

5

Estudo de Casos

5.1

Introdução

Os estudos de casos apresentados neste capítulo foram desenvolvidos com o objetivo de contemplar as diferentes variações dos modelos RL-NFHP-MA descritos no capítulo anterior e ilustrados na Tabela 5.

Tabela 5: Quadro com uma síntese dos diferentes modelos RL-NFHP-MA

Aprendizado	Coordenação	Tipo de Interação	Objetivo
Estrutura de conhecimento compartilhada	Implícita	Cooperação	Exploração paralela do ambiente
	Explícita (com Agente Central)		
Estruturas de aprendizado independentes	Implícita	Cooperação	Especialização em tarefas distintas
	Explícita (com Agente Central)	Competição	

Primeiramente, foram desenvolvidos dois sistemas em que os agentes utilizam uma estrutura de conhecimento compartilhada, ambos com o objetivo de realizar uma exploração paralela do ambiente e cooperar na realização de uma determinada tarefa. Um dos sistemas foi criado sem um mecanismo central de coordenação, ou seja, os agentes decidiram, de forma individual e independente, que objetivos deveriam ser alcançados ao longo do processo, com base nas informações captadas do ambiente. Já, no segundo estudo, foi criado um mecanismo de coordenação para sincronizar as atividades dos agentes.

Outros dois sistemas foram desenvolvidos de forma que os agentes tivessem estruturas RL-NFHP independentes, representando seu conhecimento. Um dos sistemas teve como foco a aplicação de múltiplos agentes em um problema de natureza competitiva e o outro abordou um problema de natureza cooperativa.

A seção seguinte introduz os Estudos de Caso selecionados.

5.2

Seleção e Apresentação dos Estudo de Casos

Para contemplar as situações descritas na seção anterior e seus pré-requisitos, foram selecionadas três aplicações. A primeira delas é o Jogo da Presa e Predador, também conhecido como “*Pursuit Game*” (Benda et al., 1986). Através desta aplicação foram explorados sistemas em que os agentes possuem uma estrutura única de conhecimento. A aplicação pode ser considerada de natureza colaborativa, já que ocorre a interação entre os Predadores com o objetivo único de capturar a Presa. Foram feitas ainda simulações adicionais onde a Presa, além dos Predadores, se comporta como um agente RL-NFHP inteligente, com o intuito de dificultar a sua própria captura. Assim, a aplicação também ganhou características de competição.

No segundo estudo de caso, foi desenvolvido um leilão de energia como aplicação. Cada agente é um fornecedor de energia que faz ofertas inteligentes para atender a uma determinada previsão de demanda em um período de tempo. Os agentes atuam de forma competitiva e cada um busca aprender a melhor estratégia de *bidding*, de acordo com a demanda solicitada.

O terceiro e último estudo de caso teve como objetivo o desenvolvimento de um novo Sistema Multi-Agente voltado para o Gerenciamento Ágil de Projetos de Softwares. Nesta aplicação, cada agente inteligente representa um grupo de profissionais com a mesma função. Os agentes criados colaboram para dar suporte no processo de definição de métricas e estimativas de duração, e automatizar alguns processos realizados pelo Gerente de Projetos, como a alocação de recursos, planejamento e orçamentação.

O quadro abaixo apresenta novamente os diferentes modelos RL-NFHP-MA abordados nesta Tese e os relaciona com os Estudos de Caso que foram desenvolvidos com o objetivo de evidenciar o seu funcionamento.

Tabela 6: 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)

5.3

Desenvolvimento dos Estudos de Caso

5.3.1

Jogo da Presa e Predador

5.3.1.1

Metodologia

Neste primeiro estudo de caso, os sistemas RL-NFHP-MA para o problema da Presa / Predador foram construídos seguindo as especificações, premissas e restrições abaixo:

- espaço composto por um grid ortogonal de 9x9 posições;
- os 4 agentes predadores se movem usando o RL-NFHP multi-agente, com o objetivo de capturar a presa;
- a presa e os predadores só podem se movimentar em 4 direções. Os movimentos diagonais não são permitidos;
- cada rodada do jogo é composta de uma movimentação de cada participante;
- não há o conceito de aceleração. Cada movimentação é sempre feita para uma posição imediatamente acima, abaixo, à esquerda e à direita;

- os participantes se movem de forma alternada, e não simultânea;
- cada agente predador consegue visualizar a posição dos demais agentes e a da presa;
- os agentes predadores não conhecem os objetivos dos demais agentes;
- os agentes podem compartilhar o ‘conhecimento’ obtido ao longo do aprendizado;
- em caso de colisão após uma movimentação, o agente ou a presa retornam para a posição em que estavam;
- em caso de um participante tentar se mover para fora do grid, ele também retorna para a posição em que estava.

Quanto à movimentação da presa, conforme já mencionado na seção anterior, inicialmente, ela se movimentou de forma aleatória a cada rodada. Em seguida, uma variação do SMA foi desenvolvida e, na segunda configuração do sistema, a presa se tornou um agente inteligente, assim como os predadores, e buscou agir sempre com o objetivo de fugir e dificultar a sua captura.

Outra característica que variou foi a coordenação entre os agentes predadores. Em um primeiro momento, não houve um agente centralizador para realizar a coordenação entre os agentes. Posteriormente, um mecanismo de coordenação foi introduzido com o objetivo de controlar os objetivos de chegada dos agentes em relação à presa.

Dadas as características acima, pode-se dizer que o problema consiste em uma aplicação SMA com uma dinâmica de aprendizado homogêneo, já que todos os agentes utilizaram um mesmo algoritmo de aprendizado (inclusive a presa quando atuar como um agente inteligente).

Quanto à comunicação, ela ocorreu de forma indireta. Ao usarem a mesma estrutura RL-NFHP-MA, de certa forma, os agentes compartilharam informações que aprenderam. A estrutura funciona como um repositório de conhecimento que vai se tornando mais rico, conforme os agentes se movimentam e a retro-alimentam. É através deste repositório que os agentes sinalizam para os demais que “já estiveram em um determinado estado, experimentaram uma determinada ação, e o resultado foi x”. Porém, vale sempre lembrar que os agentes decidem de forma independente onde devem chegar para capturar a presa (acima, abaixo, direita, esquerda), pois os objetivos não são “públicos”. As informações

compartilhadas na árvore representam apenas um mapeamento dos estados e ações do problema.

O estado foi definido com uma única variável: a posição relativa da presa em relação ao agente, representada pelo ângulo arco-tangente formado pela presa, considerando que o agente está sempre no eixo, ou na posição (0, 0). O valor do ângulo, que variava de 0° a 360°, foi sempre normalizado entre 0 e 1. Para um melhor entendimento, a figura abaixo mostra oito possíveis valores do estado para o agente, quando este se encontra na posição (4, 4). Por exemplo, se o agente, representado na figura abaixo pelo X na célula cinza, está na posição (4, 4), e a presa na célula (7, 7), o ângulo arco-tangente formado é de 45°. Este valor é suficiente para determinar o estado.

8									
7		135			90			45	
6									
5									
4		180			X			0	
3									
2									
1		225			270			315	
0									
	0	1	2	3	4	5	6	7	8

Figura 33: Possíveis valores do estado para o agente, quando este se encontra na posição (4, 4)

No caso da presa estar na posição (1, 4) e o agente na posição (4, 4), o estado será o valor de 180°, normalizado de 0 a 1. Como a posição relativa é que está sendo considerada, se o agente estiver nas posições (3, 4), (4, 4), (5, 4), ..., (8, 4), o valor do estado será o mesmo.

O reforço recebido pelo modelo após cada ação foi calculado de maneira inversamente proporcional à distância do agente para a presa, conforme mostram as equações abaixo.

$$\text{Distância entre o agente e a presa} \rightarrow d = |(Ax - Px)| + |(Ay - Py)| \quad (33)$$

$$\text{Reforço} \rightarrow r = 1 - d_{\text{norm}} \quad (34)$$

A_x e P_x são as posições do agente e da presa no eixo x, e A_y e P_y são as posições do agente e da presa no eixo y, respectivamente; e d_{norm} é a distância entre o agente e a presa normalizada de forma linear de 0 a 1.

O processo de treinamento foi realizado com a presa sempre fixa na posição (4, 4). A posição de cada agente no início do jogo, ou após cada captura, também foi inicializada de maneira fixa, sempre nos cantos do grid – posições (0, 0), (8, 8), (8, 0), e (0, 8). Ou seja, os agentes precisavam sair dos cantos e capturar uma presa fixa exatamente do centro do grid.

Para que a captura pudesse ser realizada, a cada passo, os agentes definiram os “objetivos” que deveriam alcançar. Considerando a posição (x, y) da presa como (Px, Py), os quatros possíveis objetivos são: (Px - 1, Py), (Px + 1, Py), (Px, Py - 1) e (Px, Py + 1). Como cada agente conseguia visualizar tanto a posição da presa como dos demais agentes, o algoritmo de escolha de objetivo usado pelos agentes buscou encontrar sempre a formação em que a soma das distâncias entre os agentes e objetivos fosse a menor possível. Eventualmente, ocorria colisão de objetivos, já que um agente não sabe o objetivo dos demais. Esse tipo de colisão deve ser solucionado naturalmente ao longo da movimentação dos agentes, conforme eles se aproximem da presa e, conseqüentemente, de seus objetivos.

8	A3								A2
7									
6									
5					Obj3				
4				Obj1	Presa	Obj2			
3					Obj4				
2									
1									
0	A1								A4
	0	1	2	3	4	5	6	7	8

Figura 34: Quatro possíveis objetivos que os agentes buscavam alcançar para capturar a presa.

5.3.1.2

Resultados Iniciais RL-NFHP-MA

O treinamento foi realizado durante 100 passos. Os passos das 2 primeiras perseguições estão detalhados abaixo. Cada passo mostra a posição em que o agente se encontrava e seu objetivo naquele momento. Por exemplo, agente2 - 7,6 (4,5), significa que, naquele passo do treinamento, o agente 2 estava na posição (7, 6), com o objetivo de chegar na posição (4, 5) para capturar a presa. Em alguns casos, o detalhamento mostra que um determinado agente não se movimentou de um passo para outro. Isso ocorre quando ele tenta sair do grid, ou quando ele está ocupando a posição que é o seu objetivo.

- início treinamento

- Início 1ª perseguição

-Passo-1 -agente1- 1,0* (3,4)** -agente2- 8,7 (4,5) -agente3- 0,7 (4,5) -agente4- 7,0 (4,3)

* posição que o agente1 está; ** posição que ele pretende chegar neste momento da perseguição

-Passo-2 -agente1- 1,0 (3,4) -agente2- 8,6 (4,5) -agente3- 0,8 (4,5) -agente4- 7,1 (4,3)

-Passo-3 -agente1- 0,0 (3,4) -agente2- 7,6 (4,5) -agente3- 0,7 (4,5) -agente4- 7,0 (4,3)

-Passo-4 -agente1- 0,0 (3,4) -agente2- 6,6 (4,5) -agente3- 0,6 (4,5) -agente4- 6,0 (4,3)

-Passo-5 -agente1- 0,0 (3,4) -agente2- 5,6 (4,5) -agente3- 0,5 (4,5) -agente4- 5,0 (4,3)

-Passo-6 -agente1- 0,1 (4,5) -agente2- 5,5 (4,5) -agente3- 0,4 (4,5) -agente4- 4,0 (5,4)

-Passo-7 -agente1- 0,2 (4,5) -agente2- 5,4 (4,5) -agente3- 0,4 (4,5) -agente4- 5,0 (4,5)

-Passo-8 -agente1- 0,1 (3,4) -agente2- 5,3 (4,5) -agente3- 0,5 (4,5) -agente4- 5,1 (4,3)

-Passo-9 -agente1- 0,2 (3,4) -agente2- 5,4 (4,3) -agente3- 0,6 (4,5) -agente4- 5,2 (4,3)

-Passo-10 -agente1- 1,2 (3,4) -agente2- 5,4 (5,4) -agente3- 0,7 (4,5) -agente4- 5,3 (4,3)

-Passo-11 -agente1- 1,3 (3,4) -agente2- 5,4 (5,4) -agente3- 1,7 (4,5) -agente4- 4,3 (4,3)

-Passo-12 -agente1- 2,3 (3,4) -agente2- 5,4 (5,4) -agente3- 2,7 (4,5) -agente4- 4,3 (4,3)

-Passo-13 -agente1- 2,4 (3,4) -agente2- 5,4 (5,4) -agente3- 3,7 (4,5) -agente4- 4,3 (4,3)

-Passo-14 -agente1- 3,4 (3,4) -agente2- 5,4 (5,4) -agente3- 3,6 (4,5) -agente4- 4,3 (4,3)

-Passo-15 -agente1- 3,4 (3,4) -agente2- 5,4 (5,4) -agente3- 4,6 (4,5) -agente4- 4,3 (4,3)

-(Captura) Passo-16 -agente1- 3,4 (3,4) -agente2- 5,4 (5,4) -agente3- 4,5 (4,5) -agente4- 4,3 (4,3)

Fim 1ª perseguição

- Início 2ª perseguição

-Passo-17 -agente1- 0,1 (3,4) -agente2- 8,7 (4,5) -agente3- 1,8 (4,5) -agente4- 8,1 (4,3)

-Passo-18 -agente1- 0,2 (3,4) -agente2- 7,7 (4,5) -agente3- 2,8 (4,5) -agente4- 8,2 (4,3)
 -Passo-19 -agente1- 1,2 (3,4) -agente2- 6,7 (4,5) -agente3- 3,8 (4,5) -agente4- 7,2 (4,3)
 -Passo-20 -agente1- 1,3 (3,4) -agente2- 6,6 (4,5) -agente3- 3,7 (4,5) -agente4- 6,2 (4,3)
 -Passo-21 -agente1- 2,3 (3,4) -agente2- 6,5 (4,5) -agente3- 3,6 (4,5) -agente4- 5,2 (4,3)
 -Passo-22 -agente1- 2,4 (3,4) -agente2- 5,5 (5,4) -agente3- 4,6 (4,5) -agente4- 5,3 (4,3)
 -(Captura) Passo-23 -agente1- 3,4 (3,4) -agente2- 5,4 (5,4) -agente3- 4,5 (4,5) -agente4- 4,3 (4,3)

Fim 2ª perseguição

Analisando especificamente este treinamento, a primeira captura ocorreu somente no passo 16. Como a estrutura ainda não aprendeu praticamente nada, o agente 1 tentou sair do grid algumas vezes quando está na posição (0, 0). Do passo 2 para o 3, os agentes 3 e 4, depois de avançarem corretamente em direção à presa, retornaram para onde estavam. Todas as movimentações, corretas ou não, colaboraram para enriquecer o conhecimento da estrutura RL-NFHP-MA, que foi usada na decisão da movimentação e foi atualizada conforme os agentes recebiam os retornos de suas ações.

