Aprendizado por Reforço em Sistemas Multiagente: Generalizando Tarefas Conjuntas

Samuel J. Waskow e Ana L. C. Bazzan

Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil
sjwaskow,bazzan@inf.ufrgs.br

Abstract. Reinforcement learning is an efficient, widely used technique in single-agent problem. In the context of multiagent systems, whose data volume tends to be huge, the use of standard techniques of reinforcement learning may not provide an appropriate solution. This paper presents a function approximation technique aiming to generalize state space with limited set of values. The results illustrate the capacity to employ this technique in multiagent learning scenarios.

1 Introdução

Como agentes podem aprender o que fazer quando não existe nenhuma entidade dizendo qual ação deve ser executada em cada circunstância? O dinamismo de alguns cenários pode exigir a capacidade de aprendizado para lidar com fatores imprevisíveis existentes nestes cenários. Um dos métodos de aprendizagem de máquina, quando um agente aprende a partir de recompensas e penalizações, é denominado aprendizado por reforço. Aprendizado por reforço é uma técnica que possibilita aprendizagem através da interação com o ambiente. Esta interação permite mensurar atributos que incidem sobre as consequências de ações tomadas e o que fazer para que objetivos sejam atingidos.

Nos cenários monoagente, como o próprio nome diz, um único agente é responsável pela realização de uma tarefa, enquanto que em um cenário multiagente um grupo de agentes é responsável pela realização de uma tarefa. Este grupo de agentes pode ter aprendizado de maneira independente, ou seja, cada agente aprende sozinho e compartilha ou não o que aprendeu com os demais ou, este aprendizado pode ser centralizado, onde vários agentes estão engajados em um mesmo processo de aprendizado.

Problemas de aprendizado por reforço quando aplicados em cenários monoagente são popularmente resolvidos com a maximização das recompensas de cada estado do ambiente. Em um cenário multiagente onde o aprendizado pode ser centralizado, os valores utilizados para se estimar uma função de valor podem gerar uma explosão combinatorial do número de pares estado-ação. A aplicação de técnicas de aproximação de função sobre estes pares estado-ação possibilita

a generalização de uma única experiência para os demais estados ainda não visitados.

Considerando uma tarefa que deve ser realizada por mais de um agente e o custo do aprendizado conjunto, a contribuição deste trabalho é implementar uma técnica de generalização para acelerar o processo de aprendizado em um cenário complexo. Este trabalho está organizado da seguinte forma: na seção 2 são fundamentados os conceitos usados neste trabalho. A seção 4 descreve a abordagem proposta no cenário em que foram implementadas as técnicas de aprendizado. A seção 5 detalha os experimentos que foram executados. A seção 6 apresenta os trabalhos relacionados e, a seção 7 apresenta uma conclusão sobre os resultados obtidos.

2 Aprendizado por Reforço

Os problemas que envolvem aprendizado por reforço são modelados usualmente como Processos de Decisão de Markov (MDP). Os MDP's são particularmente importantes para o aprendizado por reforço na medida que este método modela o ambiente através da quádrupla $\{S, A, T, R\}$ sendo S um conjunto finito de estados; A um conjunto finito de ações; $T : S\ X\ A\ X\ S \rightarrow [0, 1]$ uma função de transição que especifica a probabilidade de observar determinado estado após ser tomada alguma ação em algum estado; e $R(s, a) \rightarrow \mathbb{R}$ uma função de recompensa que especifica a recompensa esperada após ter tomada uma ação em determinado estado.

A resolução de um MDP consiste em encontrar uma política de ações que garanta o maior ganho esperado para o sistema, dado um determinado estado inicial. O MDP é definido por uma classe de tarefas e algoritmos em que sistemas aprendem através da relação mapeada por: $\pi : S \rightarrow A$ maximizando uma avaliação escalar (reforço) e sua avaliação do ambiente através de um agente. Denotamos $\pi(s)$ como a ação recomendada dado que o sistema esteja no estado s e π^* como a política que gera o maior ganho esperado.

