

T2 - Aprendizado por Reforço

Gridworld com Perigos

Ainaras Marão (182338), Douglas H. R. A. Pereira (245202), Matheus Veiga Luchi (269494),
Rafael Carro Gaudim (240879) e Yan Oliveira da Silva (236363)

Resumo

Este trabalho apresenta a modelagem de um problema de Gridworld com perigos como um Processo de Decisão de Markov e sua resolução por Value Iteration e Q-Learning. O agente deve coletar itens distribuídos no mapa, respeitar a capacidade da mochila e retornar à base maximizando a recompensa, enquanto evita armadilhas. A avaliação experimental investigou o efeito do fator de desconto, das estratégias de exploração, da convergência e da estabilidade entre sementes. Os resultados mostram que $\gamma = 0.99$ produz melhor desempenho final, enquanto $\gamma = 0.9$ converge mais rapidamente. O planejamento por Bellman gera políticas mais estáveis no mapa conhecido, ao passo que o Q-Learning depende mais da exploração e da variabilidade amostral.

1 Modelagem do Problema

O problema foi modelado como um Processo de Decisão de Markov (*Markov Decision Process* – MDP), definido pela tupla (S, A, T, R) , em que S representa o conjunto de estados, A o conjunto de ações, $T(s, a, s')$ a dinâmica de transição e $R(s, a, s')$ a recompensa imediata. Essa formulação segue a modelagem clássica de problemas sequenciais de decisão sob incerteza.

O ambiente é uma grade bidimensional com bordas bloqueadas por paredes, armadilhas, itens colecionáveis e uma base de entrega. O agente inicia sempre na mesma posição de spawn e deve coletar todos os itens disponíveis, respeitando a capacidade da mochila, para então retorná-los ao ponto de entrega. O objetivo é maximizar a recompensa acumulada total, equilibrando ganho por coleta e entrega com o custo de deslocamento e o risco de atravessar regiões perigosas.

Cada estado é representado pela tupla $s = (x, y, w, m)$, em que (x, y) representa a posição do agente, w é a quantidade de itens atualmente na mochila, e m é uma bitmask que indica quais itens ainda permanecem disponíveis no mapa. Essa representação foi escolhida para distinguir explicitamente situações com mesma posição, mas com cargas ou conjuntos de itens remanescentes diferentes. A capacidade da mochila foi fixada em 3. As ações possíveis são os quatro movimentos cardeais: $A = \{\text{LEFT}, \text{UP}, \text{RIGHT}, \text{DOWN}\}$.

Movimentos que levariam o agente a uma parede são considerados inválidos. Em cada passo, o agente pode sofrer o efeito estocástico de queda em armadilha: se houver armadilhas adjacentes, existe uma probabilidade de transição forçada para uma delas, proporcional à quantidade de perigos vizinhos. Assim, embora o deslocamento escolhido seja determinístico, a dinâmica completa do ambiente não é totalmente determinística.

Quando o agente entra em uma célula contendo item e ainda há espaço na mochila, o item é coletado automaticamente e removido da bitmask de disponibilidade. Ao chegar ao ponto de entrega, todos os itens transportados são descarregados. Um episódio termina em três situações: todos os itens foram entregues com sucesso, o limite máximo de passos foi excedido, ou o ambiente foi configurado com terminalidade por armadilha. Na configuração final adotada, a armadilha não encerra automaticamente o episódio, mas produz uma penalização severa.

A função de recompensa combina vários componentes. Cada movimento sofre um custo fixo de deslocamento ($wd = 10$), o que incentiva trajetórias curtas. Itens possuem recompensa positiva, com bônus adicional quando coletados com a mochila parcialmente cheia, favorecendo rotas de coleta eficientes. A entrega de itens na base também gera bônus crescente com a quantidade descarregada, e a conclusão bem-sucedida do episódio concede uma recompensa final alta ($\text{RETURN_REWARD} = 5000$). Em contrapartida, armadilhas geram recompensas negativas e exceder o horizonte de passos produz a penalização $\text{MAX_STEPS_PENALTY} = -1000$. Esse desenho de recompensas favorece políticas objetivas, que conciliam coleta eficiente, retorno oportuno à base e redução do risco associado às regiões perigosas.

