 a “coroa”.

Temos então que, pela figura 9, durante as 20 primeiras jogadas, todas as transições são válidas pois correspondem a transições presentes na figura 8. A transição correspondente à primeira jogada também é válida, pois o acontecimento inicial “Sai coroa” dita que o sistema transite de forma válida da casa 0 para o estado 2.

| Estado Atual | Acontecimento | Estado Seguinte | Transição |
|--------------|---------------|-----------------|-----------|
| 1 | Sai cara | 2 | a |
| | Sai coroa | 3 | b |
| 2 | Sai cara | 3 | c |
| | Sai coroa | 4 | d |
| 3 | Sai cara | 4 | e |
| | Sai coroa | 5 | f |
| 4 | Sai cara | 5 | g |
| | Sai coroa | 6 | h |
| 5 | Sai cara | 6 | i |
| | Sai coroa | 3 | j |
| 6 | Sai cara | 3 | k |
| | Sai coroa | 7 | l |
| 7 | Sai cara | 1 | m |
| | Sai coroa | 2 | n |

Figura 8: Tabela das transições válidas entre estados

| Jogada | Estado Atual | Acontecimento | Estado Seguinte | Transição |
|--------|--------------|---------------|-----------------|-----------|
| 2 | 2 | Sai cara | 3 | c |
| 3 | 3 | Sai coroa | 5 | f |
| 4 | 5 | Sai cara | 6 | i |
| 5 | 6 | Sai coroa | 7 | l |
| 6 | 7 | Sai coroa | 2 | n |
| 7 | 2 | Sai cara | 3 | c |
| 8 | 3 | Sai cara | 4 | e |
| 9 | 4 | Sai coroa | 6 | h |
| 10 | 6 | Sai cara | 3 | k |
| 11 | 3 | Sai coroa | 5 | f |
| 12 | 5 | Sai coroa | 3 | j |
| 13 | 3 | Sai coroa | 5 | f |
| 14 | 5 | Sai coroa | 3 | j |
| 15 | 3 | Sai cara | 4 | e |
| 16 | 4 | Sai cara | 5 | g |
| 17 | 5 | Sai coroa | 3 | j |
| 18 | 3 | Sai coroa | 5 | f |
| 19 | 5 | Sai cara | 6 | i |
| 20 | 6 | Sai coroa | 7 | l |

Podemos então comprovar que no nosso programa existe coerência entre a sequência de estados e a sequência de eventos.

ii) Convergência da distribuição de probabilidades dos estados com o número de runs

Tendo em vista o estudo da convergência da distribuição das probabilidades dos diferentes estados quando aumenta o número de *runs*, tomámos os seguintes procedimentos:

Em primeiro lugar, realizámos simulações de 2000 jogadas, com *Ndiscard* igual a 6, onde considerámos amostras para um número de *runs* que toma todos os valores inteiros de 1 a 100. Para cada amostra, calculámos a soma dos erros relativos de todos os estados, sendo que para a obtenção destes erros se utilizou a equação:

$$\text{Erro relativo} = \frac{|\text{Probabilidade obtida} - \text{Probabilidade equilíbrio}|}{\text{Probabilidade equilíbrio}} \quad (3)$$

Nesta expressão assumimos que o valor da probabilidade de equilíbrio, ou seja, a obtida pelo método de Monte Carlo na alínea P1b), corresponde ao seu valor exato.

Tendo em conta que geramos sequências de números aleatórios através da instrução de MATLAB “*rand('state', seed);*”, verificámos que a variável *seed* permite gerar sequências aleatórias distintas. Assim, simulámos o processo para valores de *seed* de 0 e 1, obtendo os resultados da figura 10.