A 2ª captura já foi realizada de forma ótima, em apenas 7 passos. A figura abaixo mostra o caminho exato percorrida por cada agente da largada até a 2ª captura.

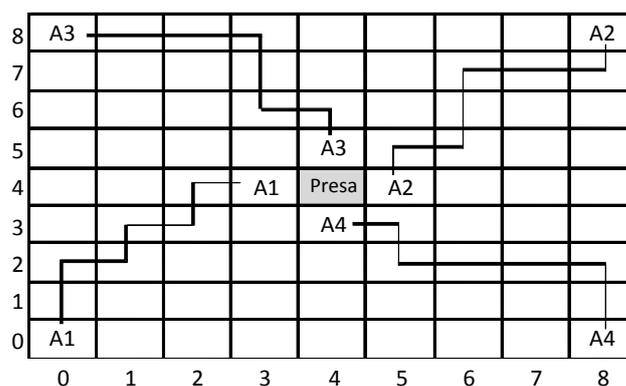


Figura 35: Sequência de movimentação dos agentes durante 2ª captura da presa.

Obviamente, nem todas as capturas seguintes levaram apenas 7 sete passos, que é a menor distância entre o agente e a presa. Ou seja, o caminho ótimo. Isso ocorreu porque o método de seleção das ações do algoritmo RL-NFHP, durante o processo de aprendizagem, nem sempre escolhe a ação que possui o maior valor de Q. Porém, pode-se ter uma idéia da velocidade de

convergência do método. No total, ao longo dos 100 ciclos de treinamento, 12 capturas foram realizadas. A figura abaixo mostra a quantidade de passos necessária para cada uma delas.

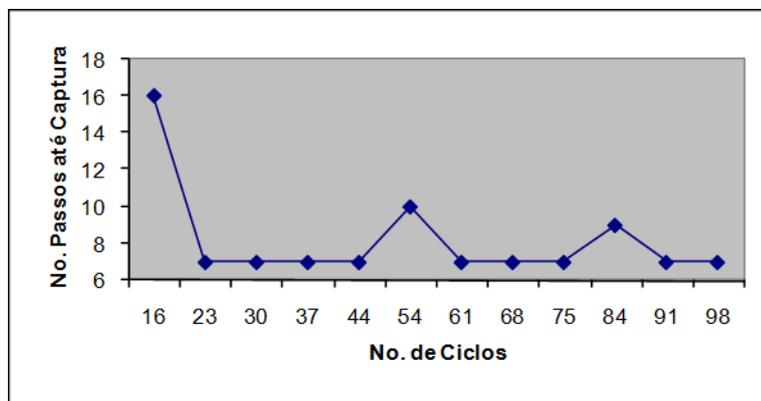


Figura 36: Número de passos que cada captura levou durante o aprendizado da versão multi-agentes.

Após o período de treinamento, o conhecimento adquirido armazenado na estrutura foi avaliado. Para a realização dos testes do modelo RL-NFHP para multi-agentes, novos procedimentos foram realizados. Desta vez, a posição da presa foi inicializada de forma aleatória. A estrutura RL-NFHP-MA foi utilizada pelos agentes a cada decisão de movimentação, em busca do objetivo.

Na primeira bateria de testes, um total de 1.000 “perseguições” foram realizadas, com os predadores largando sempre dos cantos do grid. A quantidade de passos média até a captura da presa foi de 12,983. Ou seja, foram necessários 12.983 passos para que a presa fosse capturada 1.000 vezes.

Em seguida, mais 1.000 “perseguições” foram feitas. Desta vez, a posição dos predadores na largada também foi inicializada de forma aleatória. A quantidade de passos média até a captura da presa foi de 9,476 movimentos, velocidade um pouco menor à obtida com largada fixa.

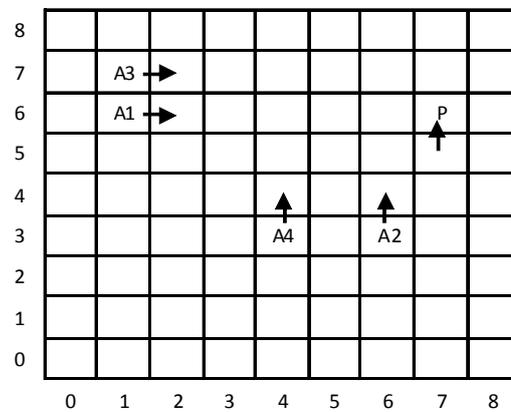
Apesar de ser difícil uma avaliação quantitativa da velocidade do processo de captura, devido às largadas randômicas dos participantes, e ainda, da movimentação aleatória da presa, o detalhamento das ações mostra que em todos os momentos os agentes foram corretamente em direção ao seu objetivo. Segue como exemplo a trilha realizada pelos participantes do jogo até a captura durante a 1ª perseguição do processo de teste. É importante perceber também a troca de objetivos dos agentes ao longo das perseguições, evidenciando uma *coordenação*

implícita entre os agentes. Conforme a movimentação dos demais agentes e da presa era alterada, o objetivo dos agentes também mudava.

- início perseguição 1

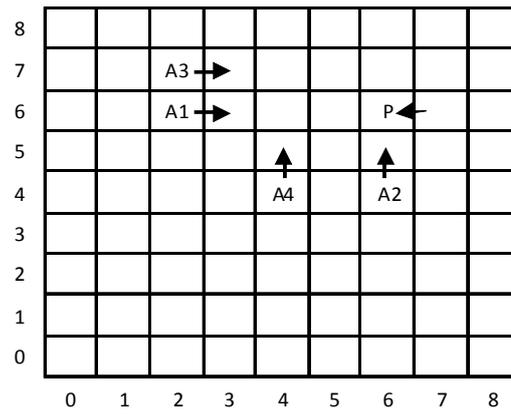
-Presa- 7,5 -agente1- 1,6 (7,6) -agente2- 6,3 (6,5) -agente3- 1,7 (8,5) -agente4- 4,3 (8,5)
 -Presa- 7,6 -agente1- 2,6 (6,6) -agente2- 6,4 (6,6) -agente3- 2,7 (7,7) -agente4- 4,4 (7,7)
 -Presa- 6,6 -agente1- 3,6 (5,6) -agente2- 6,5 (6,5) -agente3- 3,7 (6,7) -agente4- 4,5 (6,7)
 -Presa- 6,5 -agente1- 4,6 (6,4) -agente2- 6,4 (6,4) -agente3- 4,7 (7,5) -agente4- 5,5 (7,5)
 -Presa- 7,5 -agente1- 5,6 (7,6) -agente2- 7,4 (7,4) -agente3- 5,7 (8,5) -agente4- 6,5 (8,5)
 -Presa- 7,4 -agente1- 6,6 (7,3) -agente2- 7,3 (7,3) -agente3- 6,7 (8,4) -agente4- 7,5 (8,4)
 -Presa- 7,3 -agente1- 6,5 (7,4) -agente2- 7,2 (7,2) -agente3- 6,6 (7,4) -agente4- 7,4 (7,4)
 -Presa- 7,2 -agente1- 6,4 (7,3) -agente2- 7,1 (7,1) -agente3- 6,5 (7,3) -agente4- 7,3 (7,3)
 -Presa- 6,2 -agente1- 6,3 (6,3) -agente2- 6,1 (6,1) -agente3- 6,4 (6,3) -agente4- 7,2 (7,2)
 -Presa- 6,3 -agente1- 5,3 (5,3) -agente2- 6,2 (6,2) -agente3- 6,4 (6,4) -agente4- 7,3 (7,3)
 -Agarrou -Total passos: 10

A seqüência de figuras abaixo mostra o exemplo desta 1ª “caça” realizada pelos quatro agentes à presa.

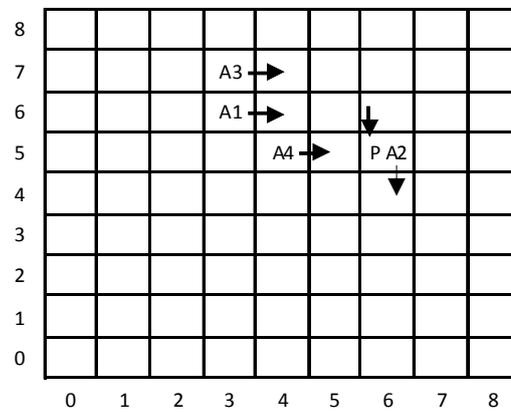


-Presa- 7,6 -agente1- 2,6 **(6,6)*** -agente2- 6,4 **(6,6)*** -agente3- 2,7 **(7,7)****
 -agente4- 4,4 **(7,7)****

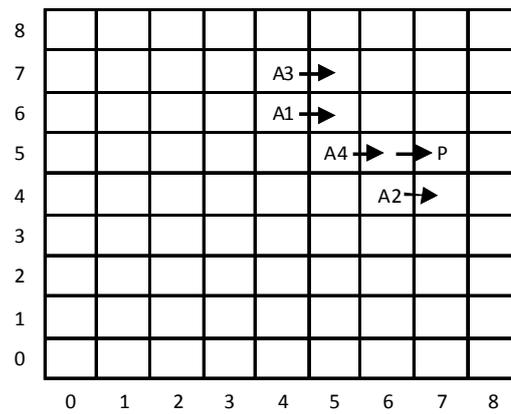
Obs: Já no primeiro momento, ocorrem 2 “colisões de objetivos” (* e **). O agente 1 e 2 perseguem a célula (6, 6) e os agentes 3 e 4 a célula (7, 7). Com a evolução do jogo, outras colisões semelhantes ocorrem até serem resolvidas naturalmente com a movimentação dos agentes.



