Instituto Tecnológico de Aeronáutica - ITA
Inteligência Artificial para Robótica Móvel - CT213
Aluno: Arthur de Sousa Vianna

Relatório do Laboratório 10 - Programação Dinâmica

1 Breve Explicação em Alto Nível da Implementação

O objetivo deste relatório é analisar a implementação de programação dinâmica para analisar políticas e obter a política ótima de um Processo Decisório de Markov (MDP). Nesse caso, o MDP consiste de um tabuleiro formado por um *grid* $n \times n$ em que em cada célula são permitidas cinco ações: *stop* (S), *up* (U), *right* (R), *down* (D) e *left* (L). A ação de *stop* tem probabilidade 1 de ocorrer e as outras ações têm probabilidade de execução p_c, em que, quando a ação não é executada, todas as ações de direção têm probabilidade $\frac{1-p_c}{4}$ de ocorrer. O *grid* contém obstáculos, células as quais não é possível ocupar, além das barreiras do *grid*. Ainda, o MDP tem fator de desconto γ e recompensa -1 para cada *timestep* em que a célula do agente não é a célula objetivo, que é única e agente recebe recompensa 0 por nela estar. Note que o MDP em questão tem um modelo matemático conhecido.

1.1 Avaliação de Política

A avaliação da política π consiste em obter a função de valor para cada estado mantendo-se fixa a política. A função valor para um estado s é obtido a partir da Equação de Bellman de Expectativa, exibida pela Equação 1, em que em cada iteração analisa-se a ação possível a por meio de sua recompensa r e analisam-se os próximos estados s' por meio da probabilidade de transição p, definida pelo modelo da MDP.

$$v_\pi(s) = \sum_{a \in A} \pi(a|s) \left(r(s|a) + \gamma \sum_{s' \in S} p(s'|s,a) v_\pi(s') \right) \tag{1}$$

A implementação de avaliação de política consiste em resolver iterativamente o sistema de equações dado pela Equação 1 para cada estado, obtendo-se $v_{k+1}(s)$ a partir de $v_k(s')$, utilizando-se o método de Gauss-Jacobi em uma atualização síncrona. O algoritmo converge para a função valor da política $v_\pi(s)$.

1.2 Iteração de Valor

A iteração de valor consiste em encontrar a função de valor ótima $v_*(s)$ a partir da Equação de Bellman de Otimalidade, dada pela Equação 2. Para cada estado, escolhe-se a ação que maximiza a função valor.

$$v_{k+1}(s) = \max_{a \in A} \left(r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) v_k(s') \right) \tag{2}$$

A implementação consiste em resolver o sistema linear iterativamente. O algoritmo converge para $v_*(s)$. A política ótima pode ser encontrada a partir da função de valor ótima por meio do algoritmo "guloso" ($\pi'(s) = greedy(v_\pi(s))$, dado pela Equação 3.

$$\pi'(s) = \underset{a \in A}{\text{argmax}} \left(r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) v_\pi(s') \right) \tag{3}$$

1.3 Iteração de Política

A iteração de política consiste em avaliar iterativamente uma política e atualizá-la de forma gulosa em relação $v_k(s)$, a partir da Equação 3. O algoritmo termina em uma condição de parada, dada pelo número de iterações ou com um valor mínimo para a diferença da função valor de iterações consecutivas. O algoritmo converge para a política ótima.

2 Tabelas Comprovando Funcionamento do Código

2.1 Caso $p_c = 1,0$ e $\gamma = 1,0$

2.1.1 Avaliação de Política

```
Evaluating random policy, except for the goal state, where policy always executes stop:
Value function:
[  -384.09,  -382.73,  -381.19,    *    ,  -339.93,  -339.93]
[  -380.45,  -377.91,  -374.65,    *    ,  -334.92,  -334.93]
[  -374.34,  -368.82,  -359.85,  -344.88,  -324.92,  -324.93]
[  -368.76,  -358.18,  -346.03,    *    ,  -289.95,  -309.94]
[    *    ,  -344.12,  -315.05,  -250.02,  -229.99,    *    ]
[  -359.12,  -354.12,    *    ,  -200.01,  -145.00,     0.00]
Policy:
[  SURDL  ,  SURDL  ,  SURDL  ,    *    ,  SURDL  ,  SURDL  ]
[  SURDL  ,  SURDL  ,  SURDL  ,    *    ,  SURDL  ,  SURDL  ]
[  SURDL  ,  SURDL  ,  SURDL  ,  SURDL  ,  SURDL  ,  SURDL  ]
[  SURDL  ,  SURDL  ,  SURDL  ,    *    ,  SURDL  ,  SURDL  ]
[    *    ,  SURDL  ,  SURDL  ,  SURDL  ,  SURDL  ,    *    ]
[  SURDL  ,  SURDL  ,    *    ,  SURDL  ,  SURDL  ,    S    ]
----------------------------------------------------------------
```

Figura 1: Tabela do *grid* com a função valor de avaliação de uma política aleatória para $p_c = 1,0$ e $\gamma = 1,0$.

