Aprendizado por Reforço

Métodos de diferenças temporais – parte 2

Panorama dos Algoritmos TD

- TD (valores de estado)
  - Estima os valores de estado de uma dada política.
  - É o algoritmo base para os demais métodos TD.
- Sarsa
  - Estima valores de ação de uma dada política.
  - Pode ser combinado com melhoria de política para encontrar políticas ótimas.
- n-step Sarsa
  - Generalização do Sarsa.
  - Sarsa e MC são casos particulares do n-step Sarsa.
- Q-learning
  - Algoritmo clássico de aprendizado por reforço.
  - Busca resolver diretamente a equação de otimalidade de Bellman para encontrar políticas ótimas.

Atualização incremental dos valores de estado

- O algoritmo TD atualiza apenas o valor do estado visitado s_t:

$$v_{t+1}(s_t) = v_t(s_t) - \alpha_t(s_t)[v_t(s_t) - (r_{t+1} + \gamma v_t(s_{t+1}))]$$

- Para todos os outros estados não visitados, os valores permanecem inalterados:

$$v_{t+1}(s) = v_t(s), \qquad \forall s \neq s_t$$

  - $t = 0,1,2,\ldots$ (tempo)
  - $v_t(s_t)$: estimativa de $v_\pi(s_t)$ no tempo t
  - $\alpha_t(s_t)$: taxa e aprendizagem para s_t no tempo t

Sarsa

- O algoritmo TD visto na aula passada estima apenas os **valores de estado.**
- O algoritmo Sarsa: estima **valores de ação.**

Por que a estimação de valores de ação é importante?

Sarsa

- O algoritmo TD visto na aula passada estima apenas os **valores de estado.**
- O algoritmo Sarsa: estima **valores de ação.**
- **Por que a estimação de valores de ação é importante?**

> A estimação de **valores de ação** é essencial porque pode ser combinada com etapas de **melhoria de política**, permitindo o aprendizado de **políticas ótimas.**

Sarsa

- Dado uma política π, o objetivo é estimar os valores de ação.
- Considere as amostras de experiência coletadas ao seguir a política π:

$$(s_0, a_0, r_1, s_1, a_1, \dots, s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1}, \dots)$$

- A atualização ocorre apenas para o par visitado (s_t, a_t):

$$q_{t+1}(s_t, a_t) = q_t(s_t, a_t) - \alpha_t(s_t, a_t)\left[q_t(s_t, a_t) - \left(r_{t+1} + \gamma q_t(s_{t+1}, a_{t+1})\right)\right]$$

  - $t = 0,1,2, \dots$
  - $q_t(s_t, a_t)$: estimativa de $q_\pi(s_t, a_t)$
  - $\alpha_t(s_t, a_t)$: taxa de aprendizado

- Todos os demais pares permanecem inalterados:

$$q_{t+1}(s, a) = q_t(s, a), \qquad \forall (s, a) \neq (s_t, a_t)$$

Sarsa

- Origem da nomenclatura: sequência usada em cada iteração

$$(s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1})$$

  - Sarsa: abreviação do termo em inglês *state-action-reward-state-action*.

- Essa sequência resume a estrutura fundamental da experiência utilizada na atualização.

Sarsa

- Aproximação estocástica para resolver uma equação de Bellman (expressa em termos de valores de ação) para uma dada política π:

$$q_\pi(s,a) = \mathbb{E}[R + \gamma q_\pi(S',A') \mid s,a]$$

- Equação de Bellman para valores de ação

$$q_\pi(s,a) = \sum_r r p(r|s,a) + \gamma \sum_{s'} \sum_{a'} q_\pi(s',a') p(s'|s,a) \pi(a'|s')$$

$$q_\pi(s,a) = \sum_r r p(r|s,a) + \gamma \sum_{s'} p(s'|s,a) \sum_{a'} q_\pi(s',a') \pi(a'|s')$$

Sarsa

- Como

$$p(s',a'|s,a) = p(s'|s,a)p(a'|s',s,a)$$

$$p(s',a'|s,a) = p(s'|s,a)p(a'|s')$$

$$p(s',a'|s,a) = p(s'|s,a)\pi(a'|s')$$

- Então

Equação de Bellman
$q_\pi(s,a) = \mathbb{E}[R + \gamma q_\pi(S',A') \mid s,a]$

$$q_\pi(s,a) = \sum_r rp(r|s,a) + \gamma \sum_{s'} \sum_{a'} q_\pi(s',a')p(s',a'|s,a)$$

Sarsa

- Convergência

Dada uma política π, pelo algoritmo Sarsa, $q_t(s,a)$ converge quase certamente para o valor de ação $q_\pi(s,a)$ quando $t \rightarrow \infty$, para todo (s,a), se

$$\sum_t \alpha_t(s,a) = \infty \qquad e \qquad \sum_t \alpha_t^2(s,a) < \infty\,, \qquad \forall(s,a)$$

Sarsa - Aprendizado de políticas ótimas

- O algoritmo Sarsa abaixo apenas estima os valores de ação para uma dada política:

$$q_{t+1}(s_t, a_t) = q_t(s_t, a_t) - \alpha_t(s_t, a_t)\left[q_t(s_t, a_t) - \left(r_{t+1} + \gamma q_t(s_{t+1}, a_{t+1})\right)\right]$$

- Para encontrar políticas ótimas, ele é combinado com um passo de **melhoria de política**.

- Essa combinação também é chamada de Sarsa e é implementada em **dois passos por iteração**.

Sarsa - Aprendizado de políticas ótimas

- Passos por iteração (seguindo a ideia da iteração generalizada de política):

1. Atualização do valor de ação do par estado-ação visitado.
2. Atualização da política tornando-a ϵ-gulosa.
    - A política não é completamente avaliada antes de atualizá-la.
    - A nova política é usada imediatamente para gerar a próxima amostra de experiência.
    - A política é ϵ-gulosa para permitir a exploração.