2 Equações de Bellman

Para o planejamento em ambiente conhecido, foi utilizado o algoritmo de Value Iteration. A atualização de Bellman aplicada sobre a função de valor foi:

$$V(s) \leftarrow \max_a \sum_{s'} T(s, a, s') [R(s, a, s') + \gamma V(s')] \quad (1)$$

Para cada estado válido (x, y, w, m) , o algoritmo constrói as transições e recompensas esperadas de cada ação considerando tanto o deslocamento principal quanto a probabilidade de queda em armadilha quando há perigos adjacentes. Em seguida, calcula-se iterativamente: $Q_a(s) = R_a(s) + \gamma P_a V$ para cada ação a , e atualizar $V(s)$ pelo máximo entre os quatro valores de ação.

O critério de convergência adotado foi: $\max_s |V_{k+1}(s) - V_k(s)| < \theta$ com $\theta = 10^{-4}$. Nos experimentos finais, o planejamento convergiu em aproximadamente 110 iterações no mapa-base utilizado para comparação com o Q-Learning. Esse número é significativamente menor do que o número de episódios necessários para que o aprendizado por experiência estabilize, o que era esperado, já que Value Iteration utiliza conhecimento explícito do modelo de transição.

A implementação utilizou o algoritmo de Value Iteration armazenando as matrizes R , P , V como matrizes esparsas, para evitar armazenamento de transições não utilizadas. A implementação matricial com as bibliotecas *numpy* e *scipy* possibilitou a paralelização dos cálculos a nível de CPU. Após a convergência a política é obtida selecionando, em cada estado, a ação de maior valor esperado, permitindo comparar diretamente o comportamento planejado com o aprendido por interação, sob o mesmo ambiente e critérios de avaliação.

3 Q-Learning

O aprendizado por experiência foi implementado por meio do algoritmo Q-Learning. Para cada transição observada durante a interação com o ambiente, a atualização aplicada foi:

$$Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q(s, a)] \quad (2)$$

Os valores $Q(s, a)$ são atualizados apenas para estados efetivamente visitados durante o treinamento, o que torna a abordagem tabular viável para o tamanho do ambiente estudado. A estratégia de exploração adotada foi ϵ -greedy. Em cada passo de treinamento, o agente executa a melhor ação conhecida para o estado atual com alta probabilidade, mas ainda escolhe ações aleatórias com probabilidade ϵ . Na configuração principal, ϵ começa em 0.6 e decai linearmente até 0.01 ao longo de 80% dos episódios, o que proporciona ampla exploração no início e comportamento mais estável ao final. A taxa de aprendizado α também sofre decaimento, de 0.5 a 0.01.

O fator de desconto padrão utilizado na política final foi $\gamma = 0.99$, embora o protocolo experimental tenha avaliado também os valores 0.5, 0.7 e 0.9. Os experimentos mostraram que γ mais alto melhora a recompensa final e a taxa média de sucesso, enquanto valores um pouco menores, em especial 0.9, tendem a acelerar a convergência. Esse comportamento é coerente com o domínio: o agente precisa aceitar custos imediatos para coletar vários itens antes de retornar à base. Os valores iniciais da tabela são zero, de modo que o agente parte sem conhecimento prévio. Ao longo dos episódios, a política aprendida passa a refletir rotas que evitam armadilhas, aproveitam melhor a capacidade da mochila e reduzem visitas desnecessárias à base de entrega.

4 Avaliação experimental

Os experimentos realizados tiveram como objetivo analisar o impacto dos principais parâmetros do algoritmo e avaliar a qualidade das políticas obtidas. Em todos os testes foram utilizados 5000 episódios de treinamento por configuração. Para os experimentos de gamma, exploração e convergência, cada configuração foi executada em três sementes distintas no mesmo mapa-base, e os valores reportados correspondem a médias entre execuções.