A aprendizagem Q [1] consiste em uma adaptação da aprendizagem de diferença temporal para o caso de não possuirmos uma política de ação fixa. A aprendizagem Q aprende uma representação ação-valor, chamada de valor Q, ao invés de aprender diretamente utilidades. A grande vantagem de se armazenar informações de utilidade na forma de valores Q é que o aprendizado passa a não depender da existência de um modelo de mundo. Por essa razão, costuma-se dizer que a aprendizagem Q é *livre de modelo*. Para que se possa aprender ações sem que se conheça um modelo de transições, podemos relacionar o valor de Q para um estado diretamente com o valor para os estados vizinhos. Dessa forma, a atualização do valor Q é dada pela equação 1, onde α é a taxa de aprendizagem que determina a medida em que informações recém-adquiridas irão substituir informações antigas, γ é o fator de desconto que regula a importância de futuras recompensas, e r é a recompensa fornecida no tempo $t + 1$.

$$Q(s, a) \leftarrow Q(s, a) + \alpha\Big[r + \gamma\, max_{a'}(s', a') - Q(s, a)\Big] \tag{1}$$

Essa equação será calculada sempre que a ação a for executada no estado s e levar ao estado s'. Uma política π é um mapeamento de cada estado s e ação a para a probabilidade $\pi(s, a)$ de efetuar a ação a.

O aprendizado em ambientes monoagente depende exclusivamente do conjunto de suas percepções e ações em relação ao conjunto de estados. Entretanto, mesmo que todas as condições sejam favoráveis ao aprendizado, este pode não ser eficaz quando aplicado em um ambiente que exija cooperação. Além disso, apesar da necessidade de exploração por parte de agentes, é inaceitável exigir que se visite todo o espaço de estados. A representação tabular dos valores Q, indexados por estado e ação, se torna rapidamente inviável em domínios com um grande número de estados. Jogos como o xadrez e o gamão poderiam facilmente exigir ao armazenamento de 10^{50} ou 10^{120} estados, respectivamente. Além disso, é inaceitável exigir que alguém tivesse que visitar todos esses estados a fim de aprender como jogar. Uma possível solução para este problema consiste na utilização de formas alternativas à representação tabular.

Uma destas alternativas é o uso de uma função de aproximação que poderia ser $V(s) = \theta_1 f_1(s) + \theta_2 f_2(s) + \ldots + \theta_n f_n(s)$, onde $f(s)$ são funções de base e θ são parâmetros variáveis a serem descobertos. Além da imensa compactação na representação da função de utilidade, o uso de uma função de aproximação que permite que o agente generalize a experiência obtida para estados ainda não visitados. A atualização dos parâmetros de $V(s)$ pode ser feita através de uma variedade de métodos existentes para aproximação de função, incluindo discretização simples, aproximadores baseados em instância e caso, funções de base radiais e redes neurais. Contudo, a técnica de generalização escolhida neste trabalho foi codificação por *Tile Coding* [2] por causa de sua robustez e por ser usualmente empregado pela comunidade de aprendizado por reforço.

3 Generalização do espaço de estados e ações

Tile Coding é um método linear para aproximar funções de valores em aplicações práticas de aprendizado por reforço, cujos conjuntos de estados e ações são usualmente contínuos, ou seja, tendem a ser conjuntos muito grandes ou infinitos. Esta técnica generaliza o espaço de estados em partições denominadas *tiling*, e cada partição é composta por subpartições denominadas *tiles*. Combinando aproximação linear com uma função de mapeamento $\phi_i(s)$ que transforma o estado s em um vetor de N características binárias, $[\phi_1, ..., \phi_N(s)]^T$. Cada *tile* recebe uma característica binária indicando se a posição de um agente se encontra ou não sobre aquele *tile*. Assim, dado um estado s, o i-nésimo *tile* associado com o componente $\phi_i(s)$ de $\boldsymbol{\phi}$ é computado através da equação 2.

