

Trabalho 2

Simulação de Monte Carlo do jogo do Monopólio



Imagens: Wikipedia

Autores:

André Teodósio : 99889

↪ andre.teodosio@tecnico.ulisboa.pt

Francisco Carvalho : 99941

↪ franciscoaesc@tecnico.ulisboa.pt

João Gonçalves : 99995

↪ jrazevedogoncalves@tecnico.ulisboa.pt

Teresa Nogueira : 100029

↪ maria.teresa.ramos.nogueira@tecnico.ulisboa.pt

Supervisor:

Prof. João Silvestre

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.

Dezembro 2022

Introdução

Um dos principais objetivos da análise de Cadeias de Markov, é a determinação das probabilidades de encontrar a cadeia em vários estados, em específicos instantes de tempo. Definimos o *state probability vector* como:

$$\boldsymbol{\pi}(k) = [\pi_1(k), \pi_2(k), \pi_3(k), \pi_4(k), \pi_5(k), \pi_6(k), \pi_7(k)]$$

onde $\pi_j(k) \triangleq \mathcal{P}r\{X_k = x_j\}$ no instante k , para o espaço de estados $\mathcal{X} = \{x_j\}$, com $j = 1, \dots, 7$. E, assim, por associação natural a um sistema dinâmico, a evolução do sistema é dada pela equação de transição de estados:

$$\boldsymbol{\pi}(k+1) = \boldsymbol{\pi}(k) \mathbf{P}$$

em que \mathbf{P} é matriz de transição (estocástica) que condensa o comportamento do jogador no tabuleiro consoante o lançamento da moeda.

3	4	5
Prisão 0€	15€	20€
2		6
10€		25€
1	7	Vá para a prisão
10€	35€	

$$\mathbf{P} = \begin{bmatrix} 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 & 0.5 \\ 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 1: Tabuleiro do Monopólio simplificado. (Imagem: Guia**).

→ **Steady-state analysis:** Qual é a probabilidade de encontrarmos a Cadeia de Markov no estado x_j "in the long run"¹?

$$\pi_j = \lim_{k \rightarrow +\infty} \pi_j(k)$$

Se π_j existir, refere-se como *steady-state*, *equilibrium*, ou *stationary state probability*. Se existir para todos os estados, definimos o *stationary state probability vector* $\boldsymbol{\pi}$.

Def.: Irredutibilidade da Cadeia de Markov[1]

A Cadeia de Markov diz-se irredutível sse todos os estados comunicam entre si.

Def.: Periodicidade dos estados[2]

A periodicidade do estado $x_j \in \mathcal{X}$ é: $d(x_j) \triangleq \gcd\{k \in \mathbb{N}^+ : \mathbf{P}^k(x_j, x_j) > 0\}$
 → O estado x_j diz-se aperiódico se $d(x_j) = 1$ e periódico se $d(x_j) > 1$.

Com base nas definições supramencionadas, invocamos o **Ergodic Theorem for Primitive Chains**[3], aplicável à Cadeia de Markov objetiva de estudo. Ergo, para $k \rightarrow +\infty$, $\mathbf{P}^k \rightarrow \mathbf{e}\boldsymbol{\pi} \implies \boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P} \wedge \boldsymbol{\pi}\mathbf{e} = 1$, em que $\mathbf{e} = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]^T$ e $\boldsymbol{\pi}$ pode ser revisto como vetor próprio da matriz estocástica \mathbf{P} , associado ao valor próprio $\lambda_0 = 1$ (→ "No other eigenvalue of \mathbf{P} has absolute value greater than 1."[4]).

Justaposto, trivialmente deduzimos (teoricamente) a *equilibrium probability distribution* da Cadeia de Markov em questão:

$$\therefore \boldsymbol{\pi} = [0.0455 \quad 0.0682 \quad 0.25 \quad 0.1591 \quad 0.2045 \quad 0.1818 \quad 0.0909]$$

¹"By "long run" we mean that the system (...) is allowed to operate for a sufficiently long period of time so that the state probabilities can reach some fixed values (...)"[2]

P1 | Distribuição de probabilidades de equilíbrio dos estados do Monopólio simplificado.

a) Estados percorridos num *run* e valores simulados para o resultado do lançamento da moeda nas sucessivas jogadas desse *run*.