4.1 Evolução da recompensa acumulada

Ao longo do treinamento, observou-se crescimento progressivo da recompensa acumulada por episódio, com redução da oscilação nas fases finais. Isso indica que o agente aprendeu a evitar rotas longas, reduzir exposição a armadilhas e aproveitar melhor a capacidade da mochila antes de retornar à base. A curva mostra claramente uma fase inicial mais exploratória, seguida por uma estabilização em patamares de recompensa mais altos, compatível com a consolidação de uma política mais eficiente.

4.2 Comparação entre valores de γ

Foi realizada uma análise do impacto do fator de desconto γ sobre o desempenho do agente. Os resultados da Tabela 1 mostram que valores mais altos de γ proporcionaram melhor desempenho final e menor variabilidade entre sementes, indicando que considerar recompensas futuras é importante para o problema proposto. O melhor resultado médio foi obtido com $\gamma = 0.99$, que apresentou maior recompensa média, maior taxa de sucesso e o menor desvio padrão dentre as configurações testadas. Em contrapartida, $\gamma = 0.9$ mostrou uma relação custo-benefício interessante por combinar desempenho alto com menor tempo de treinamento.

Tabela 1: Resultados para diferentes valores de γ

γ	Recompensa Média	Taxa de Sucesso	Tempo de Treinamento (s)
0.50	3807.75 $\pm$ 880.49	66.21%	3.94
0.70	5517.53 $\pm$ 846.05	81.66%	3.17
0.90	6037.55 $\pm$ 623.09	89.27%	2.90
0.99	6401.81 $\pm$ 173.21	90.07%	3.12

4.3 Comparação entre estratégias de exploração

Também foram avaliadas diferentes estratégias de exploração durante o treinamento. Os resultados da Tabela 2 mostram que a estratégia *greedy* apresentou maior taxa de sucesso durante o treino e menor tempo médio, mas com variabilidade mais alta e recompensa média inferior à abordagem com decaimento. A estratégia *decay* obteve a melhor recompensa média, sugerindo melhor equilíbrio entre explorar o ambiente no início e consolidar uma política estável nas etapas finais. A exploração constante foi bastante estável, mas ficou abaixo da melhor configuração em qualidade final.

Tabela 2: Comparação entre estratégias de exploração

Estratégia	Recompensa Média	Taxa de Sucesso	Tempo de Treinamento (s)
Constante	5317.45 $\pm$ 60.46	95.85%	2.60
Decay	6401.81 $\pm$ 173.21	90.07%	3.06
Greedy	5274.96 $\pm$ 514.87	97.88%	1.99

4.4 Tempo de convergência

O tempo de convergência foi estimado pelo primeiro episódio em que a média móvel de recompensa atingiu 90% do melhor valor médio observado em cada execução. Os resultados são apresentados na Tabela 3.

Tabela 3: Tempo estimado de convergência

γ	Episódio de Convergência	Tempo de Treinamento (s)
0.50	4201.67 $\pm$ 347.91	3.95
0.70	4030.00 $\pm$ 197.14	3.27
0.90	3533.67 $\pm$ 89.07	2.98
0.99	4042.00 $\pm$ 145.19	3.12

O valor $\gamma = 0.9$ apresentou convergência mais rápida e menor variabilidade no ponto de estabilização, reforçando a interpretação de que ele aprende mais cedo uma política útil. Ainda assim, $\gamma = 0.99$ manteve vantagem em recompensa final, o que evidencia um trade-off entre velocidade de convergência e qualidade da política final.

4.5 Estabilidade da política aprendida

A estabilidade da política aprendida foi avaliada a partir da média das recompensas dos 200 episódios finais de cinco execuções independentes. Os valores obtidos foram 6817.05, 6342.75, 6518.15, 6519.05 e 6639.25, resultando em recompensa média final de aproximadamente 6567.25 e desvio padrão de aproximadamente 155.14. Esses resultados indicam que a política aprendida apresenta comportamento relativamente estável entre sementes, sem a repetição artificial observada em versões anteriores. A variação restante é esperada, pois o treinamento permanece dependente da trajetória de exploração e das quedas estocásticas em armadilhas.

4.6 Comparação entre Bellman e Q-Learning

A abordagem baseada em planejamento com Value Iteration e a baseada em aprendizado via Q-Learning foram comparadas, com utilização do mesmo mapa-base. Os resultados são apresentados na Tabela 4.