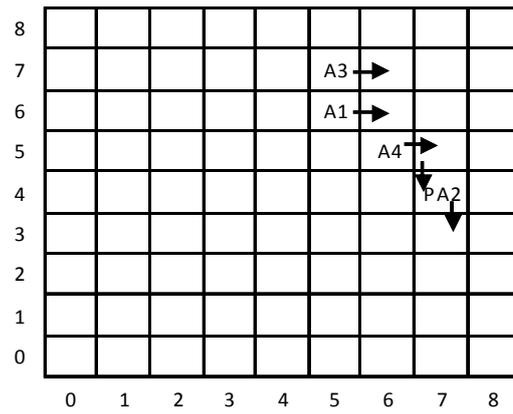
-Presas- 6,6 -agente1- 3,6 (5,6) -agente2- 6,5 (6,5) -agente3- 3,7 (6,7)
-agente4- 4,5 (6,7)



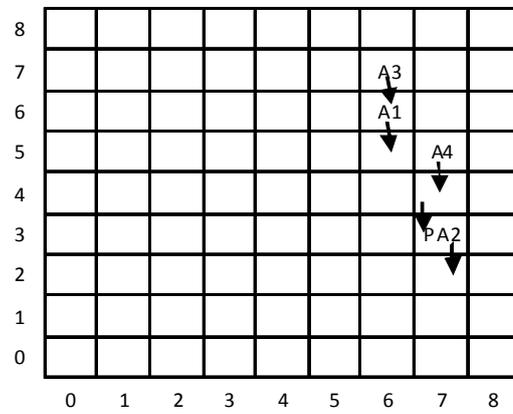
-Presas- 6,5 -agente1- 4,6 (6,4) -agente2- 6,4 (6,4) -agente3- 4,7 (7,5)
-agente4- 5,5 (7,5)



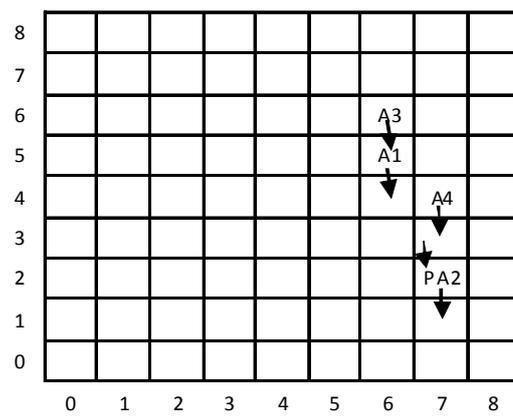
-Presas- 7,5 -agente1- 5,6 (7,6) -agente2- 7,4 (7,4) -agente3- 5,7 (8,5)
-agente4- 6,5 (8,5)



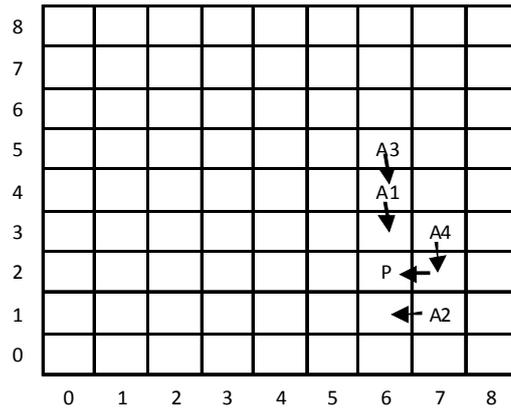
-Presa- 7,4 -agente1- 6,6 (7,3) -agente2- 7,3 (7,3) -agente3- 6,7 (8,4)
-agente4- 7,5 (8,4)



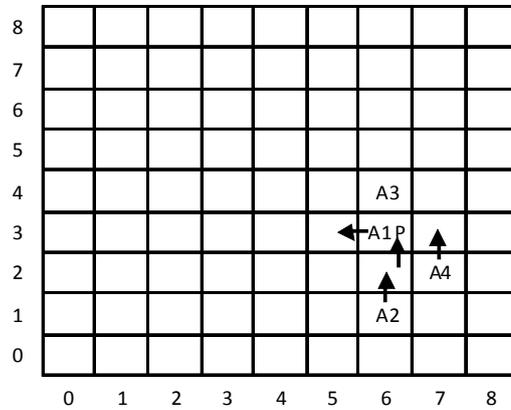
-Presa- 7,3 -agente1- 6,5 (7,4) -agente2- 7,2 (7,2) -agente3- 6,6 (7,4)
-agente4- 7,4 (7,4)



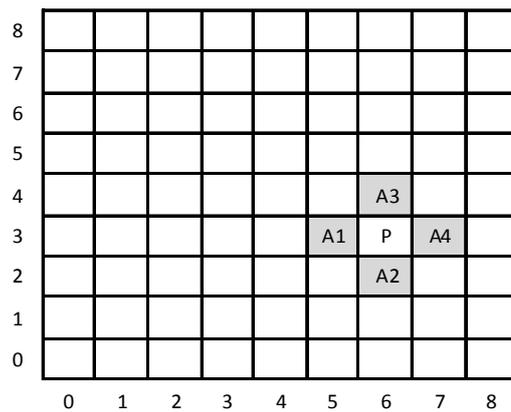
-Presa- 7,2 -agente1- 6,4 (7,3) -agente2- 7,1 (7,1) -agente3- 6,5 (7,3)
-agente4- 7,3 (7,3)



-Presa- 6,2 -agente1- 6,3 (6,3) -agente2- 6,1 (6,1) -agente3- 6,4 (6,3) -agente4- 7,2 (7,2)



-Presa- 6,3 -agente1- 5,3 (5,3) -agente2- 6,2 (6,2) -agente3- 6,4 (6,4) -agente4- 7,3 (7,3)



Abaixo, estão detalhadas as quatro primeiras perseguições, com largada randômica, passo a passo. Além da informação da posição dos agentes e seus objetivos, também são sendo exibidas as movimentações da presa ao longo da perseguição.

- início perseguição 1

-Presas- 7,5 -agente1- 1,6 (7,6) -agente2- 6,3 (6,5) -agente3- 1,7 (8,5) -agente4- 4,3 (8,5)
 -Presas- 7,6 -agente1- 2,6 (6,6) -agente2- 6,4 (6,6) -agente3- 2,7 (7,7) -agente4- 4,4 (7,7)
 -Presas- 6,6 -agente1- 3,6 (5,6) -agente2- 6,5 (6,5) -agente3- 3,7 (6,7) -agente4- 4,5 (6,7)
 -Presas- 6,5 -agente1- 4,6 (6,4) -agente2- 6,4 (6,4) -agente3- 4,7 (7,5) -agente4- 5,5 (7,5)
 -Presas- 7,5 -agente1- 5,6 (7,6) -agente2- 7,4 (7,4) -agente3- 5,7 (8,5) -agente4- 6,5 (8,5)
 -Presas- 7,4 -agente1- 6,6 (7,3) -agente2- 7,3 (7,3) -agente3- 6,7 (8,4) -agente4- 7,5 (8,4)
 -Presas- 7,3 -agente1- 6,5 (7,4) -agente2- 7,2 (7,2) -agente3- 6,6 (7,4) -agente4- 7,4 (7,4)
 -Presas- 7,2 -agente1- 6,4 (7,3) -agente2- 7,1 (7,1) -agente3- 6,5 (7,3) -agente4- 7,3 (7,3)
 -Presas- 6,2 -agente1- 6,3 (6,3) -agente2- 6,1 (6,1) -agente3- 6,4 (6,3) -agente4- 7,2 (7,2)
 -Presas- 6,3 -agente1- 5,3 (5,3) -agente2- 6,2 (6,2) -agente3- 6,4 (6,4) -agente4- 7,3 (7,3)

-Agarrou -Total passos: 10

- início perseguição 2

-Presas- 7,2 -agente1- 5,6 (8,2) -agente2- 2,4 (8,2) -agente3- 7,1 (7,1) -agente4- 4,4 (7,3)
 -Presas- 7,1 -agente1- 5,5 (8,1) -agente2- 3,4 (8,1) -agente3- 7,0 (7,0) -agente4- 5,4 (7,2)
 -Presas- 6,1 -agente1- 5,5 (7,1) -agente2- 4,4 (7,1) -agente3- 6,0 (6,0) -agente4- 5,3 (6,2)
 -Presas- 7,1 -agente1- 6,5 (8,1) -agente2- 5,4 (8,1) -agente3- 7,0 (7,0) -agente4- 6,3 (7,2)
 -Presas- 6,1 -agente1- 6,4 (6,2) -agente2- 5,3 (5,1) -agente3- 6,0 (6,0) -agente4- 6,2 (6,2)
 -Presas- 5,1 -agente1- 6,3 (5,2) -agente2- 5,2 (5,2) -agente3- 5,0 (5,0) -agente4- 6,2 (5,2)
 -Presas- 5,2 -agente1- 5,3 (5,3) -agente2- 4,2 (4,2) -agente3- 5,1 (5,1) -agente4- 6,2 (6,2)

-Agarrou -Total passos: 7

- início perseguição 3

-Presas- 5,6 -agente1- 7,6 (6,6) -agente2- 5,5 (5,7) -agente3- 3,4 (5,7) -agente4- 3,6 (5,7)
 -Presas- 5,5 -agente1- 7,5 (6,5) -agente2- 5,6 (5,6) -agente3- 3,5 (5,6) -agente4- 4,6 (5,6)
 -Presas- 5,4 -agente1- 7,4 (6,4) -agente2- 5,5 (5,5) -agente3- 4,5 (5,3) -agente4- 4,6 (5,3)
 -Presas- 6,4 -agente1- 7,4 (7,4) -agente2- 5,4 (6,3) -agente3- 5,5 (6,3) -agente4- 5,6 (6,3)
 -Presas- 7,4 -agente1- 7,3 (7,3) -agente2- 6,4 (7,3) -agente3- 6,5 (7,5) -agente4- 6,6 (7,5)
 -Presas- 6,4 -agente1- 6,3 (6,3) -agente2- 5,4 (5,4) -agente3- 6,5 (6,5) -agente4- 6,6 (6,5)
 -Presas- 6,5 -agente1- 6,4 (6,4) -agente2- 5,5 (5,5) -agente3- 7,5 (7,5) -agente4- 6,6 (6,6)

-Agarrou -Total passos: 7

- início perseguição 4

-Presas- 7,4 -agente1- 1,0 (6,4) -agente2- 5,1 (6,4) -agente3- 4,6 (7,5) -agente4- 6,5 (7,5)
 -Presas- 7,3 -agente1- 2,0 (6,3) -agente2- 5,2 (6,3) -agente3- 5,6 (7,4) -agente4- 7,5 (7,4)
 -Presas- 6,3 -agente1- 2,1 (5,3) -agente2- 5,3 (5,3) -agente3- 5,5 (6,4) -agente4- 7,4 (7,3)

-Presas- 5,3 -agente1- 2,2 (4,3) -agente2- 4,3 (4,3) -agente3- 5,4 (5,4) -agente4- 7,3 (6,3)
 -Presas- 6,3 -agente1- 3,2 (6,2) -agente2- 5,3 (5,3) -agente3- 6,4 (6,4) -agente4- 7,3 (7,3)
 -Presas- 6,2 -agente1- 4,2 (5,2) -agente2- 6,3 (6,3) -agente3- 6,4 (6,3) -agente4- 7,2 (7,2)
 -Presas- 7,2 -agente1- 5,2 (6,2) -agente2- 7,3 (7,3) -agente3- 7,4 (7,3) -agente4- 7,1 (7,1)
 -Presas- 7,3 -agente1- 5,3 (6,3) -agente2- 8,3 (8,3) -agente3- 7,4 (7,4) -agente4- 7,2 (7,2)
 -Presas- 7,2 -agente1- 6,3 (6,2) -agente2- 8,2 (8,2) -agente3- 7,3 (7,3) -agente4- 7,1 (7,1)
 -Presas- 7,1 -agente1- 6,2 (6,1) -agente2- 8,1 (8,1) -agente3- 7,2 (7,2) -agente4- 7,0 (7,0)
 -Presas- 7,2 -agente1- 6,2 (6,2) -agente2- 8,2 (8,2) -agente3- 7,3 (7,3) -agente4- 7,1 (7,1)

-Agarrou -Total passos: 11

5.3.1.2.1

Comparações com RL-NFHP para um único Agente

Com o objetivo de avaliar o ganho no tempo de convergência da versão multi-agentes em relação à versão original, um problema semelhante foi construído com apenas 1 agente predador e a presa. Foram realizados 50 ciclos de treinamento com 300 iterações cada. Em todos, a presa também ficava estática na posição central do grid (4, 4) e um único agente largava sempre de um dos corners do grid – posições (0, 0), (0, 8), (8, 0) e (8, 8). O objetivo era chegar até a posição (4, 4), onde a presa se localizava. Neste caso, a quantidade mínima de passos necessários são 8, em qualquer das posições de partida do agente.

Um dos ciclos de treinamento da versão original do RL-NFHP pode ser visualizado no gráfico abaixo. A figura mostra que a convergência ocorreu apenas por volta do passo 163 do treinamento, tempo 10 vezes maior que o algoritmo multi-agentes precisou. Na média obtida após os 50 ciclos, as capturas ocorreram em 166,82 passos.

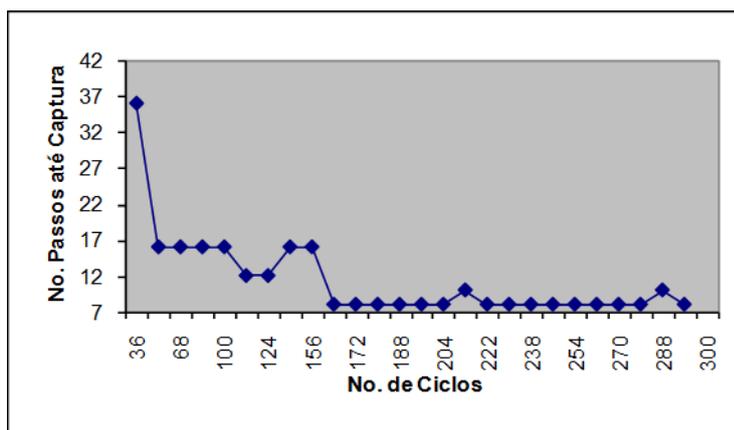


Figura 37: Número de passos que cada captura levou durante o aprendizado da versão original do modelo RL-NFHP.

5.3.1.2.2

Introdução dos Conceitos de *Não-Dominação* e *Satisfatoriedade*

Os testes seguintes tiveram como finalidade avaliar a introdução dos princípios de *Satisfatoriedade* e *Não-Dominação*, propostos por Goodrich e Quigley (2004). Primeiramente, o algoritmo RL-NFHP para um único agente foi modificado. A atualização dos valores-Q foi alterada e uma estrutura separada para as recompensas e penalidades foi criada. Os detalhes da alteração no algoritmo estão apresentados e detalhados na seção 4.2.

