1

Introdução

1.1

Motivação

Convencionalmente, as ações executadas por computadores devem ser antecipadas, planejadas e codificadas pelos programadores. Caso ocorra uma situação não prevista, estará configurado um erro que deverá ser corrigido pelo programador. Para muitas aplicações isto pode ser aceitável. No entanto, para um número cada vez maior de aplicações há a necessidade de que os sistemas tenham maior grau de autonomia, isto é, que incorporem a capacidade de decidir o que deve ser feito para satisfazer os objetivos determinados pelo programador (Weiss, 1999).

Agentes são o paradigma utilizado no desenvolvimento de aplicações de software que visam a atender a esta necessidade, pois percebem o ambiente através de sensores e agem neste ambiente através de atuadores (Jenning e Wooldridge, 1997; Russel e Norvig, 2010).

Entre as características que um agente pode possuir, a autonomia é aquela que determina que programas (códigos ou algoritmos) podem ser classificados como agentes. A autonomia é a capacidade de o agente seguir seus objetivos automaticamente, isto é, sem comandos que venham do ambiente. Outra característica de grande importância, por aumentar a autonomia de um agente, é a inteligência. Brenner et. al. (1998) são enfáticos ao afirmar que o agente deve ser capaz de aprender para ser considerado autônomo. O aprendizado, juntamente com a base de conhecimento e o raciocínio, integram a característica inteligência.

Um agente inteligente deve interagir com o ambiente para atingir seus objetivos: deve ser capaz de reunir informações do seu ambiente, tomar decisões baseadas nestas informações e iniciar uma ação específica baseada nessas decisões.

Recentemente, um novo campo de pesquisa na área de Inteligência Computacional vem emergindo, com o objetivo de desenvolver sistemas complexos envolvendo *múltiplos agentes* e os mecanismos de coordenação entre eles: os chamados Sistemas Multi-Agentes (SMA). Um SMA pode ser definido como um grupo de agentes autônomos, interagindo entre si e compartilhando um mesmo ambiente, que é percebido através de sensores, e onde eles agem realizando ações (Vlassis, 2003). Este tipo de sistema vem encontrando uma grande variedade de campos de aplicação, como controle de processos (Stephan et al., 2000), controle de sinais de trânsito (Wiering, 2000; Bakker et al., 2005), controle de redes de energia elétrica (Riedmiller et al., 2000), times robóticos (Ishiwaka et al., 2003; Kok et al., 2005; Xiao e Tan, 2007; Hladek, 2007), controle de navegação (Mataric, 1997; Bowling e Veloso, 2002; Hu e Wellman, 2003; Calvo e Romero, 2007), definição de estratégias de negociação e *trading* (Lee e Oo, 2002; Oo et al., 2002; Lee et al., 2007; Wellman et al., 2003), etc.

Os benefícios trazidos pela aplicação de múltiplos agentes são diversos. Em primeiro lugar, através da computação paralela, vários agentes podem trabalhar em conjunto para melhor explorar a estrutura descentralizada de uma determinada tarefa e acelerar sua conclusão (Lagoudakis, 2002; Kok et al., 2005; Fitch et al., 2005). Além disso, agentes podem trocar experiências se comunicando (Tan, 1993), podem apenas observar os mais habilidosos e aprender (Price e Boutilier, 2003), os mais inteligentes podem servir de professores para outros agentes (Clouse, 1995), etc. Os SMA também podem fornecer alto grau de escalabilidade, através da inclusão de novos agentes quando necessário, e ainda fazer com que agentes assumam as atividades de outros agentes em casos de falha (Busoniu, 2008). Os SMA também se destacam no estudo e compreensão da inteligência (Weiss e Sen, 1996). Como a inteligência está profundamente acoplada à interação, a melhor forma de criar máquinas inteligentes pode ser através da construção de “redes sociais” de máquinas.

Vários modelos de agentes desenvolvidos até o momento usam o aprendizado por reforço (*Reinforcement Learning*) como algoritmo base no processo de aprendizado. Quando o agente está inserido em ambientes pequenos e/ou discretos, os resultados obtidos com o uso de métodos como *Q-learning* são satisfatórios (Weiss, 1993; Sen et. al., 1994; Sen e Sekaram, 1995; Claus e Boutilier, 1998; Bowling e Veloso, 2002; Melo e Ribeiro, 2008).

No entanto, quando o ambiente é grande e/ou contínuo, a aplicação de métodos de *Reinforcement Learning* (RL) utilizando *lookup table* (tabela de armazenamento das funções de valor para espaço de estados pequeno ou discreto) torna-se inviável, devido à grande dimensão do espaço de estados. Este problema é conhecido como *curse of dimensionality* (Kaelbling et. al. 1996; Sutton e Barto, 1998, Ribeiro, 1999).

