5
CONCLUSÕES

Este capítulo apresenta as conclusões deste trabalho e uma síntese de sugestões para possíveis trabalhos futuros com base no que foi detalhado nos itens anteriores.

5.1
Conclusões

O objetivo deste trabalho foi desenvolver novas técnicas que contribuíssem para que o modelo *Reinforcement Learning Neuro-Fuzzy Hierarchical Politree* (RL-NFHP) aprendesse mais rapidamente, obtendo uma ferramenta mais eficiente para o aprendizado de agentes inteligentes. A utilização deste modelo em ambientes grandes e/ou contínuos constitui uma grande vantagem sobre os demais, uma vez que é desnecessário o conhecimento da dinâmica do ambiente, sendo preciso apenas um mínimo de definições prévias.

A melhoria do modelo, dito modificado, foi viabilizada através de um estudo inicial de seu comportamento com relação a seus parâmetros intrínsecos, paralelamente ao desenvolvimento de métodos de aceleração do aprendizado. Ressalta-se que a reescrita do código computacional também trouxe maior rapidez de processamento e maior generalização.

As variações do parâmetro ε de probabilidade de escolha aleatória para punição, assim como do parâmetro n da função de crescimento Φ (eq. 3.19) utilizada no particionamento da célula, não acarretaram mudanças significativas no desempenho de treinamento.

A alteração do valor do parâmetro α da equação de atualização da função de valor Q do par estado-ação para valores fixos também não trouxe ganho em relação ao número de épocas necessário para treinamento. Desta forma é recomendado continuar com o parâmetro α proporcional ao reforço da polipartição.

O uso da equação de punição na atualização da função valor Q acarretou em treinamentos mais rápidos quando se utilizou a política *Q-DC-roulette*.

O método de poda da estrutura durante o processo de aprendizado resultou em um maior número de épocas para treinamento e gerou, como esperado, favoravelmente uma estrutura final do modelo com menor número de células. A poda das células recentemente criadas e não visitadas gera uma dificuldade extra, além dos critérios de particionamento, para o crescimento da estrutura. Isto evita o refinamento demasiado do espaço de estados – crescimento da estrutura –, o que encareceria os cálculos computacionais, como ocorre com frequência não desprezível quando não se utiliza este método.

A proposta de *early stopping* para interrupção do aprendizado revelou-se de suma importância para a diminuição do número de épocas necessárias de treinamento.

O método de *eligibility trace* implementado neste trabalho é cumulativo, o qual desconsidera que o estado anterior possa ter sido particionado e que a ação possa ter sido escolhida de maneira aleatória. A utilização deste *eligibility trace* cumulativo gerou aprendizados com maior número de épocas. Consequentemente, recomenda-se a não utilização deste *eligibility trace* cumulativo durante o treinamento. Outros tipos de *eligibility trace* podem trazer desempenho superior ao apresentado.

A proposta de uma nova política de escolha de ação – *Q-DC-roulette* –, baseada em uma roleta mista entre visita e função valor Q, obteve excelente desempenho.

Pode-se concluir que o modelo RL-NFHP demonstrou ser capaz de aprender o comportamento desejado, mesmo quando submetido a um problema grande e contínuo de várias entradas (neste trabalho, 5 entradas no caso do Khepera), o que torna ainda mais complexa a capacidade de aprendizado, devido ao alto número de consequentes.

Como esperado, o número de regras linguísticas geradas aumenta com a complexidade do problema. Este fato pode ser observado para o caso Khepera: no caso sem obstáculo foi construída uma estrutura menor do que com obstáculos. Isto se deve à maior complexidade nas posições próximas ao obstáculo. Pelo mesmo motivo, o número de ciclos necessários para o aprendizado, no caso sem obstáculo, também foi menor.

A normalização das entradas também permitiu que o modelo respondesse adequadamente a mudanças nos limites das variáveis de entrada. Esta

característica também foi bastante interessante, pois aumentou ainda mais a autonomia desejada para este agente.

Pode-se concluir que o modelo RL-NFHP demonstrou ser capaz de aprender e generalizar o comportamento desejado no simulador e, com algumas adaptações, ser utilizado para equipar um agente real dotando-o de comportamento inteligente. O uso de simuladores realistas evita o dispêndio de tempo de treinamento. Vale ressaltar que o experimento conduzido com o robô Lego *MindStorms* NXT gerou desempenho acima do esperado, uma vez que o aprendizado do par estado-ação foi feito apenas no simulador e o modelo já pronto foi utilizado em um robô e ambiente ligeiramente diferentes.