Sarsa - Algoritmo

Algorithm 7.1: Optimal policy learning by Sarsa

Initialization: $\alpha_t(s,a) = \alpha > 0$ for all (s,a) and all t. $\epsilon \in (0,1)$. Initial $q_0(s,a)$ for all (s,a). Initial ϵ-greedy policy π_0 derived from q_0.
Goal: Learn an optimal policy that can lead the agent to the target state from an initial state s_0.

For each episode, do
    Generate a_0 at s_0 following $\pi_0(s_0)$
    If s_t $(t = 0, 1, 2, \dots)$ is not the target state, do
        Collect an experience sample $(r_{t+1}, s_{t+1}, a_{t+1})$ given (s_t, a_t): generate r_{t+1}, s_{t+1} by interacting with the environment; generate a_{t+1} following $\pi_t(s_{t+1})$.
        Update q-value for (s_t, a_t)*:*

$$q_{t+1}(s_t,a_t) = q_t(s_t,a_t) - \alpha_t(s_t,a_t)\Big[q_t(s_t,a_t) - (r_{t+1} + \gamma q_t(s_{t+1},a_{t+1}))\Big]$$

        Update policy for s_t*:*

$$\pi_{t+1}(a|s_t) = 1 - \frac{\epsilon}{|\mathcal{A}(s_t)|}(|\mathcal{A}(s_t)| - 1) \text{ if } a = \arg\max_a q_{t+1}(s_t,a)$$

$$\pi_{t+1}(a|s_t) = \frac{\epsilon}{|\mathcal{A}(s_t)|} \text{ otherwise}$$

        $s_t \leftarrow s_{t+1}$, $a_t \leftarrow a_{t+1}$

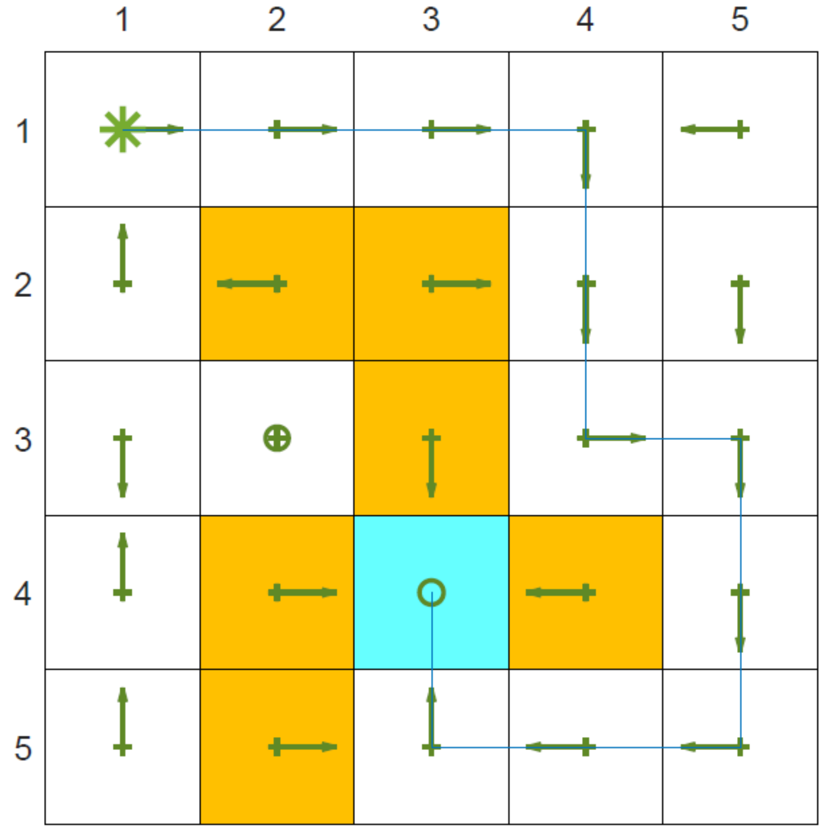
Sarsa – Exemplo no mundo em grade

- Objetivo: encontrar o caminho ótimo entre um estado inicial e um estado final (ambos fixos).
  - Precisamos explorar somente estados próximos ao caminho e não todos os estados.
  - Porém, se o agente não explorar todos os estados o caminho final pode ser um ótimo local ao invés de ótimo global.
  - Recompensas:
    - $r_{alvo} = 0$
    - $r_{proibido} = -10$
    - $r_{fronteira} = -10$
    - $r_{outros} = -1$
  - Parâmetros:
    - $\alpha_t(s_t, a_t) = 0.1$
    - $\epsilon = 0.1$
    - $q_0(s, a) = 0, \ \ \forall(s, a)$
    - $\pi_0(s, a) = 0.2, \ \ \forall(s, a)$ (distribuição uniforme)

