Relatório - Planejamento Automatizado
Planejamento Probabilístico - Algoritmo Value Iteration

Alexandre Del Rey, Paula Moraes e Vinícius Almeida

November 20, 2017

Iteração de Valor

Um Processo de Decisão Markoviano (MDP) M é caracterizado por uma quíntupla $< S, D, A, T, R >$, sendo:

S: um conjunto finito de estados

D: uma sequência de passos discretos (horizonte) que pode ser finita ou infinita

A: um conjunto finito de ações

$T : S \times A \times S' \rightarrow [0, 1]$ uma função de transições probabilísticas estacionária

$R : S \rightarrow R$ uma função de recompensa estacionária

O algoritmo de Iteração de Valor[1], descrito no código abaixo, é um método para computar a política e valor ótimos de um MDP.

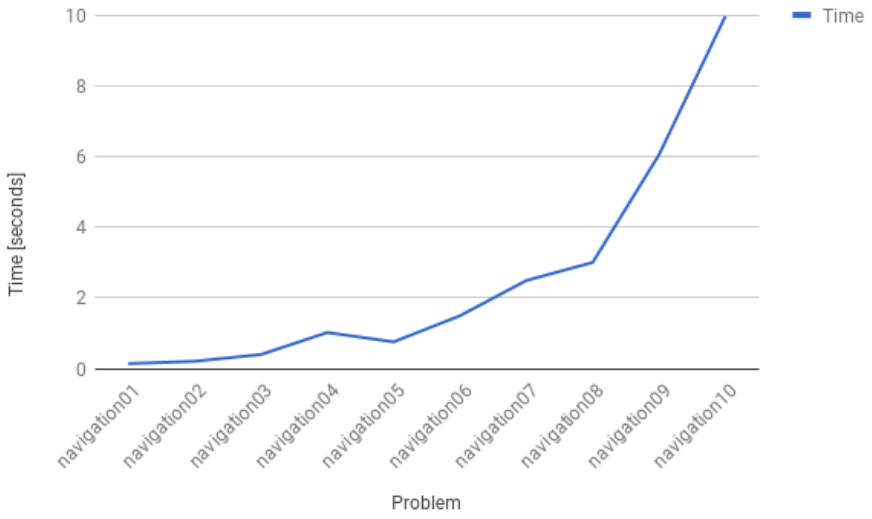
Algorithm 1 Value Iteration

```
1:  function ITERVAL(M)
2:      for each s in S do
3:          v_0(s) ← r(s)
4:      n ← 0
5:      repeat
6:          n ← n + 1
7:          for each s in S do
8:              for each a in A do
9:                  q_n(s,a) ← r(s) + γ Σ_{s'∈S} τ(s,a,s') v_{n-1}(s')
10:             v_n(s) ← max_{a∈A} q_n(s,a)
11:             π_n(s) ← argmax_{a∈A} q_n(s,a)
12:     until |v_n(s) − v_{n-1}(s)| < ε, ∀s ∈ S
13:     return π_n
```

[1] Código escrito em Python, disponível em `https://github.com/pksm/VI_planejamento`

Resultados - Domínio *Navigation*

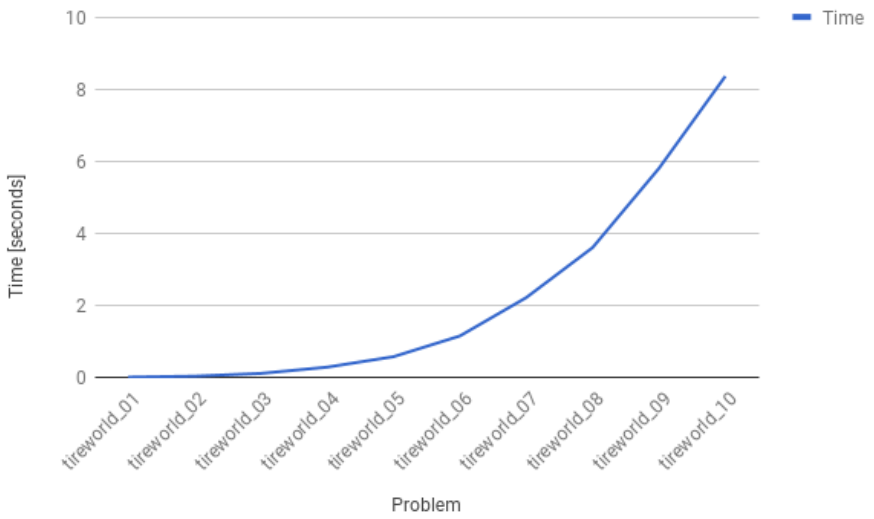
No domínio *Navigation*, um robô deve ir de uma localização origem a uma localização destino percorrendo um grid. O robô pode executar quatro tipos de ação de movimentação para cada direção (norte, sul, leste e oeste). Durante o percurso o robô deve evitar regiões nas quais ele possa se danificar e assim ficar incapacitado de se movimentar. A cada estágio que o robô não alcançar seu destino, o robô será penalizado.



Neste domínio, o tempo de convergência para um valor residual $\varepsilon = 0.0$ é proporcional ao aumento da complexidade dos problemas analisados, com exceção do *navigation05*. Uma possível hípotese está em uma convergência mais rápida dos estados deste se comparado ao problema *navigation04*.

Resultados - Domínio *Triangle-Tire World*

O domínio *Triangle-Tire World* também consiste em um problema de navegação em grid, onde um carro deve ir de uma célula a outra. O carro inicia sua trajetória em uma célula ao sul e deve escolher entre três trajetórias possíveis, que são: ir para norte, nordeste ou noroeste. Durante a movimentação entre os grids, o carro corre o risco de ter um de seus pneus furados, não podendo mais seguir viagem. Nesta situação, ele poderá trocar o seu pneu furado por um estepe e voltar a fazer o percurso normalmente. Em algumas posições, há pontos onde o carro poderá obter um estepe extra, caso já o tenha utilizado. Assim como no domínio *Navigation*, o carro será penalizado enquanto não alcançar seu destino.



Neste domínio, o tempo de convergência para um valor residual $\varepsilon = 0.0$ é proporcional ao aumento da complexidade dos problemas analisados, como é possível observar pela curva exponencial em relação ao tempo.

Conclusão

O algoritmo de iteração de valores utiliza um cálculo dinâmico e iterativo para determinar o valor V de cada estado s pertencente ao conjunto de estados S do MDP, em cada estágio de decisão. O algoritmo busca a convergência de valores, através da utilização da equações de Bellman em passos de programação dinâmica, melhorando a utilidade a cada iteração (Bellman backup). Também é importante considerar que ao se utilizar o ELAU (Expected Linear Addictive Utility) com um fator de desconto de 0.9, o algoritimo privilegia recompensas mais imediatas do que recompensas futuras.

Quanto a complexidade do iteração de valor, verificamos que é quadrática no número de estados e linear no número de ações.

Referência

[1] Andrey Kolobov. Planning with markov decision processes: An ai perspective. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–210, 2012.