APRENDIZADO POR REFORÇO ACELERADO POR HEURÍSTICAS NO DOMÍNIO DO FUTEBOL DE ROBÔS SIMULADO

LUIZ A. CELIBERTO JR [†*], REINALDO A. C. BIANCHI [*], JACKSON P. MATSUURA [†]

[†]*Departamento de Sistemas e Controle, Instituto Tecnológico da Aeronática*
Pça Mal-do-Ar Eduardo Gomes, 50
São José dos Campos – SP – Brasil CEP: 12228-900

[*]*Departamento de Engenharia Elétrica, Centro Universitário da FEI*
Av. Humberto de Alencar Castelo Branco, 3972
São Bernardo do Campo – SP – Brasil CEP: 09850-901

E-mails: celibertojr@uol.com.br, rbianchi@fei.edu.br, jackson@ita.br

Abstract — Reinforcement Learning is a very well known technique for the solution of problems when the agent needs to act with success in an unknown environment through trial and error. However, this technique is not efficient enough to be used in applications with real world demands of the real world, due to the time that the agent needs learn. This work presents the use of the heuristics accelerated reinforcement learning in the mobile robotics domain, using the RoboCup simulation 2D as a testbed. The experiences were made without the use of the heuristics and later with the use of the heuristics and their results compared through a statistical analysis. The results indicate some advantages in the use of the heuristics, making possible the definition of important guidelines for the application of the use of heuristics in the domain of the simulated soccer of robots.

Keywords— Artificial intelligence, Intelligent Robotics, Mobile Robotics, Reinforcement Learning, RoboCup soccer 2D.

Resumo— O Aprendizado por Reforço é uma técnica muita conhecida para a solução de problemas quando o agente precisa atuar com sucesso em um local desconhecido por meio de tentativa e erro. Porém, esta técnica não é eficiente o bastante para ser usada em aplicações com exigências do mundo real, devido ao tempo que o agente leva para aprender. Este artigo apresenta o uso do Aprendizado por Reforço, acelerado por heurística, no domínio da robótica móvel, utilizando para testes a plataforma do *RoboCup* simulação 2D. Foram realizadas experiências comparando o uso de um algoritmo de aprendizado por reforço bem conhecido, o Q-learning, com uma versão acelerada por heurísicas do mesmo algoritmo, sendo que os resultados indicam algumas vantagens no uso das heurísticas, possibilitando a definição de diretrizes importantes para a aplicação do uso de heurísticas no domínio do futebol de robôs simulado.

Palavras-chave— Inteligência Artificial, Robótica Inteligente, Robótica Móvel, Aprendizado por Reforço, RoboCup 2D.

1 Introdução

Aprendizado por Reforço (AR) é uma técnica muito atraente quando se deseja solucionar uma variedade de problemas de controle e planejamento quando não existem modelos disponíveis *a priori*, pois o agente irá aprender a cumprir uma função de maneira correta em um ambiente desconhecido por meio de tentativa e erro (Pegoraro, 2001).

No aprendizado por reforço, o agente aprende por meio da interação direta entre o agente e o ambiente através de recompensas. Estas recompensas são dadas através de reforços positivos e negativos, que são usadas para sinalizar ao agente se ele esta tomando as ações corretas ou não. O objetivo do agente sempre será acumular o máximo reforço positivo.

Porém, aprender não é suficiente: o agente precisa aprender rapidamente, pois o ambiente em que ele esta localizado pode sofrer constantes mudanças. Infelizmente, o aprendizado por reforço é muito lento, necessitando de enorme quantidade de ações para que um agente aprenda corretamente: a convergência dos algoritmos de AR só pode ser atingida após uma extensiva exploração do espaço de estados-ações.

Bianchi (2004) mostrou que a velocidade de convergência de um algoritmo de AR pode ser acelerada ao se utilizar funções heurísticas para guiar a exploração do espaço de estados-ações. Ele também propôs diversos algoritmos acelerados por heurísticas, entre eles, o *Q-Learning* Acelerado por Heurísticas, baseado no conhecido algoritmo *Q-Learning* (Watkins, 1989).