2.1.2 Iteração de Valor

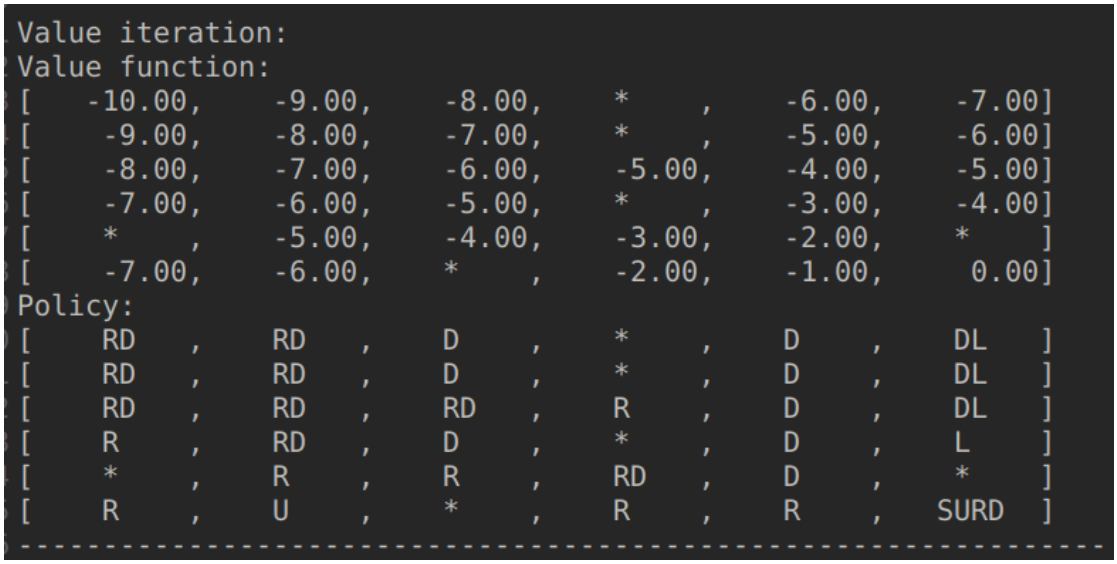

Figura 2: Tabela do *grid* com a função valor e política ótimas a partir da iteração de valor para $p_c = 1,0$ e $\gamma = 1,0$.

2.1.3 Iteração de Política

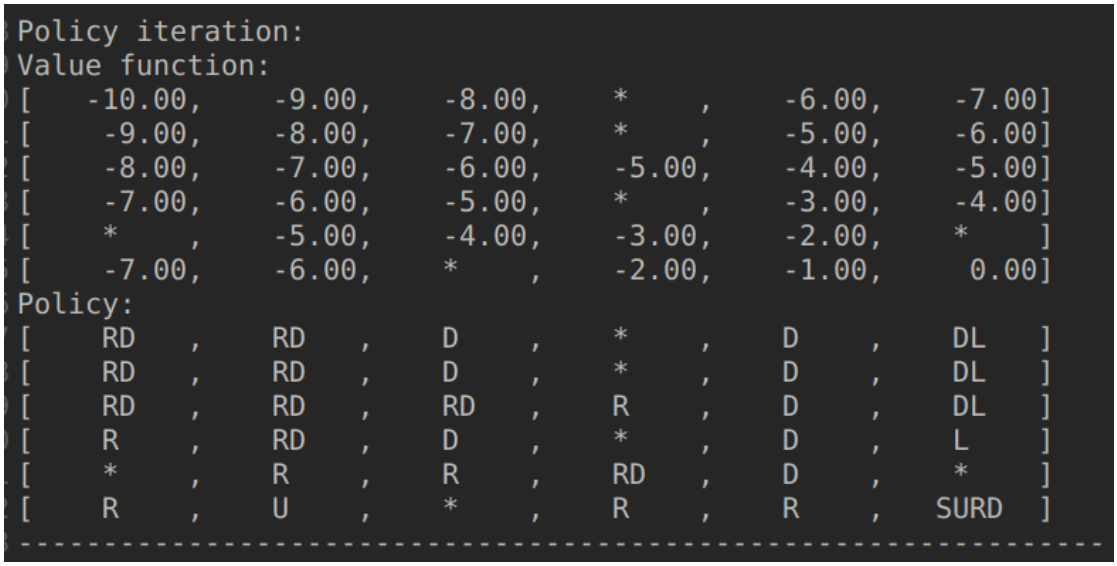

Figura 3: Tabela do *grid* com a função valor e política ótimas a partir da iteração de política para $p_c = 1,0$ e $\gamma = 1,0$.