Nos SMA, este problema é consideravelmente maior, já que o crescimento da memória poderá ser exponencial em relação à quantidade de agentes envolvidos na aplicação (as funções de valor devem representar uma combinação dos estados e ações de todos os agentes em conjunto, e não apenas de um agente).

Para contornar este problema, alguma forma de generalização deve ser incorporada à representação de estados. A generalização consiste em produzir modificações nos métodos RL por intermédio de aproximação de funções, permitindo a atualização através de reforços não apenas do estado relacionado à iteração atual, mas também de outros estados correlacionados por alguma característica comum. A aproximação de funções pode ser feita por diversas formas (Moore, 1991; Kaelbling et. al., 1996; Ribeiro, 1999); atualmente, empregam-se técnicas inteligentes, como Redes Neurais e Lógica Fuzzy (Sutton, 1996; Jouffe, 1998; Stephan et al, 2000; Xiao e Tan, 2007).

Alguns dos modelos que utilizam aproximação de funções, necessitam de um grande número de informações ou de definições prévias. De uma forma geral, as limitações dos principais modelos baseados em RL e Lógica Fuzzy, como por exemplo, NEFCON (Neuro-Fuzzy Controller) (Nauck e Kruse, 1997) e FQL (Fuzzy-Q-Learning) (GLORENNEC, 1994) são:

- necessidade de pré-definição do número de conjuntos fuzzy e das funções de pertinência destes conjuntos, usados nos antecedentes e conseqüentes das regras fuzzy;
- necessidade de pré-definição do número de regras e até mesmo dos antecedentes que compõem as regras;
- limitação quanto ao número de variáveis de entrada no modelo.

Com o objetivo de superar algumas das limitações apresentadas, em Figueiredo (2003), foram desenvolvidos os modelos de *Reinforcement Learning*

Neuro-Fuzzy Hierárquicos (RL-NFH), contemplando o aprendizado autônomo de um único agente. Originariamente, Souza et al. (2000) criou esta família de Sistemas Neuro-Fuzzy, denominados Sistemas Neuro-Fuzzy Hierárquicos, para realizar aprendizado com treinamento supervisionado baseado em Gradiente Decrescente.

Após a avaliação de diversos modelos durante a etapa de levantamento bibliográfico, optou-se pela utilização desta família de sistemas como base para o desenvolvimento de novos Sistemas Multi-Agente.

1.2

Descrição do trabalho e contribuições

A proposta central deste trabalho foi o desenvolvimento de um novo modelo considerando múltiplos agentes, colaborativos ou competitivos, através de modelos neuro-fuzzy híbridos da família RL-NFH. Entre as principais contribuições do novo modelo estão:

- criação e expansão da estrutura de regras dos múltiplos agentes inteligentes, sem o conhecimento prévio do número de regras;
- extração de conhecimento a partir da interação direta de multi-agentes com o ambiente, sem utilizar algoritmos supervisionados, de forma a aprender quais ações devem ser executadas em ambientes grandes e/ou contínuos;
- produção de resultados linguisticamente interpretáveis, sob a forma de regras fuzzy, que constituam o raciocínio que os múltiplos agentes devem construir para atingir suas metas;
- capacidade de realizar o aprendizado eficiente de múltiplos agentes, fazendo com que estes possam explorar o espaço de estados-ações de forma simultânea;

- permitir diferentes formas de coordenação entre os agentes, mais especificamente de forma explícita, através da existência de um agente central, ou implícita;

- viabilizar o desenvolvimento de aplicações em que múltiplos agentes atuem tanto de forma cooperativa, como competitiva;

- permitir que agentes trabalhem com bases de conhecimento compartilhadas, com o objetivo de agilizar o aprendizado de determinadas tarefas, ou independentes.

Além disso, também podem ser destacadas as seguintes contribuições secundárias geradas por este trabalho :

- 1- Alteração da versão atual do modelo RL-NFHP, introduzindo o princípio de *satisfatoriedade* (Simon, 1996; Goodrich e Quigley, 2004), visando acelerar o algoritmo de aprendizado.

- 2- Desenvolvimento de um ambiente de testes para os sistemas multi-agentes RL-NFH. Foi criado um ambiente para desenvolvimento e testes, em linguagem Java, com um compilador NetBeans, para Windows.

- 3- Desenvolvimento de um estudo de caso da aplicação *benchmark* do Jogo da Presa/Predador (Benda et al., 1986). O desenvolvimento deste estudo de caso permitirá uma análise dos ganhos obtidos, tanto pela introdução do conceito de *satisfatoriedade*, como pela utilização de múltiplos agentes no processo de aprendizado. Foram avaliadas a velocidade de convergência para uma política ótima, a quantidade de passos necessárias para a captura da presa, e outras diferenças entre o novo modelo e o modelo anterior. Neste jogo, os Predadores se comportam como agentes inteligentes que cooperam entre si, com a finalidade de capturar a Presa. Esta, por sua vez, se movimenta de duas formas: de maneira aleatória ou também como um agente inteligente, permitindo assim a avaliação de aspectos envolvendo a competição entre agentes.

