UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

Aprendizado por Reforço com Valores de Influência em Sistemas Multi-Agente

Dennis Barrios Aranibar

Orientador: Prof. Dr. Luiz Marcos Garcia Gonçalves

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Computação da UFRN (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Doutor em Ciências.

Natal, RN, março de 2009

Divisão de Serviços Técnicos

Catalogação da publicação na fonte. UFRN / Biblioteca Central Zila Mamede

Pereira, Fulano dos Anzóis.

Sobre a Preparação de Propostas de Tema, Dissertações e Teses no Programa de Pós-Graduação em Engenharia Elétrica da UFRN / Fulano dos Anzóis Pereira - Natal, RN, 2006

23 p.

Orientador: Sicrano Matosinho de Melo

Co-orientador: Beltrano Catandura do Amaral

Tese (doutorado) - Universidade Federal do Rio Grande do Norte. Centro de Tecnologia. Programa de Pós-Graduação em Engenharia Elétrica.

1. Redação técnica - Tese. 2. LaTeX- Tese. I. Melo, Sicrano Matosinho de. II. Amaral, Beltrano Catandura do. III. Título.

RN/UF/BCZM CDU 004.932(043.2)

Sobre a Preparação de Propostas de Tema, Dissertações e Teses no Programa de Pós-Graduação em Engenharia Elétrica da UFRN

Fulano dos Anzóis Pereira

Dissertação de Mestrado aprovada em 31 de fevereiro de 2006 pela banca examinadora composta pelos seguintes membros:

Prof. Dr. Sicrano Matosinho de Melo (orientador) . . . . . DCA/UFRN

Prof. Dr. Beltrano Catandura do Amaral (co-orientador) . . . . . DCA/UFRN

Prof. Dr. Clint Stallone da Silva . . . . . DCEP/UFFN

Prof[a] Dr[a] Florisbela do Amaral . . . . . DCA/UFRN

A minha mãe Hermelinda, minha esposa Raquel e meus filhos, Sebastian e Sofia; o nosso amor me levou a concluir este trabalho.

Agradecimentos

Eu sem Deus não poderia fazer alguma coisa sequer, por este motivo eu agradeço primeiramente a ELE, por me fazer seu filho e me permitir cumprir meus objetivos.

Ao meu orientador, professor Luiz Marcos Garcia Gonçalves, sou grato pela orientação e especialmente pela amizade.

A minha familia, meu pai (in memoriam), meus irmãos, minha sogra e meus cunhados. Pelo apoio durante esta jornada.

Aos meus colegas e amigos Roque, Alzira, João Gabriel e José que me ajudaram com sua amizade, idéias, críticas e apoio na conclusão efetiva desta jornada.

Ao CNPq, pelo apoio financeiro.

Resumo

Propomos um novo paradigma de aprendizado coletivo em sistemas multi-agentes (SMA) como uma solução para o problema em que vários agentes devem aprender como realizar tarefas agindo sobre um mesmo ambiente, simultaneamente, baseando-se em retornos fornecidos por cada um dos outros agentes. Introduzimos o paradigma proposto na forma do algoritmo de aprendizado por reforço, denominando-o de *aprendizado por reforço com valores de influência*. Cada agente aprendendo por reforço avalia a relação existente entre o valor do seu estado atual e/ou a ação executada no estado (crenças atuais) junto com a recompensa obtida após todos os agentes que estão interagindo executarem suas ações (resultado da interferência dos outros). O agente pondera as opiniões de todos os seus colegas na tentativa de mudar os valores dos seus estados e/ou ações. A idéia é que o sistema como um todo deve chegar a um equilíbrio, onde todos os agentes se sentem satisfeitos com os resultados obtidos, significando que os valores dos estados ou pares estado/ação casam-se com a recompensa obtida por cada agente. Esta forma dinâmica de atualizar o valor dos estados e/ou ações faz deste novo paradigma de aprendizado por reforço o primeiro a incluir, naturalmente, o fato de que a presença de outros agentes no ambiente o torna dinâmico. Como resultado direto, incluímos implicitamente o estado interno, as ações e a recompensa obtida por todos os outros agentes dentro do estado interno de cada agente. Isso faz de nossa proposta a primeira solução completa para o problema conceitual que surge ao aplicar aprendizado por reforço em sistemas multi-agente, causado pela diferença existente entre o modelo do ambiente e o modelo do agente. Com base no modelo proposto, criamos o algoritmo IVQ-Learning, testado exaustivamente em jogos repetitivos com dois, três e quatro agentes e em jogos estocásticos que exijam cooperação e em jogos que exijam colaboração. Este algoritmo mostra-se como uma boa opção na tentativa de obter soluções que garantam a convergência para o equilíbrio de Nash ótimo em problemas cooperativos. Os experimentos realizados deixam claro que o paradigma proposto é teórica e experimentalmente superior aos paradigmas tradicionais. Ainda, com a criação deste novo paradigma, o conjunto de aplicações de aprendizado por reforço em SMA foi ampliado. Ou seja, além da possibilidade de aplicar os algoritmos nos problemas tradicionais de aprendizado em SMA, como por exemplo coordenação de tarefas em sistemas multi-robô, é possível aplicar aprendizado por reforço nos problemas essencialmente colaborativos.

Palavras-chave: Sistemas Multi-Agente, Coordenação, Colaboração, Aprendizado por Reforço, Auto-Organização.

Abstract

We propose a new paradigm for collective learning in multi-agent systems (MAS) as a solution to the problem in which several agents acting over the same environment must learn how to perform tasks, simultaneously, based on feedbacks given by each one of the other agents. We introduce the proposed paradigm in the form of a reinforcement learning algorithm, nominating it as *reinforcement learning with influence values*. While learning by rewards, each agent evaluates the relation between the current state and/or action executed at this state (actual believe) together with the reward obtained after all agents that are interacting perform their actions. The reward is a result of the interference of others. The agent considers the opinions of all its colleagues in order to attempt to change the values of its states and/or actions. The idea is that the system, as a whole, must reach an equilibrium, where all agents get satisfied with the obtained results. This means that the values of the state/actions pairs match the reward obtained by each agent. This dynamical way of setting the values for states and/or actions makes this new reinforcement learning paradigm the first to include, naturally, the fact that the presence of other agents in the environment turns it a dynamical model. As a direct result, we implicitly include the internal state, the actions and the rewards obtained by all the other agents in the internal state of each agent. This makes our proposal the first complete solution to the conceptual problem that rises when applying reinforcement learning in multi-agent systems, which is caused by the difference existent between the environment and agent models. With basis on the proposed model, we create the IVQ-learning algorithm that is exhaustive tested in repetitive games with two, three and four agents and in stochastic games that need cooperation and in games that need collaboration. This algorithm shows to be a good option for obtaining solutions that guarantee convergence to the Nash optimum equilibrium in cooperative problems. Experiments performed clear shows that the proposed paradigm is theoretical and experimentally superior to the traditional approaches. Yet, with the creation of this new paradigm the set of reinforcement learning applications in MAS grows up. That is, besides the possibility of applying the algorithm in traditional learning problems in MAS, as for example coordination of tasks in multi-robot systems, it is possible to apply reinforcement learning in problems that are essentially collaborative.

Keywords: Multi Agent Systems, Coordination, Collaboration, Reinforcement Learning, Self-Organization.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos e Abreviaturas

A_i	Conjunto de todas as ações possíveis para o agente *i*
IV	Valor de influência
OP	Opinião de um agente em relação a outro
$Pe(s,a)$	Percentagem do número de vezes que a ação *a* é executada no estado *s*
$Q(s,a)$	Valor do par estado-ação (*s*-*a*)
$V(s)$	Valor do estado *s*
Σ_i	Conjunto de todas as estratégias do agente *i* em MAS
β	Coeficiente de influência
σ_i	Estratégia de um agente *i* em MAS
σ_i^*	Estratégia escolhida pelo agente *i* em MAS
$a_i(t)$	Ação do agente *i* executada no tempo *i*
r_i	Recompensa obtida pelo agente *i* em MAS
$s(t)$	Estado do jogo no tempo *t*
IL	Aprendizado independente - Independent Learning
IVRL	Aprendizado por Reforço com Valores de Influência
JAL	Aprendizado de ações conjuntas - Joint Actions Learning
MAS	Sistema Multi-Agente
MAS	Sistemas Multi-Agente
MDP	Processos de Decisão Markovianos (*Markov Decission Process*)
MRS	Sistemas Multi-Robô
RG	Jogos Repetitivos (*Repetitive Games*)
RL	Aprendizado por Reforço (Reinforcement Learning)

SG Jogos Estocásticos (*Stochastic Games*)

TL Aprendizado como um time - Time Learning

Capítulo 1

Introdução

Softwares que empregam a tecnologia de agentes são cada vez mais utilizados na solução de problemas em diferentes áreas de aplicação. Naturalmente, há inúmeras definições para o termo agente, que pode ser visto como uma abstração de entidades computacionais [Fisher et al. 2007], como uma extensão do paradigma de programação orientado a objetos [Aguilar-Ponce et al. 2007], como um paradigma para o projeto de arquiteturas flexíveis para sistemas com necessidades diferentes [Anthony & Jannett 2006] e assim por diante.

No presente trabalho, consideramos um agente como um programa que controla um determinado hardware (agente robótico) ou como um programa embutido em um ambiente simulado (agente de software), ambos com capacidade de perceber e modificar o ambiente mediante utilização de sensores e ações específicas.

Consideramos um sistema multi-agente (MAS) como um conjunto de agentes interagindo num determinado ambiente, de forma a cooperarem e/ou competirem na solução de um determinado problema. No caso de um sistema multi-agente desenvolvido para controlar hardwares (robôs), pode-se também denominá-lo de sistema multi-robô (MRS) [Barrios-Aranibar & Alsina 2007].