2.2 Caso $p_c = 0,8$ e $\gamma = 0,98$

2.2.1 Avaliação de Política

```
Evaluating random policy, except for the goal state, where policy always executes stop:
Value function:
[   -47.19,   -47.11,   -47.01,    *    ,   -45.13,   -45.15]
[   -46.97,   -46.81,   -46.60,    *    ,   -44.58,   -44.65]
[   -46.58,   -46.21,   -45.62,   -44.79,   -43.40,   -43.63]
[   -46.20,   -45.41,   -44.42,    *    ,   -39.87,   -42.17]
[    *    ,   -44.31,   -41.64,   -35.28,   -32.96,    *    ]
[   -45.73,   -45.28,    *    ,   -29.68,   -21.88,     0.00]
Policy:
[  SURDL  ,  SURDL  ,  SURDL  ,    *    ,  SURDL  ,  SURDL  ]
[  SURDL  ,  SURDL  ,  SURDL  ,    *    ,  SURDL  ,  SURDL  ]
[  SURDL  ,  SURDL  ,  SURDL  ,  SURDL  ,  SURDL  ,  SURDL  ]
[  SURDL  ,  SURDL  ,  SURDL  ,    *    ,  SURDL  ,  SURDL  ]
[    *    ,  SURDL  ,  SURDL  ,  SURDL  ,  SURDL  ,    *    ]
[  SURDL  ,  SURDL  ,    *    ,  SURDL  ,  SURDL  ,    S    ]
------------------------------------------------------------
```

Figura 4: Tabela do *grid* com a função valor de avaliação de uma política aleatória para $p_c = 0,8$ e $\gamma = 0,98$.

2.2.2 Iteração de Valor

```
Value iteration:
Value function:
[   -11.65,   -10.78,    -9.86,    *    ,    -7.79,    -8.53]
[   -10.72,    -9.78,    -8.78,    *    ,    -6.67,    -7.52]
[    -9.72,    -8.70,    -7.59,    -6.61,    -5.44,    -6.42]
[    -8.70,    -7.58,    -6.43,    *    ,    -4.09,    -5.30]
[    *    ,    -6.43,    -5.17,    -3.87,    -2.76,    *    ]
[    -8.63,    -7.58,    *    ,    -2.69,    -1.40,     0.00]
Policy:
[    D    ,    D    ,    D    ,    *    ,    D    ,    D    ]
[    D    ,    D    ,    D    ,    *    ,    D    ,    D    ]
[    RD   ,    D    ,    D    ,    R    ,    D    ,    D    ]
[    R    ,    RD   ,    D    ,    *    ,    D    ,    L    ]
[    *    ,    R    ,    R    ,    D    ,    D    ,    *    ]
[    R    ,    U    ,    *    ,    R    ,    R    ,    S    ]
------------------------------------------------------------
```

Figura 5: Tabela do *grid* com a função valor e política ótimas a partir da iteração de valor para $p_c = 0,8$ e $\gamma = 0,98$.

