Aceleração do Aprendizado por Reforço em Problemas com Múltiplos Objetivos

Helen C. de Mattos Senefonte[1], Reinaldo A. C. Bianchi[2], Carlos H. C. Ribeiro[3]

[1]Universidade Estadual de Londrina (UEL)
Londrina, PR – Brazil

[2]Centro Universitário da FEI (FEI)
São Bernardo do Campo, SP – Brazil

[3]Instituto Tecnológico de Aeronáutica (ITA)
São José dos Campos, SP – Brazil

helen@uel.br, carlos@ita.br, rbianchi@fei.edu.br

Abstract. *We present a new variation of an action-selection approach to be used in Multi Objective Reinforcement Learning problems, in order to speed up the learning process. To do so, we propose to use the action selection method of the Heuristically Accelerated Q-learning algorithm (HAQL), in which a heuristic function $\mathcal{H}$ is used to influence the choice of actions during the learning, in traditional multi objective RL algorithms. This proposal was evaluated using a traditional research task: the predator-prey problem, and the results indicate that the use of heuristics is able to provide the desired learning acceleration.*

Resumo. *Apresentamos neste artigo uma nova variação de uma abordagem para a seleção de ações em problemas de Aprendizado por Reforço com Múltiplos Objetivos, visando à diminuição do tempo necessário para a realização do aprendizado. Para tanto, é proposta a utilização do método de seleção de ações do algoritmo Q-learning Acelerado por Heurísticas (HAQL), no qual uma função heurística $\mathcal{H}$ é usada para influenciar o agente na escolha de suas ações durante o aprendizado, em algoritmos tradicionais de RL multiobjetivos. A proposta foi avaliada utilizando um domínio tradicional em pesquisas com múltiplos objetivos: o problema Presa-Predador, e os resultados obtidos indicam que o uso de heurísticas é capaz de proporcionar a aceleração do aprendizado desejada.*

1. Introdução

Para permitir a utilização de algoritmos de Aprendizado por Reforço (*Reinforcement Learning* – RL) [Sutton e Barto, 1998] em problemas onde existem múltiplos objetivos (*Multiple Objective Reinforcement Learning* – MORL), por vezes concorrentes, diversos autores têm proposto soluções modularizadas para as tarefas, decompondo o problema original em sub-tarefas, preferencialmente independentes entre si, cada uma representada por um Processo Decisório de Markov (*Markov Decision Process* – MDP). Essa divisão traz consigo uma questão a ser tratada: como selecionar as ações a serem executadas, considerando que cada MDP propõe uma ação distinta a cada instante de tempo para solucionar cada sub-tarefa.

Estudos recentes apresentam diferentes técnicas para a solução deste problema: Barrett e Narayanan [2008] propõe um método que garante encontrar a política ótima para qualquer função de valor linear; já outra abordagem é apresentada por Vamplew et al. [2008] que projeta problemas MORL como um conjunto de soluções baseada no princípio de Pareto, e mostra que o uso de reforços escalares – ao invés de um reforço vetorial em que cada elemento do vetor é o reforço de cada objetivo do problema MORL – tem limitações fundamentais, quando usada para encontrar políticas ótimas de Pareto.

Este trabalho propõe a utilização do método de seleção de ações do algoritmo Q-learning Acelerado por Heurísticas (HAQL) [Bianchi et al., 2008], uma variante do Q-learning [Watkins e Dayan, 1992] que incorpora a utilização de heurísticas aproveitando indícios existentes no processo de aprendizagem da estrutura do domínio, em problemas de RL com múltiplos objetivos. A análise desta proposta é feita para problemas em que um único agente deve aprender simultaneamente várias sub-tarefas independentes, utilizando um domínio tradicional em pesquisas com múltiplos objetivos: o problema Presa-Predador.

O artigo está estruturado da seguinte maneira: na Seção 2 são apresentados definições básicas de RL; a Seção 3 apresenta algoritmos utilizados no problema de seleção de ações em MORL. A seção 4 descreve o uso de heurísticas no aprendizado por reforço. A Seção 5 apresenta algumas características do ambiente experimental, juntamente com a descrição da heurística utilizada. Resultados obtidos na etapa experimental são apresentados na Seção 6. A Seção 7 apresenta as conclusões do artigo.