Sistemas multi-agente podem ser aplicados em problemas onde as tarefas não podem ser executadas por um único agente, sendo necessária uma *coordenação* entre os agentes para completar a resolução do problema, ou então em problemas em que as tarefas podem ser divididas em sub-tarefas independentes e assim, os agentes podem prover *colaboração*, visando a sua solução. Exemplos de tais aplicações incluem a modelagem de problemas de comércio eletrônico [Chen et al. 2008, Nassiri-Mofakham et al. 2009, Wei et al. 2008], o envio de cartas utilizando robôs [Carrascosa et al. 2008], o agendamento de rotas de caminhões [Mes et al. 2007], a modelagem da demanda de energia em cidades ou países [Toksari 2007], missões de resgate de vítimas de desastres [Rooker & Birk 2005, Zhang et al. 2008], mapeamento de ambientes desconhecidos [Li et al. 2008, Pan et al. 2008, Rekleitis et al. 2001, Fenwick et al. 2002, Sujan & Dubowsky 2002, Huntsberger et al. 2003, Ryde & Hu 2005, Rocha et al. 2005, Yun et al. 2005], implementação de sistemas de vigilância usando robôs [Fernández et al. 2008], entre outras.

Geralmente, entre as etapas executadas na solução de problemas utilizando sistemas multi-agente, encontram-se as de agendamento e de execução das tarefas. O agendamento deve ser realizado de forma que todos os agentes no sistema sejam utilizados e

que o problema seja totalmente resolvido. A etapa de execução propriamente dita deve ser realizada de forma que os agentes não atrapalhem uns aos outros (coordenação e/ou colaboração), resolvendo o problema em questão.

Ambas as etapas são realizadas de forma independente e diferentes técnicas estatísticas e de aprendizado de máquina são utilizadas na implementação das mesmas, muitas vezes comprometendo o rendimento do sistema. Por exemplo, ao se implementar um algoritmo genético para o agendamento de tarefas e técnicas de enxames inteligentes para a execução das mesmas, muito processamento pode ser dedicado aos cálculos necessários por esta técnicas, não priorizando o tempo.

Desta forma, o desenvolvimento de novos algoritmos que permitam realizar estas duas etapas de forma conjunta torna-se importante, sendo necessário reduzir a complexidade dos sistemas sem comprometer a eficiência dos mesmos.

1.1 Motivação

A idéia de usar agentes que possam *aprender* a resolver problemas tornou-se popular na área de Inteligência Artificial, mais especificamente em Aprendizado de Máquina. Ao utilizar-se um sistema multi-agente, tradicionalmente, o aprendizado pode ser implementado nas duas etapas explicadas na seção anterior, isto é, instanciando-se agentes que aprendem a alocar tarefas para a primeira etapa e agentes que aprendem a resolver as suas próprias tarefas na segunda etapa.

Uma das formas tradicionais de dotar agentes da capacidade de aprender é com o uso de técnicas de aprendizado por reforço (RL). Esse tipo de algoritmo é bem sucedido quando aplicado à problemas que podem ser modelados com um único agente [Dalamagkidis et al. 2007, Tanaka et al. 2007, Usaha & Barria 2007, Vengerov 2007]. Porém, quando aplicados, sem nenhuma modificação, em problemas que devem ser modelados com vários agentes, mesmo garantindo a convergência, nem sempre esta ocorre para a solução ótima [Claus & Boutilier 1998]. Dessa forma, muitas propostas têm sido desenvolvidas com o intuito de poder convergir para as soluções ótimas. As mesmas baseiam-se na idéia de modificar o comportamento do próprio agente para poder se adaptar ao comportamento dos outros [Panait & Luke 2005, Shoham et al. 2007].

Uma idéia ainda não explorada dentro do aprendizado por reforço em sistemas multi-agente é tentar influenciar os outros agentes para que estes mudem seu comportamento e não mudar o próprio. Assim, a pesquisa direcionada ao desenvolvimento de algoritmos baseados na idéia anterior foi uma das principais motivações desta tese, uma vez que, no início da nossa pesquisa, conjeturamos que existia uma grande probabilidade de se obter melhores resultados em relação às técnicas tradicionais, o que foi corroborado durante o desenvolvimento dos nossos trabalhos.

Além disso, inspirados pelas idéias colocadas por Vygotsky [Jars et al. 2004, de Oliveira 1997], conjeturamos que as pessoas tentam mudar suas atitudes em função das opiniões dos outros, sejam estas negativas ou positivas, na tentativa de atingir um certo equilíbrio, nas interações sociais. Isto significa que as opiniões das pessoas podem gerar mudanças no comportamento dos outros. Desse modo, essas teorias reforçaram mais ainda a idéia

de desenvolver algoritmos que mimetizem as interações sociais das pessoas no tocante às opiniões e sua influência no comportamento dos outros.

Finalmente, um outro aspecto motivador da presente pesquisa foi a possibilidade de verificar se, ao implementar algoritmos de aprendizado por reforço usando as idéias sobre interações sociais colocadas por Vygotsky, os agentes conseguiriam adquirir a capacidade de se auto-organizar. Notamos que isto representou uma possibilidade (grande) de juntar as duas etapas geralmente executadas na solução de problemas utilizando sistemas multi-agente, explicadas na seção anterior. Como será visto no decorrer desta tese, isto se tornou realidade, após a realização dos desenvolvimentos inerentes à nossa pesquisa.

1.2 A Proposta

Nossa principal motivação para desenvolvimento deste trabalho surgiu a partir de uma série de questionamentos que vieram à tona após análise mais detalhada do estado da arte sobre as técnicas de aprendizado em sistemas multi-agente, assim como sobre as teorias de Vigotsky. Esses questionamentos são detalhados a seguir:

- **Q1:** É possível criar algoritmos de aprendizado por reforço usando as idéias de Vigotsky?
- **Q2:** Esses algoritmos terão a capacidade de convergir quando aplicados a problemas de coordenação?
- **Q3:** Esses algoritmos terão a capacidade de convergir quando aplicados a problemas de colaboração?
- **Q4:** Agentes implementados usando esses algoritmos terão a capacidade de se auto-organizar?
- **Q5:** A capacidade de convergência desses algoritmos para uma solução ótima superará a capacidade dos algoritmos tradicionais?
- **Q6:** A velocidade de convergência destes algoritmos superará a velocidade de convergência dos algoritmos tradicionais?
- **Q7:** O esforço computacional destes algoritmos será maior que nos algoritmos tradicionais?
- **Q8:** Qual a influência dos parâmetros no rendimento destes algoritmos?

1.2.1 Objetivos

Tendo como base os questionamentos acima, podemos formular o objetivo principal desta tese de doutorado como: *propor técnicas de aprendizado por reforço, baseadas nas idéias sobre interações sociais de Levy Vigotsky, que permitam construir sistemas multi-agente com capacidade de resolver problemas de coordenação e colaboração, tendo sempre como premissa que as opiniões de cada agente têm a capacidade de gerar mudanças no comportamento dos colegas.*

Assim, podemos adiantar que nossa principal contribuição é o desenvolvimento de um novo paradigma de aprendizado por reforço em sistemas multi-agente que tem maior probabilidade de convergência para soluções ótimas do que os paradigmas tradicionais.

Além disso, o mesmo permite que os agentes se auto-organizem, ou seja, dividam de forma autônoma as tarefas. Neste modelo, inspirado nas interações sociais, os agentes tentam mudar o comportamento dos colegas opinando em relação às ações que os últimos executaram.

Para atingir o objetivo principal do trabalho, foram definidos e cumpridos os seguintes objetivos específicos (ou metas):

1. Formalizar um modelo de aprendizado por reforço onde os agentes aprendem através das opiniões de outros agentes. Denominamos o modelo em questão de *aprendizado por reforço com valores de influência* ou *IVRL*, da sigla em inglês *Influence Value Reinforcement Learning*. Convém ressaltar que esta é uma denominação nossa.
2. Desenvolver algoritmos baseados no modelo proposto e em eventuais modificações de algoritmos tradicionais.
3. Mostrar experimentalmente a convergência a soluções ótimas destes algoritmos em problemas com número finito de estados, ações e agentes.
4. Testar em diferentes problemas de laboratório o modelo e algoritmos propostos, mostrando experimentalmente sua capacidade de convergência e avaliando o seu comportamento.
5. Mostrar experimentalmente a capacidade de auto-organização destes algoritmos e compará-la com a capacidade dos algoritmos tradicionais.
6. Avaliar e determinar a influência dos parâmetros do modelo proposto na sua capacidade de convergência.

1.2.2 Contribuições Principais

Com o desenvolvimento do presente trabalho, contribuímos diretamente para a área de aprendizado em sistemas multi-agente, principalmente melhorando a capacidade de convergência de algoritmos de aprendizado para soluções ótimas, resolvendo os problemas de alocação de tarefas e os de colaboração e de coordenação entre agentes. Temos ainda nesta tese, algumas contribuições secundárias relacionadas aos paradigmas tradicionais, assim como aos ambientes de teste usados em sistemas multi-agente. Neste sentido, podemos considerar as seguintes contribuições pontuais:

- **C1:** Desenvolvemos um novo paradigma de aprendizado em sistemas multi-agente inspirado nas idéias sobre interações sociais de Levy Vigotsky.
- **C2:** Desenvolvemos algoritmos de aprendizado por reforço, utilizando o paradigma proposto e baseados no algoritmo Q-Learning, para jogos repetitivos (IVQ-Learning).
- **C3:** Desenvolvemos algoritmos de aprendizado por reforço, utilizando o paradigma proposto e baseados no algoritmo Q-Learning, para jogos estocásticos (IVQ-Learning).
- **C4:** Realizamos uma revisão detalhada das contribuições realizadas até o momento em aprendizado por reforço em sistemas multi-agente, identificando nestas contribuições paradigmas tradicionais de implementação e fazendo uma análise das fortalezas e debilidades dos mesmos baseados nesta classificação.