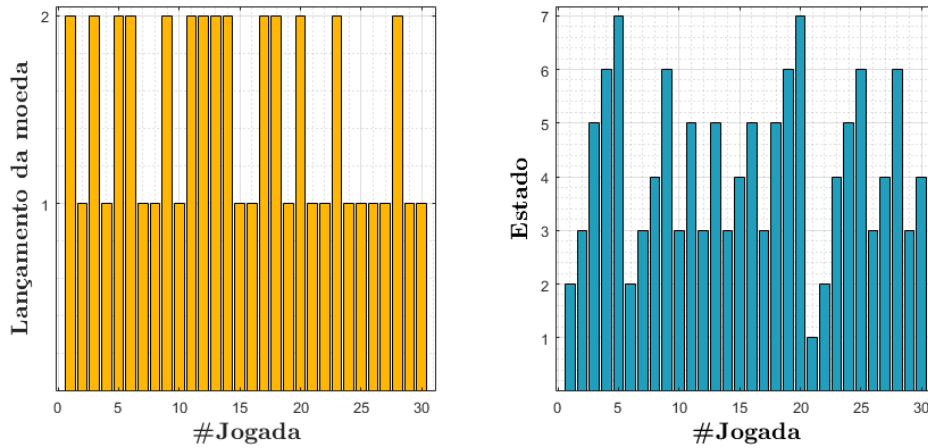


Fig. 2: Resultado do lançamento da moeda e estados percorridos numa *run* com 30 jogadas².

→ Observações

- ▲ O reduzido valor de iterações (jogadas) é escolhido para uma fácil visualização dos gráficos, já que, como veremos em seguida, este valor mostra-se insuficiente para garantir um regime estocástico estacionário.
- ▲ De notar a elevada incidência no estado 3 e 5, resultado congruente com o vetor de equilíbrio (*steady-state vector*) da cadeia em questão (vide secção introdutória)

b) Frequências relativas dos diferentes estados (muitas jogadas), descartando um transitório inicial.

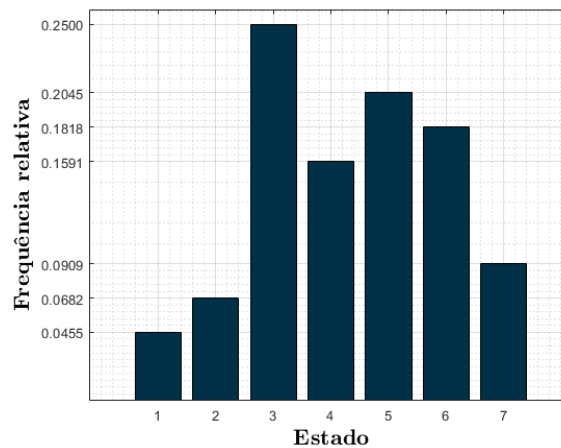


Fig. 3: Frequências relativas para uma simulação de 1000000 *runs* cada uma com 1000 jogadas, de onde 20 são descartadas.

²Vide secção P2 iii).