O objetivo deste artigo é apresentar o uso do aprendizado por reforço acelerado por heurísticas no domínio da robótica móvel, utilizando para testes um ambiente de futebol de robôs simulado denominado *RoboCup* 2D (Noda, 1995) para fazer o aprendizado e avaliação de seu funcionamento. O *RoboCup* 2D é uma das categorias das competições da *RoboCup*. A *RoboCup* (Kitano et al.,1995), foi inicialmente proposta para ser um meio de divulgação da robótica e da pesquisa em inteligência artificial, e fornecer meios para a avaliação de várias teorias, algoritmos e arquitetura, servindo também como uma ferramenta para a integração e estudos de como várias tecnologias podem trabalhar em conjunto (BOER; Kok, 2002), a Figura 1, mostrada um exemplo de um jogo no *RoboCup* 2D

Para este trabalho, foram implementados dois sistemas dotados de capacidade de aprender: um utilizando um goleiro e o outro utilizando um atacante, ambos irão jogar contra um adversário sem aprendizado que é implementado com um algoritmo básico.

Nos dois casos de aprendizado foi inicialmente utilizado um algoritmo de AR tradicional e depois utilizado um algoritmo acelerado por heurísticas. As heurísticas utilizadas visam melhorar o aprendizado do agente e foram escolhidas de modo a determinar a melhor ação a ser feita em um determinado estado.

Figura 1. Jogo sendo realizado no *RoboCup* 2D.

Este artigo é organizado da seguinte maneira: na próxima seção são apresentadas, de forma sucinta, características do algoritmo de aprendizado por reforço e a seção 3 introduz a aceleração heurística. A seção 4 descreve a proposta da experiência, a seção 5 discute os resultados e a última seção apresenta a conclusão deste trabalho.

2 Aprendizado por Reforço

No Aprendizado por Reforço, um agente sem conhecimentos prévios aprende por meio de interações com o ambiente, recebendo recompensas por suas ações e assim descobrindo a política ótima para a resolução de um determinado problema. A suposição principal do Aprendizado por Reforço é a existência de um agente que pode aprender a escolher suas ações que resultarão em um melhor resultado futuro na realização de uma tarefa (Pegoraro, 2001).

O aprendizado por reforço é uma técnica de aprendizado não supervisionado devido a não existência de uma representação de pares de entrada e de saída. Para cada movimentação do agente não é fornecida nenhum tipo de informação externa que ajude seu deslocamento, tirando aquela que ele mesmo percebe da sua interação com o ambiente (Kaelbling; Littman; Moore, 1996).

O Aprendizado por Reforço funciona da seguinte maneira: em um ambiente, a cada intervalo de tempo o agente executa uma ação a_t. Esta ação é determinada pela política já aprendida e faz o agente ir para o estado s_{t+1} e tendo em vista a recompensa $r_{st,at}$ que irá ganhar. A recompensa pode ser dada por valores positivos ou negativos, indicando se o agente está seguindo corretamente para o objetivo ou não. A Figura 2 apresenta um esboço do funcionamento do aprendizado por reforço.

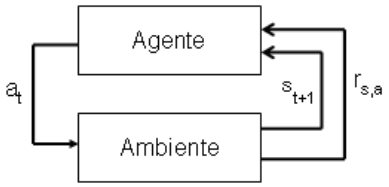

Figura 2. Aprendizado por reforço.

Entre os diversos algoritmos de Aprendizado por Reforços existentes, o mais conhecido é o *Q-Learning* (Watkins, 1989; Watkins; Dayan, 1992), considerado por vários autores um algoritmo de fácil implementação (Kaelbling; Littman; Moore, 1996) e por este motivo, utilizado em uma grande quantidade de domínios (Mitchell, 1997). No *Q-Learning*, para cada ação realizada pelo agente é computado sua recompensa e o valor esperado ao seguir a melhor política com um desconto. Esta política é aprendida por meio da interação com o ambiente e, assim, aprendidos quais as melhores ações para chegar a um objetivo. A informação da política é armazenada em uma matriz *Q(s,a)*, que guarda os valores estimados para cada par de estado,ação (Uther; Veloso, 2003).