- **C5:** Implementamos e analisamos o rendimento de algoritmos baseados nos paradigmas tradicionais (aprendizado independente e aprendizado de ações conjuntas) e no algoritmo Q-Learning, em problemas com jogos repetitivos e jogos estocásticos. Principalmente se avaliou a capacidade de convergência dos mesmos para um equilíbrio de Nash ótimo.
- **C6:** Estendemos a definição de dois problemas de teste padrão em sistemas multi-agente para jogos repetitivos para o teste de algoritmos com N-Agentes (o *Climbing Game* e o *Penalty Game*).
- **C7:** Mostramos experimentalmente a superioridade do algoritmo IVQ-Learning sobre suas versões baseadas nos paradigmas tradicionais em relação a sua capacidade de convergência para o equilíbrio de Nash ótimo em jogos repetitivos.
- **C8:** Mostramos experimentalmente a superioridade do algoritmo IVQ-Learning sobre suas versões baseadas nos paradigmas tradicionais em relação a sua capacidade de convergência para o equilíbrio de Nash ótimo em jogos estocásticos.
- **C9:** Mostramos experimentalmente, a capacidade de auto-organização do paradigma proposto, especificamente do algoritmo IVQ-Learning. Capacidade que não existe nos paradigmas tradicionais. Ainda é importante observar que esta capacidade do paradigma proposto vai possibilitar a junção das etapas de agendamento e execução de tarefas em sistemas multi-agente com capacidade de aprendizado.
- **C10:** Avaliamos a influência dos parâmetros próprios do paradigma *IVRL* na capacidade e tempo de convergência do algoritmo IVQ-Learning.
- **C11:** Avaliamos a influência da exploração de novas soluções versus a exploração do conhecimento prévio (*exploration* Vs. *explotation*) na capacidade e velocidade de convergência tanto dos paradigmas tradicionais como do nosso paradigma. Especificamente se trabalhou em base às versões modificadas do algoritmo Q-Learning.

Contribuições para o estado da arte (publicações)

Durante o desenvolvimento das pesquisas desta tese, em resposta aos questionamentos definidos na Seção 1.2, acima, pudemos efetivamente contribuir para a comunidade com publicações significativas em eventos científicos com visibilidade nacional e internacional assim como em capítulos em livros internacionais. A lista dessas publicações, que foram 13 ao todo, pode ser encontrada junto às referencias bibliográficas, ao final da tese [Barrios-Aranibar & Gonçalves 2009, Barrios-Aranibar et al. 2008, Barrios-Aranibar & Alsina 2007, Barrios-Aranibar & Gonçalves 2008, de Silva et al. 2008, Barrios-Aranibar & Gonçalves 2007*a*, Barrios-Aranibar & Gonçalves 2007*b*, Barrios-Aranibar & Gonçalves 2007*c*, Barrios-Aranibar & Gonçalves 2007*d*, Barrios-Aranibar & Alsina 2006, Barrios-Aranibar, Gurgel, Santos, Araujo, Roza, Nascimento, da Silva, Silva & Gonçalves 2006, Barrios-Aranibar, Gurgel, Burlamaqui, Gonçalves, Santos, Araujo, Roza & Nascimento 2006, Gurgel et al. 2006].

A Figura 1.1 mostra a relação existente entre as etapas de desenvolvimento da nossa tese e as contribuições (*Ci*), publicações (*Pi*) e questionamentos formulados (*Qi*). É importante observar que na primeira etapa da tese, trabalhamos no estudo do estado da arte e na análise da possibilidade de desenvolver algoritmos usando a teoria de Vigotsky. A segunda etapa foi onde se desenvolveu efetivamente nossa proposta e onde fizemos os tes-

tes iniciais. Já na etapa 3 e no final da nossa tese, foram feitos os testes de convergência e de aplicabilidade em problemas de colaboração. A lista de nossas publicações científicas conseguidas durante o doutorado inclui também outras que, embora não estejam diretamente relacionadas às contribuições desta tese, estão relacionadas à área de investigação (Aprendizado e Robótica) e que foram muito relevantes na nossa formação (foram quatro publicações não diretamente relacionadas com o trabalho, ao todo). Desse modo, temos 9 publicações resultantes diretamente desta tese.

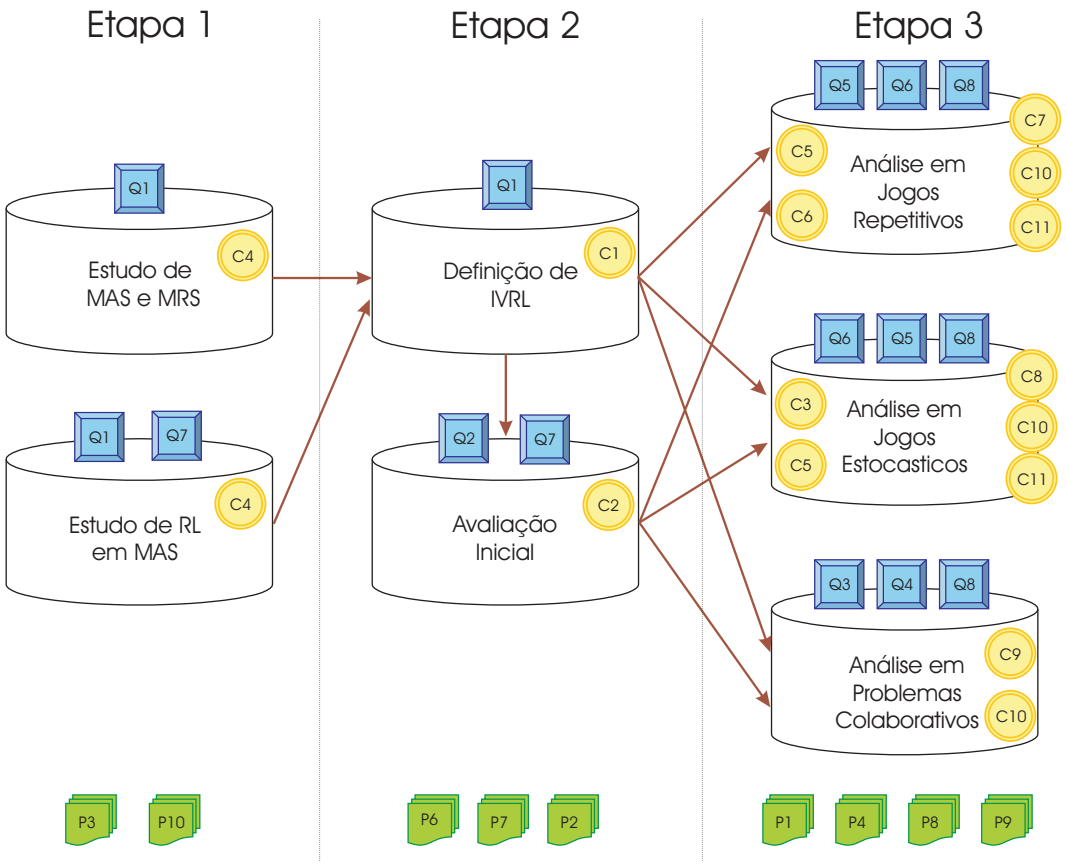


Figura 1.1: Etapas do desenvolvimento da tese de doutorado

1.3 Aplicações

O paradigma e os algoritmos propostos nesta tese, embora tenham sido testados apenas em problemas padrões de laboratório, podem ser usados em aplicações que envolvam problemas que podem ser resolvidos utilizando sistemas multi-agente cooperativos (coordenação e/ou colaboração). Merecem especial realce as aplicações em problemas onde sistemas multi-robô são necessários, uma vez que, ao se usar o paradigma proposto, a possibilidade de se ter agentes totalmente autônomos com uma boa capacidade de convergência a soluções ótimas é maior. Assim, a nossa proposta daria total autonomia a cada

indivíduo do time e a eventual perda de um elemento do mesmo poderia ser superada com facilidade. Como exemplo de tais aplicações, podemos citar o mapeamento coordenado de ambientes, uso de robôs em áreas de risco ao ser humano (escombros, minas, locais adversos), uso de um time de robôs como guias em museus, implementação de robôs humanóides tendo como premissa que cada uma das juntas do robô são implementadas como sendo um agente usando o paradigma IVRL, entre outras.

1.4 Organização do trabalho

O presente documento está divido em 6 capítulos. Após a introdução ao trabalho, que foi apresentada neste capítulo, o capítulo 2 explora o estado da arte, com especial ênfase no aprendizado em sistemas multi-agente. O capítulo 3 formaliza e propõe o paradigma de aprendizado em sistemas multi-agente: *aprendizado por reforço com valores de influência*. O capítulo 4 mostra as implementações realizadas no decorrer do desenvolvimento da tese de doutorado. Os resultados de testes realizados com estas implementações são analisados no capítulo 5. Finalmente, no capítulo 6, são esboçadas as conclusões e perspectivas desta tese.

Capítulo 2

Sistemas Multi-Agente

Várias técnicas computacionais são inspiradas pela realidade humana, desenvolvidas com o objetivo de modelar sistemas de forma eficiente. Os sistemas multi-agente são influenciados pela sociologia, com suas principais características funcionais definidas a partir de análises realizadas sobre as propriedades da sociedade e as propriedades dos indivíduos.

Podemos descrever uma sociedade como um grupo formado por indivíduos interagindo entre eles de tal forma que cada um busca cumprir seus objetivos traçados. A interação social ajuda a superar limitações tanto em termos físicos como em termos cognitivos dos indivíduos e a conformação de organizações e grupos facilita a solução de problemas [Mataric 1995]. Neste sentido, os sistemas multi-agente tem por objeto de estudo a coletividade onde o foco é a interação e organização dos indivíduos que no caso são chamados de agentes.

Devido à popularização dos sistemas multi-agente em diversas áreas, resulta em uma certa complexidade a tentativa de se dar uma definição universal de agente. Assim, Maes [Maes 1994] e Hendler [Hendler 1996] definem agentes inteligentes como programas de inteligência artificial cuja finalidade é atuar em diversos ambientes que são importantes para o ser humano. Os agentes são classificados em agentes físicos e agentes de informação. Por outro lado, Russel e Norvig definem agente como algo que pode perceber seu ambiente através de sensores e agir através de atuadores [Russel & Norvig 1995].