→ **Observações**

- ▲ Os *steady-state values* simulados de cada estado aproximam-se rigorosamente dos valores teóricos expectáveis (secção introdutória).
- ▲ O valor de *burn-in* (*Ndiscard*) estipulado é o que garante a melhor eliminação do *bias* inicial imposta pela influência da casa de partida (x_0).
- ▲ A distribuição das probabilidades estacionárias por estado são intuitivamente explicadas pelo diagrama da cadeia: Sucintamente, x_1 é o estado menos provável, a transição (fora a eventual transição inicial do estado x_0) só é possível através de uma lançamento com resultado coroa do estado 7. x_3 é o estado mais provável, a transição é possível através do estado x_1 (coroa), x_2 (cara), x_5 (coroa) e x_6 (cara).

c) **Renda média em regime estocástico estacionário.**

Trivialmente se obtêm os valores de renda média efetuando o produto das entradas dos vetores *zfreq* e *Aluguer* (nomenclatura conforme o Guia Laboratorial):

$$\text{Renda média} = [\text{zfreq}(i) \cdot \text{Aluguer}(i)], i = 1, \dots, 7$$

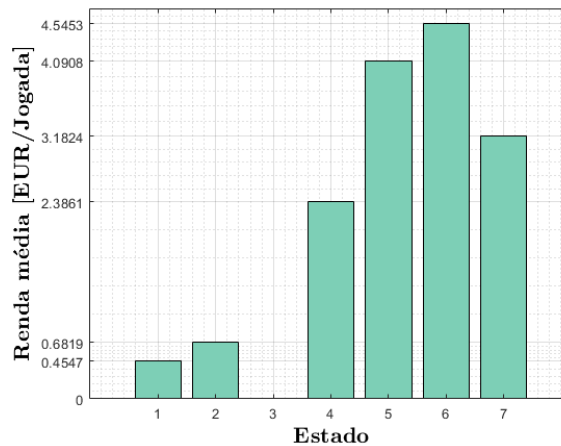


Fig. 4: Renda média para uma simulação de 1000000 *runs* cada uma com 1000 jogadas, de onde 20 são descartadas.

→ **Observações**

- ▲ O estado 3 possui renda média nula, já que o pagamento é inexistente na prisão (como veremos na secção P4 o pagamento poderá equacionar em tempo).
- ▲ A renda é tanto maior quanto maior for a frequência relativa e o preço de aluguer do respetivo estado.

P2 | Validação do programa.i) **Coerência entre a sequência dos estados ao longo das várias jogadas e a sequência de eventos que a determinam.**

A transição entre estados depende de um acontecimento definido probabilisticamente (o lançamento da moeda). Complementamos a matriz de transição com a Tab. 1.:

Tab. 1: Transição de estados.

x_j	x_{j+1}	
	Cara	Coroa
x_1	x_2	x_3
x_2	x_3	x_4
x_3	x_4	x_5
x_4	x_5	x_6
x_5	x_6	x_3
x_6	x_3	x_7
x_7	x_1	x_2

→ **Observações**▲ Afluem quatro estados para x_3 :

$$\pi_3 = \frac{1}{2}(\pi_1 + \pi_2 + \pi_5 + \pi_6)$$

▲ É apenas possível transitar para os estados x_2 , x_4 , x_5 e x_6 através dos dois estados anteriores:

$$\pi_2 = \frac{1}{2}(\pi_1 + \pi_7) \quad \pi_4 = \frac{1}{2}(\pi_2 + \pi_3)$$

$$\pi_5 = \frac{1}{2}(\pi_3 + \pi_4) \quad \pi_6 = \frac{1}{2}(\pi_4 + \pi_5)$$

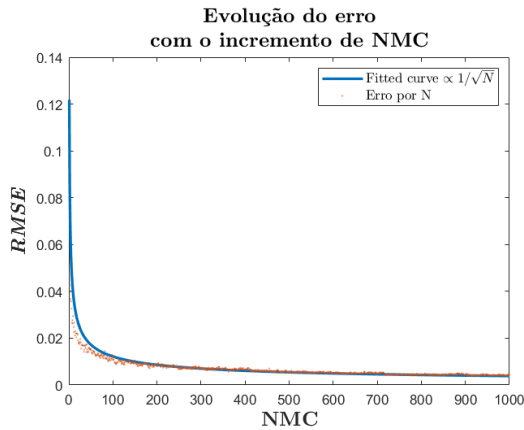
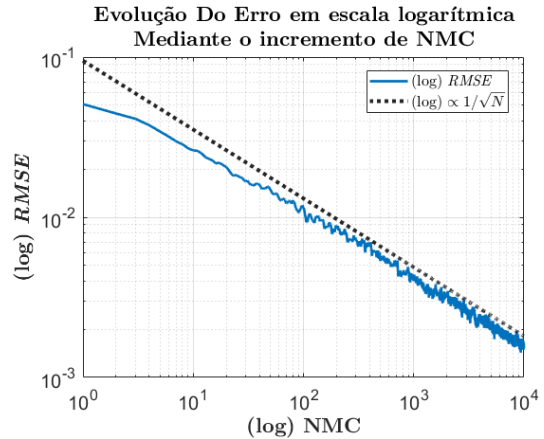
▲ Transita-se para x_1 de x_7 e para x_7 de x_6 :