O *Q-Learning* tem como proposta principal aprender através da interação uma política ótima π^* quando não se tem um modelo do sistema. Esta política ótima é encontrada escolhendo a ação que maximiza os valores de Q, para um determinado estado. O valor de custo de um estado (função valor *V(s)*) também é encontrado facilmente, sendo que o custo de um estado *V(s)* é o valor máximo de *Q(s,a)* de um estado *s*, para todas as ações possíveis de serem executadas nesse estado.

Nesta técnica de aprendizado uma função de estimação Q que é calculada através da Exploração on-line do agente. A função de estimação Q é computada através da equação 1:

$$Q(s, a) = Q(s, a) + \alpha (r(s_t, a_t) + \gamma \max Q_t(s_t, a_t) - Q(s, a)) \quad (1)$$

O parâmetro γ é compreendido entre os valores de 0 a 1. Caso o valor de γ esteja muito perto de 0 o agente tende a considerar apenas valores imediatos de recompensa, caso os valores sejam muito perto de 1 o agente tem em vista as recompensas futuras com o maior peso.

Para a escolha das ações é usada uma estratégia conhecida como exploração aleatória ε-Greedy. O agente irá executar a ação que contenha o maior valor Q e probabilidade 1-ε e escolhe uma ação aleatória com probabilidade ε. A regra de transição de estados será dada pela seguinte equação:

$$\pi(s_t) = \begin{cases} a_{random} & \text{se } q \le \varepsilon, \\ \arg\max_{at} \hat{Q}_t(s_t, a_t) & \text{caso contrário.} \end{cases} \quad (2)$$

Sendo q um valor aleatório e com distribuição de probabilidade uniforme, ε compreendido entre os valores de 0 e 1 é o parâmetro que irá definir a taxa de exploração e de explotação e a_{random} é a ação aleatória selecionada.

O agente, através das interações com o ambiente, irá se deslocando de um estado ao outro e assim computando a função de estimação Q até conseguir chegar ao seu objetivo. Ao conseguir chegar ao seu objetivo é dito que ele completou um episódio.

Apesar de ter sido aplicado em um vasta gama de problemas de maneira bem sucedida, uma desvantagem do Aprendizado por Reforço é o tempo que o agente leva para aprender. Em muitos casos são necessárias uma enorme quantidade de interações (ou episódios) para o agente poder

aprender sobre o ambiente onde ele está ou sobre a tarefa que deve realizar.

Devido a esta demora, é comum o uso de alguma forma de aceleração do aprendizado. A seção a seguir apresenta algumas técnicas que podem ser usadas para diminuir o tempo de aprendizado dos algoritmos de AR.

3 Aceleração do Aprendizado por Reforço

Um dos métodos de acelerar o algoritmo *Q-Learning* é aplicar uma heurística que acelere o aprendizado. Heurísticas são métodos ou principio de decisões, que indicam as melhores alternativas de ações que levem a solução de um problema mais facilmente (Pearl, 1984). Utilizar uma heurística em um aprendizado por reforço é fornecer uma ajuda ao agente para que ele possa conseguir chegar a um objetivo mais facilmente.

Bianchi (2004), define a heurística como uma técnica que no caso médio melhora a eficiência do algoritmo. Em seu trabalho, a heurística é usada para que o agente seja influenciado a escolher as ações durante o aprendizado.

Assim é obtida uma heurística definida por $H_t(s_t,a_t)$ no RL que irá indicar a importância de executar em um estado s_t uma ação a_t. A política estará fortemente vinculada a sua função heurística. Sendo assim podemos dizer que teremos uma função heurística que é definida por uma "Política Heurística" (Bianchi, 2004).

Conhecendo as informações existentes no domínio ou em estágios iniciais, pode ser definida uma heurística que poderá ser usada para acelerar o aprendizado, porém devido as características do AR o uso de uma heurística inadequada pode causar um atraso no sistema, mas não impede o algoritmo de convergir para uma política ótima (Bianchi, 2004).