Já, Wooldridge e Jennigs distinguem dois conceitos: agentes segundo uma definição simples seriam considerados entes (hardware ou software) que tem propriedades de autonomia, sociabilidade, reatividade e pro-atividade. Segundo a definição maior, agentes seriam sistemas de computador que tem as propriedades anteriormente citadas. Ademais, podem ser implementados com características como aprendizagem, conhecimento, sentimentos, entre outras. [Wooldridge & Jennings 1995].

Similarmente à definição de agentes, sistema multi-agente (MAS) não tem uma definição universal. Segundo Hoe et.al. são definidos como um área da inteligência artificial que enfatiza o comportamento conjunto dos agentes, com algum grau de autonomia e com certa complexidade decorrente de suas interações [´t Hoen et al. 2006]. Já, Panait e Luke definem MAS como o ambiente onde existe mais de um agente, onde os agentes interagem uns com outros e além disso existem certas restrições no ambiente tais como que os agentes conhecem nada do ambiente que outros agentes conhecem [Panait & Luke 2005].

No presente trabalho, definimos um sistema multi-agente (MAS - *Multi-Agent System*)

como um software no qual dois ou mais agentes interagem num mesmo ambiente de forma a cooperarem e/ou competirem na solução de problemas. Os agentes devem ter capacidades básicas de percepção e de ação sobre o ambiente. Esses sistemas têm duas propriedades fundamentais: a autonomia dos agentes e sua organização.

O uso de sistemas multi-agente tornou-se comum na solução de problemas computacionais como por exemplo comércio eletrônico [Chen et al. 2008], agendamento de rotas de caminhões [Mes et al. 2007], modelagem da demanda de energia em cidades ou países [Toksari 2007], recuperação de imagens baseada em conteúdo [Dimitriadis et al. 2007], entre outros, e na solução de problemas envolvendo robôs como envio de cartas utilizando robôs [Carrascosa et al. 2008], missões de resgate de vítimas de desastres [Rooker & Birk 2005], geração de mapas de ambientes estruturados [Rocha et al. 2005] entre outros.

As interações em sistemas multi-agente podem ser de natureza cooperativa, competitiva ou mista, e é neste aspecto que os conceitos de inteligência artificial e teoria de jogos se misturam para poder explicar as mesmas, e encontrar formas de otimizá-las.

2.1 Teoria dos Jogos em MAS

A teoria dos jogos visa ao estudo da tomada de decisões em ambientes interativos, baseando-se na matemática, estatística investigação de operações, engenharia, biologia, economia, ciências políticas e outras disciplinas [Yeung & Petrosyan 2006].

Em teoria dos jogos, as interações entre agentes e as suas decisões são modeladas como *jogos estocásticos*. Na existência de um único agente, o jogo estocástico vira um *processo de decisão markoviano* e na presença de um único estado, ele se chamará *jogo repetitivo* [Shoham et al. 2007].

Os jogos estocásticos também são conhecidos como *jogos markovianos* e são classificados em *jogos de soma zero*, os quais envolvem dois agentes agindo de forma totalmente competitiva, e em *jogos de soma geral*, que incluem vários agentes cooperando e/ou competindo [Filar & Vrieze 1997]. Ainda, podem se diferenciar aqueles jogos onde a recompensa é a mesma para todos os agentes, chamados de jogos totalmente cooperativos [Panait & Luke 2005]. Assim, no presente trabalho, serão abordados unicamente os algoritmos de aprendizado aplicados a jogos estocásticos totalmente cooperativos.

Quando se trabalha com jogos estocásticos onde os agentes têm que prover cooperação, um outro conceito oriundo da teoria dos jogos se aplica em sistemas multi-agente , o *Equilíbrio de Nash*. É importante pensar que existem níveis de cooperação, assim, é desejável que os agentes cooperem em igualdade (e.g. que um agente não se aproveite do esforço dos outros e deixe de trabalhar na execução da sua tarefa) e especialmente que todos os agentes recebam ganhos equitativos ao resolverem a tarefa global (e.g. que o desejo de obter ganhos por parte de um agente não atrapalhe a obtenção de ganho dos outros agentes). É neste contexto que o conceito de *Equilíbrio de Nash* pode ser aplicado na avaliação do rendimento dos diferentes algoritmos utilizados em sistemas multi-agente cooperativos.

Seja um sistema multi-agente composto por N agentes. Define-se σ_i^* como a estratégia escolhida pelo agente i, σ_i como uma estratégia qualquer do agente i e Σ_i como o conjunto

de todas as possíveis estratégias de i. Diz-se que as estratégias $\sigma_1^*, ..., \sigma_N^*$ constituem um *equilíbrio de Nash* se a seguinte inequação é verdadeira, para todos os $\sigma_i \in \Sigma_i$ e para todos os agentes i:

$$r_i(\sigma_1^*, ..., \sigma_{i-1}^*, \sigma_i, \sigma_{i+1}^*, ..., \sigma_N^*) \leq r_i(\sigma_1^*, ..., \sigma_N^*) \tag{2.1}$$

onde r_i é a recompensa obtida pelo agente i.

Isto quer dizer que a idéia do *equilíbrio de Nash* é que a estratégia de cada agente seja a melhor resposta para as estratégias dos seus colegas e/ou adversários [Kononen 2004]. Neste ponto, é importante observar que este conceito não é aplicável só em sistemas cooperativos mas também é utilizado para sistemas competitivos que foi onde se originou [Nash 1950*a*, Nash 1950*b*, Nash 1953].

Então, em sistemas multi-agente cooperativos, é esperado que os algoritmos possam convergir para um *equilíbrio de Nash* e é desejado que possam convergir para o *equilíbrio de Nash ótimo* que é aquele equilíbrio onde a recompensa para os agentes é a melhor possível no jogo.

Neste sentido, muitos algoritmos de aprendizado de máquina são aplicados ao aprendizado em sistemas multi-agente cooperativos com o intuito de atingir um equilíbrio de Nash.

2.2 Cooperação em sistemas multi-agente

Cooperação, coordenação e colaboração são três termos indistintamente utilizados em sistemas multi-agente. Neste trabalho, usamos como base as definições propostas por Noreils [Noreils 1993], onde a cooperação acontece quando dois ou mais agentes interagem com o intuito de executar uma tarefa global. Já coordenação e colaboração são duas formas de cooperação [Botelho & Alami 2000].

Coordenação acontece quando os agentes coordenam suas ações um com o outro (sincronizam suas ações em relação às dos outros agentes), trocando informação, sinais, etc. Colaboração, por outro lado, ocorre quando os agentes dividem a tarefa global em sub-tarefas, sendo cada sub-tarefa executada por um agente específico.

Problemas que exigem coordenação são amplamente usados para testar algoritmos de aprendizado em sistemas multi-agente, porém o mesmo não acontece com problemas que exigem colaboração. Neste tipo de problema geralmente são aplicados algoritmos sem capacidade de aprendizado. Sendo os principais expoentes desta classe de algoritmos, os algoritmos inspirados biologicamente. Por exemplo, temos os algoritmos de enxames inteligentes (*swarm intelligence*) que se baseiam na idéia de que uma inteligência coletiva pode surgir das interações de um grande número de unidades simples [Beni 2005, He et al. 2006]. Estas interações são de natureza colaborativa posto que todos os indivíduos colaboram com um mesmo objetivo subdividindo as tarefas em sub-tarefas independentes. É importante observar que dentro destes sistemas também existem interações que exigem coordenação, porém a colaboração se dá em maior percentual. Esta abordagem inspira-se em muitas conclusões da inteligência coletiva em organismos biológicos, tais como peixes, aves, formigas, cupins, abelhas e micetozoários [Schmickl et al. 2009]. Sendo

a principal característica dos mesmos a sua capacidade de auto-organização [Bonabeau et al. 1999, Camazine et al. 2003].

Um exemplo de problema colaborativo muito usado em MAS, é o problema de busca de provisões (*Foraging*) [Hayat & Niazi 2005]. Muitos problemas computacionais podem ser vistos como um problema de busca de provisões; dentre eles podemos citar a busca de informação na web, roteamento numa rede, planejamento de caminhos e assim por diante. Por ser um problema colaborativo, as técnicas inspiradas biologicamente como enxames inteligentes e sistemas imunológicos são bastante aplicadas nele. Em geral, a idéia de aplicar qualquer algoritmo na solução de problemas colaborativos tem como objetivo a otimização do processo (e.g. fazer a tarefa no menor tempo possível usando a menor quantidade de recursos).

Neste sentido, Liu e Pasino mostraram que a solução de problemas colaborativos usando grupos de indivíduos tem vantagens em relação a fazê-lo sozinho [Liu & Passino 2004]. Especificamente, o problema de busca de provisões inspirou algoritmos que foram aplicados no planejamento dos movimentos para robôs humanoides [Mohamad et al. 2006], atribuição de tarefas em sistemas multi-robô [Zhang et al. 2007], desenvolvimento de motores de busca [Walker 2007] e muitos outros.

É importante observar que também existem aqueles problemas que precisam tanto de colaboração quando de coordenação, um exemplo simples deste tipo, seria um problema de busca por provisões onde as mesmas terão de ser carregadas por dois agentes ao mesmo tempo.

2.3 Aprendizado em Sistemas Multi-Agente

Em aprendizado de máquina, existem três tipos de aprendizado: O aprendizado supervisionado, o aprendizado não supervisionado e o aprendizado baseado em recompensas [Panait & Luke 2005]. Muitos destes algoritmos não são facilmente aplicáveis no aprendizado em sistemas multi-agente.

Algumas técnicas inspiradas em sistemas biológicos como, por exemplo as técnicas de *enxames inteligentes* e *sistemas imunológicos*, são também aplicadas a sistemas multi-agente devido a suas capacidades de auto-organização, descentralização e adaptação às mudanças do ambiente [Martinoli et al. 2004, Oliveira & Bazzan 2006, Lau & Wong 2007]. Porém, devido a estas técnicas não incluírem aprendizado nos agentes, elas serão desconsideradas na análise realizada na presente seção. É importante observar que alguns sistemas com aprendizado, desenvolvidos utilizando estas técnicas, são sistemas híbridos que misturam as mesmas com técnicas de aprendizado de máquina [Aardt & Marwala 2005, Zhang et al. 2007].