$$\pi_1 = \frac{1}{2}\pi_7 = \frac{1}{2}\left(\frac{1}{2}\pi_6\right)$$

A validação dos resultados advindos da simulação e deduzidos teoricamente é naturalmente observada com as relações expostas acima.

ii) Convergência da distribuição das probabilidades dos diferentes estados com o aumento do número de *runs* de Monte Carlo.

A convergência da distribuição das probabilidades dos diferentes estados é estudada através do *root mean square error* (*RMSE*) acumulado ao longo de N *runs* de Monte Carlo (NMC)³:

(a) Evolução do erro ao longo de N *runs* de Monte Carlo

(b) Evolução do erro em escala logarítmica.

Fig. 5: Ilustração da *convergence rate* da Cadeia de Markov.→ **Observações**

▲ A *fitted curve* demonstra que o decaimento do erro tem um *rate* $\propto 1/\sqrt{N}$ (onde N são as *runs* de Monte Carlo) → O *convergence rate* dos diferentes estados é da ordem de $\mathcal{O}(1/\sqrt{N})$, realidade independente da dimensão da cadeia:

³Para *runs* de 200 jogadas totais.

“Monte Carlo’s convergence rate, $\mathcal{O}(1/\sqrt{N})$, is independent of dimension.”[5]

- ▲ A visualização dos eixos no espaço logarítmico (vide Fig. 5 b)) sugere que para valores elevados de N a aproximação à reta $\propto 1/\sqrt{N}$ torna-se cada vez mais refinada.

Estudou-se ainda a evolução do erro para *seeds* diferentes, de forma a variar o gerador de números aleatórios:

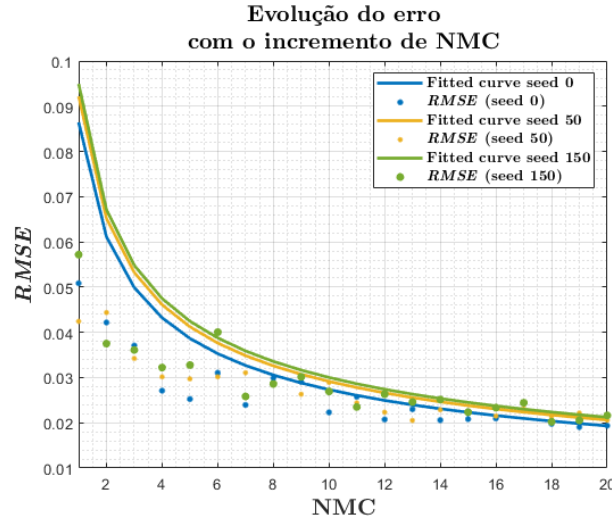


Fig. 6: Convergence rate para *seeds* diferentes.

→ Observações

- ▲ A variação do gerador de números aleatórios não afeta a evolução do erro mediante o número de NMC's: as *fitted curves* demonstram sempre que o decaimento do erro tem um $\text{rate} \propto 1/\sqrt{N}$. A não sobreposição aparente das *fitted curves* advém da sequência de números aleatórios distintos, consequentemente, valores de *RMSE* distintos.

iii) Valor adequado da variável *Ndiscard*.