Posteriormente, novos treinamentos foram realizados. A presa permanecia fixa na posição central do grid (4, 4) e um único agente largava de um dos *corners* do grid. O objetivo era chegar até a posição (4, 4), onde a presa se localizava. Este treinamento foi realizado por 50 vezes. Enquanto a convergência na versão anterior do RL-NFHP ocorreu em média após 166,82 passos, após a modificação do algoritmo, o tempo médio necessário para se chegar a estrutura ótima foi reduzido para 143,76 passos, evidenciando um ganho de 13,82%.

A etapa seguinte foi a modificação da versão RL-NFHP-MA, criada para ambientes com múltiplos agentes. Exatamente os mesmos procedimentos foram realizados. Durante o treinamento, a presa esteve sempre fixa na posição (4, 4). A posição de cada agente no início do jogo e após cada captura foi inicializada sempre nos cantos do grid – posições (0, 0), (8, 8), (8, 0), e (0, 8). Os agentes

tenham como objetivo sair dos *corners* e capturar uma presa fixa no centro do grid. Na versão original, o algoritmo conseguiu chegar ao modelo ótimo após uma média de 16,13 passos. Na versão modificada, a convergência foi reduzida em 23,19%, para 12,39 passos.

O resumo dos novos resultados obtidos com a introdução dos princípios de *Satisfatoriedade* e *Não-Dominação* pode ser observado no quadro abaixo.

Tabela 7: Quadro com resultados alcançados pela versão modificada do algoritmo RL-NFHP, após a introdução dos conceitos de *Satisfatoriedade* e *Não-Dominação*

Ambiente	Algoritmo	Passos para convergência		% Redução
		Antes da Modificação	Após Modificação	
1 Agente - 1 Predador x 1 Presa	RL-NFHP	166,82	143,76	13,82%
Multi-Agente - 4 Predadores x 1 Presa	RL-NFHP-MA	16,13	12,39	23,19%

Devido aos resultados alcançados, a versão modificada do algoritmo passou a ser usada no restante deste Estudo de Caso, assim como nos Estudos de Casos posteriores.

5.3.1.3

Resultados após a Introdução de um Mecanismo de Coordenação

Em todos os testes realizados e descritos anteriormente, cada agente definia seus objetivos individualmente, com base na sua percepção do ambiente. A escolha do objetivo usado pelos agentes buscava encontrar sempre a formação em que soma das distâncias entre os agentes e objetivos fosse a menor possível. Logo, ao longo das perseguições, ocorreram colisões dos objetivos dos agentes. Ou seja, como um agente não sabia o objetivo dos demais, dois ou mais agentes poderiam escolher a mesma posição de chegada em um determinado momento do jogo. Essas colisões foram solucionadas naturalmente ao longo da movimentação dos agentes, conforme eles se aproximassem da presa e, conseqüentemente, de seus objetivos.

Novos testes foram realizados, após a introdução de um mecanismo de coordenação entre os agentes. Desta vez, os objetivos deixaram de ser escolhidos pelos próprios agentes, e passaram a ser definidos por um agente central, que possuía uma visão geral do ambiente. Com isso, a colisão de objetivos deixou de ocorrer ao longo das perseguições. O agente central, a cada rodada do jogo, escolhia diferentes posições de chegada dos agentes, de acordo com o posicionamento de cada um.

Inicialmente, os testes foram feitos com os agentes partindo sempre dos quatro cantos dos grids. Após 1.000 capturas, a média de passos obtidos foi de 11,236, valor 13,45% menor que a média de 12,983 alcançada quando o jogo foi realizado sem uma coordenação explícita entre os agentes.

Em seguida, foram realizadas mais 1.000 perseguições com os predadores sendo inicializados de maneira aleatória. A média de passos para captura foi de 7,695, número 18,79% menor que a média de 9,476 passos obtida sem o mecanismo de coordenação. O quadro abaixo mostra um resumo dos resultados obtidos e dos benefícios trazidos pela introdução do mecanismo de coordenação entre os agentes.

Tabela 8: Resultados comparativos do total de passos necessários para realização de 1.000 capturas, com e sem um mecanismo de coordenação explícito, ou “agente central”

Largada Predadores	Largada Presa	Coordenação Implícita	Coordenação Explícita (Agente Central)	% Redução
Aleatória	Aleatória	9.476	7.695	18,79%
Fixa (corners)	Aleatória	12.983	11.236	13,46%
Aleatória	Fixa (centro)	9.548	7.794	18,37%
Fixa (corners)	Fixa (centro)	13.161	11.294	14,19%

5.3.1.4

Introdução de Presa Inteligente – SMA Competitivo

Finalmente, com o objetivo de introduzir características competitivas a aplicação, o sistema foi alterado para que a presa deixasse de se movimentar de forma aleatória e se tornasse um agente inteligente RL-NFHP. Desta forma, a presa passou a competir com o grupo de agentes inteligentes predadores, com o objetivo de tornar o processo de captura o mais longo possível.

Após o aprendizado da presa, novas simulações foram realizadas no intuito de comparar a variação nos tempos das perseguições. Em todos os cenários testados, que contemplavam diferentes largadas e formas de coordenação entre os predadores, o número de passos necessários para as capturas aumentou significativamente. Este aumento evidenciou os benefícios da utilização do algoritmo RL-NFHP pela presa. Os resultados finais podem ser observados na Tabela 9.

Tabela 9: Resultados comparando o total de passos necessários para realização de 1.000 capturas com a presa inteligente e aleatória

		Presa Aleatória		Presa 'Inteligente'		Aumento de Tempo na Perseguição (%)	
Largada Predador	Largada Presa	Coordenação Implícita	Coordenação Explícita (Ag. Central)	Coordenação Implícita	Coordenação Explícita (Ag. Central)		
Largada aleatória	Largada aleatória	9.476	7.695	21.323	18.437	125,02%	139,60%
Largada fixa	Largada aleatória	12.983	11.236	18.978	16.175	46,18%	43,96%
Largada aleatória	Largada fixa	9.102	7.404	19.812	17.367	117,67%	134,56%
Largada fixa	Largada fixa	11.583	9.654	N/D	278.323	N/D	2782,98%

Nos testes realizados com largadas aleatórias da presa e dos predadores, o tempo de perseguição aumentou em 125,02%, quando não havia um agente central, e 139,60%, com a existência de um agente central coordenando os predadores. Isso significa que o tempo médio para realizar as capturas mais que dobrou, quando a presa deixou de se movimentar de forma aleatória e passou a ser um agente inteligente RL-NFHP.

Nos testes feitos com os predadores largando sempre dos *corners* do grid e a presa largando de forma aleatória, o tempo médio das perseguições aumentou em 46,18% e 43,96%. Ao inverter as largadas, ou seja, a presa largando sempre fixa do centro do grid e os predadores largando de pontos aleatórios, o aumento ficou respectivamente em 117,67% e 134,56%.

Finalmente, no experimento considerando largada fixa dos predadores dos quatro *corners* do grid e largada fixa da presa do centro do grid, a captura ficou praticamente impossível. Sem uma coordenação explícita, não foi possível capturar a presa (“N/D”) e, com um agente central, o tempo da perseguição

aumentou muito (2782,98%). Isso ocorreu porque na grande maioria das perseguições a presa acabou ficando colada nas bordas do grid.

5.3.1.5

Conclusões

Este primeiro estudo de caso do modelo RL-NFHP para multi-agentes, desenvolvido através do *pursuit game*, teve como principal objetivo mostrar como vários agentes podem explorar um ambiente de forma simultânea e contribuir para um aumento na velocidade da convergência do aprendizado.

Os testes realizados comparando o desempenho do aprendizado de um único agente com o desempenho de quatro agentes aprendendo em paralelo evidenciaram a grande diferença na quantidade de iterações necessárias para que o algoritmo conseguisse chegar à estrutura ótima. Além disso, também foi possível mostrar a coordenação implícita entre os agentes durante a captura da presa, mesmo sem que os agentes tivessem acesso aos objetivos, ou “pontos de chegada” em relação à presa, dos demais participantes do jogo.

Novos testes foram feitos após a introdução de um mecanismo explícito de coordenação, funcionando como um “agente central”, com o objetivo de sincronizar os objetivos dos agentes predadores a cada rodada do jogo. Os resultados mostraram a redução da quantidade média de rodadas necessárias para a captura, evidenciando os ganhos trazidos pela criação do mecanismo.

Finalmente, a aplicação foi modificada para que a presa também passasse a ser um agente inteligente RL-NFHP. Os resultados finais mostraram que, ao deixar de se movimentar aleatoriamente, a presa conseguiu dificultar consideravelmente a sua captura, aumentando de forma significativa o número de passos necessários para a conclusão dos ciclos de perseguição.

5.3.2

Leilão de Energia

5.3.2.1

Introdução

Neste segundo Estudo de Caso, foi desenvolvido um sistema multi-agente de natureza competitiva para simular um leilão de energia, em que os fornecedores de energia fazem ofertas para atender a uma determinada previsão de demanda durante um período de tempo.

5.3.2.1.1

Funcionamento dos Leilões

Um leilão pode ser considerado uma instituição de mercado com um conjunto de regras explícitas determinando uma alocação de recursos e preços com base em lances dos participantes no mercado (McAfee & McMillan 1987).

Existem diversos tipos de leilão. Eles podem ser classificados a partir de sua natureza (oferta, demanda ou duplo), pela forma como os lances são oferecidos (aberto ou fechado) e pela determinação do preço de fechamento (primeiro ou segundo preço) (Masili, 2004).

A natureza é definida de acordo com o papel exercido pelos participantes. Um leilão de demanda ocorre quando participam apenas 1 vendedor e m compradores. Havendo n vendedores e 1 comprador, o leilão pode ser classificado como um leilão de oferta ou reverso. Finalmente, o leilão duplo é caracterizado pela participação de n vendedores e m compradores.

Em relação à forma de oferta dos lances, existem basicamente dois tipos de leilão: leilão aberto (*open-outcry*) ou fechado (*sealed bid*). Nos leilões abertos, os lances são de conhecimento dos participantes e o preço final é definido após um processo dinâmico de estabelecimento do preço vencedor. Este processo pode ocorrer de maneira ascendente (leilão inglês) ou descendente (leilão holandês). Nos leilões fechados, por outro lado, os preços são enviados em paralelo pelos participantes para o leiloeiro. Com isso, os participantes não possuem acesso aos demais lances. O vencedor é determinado pelo melhor lance enviado.

O preço que define o valor pelo qual o bem leiloadado será negociado é o preço de fechamento. Ele pode ser considerado uniforme ou discriminatório. No leilão uniforme, os participantes exercem o mesmo preço, independente do valor de seus lances. Eles podem ainda ser subdivididos em leilões uniforme de primeiro e segundo preço. Nos leilões de primeiro preço, o preço de fechamento é definido pelo valor oferecido pelo lance vencedor. Já nos leilões de segundo preço (leilão de *Vickrey*), o preço de fechamento correspondente ao melhor lance perdedor, ou segundo melhor lance. Nos leilões discriminatórios (leilão *Yankee* ou *pay as bid*), cada participante deverá pagar ou receber o valor do seu lance.

5.3.2.1.2

Leilão de Energia Elétrica

Em diversos países, a indústria de energia elétrica foi desverticalizada e reestruturada, introduzindo uma maior competitividade entre os agentes participantes. Uma grande variedade de leilões foi implementada nos diferentes modelos de reestruturação. No geral, são perseguidos dois objetivos principais: aumentar a eficiência da alocação de recursos e da formação de preços da energia.

Neste estudo de caso, foi desenvolvido um sistema RL-NFHP-MA com o objetivo de maximizar o lucro dos agentes (fornecedores de energia) envolvidos nos leilões de demanda de energia de curto prazo. Foram utilizados dados do mercado brasileiro correspondente ao sub-mercado Sul, divulgados pelo Operador Nacional do Sistema em 2002.

Neste tipo de leilão, a demanda é negociada através de leilões reversos, onde o foco 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. O leilão é ainda de preço fechado (*sealed bid auction*), não sendo os lances de conhecimento público, e de preço uniforme, ou seja, os lances aceitos serão pagos ao preço do último e maior lance aceito.

Por exemplo, supondo que em um determinado dia a demanda é de 100MW e seis fornecedores participam do leilão ofertando 25MW cada, aos preços de 10 (R\$/MWh), 20, 30, 40, 50, 60, respectivamente. Ao final do leilão, o serão aceitos os lances de 10, 20, 30 e 40. Com isso, cada participante fornecerá

25MW, suprimindo a demanda total de 100MW. Porém, será pago o valor de 40 R\$/MWh para todos os agentes (preço do último e maior lance aceito).

Conforme mencionado anteriormente, o objetivo 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 hora. A figura abaixo mostra o perfil da demanda horária de energia, durante um período de 528 horas.