No caso de algoritmos de aprendizado supervisionado assume-se que é possível indicar a melhor resposta diretamente ao agente. Porém, no aprendizado em sistemas multi-agente, devido às interações dinâmicas e à mudanças no comportamento dos colegas e/ou adversários, torna-se muito difícil indicar a melhor resposta. Mesmo assim, podem ser encontrados alguns trabalhos que aplicam este tipo de algoritmo diretamente em sistemas multi-agente [Goldman & Rosenschein 1996, Wang & Gasser 2002, Ŝniezyński & Koźlak 2006]. Há também alguns sistemas híbridos que realizam a fusão com outros

tipos de aprendizado de máquina, como por exemplo, Ribeiro e colaboradores propõem um sistema híbrido que utiliza o algoritmo dos *k* vizinhos mais próximos com Q-learning [Ribeiro et al. 2006].

A idéia dos algoritmos baseados em recompensas não é indicar qual a melhor resposta mas sim indicar qual o resultado esperado, assim o agente terá de descobrir qual a melhor estratégia para obter o resultado desejado. Dentro deste tipo de algoritmos podemos encontrar os algoritmos evolutivos, os quais fazem uma busca no espaço de soluções com a finalidade de encontrar a que dê melhores resultados, por exemplo avaliando uma função de ajuste (*fitness*). Também encontramos os algoritmos de aprendizado por reforço, que calculam uma função de valor para predicados de estados ou para pares de estados-ações com a finalidade de definir uma política que melhor aproveite esses valores. A análise realizada neste capítulo priorizará os algoritmos baseados em recompensas e especialmente os algoritmos de aprendizado por reforço.

2.4 Aprendizado através recompensas em MAS

Os algoritmos de aprendizado por recompensa (ou por reforço) foram introduzidos inicialmente para aprendizado de um único agente. Eles são também aplicados em sistemas multi-agente segundo diferentes modos. A primeira maneira é modelar o time de agentes como um agente só, o que neste trabalho denominamos de *aprender como um time*. Um outro modo é aplicar os algoritmos sem nenhuma modificação, mas de forma independente para cada agente o que denominamos de *aprender independentemente*. A terceira e última forma, até o momento realizada, é a de *aprender ações conjuntas*. Isto quer dizer que cada agente aprende a executar uma ação pensando em combiná-la com as ações que os outros agentes irão executar. Estes três paradigmas são explicados a seguir.

2.4.1 Aprendendo como um time (TL)

O paradigma de aprendizado em sistemas multi-agente onde os agentes aprendem como um time baseia-se na modelagem do time todo como um agente único. A grande vantagem deste paradigma é que os algoritmos não precisam ser modificados. Porém, em aplicações reais como em robótica podem ser muito difíceis de implementar pelo fato de que eles precisam de ter todo o processamento e a informação centralizados.

Um exemplo deste paradigma usando aprendizado por reforço é o trabalho de Kok e Vlasis, que modelam o problema de colaboração em sistemas multi-agente como um processo de decisão markoviano [Kok & Vlassis 2004]. Neste e outros trabalhos similares o principal problema é o fato que a aplicabilidade dos mesmos torna-se inviável quando o número de agentes aumenta, já que o número de estados e ações também aumenta de forma exponencial.

Os algoritmos genéticos, por sua vez, não são muito afetados pelo crescimento exponencial no número de estados e ações [Sen & Sekaran 1996, Salustowicz et al. 1998, Iba 1999]. Porém, a necessidade de ter o processamento e a informação centralizados ainda é um problema impossível de ser resolvido neste paradigma, sendo uma de suas desvantagens.

Ainda, é importante observar que este paradigma foge do conceito de sistema multi-agente, já que, em sistemas multi-agente cada agente deve ser autônomo, não podendo existir um controle global dos mesmos [Shen & Wu 2008]. Fato pelo qual este paradigma não foi considerado nos experimentos realizados desta tese.

2.4.2 Aprendendo independentemente (IL)

Os problemas relatados no paradigma de aprendizado como um time podem ser resolvidos ao implementar o algoritmo de aprendizado de forma independente em cada agente. Vários trabalhos mostram resultados promissores ao aplicar este paradigma [Sen et al. 1994, Kapetanakis & Kudenko 2002, Tumer et al. 2002, Barrios-Aranibar & Alsina 2006].

A pergunta que logo aparece ao se descrever este paradigma é se os algoritmos tradicionalmente introduzidos para tratar problemas com um agente único terão resultados ótimos ou pelo menos condizentes com a solução do problema. Os trabalhos de Sen [Sen et al. 1994] e de Barrios-Aranibar [Barrios-Aranibar & Alsina 2006] mostram que estes algoritmos são aplicáveis em sistemas multi-agente. Claus e Bouitilier [Claus & Boutilier 1998] exploram o uso de aprendizes independentes em jogos repetitivos, mostrando empiricamente que a proposta têm condições de atingir um equilíbrio de Nash mas não o equilíbrio ótimo.

Estes resultados são importantes se os analisarmos no tocante à natureza dos algoritmos utilizados. Pode ser observado que os algoritmos de aprendizado por reforço visam levar o agente a executar um conjunto de ações que lhe proporcionem a maior utilidade (maiores recompensas). Segue que, em problemas envolvendo vários agentes, existe a possibilidade que a combinação de estratégias individuais ótimas não representem necessariamente uma estratégia do time também ótima.

Na tentativa de resolver este problema, diversos trabalhos foram desenvolvidos. Lão, aqui achei complicado mexer: mas não entendi o sentido de atacar, seria abordar negativamente?, etc. Estes trabalhos atacam a forma de calcular as recompensas para os agentes individualmente, já, ao paradigma trabalhar com independência dos agentes, não é possível incluir informações dos colegas. Assim, Um exemplo destas tentativas é o trabalho de Kapetanakis e Kudenko [Kapetanakis & Kudenko 2002] que propõem uma nova heurística para calcular os valores de recompensa para as ações baseada na freqüência com que uma ação teve sua máxima recompensa. Eles mostram, empiricamente, que a heurística tem capacidade de convergir para o equilíbrio de Nash ótimo em jogos repetitivos com dois agentes. Ainda no trabalho em questão, a heurística é testada com até quatro agentes em jogos repetitivos, onde só um deles utiliza a heurística de freqüência, mostrando que a capacidade de convergir para o equilíbrio ótimo aumenta, porém não é garantida [Kapetanakis & Kudenko 2004].

Um outro trabalho que explora modificações nas escolhas de recompensas é o de Tumer et. al. onde é abordado o problema de escolha das recompensas corretas em aprendizado independente [Tumer et al. 2002]. Os autores propõem um algoritmo que utiliza conceitos de inteligência coletiva para obter melhores resultados que os obtidos ao aplicar os algoritmos sem modificação alguma e que o de aprendizado como um time.

Um outro exemplo nesta linha é o trabalho de Mataric [Mataric 1997], o qual propõe uma metodologia que envolve minimização do espaço de aprendizado através do uso de comportamentos e condições, e tratando diretamente o problema de atribuição de recompensas usando funções de recompensa heterogêneas e estimadores de progresso.

Mesmo obtendo bons resultados em problemas simples como os jogos repetitivos ou jogos estocásticos com poucos agentes, um outro problema neste paradigma, que ocorre à medida que o número de agentes aumenta, é que os algoritmos tradicionais foram criados para casos em que o ambiente não muda, quer dizer para ambientes onde as recompensas são estáticas. Porém, em sistemas multi-agente, as recompensas podem mudar com o tempo, uma vez que as ações dos outros agentes irão influenciar nas mesmas.

2.4.3 Aprendendo ações conjuntas (JAL)

Uma das formas de resolver o problema do paradigma anterior é aprender a melhor resposta às ações dos outros agentes. Neste contexto, aparece o paradigma de aprender ações conjuntas, onde cada agente deverá aprender qual o melhor valor ao executar suas ações em combinação com as ações dos outros (ação conjunta). Ao intuir um modelo dos outros agentes, ele deverá calcular qual a melhor ação às supostas ações a serem executadas pelos colegas e/ou adversários [Kapetanakis et al. 2003, Chalkiadakis & Boutilier 2003, Guo et al. 2007].

Claus e Bouitilier exploram o uso deste paradigma em jogos repetitivos [Claus & Boutilier 1998] mostrando empiricamente que a forma básica do algoritmo converge para um equilíbrio de Nash mas não garante a convergência para o equilíbrio ótimo. Porém, os autores indicam que, diferentemente dos algoritmos aplicados para aprender de forma independente, este paradigma pode ser melhorado se forem melhorados os modelos dos outros agentes.

Outros exemplos de algoritmos baseados neste paradigma incluem o trabalho de Suematsu e Hayashi que propõem um algoritmo para aprender ações conjuntas que garante a convergência para um equilíbrio de Nash [Suematsu & Hayashi 2002]. O trabalho de Banerjee e Sen [Banerjee & Sen 2007] propõe um algoritmo de aprendizado de ações conjuntas Condicional, no qual os agentes aprendem a probabilidade condicional de uma ação executada pelo oponente ser ótima, dadas as suas próprias ações e usa esta probabilidade para escolher suas ações futuras.

Outra forma encontrada deste paradigma é o aprendizado implícito de ações conjuntas, proposto por Lauer e Riedmiller [Lauer & Riedmiller 2004]. Nesta proposta eles armazenam os valores Q das ações conjuntas, porém eles não as definem explicitamente como nos trabalhos anteriores.

O principal problema deste paradigma é dado pela quantidade de combinações de estados e ações que cresce exponencialmente a medida que número de estados, ações e/ou agentes cresce.