Ndiscard, normalmente conhecido pelo termo *burn-in/warm-up*, destina-se a dar tempo suficiente à Cadeia de Markov para atingir o seu *steady-state/equilibrium distribution*, livre do *bias* inicial imposto pelo estado de partida:

“The idea is that a “bad” starting point may over-sample regions that are actually very low probability under the equilibrium distribution before it settles into the equilibrium distribution. If you throw those points away, then the points which should be unlikely will be suitably rare.”[6]

Procuramos um número de iterações grande o suficiente para garantir uma cadeia bem misturada, o que se traduz num *RMSE* (*root mean square error*) mínimo:

→ Ao realizar N simulações de Monte Carlo⁴, cada uma com *burn-in* incrementalmente diferente, pretendemos depreender uma gama de valores de $N_{discard}$ para o qual o erro é mínimo e estável.

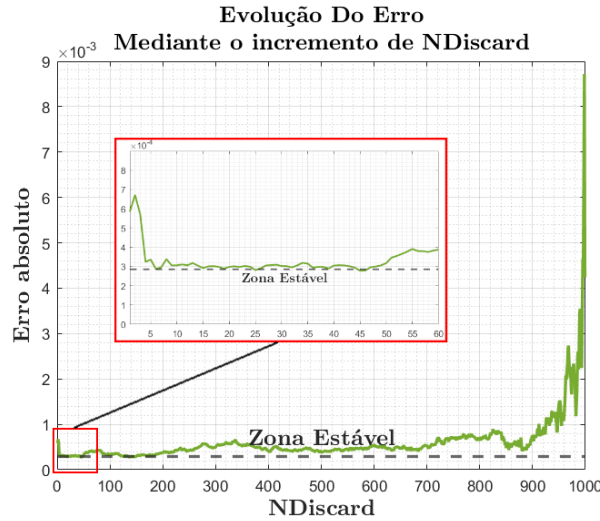


Fig. 7: Erro absoluto mediante o incremento de $N_{discard}$ para simulações de 1000000 *runs* e 1000 jogadas totais. (**Nota:** foi omitido o valor do erro absoluto para $N_{discard} = 1000$ de forma a uma melhor visualização do comportamento da evolução do erro. Tal será esperado nas subseqüentes figuras.)

→ Observações

- ▲ Para $N_{discard} < 5$ é identificada uma forte flutuação de erros → o regime transitório influencia os valores de frequência relativa estimados.
- ▲ Para $N_{discard} \in]5, 60]$ o erro calculado atinge uma estabilidade mínima correspondente à gama de valores já supramencionada.
- ▲ Para $N_{discard} > 60$ A regressão do erro torna-se progressivamente maior e mais instável, com um pico na gama $[900, 1000]$. Este comportamento é trivialmente explicado pela natureza da Cadeia de Markov: Ao descartar um número de jogadas próximo do seu valor total (1000), a ponderação da probabilidade tem por base um número de amostras reduzido. Reconhecendo que a *equilibrium distribution* da Cadeia de Markov só é atingida para um elevado valor de transições⁵ a estimativa da probabilidade apresenta maior incidência de erro.

Por outro lado, a literatura indica que:

“The amount that should be thrown away is usually less than 1% of a run whenever a run is long enough to give enough precision. So routinely throwing away the initial 1% or 2% of runs will usually suffice.”[1]

Logo é admitido um valor ótimo de $N_{discard}$ de 20.