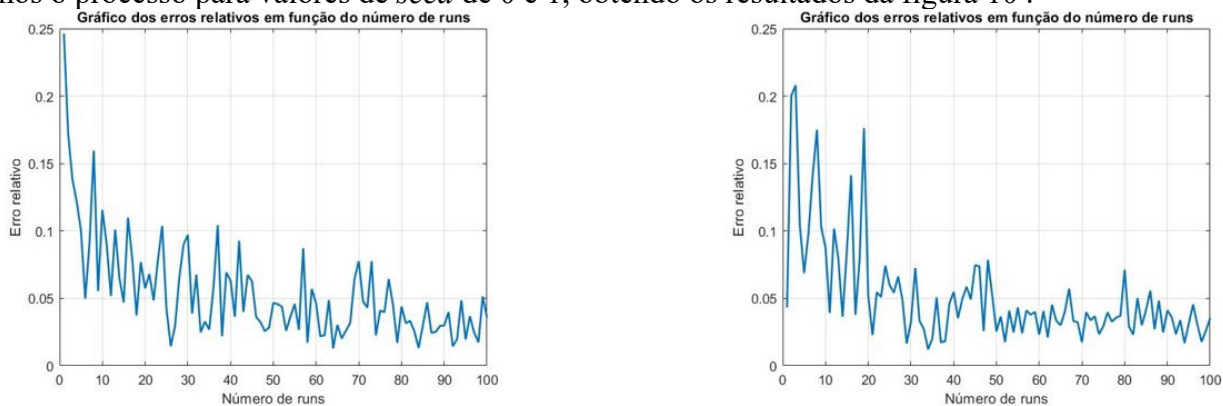


Figura 10: Gráfico do erro relativo em função do número de *runs* para *seed* = 0 (esquerda) e para *seed* = 1 (direita)

Podemos observar que os resultados obtidos dependem da maneira como se inicializa o gerador de números aleatórios, pois cada sequência irá ditar uma cadeia de acontecimentos distintos. Isto é particularmente relevante para um número baixo de amostras (número baixo de *runs*), pois aí nem todas as movimentações foram contempladas. Assim, será o início da sequência aleatória que irá decidir quais delas serão consideradas, afetando significativamente os resultados.

No entanto, à medida que o número de *runs* aumenta, um maior número de possibilidades de movimentação é considerado, o que faz com que a importância dos primeiros valores da sequência aleatória vá diminuindo. Assim, à medida que o número de *runs* aumenta, não só a probabilidade de cada estado vai tendendo para um equilíbrio, como também os valores gerados por sequências aleatórias distintas vão-se assemelhando cada vez mais.

Apresenta-se ainda, na figura 11, o gráfico correspondente à média dos valores do erro relativo obtidos para cada *seed*. Foram utilizados todos os valores inteiros de *seed*, entre 0 e 100, de modo a reduzir substancialmente as oscilações.

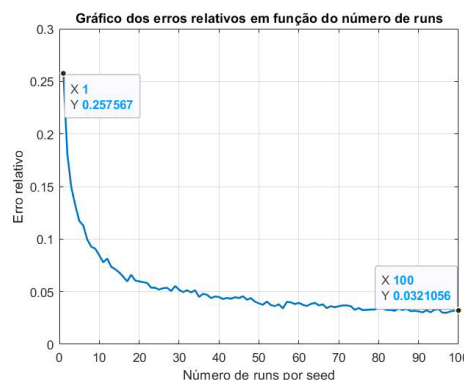


Figura 11: Gráfico do erro relativo em função do número de *runs* por *seed*, utilizando a média dos valores obtidos para todos os valores inteiros de *seed* entre 0 e 100

A partir da figura 11, é possível verificar que o erro relativo vai diminuindo de acordo com a equação $a \frac{1}{\sqrt{n}} + b$, sendo " a " um número real positivo (aproximadamente 0,258) e " b " o limite quando $n \rightarrow \infty$ (aproximadamente 0).

Isto indica que existe uma proporcionalidade do erro relativo com o inverso da raiz quadrada do número de *runs*, logo a convergência da distribuição das probabilidades dos diferentes estados dá-se com a raiz quadrada do número de *runs*, sendo esse o resultado que se pretendia demonstrar.

Isto era previsível, pois através do método de Monte Carlo obtemos um número n de amostras (número de *runs*) para a frequência relativa de cada estado (X_i), sendo a estimativa para a probabilidade do jogador estar num determinado estado, feita através do cálculo da média dos valores das amostras:

$$\bar{X}_n = \frac{\sum_{i=1}^n X_i}{n} \quad (4)$$

Desta forma, conseguimos obter o desvio-padrão de \bar{X}_n , tendo como hipóteses que as amostras X_i são independentes, e que $X_i \sim X$.

$$Var(\bar{X}_n) = Var\left(\frac{\sum_{i=1}^n X_i}{n}\right) = \frac{1}{n^2} n Var(X_i) = \frac{Var(X)}{n} \quad (5)$$

$$\text{desvio-padrão}(\bar{X}_n) = \sqrt{Var(\bar{X}_n)} = \sqrt{\frac{Var(X)}{n}} \quad (6)$$