Tabela 2.1: Contextualizando o paradigma IVRL

	TL	IL	JAL	IVRL
Processamento	Centralizado	Distribuido	Distribuido	Distribuido
Informação do Colega?	—	Não	Sim	Sim
Comunicação?	—	Não	Sim	Sim
Auto-organização?	—	Não	Não	Sim
Ambiente	Estático	Estático	Estático	Dinâmico
Exige sincronia?	Sim	Não	Sim	Não
Ações do Colega	—	—	Reais	Observadas
Mais agentes				
Ações aumentam?	Exponencial	Não	Exponencial	Não
Armazenamento Cresce?	Exponencial	Não	Exponencial	Linear
Processamento Cresce?	Exponencial	Não	Exponencial	Linear
Mais ações				
Armazenamento Cresce?	Exponencial	Linear	Exponencial	Linear
Processamento Cresce?	Exponencial	Linear	Exponencial	Linear

2.5 Contextualizando o IVRL

O paradigma de aprendizado que propomos nesta tese de doutorado baseia-se na idéia de que cada agente pode mudar seu comportamento em função da opinião dos outros. Já que esta proposta é desenvolvida no contexto do aprendizado através de recompensas, então o valor das soluções propostas será dado pela recompensa individual de cada agente e pela opinião que os outros agentes têm sobre a solução que o agente deu individualmente. Esta opinião deverá ser uma função da recompensa obtida pelos agentes. Isto é, se um agente prejudica os outros agentes ao fazer com que eles obtenham uma recompensa menor às recebidas anteriormente eles terão uma opinião negativa de suas ações.

Este trabalho é considerado um novo paradigma, uma vez que não pode ser classificado como sendo o paradigma de aprendizado em time, porque nele cada agente terá seu próprio processo de aprendizado. Também não pode ser considerado aprendizado independente, porque, ao se considerar qualquer informação vinda dos outros agentes, os algoritmos deixam de ser independentes. Finalmente não pode ser considerado como parte do paradigma de aprendizado de ações conjuntas porque os agentes aprenderão simplesmente o valor de suas ações. Aliás, eles nem precisam ter certeza de que o que eles acham que os outros agentes fizeram é exatamente o que eles fizeram. Eles simplesmente têm que opinar se gostaram ou não daquilo que aconteceu externo a eles.

Também é importante observar que este paradigma não necessariamente deve ser implementado como um algoritmo de aprendizado por reforço, como fazemos neste trabalho, mas também pode se utilizar algoritmos genéticos ou outros algoritmos baseados em recompensas.

Características de todos os algoritmos citados podem ser observadas na Tabela 2.1. Podemos observar que o paradigma de aprendizado como um time exige processamento centralizado e os outros dois paradigmas tradicionais (IL e JAL), junto com o nosso

(IVRL) podem ser implementados de forma completamente distribuída. Em relação à necessidade de um agente requerer informações sobre os outros agentes, notamos que isto não é aplicável a aprendizado como um time já que este implementa um único agente. Já no caso do aprendizado independente, vemos que não é necessária nenhuma informação do colega e que no aprendizado de ações conjuntas e no nosso paradigma, esta informação é necessária e relevante para o bom funcionamento dos mesmos. O fato anterior está diretamente relacionado à necessidade de comunicação dos agentes, como mostrado na mesma Tabela. Emquanto no JAL os agentes precisam se comunicar para informar suas ações aos colegas, no IVRL precisam se comunicar para comunicar sua opinião.

Uma outra propriedade da nossa proposta é a sua capacidade de auto-organização, a qual não existe nos paradigmas tradicionais. Esta capacidade foi comprovada experimentalmente, porém se fizermos uma análise, podemos observar nos nossos algoritmos um comportamento parecido ao dos algoritmos de enxames inteligentes, já que ambos os casos o comportamento dos outros agentes se vê influenciado pelo comportamento e decisões de cada agente. Nos enxames inteligentes, especificamente nos algoritmos baseados em colônias de formigas, por exemplo, isto é implementado com o conceito de feromônio e nos nossos algoritmos é implementado usando o conceito de opinião. Esta nova característica inserida nos algoritmos de aprendizado vai lhes dar a possibilidade de aplicação direta em problemas colaborativos eliminando a etapa de alocação de tarefas.

Outro aspecto importante é que nos paradigmas tradicionais trabalhava-se a idéia de que o ambiente é estático e não muda com o tempo. Porém, ao existirem vários agentes interagindo no mesmo ambiente, o último se torna naturalmente dinâmico. Este fato se reflete nas recompensas que cada agente pode receber, já que para uma mesma ação, num determinado estado, o agente nem sempre vai receber a mesma recompensa. A recompensa recebida depende das ações dos colegas. Já no nosso paradigma, é aceito que o ambiente é dinâmico ao estabelecer que os outros agentes podem ter participação positiva ou negativa nas recompensas que cada agente recebe.

É importante o fato das propostas TL e JAL exigirem sincronia no funcionamento dos agentes, o que não é necessário na nossa proposta juntamente com o IL. Além disso, já que o JAL e o IVRL trabalham baseados na informação dos colegas, especificamente observando o ‘"agir"dos mesmos, é importante notar que na nossa proposta não é relevante saber realmente o que os colegas fizeram em quanto no JAL é relevante. Este conhecimento não é relevante posto que pode se trabalhar em base a ações observadas e não reais, discretizando Lão, aqui tbem: é de discreto? Ou descrer? as ações dos colegas em função da discretização que cada agente faz do mundo real.

Finalmente, comparando o tempo de processamento e a capacidade de armazenamento necessária para cada método, vemos que, ao aumentar o número de agentes, o número de ações processadas por cada agente aumenta exponencialmente para os paradigmas TL e JAL, enquanto que para o IL e nossa proposta as ações para cada agente não aumentam. Já em relação às necessidades de processamento e armazenamento, vemos que nos paradigmas TL e JAL esta necessidade cresce exponencialmente enquanto que para o IL estas necessidades não crescem e para nosso paradigma o crescimento é linear. Porém, ao aumentar o número de ações por agente real, o crescimento em necessidades de armazenamento e processamento também cresce exponencialmente para os paradigmas TL e JAl

enquanto que para nosso paradigma e o IL o crescimento é linear.

Capítulo 3

Aprendendo através da opinião de outros agentes

Neste capítulo, apresentamos o paradigma de aprendizado através de recompensas com valores de influência. Este paradigma será explicado a partir de uma visão dos algoritmos de aprendizado por reforço. Porém, deixamos claro que o mesmo poderia ser implementado também utilizando outros tipos de algoritmos baseados em recompensas.

Aprendizado é um conceito que pode variar dependendo do contexto. Então, nosso primeiro passo é definir o conceito de aprendizado através de recompensas para sistemas multi-agentes utilizado no presente trabalho.

Definição 3.1 *Considera-se que um conjunto de N agentes aprenderam se pelo menos uma das seguintes inequações (Equações 3.1 e 3.2) é verdadeira para pelo menos um dos agentes i dentro do sistema:*

$$r_i(\sigma_1^*(t+T),...,\sigma_N^*(t+T)) > r_i(\sigma_1^*(t),...,\sigma_N^*(t)) \tag{3.1}$$

$$size(\sigma_i^*(t+T)) < size(\sigma_i^*(t)) \tag{3.2}$$

onde T é o tempo de aprendizado, t é um instante de tempo qualquer dentro da vida do sistema, $\sigma_i^*(t)$ *é a estratégia escolhida pelo agente i no tempo t,* r_i *é o retorno obtido pelo agente i e size calcula o tamanho de uma estratégia em função do número de ações que a compõem.*

Da Definição 3.1, podemos concluir que, para poder-se afirmar que um sistema multi-agente aprendeu, a recompensa dos indivíduos que o compõem deverá aumentar após o processo de aprendizado, ou então o tempo que os mesmos levam para executar suas tarefas deve ser menor que o anterior.

3.1 Formalização do problema

Aprendizado por Reforço é aprender o que fazer–como mapear situações em ações– com o intuito de maximizar um sinal numérico de recompensa. O aprendiz não recebe informações sobre quais ações deve executar, como a maioria de formas de aprendizado

de máquina, pelo contrário, ele deve descobrir quais ações levam a um ganho maior pela experimentação das mesmas [Suton & Barto 1998].

Uma tarefa de aprendizado por reforço é implementada usando um processo de decisão Markoviano (*MDP - Markov Decission Process*). Isto significa que os algoritmos de aprendizado por reforço foram criados para serem aplicados a esse tipo de processo. A figura 3.1 mostra as interações de um agente e o ambiente em um MDP. Como mostrado na figura, o ambiente pode ser modelado como uma máquina de estados finitos estocástica com entradas e saídas. Como entradas, temos as ações do agente e, como saídas, temos as observações e recompensas enviadas ao agente. Assim, as seguintes três funções modelam o ambiente [Murphy 2000]:

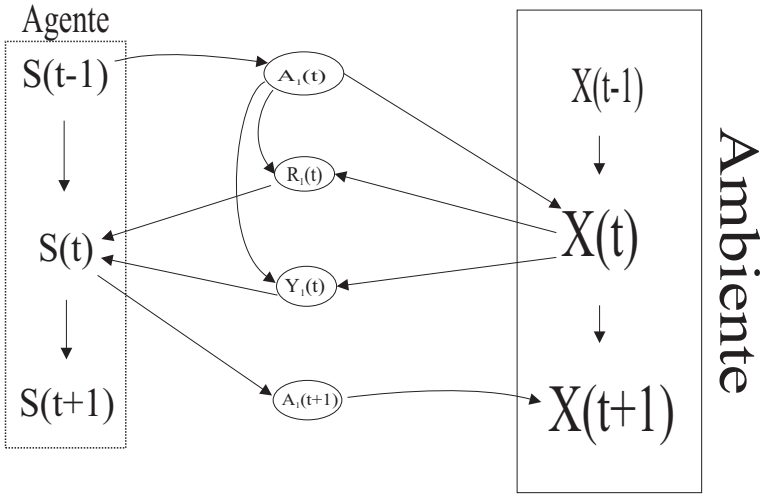


Figura 3.1: Interação de um Agente com o Ambiente num MDP
[Murphy 2000]

- Função de transição de estados: modela qual a probabilidade de chegar num estado $X(t)$ partindo de um estado $X(t-1)$, sabendo que o agente executou a ação $A(t)$. Isto é $P(X(t)|X(t-1),A(t))$.
- Função de observação (saída): modela a observação $Y(t)$ que pode ser realizada do ambiente como um processo ativo. É calculada a probabilidade de observar $Y(t)$ sabendo que $X(t)$ é o estado atual do sistema e que o agente executou a ação $A(t)$ para chegar nesse estado. Isto é $P(Y(t)|X(t),A(t))$. No caso especial desta função, se $Y(t) \neq X(t)$ temos que o mundo é parcialmente observável e o MDP se converte num POMDP (processo de decisão markoviano parcialmente observável).
- Função de recompensa: Que modela o valor numérico esperado como recompensa $R(t)$, sabendo que $X(t)$ é o estado atual do sistema e que o agente executou a ação $A(t)$ para chegar nesse estado. Isto é $E(R(t)|X(t),A(t))$.