→ Nota:

O método de Monte Carlo supõe um elevado número de *runs*⁶ para um valor de $N_{jogadas}$ satisfatório, podemos ainda admitir o estudo do comportamento do número de erros para duas instâncias diferentes, *Many short runs* e *Few long runs*:

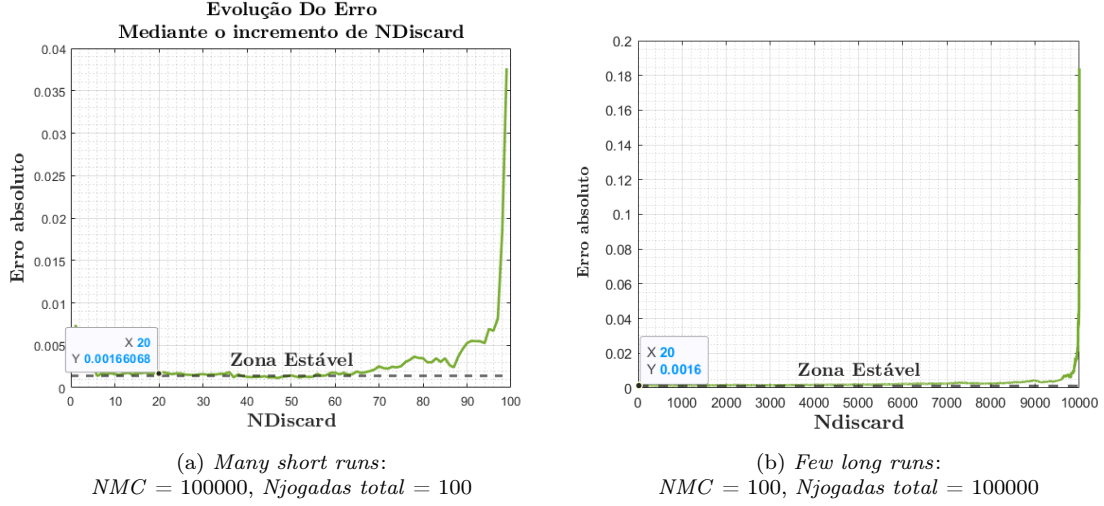


Fig. 8: Comportamento do número de erros para as duas instâncias.

Embora o erro para um $Ndiscard$ de 20 seja satisfatório para as duas instâncias já acima referidas é importante compreender que o valor de *burn-in* escolhido é totalmente inerente ao tipo de simulação realizada⁷. Para o P1 são admitidas simulações de 1000000 *runs*, 1000 *Njogadas total* e 20 *Ndiscard* (**Many short runs**) já que a *convergence rate* é de $O(1/\sqrt{N})$ onde N é o número de *runs* de Monte Carlo.

iv) Outros aspetos importantes.

→ Métodos de diagnóstico

Na literatura, a estimação do *burn-in*/*warm-up*, bem como, da região de convergência da Cadeia de Markov é efetuada mediante diversos padrões de diagnóstico, tais como: *trace plot*, *autocorrelation*... Infelizmente não abordados, devido à *scope* do problema.

→ Fórmulas de erro escolhidas

$$RMSE \triangleq \sqrt{\frac{1}{NMC} \sum_{n=1}^{NMC} \|\hat{\pi}_n - \pi_{teo.}\|}$$

$$Erro\ absoluto \triangleq \sum_{j=1}^{Ncasas} |\hat{\pi}_j - \pi_{teo.,j}|$$

O *RMSE* foi escolhido para visualizar a *convergence rate* da distribuição das probabilidades dos diferentes estados dado que pode ser visto como a distância (assemelhando-se à Euclidiana, mas *rescaled/normalized*) entre o vetor simulado e o vetor teórico esperado.

O erro absoluto propiciou uma melhor visualização da região de transição prevista na evolução do erro mediante o incremento de *Ndiscard*.

⁴Entendemos por simulação uma simulação onde decorrem n *runs* de Monte Carlo.

⁵Vide secção introdutória.

⁶Procuramos simular a Cadeia de Markov NMC (N de Monte Carlo, número de *runs*) vezes.

⁷Tipo de Cadeia de Markov, número de *runs* e número de *Njogadas total*.

P3 | Probabilidade de estar no estado 4 ao longo das sucessivas jogadas.