Bianchi (2004) também propôs diversos algoritmos acelerados por heurísticas, entre eles, o *Q-Learning* Acelerado por Heurísticas (HAQL), baseado no conhecido algoritmo *Q-Learning*. O HAQL utilizada uma modificação da regra ε-*Greedy* para a regra de transição de estados, e incorpora a função heurística como uma somatória simples ao valor da função valor-ação. A regra de transição de estados é mostrada em 3.

$$\pi(s_t) = \begin{cases} a_{random} & \text{se q} \le \varepsilon, \\ \arg\max_{at}\left[Q_t(s_t,a_t) + H_t(s_t,a_t)\right] & \text{caso contrário.} \end{cases} \quad (3)$$

As únicas modificações realizadas no algoritmo HAQL em comparação ao *Q-Learning* é a modificação ao uso da função heurística para a escolha da ação a ser executada e a existência de um passo para atualização da função $H_t(s_t,a_t)$. Mais detalhes podem ser encontradas em (Bianchi, 2004).

4 O Experimento Proposto

Neste artigo são apresentados dois experimentos: o primeiro com o uso de um goleiro com aprendizado que irá jogar contra um atacante sem aprendizado e um segundo experimento que consta da situação inversa, um atacante com aprendizado que jogará contra um goleiro sem aprendizado. Os adversário são formados por jogadores do UVA Trilearn Basic (Kok; Vlassis; Groen, 2003), com funções básicas de chutar a bola. A escolha dos reforços e a quantidade de reforços necessários foram determinadas por meio de testes empíricos.

Para a situação do goleiro, foi implementada uma grade de 4 x 4 células que será a área de defesa do goleiro (Figura 3), esta área compreende a grande e pequena área do jogo. O agente sabe a todo tempo sua posição e também a localização da bola dentro de um plano cartesiano, podendo estimar em que célula o agente e a bola se encontram, quando ela estiver dentro da área de defesa do agente.

O goleiro tem como possibilidade executar as ações: ficar parado, interceptar a bola, conduzir a bola, pegar a bola e ficar em uma posição de defesa, na linha de trajeto da bola. As recompensas que o goleiro pode receber são: chutar, pegar ou conduzir a bola = 25; gol tomado= -100; bola com o oponente = -25.

Para a heurística do goleiro foi estipulada a seguinte regra: se o agente goleiro e a bola estiverem em células diferentes, então o agente deve correr em direção a bola, caso o agente goleiro e a bola estiverem na mesma célula, o agente deve escolher alguma ação que trabalhe com a bola.

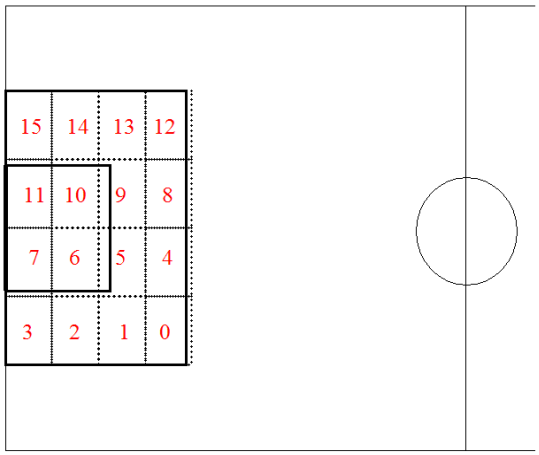

Figura 3. Grade utilizada pelo agente goleiro.

De forma similar, para o atacante, é criado uma grade que representará a área em que este irá atacar. Esta área é formada por uma grade de 8 x 7 células, mostrada na Figura 4 e compreende praticamente todo o campo adversário, as células próximas a área do gol adversário possuem tamanho reduzido, proporcionando um aprendizado mais detalhado nesta zona. As ações que podem ser realizadas pelo agente são: parado, interceptar a bola, chutar para o gol, chutar a bola para dentro da grande área do adversário, driblar e por último segurar a bola. As suas recompensas são: chutar a bola para frente e chutar a bola para o gol = 3; fazer gol = 100; bola com o oponente = -4.