Neste paradigma de aprendizado, um agente também pode ser modelado como uma máquina de estados finitos estocástica, onde as entradas são as observações e recompensas enviadas pelo ambiente e as saídas são as ações realizadas sobre o ambiente. Assim, as seguintes duas funções modelam o agente [Murphy 2000]:

- Função de transição de estados: modela o estado interno do agente $S(t)$ como sendo uma função do estado interno anterior $S(t-1)$, da observação do estado do mundo $Y(t)$, da recompensa obtida pelo agente $R(t)$ e da ação $A(t)$ executada pelo mesmo quando estava no estado interno $S(t-1)$. Isto é $f(S(t-1),Y(t),R(t),A(t))$.
- Função de seleção de ações: modela a ação $A(t+1)$ a ser escolhida pelo agente conhecendo o estado interno do mesmo ($S(t)$). Isto é $A(t) = \pi(S(t))$.

Para poder aplicar aprendizado por reforço a um determinado problema, o mesmo deve ser modelado usando as funções anteriores (e.g. como um MDP). Este fato representa um problema quando se trabalha com problemas no escopo de sistemas multi-agentes, já que os mesmos são geralmente modelados como jogos estocásticos (*SG - Stochastic Games*) e são generalizações dos MDPs. Para entender melhor o problema relatado remetemo-nos à Figura 3.2 a qual mostra as interações dos agentes com o ambiente num SG.

Analisando a Figura, podemos ver que as funções que modelam o ambiente e os agentes são inseridas como máquinas de estados finitos estocásticas, com novos parâmetros e novas restrições. Assim, temos as seguintes versões das funções que modelam o ambiente para este tipo de problema:

- Função de transição de estados: modela qual a probabilidade de chegar num estado $X(t)$ partindo de um estado $X(t-1)$, sabendo que os N agentes dentro do sistema executaram as ações $A_1(t),A_2(t),...,A_n(t)$. Isto significa que a definição desta função muda para $P(X(t)|X(t-1),A_1(t),A_2(t),...,A_n(t))$.
- Função de observação (saída): modela a observação $Y_i(t)$ que pode ser realizada do ambiente por um agente i como um processo ativo. O que significa que temos N funções de observação. Nas mesmas, é calculada a probabilidade de observar $Y_i(t)$ sabendo que $X(t)$ é o estado atual do sistema e que os N agentes dentro do sistema executaram as ações $A_1(t),A_2(t),...,A_n(t)$. Assim esta função definida para o agente i é $P(Y_i(t)|X(t),A_1(t),A_2(t),...,A_n(t))$. Embora esta função esteja definida com os mesmos parâmetros para todos os agentes, é importante esclarecer que as observações realizadas por cada agente não necessariamente são as mesmas que realizaram seus colegas, acrescendo maior complexidade a estes sistemas.
- Função de recompensa: Que modela o valor numérico esperado pelo agente i como recompensa $R_i(t)$, sabendo que $X(t)$ é o estado atual do sistema e que os N agentes dentro do sistema executaram as ações $A_1(t),A_2(t),...,A_n(t)$ para chegar nesse estado. Isto é $E(R_i(t)|X(t),A_1(t),A_2(t),...,A_n(t))$. Nesta função ocorre o mesmo que na função de observação. As recompensas não são necessariamente as mesmas para os agentes dentro do sistema, porém, no caso de serem a mesma, remete a ambientes totalmente cooperativos.

No caso das duas funções que modelam o ambiente observamos as seguintes mudanças:

- Função de transição de estados: modela o estado interno do agente i dentro do sistema $S_i(t)$ como sendo uma função do estado interno anterior $S_i(t-1)$, da observação feita por ele do estado do mundo $Y_i(t)$, da recompensa obtida pelo agente $R_i(t)$

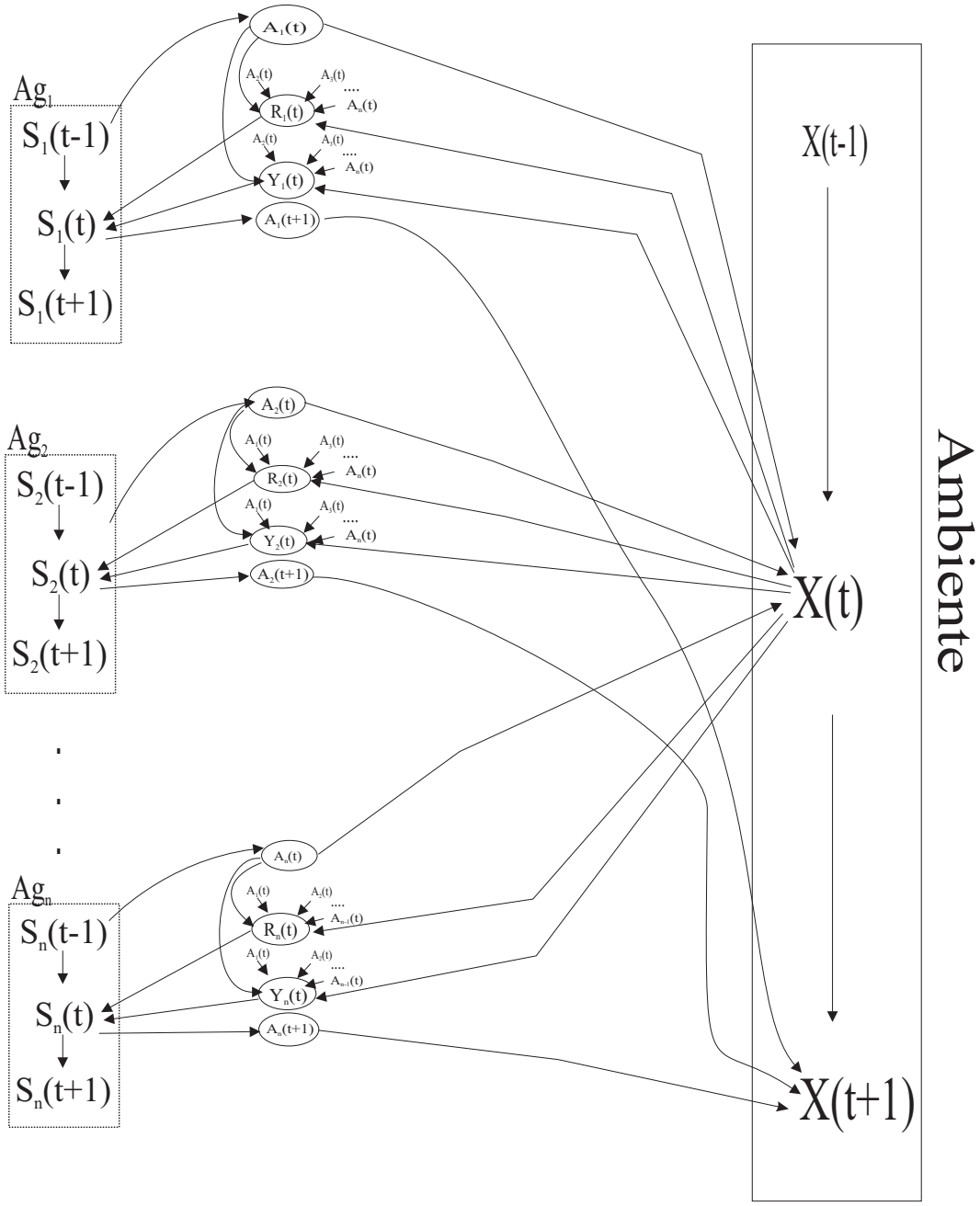


Figura 3.2: Interação de N Agentes com o Ambiente em SG

e da ação $A_i(t)$ executada pelo mesmo quando estava no estado interno $S_i(t-1)$. Isto é $f(S_i(t-1), Y_i(t), R_i(t), A_i(t))$.

- Função de seleção de ações: modela a ação $A_i(t+1)$ a ser escolhida pelo agente conhecendo o estado interno do mesmo ($S_i(t)$). Isto é $A_i(t) = \pi(S_i(t))$.

Se observarmos as funções anteriores, podemos verificar que processos de decisão markovianos são diferentes de jogos estocásticos e que existe uma defasagem entre o modelo do ambiente e o modelo do agente quando aplicamos aprendizado por reforço em jogos estocásticos (SG). Por exemplo, embora as observações e interações no ambiente baseiem-se nas ações executadas por todos os agentes dentro do sistema, o agente, internamente, trabalha com base no seu próprio estado interno e nas suas próprias ações. Então, um dos desafios é justamente incluir as novas restrições e características inseridas ao modelo nos algoritmos de aprendizado por reforço. É importante também notar que o paradigma de aprendizado de ações conjuntas já inclui mudanças ao modelo para poder se ajustar às novas restrições. Por exemplo, nesse paradigma, a função de transição de estados para o agente é função das ações de todos os agentes dentro do sistema ($f(S_i(t-1), Y_i(t), R_i(t), A_1(t), A_2(t), ..., A_n(t))$), já que ele aprende o valor das ações conjuntas e não, unicamente, o valor da sua ação, como é feito no aprendizado por reforço tradicional.