4- Desenvolvimento de um SMA de natureza competitiva para simular um leilão de energia, em que os fornecedores de energia fazem ofertas para atender uma determinada previsão de demanda em um período de tempo. Existem vários tipos de leilões. Neste trabalho, foi simulado um leilão reverso, onde o objetivo é comprar energia dos agentes que a oferecem pelo menor preço, diferentemente do leilão inglês, em que o produto é vendido diretamente para quem oferece o maior lance. Além disso, o leilão simulado teve como características o preço fechado (*sealed bid auction*), não sendo os lances de conhecimento público, e o preço uniforme, ou seja, os lances aceitos são pagos ao preço do último e maior lance aceito. O objetivo principal do leilão é atender à demanda de curto prazo. Os leilões de curto prazo são realizados no dia anterior ao consumo (*day ahead market*) e estabelecem os preços para a demanda de 24 períodos de 1 hora ou 48 períodos de meia. Nesta aplicação, agentes ou fornecedores de energia desenvolveram sua própria estratégia, utilizando uma estrutura RL-NFHP, visando sempre maximizar o lucro. Os agentes RL-NFHP inteligentes desenvolvidos competem nos leilões entre eles e também com agentes que utilizam estratégias conservadoras.

5- Desenvolvimento de um SMA voltado para o Gerenciamento Ágil de Projetos de Softwares. Este terceiro SMA teve como objetivo a criação de agentes inteligentes colaborativos para dar suporte no processo de definição de métricas e estimativas de duração, e automatizar alguns dos processos realizados pelo Gerente de Projetos, como a alocação de recursos, planejamento e orçamentação.

De maneira resumida, o quadro abaixo relaciona os diferentes Estudos de Caso descritos nos itens acima e desenvolvidos através do novo modelo multi-agente elaborado nesta tese (RL-NFHP-MA), com as possíveis formas de coordenação e interação entre os agentes.

Tabela 1: Quadro relacionando os diferentes modelos RL-NFHP-MA com os Estudos de Caso que foram desenvolvidos.

Aprendizado	Coordenação	Tipo de Interação	Objetivo	Aplicação
Estrutura de conhecimento compartilhada	Implícita	Cooperação	Exploração paralela do ambiente	Presas x Predador (Cooperação entre Predadores)
	Explícita (com Agente Central)			
Estruturas de aprendizado independentes	Coordenação Individual	Cooperação	Especialização em tarefas distintas	Gerenciamento de Projetos
		Competição		Leilão de Energia
				Presas x Predador (Presas Inteligentes)

No intuito de alcançar estes objetivos, foram realizadas três macro etapas:

- 1- Levantamento bibliográfico e estudo de agentes inteligentes e Sistemas Multi-Agentes, buscando identificar as propriedades e limitações dos algoritmos já desenvolvidos, as aplicações existentes, e as características desejadas em um SMA.
- 2- Desenvolvimento um novo modelo de SMA, a partir de um modelo capaz de gerar sua própria estrutura e de aprender dinamicamente, viabilizando aplicações em ambientes com espaços de estados grandes e/ou contínuos
- 3- Desenvolvimento de um ambiente computacional adequado para a avaliação do modelo proposto, permitindo a realização de simulações para demonstrar a potencialidade e a aplicabilidade do modelo proposto.

1.3

Organização da Tese

O restante desta tese está organizado em mais cinco capítulos.

No capítulo 2, faz-se uma breve introdução à área de agentes e de modelos de aprendizado, destacando-se o aprendizado por reforço. Neste capítulo também são explorados os principais conceitos, aspectos teóricos, aplicações e desafios relacionados aos Sistemas Multi-Agentes.

O capítulo 3 apresenta o modelo denominado *Reinforcement Learning Neuro-Fuzzy Hierárquico Polítree* (RL-NFHP), apresentando também sua célula básica, sua arquitetura, as regras hierárquicas associadas e o seu algoritmo de aprendizado.

O capítulo 4 introduz o conceito de *satisfatoriedade* na versão atual do modelo RL-NFH e apresenta, de forma detalhada, a nova versão para multi-agentes dos modelos da família RL-NFH (RL-NFH-MA). São explorados aspectos como o aprendizado em diferentes tarefas de forma simultânea, o compartilhamento do conhecimento dos agentes em uma mesma estrutura, a utilização do modelo em ambientes competitivos e cooperativos, o ganho de velocidade na convergência para uma política ótima, as formas de coordenação entre os agentes, e a homogeneidade no aprendizado.

No capítulo 5, são apresentados os estudos de caso e os resultados das simulações obtidos através do SMA neuro-fuzzy hierárquico criado.

Finalmente, o capítulo 6 apresenta as conclusões finais e aponta possíveis trabalhos futuros que podem ser desenvolvidos a partir deste trabalho.