Sarsa - Exemplo

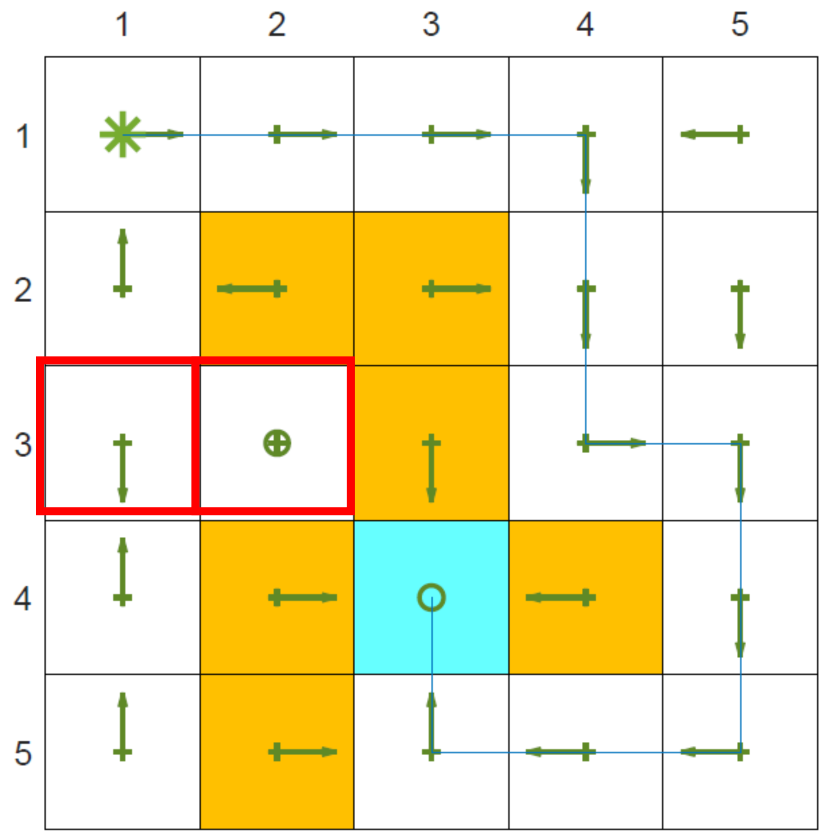
- Política Aprendida

O que podemos observar?



Sarsa - Exemplo

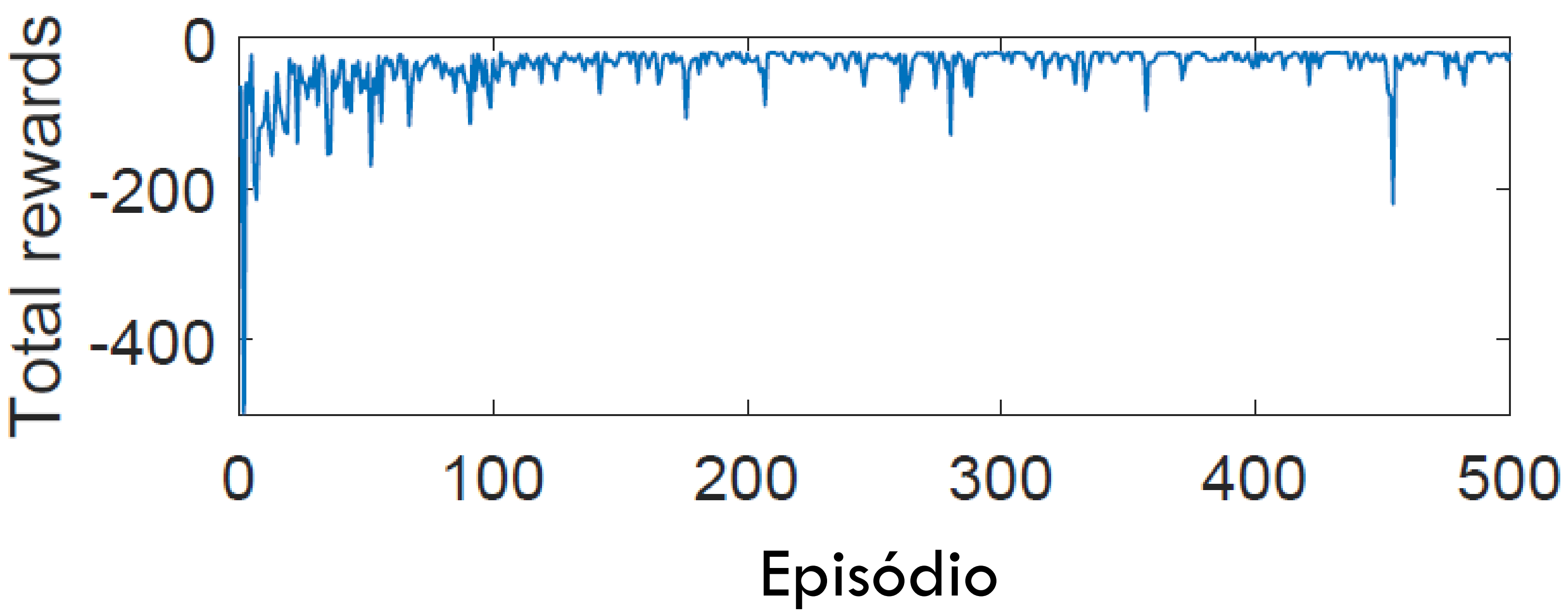
- Política Aprendida
  - Consegue guiar o agente do estado inicial ao estado alvo.
  - Nos estados pouco explorados ações ótimas podem não ser aprendidas.