2. MDPs e o Aprendizado por Reforço

O Aprendizado por Reforço (RL) é uma técnica muito atraente para solucionar uma variedade de problemas de controle e planejamento quando não existem modelos disponíveis *a priori*, já que seus algoritmos têm a convergência para uma situação de equilíbrio garantida [Sutton e Barto, 1998], além de permitirem o aprendizado de estratégias de controle adequadas. No RL o aprendizado acontece por meio da interação direta entre o agente e o ambiente. Infelizmente, a convergência dos algoritmos de RL só pode ser atingida após uma extensiva exploração do espaço de estados-ações, que é geralmente demorada.

No RL, um agente aprendiz interage com o ambiente em intervalos de tempos discretos em um ciclo de percepção-ação. A maneira mais tradicional para formalizar o RL é utilizando o conceito de Processo Markoviano de Decisão (*Markov Decision Process* – MDP), que pode ser definido formalmente [Mitchell, 1997] pela quádrupla $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$, onde: $\mathcal{S}$ é um conjunto finito de estados do ambiente; $\mathcal{A}$ é um conjunto finito de ações que o agente pode realizar; $\mathcal{T} : \mathcal{S} \times \mathcal{A} \mapsto P_{s,a}(\mathcal{S})$ é a função de transição de estados ($P_{s,a}(\mathcal{S})$ é uma distribuição de probabilidades sobre o conjunto de estados $\mathcal{S}$, e portanto $\mathcal{T}(s_t, a_t, s_{t+1})$ define a probabilidade de realizar a transição do estado s_t para o estado s_{t+1} quando se executa a ação a_t); e $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \Re$ é a função de reforço ($\Re$ denota o conjunto dos números reais).

Assim, no ciclo percepção-ação, o agente aprendiz observa, a cada passo de iteração, o estado corrente s_t do ambiente e escolhe a ação a_t para realizar. Ao executar esta ação a_t – que altera o estado do ambiente – o agente recebe um sinal escalar de reforço (penalização ou recompensa) $r(s_t, a_t)$. Resolver um MDP consiste em computar a política $\pi : \mathcal{S} \mapsto \mathcal{A}$ que maximiza (ou minimiza) o valor esperado de um funcional

dos reforços, geralmente uma soma ponderada destes ao longo do tempo. Na formulação mais comum, o objetivo do agente de aprendizado é encontrar uma política de ações estacionária $a_t^* = \pi^*(x_t)$ que maximize a função de valor esperado da Equação 1 para cada estado x_0:

$$V^{\pi}(x_0) = \lim_{M \to \infty} \mathrm{E}\left[\sum_{t \to 0}^{M} \gamma^t r(x_t, \pi(x_t))\right] \tag{1}$$

O fator de desconto $0 < \gamma < 1$ define se os reforços recentes são mais importantes ou não. A política ótima estacionária é dada por $\pi^*(x) = \arg\max_{\pi} V^{\pi}(x), \forall x$.

Provavelmente mais popular algoritmo de RL, o algoritmo *Q–Learning* foi proposto como uma maneira de aprender iterativamente a política ótima π^* quando o modelo do sistema não é conhecido [Watkins e Dayan, 1992]. O algoritmo *Q–Learning* propõe que o agente aprenda uma função Q de soma de recompensas esperadas com descontos, conhecida como função valor–ação. O algoritmo opera aproximando iterativamente $\hat{Q}$ – a estimativa de $Q^*(s, a)$ no instante t – da seguinte maneira: estando o agente num determinado estado x, ele escolhe uma ação a disponível neste estado usando uma dada estratégia de exploração, executa essa ação a, recebe o respectivo reforço r, e visita o estado resultante y. No estado y é feita uma busca, entre as ações disponíveis, para encontrar a ação b que produza o maior valor esperado. Assim, o valor do par estado-ação é atualizado, segundo a Equação 2 [Watkins e Dayan, 1992]:

$$Q(x, a) \;=\; Q(x, a) \;+\; \alpha \;\left[r \;+\; \gamma \;\max_{b} Q(y, b) \;-\; Q(x, a)\right]. \tag{2}$$

O parâmetro α é o fator de aprendizado ($0 < \alpha \leq 1$), frequentemente definido pela Equação 3:

$$\alpha = \frac{1}{1 + visitas(x, a)} \tag{3}$$

Sendo $visitas(x, a)$ o número de visitas já realizadas ao estado x, com ação a escolhida e realizada.