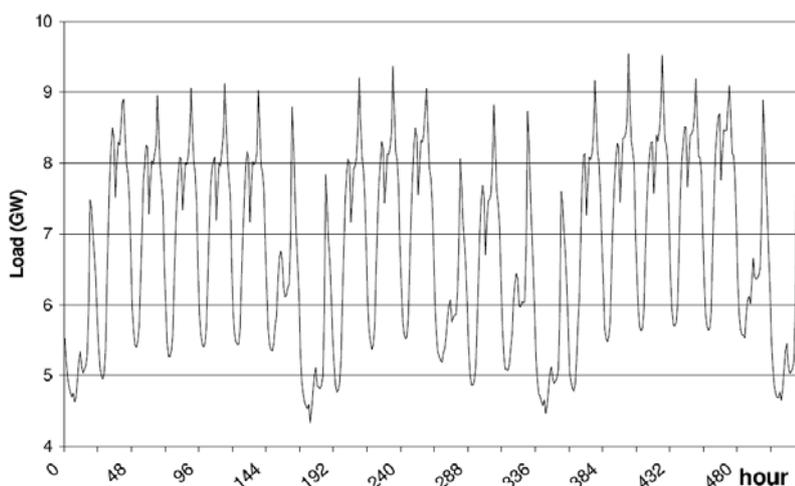


Figura 38: Perfil horário da demanda da região Sul.

Em um leilão com as características acima descritas, os procedimentos seguidos pelo leiloeiro são os seguintes:

- abrir o leilão
- publicar a previsão de demanda para o dia seguinte
- aceitar lances dos fornecedores de energia ou “agentes”
- encerrar recebimento de lances
- aplicar algoritmo de formação de preços
- publicar preço horário
- informar a cada agente a quantidade de energia a ser produzida
- encerrar leilão

O algoritmo de formação de preço funciona através da ordem do mérito (*merit order*). Neste método, os preços são ordenados do menor para o maior, até

atender à demanda do período. O preço horário é definido pelo último lance aceito para atender a demanda.

Em Walter e Gomide (2006), um sistema genético baseado em regras nebulosas (GFRBS) foi desenvolvido com o objetivo de gerar ofertas estratégicas para os fornecedores. Diferentemente das ofertas conservadoras, onde os lances são sempre equivalentes à sua capacidade ao custo marginal, um *bidding* estratégico pode ter outros objetivos, como maximizar o lucro do fornecedor. No trabalho citado, o fornecedor que seguiu as estratégias definidas pelo modelo GFRBS conseguiu obter lucros acima dos obtidos por uma estratégia conservadora. Ainda, ao fazer com que 2 agentes evoluíssem suas estratégias, Walter e Gomide (2006) mostraram que a evolução de um melhorou o desempenho do outro, que também utilizava o sistema GFRBS.

Neste trabalho, apresenta-se um SMA onde cada agente ou fornecedor de energia desenvolve sua estratégia, utilizando uma estrutura RL-NFHP, que visa sempre maximizar o lucro. Além de agentes RL-NFHP inteligentes, também participaram do leilão agentes com estratégias conservadoras.

O sistema GFRBS, desenvolvido por Walter e Gomide (2006), utiliza a teoria de conjuntos nebulosos para modelar as estratégias de ofertas. Assume-se que as estratégias são modeladas por sistemas baseados em regras nebulosas (FRBS, *fuzzy rule-based systems*). O sistema nebuloso de regras realiza uma aprendizagem evolutiva com o objetivo de encontrar regras transparentes e efetivas. É adotado um algoritmo evolutivo genético para otimizar a base de conhecimento para uma dada configuração de mercado.

Já no Estudo de Caso proposto neste trabalho, as estratégias de ofertas são definidas por agentes inteligentes, que utilizam o modelo RL-NFHP no processo de aprendizagem. Com isso, os agentes utilizam o mecanismo de aprendizagem por reforço hierárquico detalhado nos Capítulos 3 e 4.

5.3.2.1.3

Ambiente do Sistema

Nesta seção são apresentadas algumas características do sistema de potência e do ambiente competitivo utilizado para o desenvolvimento do SMA proposto. O ambiente é exatamente o mesmo utilizado em Walter e Gomide

(2006). A demanda do sistema é supostamente de conhecimento público e o leilão é um mecanismo que tem como finalidade alocar a potência a ser produzida por cada usina e definir o preço horário da energia. Como descrito anteriormente, os dados utilizados correspondem à carga real do sub-mercado Sul brasileiro, divulgada pelo Operador Nacional do Sistema (ONS 2002), durante o período de 3 semanas, a partir de 19 de maio de 2002. A Tabela abaixo mostra a lista das 24 termoeletricas consideradas no trabalho e seus respectivos custos marginais de produção.

Tabela 10: Relação das termoeletricas e suas características .

Usina	Tipo	Capacidade (MW)	Cap. Agregada (MW)	Custo marginal (R\$/MWh)
Angra 1	Nuclear	657	657	8,500
Angra 2	Nuclear	1.309	1.966	8,500
P.Médici 3-4	Carvão	320	2.286	32,946
P.Médici 1-2	Carvão	126	2.412	33,330
TermoBahia	Gás	171	2.583	34,378
TermoCeara	Gás	153	2.736	34,719
Canoas	Gás	450	3.186	37,543
Norte Fluminense	Gás	426,6	3.612,6	37,631
Araucária	Gás	441,6	4.054,2	37,697
Três Lagoas	Gás	324	4.378,2	37,762
Corumba	Gás	79,2	4.457,4	38,034
Juiz de Fora	Gás	103	4.560,4	38,732
Ibirité	Gás	766,5	5.326,9	39,065
TermoRio	Gás	824,7	6.151,6	39,109
Argentina I	Gás	1.018	7.169,6	41,045
Argentina II	Gás	1.000	8.169,6	41,046
J.Lacerda C	Carvão	363	8.532,6	52,638
J.Lacerda B	Carvão	262	8.794,6	63,302
J.Lacerda A 1-2	Carvão	100	8.894,6	67,105
J.Lacerda A 3-4	Carvão	132	9.026,6	67,350
Charqueadas	Carvão	69,1	9.095,7	67,723
FAFEN	Gás	57,6	9.153,3	74,782
Uruguaiana	Gás	582	10.266,3	82,765
Nova Piratininga	Gás	531	10.298,3	88,794

Assim como em Walter e Gomide (2006), os custos operacionais da produção de potência em usinas movidas a carvão, gás e óleo foram modelados como funções quadráticas da potência suprida pela usina $g_{j,h}$. Já as usinas nucleares e a de bagaço de cana foram modeladas como funções de custos lineares.

Na Tabela 11, encontram-se as funções de custo das termoelétricas. O custo marginal (R\$/MWh) na capacidade total, quarta coluna da Tabela anterior, é calculado a partir das funções de custo e corresponde ao preço da energia quando da oferta conservadora. Os participantes “conservadores” escolhem apenas o preço ao qual a energia é ofertada, pois assume-se que a quantidade será sempre equivalente à sua capacidade disponível.

Tabela 11: Funções de custo das termoelétricas.

Usina	$C_j(\cdot)$
Angra 1	8,5g
Angra 2	8,5g
P.Médici 3-4	$865,3 + 28,914g + 0,0063g^2$
P.Médici 1-2	$343,34 + 28,53g + 0,01905g^2$
TermoBahia	$580,54 + 30,985g + 0,00992g^2$
TermoCeara	$505,29 + 30,558g + 0,0136g^2$
Canoas	$1.575,22 + 33,869g + 0,00408g^2$
Norte Fluminense	$1.484,92 + 33,759g + 0,00454g^2$
Araucária	$1.505,65 + 33,782g + 0,00443g^2$
Três Lagoas	$1.115,29 + 33,595g + 0,00643g^2$
Corumba	$278,97 + 33,256g + 0,03016g^2$
Juiz de Fora	$323,68 + 33,088g + 0,0274g^2$
Ibirité	$3.632,08 + 31,966g + 0,00463g^2$
TermoRio	$3.904,05 + 31,912g + 0,00436g^2$
Argentina I	$4.459,61 + 32,775g + 0,00406g^2$
Argentina II	$4.379,82 + 32,774g + 0,00414g^2$
J.Lacerda C	$1.547,15 + 45,962g + 0,00919g^2$
J.Lacerda B	$1.407,65 + 56,198g + 0,01356g^2$
J.Lacerda A 1-2	$549,89 + 57,895g + 0,04605g^2$
J.Lacerda A 3-4	$728,6 + 57,65g + 0,03674g^2$
Charqueadas	$414,59 + 60,037g + 0,05559g^2$
FAFEN	$417,18 + 66,857g + 0,06879g^2$
Uruguaiana	$4.306,82 + 76,729g + 0,00519g^2$
Nova Piratininga	$4.547,35 + 82,418g + 0,006g^2$

A dinâmica dos preços segue a lógica de um leilão fechado de preço uniforme para o mercado do dia seguinte. A cada dia de atividade, o operador realiza as funções descritas em 5.3.2.1.2. Desta forma, o operador do leilão seleciona os geradores de hora em hora, visando minimizar o custo da energia. A demanda D_h é inelástica ao preço. O operador do leilão deve assegurar que para cada hora h , a demanda seja igual ao somatório das potências geradas pelas usinas

selecionadas. Considerando um preço uniforme, esta alocação custa $D_h v_h$, onde v_h é o preço de fechamento da hora h . Esta é a função que o leiloeiro busca minimizar.

A formação de preço funciona através da ordem do mérito (*merit order*). Os preços são ordenados do menor para o maior, até atender à demanda do período. A interseção da oferta com a demanda define o preço horário. Com isso, usinas que oferecem lances acima do preço horário não produzem energia. Produzem energia apenas as usinas que ofertarem preços iguais ou menores que o preço horário. Os lances dos fornecedores são dados pelo par $(q_{j,h}, p_{j,h})$, onde $q_{j,h}$ é a quantidade de energia oferecida pelo agente j , na hora h (em MW), e $p_{j,h}$ o preço.

5.3.2.2

Metodologia

A entrada utilizada pelos agentes foi a demanda horária D_h . Com base na demanda e no ‘ambiente’, composto pelas características das demais usinas, os agentes inteligentes aprendem simultaneamente qual preço traz o melhor resultado ou lucro em cada situação. Foi considerada apenas uma variável de decisão para os agentes inteligentes, que é o preço da oferta. A quantidade de energia ofertada foi sempre igual à capacidade do agente. A função de reforço usada pelo algoritmo foi o lucro do agente gerador de energia, dada por:

$$P_{j,h} = v_h g_{j,h} - C_j(g_{j,h}) \quad (35)$$

Na equação acima, v_h é o preço *spot* do sub-mercado, segundo a quantidade despachada $g_{j,h}$, e $C_j(g_{j,h})$ é a função de custo do agente j .

No primeiro experimento, apenas um agente, no caso Argentina II, utilizou o algoritmo RL-NFHP para tomar as decisões e realizar seus lances. Os demais agentes utilizaram estratégias conservadoras, fazendo ofertas de toda a sua capacidade ao preço definido pelo custo marginal à capacidade plena. A figura 39 mostra o funcionamento do sistema. À esquerda, está representada a sequência de demandas do sistema que precisa ser atendida a cada leilão. Na parte central, o

leiloeiro recebe as demandas do sistema e os lances dos fornecedores participantes do Leilão. E, à direita, estão representados os agentes (fornecedores), sendo um deles inteligente e os demais conservadores.

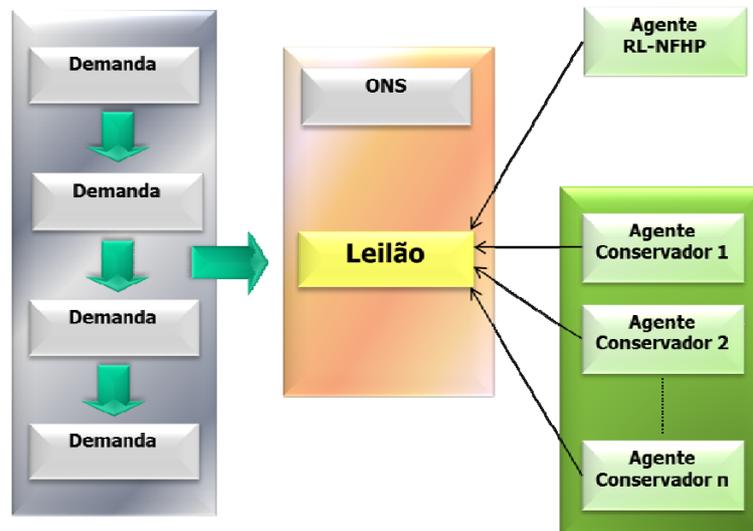


Figura 39: Funcionamento do Leilão de Energia multi-agentes, sendo 1 agente RL-NFHP.

Conforme mencionado anteriormente, foram utilizados dados correspondentes à carga real do sub-mercado Sul brasileiro, durante o período de 3 semanas, a partir de 19 de maio de 2002. Assim como em Walter e Gomide (2006), as primeiras 168 horas (1 semana) foram usadas para o treinamento e aprendizado do RL-NFHP-MA. Depois de criado o modelo, as 336 horas seguintes (2 semanas) foram usadas para testes do modelo.

O treinamento foi realizado em 500 ciclos. Em cada ciclo, todos os registros usados (primeiras 168 horas) foram apresentados ao algoritmo RL-NFHP-MA. A estrutura final do modelo possuía 18 células. A tabela 12 mostra os resultados obtidos pelo agente que tomou decisões através do algoritmo RL-NFHP, em relação aos agentes que utilizaram estratégias conservadoras. Podem ser observados a quantidade de horas em que foi produzida energia, o lucro e a quantidade de energia gerada por cada estratégia utilizada. O lucro líquido foi calculado subtraindo os custos fixos, durante as horas que o agente não produziu energia, do lucro bruto.