Sarsa - Exemplo

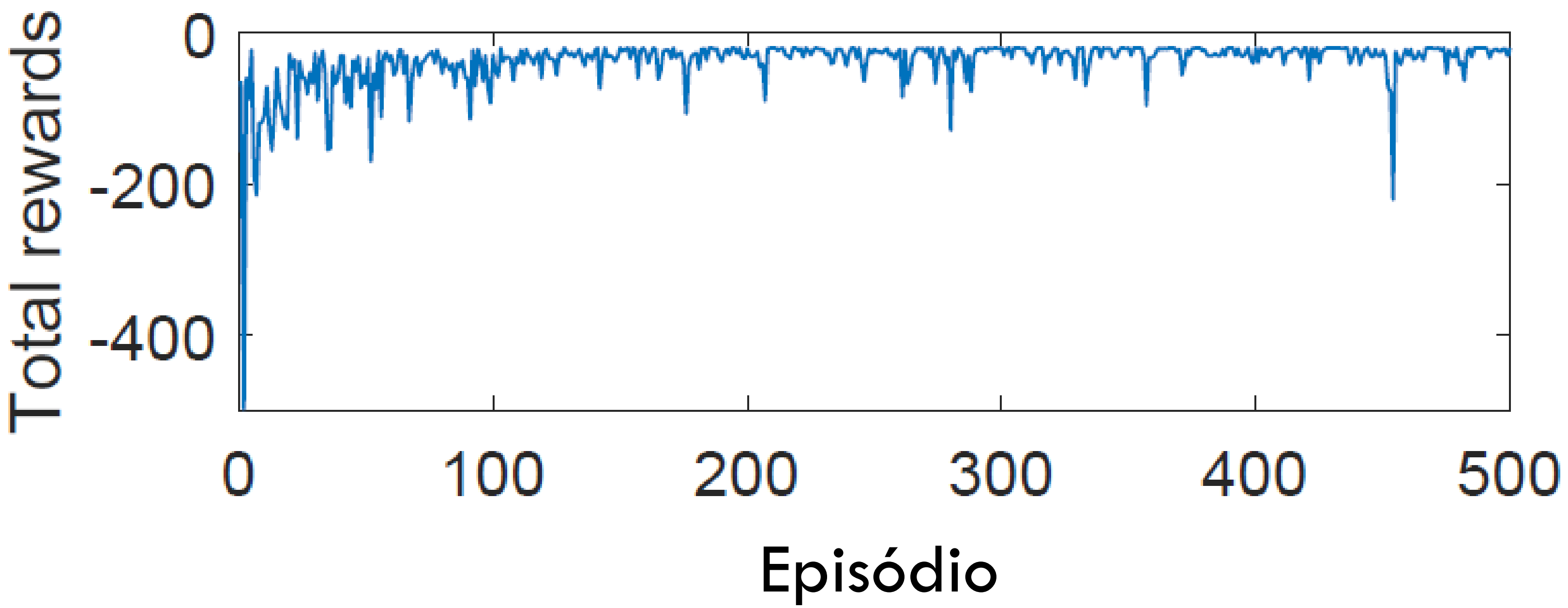
- Recompensa total por episódio

O que podemos observar?



Sarsa - Exemplo

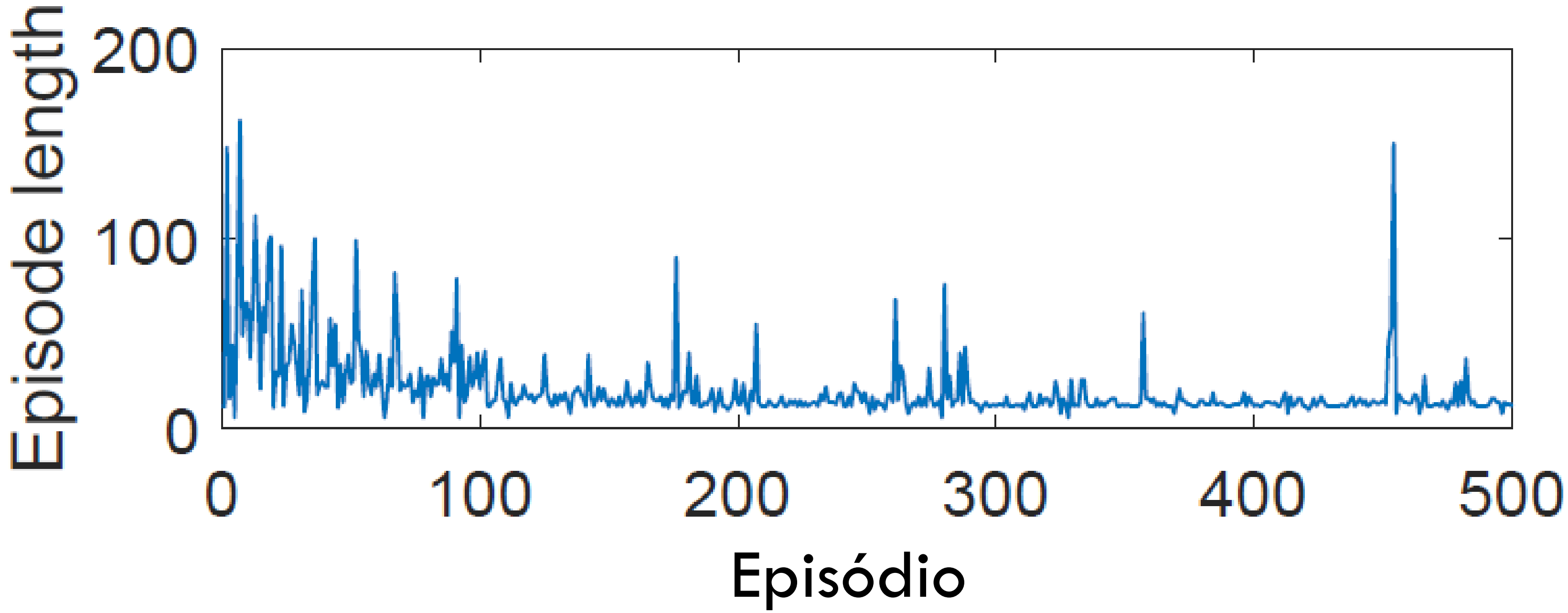
- Recompensa total por episódio
  - Ao longo dos episódios, o retorno aumenta gradualmente.
    - A política inicial é ineficiente e resulta em penalizações.
    - A melhoria da política melhora eleva o retorno.