$$\phi_i(s) = \begin{cases} 0 \text{ se } s \notin tile_i, \\ 1 \text{ se } s \in tile_i. \end{cases} \tag{2}$$

Desta forma, o valor de $Q(s, a)$ é obtido através de $\phi(s) \times \theta_a$, onde θ_a é um vetor de N parâmetros associados com a ação a do estado s. Para aproximar a

função que determina o valor de um estado utiliza-se sobreposição de *tilings* e, para cada *tiling* são computadas variáveis que indicam o peso de cada *tile*.

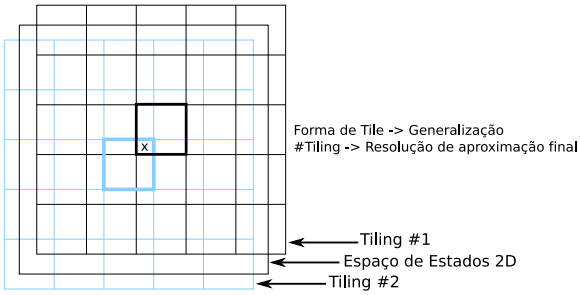

Fig. 1. *Tilings* sobrepostos [3]

Conforme a figura 1, o valor aproximado de um determinado estado é obtido pela soma dos pesos θ_a dos *tiles* (situados em diferentes *tilings*) em que o estado está localizado. Sendo que o número de *tilings* sobrepostos influencia diretamente na função que se deseja aproximar e no custo computacional.

Os *tilings* das extremidades representam os limites inferior e superior do intervalo de *tilings*. A partir destes limites, a quantidade de *tilings* utilizados é definida de acordo com o número de ações possíveis dos agentes. Caso a quantidade de atributos de um estado exceda 3 dimensões, o modo de generalização pode ser ajustado para *tiles* hiperplanos.

Através dos subsídios que *Tile Coding* fornece é possível estender métodos de predição de valor usando aproximação de função para métodos de controle. Para implementação de *Tile Coding* em um espaço de estados e ações um dos métodos utilizados é o algoritmo de controle de aproximação de função $Q(\lambda)$ proposto por Watkins's em [3], onde λ é o parâmetro que indica o decaimento de vestígios temporais. Este algoritmo processa o conjunto de parâmetros pertencentes ao par estado-ação corrente como: características ϕ_a, ações possíveis e traços de eligibilidade $\boldsymbol{e}$ de cada *tile*.

Mesmo aproximando o valor de um estado, a dependência da avaliação das ações cria a necessidade de exploração do ambiente. Esta necessidade é diretamente relacionada com um método que introduza um grau de exploração e não considere apenas decisão gulosa. Por exemplo, o método ε-guloso se comporta gulosamente na maior parte das transições; porém, com uma pequena probabilidade regulada pelo parâmetro ε uma ação é selecionada aleatoriamente. Conforme descrito na equação 3, onde p é um número gerado aleatoriamente para ser comparado ao fator ε.

$$\pi(s) = \begin{cases} arg\,max\,_{a \in A(x)}\, \hat{Q}(s,a) \text{ se} & \varepsilon \geq p(0,1) \\ p(a \in A(x)) \text{ se} & \varepsilon < p(0,1) \end{cases} \tag{3}$$

Além da estratégia para seleção de ações anteriormente descrita existem alguns métodos que atuam indiretamente na transição entre estados, como traços de eligibilidade e método do ator e crítico. Traços de eligibilidade ou vestígios [4] são registros temporários da ocorrência de um evento, como por exemplo, a visita de um estado ou a realização de uma ação. Isto possibilita que os parâmetros associados a estes eventos "elegíveis" sejam modificados durante o treinamento. Métodos ator e crítico são métodos de Diferença Temporal (DT) em que a política de seleção é alocada em uma estrutura de memória independente da função de valor. Esta estrutura é denominada *ator*, pois é responsável pela seleção de ações. A função de valor estimada é chamada de *crítico* justamente porque critica as ações realizadas pelo ator e expõe esta crítica na forma de um erro DT δ. Erro DT é uma avaliação que o *crítico* realiza para aprender uma função de valor do novo estado verificando se o resultado é melhor ou pior que o esperado.