Para a heurística do atacante foi estipulada a seguinte regra: se o agente atacante e a bola estiverem em células diferentes, então o agente deve correr em direção da bola, caso o agente atacante e a bola estiverem na mesma célula, o agente deve escolher alguma ação que trabalhe com a bola.

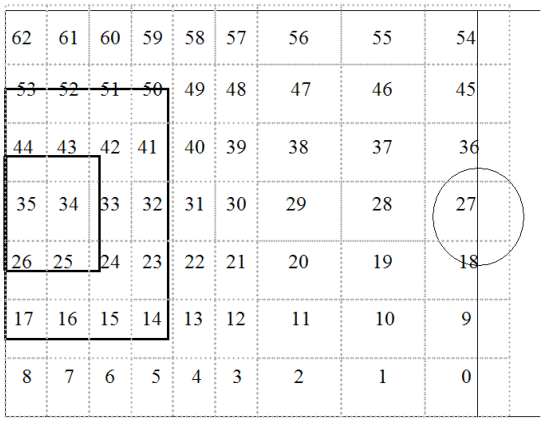

Figura 4. Grade utilizada pelo agente atacante.

5 Resultados Obtidos

Os agentes com aprendizados, podem escolher qualquer ação das disponíveis para serem usadas, ao escolher uma ação, está irá determinar o que o agente tem que fazer e assim ele irá receber uma recopensa por ter realizado esta ação. Cada episódio foi formado de 3.000 ciclos (aproximadamente 5 minutos) com todos os jogos dos agentes aprendizes começando do lado esquerdo do campo.

Para o goleiro o resultado da execução dos algoritmos *Q-Learning*, HAQL (média de 10 jogos) e de somente as heurísticas é apresentado na Figura 5. Na Figura 6 é apresentada uma comparação estatística do aprendizado com e sem o uso de heurísticas, usando o teste t de Student (Spiegel, 1984).

Pode-se perceber que utilizando o algoritmo *Q-Learning*, o goleiro inicia o aprendizado sofrendo uma média de 13 gols por jogo e no episódio número 100 este valor cai para aproximadamente 8 gols por jogo. Comparando com o algoritmo HAQL, pode-se ver que o goleiro já começa sofrendo um valor menor de gols (10 gols por jogo, em média) e que no episódio número 40 este valor fica em aproximadamente 7. Também se pode observar que o uso das heurísticas em um agente sem aprendizado não evolui com o tempo, e que não produz um goleiro eficiente, sofrendo uma média de 21 gols por partida (valor da média da partida). Com isso, pode-se concluir que a heurística acelera o aprendizado, mas não é uma informação tão forte que elimina a necessidade do *Q-Learning*.

No teste t de Student é possível ver que entre o décimo e o centésimo episódio, os algoritmos são significativamente diferentes, com nível de confiança maior que 5%. Isto confirma que a heurística usada torna o algoritmo HAQL mais rápido que o *Q-Learning*.