Uma propriedade importante deste algoritmo é que as ações usadas durante o processo iterativo de aproximação da função Q podem ser escolhidas usando qualquer estratégia de exploração (ou explotação). Uma estratégia para a escolha das ações bastante utilizada em implementações do *Q–Learning* é a exploração aleatória $\epsilon - Greedy$, na qual o agente executa a ação com o maior valor de Q com probabilidade $1 - \epsilon$ e escolhe uma ação aleatória com probabilidade ϵ. Neste caso, a transição de estados é dada pela seguinte equação:

$$\pi(s_t) = \begin{cases} a_{random} & \text{se } q \leq \epsilon, \\ \arg\max_{a_t} \hat{Q}_t(s_t, a_t) & \text{caso contrário,} \end{cases} \tag{4}$$

onde: q é um valor escolhido de maneira aleatória com distribuição de probabilidade uniforme sobre [0,1] e ϵ ($0 \leq \epsilon \leq 1$) é o parâmetro que define a taxa de exploração: quanto menor o valor de ϵ, menor a probabilidade de se fazer uma escolha aleatória e a_{random} é uma ação aleatória selecionada entre as ações possíveis de serem executadas no estado s_t.

3. Métodos de Seleção de Ações

As técnicas para a seleção de que ação executar quando o aprendizado do agente envolve uma única tarefa é um tema bem estudado, com resultados teóricos e técnicas conhecidas [Sutton e Barto, 1998]. Em contrapartida, no aprendizado por reforço com múltiplos objetivos estes métodos possuem uma alta complexidade e ainda apresentam desafios.

Partindo da idéia de que nos sistemas multiobjetivos cada MDP corresponde apenas a uma determinada sub-tarefa, surge a necessidade de desenvolver um mecanismo que considere o conhecimento obtido em cada um dos MDPs para encontrar uma política ótima (ou pelo menos adequada) conjunta, durante o processo de aprendizagem.

Os métodos de seleção de ações são mecanismos que têm como objetivo escolher uma ação (idealmente ótima para uma dada função de custo definida *ex-ante*) entre as ações possíveis de ser executadas a cada instante, num estado x, em cada um dos MDPs. Neste artigo foram considerados dois métodos propostos por Humphrys [1997]: "Maximiza a Maior Satisfação" e "*W(x,a)-learning*".

Os métodos de seleção de ações são baseados em uma competição entre os agentes, utilizando-se uma variável W que avalia a importância das possíveis ações. Considera-se vitoriosa a ação que tiver o maior valor W, definido de acordo com o método de seleção considerado e a partir dos valores $Q(x, a)$ obtidos independentemente por cada agente, atuando em seu MDP.

3.1. Maximiza a Maior Satisfação

Por conveniência notacional, denotemos a_i como sendo a ação preferida para o agente que opera no MDP i (ou seja, $a_i = \arg\max_{a \in A_i} Q_i(x, a)$, onde A_i é o conjunto de ações disponíveis para o MDP i). O método Maximiza a Maior Satisfação é baseado na maior satisfação individual, e ignora eventuais penalizações que a escolha de uma ação possa causar aos demais MDPs. O valor W_i para um agente operando no seu MDP i é obtido como $W_i(x) = Q_i(x, a_i) = \max_{a \in A_i} Q_i(x, a)$, e o objetivo é escolher a ação que maximiza este valor entre todos os agentes, de acordo com a Equação 5:

$$a_k = \arg\max_{a \in A'} W_i(x) = \arg\max_{a \in A'} [\max_{a \in A_i} Q_i(x, a)] \tag{5}$$

Onde $A' = A_1 \cup A_2 \cup \ldots \cup A_n$ é o conjunto de ações disponíveis no estado x, n é o número de MDPs e k identifica o MDP cuja ação preferida é a_k e que venceu a competição pela escolha da ação no estado x, nesse instante. Note que a ação a_k pode produzir baixos valores $Q_i(x, a_k)$ para MDPs não vitoriosos.