Tabela 12: Quadro com os resultados do primeiro experimento, com apenas um agente inteligente, competindo contra agentes de estratégias conservadoras.

Estratégia	Horas	Quantidade (MWh)	Lucro Bruto (R\$)	Lucro Líquido (R\$)
Conservadora	149	116.381	1.049.818	230.791
GFRBS	316	295.660	1.088.983	1.001.386
RL-NFHP	248	221.296	1.063.423	792.432
RL-NFHP x Conservadora	66%	90%	1%	243%
RL-NFHP x GFRBS	-22%	-25%	-2%	-21%

Observa-se que o modelo RL-NFHP conseguiu superar amplamente a estratégia conservadora, obtendo um lucro líquido 243% maior. Também foram incluídos na comparação os resultados obtidos pelo modelo GFRBS (Sistema Genético baseado em Regras Nebulosas), desenvolvido por Walter e Gomide (2006) e aplicado ao mesmo problema.

Apesar de ligeira superioridade do modelo GFRBS, pode ser observado que ambos os modelos geraram estratégias com lances de preços menores quando a demanda era baixa e lances com preços maiores que seu custo marginal quando havia a oportunidade do agente inteligente ser o gerador marginal (última usina a ser despachada que, conseqüentemente, determina o preço horário v_h). Isto porque ofertas a preços pequenos garantem que o agente inteligente seja despachado mesmo quando o preço da energia é menor que seu custo marginal a plena capacidade, porém, maior que o custo fixo. Como resultado, o agente reduz perdas em relação ao caso conservador, quando o agente não é despachado, e arca com custos fixos sem nenhuma produção e, com isso, sem receita.

Em um segundo experimento, foi inserido mais um agente RL-NFHP para tomar as decisões e realizar seus lances. Com isso, o ambiente passou a ser composto por dois agentes RL-NFHP e outros agentes conservadores, conforme mostra a figura abaixo.

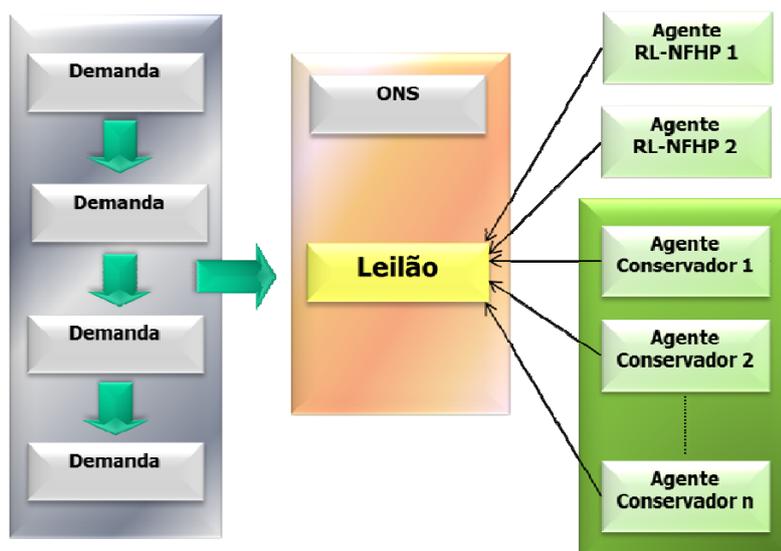


Figura 40: Funcionamento do Leilão de Energia multi-agentes, sendo 2 agentes RL-NFHP.

Em Walter e Gomide (2006), uma variação semelhante do experimento também é realizada. Ocorre a evolução de um segundo agente, Argentina I, supondo Argentina II como agente conservador. Ou seja, Argentina I sofre evolução em competição com o conjunto de agentes conservadores do mesmo modo que a evolução de Argentina II. Todos os parâmetros do algoritmo evolutivo foram mantidos com os mesmos valores utilizados na evolução de Argentina II. Ao final do processo de evolução de Argentina I, são tomados os melhores indivíduos encontrados em ambos os processos e adicionados ao *pool* de geradores. Novos resultados para o agente inteligente Argentina II são então extraídos. Desta vez, o agente compete com o agente inteligente Argentina I e com o *pool* de agentes conservadores. O trabalho mostra que os resultados obtidos por Argentina II são melhores que os previamente obtidos quando em competição com uma estratégia conservadora para Argentina I. Isto ocorre principalmente pelo fato de Argentina I fazer ofertas a preços maiores, mas é o agente Argentina II que produz mais energia a estes preços, se apropriando do ganho proporcionado por estes preços.

Já, neste trabalho, por se tratar de um Sistema Multi-Agente baseado na aprendizagem por reforço, os agentes podem aprender simultaneamente, de acordo com o algoritmo RL-NFHP-MA, descrito no Capítulo 4. Ou seja, ao invés de apenas um agente por vez ser treinado contra agentes conservadores, o modelo

permite que os agentes sejam treinados ao mesmo tempo, contra o *pool* de agentes conservadores.

Com isso, este segundo experimento foi dividido em duas etapas. Na primeira, os agentes RL-NFHP foram treinados separadamente considerando apenas agentes conservadores no ambiente. Sendo assim, o agente Argentina I foi treinado em um ambiente com os demais 23 agentes conservadores e, em seguida, Argentina II foi treinado da mesma forma. Durante o treinamento de Argentina I, o agente Argentina II utilizou a estratégia conservadora, e vice-versa. Na segunda etapa deste experimento, os dois agentes aprenderam simultaneamente a definir suas estratégias de *bidding*, considerando o ambiente com outros 22 agentes conservadores. Os resultados e comparações da primeira etapa do segundo experimento podem ser observados na tabela abaixo.

Tabela 13: Quadro com os resultados do segundo experimento, em que são utilizados dois agentes inteligentes.

Agente	Estratégia	Competição	Horas	Quantidade (MWh)	Lucro Bruto (R\$)	Lucro Líq (R\$)
Arg II	RL-NFHP	Conserv.	248	221.296	1.063.423	792.432
Arg II	GFRBS	GFRBS	316	307.523	1.479.871	1.392.274
Arg II	RL-NFHP	RL-NFHP	271	242.545	1.169.067	888.491
Arg I	RL-NFHP	Conserv.	159	143.123	1.892.312	1.021.323
Arg I	GFRBS	GFRBS	151	133.471	1.838.689	1.013.662
Arg I	RL-NFHP	RL-NFHP	168	152.321	1.912.323	1.033.232
Arg II - RL-NFHP/Conser x RL-NFHP/RL-NFHP			9,3%	9,6%	9,9%	12,1%
Arg I - RL-NFHP/Conser x RL-NFHP/RL-NFHP			5,7%	6,4%	1,1%	1,2%
Arg II - GFRBS/GFRBS x RL-NFHP/RL-NFHP			-14%	-21%	-21%	-36%
Arg I - GFRBS/GFRBS x RL-NFHP/RL-NFHP			11,3%	14,1%	4,0%	1,9%

Através dos resultados obtidos, algumas diferenças entre os modelos utilizados podem ser observadas. Primeiramente, nos dois experimentos em que foram utilizados apenas um agente inteligente RL-NFHP (Argentina II e Argentina I), foi possível superar os resultados alcançados por estes mesmos agentes no ambiente apenas com estratégias conservadoras. Em relação ao modelo GFRBS, considerando todos os demais agentes como conservadores, o RL-NFHP

obteve resultados inferiores no experimento com o agente Argentina II e superiores com Argentina I.

Assim como na utilização do modelo GFRBS por dois agentes, a introdução de um segundo agente utilizando estratégias inteligentes melhorou o desempenho do agente que havia aprendido (ou evoluído) em um cenário composta apenas de agentes conservadores.

Conforme descrito anteriormente, na segunda etapa deste experimento, foi desenvolvido um SMA com dois agentes, Argentina I e Argentina II, utilizando o algoritmo RL-NFHP com aprendizado simultâneo. Ou seja, ao invés do treinamento dos agentes ocorrer separadamente contra grupos de agentes com estratégias conservadoras, os dois agentes foram treinados ao mesmo tempo. Os novos resultados mostram que os dois agentes tiveram desempenhos superiores aos mesmos agentes treinados separadamente contra estratégias conservadoras. Os agentes Argentina I e Argentina II obtiveram um aumento de 4,2% e 22,8% no Lucro Líquido, respectivamente.

Tabela 14: Quadro com os resultados do segundo experimento, em que são utilizados dois agentes inteligentes com aprendizado simultâneo

Agente	Estratégia	Competição	Aprendizado	Horas	Qtd (MWh)	Lucro Bruto (R\$)	Lucro Líq (R\$)
Arg II	RL-NFHP	RL-NFHP	Individual	248	221.296	1.063.423	792.432
Arg II	RL-NFHP	RL-NFHP	Simultâneo	311	302.319	1.421.231	973.321
Arg I	RL-NFHP	RL-NFHP	Individual	168	152.321	1.912.323	1.033.232
Arg I	RL-NFHP	RL-NFHP	Simultâneo	176	160.231	1.982.123	1.076.212
Arg II - RL-NFHP/Indiv x RL-NFHP/Simult				25,4%	36,6%	33,6%	22,8%
Arg I - RL-NFHP/Indiv x RL-NFHP/Simult				4,8%	5,2%	3,7%	4,2%

5.3.2.3

Conclusões

Neste Estudo de Caso, foi desenvolvido um sistema multi-agente de natureza competitiva para simular um leilão reverso de energia, em que os fornecedores de energia fazem ofertas para atender a uma determinada previsão de demanda durante um período de tempo.

No SMA desenvolvido, foram criados agentes inteligentes RL-NFHP para atuar como fornecedores, que visavam maximizar o lucro através de ofertas estratégicas. Diferentemente das ofertas conservadoras, onde os lances são sempre equivalentes à sua capacidade ao custo marginal, um *bidding* ou oferta estratégica geralmente tem como finalidade maximizar o lucro do fornecedor. Além de agentes RL-NFHP inteligentes, também participaram do leilão agentes com estratégias conservadoras. Os resultados alcançados pelos agentes RL-NFHP foram comparados com os resultados obtidos por agentes conservadores e também pelos resultados obtidos por Walter e Gomide (2006), que desenvolveram um sistema genético baseado em regras nebulosas (GFRBS) com objetivos similares.

Vale reforçar que os modelos possuem características e propriedades distintas. No sistema GFRBS, as estratégias são modeladas por sistemas baseados em regras nebulosas (FRBS, *fuzzy rule-based systems*). Sobre o sistema nebuloso de regras, é realizada uma aprendizagem evolutiva com o objetivo de encontrar regras transparentes e efetivas. É adotado um algoritmo evolutivo genético para otimizar a base de conhecimento para uma dada configuração de mercado. Já, o RL-NFHP-MA é um modelo voltado para o aprendizado de múltiplos agentes e seu algoritmo de treinamento usa técnicas de aprendizado por reforço como base.

Comparando-se os resultados, observou-se que o novo modelo foi capaz de superar estratégias conservadoras utilizando os *bidding* estratégicos gerados pelo algoritmo. Em relação ao modelo GFRBS (Walter e Gomide, 2006), o RL-NFHP-MA obteve resultados similares, sendo superior em algumas situações e inferior em outras. Uma das vantagens apresentadas pelo novo modelo foi a capacidade de realizar um aprendizado paralelo dos agentes.

Outras simulações podem ser realizadas ainda dentro desta aplicação. Por exemplo, ao invés de decidir apenas o preço das ofertas, o *bidding* estratégico pode levar em conta também a quantidade ofertada. Para isso, o modelo precisa ser estendido para gerar duas saídas, ao invés de apenas uma. Além disso, o RL-NFHP-MA também pode vir a ser utilizado em outros tipos de leilão, como leilões de demanda, em que participam apenas 1 vendedor e m compradores, e leilões duplos, caracterizados pela participação de n vendedores e m compradores.

5.3.3

Sistema Multi-Agente para Gerenciamento Ágil de Projetos de Softwares em Ambientes Distribuídos

5.3.3.1

Introdução

Um Projeto é um esforço temporário, geralmente com restrições de tempo e recursos, que tem como finalidade produzir resultados específicos. Já o Gerenciamento de Projetos pode ser considerado como o processo através do qual se aplicam conhecimentos, capacidades, instrumentos e técnicas às atividades de Projetos, de forma a satisfazer as necessidades e expectativas dos diversos participantes envolvidos no mesmo (Nokes, 2007). De acordo com pesquisa conduzida pelo Standish Group em 2009, apenas 32% dos projetos são bem sucedidos, enquanto 44% não terminam dentro das restrições de tempo e orçamento previstas, e 24% falham por inteiro (Standish, 2009). Por isso, existe uma grande demanda por melhorias na área de gestão de projetos, especialmente em práticas de planejamento de tempo e recursos.