Sarsa - Exemplo

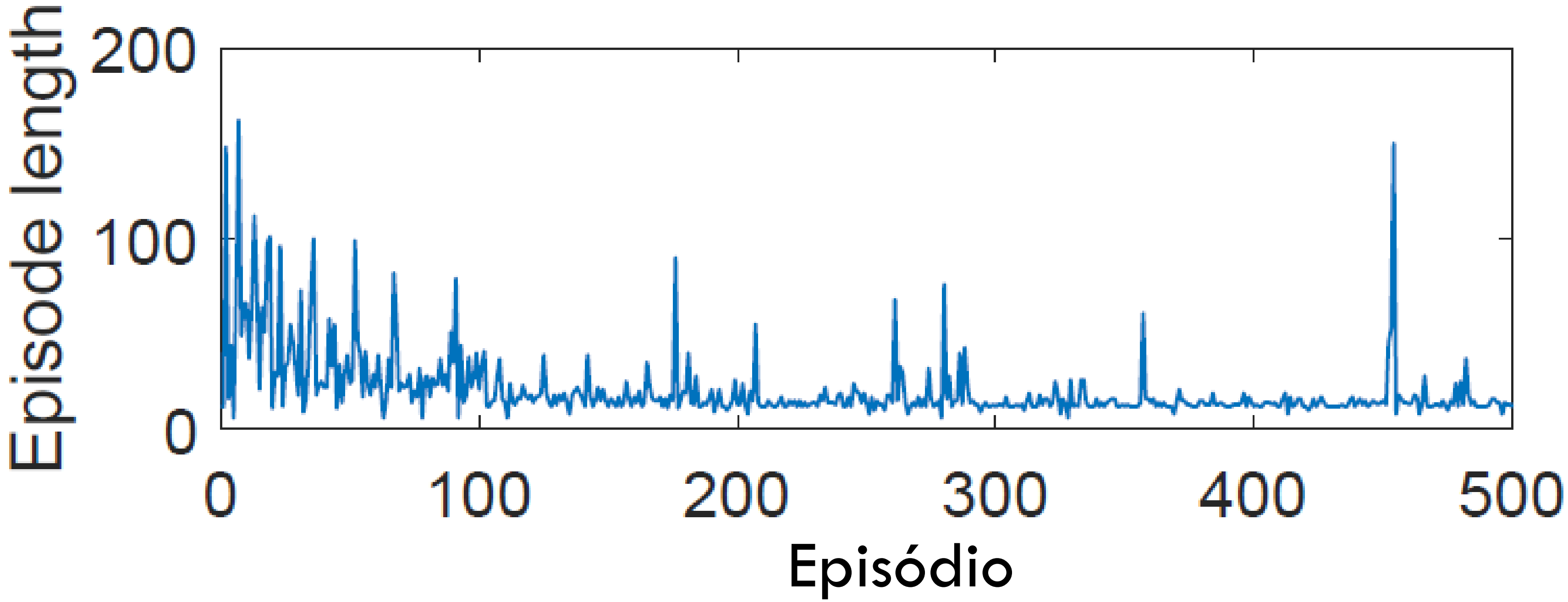
- Duração dos episódios

O que podemos observar?



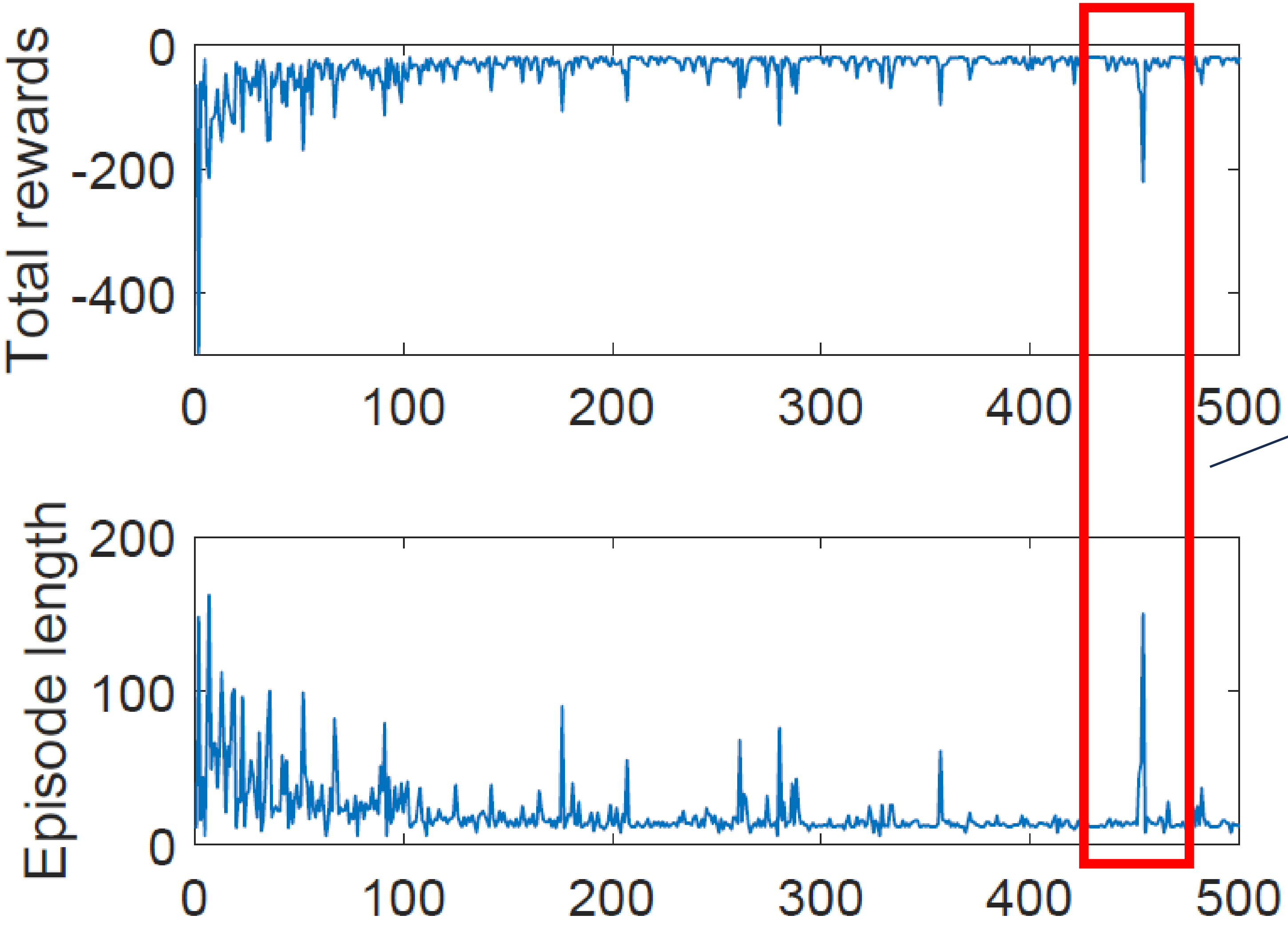
Sarsa - Exemplo

- Duração dos episódios
  - A duração dos episódios diminui progressivamente
    - A política inicial é ineficiente e pode resultar em trajetórias mais longas.
    - A melhoria da política reduz a duração dos episódios.