3.2. W(x,a)-learning

Este método considera uma atualização dos valores W e uma expressão mais fidedigna destes como indicadores de preferência, pois considera-se não apenas a satisfação individual: o valor W_i representa a diferença entre o valor que seria recebido se o MDP i fosse o vencedor e o valor recebido pela execução da ação vencedora. Sousa [2006] propõe, de modo a se evitarem problemas causados pela estocasticidade da atualização dos valores Q em MDPs, a armazenagem de valores $W_i(x, a)$ e não $W_i(x)$, tal como originalmente proposto por Humphrys [1997]. A atualização é feita de acordo com a Equação 6:

$$W_i(x, a) = (1 - \alpha)\, W_i(x, a) + \alpha\, [Q_i(x, a_i) - (r_i + \gamma \max_{a \in A_i} Q_i(y, a))] \tag{6}$$

e a ação é então escolhida a cada iteração como $a_k = \arg\max_{a \in A'} W_i(x, a)$.

4. Q-Learning Acelerado por Heurísticas (HAQL)

Por ser o mais popular algoritmo de RL e possuir uma grande quantidade de dados na literatura para a realização de uma avaliação comparativa, o primeiro algoritmo que utiliza heurísticas a ser implementado foi uma extensão do algoritmo Q-Learning, denominado Q-Learning Acelerado por Heurísticas -– HAQL [Bianchi et al., 2008]. Este algoritmo faz uso de uma função heurística como uma maneira de usar o conhecimento sobre a política de um sistema para acelerar o aprendizado. Este conhecimento pode ser derivado diretamente do domínio ou construído a partir de indícios existentes no próprio processo de aprendizagem.

No HAQL a função heurística é usada somente na regra de transição de estados que escolhe a ação a a ser tomada no estado x. A estratégia para a escolha das ações utilizada no HAQL é a exploração aleatória $\epsilon - Greedy$, na qual, além da estimativa das funções de probabilidade de transição $\mathcal{T}$ e da recompensa $\mathcal{R}$, a função $\mathcal{H}$ também é considerada. A regra de transição de estados aqui proposta é dada pela Equação 7:

$$\pi(x) = \begin{cases} a_{random} & \text{se } q \leq p; \\ \arg\max_a \left[Q(x,a) + \xi \, H(x,a)^\beta \right] & \text{caso contrário,} \end{cases} \tag{7}$$

onde:

- q é um valor escolhido de maneira aleatória com distribuição de probabilidade uniforme sobre $[0, 1]$ e p $(0 \leq p \leq 1)$ é o parâmetro que define a taxa de exploração: quanto menor seu valor, menor a probabilidade de se fazer uma escolha aleatória,
- a_{random} é uma ação selecionada de maneira aleatória entre as ações executáveis no estado x,
- ξ e β são variáveis reais usadas para ponderar a influência da função heurística, e
- $H(x, a)$ é definido pela Equação 8:

$$H(x,a) = \begin{cases} \max_a \left[Q(x,a) \right] - Q(x,a) + \eta & \text{se } a = \pi^H(x); \\ 0 & \text{caso contrário.} \end{cases} \tag{8}$$

onde η é um valor pequeno e $\pi^H(x)$ é a heurística obtida usando um método apropriado.

Uma heurística pode ser definida como uma técnica que melhora, no caso médio, a eficiência na solução de um problema [Bianchi et al., 2008]. Funções heurísticas são a forma mais comum de se aplicar um conhecimento adicional do problema a um algoritmo de busca [Russell e Norvig, 1995].

A hipótese central deste trabalho é que a aceleração do aprendizado por reforço em problemas com múltiplos objetivos pode ser alcançada a partir do uso de heurísticas para escolha de ações nos métodos apresentados na seção anterior. Para tanto, a próxima seção apresenta como se pode fazer uso de heurísticas nos métodos de seleção de ações "Maximiza a Maior Satisfação" e "*W(x,a)-learning*".

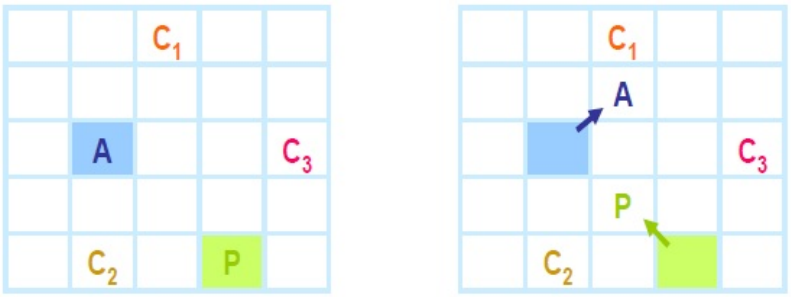

Figura 1. Ambiente padrão de testes.