A responsabilidade pela gestão de um projeto é atribuída ao Gerente de Projeto (GP). O GP tem como missão coordenar e monitorar o progresso do desenvolvimento do projeto e verificar os desvios através de variáveis como custo, tempo e qualidade. De maneira geral, o principal objetivo do Gerente de Projeto é minimizar os riscos de fracasso. A gerência de projetos, como disciplina, vem sendo desenvolvida gradativamente, a partir de diversos campos de aplicação, como a construção civil, a engenharia mecânica, projetos militares, desenvolvimento de softwares, etc.

Neste trabalho, o foco será dado aos Projetos de Desenvolvimento de Sistemas de Informação, realizados por grandes equipes trabalhando em ambientes distribuídos. Nos dias atuais, o ambiente de desenvolvimento de Sistemas de Informação é extremamente competitivo e complexo. A presença de multinacionais, especialmente indianas e mais recentemente chinesas, que contam com mão-de-obra barata e especializada, faz com que as margens sejam pequenas, não havendo espaço para erros.

Além da competição global, problemas como limitações de recursos e entrega tardia de tarefas geram uma série de impactos financeiros negativos nos projetos. Isto ocorre porque, na prática, os projetos são orçados e cobrados de acordo com a estimativa de horas que serão gastas na execução das atividades. Estimativas erradas significam aumento de despesas sem haver uma contrapartida de receita. Dependendo do tamanho do erro e do atraso, os projetos podem sofrer grandes prejuízos financeiros.

De uma maneira geral, as estimativas são realizadas através de métodos heurísticos, usando opiniões especializadas, analogias e ponderações. O método PERT (*Program or Project Evaluation and Review Technique*) (Malcolm et al., 1959), por exemplo, baseia-se na determinação de três tipos de estimativas: mais provável, otimista e pessimista. Depois de colher as três estimativas de cada atividade, é utilizada uma fórmula para encontrar o tempo médio estimado para conclusão da atividade. A fórmula utilizada mais frequentemente está expressa na equação 36.

$$T_E = (O + 4M + P) \div 6 \quad (36)$$

T_E é o tempo estimado, O é a estimativa otimista, M é a estimativa mais provável e P a pessimista.

Já a técnica Delphi (Turoff e Linstone, 1975), batizada com este nome pela inspiração no "Oráculo de Delfus", utiliza a opinião de um grupo de especialistas, que muitas vezes podem não estar envolvidos nas dificuldades do dia-a-dia do projeto. Além disso, a alta rotatividade dos profissionais de Tecnologia da Informação, gerada pelo aquecimento do mercado, também contribui para o agravamento do problema. Ao deixar a empresa, o profissional leva com ele a 'opinião especializada' formada ao longo do tempo.

A falta de ferramentas e métodos capazes de estimar com maior precisão a duração das atividades gera um grande aumento na incerteza do sucesso no desenvolvimento dos projetos e faz com que as empresas percam competitividade, já que, quanto maior a imprecisão, maior será o risco do projeto, e maior deverá ser o valor cobrado para compensar o mesmo.

Ao longo dos últimos anos, diversas iniciativas foram desenvolvidas com o objetivo de minimizar os problemas descritos acima através de Sistemas Multi-

Agentes. Entre as principais motivações para a utilização de SMAs estão o aumento do tamanho e da complexidade dos projetos (Shu-guang, 2005; Gorodetsky et al., 2006; Blaschke et al., 2010) e ocorrência cada vez maior de equipes trabalhando em ambientes fisicamente distribuídos (Yan et al., 2000).

Blaschke et al. (2010) construíram um Sistema Multi-Agente com o objetivo de gerenciar projetos de desenvolvimento de circuitos integrados. Em seu trabalho, os agentes representam projetistas de circuitos que, em conjunto, formam uma equipe para criar uma aplicação composta de chips integrados. Os resultados mostraram que o SMA desenvolvido foi capaz de fornecer previsões mais confiáveis sobre o curso do projeto, identificar pontos fracos e gerar sugestões sobre como planejar e executar projetos futuros.

De forma semelhante, Gorodetsky et al. (2006) criaram um SMA focado no gerenciamento do desenvolvimento de microeletrônicos. Através do protótipo construído, foi possível automatizar o planejamento de projetos e a elaboração de cronogramas.

Já Wu et al. (2009) desenvolveram um SMA que auxilia Gerentes de Projetos de Softwares na elaboração de planos e na organização de projetos. Os agentes inteligentes coletam métricas sobre o andamento das atividades e oferecem previsões sobre a eficiência do projeto, além de recomendações para reduzir desvios e alcançar os objetivos estabelecidos no plano original do projeto.

Dosedla (2009) descreve a utilização de SMAs em diversas áreas do Gerenciamento de Projetos, incluindo o desenvolvimento de planejamentos, elaboração de cronogramas, simulação de cenários e, especialmente, gerenciamento de riscos. Eles destacam ainda os processos adaptativos de gerenciamento como principais beneficiários de SMAs.

Rabelo et al. (1999) apontam os SMAs como um dos mecanismos mais promissores no desenvolvimento de planejamentos e cronogramas em projetos de indústrias manufatureiras. Eles destacam a capacidade de integração de informações, comunicação e coordenação entre os agentes, como principais benefícios que podem ser extraídos dos SMAs neste tipo de aplicação.

Bodea et al. (2010) e Bodea, Badea (2010) modelam projetos como Sistemas Adaptativos Complexos, devido aos impactos dinâmicos gerados por fatores econômicos, e desenvolvem um SMA com foco na otimização das estimativas de tempo em projetos de pesquisa.

No trabalho de Yan et al. (2000), a ênfase é dada na aplicação de SMAs no suporte ao Gerenciamento de Projetos em ambientes distribuídos. São modelados agentes que representam atividades e recursos. O SMA proposto apresenta métodos para estimar as atividades e resolver conflitos na utilização de recursos através da troca de mensagens e negociação entre os agentes.

5.3.3.2

Metodologia Ágil de Gerenciamento de Projetos

Uma das metodologias que vem ganhando destaque no Gerenciamento de Projetos de Software, que é o foco deste Estudo de Caso, é a Metodologia Ágil de Desenvolvimento, especialmente o SCRUM (Takeuchi e Nonaka, 1986). Através deste tipo de metodologia, o desenvolvimento do projeto ocorre de forma iterativa e incremental. Estas características auxiliam o gerenciamento e controle de projetos complexos, que costumam sofrer diversas mudanças durante o seu andamento.

O nome SCRUM tem sua origem no Rugby, esporte em que jogadores colocam a bola em jogo, repetidamente, através de trabalho em equipe. Devido a suas propriedades, a metodologia enfatiza a comunicação, a auto-organização, a flexibilidade e a entrega incremental do resultado do projeto, o que reduz risco de problemas no projeto como um todo.

No SCRUM, participam diretamente do projeto três entidades:

- *Scrum Master*: responsável por remover qualquer impedimento no desenvolvimento das atividades pela equipe;
- *Product Owner*: mantém lista priorizada de itens de *Backlog* (lista de todas as funcionalidades desejadas no produto);
- Equipe: responsável por executar as atividades na ordem priorizada.

O desenvolvimento do projeto se dá através de *Sprints*. Um *Sprint* consiste em uma interação curta de desenvolvimento que tem como objetivo entregar parte do produto (ou um “conjunto de itens do *Backlog*”). Antes de cada *Sprint*, os itens do *Backlog* são selecionados pelo *Product Owner* e *Scrum Master*, de acordo com esforço e prioridade. Ao longo do desenvolvimento de um *Sprint*, podem ser

realizadas breves reuniões diárias, em que cada participante fala sobre o progresso conseguido, o trabalho a ser realizado, e o que o impede de seguir avançando. A figura 41 ilustra o mecanismo de funcionamento dos projetos que seguem a metodologia SCRUM.

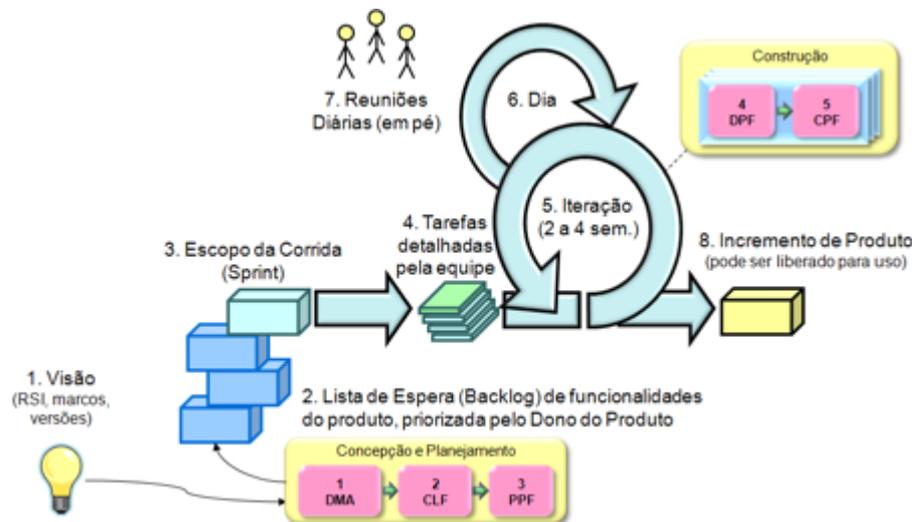


Figura 41: Resumo do desenvolvimento de projetos através de metodologia SCRUM (Schwaber, Beedle, 2002).

Apesar de trazer evoluções no Gerenciamento de Projetos de Software, as metodologias ágeis de desenvolvimento ainda apresentam algumas limitações, entre elas:

- as estimativas de duração das atividades continuam sendo realizadas através de métodos heurísticos por membros da equipe;
- dificuldade de trabalhar com equipes maiores ;
- dificuldade de trabalhar com equipes em ambiente distribuídos.

Visando superar estas limitações, este Estudo de Caso teve como objetivo o desenvolvimento de um novo Sistema Multi-Agente voltado para o Gerenciamento Ágil de Projetos de Softwares. Entre os objetivos do novo SMA, estão o suporte no processo de definição de métricas e estimativas de duração e a automatização de alguns dos principais processos realizados pelo Gerente de Projetos, como a alocação de recursos, planejamento de *sprints* e orçamentação.

Para realização dos experimentos, foram utilizados dados de projetos de uma das dez maiores integradoras de tecnologia brasileira, contendo informações de atividades executadas ao longo de diversos projetos. Os resultados obtidos pelo sistema proposto foram comparados com métodos tradicionais utilizados na área de Gerenciamento de Projetos.

5.3.3.3

Análise da Situação Atual

Antes da implantação do novo SMA, uma análise da situação atual do andamento dos projetos foi realizada. Foram coletados os dados dos últimos 60 projetos desenvolvidos pelos profissionais dos Escritórios de Projetos (EGPs) de RJ e SP. Ao longo dos 60 projetos foram concluídas 1.494 atividades agrupadas em *Sprints*, sendo 577 (38,6%) realizadas no RJ e 917 (61,4%) em SP.

As atividades foram desenvolvidas por 3 tipos de profissionais:

- Analistas de Requisitos: tinham como função principal entender os problemas junto aos clientes e desenvolver especificações funcionais para Programadores e Analistas de Testes ou QA (*Quality Assurance*);
- Programadores: responsáveis por desenvolver programas a partir das especificações geradas pelos Analistas de Requisitos;
- Analistas de QA: responsáveis pela realização de testes integrados dos programas desenvolvidos.

A assertividade das estimativas também foi analisada. No geral, a duração média de cada atividade foi de 6,44 dias, havendo um desvio médio de 2,02 dias por atividades, o que representa um desvio médio percentual de 31,33% em relação às durações estimadas.

A Tabela 15 mostra um resumo geral do desempenho das equipes de desenvolvimento de software ao longo dos 60 projetos observados nas filiais do RJ e SP, dividido pelos diferentes profissionais.

Tabela 15: Resumo geral do desempenho das equipes de desenvolvimento de software.

	Qtd Atividades	Duração Total Estimada	Duração Total Real	Desvio Total (Dias)	Duração Média Atividades	Desvio Méd. por Ativ. (Dias)	Desvio Médio Atividades (%)
Analista Req-RJ	151	604	878	274	5,82	1,82	31,24%
Programador-RJ	340	1.682	2.443	761	7,19	2,24	31,16%
Analista QA-RJ	86	516	734	218	8,54	2,54	29,74%
Totais RJ	577	2.802	4.056	1.254	7,03	2,17	30,92%
Analista Req-SP	271	1.084	1.618	534	5,97	1,97	32,99%
Programador-SP	508	1.896	2.753	857	5,42	1,69	31,12%
Analista QA-SP	138	828	1.200	372	8,70	2,70	31,00%
Totais SP	917	3.808	5.570	1.762	6,07	1,92	31,64%
Totais Gerais	1.494	6.610	9.626	3.016	6,44	2,02	31,33%

5.3.3.4

Metodologia

No SMA proposto neste Estudo de Caso, para cada tipo de profissional (foram considerados: Analista de Requisitos, Programador e Analista de QA) presente nos Escritórios de Projetos (EGPs), foi criado um Agente RL-NFHP. O objetivo individual dos Agentes era aprender qual a estimativa mais precisa para execução de uma determinada atividade.

Um Agente Central, representando o *Product Owner*, também foi desenvolvido. Ele tinha como objetivos distribuir as informações sobre as atividades, centralizar as informações sobre as estimativas de esforço e disponibilidade de recursos, e elaborar o planejamento das *Sprints*.