2.2.3 Iteração de Política

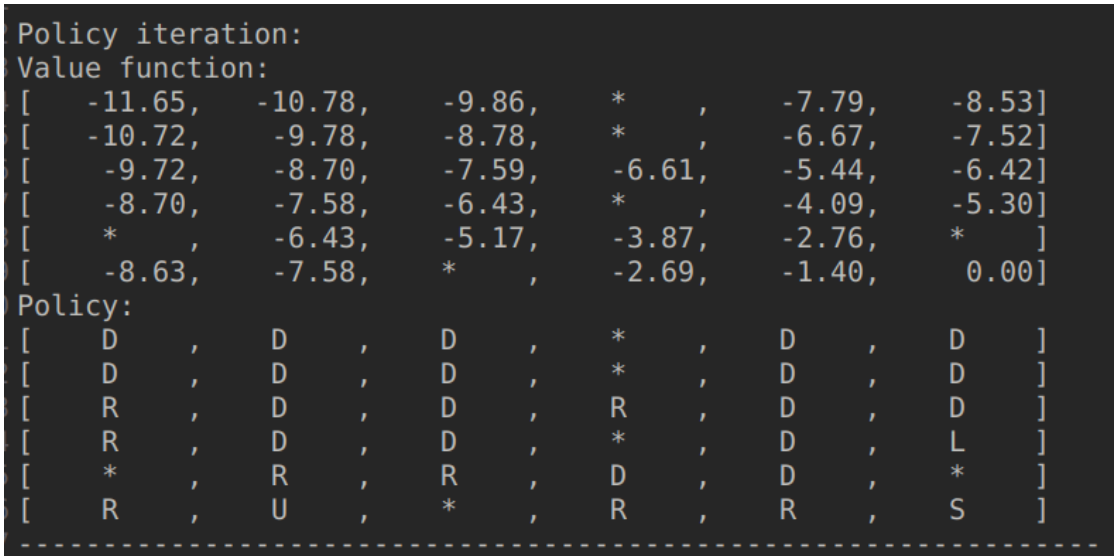

Figura 6: Tabela do *grid* com a função valor e política ótimas a partir da iteração de política para $p_c = 0,8$ e $\gamma = 0,98$.

3 Discussão dos Resultados

A implementação e os testes dos algoritmos de programação dinâmica forneceram uma visão clara sobre a resolução de Processos Decisórios de Markov (MDPs) com modelo conhecido. A análise dos resultados, tanto para o ambiente determinístico quanto para o estocástico, permite extrair conclusões importantes sobre o comportamento dos métodos e a natureza das políticas ótimas resultantes.

Primeiramente, é fundamental observar que tanto a Iteração de Valor quanto a Iteração de Política convergiram para a mesma função de valor ótima e, consequentemente, para a mesma política ótima em ambos os cenários testados ($p_c = 1,0$ e $p_c = 0,8$). Este resultado era esperado e valida a correção teórica e prática de ambas as implementações, pois os dois algoritmos são garantidos a encontrar a solução ótima. A principal diferença entre eles reside na sua abordagem computacional: a Iteração de Política alterna entre uma avaliação completa da política atual e um passo de melhoria gulosa, enquanto a Iteração de Valor funde essas duas etapas em uma única equação de atualização (a equação de Bellman de Otimalidade). Na prática, a Iteração de Política pode convergir em menos iterações, mas cada iteração é mais custosa. A escolha entre os dois métodos depende, portanto, das características específicas do problema, como o número de estados e ações.

Analisando o caso determinístico ($p_c = 1,0$, $\gamma = 1,0$), os resultados são bastante intuitivos. A função de valor ótima, conforme exibida na Figura 2 e na Figura 3, apresenta valores inteiros negativos. Isso ocorre porque, com um fator de desconto $\gamma = 1$ e uma recompensa de -1 por passo, o valor de um estado é exatamente o negativo do número de passos no caminho mais curto até o estado objetivo. A política ótima reflete essa realidade, sendo direta e "agressiva", traçando

o caminho geometricamente mais curto sem hesitar em se mover adjacente a obstáculos, visto que não há risco de uma ação falhar e resultar em uma penalidade de tempo.

O caso estocástico ($p_c = 0,8$, $\gamma = 0,98$) revela um comportamento muito mais rico e interessante. A introdução da incerteza (20% de chance de uma ação aleatória ocorrer) e do fator de desconto ($\gamma < 1$) altera fundamentalmente a natureza da política ótima. A função de valor não é mais uma simples contagem de passos, e a política resultante, exibida nas Figuras 5 e 6, torna-se visivelmente mais "cautelosa". O agente agora evita trajetórias que, embora geometricamente mais curtas, o colocam em risco. Por exemplo, ele pode preferir um caminho que o mantenha mais centralizado no corredor em vez de um que "raspa" a parede de um obstáculo. Isso ocorre porque o custo esperado de uma ação que beira um obstáculo é maior, pois a probabilidade de uma ação aleatória resultar em uma colisão (e, portanto, em perda de tempo) é significativa. A política ótima não busca mais o caminho mais curto, mas sim o caminho que minimiza a soma descontada de recompensas negativas a longo prazo, efetivamente buscando o caminho mais seguro e eficiente em termos de expectativa.

Em suma, os experimentos demonstram com sucesso a capacidade dos algoritmos de programação dinâmica em resolver MDPs e ilustram de forma eficaz o impacto da estocasticidade na formulação de uma estratégia ótima, forçando a transição de um planejamento puramente geométrico para um planejamento robusto à incerteza.