5. Uso de heurísticas em métodos de seleção de ações

Para fazer uso de heurísticas para escolha de ações nos métodos apresentados na Seção 3, é necessário introduzir a função heurística nas equações de seleção de ação. Assim, propomos que as funções de seleção de ações usadas passem a ser, em analogia com a modificação introduzida no algoritmo HAQL para incluir a função heurística:

- Para o método "Maximiza a Maior Satisfação":

$$a_k = \arg\max_{a \in A'} \{ \max_{a \in A_i} \left[Q_i(x,a) + \xi H(x,a)^\beta \right] \} \tag{9}$$

- E para o método "*W(x,a)-learning*":

$$a_k = \arg\max_{a \in A'} \left[W_i(x,a) + \xi \, H(x,a)^\beta \right] \tag{10}$$

Definir o ambiente onde os testes foram realizados facilita a descrição de como a heurística $H(x,a)$ é computada. Apresentamos portanto o domínio considerado, a seguir.

5.1. O problema da Presa-Predador

O domínio para testes implementado é baseado no conhecido problema da Presa-Predador, introduzido por Singh e Cohn [1998] e utilizado por autores (Sprague e Ballard [2003]; Sousa [2006]), que realizaram testes com diversos métodos de seleção de ações.

No problema da Presa-Predador, um agente A e um predador P se movem num mundo representado por uma matriz 5×5. O ambiente possui três células de reforço positivo C1, C2 e C3 estáticas, conforme apresentado na Figura 1. O agente move-se em 8 direções possíveis (N, NE, E, SE, S, SO, O e NO), e tem por objetivo procurar as células de reforço positivo presentes no ambiente, e ao mesmo tempo, evitar o predador. O agente encontra um reforço positivo ou é apanhado pelo predador quando partilha a mesma posição da matriz. Quando o agente executa uma ação que o leva contra os limites do ambiente, permanece na mesma posição.

Foram desenvolvidos três ambientes experimentais baseados nesse ambiente, denominados (em concordância com Sousa [2006]) R1110, R2340 e R2343. Os nomes referem-se aos reforços recebidos pelo agente quando encontra as células de reforço positivo e o predador, de acordo com a Tabela 1.

O desempenho do agente é medido através do reforço médio $R_m = \frac{1}{100} . \sum_{t=1}^{100} r_t$ recebido ao longo de 100 passos simulados, correspondendo a um episódio. Para geração dos resultados de desempenho foram executados testes a intervalos regulares, intercalando períodos de treinos. As diferenças entre treino e teste são:

Ambientes	Célula 1	Célula 2	Célula 3	Predador
R1110	+1	+1	+1	0
R2340	+2	+3	+4	0
R2343	+2	+3	+4	-3

Tabela 1. Tabela de reforços para ambientes experimentais testados.

1. Treinos: há exploração para aumentar o grau de conhecimento de cada MDP do ambiente; período em que ocorre a extração de estrutura para a construção da heurística e atualização dos valores Q.
2. Testes: Não há exploração e sim a execução da melhor política encontrada até então; período em que são obtidos os resultados dos experimentos, ou seja, são gerados os valores de R_m.

5.2. Uso de métodos "Heurísticas a partir de X"

A função heurística H pode ser definida de maneira *ad hoc* ou pode ser extraída durante o processo de aprendizado. Para a extração automática iterativa das heurísticas, Bianchi et al. [2008] propõe uma classe de métodos chamados "Heurísticas a partir de X", que funcionam em dois estágios: Extração de Estrutura, que retira da estimativa da função de valor informações sobre a estrutura do domínio; e Composição de Heurística, que encontra a heurística para a política usando as informações extraídas no estágio anterior.

5.2.1. Extração de Estrutura

Segundo Bianchi et al. [2008], pode-se extrair uma heurística útil a partir do momento em que o agente está recebendo reforços do ambiente. A técnica de extração utilizada para os métodos de seleção de ações baseia-se na seguinte descrição.