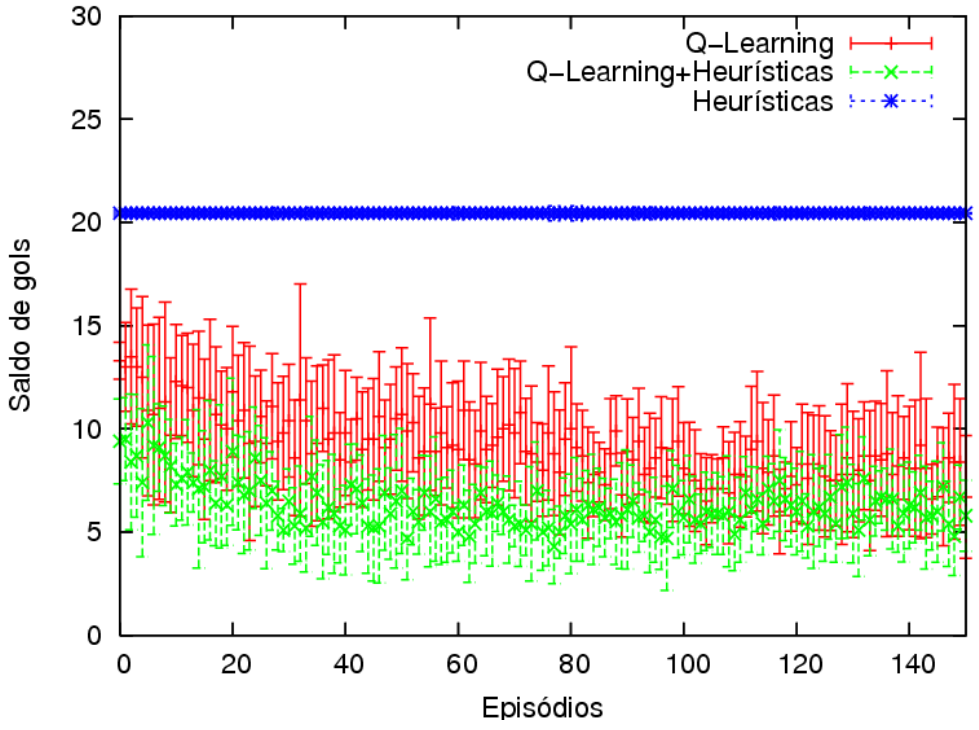

Figura 5. Evolução do saldo de gols para os algoritmos *Q-Learning*, HAQL e somente heurísticas para o agente goleiro com barras de erro.

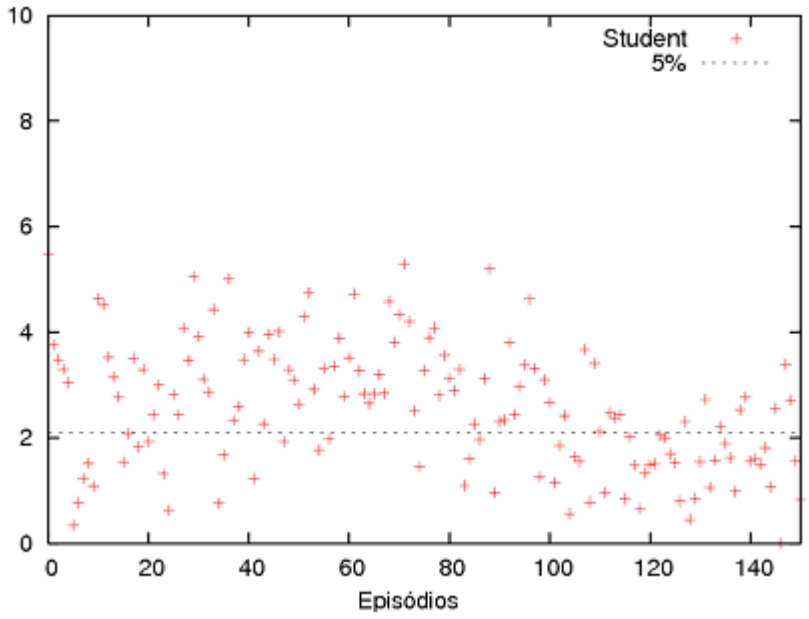

Figura 6. Resultado do teste *t* de Student para os algoritmos *Q-Learning* e HAQL.

Para o aprendizado do atacante, os resultados do uso do *Q-Learning* e do HAQL são mostrados na Figura 7. O uso de somente o *Q-Learning*, o atacante começa fazendo uma média de 7 gols e no episódio número 100 este valor fica em aproximadamente 18 gols. Comparando com o algoritmo *HAQL*, pode-se ver que o atacante começa com um valor de aproximadamente 12 gols e no final faz um média de 18 gols no goleiro do UVA trilearn. Também se pode observar que o uso de somente as heurísticas em um agente não evolui com o tempo, e que não produz um atacante eficiente, não fazendo praticamente gols. Com isso, pode-se concluir, como no caso anterior, que a heurística acelera o aprendizado, mas não é uma informação tão forte que elimina a necessidade da existência do *Q-Learning*.