4 Uso de Aproximação de Função em um Cenário Multiagente

Neste trabalho foi escolhido um cenário do tipo Presa-Predador. Este cenário proposto por [5] consiste em um caso de tarefa conjunta envolvendo dois agentes "predadores", cujo objetivo é capturar um agente "presa" em uma grade. Nesta grade não há limitações (paredes) no deslocamento e percepção dos agentes, pois este ambiente foi modelado toroidalmente. Em cada unidade discreta de tempo, cada agente (predador ou presa) possui quatro ações possíveis: se movimentar para cima, para a direita, para baixo e para a esquerda.

O estado terminal de cada episódio (captura da presa) é alcançado somente se ambos predadores estiverem em células adjacentes a presa (figura 3(b)), sendo que a percepção destes agentes é conjunta, ou seja, os predadores cooperam ativamente compartilhando percepções. Cada ação é uma ação conjunta dos dois agentes, assim como a recompensa recebida é recompensa total dos agentes. A cada passo, caso a presa não seja capturada, os predadores são penalizados com -1.0 pontos. A presa se desloca pelo ambiente escolhendo uma das 4 ações disponíveis aleatoriamente. Caso o estado objetivo seja alcançado (captura da presa), a recompensa recebida por ambos é 10.0 pontos. Não há restrições quanto aos agentes ocuparem a mesma célula. A cada novo episódio a posição dos agentes é conforme mostrado na figura 2.

A percepção de um predador é composta por um número de células em torno da célula na qual o predador se encontra. Nos experimentos realizados o campo perceptivo atinge um raio de 2 níveis de células em torno do agente. Assim, conforme a figura 3(a), cada agente percebe 25 células da grade. O estado perceptivo do predador A_1 é representado por coordenadas cartesianas, onde (x, y) é a posição da presa B ou outro predador A_2 relativa à posição de A_1, posição esta que é considerada como $(0, 0)$. Por exemplo, na figura 3(a) está exemplificado o estado $(2, 2)$ representando que a presa se encontra na célula superior direita em relação ao estado do predador. Caso não haja nenhum agente

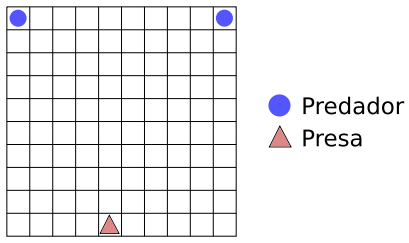

Fig. 2. Ambiente Grid 10 por 10 em seu estado inicial.

dentro do campo perceptivo de um predator, o estado é arbitrado como estado-nulo conforme [6]. Em contrapartida, caso existam um ou mais agentes no campo perceptivo, existe a necessidade do mapeamento de um estado correspondente à esta situação.

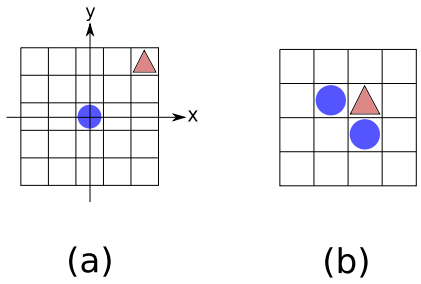

Fig. 3. (a): estado perceptivo representado por (2,2). (b): possível posição de captura da presa pelos predadores

No cenário anteriormente descrito percebe-se que o processo de coordenação entre dois agentes exige uma estrutura de dados com dimensões muito maiores do que uma estrutura necessária para aprendizado independente. Para um único agente, a dimensão da tabela $\hat{Q}(s, a)$ é de 26 estados (25 células $+1$ estado nulo). Considerando a combinação de estados entre os dois predadores, a dimensão passa a ser de 26^2 estados por agente, pois as posições tanto da presa quanto do outro predador devem ser referenciadas, totalizando 26^4 estados. Como cada agente pode realizar 4 ações, a combinação de ações conjuntas é 4^2. Desta forma, ambos os predadores geram uma tabela $\hat{Q}(s, a)$ de dimensão 7.311.616 ($26^4 \times 4^2$).