A evolução da probabilidade do estado 4 é avaliada para um valor fixo de $N_{jogadas}$ (50) e um valor sucessivamente maior de $runs$ de Monte Carlo:

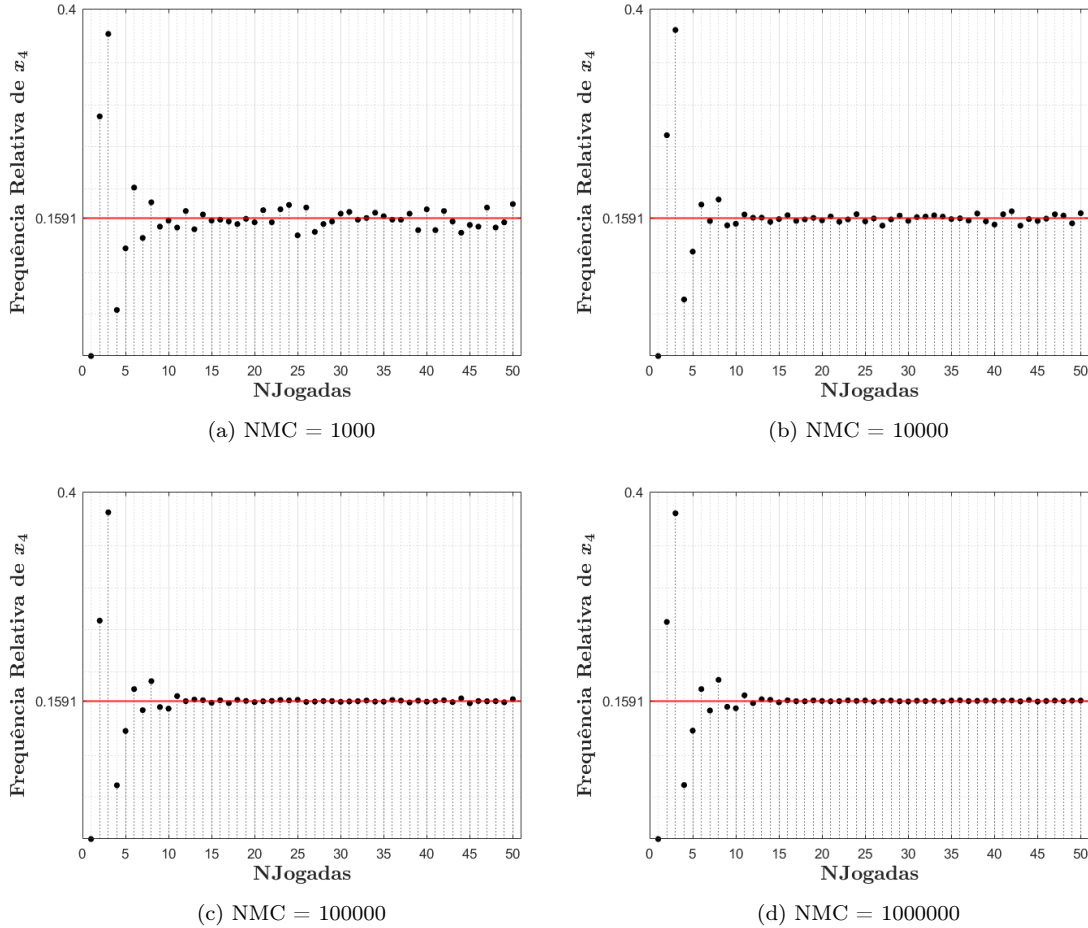


Fig. 9: Evolução da probabilidade do estado 4

→ Observações

- ▲ Para a primeira jogada a probabilidade de estar no estado 4 é sempre nula, condição imposta pela casa de partida (apenas é possível transitar para o estado 1 ou 2).
- ▲ A convergência da probabilidade para o seu *steady-state value* é tanto mais rápida e mais exata quanto maior for o número de *runs* de Monte Carlo (relembremos o *convergence rate* de $\mathcal{O}(1/\sqrt{N})$, vide secção P2 ii).
- ▲ O número de $N_{jogadas}$ não necessita de ser muito grande para atingir uma boa estimativa do *steady-state value*, sendo o número de *runs* o parâmetro de maior peso nesta tarefa, reforçando novamente o uso de *Many short runs*.