O resultado para o teste *t* de Student é apresentado na Figura 8. Nesta figura é possível ver que o começo do aprendizado ocorreu mais rápido, porém depois do episódio número 30 a diferença dos gráficos não são grandes. Para este caso, o inicio do gráfico mostra que os algoritmos são significativamente diferentes, com nível de confiança maior que 5% até o episódio número 18. Isto confirma que a heurística usada torna o algoritmo HAQL mais rápido que o *Q-Learning*.

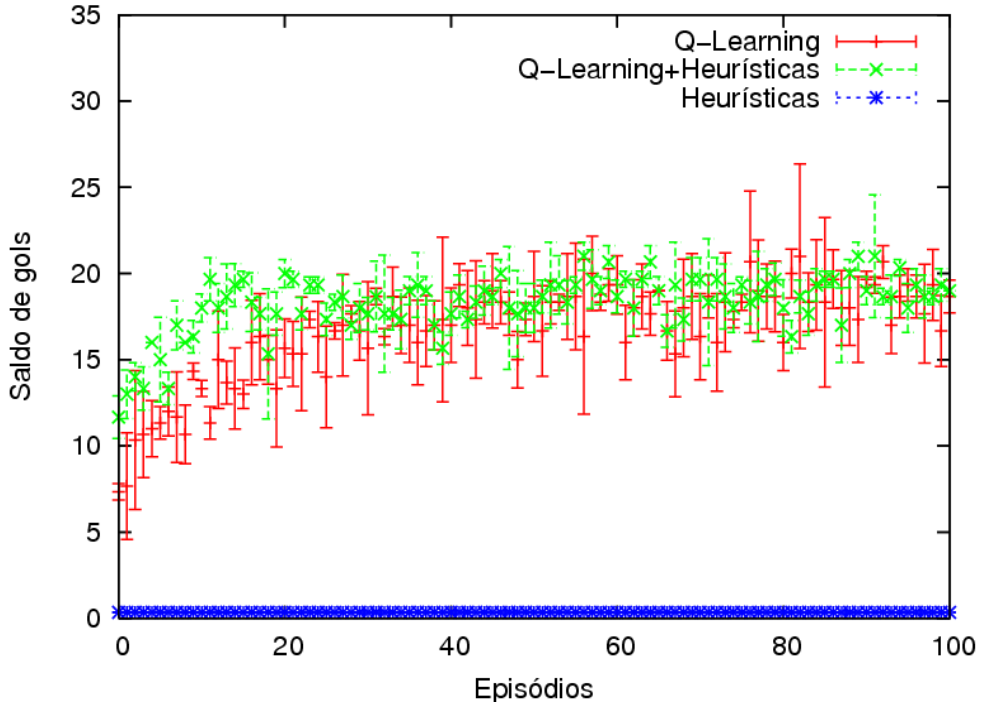

Figura 7. Evolução do saldo de gols para os algoritmos *Q-Learning*, HAQL e somente heurísticas para o atacante com barras de erro.