A tabela $\hat{Q}(s, a)$ neste cenário representa as coordenadas cartesianas (x, y) relativas entre os predadores e a presa. Os atributos que identificam cada par estado-ação são as posições relativas entre os agentes. Por exemplo, no caso de ambos predadores estarem situados na mesma célula que a presa, as coordenadas relativas entre os 3 agentes é $(0, 0)$, pois todos se encontram na mesma célula. Neste caso, o estado é identificado como $(0, 0), (0, 0), (0, 0), (0, 0)$. Sendo

o primeiro e segundo pares de coordenadas referentes aos primeiro e segundo predadores, respectivamente.

Na prática, os requisitos formais para a convergência da aprendizagem Q neste cenário dificilmente acontecem, pois o espaço de estados e ações é grande demais para permitir uma representação independente do valor de cada estado e ação. Mesmo conseguindo a representação de todos os estados e ações possíveis, o emprego desta técnica leva a um número de passos muito grande por episódio, sendo inviável utilizar esta técnica em aplicações de tempo real.

5 Experimentos

O número de conjuntos de *tilings*, conforme indicado por [3], é parametrizado de acordo com o número de ações possíveis a cada unidade de tempo discreto. No caso deste trabalho são 4^2. Devido ao compartilhamento sensorial por parte dos predadores, o grande espaço de estados faz com que o aprendizado inicial seja lento. Os parâmetros de aprendizagem Q foram ajustados para $\alpha = 0.8$ e $\gamma = 0.9$. Segundo [5], estes valores foram escolhidos objetivando que os agentes considerem informações mais recentes e recompensas a longo prazo, respectivamente. O parâmetro do método de seleção de ações ε-guloso foi ajustado para 0.1, este valor foi escolhido para permitir uma pequena característica exploratória e não percorrer somente os estados com as recompensas mais elevadas.

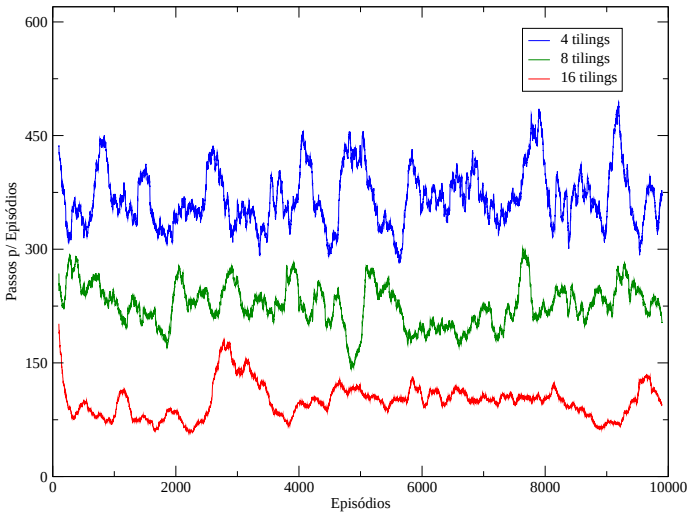

Fig. 4. Número de passos por episódio em função do número de tilings

Primeiramente, foram realizados testes para verificar a influência do número de *tiles* sobre as aproximações dos valores da tabela. Para isso, foram testadas aproximações de 16 *tilings* com divisões de 20 a 26 unidades por *tiling*. Porém foi

verificado que o uso de diferentes quantidades de *tiles*, respeitando o intervalo utilizado, não exerce influência significativa sobre as aproximações de valores. Sendo assim, foram descartados experimentos relativos ao número de *tiles* e foram realizados somente experimentos com sobreposição de *tilings*.