Sarsa - Exemplo

- Problema

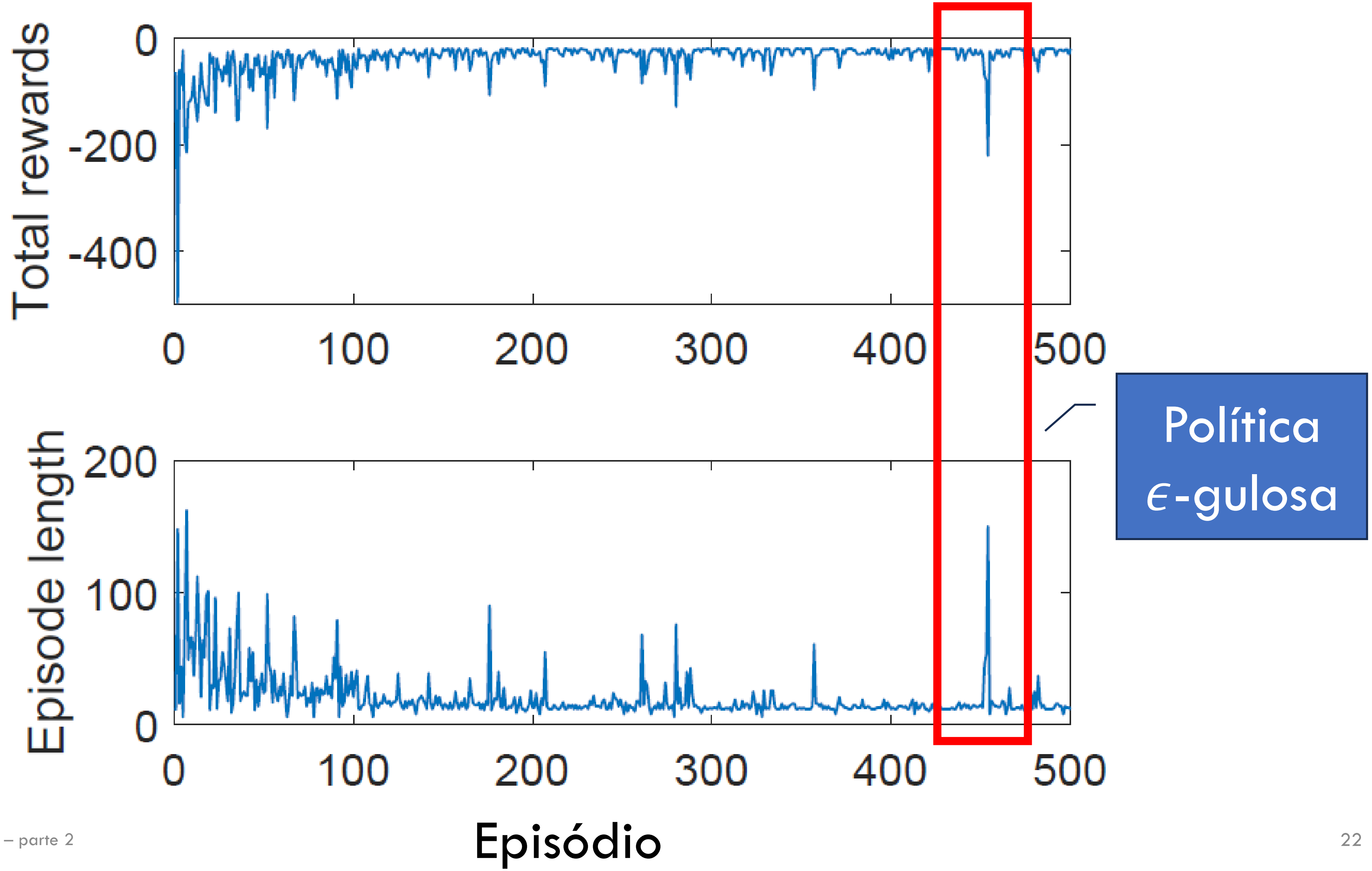


Episódio

?

Sarsa - Exemplo

- Problema
  - Pode ser mitigado usando um ϵ decrescente, cujo valor converge gradualmente para zero.



Expected Sarsa

- Variante do Sarsa usada para avaliar valores de ação de uma política π.
- Ideia-chave: trocar o alvo baseado em **uma única ação amostrada** por um **valor esperado** sobre todas as ações do próximo estado.
- Regra de atualização
  - Par visitado (s_t, a_t):

$$q_{t+1}(s_t, a_t) = q_t(s_t, a_t) - \alpha_t(s_t, a_t)[q_t(s_t, a_t) - (r_{t+1} + \gamma \mathbb{E}[q_t(s_{t+1}, A)])]$$

$$\mathbb{E}[q_t(s_{t+1}, A)] = \sum_a \pi(a \mid s_{t+1}) q_t(s_{t+1}, a) \triangleq v_t(s_{t+1})$$

  - Todos os demais pares $(s, a) \neq (s_t, a_t)$:

$$q_{t+1}(s, a) = q_t(s, a)$$

Expected Sarsa

- O alvo de Expected Sarsa é um **valor médio,** eliminando a variabilidade da amostra a_{t+1}.

Algoritmo	Alvo TD
Sarsa	$r_{t+1} + \gamma q_t(s_{t+1}, a_{t+1})$
Expected Sarsa	$r_{t+1} + \gamma \mathbb{E}[q_t(s_{t+1}, A)]$

- Tem um custo computacional adicional.
- Vantagens
  - Reduz o conjunto de variáveis aleatórias de $\{s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1}\}$ para $\{s_t, a_t, r_{t+1}, s_{t+1}\}$.
  - Menor variância na estimativa das atualizações.