Tabela 4: Comparação entre Bellman e Q-Learning

Método	Passos de otimização	Tempo (s)	Taxa de sucesso
Bellman Planning	110.0	2.38	100.00%
Q-Learning	4042.0	3.08	40.17%

A abordagem baseada em Bellman apresentou política final mais consistente no mapa conhecido, com taxa de sucesso igual a 1.0, recompensa média de 7074.15 e desvio padrão de 384.91. Já o Q-Learning obteve tempo médio de treinamento comparável, mas com maior variação e qualidade inferior quando avaliado após o treino, refletindo dependência da exploração e da qualidade da política aprendida em cada semente. Essa diferença é esperada: Bellman resolve o problema a partir do modelo completo do ambiente, enquanto Q-Learning aprende apenas por interação. Assim, a comparação reforça a distinção entre planejamento em ambiente conhecido e aprendizado incremental em ambiente inicialmente desconhecido.

5 Visualizações

As Figuras 1 e 2 resumem os principais resultados quantitativos do processo de treinamento, destacando o efeito de γ , da exploração e da variabilidade entre execuções.

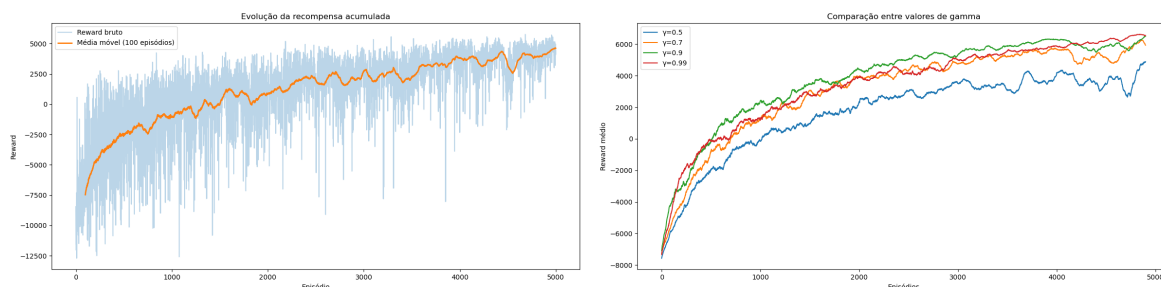


Figura 1: Curvas de aprendizado. À esquerda, observa-se o ganho gradual de desempenho ao longo do treino. À direita, descontos mais altos produzem recompensas finais superiores, embora não necessariamente a convergência mais rápida.