Na figura 4 é possível observar a comparação entre o número de *tilings* (4, 8, 16) e o número de passos por episódio. Quanto maior for o número de *tilings*, mais detalhada é a representação dos estados e, consequentemente, o processo de aprendizagem é agilizado em função do menor número de iterações para atingir o estado terminal.

Em relação a recompensa recebida por episódio, conforme visualizado na figura 5, o número de *tilings* utilizados para generalizar o espaço de estados influenciam diretamente na recompensa recebida por episódio. Quanto maior for o número de tilings, mais detalhada é a representação de um estado e maior é a recompensa por episódio.

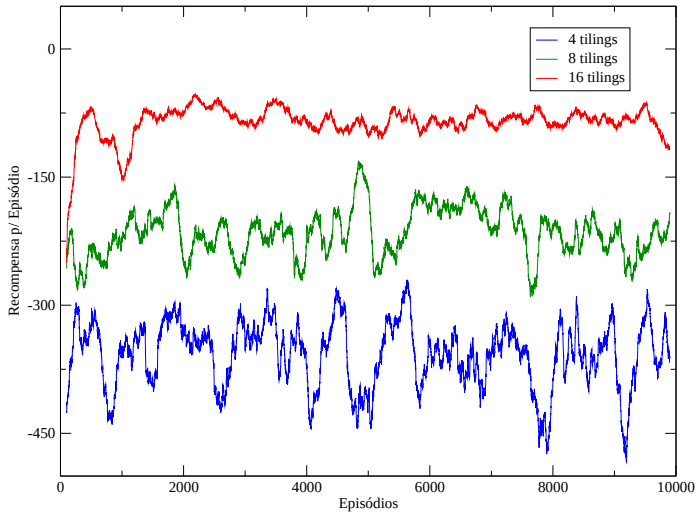

Fig. 5. Recompensa por episódio em função do número de tilings

Apesar de não ser visualizada uma curva de convergência de aprendizado em ambos os gráficos, o número de passos por episódio permanece oscilando entre valores específicos de cada quantidade de *tilings* utilizada. Isto é causado pelo movimento da presa, pois como a presa se desloca aleatoriamente pela grade, o estado terminal (captura) é alcançado com um número de passos diferentes.

Ainda é possível observar que, em ambos experimentos realizados, o número de *tilings* faz com que a oscilação do número de passos e da média de recompensas por episódio seja menor. Quanto maior o número de *tilings*, menor a oscilação. Por exemplo, na figura 5, a linha que representa a recompensa por episódio correspondente a 16 *tilings* possui uma oscilação menor do que a linha que

representa 4 *tilings*. Isto deve-se ao fato de que a utilização de um número maior de *tilings* reduz a generalização do espaço de estados, ou seja, os estados são representados de uma forma mais detalhada. Com isso, o aprendizado é mais efetivo, pois os parâmetros utilizados (*tilings*) possuem uma aproximação muito próxima de um estado na tabela Q, por exemplo.

6 Trabalhos Relacionados

No trabalho [5] é feita a comparação entre o desempenho de aprendizado independente e conjunto. Entretando, a forma com que o autor lida com a questão do aprendizado é direcionada à comparação de aprendizado de agentes com diferentes tipos de percepções e ações. No caso de realização de tarefas conjuntas, as técnicas de aprendizado (Aprendizagem Q) foram implementadas de maneira eficiente, porém, por terem sido aplicados métodos padrão de aprendizado por reforço, a realização de tarefas conjuntas gera uma tabelas de estados e ações muito grande. O problema de coordenação em sistemas multiagente (SMA) é conduzido por [7] pela utilização de jogos para o equilíbrio do aprendizado em agentes através da realização de tarefas conjuntas. Entretanto, este trabalho utiliza um cenário multiagente com um número de estados e ações muito limitado para implementar o método proposto.