O ambiente experimental desse estudo possui algumas características importantes: assumindo que o algoritmo *Q-Learning* já quase convergiu, o Predador só alcança o Agente se este escolher uma ação que forme um par estado-ação de perigo. Tal situação é caracterizada quando o Agente tenta executar um movimento inválido (para fora do ambiente), aumentando suas chances de ser capturado, pois continuará na mesma posição e o Predador caminhará um passo em sua direção.

Considerando o ambiente ilustrado na Figura 2, o número total de pares estado-ação é (= Posições possíveis do Agente $\times$ Posições possíveis do Predador $\times$ Ações = 25 $\times$ 25 $\times$ 8 = 5.000). A análise individual de cada posição do ambiente experimental permite a identificação das posições da grade que correspondem a estados com características distintivas, ainda conforme a Figura 2. Considerando:

- Identificador: identificação da posição que está sendo analisada e calculada na grade na Figura 2,
- QOG: quantidade de ocorrências do respectivo identificador na grade,
- QPP: quantidade de posições em que o predador pode causar perigo caso o agente execute uma ação que o leve aos limites do ambiente,
- QAL: quantidade de ações que levam o agente aos limites do ambiente,

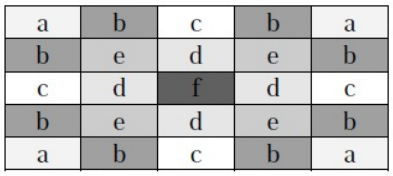

Figura 2. Identificação das posições do ambiente experimental.

- QPQ: quantidade de posições em que o predador pode estar e causar perigo em sua próxima movimentação,

tem-se:

Identificador = QOG x [(QPP x QAL) + QPQ]
a = 4 x [(4 x 5) + 5] = 100
b = 8 x [(6 x 3) + 6] = 192
c = 4 x [(6 x 3) + 9] = 108
d = 4 x [(0 x 0) + 11] = 44
e = 4 x [(0 x 0) + 7] = 28
f = 1 x [(0 x 0) + 16] = 16
Total = 488 pares estado-ação

O agente deverá evitar as ações que levam aos limites do ambiente, e a informação a ser extraída da estrutura durante os treinos é definir portanto quais ações evitam que o agente seja levado a estes limites. Essas ações são anotadas em uma estrutura $A^H(x)$, inicializada com valores 1. Para cada ação que evita que o agente seja levado aos limites, o par estado-ação continua com o valor 1, caso ele tome alguma ação que o leve aos limites, recebe o valor 0.

5.2.2. Composição de Heurística

As heurísticas são extraídas durante o treino, e é nesse período que a estrutura $A^H(x)$ é construída. Com as informações da estrutura $A^H(x)$, a heurística é aplicada durante as fases de teste, influenciando na decisão do agente e permitindo que a aceleração seja observada durante todo o estágio do aprendizado. Na aplicação da heurística, $A^H(x)$ é analisada, e se o valor respectivo do par estado-ação for igual a 1, indica que a ação não é perigosa. Então, $H(x, a)$ é atualizada de acordo com a Equação 11.

$$H(x,a) = \begin{cases} \max_a [Q(x,a)] - Q(x,a) + \eta & \text{se } A^H(x) = 1; \\ 0 & \text{caso contrário.} \end{cases} \quad (11)$$

As ações não-perigosas são selecionadas seguindo as regras dos métodos de seleção de ações.

6. Experimentos

Para gerar os resultados foram executados testes sobre 1.000 episódios intercalados com períodos de treinos de 10.000 episódios. Os dados demonstrados nos gráficos são gerados nos períodos de testes. No eixo x os valores são multiplicados por 10^5, por exemplo,

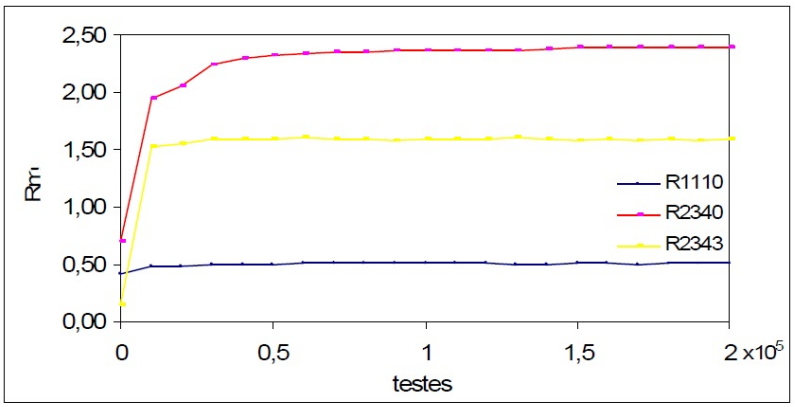

Figura 3. Maximiza Maior Satisfação.