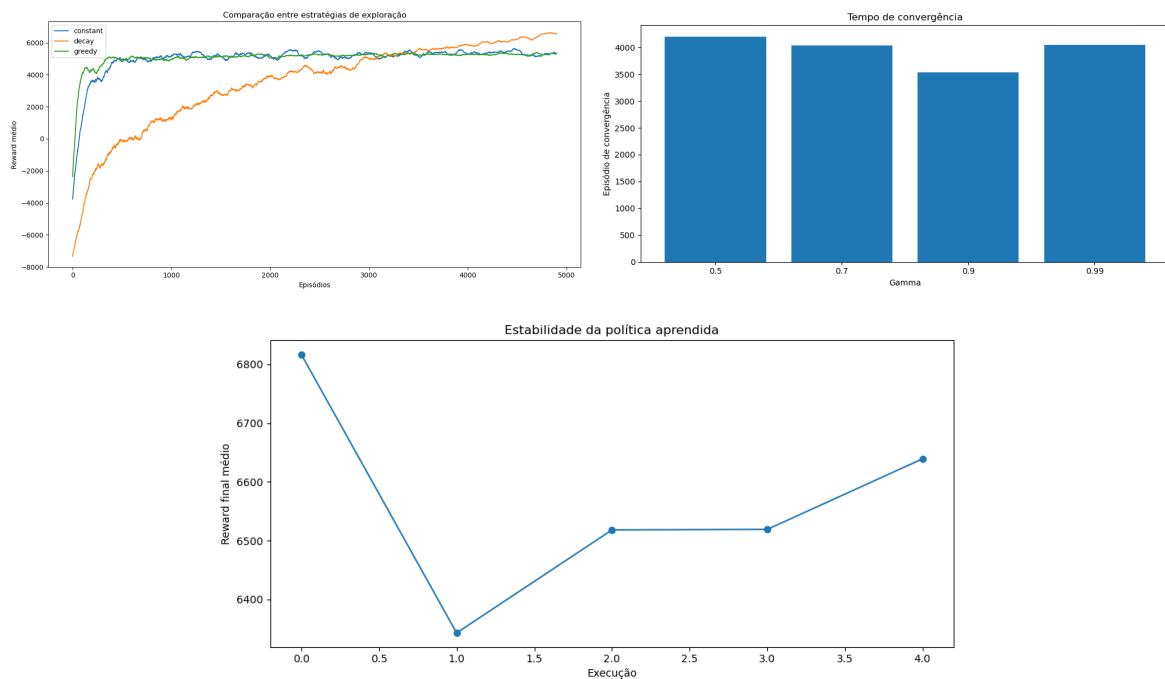


Figura 2: Diagnósticos de treinamento. Os gráficos superiores comparam exploração e convergência; o gráfico inferior resume a estabilidade da política entre sementes.

A Figura 3 reúne três perspectivas complementares sobre o comportamento do agente. O mapa de valores destaca as regiões mais promissoras do estado inicial, o mapa de política mostra as direções preferidas pela política aprendida, e a comparação de trajetórias ajuda a interpretar qualitativamente os resultados numéricos.

6 Discussão

A principal dificuldade do trabalho esteve na definição da função de recompensa e da representação de estado. Uma modelagem mais simples, baseada apenas na posição do agente, não era suficiente para distinguir situações em que ainda havia itens no ambiente ou em que a mochila estava cheia. Por isso, foi necessário incluir no estado tanto a quantidade de itens carregados quanto a bitmask de itens disponíveis, o que aumentou substancialmente o espaço de estados, mas tornou o problema corretamente observável.

Os experimentos também evidenciaram trade-offs relevantes. Valores mais altos de γ melhoraram a recompensa final e a taxa de sucesso, pois o agente passou a valorizar melhor trajetórias de longo prazo. Em compensação, $\gamma = 0.9$ convergiu mais rápido do que $\gamma = 0.99$. De forma semelhante, a exploração com decaimento obteve melhor recompensa média, enquanto a estratégia greedy foi mais rápida e manteve alta taxa de sucesso durante o treinamento, embora com desempenho final menos consistente.

O impacto da exploração ficou claro tanto no treinamento quanto na comparação final com Bellman. O Q-Learning depende da qualidade das experiências coletadas ao longo dos episódios, e isso introduz variabilidade entre sementes. Já o planejamento por Bellman, por operar diretamente sobre o modelo do ambiente, produz uma política muito mais estável no mapa conhecido. Essa diferença ajuda a ilustrar, na prática, a distinção entre aprender por interação e resolver um MDP quando a dinâmica já é conhecida, como discutido na literatura clássica de planejamento e aprendizado por reforço.

Entre as limitações do modelo, destacam-se a natureza tabular da solução, o rápido crescimento do espaço de estados e a dependência de hiperparâmetros. Embora o ambiente inclua estocasticidade local devido a quedas em armadilhas, ele ainda está longe da complexidade de cenários reais com observabilidade parcial ou mudanças estruturais no mapa. Mesmo assim, o domínio adotado foi suficiente para demonstrar os conceitos centrais de Bellman, Q-Learning, exploração e convergência.