Entretanto alguns aspectos menos favoráveis deste modelo merecem menção. O modelo tem dificuldade para solucionar adequadamente problemas que exijam saídas discretas, devido à generalização inerente ao particionamento fuzzy do espaço de entrada. O comportamento aprendido pelo agente deve responder adequadamente a um domínio que corresponde a um número significativo de estados do ambiente. Dessa forma, o agente não consegue ter um comportamento discreto. Duas formas distintas foram adotadas para resolver este problema no estudo de caso do carro da montanha: utilizar um conjunto de ações (associadas às polipartições) com valores absolutos maiores do que os usados na solução do problema, porém não permitindo que fosse aplicada, ao mesmo, uma ação superior à permitida; ou usar um conjunto de ações (associadas às polipartições) com valores normalmente utilizados na solução do problema, porém acoplando uma função à saída da estrutura, de forma a aumentar, dentro dos limites permitidos, o valor de saída.

Outro aspecto menos favorável é o fato de que a função de reforço deve ser mais elaborada do que as funções requeridas por outros modelos NF baseados em RL, pois, sem esta informação, a solução seria extremamente difícil de ser encontrada, uma vez que o sistema necessita, paralelamente ao aprendizado do comportamento do agente, aprender a identificar seus domínios.

Por fim, a estrutura do RL-NFHP pode alcançar tamanhos maiores do que o necessário devido às dificuldades de aprendizado inseridas pelo próprio RL somadas às do modelo RL-NFHP, causando maior dificuldade do aprendizado, aumentando o tempo computacional e gerando regras mais especializadas do que o necessário, ou seja, uma estrutura com mais células.

Em resumo, estas propostas de melhoria e aceleração de aprendizado, em relação ao modelo original hierárquico neuro-fuzzy RL-NFHP, permitiram: manter a capacidade do modelo de criar e expandir a estrutura de regras sem qualquer conhecimento a priori (regras ou conjuntos fuzzy); extrair conhecimento a partir da interação direta do agente com ambientes grandes e/ou contínuos, utilizando aprendizado por reforço, de modo a aprender quais ações devem ser executadas; e produzir resultados linguisticamente interpretáveis, sob a forma de regras fuzzy, que constituam o raciocínio que o agente deve inferir para atingir suas metas.

Estes estudos de casos mostram a boa aplicabilidade do modelo RL-NFHP na área de controle e robótica, encorajando o prosseguimento da pesquisa de aprendizado automático utilizando Sistemas Neuro-Fuzzy Hierárquicos.

5.2
Sugestões para Trabalhos Futuros

As sugestões abaixo elencadas objetivam nortear futuras abordagens no modelo RL-NFHP. As principais linhas de pesquisa para a continuação deste trabalho são:

- Continuar os estudos do modelo RL-NFHP modificado em outros casos *benchmark* para corroborar a melhoria e a aceleração do treinamento através da inserção das modificações propostas neste trabalho.
- Melhorar o crescimento da estrutura do modelo RL-NFHP, modificando os critérios de particionamento das células. A função de crescimento (eq. 2.19) deve ser repensada.
- Tornar os poucos parâmetros de ajuste do modelo RL-NFHP automáticos. Variando de acordo com a quantidade de entradas e dificuldade do problema.
- Estudar a influência na aceleração do aprendizado ao se usar a política de escolha de ação desenvolvida neste trabalho, *Q-DC-roulette* em outros modelos baseados em *Reinforcement Learning*.

- Adequar este modelo de forma a possibilitar que o mesmo seja acoplado como módulos de inteligência em sistemas como, por exemplo, o *Java Application Framework for Intelligent and Mobile Agents* (JAFIMA) (Kendall et al, 1997), que já possui várias características de agentes, tais como mobilidade, comunicação e capacidade de instanciar vários agentes.
- Adequar o modelo para aplicação em jogos, por meio do uso do modelo já integrado ao JAFIMA, por exemplo. Isso permitiria novas avaliações considerando ou não a troca de conhecimento adquirido pelos agentes rivais.
- Adequar o modelo para aplicações onde houvesse necessidade de cooperação entre os agentes. Isso permitiria novas avaliações, considerando ou não a troca de conhecimento adquirido pelos agentes parceiros, através de comunicação implícita ou explícita entre os agentes.
- Uma nova modelagem pode ser feita a partir do RL-NFHP: um modelo *Multi-Tree*. Neste novo modelo, cada variável de entrada teria uma estrutura própria, sem compartilhar a célula ou a estrutura com outra variável. Isso deve levar à construção de árvores menores. O reforço do ambiente deve ser tratado separadamente, um para cada variável de estado, permitindo que as ações sejam avaliadas em relação a sua contribuição. A saída de cada estrutura é ponderada percentualmente pelo valor da entrada em relação ao valor máximo permitido para a mesma.