P4 | Diagrama de transição modificado.

De modo a emular a permanência no estado prisão durante uma jogada, acrescenta-se um estado *dummy*, denominado por x_P , para o qual se transita, (possivelmente) dos estados 5 ou 6. Nestes moldes, uma jogada é desperdiçada entre a transição obrigatória do estado x_P para o estado 3 (transição esta independente do resultado do lançamento da moeda).

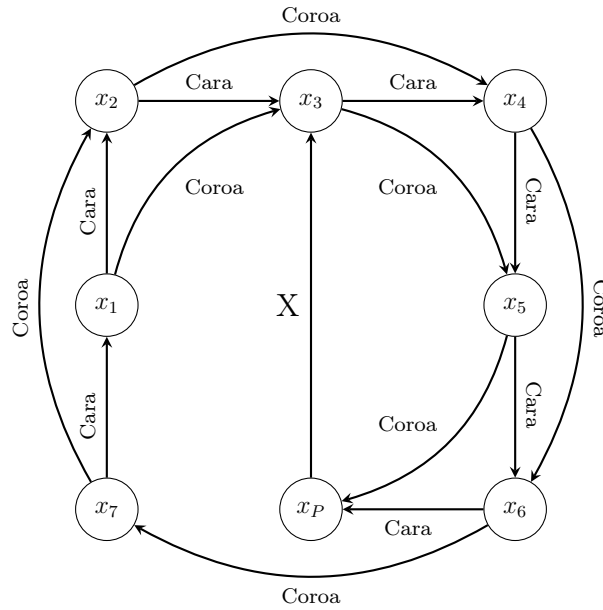


Fig. 10: Diagrama de transição de estados modificado com a introdução de um *dummy state*, de modo a descartar uma possível jogada, na eventualidade do jogador ser enviado para a prisão.

Seguindo o mesmo raciocínio, é trivialmente generalizado o esquema para um número arbitrário de jogadas em que se pretende que o jogador permaneça no estado prisão. Esta façanha é concretizável introduzindo os *dummy states* desejáveis entre o estado x_P e o estado 3, de forma a gastar jogadas com estas transições impostas (sempre independentes do lançamento da moeda).

Referências

- [1] Charles J. Geyer. Practical Markov Chain Monte Carlo. *Statistical Science*, 7(4), November 1992. doi: 10.1214/ss/1177011137. URL <https://doi.org/10.1214/ss/1177011137>.
- [2] Christos G. Cassandras and Stéphane Lafortune. *Introduction to Discrete Event Systems*, pages 369–428. Springer, 2 edition, 2008. ISBN 978-0-387-68612-7. Chapter 7.
- [3] J. Medhi. Stochastic Processes. In *Stochastic Models in Queueing Theory*, pages 1–46. Elsevier, 2003. doi: 10.1016/b978-012487462-6/50001-1. URL <https://doi.org/10.1016/b978-012487462-6/50001-1>.
- [4] David G. Luenberger. *Introduction to Dynamic Systems*, pages 224–253. John Wiley & Sons, 1979. ISBN 978-0-471-02594-8. Chapter 7.
- [5] Russel E. Caflisch. Monte Carlo and quasi-monte Carlo methods: Acta Numerica, Nov 2008. URL <https://www.cambridge.org/core/journals/acta-numerica/article/abs/monte-carlo-and-quasimonte-carlo-methods/FE7C779B350CFEA45DB2A4CCB2DA9B5C>.
- [6] Amelio Vazquez, Matt KrauseMatt, and Eric PetersonEric. "MCMC methods - burning samples?", Apr 2015. URL <https://stats.stackexchange.com/questions/88819/mcmc-methods-burning-samples>.