Como a realização de tarefas conjuntas enfrenta o problema da dimensionalidade do espaço de estados, alguns autores buscam alternativas para resolver este problema sem comprometer o processo de aprendizado. Na abordagem proposta por [8], os agentes escolhem a melhor ação conjunta sem explicitamente considerar cada ação possível do espaço conjunto de ações. Esta abordagem desfavorece ações exploratórias e, por utilizar iteração de política dos mínimos quadrados, requer o conhecimento prévio do vetor de características $\boldsymbol{\phi}$ de cada agente para todo espaço de estados. Isto resulta em uma explosão combinatorial de ações, pois além desta avaliação ser realizada para cada ação, é avaliada também cada ação recomendada pela política utilizada.

7 Conclusão

É essencial que as novas técnicas a serem usadas em aprendizado multiagente sejam estendidas. Contudo, estas novas técnicas não podem limitar a capacidade de adaptação dos modelos de aprendizado em função da interação de agente e ambiente. O limite desta adaptação está diretamente associado ao número de possibilidades entre ações de agentes e alterações no ambiente. Assim, visando minimizar o problema da dimensionalidade gerado pela abordagem relacional de representação de estados e ações neste trabalho foi proposto o uso de aproximação de funções como ferramenta para otimizar o processo de aprendizagem conjunta em SMA.

Através dos testes realizados e de posse dos resultados obtidos observou-se a utilização de uma técnica de generalização em um problema de aprendizado caracterizado por espaço de estados muito grande, cujo uso de um modelo

linear facilita a aproximação de funções de estados e ações. Este método permite aplicação de aprendizado multiagente centralizado em alguns cenários não tratáveis por técnicas comuns de aprendizado por reforço.

A capacidade de generalização do método empregado neste trabalho possui limitações com relação ao ajuste de parâmetros utilizados para os algoritmos de aproximação linear (*tile coding*), pois estes algoritmos são muito dependentes das características do problema a ser solucionado, o que dificulta a definição de um esquema de *tiling* independente do ambiente. Neste sentido, como trabalho futuro deseja-se aprimorar esta técnica alinhando as características que *tile coding* utiliza com características que sejam mais adequadas a uma abordagem livre de contexto.

References

[1] Watkins, C.J.C.H., Dayan, P.: Q-learning. Machine Learning **8**(3) (1992) 279–292

[2] Sutton, R.S.: Generalization in reinforcement learning: Successful examples using sparse coding. In David S. Touretzky, M.C.M., Hasselmo, M.E., eds.: Advances in Neural Information Processing Systems. Volume 8., Cambridge, MA, MIT Press (1996) 1038–1044

[3] Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA (1998)

[4] Singh, S.P., Sutton, R.S., Kaelbling, P.: Reinforcement learning with replacing eligibility traces. In: Machine Learning. (1996) 123–158

[5] Tan, M.: Multi-agent reinforcement learning: Independent vs. cooperative agents. In: Proceedings of the Tenth International Conference on Machine Learning (ICML 1993), Morgan Kaufmann (June 1993) 330–337

[6] Kok, J.R., Vlassis, N.: Sparse cooperative q-learning. In: Proceedings of the 21st. International Conference on Machine Learning (ICML), New York, USA, ACM Press (July 2004) 481–488

[7] Claus, C., Boutilier, C.: The dynamics of reinforcement learning in cooperative multiagent systems. In: Proceedings of the Fifteenth National Conference on Artificial Intelligence. (1998) 746–752

[8] Guestrin, C., Lagoudakis, M., Parr, R.: Coordinated reinforcement learning. In: In Proceedings of the ICML-2002 The Nineteenth International Conference on Machine Learning. (2002) 227–234