Assim, podemos verificar que a convergência da distribuição das probabilidades dos diferentes estados se dá com a raiz quadrada do número de *runs*, pois o desvio-padrão é proporcional ao inverso de \sqrt{n} . Ou seja, quanto maior for \sqrt{n} , menor será o desvio-padrão, e maior será a convergência da frequência relativa, em cada estado, para o valor da sua probabilidade de equilíbrio.

iii) Estudo ao valor de *Ndiscard*

Na alínea P1b) observámos que com o aumento do número de *runs*, ocorre uma diminuição das oscilações no gráfico das frequências relativas ao longo das suas jogadas devido ao consequente aumento do número de amostras. Assim, nas simulações para a análise do valor de *Ndiscard*, utilizou-se um número constante de amostras de modo a eliminar a influência deste parâmetro.

Realizámos então uma análise para um valor de jogadas tal que o número de jogadas contabilizadas fosse sempre 50, ou seja, cuja diferença entre o número de jogadas e *Ndiscard* seja 50. Para tal, foi feita uma simulação em que foi calculada a soma dos erros relativos da probabilidade dos estados em função de *Ndiscard*, para valores deste entre 0 e 49. Realizámos a simulação para 1000 *runs*, onde as oscilações eram bastante acentuadas, e para 100000 *runs*.

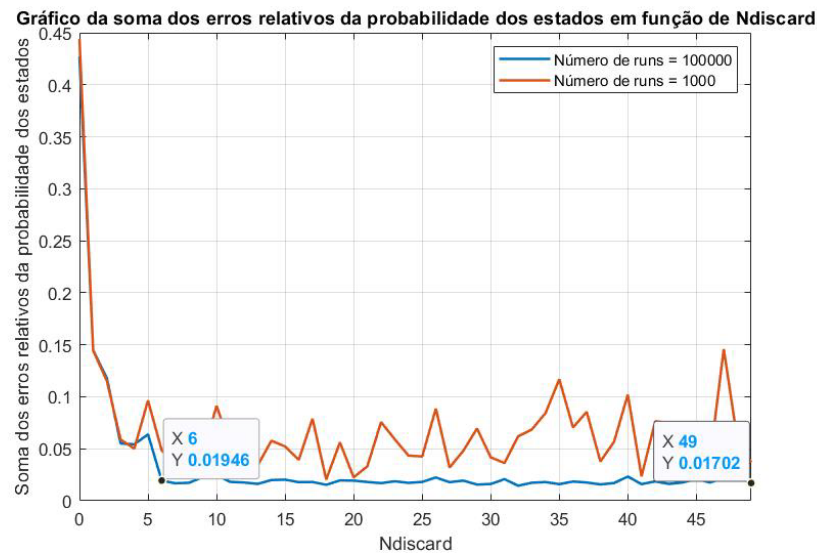


Figura 12: Gráfico da soma dos erros relativos da probabilidade dos estados em função de *Ndiscard*, para 50 jogadas contabilizadas por *run*

A partir da figura 12, para a curva correspondente a 100000 *runs*, podemos verificar que os erros relativos diminuem com o aumento de *Ndiscard*. Inicialmente esta diminuição é bastante rápida, mas a partir do valor 6 observamos que esta diminuição passa a ser muito lenta, visto que a soma dos erros relativos da probabilidade

dos estados apenas diminui aproximadamente 0,25% entre $N_{discard} = 6$ e $N_{discard} = 49$. Concluimos então que um bom valor para $N_{discard}$ será 6, pois permite obter resultados bastante exatos sem sacrificar muito tempo de computação, pois para valores superiores de $N_{discard}$, para manter o número de amostras, terão de ser realizadas mais jogadas.

Iremos agora analisar o efeito de $N_{discard}$ quando possui um valor próximo do número de jogadas. Para tal, foi feita uma simulação em que foi calculada a soma dos erros relativos da probabilidade dos estados em função de $N_{discard}$, para valores deste entre 0 e 49, mas mantendo o número de jogadas fixo e igual a 50. Realizámos a simulação para 1000 *runs*, onde as oscilações eram bastante acentuadas, e para 100000 *runs*.