As informações entre os profissionais do EGP sobre estimativas de duração das atividades e alocação eram disponibilizadas em um *Blackboard* através de mensagens. As mensagens continham as seguintes informações:

- Estimativa de Duração da Atividade: código do projeto, código da atividade e duração estimada em dias;
- Disponibilidade: vetor com data do início e fim das alocações já determinadas pelo Gerente de Projetos.

O Agente representando o Gerente de Projetos acessava o *Blackboard* para obter as informações disponíveis e elaborar o cronograma do projeto. A figura 42 mostra o esquema de funcionamento do novo SMA proposto:

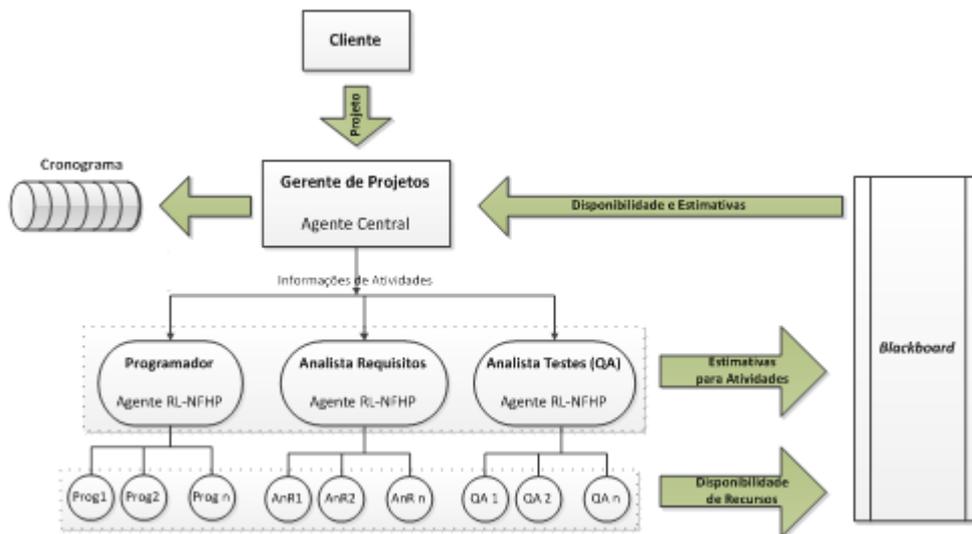


Figura 42: Esquema de funcionamento do novo SMA proposto.

O aprendizado dos Agentes ocorreu a partir da execução das atividades pelos profissionais do EGP. Cada atividade continha as seguintes informações:

- Nível do Profissional – Júnior, Pleno, Senior
- Linguagem - .NET (C#), Linguagem SQL
- Tipo de Funcionalidade - rotina batch, tela, serviço, procedure
- Complexidade - muito simples, simples, médio, complexo, muito complexo
- Tipo de Atividade – Nova, Alteração

Este conjunto de variáveis foi usado para definir os estados durante o aprendizado de cada Agente. Em relação às ações, foram criadas sete possíveis estimativas de duração, que deveriam ser atribuídas pelos agentes, a partir das características das atividades (ou estado). Cada estimativa poderia ser de 4 horas (ou 0,5 dias), 1, 2, 4, 6, 8 ou 10 dias úteis.

Após a verificação da duração real gasta na execução de cada atividade, cada Agente RL-NFHP correspondente ao profissional que realizou a tarefa recebia o reforço e aprendia, conforme algoritmo descrito nos Capítulos 3 e 4. O

reforço do modelo foi calculado com base nas estimativas previstas e reais das atividades. Quanto mais precisa a estimativa feita pelo Agente, maior era o reforço ou recompensa atribuída ao par de estado e ação correspondente. A Figura 43 ilustra, como exemplo, o funcionamento do Agente correspondente ao profissional com a função de Programador.

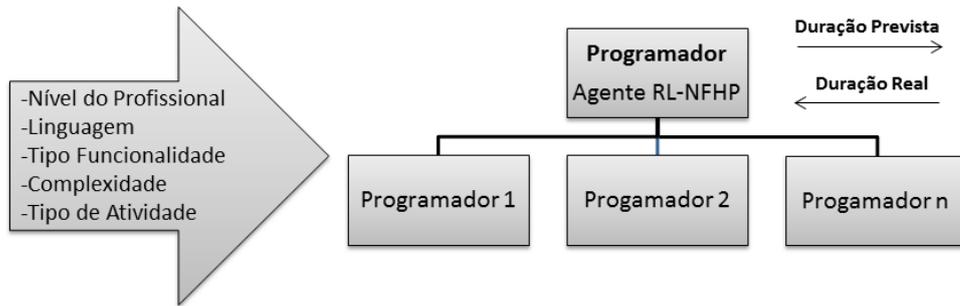


Figura 43: Esquema representando o processo de aprendizado do agente 'Programador'.

Para contemplar o desenvolvimento de projetos em múltiplos EGPs, foram criados Agentes distintos para profissionais alocados em outros EGPs, mesmo que desempenhassem a mesma atividade. Isso permite a modelagem de possíveis diferenças no desempenho dos profissionais em diferentes regiões. Abaixo, a figura mostra o funcionamento do modelo considerando dois EGPs.

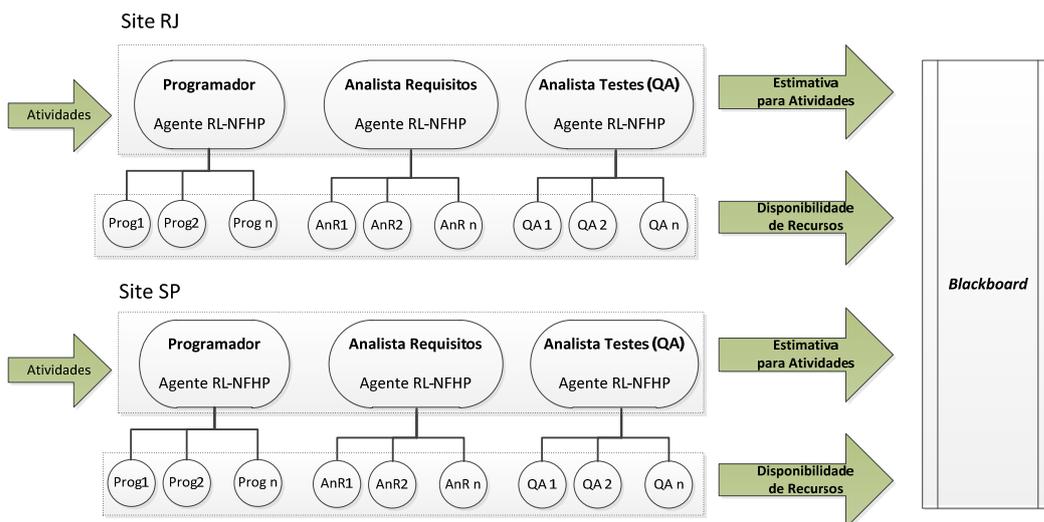


Figura 44: Esquema do SMA proposto considerando dois Escritórios de Projetos (EGPs)

Com isso, o SMA final foi construído com um Agente Central e seis agentes RL-NFHP, os quais possuíam estruturas independentes e eram alimentados com informações de diferentes profissionais de mesma função.

5.3.3.5

Resultados

Após a análise da situação atual (seção 5.3.3.3), onde foram analisadas os desvios ocorridos nos projetos e nas atividades, o novo SMA foi desenvolvido. Após a construção e treinamento das seis estruturas RL-NFHP, as estimativas passaram a ser feitas de forma automática pelo modelo.

A ideia era que, com a evolução das atividades, o SMA fosse capaz de se adaptar e gerar estimativas mais assertivas. Com isso, ele poderia substituir os métodos convencionais de planejamento usados nos EGPs.

O processo de aprendizado foi realizado utilizando dados de 778 atividades de 32 projetos, equivalente a cerca de 60% da base. Os padrões de treinamento foram apresentados durante 500 ciclos. Após a apresentação de cada padrão, a estrutura RL-NFHP era atualizada de acordo com o algoritmo descrito na seção 4.2.

Com o término do treinamento e obtenção da estrutura final, que possuía um tamanho de 32 células, foram iniciados os testes. Foram apresentadas as informações dos outros 21 projetos, que totalizavam 519 atividades, ao modelo obtido durante o processo de aprendizado.

Após o término deste processo, foram observados os ganhos trazidos pelo novo SMA de cada profissional das filiais abordadas no trabalho. O quadro abaixo mostra a precisão das estimativas realizadas separadamente pelos agentes inteligentes que representavam os profissionais nos EGPs do Rio de Janeiro e São de Paulo. Foram considerados somente os projetos estimados automaticamente pelo SMA após a fase de aprendizado (a partir do 33º projeto).

Tabela 16: Resumo geral do desempenho das equipes de desenvolvimento de software, após a implantação e aprendizado do novo SMA.

	Qtd Atividades	Duração Total Estimada	Duração Total Real	Desvio Total (Dias)	Duração Média Atividades	Desvio Méd. por Ativ. (Dias)	Desvio Médio Atividades (%)
Analista Req-RJ	52	255,2	301,5	46,3	5,75	0,88	15,36%
Programador-RJ	118	721,6	848,0	126,4	7,19	1,07	14,91%
Analista QA-RJ	41	221,2	252,3	31,1	6,15	0,76	12,33%
Totais RJ	211	1.198,0	1.401,8	203,8	6,63	0,96	14,54%
Analista Req-SP	89	476,2	564,1	87,9	6,34	0,99	15,58%
Programador-SP	171	814,5	952,6	138,1	5,57	0,81	14,50%
Analista QA-SP	48	338,7	401,1	62,4	8,37	1,30	15,56%
Totais SP	308	1.629,4	1.917,8	288,4	6,23	0,94	15,04%
Totais Gerais	519	2.827,4	3.319,6	492,2	6,39	0,95	14,83%

Os resultados mostram que em todas as classes de profissionais houve uma melhora significativa na precisão das estimativas geradas. As estimativas mais precisas vieram do agente ‘Analista QA-RJ’, com um desvio de 12,33%. No cenário anterior, este desvio era de 29,74%. O desvio médio geral por atividade foi reduzido de 2,02 dias para 0,95. Na prática, isso significa uma redução de 1,07 dias no erro da estimativa por atividade. Considerando o grande número de atividades desenvolvidas ao longo dos *sprints* e projetos, o ganho significa uma expressiva redução de custos.

A tabela 17 mostra uma síntese dos resultados da situação anterior e após a implantação do novo SMA baseado no modelo RL-NFHP-MA.

Tabela 17: Resultados comparativos do desempenho do modelo RL-NFHP-MA com técnicas convencionais.

	Métodos Convencionais		RL-NFHP-MA		Redução Desvio Médio por Atividade	
	Desvio Méd. por Ativ. (Dias)	Desvio Méd. Atividades (%)	Desvio Méd. por Ativ. (Dias)	Desvio Méd. Atividades (%)	Em dias	%
Analista Req-RJ	1,82	31,24%	0,88	15,36%	0,93	49,16%
Programador-RJ	2,24	31,16%	1,07	14,91%	1,17	47,84%
Analista QA-RJ	2,54	29,74%	0,76	12,33%	1,78	41,45%
Totais RJ	2,17	30,92%	0,96	14,54%	1,21	47,02%
Analista Req-SP	1,97	32,99%	0,99	15,58%	0,98	47,24%
Programador-SP	1,69	31,12%	0,81	14,50%	0,88	46,58%
Analista QA-SP	2,70	31,00%	1,30	15,56%	1,39	50,18%
Totais SP	1,92	31,64%	0,94	15,04%	0,99	47,53%
Totais Gerais	2,02	31,33%	0,95	14,83%	1,07	47,32%

5.3.3.6

Conclusões

Este terceiro Estudo de Caso teve como objetivo o desenvolvimento de um novo Sistema Multi-Agente voltado para o Gerenciamento Ágil de Projetos de Softwares. O novo SMA teve como objetivos primários o suporte no processo de definição de métricas e estimativas de duração e a automatização de alguns dos principais processos realizados pelo Gerente de Projetos.

Analisando os tipos de aprendizado, formas de interação, coordenação e objetivos apresentados na tabela 4 da seção 5.1 deste capítulo, pode ser observado que:

- o aprendizado dos agentes ocorreu através estruturas independentes RL-NFHP. Ao mesmo tempo, cada estrutura independente foi atualizada por múltiplos agentes;
- estas diferentes formas de aprendizado tiveram como objetivos, respectivamente, a especialização dos agentes em tarefas distintas e a exploração paralela do ambiente;
- a coordenação entre os agentes foi realizada de forma explícita, por um Agente Central correspondente ao Gerente de Projetos;
- como todos os agentes criados possuíam o mesmo objetivo, o SMA pode ser considerado de natureza colaborativa.

Os resultados mostraram que o novo sistema foi capaz de superar técnicas convencionais utilizadas na área de Gerenciamento de Projetos. O SMA desenvolvido conseguiu reduzir o erro das estimativas em todas as classes de profissionais envolvidas nos projetos nos dois Escritórios de Projetos contemplados.