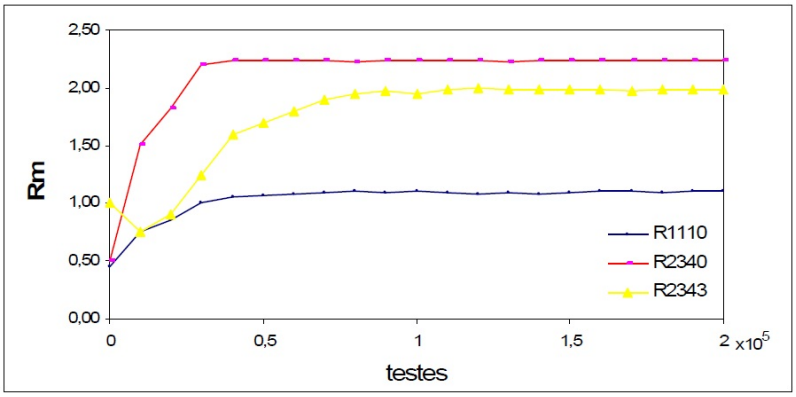

Figura 4. *W(x,a)-Learning.*

$1,5$ equivale a 150.000 testes e o eixo y apresenta os valores do reforço médio R_m. Os parâmetros utilizados para a atualização dos valores-Q foram α (conforme Equação 3), $\gamma = 0,9$ e ε variando linearmente de $0,4$ a 0 durante a primeira metade dos episódios de cada período de treino. Os valores de ξ e β foram definidos como sendo iguais a 1.

A Figura 3 apresenta os testes de desempenho com o método Maximiza Maior Satisfação nos ambientes R1110, R2340 e R2343. O desempenho nos testes com o método *W(x,a)-Learning* é visualizado na Figura 4.

A Figura 5 apresenta a comparação do desempenho do método Maximiza a Maior Satisfação com e sem heurística, no ambiente R2340. Pode-se notar que o reforço médio gerado é sensivelmente maior com a utilização de heurística. A comparação entre os testes realizados com o método *W(x,a)-Learning* com e sem heurística no ambiente R2340 é apresentada na Figura 6. Com a utilização da heurística, também foram obtidos reforços médios maiores. Tal resultado se verificou também em testes comparativos realizados nos outros dois ambientes (R1110 e R2343), com melhoria similar de desempenho causado pelo uso da heurística na seleção de ações.

O aumento nos valores de reforço médio obtidos com a utilização de heurística comprova que a heurística é adequada e que as informações retiradas do domínio são

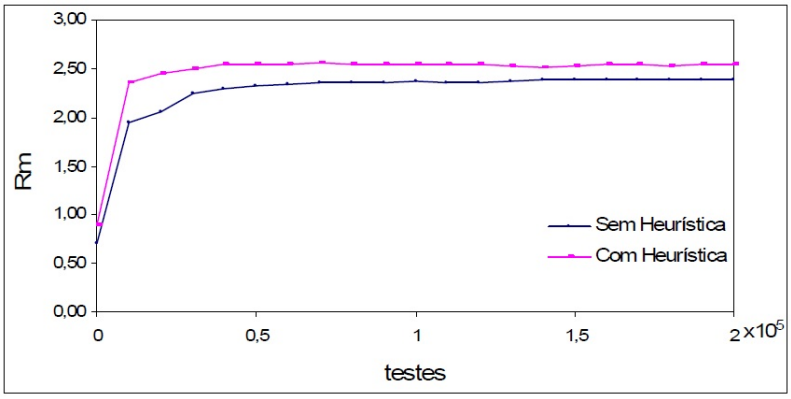

Figura 5. Maximiza a Maior Satisfação, com e sem heurística.