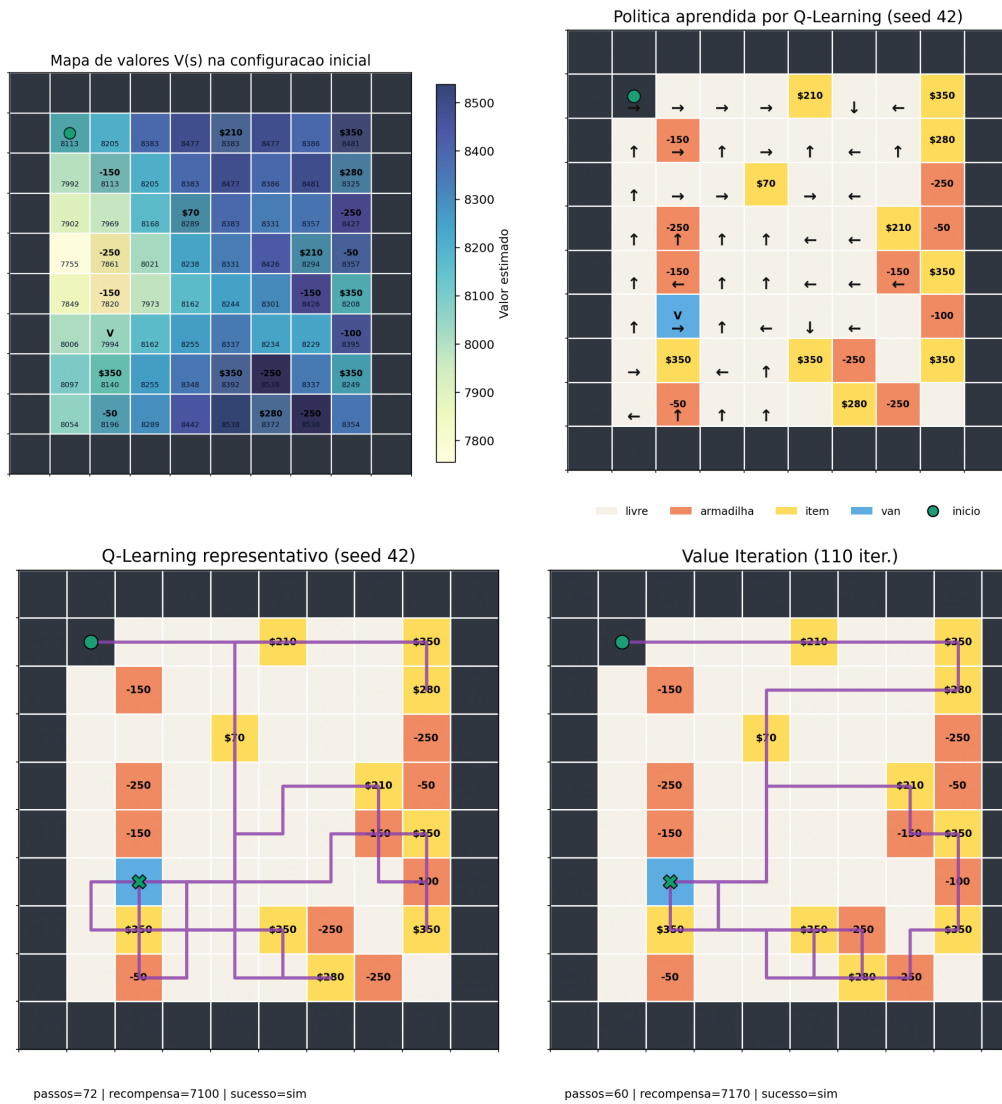


Figura 3: Mapa de valores $V(s)$, mapa da política aprendida por Q-Learning e comparação de trajetórias entre Q-Learning e Value Iteration no mesmo mapa-base. A política aprendida recupera a estrutura geral do planejamento, embora com pequenas redundâncias de percurso.

No conjunto, os resultados foram coerentes com a teoria. A penalização por movimento foi essencial para evitar políticas com trajetórias longas, o fator de desconto controlou o equilíbrio entre ganho imediato e planejamento, e a exploração ϵ -greedy impediu convergência prematura. O comportamento observado do agente, tanto nos gráficos quanto no visualizador interativo, confirma que a formulação do problema e a implementação dos algoritmos estão alinhadas com os objetivos do trabalho.

Referências

- [1] George D. Greenwade. The Comprehensive Tex Archive Network (CTAN). *TUGBoat*, 14(3):342–351, 1993.
- [2] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (4th ed.)*. Pearson, 2021.