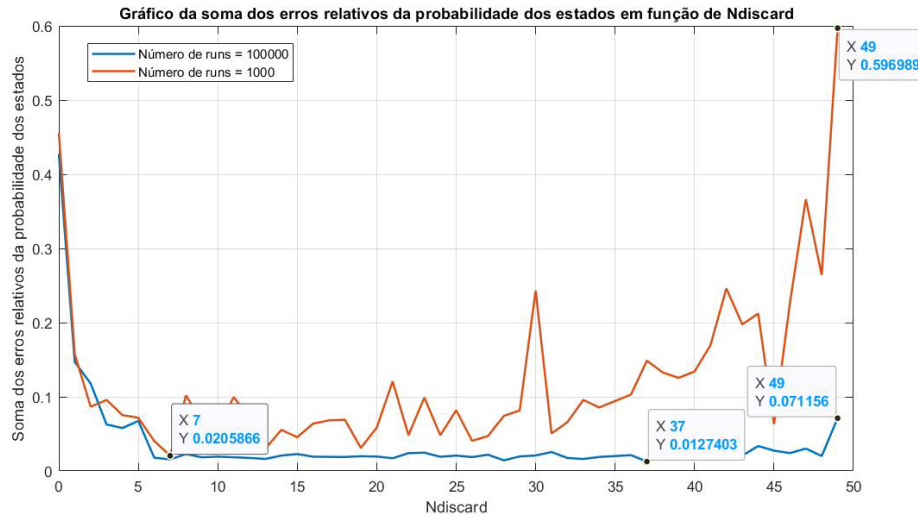


Figura 13: Gráfico da soma dos erros relativos da probabilidade dos estados em função de $N_{discard}$, com número de jogadas fixo igual a 50

A partir da figura 13 podemos verificar que, tal como observámos anteriormente, numa fase inicial os erros relativos diminuem com o aumento de $N_{discard}$. Também se verifica que para $N_{discard}$ próximo do número de jogadas fixo, os erros relativos aumentam, pois, neste caso, temos um número de amostras insuficiente para que as probabilidades sejam estimadas com exatidão elevada. Podemos confirmar esta observação verificando que este aumento é mais acentuado para o caso em que temos um menor número de *runs*, ou seja, um menor número de amostras. Confirma-se assim que o procedimento utilizado anteriormente para calcular um valor adequado para a variável $N_{discard}$ foi correto.

iv) Validação do programa e dos resultados

Até agora os valores da probabilidade de equilíbrio de cada estado foram calculados utilizando o método de Monte Carlo. No entanto, podemos também utilizar um modelo analítico para estudar esta distribuição probabilística, o que nos permite validar os valores obtidos anteriormente.

Começemos por definir $P^T(k) = [P_1(k) \dots P_n(k)]$, sendo $P_j(k)$ a probabilidade de se estar no estado j na jogada k .

Neste modelo, o vetor $P(k)$ evolui do seguinte modo:

$$P(k+1) = A^T P(k) \quad (7)$$

Cada elemento a_{ij} da matriz A corresponde à probabilidade de o modelo transitar do estado i para o estado j , pelo que esta matriz A corresponde à matriz de transição de uma cadeia de Markov.

Considerando os acontecimentos “Sai cara” e “Sai coroa” equiprováveis, obtemos a seguinte matriz A :

$$A = \begin{bmatrix} 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 & 0.5 \\ 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Neste caso, temos uma cadeia de Markov regular, pois temos que $A^m > 0$ para algum inteiro positivo m . Por exemplo, todas as entradas da matriz A^{16} são positivas.

Sendo A uma matriz de transição de uma cadeia de Markov regular, o Teorema da distribuição de equilíbrio diz que quando $k \rightarrow \infty$, a distribuição de probabilidades dos estados tende para um valor limite P que satisfaz o sistema de equações lineares:

$$A^T P = P \quad (8)$$

Assim, começando com qualquer distribuição inicial de probabilidade dos estados, após um número muito grande de jogadas, a probabilidade de estar nos diversos estados é dada por P , sendo independente da distribuição inicial.

Resolvendo a equação (8), obtemos as seguintes probabilidades para cada estado, sendo que estas são iguais às obtidas pelo método de Monte Carlo na alínea P1b), comprovando assim a exatidão dos resultados obtidos com recurso a esse método:

$$P_1 \approx 0,045 \quad P_2 \approx 0,068 \quad P_3 \approx 0,250 \quad P_4 \approx 0,159 \quad P_5 \approx 0,205 \quad P_6 \approx 0,182 \quad P_7 \approx 0,091$$

P3.