Expected Sarsa

- Expected Sarsa pode ser visto como uma aproximação estocástica para resolver a equação de Bellman:

$$q_\pi(s,a) \;=\; \mathbb{E}[R_{t+1} + \gamma \mathbb{E}[q_\pi(S_{t+1}, A_{t+1}) | S_{t+1}] \mid S_t = s, A_t = a]$$

Mas,

$$\mathbb{E}[q_\pi(S_{t+1}, A_{t+1}) | S_{t+1}] = \sum_{A'} q_t(S_{t+1}, A')\pi(A' \mid S_{t+1}) = v_t(S_{t+1})$$

Então chegamos à equação de Bellman para valores de ação:

$$\boxed{q_\pi(s,a) \;=\; \mathbb{E}[R_{t+1} + \gamma v_t(S_{t+1}) \mid S_t = s, A_t = a]}$$

n-step Sarsa

- Extensão multietapa do Sarsa usada para avaliar valores de ação de uma política π com um compromisso controlável entre viés e variância.
- Utiliza n recompensas futuras.
- Reúne, num mesmo quadro teórico, Sarsa e Monte Carlo.

n-step Sarsa

- Definição de valor de ação:

$$q_\pi(s,a) \;=\; \mathbb{E}[G_t \mid S_t = s, A_t = a]$$

Onde,

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$$

n-step Sarsa

- Decomposição do retorno G_t

$$
\begin{array}{lcl}
\text{Sarsa} & \leftarrow & G_t^{(1)} = R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \\
 & & G_t^{(2)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 q_\pi(S_{t+2}, A_{t+2}) \\
 & & \vdots \\
\text{n-step Sarsa} & \leftarrow & G_t^{(n)} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^n q_\pi(S_{t+n}, A_{t+n}) \\
 & & \vdots \\
\text{Monte Carlo} & \leftarrow & G_t^{(\infty)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots
\end{array}
$$

- Note que $G_t = G_t^{(1)} = G_t^{(2)} = \cdots = G_t^{(n)} = \cdots = G_t^{(\infty)}$.

n-step Sarsa

- $n = 1$

$$q_\pi(s,a) = \mathbb{E}\left[G_t^{(1)} \mid s,a\right] = \mathbb{E}[R_{t+1} + \gamma \mathrm{q}_\pi(S_{t+1}, A_{t+1}) \mid s,a]$$

- Algoritmo de aproximação estocástica para resolver a equação acima: Sarsa

$$q_{t+1}(s_t,a_t) = q_t(s_t,a_t) - \alpha_t(s_t,a_t)\left[q_t(s_t,a_t) - \left(r_{t+1} + \gamma q_t(s_{t+1},a_{t+1})\right)\right]$$

n-step Sarsa

- $n = \infty$

$$q_\pi(s,a) = \mathbb{E}\left[G_t^{(\infty)} \mid s,a\right] = \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \mid s,a]$$

  - Algoritmo para resolver a equação acima: Monte Carlo

$$q_{t+1}(s_t,a_t) = g_t = \sum_{k=0}^{\infty} \gamma^k r_{t+1+k}$$

n-step Sarsa

- Para um n qualquer:

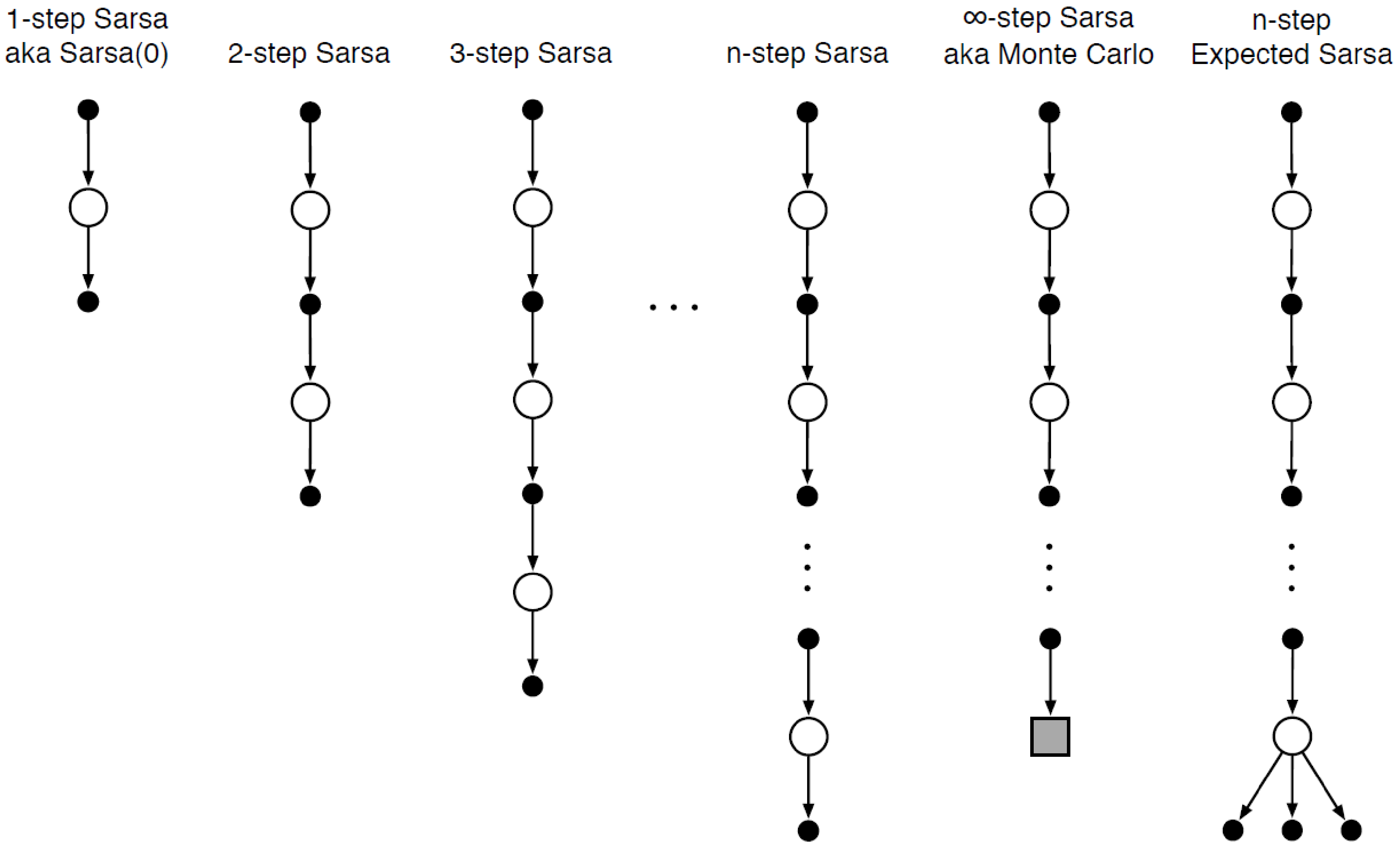
$$q_\pi(s,a) = \mathbb{E}\left[G_t^{(n)} \mid s,a\right] = \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^n q_\pi(S_{t+n}, A_{t+n}) \mid s,a]$$