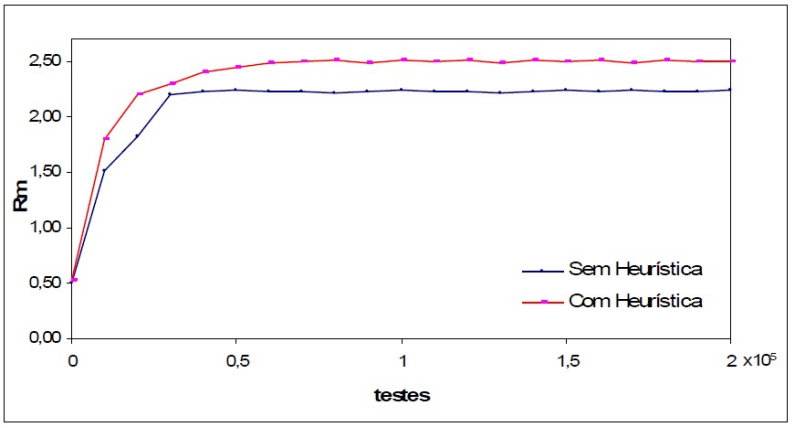

Figura 6. *W(x,a)-Learning* com e sem heurística.

suficientes para acelerar o aprendizado. O teste estatístico T de Student [Spiegel, 1998] foi utilizado nos resultados obtidos, e comprovou que os desempenhos obtidos com e sem heurística diferem com nível de confiança maior que 90%.

7. Conclusões

Esse estudo propôs a aceleração do aprendizado por reforço em sistemas com múltiplos objetivos, aplicando uma heurística para seleção de ações em dois métodos de seleção de ações conhecidos. A heurística aplicada explora características retiradas do ambiente durante o processo de aprendizado, e os resultados mostraram que os métodos que a utilizaram apresentaram melhores resultados que os métodos originais, confirmando a afirmação em Bianchi et al. [2008] que, mesmo com a utilização de heurísticas adequadas simples, o aprendizado por reforço pode ser acelerado.

Deve-se porém considerar que a heurística considerada nesse estudo explorou um problema que abrange apenas os pares estado-ação de perigo, que equivalem a 10% dos pares estado-ação possíveis do ambiente considerado. Isso sugere que resultados ainda melhores podem ser obtidos com a análise e aplicação de heurísticas mais sofisticadas, cujo estudo é sugerido como trabalho futuro.

Referências

Barrett, L. e Narayanan, S. (2008). Learning all optimal policies with multiple criteria. In *Proceedings of the 25th International Conference on Machine learning*, ICML '08, pages 41–47, New York, NY, USA. ACM.

Bianchi, R. A. C., Ribeiro, C. H. C. e Costa, A. H. R. (2008). Accelerating autonomous learning by using heuristic selection of actions. *Journal of Heuristics*, 14(2):135–168.

Humphrys, M. (1997). *Action selection methods using reinforcement learning*. PhD thesis, University of Cambridge, Trinity Hall, Cambridge, UK.

Mitchell, T. (1997). *Machine Learning*. McGraw Hill, New York.

Russell, S. e Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ.

Singh, S. e Cohn, D. (1998). How to dynamically merge markov decision processes. In *Proceedings of the 1997 conference on Advances in neural information processing systems 10*, NIPS '97, pages 1057–1063, Cambridge, MA, USA. MIT Press.

Sousa, C. A. O. (2006). *Aprendizagem por Reforço de Sistemas com Múltiplos Objectivos: o Problema da Selecção de Acções*. Tese de Doutorado, Universidade Técnica de Lisboa, Instituto Superior Técnico, Lisboa.

Spiegel, M. R. (1998). *Statistics*. McGraw-Hill.

Sprague, N. e Ballard, D. (2003). Multiple-goal reinforcement learning with modular sarsa(o). In *Proceedings of the 18th international joint conference on Artificial intelligence*, pages 1445–1447, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Sutton, R. e Barto, A. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

Vamplew, P., Yearwood, J., Dazeley, R. e Berry, A. (2008). On the limitations of scalarisation for multi-objective reinforcement learning of pareto fronts. In *Proceedings of the 21st Australasian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence*, AI '08, pages 372–378, Berlin, Heidelberg. Springer-Verlag.

Watkins, C. e Dayan, P. (1992). Q-learning. *Machine Learning*, 8:279–292.