De modo a estimar a probabilidade de o sistema estar no estado 4 ao longo das sucessivas jogadas, modificámos o programa da seguinte forma:

Criámos um vetor denominado *zfinal* com a finalidade de conter o número de ocorrências em que o jogo acabou em cada estado. Este é obtido através duma atualização no fim de cada *run*, na qual é aumentada em 1 unidade a posição do vetor correspondente àquela em que o jogador se encontrava no fim do jogo.

Também criámos um vetor denominado *zfreqfinal*, com o objetivo de obter a probabilidade de o jogo terminar em cada um dos estados, sendo esta calculada como o quociente entre *zfinal* e o número de *runs*.

Pode verificar-se a implementação destas alterações na figura 3.

Por fim, foi definido um novo vetor denominado *prob4*, com o intuito de guardar a probabilidade de o jogo acabar na posição 4, para cada valor do número de jogadas, como se apresenta na figura 14.

```
for Njogadas = 1:20
    [z, y, avancador, zfreg, lucro, zfregfinal] = monopoly(Njogadas, NMC, Ndiscard, Ncasas);
    prob4(Njogadas) = zfregfinal(4);
end
```

Figura 14: Código relativo à pergunta 3

Realizámos uma simulação deste novo programa com 100000 *runs*, sem descartar um transitório inicial, para um número de jogadas entre 1 e 20, de modo a estimar, pelo método de Monte Carlo, a probabilidade de estar no estado 4 ao longo das sucessivas jogadas, estando os resultados apresentados em seguida.

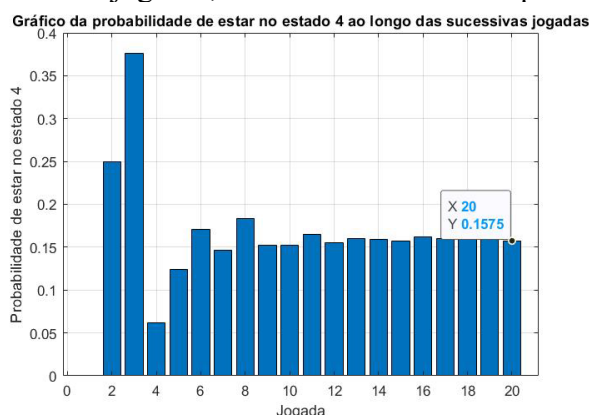


Figura 15: Gráfico da probabilidade de estar no estado 4 ao longo das sucessivas jogadas, para 100 mil *runs*

Pela análise da figura 15, podemos verificar que os resultados obtidos são coerentes com os acontecimentos possíveis. Nas jogadas 0 e 1 é impossível estar o jogador estar no estado 4, pois na jogada 0 está-se, invariavelmente, no estado 0, e na jogada 1 só se pode estar nos estados 1 ou 2. Na 2ª jogada é possível atingir o estado 4 através de dois acontecimentos "Sai coroa" consecutivos (probabilidade = $0,5 \cdot 0,5 = 0,25$). Na 3ª jogada, o quociente entre o número de combinações favoráveis a atingir o estado 4 e o número de combinações possíveis aumenta, aumentando essa probabilidade. Pelo contrário, na quarta jogada essa probabilidade diminui, dado que apenas existe um caso favorável (sair "cara" quatro vezes consecutivas, cuja probabilidade é $(0,5)^4 = 0,0625$).

Após ultrapassarmos este transitório inicial, a probabilidade de o jogador estar no estado 4 numa dada jogada, tende a convergir com o aumento do número de jogadas, atingindo uma probabilidade de equilíbrio.

Isto acontece porque, inicialmente, devido ao número reduzido de jogadas realizadas, ainda nem todas as possibilidades de movimentação ao longo do tabuleiro foram contempladas. Assim, à medida que mais possibilidades vão sendo consideradas, a probabilidade de o jogador se encontrar no estado 4 vai oscilando. Quando já grande parte das movimentações possíveis foram consideradas, esta probabilidade atinge valores próximos do equilíbrio, sendo que ao fim de 20 jogadas o seu valor é 0,1575. Este valor encontra-se bastante próximo da frequência relativa obtida para o estado 4 na alínea P1b), cujo valor é 0,159.

P4.

Na situação em que, se o jogador for enviado para a prisão, terá de permanecer lá após a próxima jogada, é necessário criar um novo estado correspondente à jogada em que o jogador está preso. A esse estado chamamos estado P.