  - Algoritmo para resolver a equação acima: n-step Sarsa

$$q_{t+1}(s_t,a_t) = q_t(s_t,a_t) - \alpha_t(s_t,a_t)\left[q_t(s_t,a_t) - \left(r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^n q_t(s_{t+n}, a_{t+n})\right)\right]$$

- n-step Sarsa inclui o Sarsa (para $n = 1$) e o MC (para $n = \infty$ e $\alpha_t = 1$) como casos extremos.

n-step Sarsa


1-step Sarsa
aka Sarsa(0)
2-step Sarsa
3-step Sarsa
n-step Sarsa
∞-step Sarsa
aka Monte Carlo
n-step
Expected Sarsa

n-step Sarsa

- Implementação (atraso de n passos)
  - Amostras de experiência necessárias: $(s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1}, \ldots, r_{t+n}, s_{t+n}, a_{t+n})$
  - O valor de ação para o par (s_t, a_t) só poderá ser atualizado no instante $t + n$.
  - Mantém-se um buffer de tamanho n com transições recentes.
  - Fórmula operacional (tempo $t + n$):

$$q_{t+n}(s_t, a_t) = q_{t+n-1}(s_t, a_t) - \alpha_{t+n-1}(s_t, a_t)\delta_t^{(n)}$$

Onde

$$\delta_t^{(n)} = \left[q_{t+n-1}(s_t, a_t) - \left(r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^n q_{t+n-1}(s_{t+n}, a_{t+n})\right)\right]$$

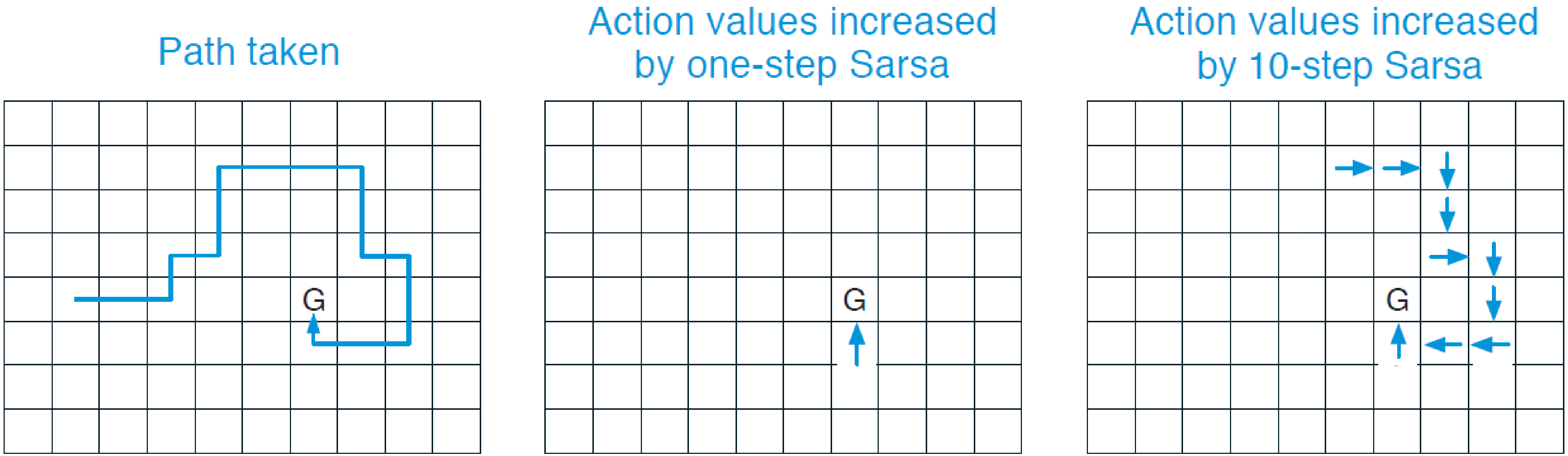
n-step Sarsa

- Viés × Variância
  - n pequeno
    - Baixa variância, maior viés (parecido com Sarsa)
  - n grande
    - Alta variância, baixo viés (aproxima MC)

- O n-step Sarsa descrito faz apenas avaliação de uma dada política π.
  - Para aprender políticas ótimas, adiciona-se, após cada atualização, um **passo de melhoria de política.**

n-step Sarsa

- Exemplo: agente percorre um mundo em grade.
  - A trajetória de um único episódio termina no estado de alta recompensa marcado por G.



n-step Sarsa

- No método de 1 passo, apenas a última ação da sequência que levou até G tem seu valor reforçado.
- No método n passos, o reforço retrocede pelos n últimos passos da mesma trajetória.
- Tem-se um maior aprendizado a partir do mesmo episódio, acelerando o aprendizado da política.

Referências

- S. Zhao. *Mathematical Foundations of Reinforcement Learning*. Springer Singapore, 2025. [capítulo 7]
  - **disponível em:** https://github.com/MathFoundationRL/Book-Mathematical-Foundation-of-Reinforcement-Learning

- R. S. Sutton e A. G. Barto. *An Introduction Reinforcement Learning*, Bradford Book, 2018. [capítulos 6 e 7]
  - **disponível em:** http://incompleteideas.net/book/the-book-2nd.html

Slides construídos com base nos livros supracitados, os quais estão disponibilizados publicamente pelos seus respectivos autores.