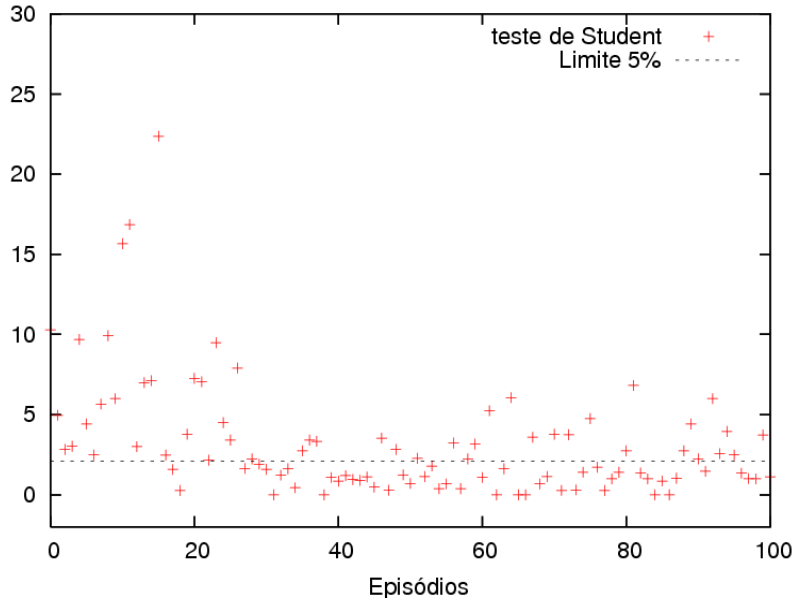

Figura 8. Resultado do teste *t* de Student para os algoritmos *Q-Learning* e HAQL.

Todos os experimentos foram realizados utilizando o algoritmo *Q-Learning*, com os seguintes parâmetros: γ = 0.9, α = 0.125, e taxa de exploração/explotação igual a 0.05. Os experimentos realizados foram codificados em linguagem C++ e executados em um Microcomputador Pentium 4 2.8 Ghz HT, com 1GB de memória RAM e sistema operacional Linux.

6 Conclusão

Este artigo propôs o uso da heurística na aceleração do aprendizado por reforço no domínio do futebol de robôs, usando a plataforma do *RoboCup* para a simulação dos testes.

Os resultados obtidos neste domínio mostram que usando o HAQL os agentes aprendem mais rápido que usando apenas o *Q-Learning*, quando treinados contra o mesmo oponente. Devido ao menor espaço de busca explorado pelo agente aprendiz, proporcionado por uma heurística correta o agente não precisava ficar passando por todos os estados para aprender o que fazer e assim é possível obter um aprendizado com uma performance melhor, usando apenas heurísticas simples.

Outro ponto importante é que por intermédio do aprendizado acelerado por heuristícas é possível obter jogadores que melhoram suas abilidades durante um jogo, mais rapidamente, que mesmos jogadores com aprendizado sem heurísticas.

É possível também afirmar que este jogadores com apredizado acelerado por heurísticas, podem também aprender a se adaptar ao jogo, quando seus oponenstes mudam de estratégia, independentemente das heurísticas utilizada no começo do jogo.

Referências Bibliográficas

Bianchi, R. A. C. Uso de Heurística para a aceleração do aprendizado por reforço. Tese (Doutorado) — Escola Politécnica da Universidade de São Paulo, 2004.

Boer; Kok, The Incremental Development of a Synthetic Multi-Agente System: The UvA Trilearn 2001 RoboCup Soccer Simulation Team, Tese (Mestrado) , 2002.

Kaelbling, L.P.; Littman M. L.; Moore, W.A. Reinforcement Learning: A Survey, Journal of Artificial Intelligence Research 4, May 1996, p. 237-285

Kitano, H.; Tambe, M.; Stone, P.; Veloso, M.; Noda, I.; OSAWA, E.; ASADA, M., The RoboCup Synthetic Agents' Challenge. Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 1995.

Kok, J.;Vlassis, N.; Groen, F. UvA Trilearn 2003 Team Description. In Proceedings CD RoboCup 2003, Springer-Verlag, Padua, Italy, July 2003.

Mitchell, T. Machine Learning. New York: McGraw Hill, 1997.

Pearl, J., Heuristics - Intelligent Search Strategies for Computer Problem Solving, Addison-Wesley Publishing Company, 1984.

Pegoraro, R. EPUSP/PCS. Agilizando Aprendizagem por Reforço em Robótica Móvel Através do Uso de Conhecimento Sobre o Domínio. Tese (Doutorado) , Setembro 2001.

Spiegel, M. R. Estatística. 2. ed. São Paulo: McGraw-Hill, 1984.

Uther, W; Veloso, M. Adversial Reinforcement Learning, Carnegie Mellon University, 2003

Watkins, C. J. C. H. Learning from Delayed Rewards. Tese (Doutorado) — University of Cambridge, 1989.

Watkins, C. J. C. H.; DAYAN, P. Q-learning. Machine Learning, v. 8, p.279–292, 1992.