Este estado é alcançado quando o jogador cai na casa “Vá para a prisão”, o que pode acontecer de duas formas: estar no estado 5 e o resultado do lançamento da moeda ser “coroa”, ou estar no estado 6 e o resultado do lançamento ser “cara”. Assim, no diagrama de transição de estado, teremos duas setas a apontar para o estado P, uma proveniente do estado 5 associada ao acontecimento “Sai coroa”, e outra proveniente do estado 6 associada ao acontecimento “Sai cara”.

Encontrando-se no estado P, o jogador terá de esperar uma jogada até poder voltar a jogar. Isto significa que do estado P terá de se prosseguir para um estado onde o jogador ainda se encontre na casa 3, onde se localiza a prisão, mas já possibilitado a jogar. Esse é o estado 3. Assim, independentemente do resultado do lançamento da moeda, o estado P irá transitar para o estado 3, pelo que no diagrama de transição de estado deverá haver duas setas do estado P para o estado 3, associadas a ambos os resultados possíveis do lançamento da moeda.

Com estes aspetos em consideração, obtém-se o diagrama de transição de estado da figura 16.

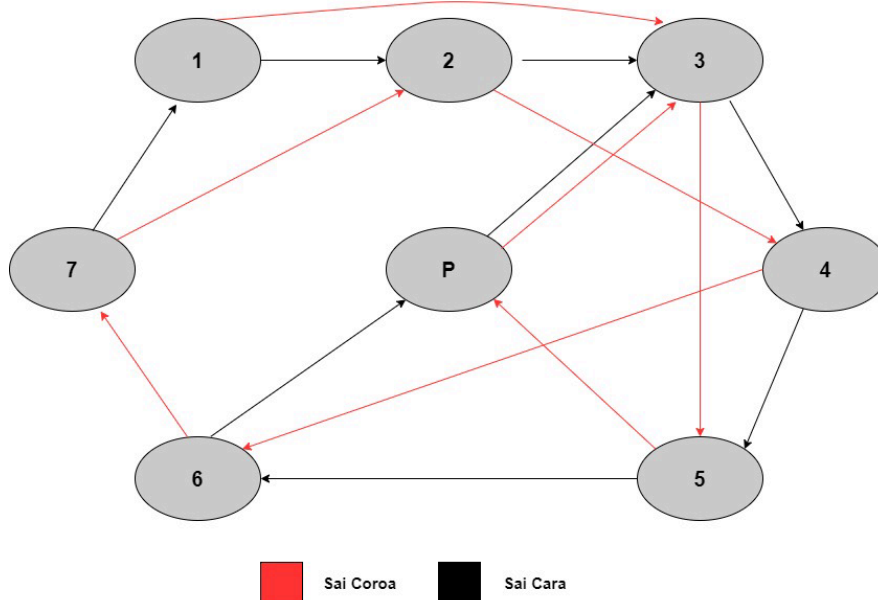


Figura 16: Diagrama de transição de estado para a situação apresentada na pergunta 4

Conclusão

Neste relatório, estudámos a distribuição de probabilidades de equilíbrio dos estados do Monopólio simplificado, através do método de Monte Carlo. Recorremos à elaboração de um programa em MATLAB visando a recolha de amostragens aleatórias massivas com o propósito de obter resultados numéricos para as frequências relativas de cada estado. Desta forma, utilizámos a aleatoriedade dos acontecimentos para obter resultados para um problema que, a priori, é determinístico.

Este método revela-se bastante útil, sendo utilizado em diversas áreas e podendo ser aplicado a conceitos complexos e abrangentes. Porém, apresenta algumas desvantagens, tais como a necessidade de uma enorme capacidade computacional e de elevado tempo de processamento para problemas mais complexos.

Procedemos à validação do nosso programa através de uma análise à coerência entre a sequência dos estados ao longo das várias jogadas e a sequência de eventos, e através de um estudo analítico pelo Teorema da distribuição de equilíbrio para uma cadeia de Markov regular.

Bibliografia

- [1] Modelação e Simulação - Slides das Aulas Teóricas 2021/2022, 1º Semestre (LEEC), do professor João Miranda Lemos, IST-DEEC;
- [2] 1º Semestre 2021/2022, Modelação e Simulação, 2º Trabalho de Laboratório -Simulação de Monte Carlo do jogo do Monopólio - Enunciado;
- [3] MATLAB, MATLAB R2021a, versão 9.10.0.1739362, 2021.