Uma especialização muito importante dos jogos estocásticos são os jogos repetitivos (*RG - Repetitive Games*). Este tipo de jogo se tornou importante por representar um espaço simplificado onde as técnicas de aprendizado por reforço propostas para sistemas multi-agentes podem ser testadas e avaliadas. Sua simplicidade pode ser observada na Figura 3.3, que mostra como são as interações dos agentes com o ambiente. As versões das funções que modelam o ambiente para este tipo de problema são as seguintes:

- Função de transição de estados: não existe porque neste tipo de problema considera-se que o mundo tem um único estado.
- Função de observação (saída): modela a observação Y_i que pode ser realizada do ambiente por um agente i. Esta observação não é mais um processo ativo, pois não depende das ações dos agentes. Isto significa que temos N funções de observação. Nas mesmas, é calculada a probabilidade de observar Y_i sabendo que X é o estado permanente do sistema. Assim esta função, definida para o agente i, é $P(Y_i|X)$.
- Função de recompensa: modela o valor numérico esperado pelo agente i como recompensa R_i, sabendo que X é o estado do sistema e que os N agentes dentro do sistema executaram as ações $A_1, A_2, ..., A_n$. Isto é $E(R_i|X, A_1, A_2, ..., A_n)$. Nesta função, ocorre o mesmo que na função de observação. As recompensas não necessariamente são as mesmas para os agentes dentro do sistema, e, no caso de serem a mesma, referimo-nos a ambientes repetitivos totalmente cooperativos.

No caso das duas funções que modelam o ambiente, observamos as seguintes mudanças:

- Função de definição de estado: muda de nome porque neste tipo de problemas existe um único estado. Então, agora, modela-se o estado interno do agente i dentro do

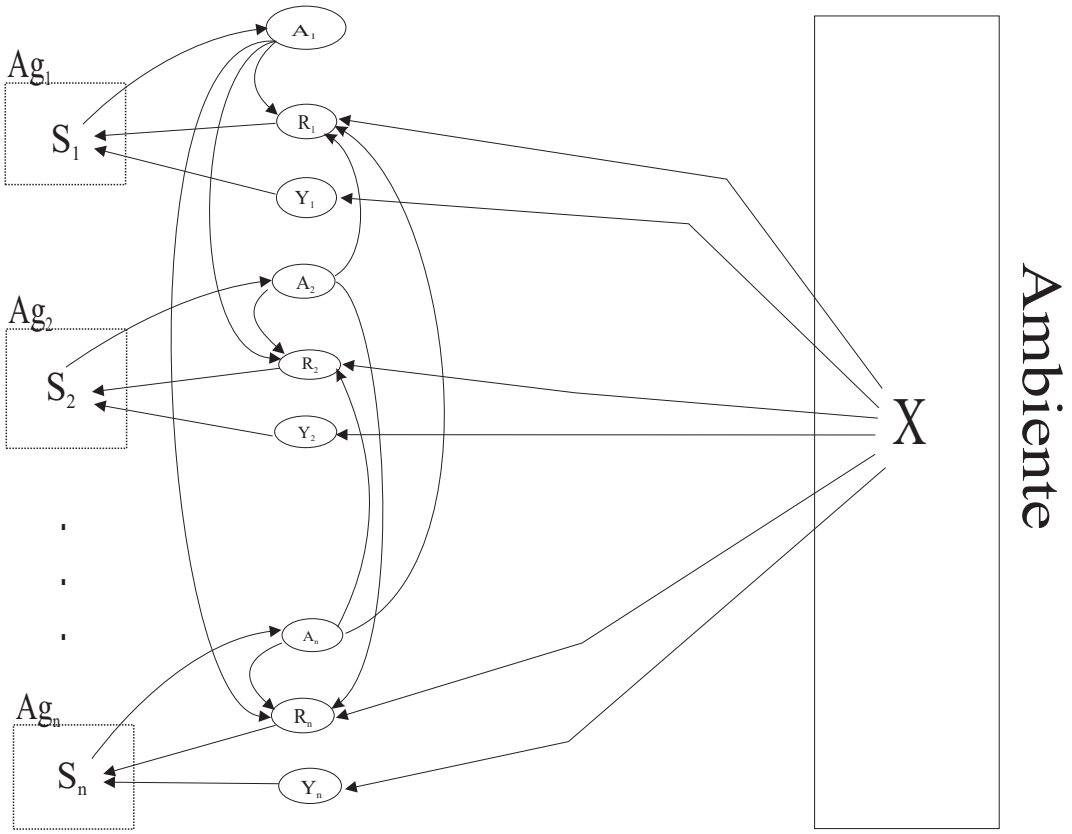


Figura 3.3: Interação de N Agentes com o Ambiente em RG

sistema S_i como sendo uma função da observação feita por ele do estado do mundo Y_i, da recompensa obtida pelo agente R_i e da ação A_i executada pelo mesmo, isto é, $f(S_i, Y_i, R_i, A_i)$.

- Função de seleção de ações: modela a ação A_i a ser escolhida pelo agente conhecendo o estado interno do mesmo (S_i). Isto é $A_i = \pi(S_i)$.

Por fim, de posse dos conceitos acima abordados, podemos enunciar formalmente o problema a ser abordado nesta tese. Buscamos justamente a definição de elementos que permitam eliminar a defasagem existente entre o modelo do ambiente e o modelo do agente quando aplicamos aprendizado por reforço em jogos estocásticos. Ainda, é importante observar que nosso principal objetivo é manter a simplicidade do modelo do agente no tocante a manter a função de transição de estados no agente como sendo $f(S_i(t-1), Y_i(t), R_i(t), A_i(t))$. E não complicá-la, tal como no paradigma de aprendizado de ações conjuntas.

3.2 Aprendizado por reforço com valores de influência

Neste trabalho buscamos converter o paradigma proposto numa solução alternativa ao problema relatado na seção anterior. Nosso modelo é inspirado nas interações sociais

que as pessoas têm em uma comunidade. Algumas teorias a respeito da interação social podem ser verificadas em trabalhos teóricos da área de educação e psicologia, como, por exemplo, o trabalho de Levi Vygotsky [Oliveira & Bazzan 2006, Jars et al. 2004]. Baseados nesses estudos preliminares sobre o processo de interações sociais dentro do aprendizado, conjeturamos que, quando as pessoas interagem, elas comunicam, umas às outras, o que acham das ações das outras, seja através de crítica direta ou de elogios. Isto significa que, se uma pessoa *B* não gosta da ação executada pela pessoa *A*, então *B* irá protestar contra *A*. Se a pessoa *A* continuar a realizar a mesma ação, então *B* pode se tornar irritado e protestar bravamente. Note-se que isto significa que a força do protesto pode ser considerada como sendo proporcional ao número de vezes que a ação que o pode originar é repetida.

Por outro lado, se a pessoa *B* gostar da ação executada pela pessoa *A*, então *B* deve elogiar *A*. Mais ainda, se a ação realizada for considerada como muito boa, então *B* deve elogiar a *A* com veemência. Mas, do mesmo modo, se *A* continua a realizar a mesma ação, então *B* ficará habituada, e, com o passar do tempo, ela deixará de elogiar *A*. Isto significa que a força dos elogios pode ser considerada como sendo inversamente proporcional ao número de vezes que a ação que o originou é repetida. A Figura 3.4 representa um exemplo das interações sociais relatadas, pela perspectiva de que quem está agindo é o agente A.

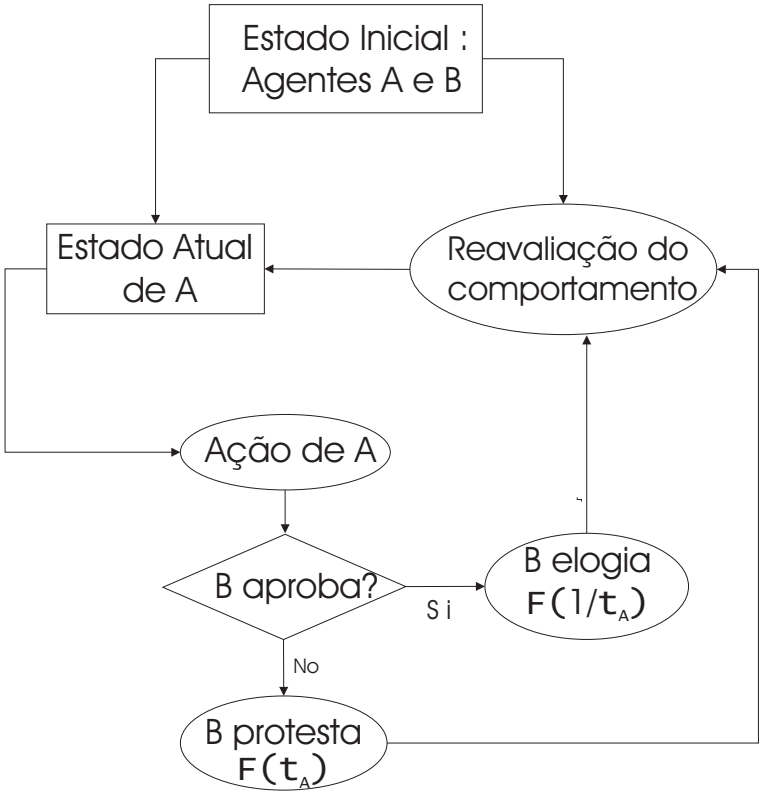


Figura 3.4: Interação Social segundo o agente A

Um outro aspecto muito importante é que os protestos e elogios de outras pessoas po-

dem *influenciar* o comportamento de uma pessoa. Nesse sentido, quando outras pessoas protestam, uma pessoa tentará evitar as ações que causaram esses protestos e, quando outras pessoas fazem elogios, a pessoa tentará repetir estas ações mais vezes.

Inspirados neste fato, propomos a criação de um novo paradigma de aprendizado através de recompensas com valores de influência, em termos do aprendizado por reforço, que denominamos de "aprendizado por reforço com valores de influência" (IVRL - *Influence Value Reinforcement Learning*). Neste paradigma, os agentes calculam os valores das suas ações baseados no retorno local obtido e em um valor numérico chamado de *valor de influência*.

3.2.1 O Modelo

Com o intuito de resolver o problema relatado anteriormente em relação à aplicação de algoritmos de aprendizado por reforço em sistemas multi-agente, modificamos o modelo das interações entre os agentes e o ambiente num SG. Estas interações são mostradas na Figura 3.5.

Como pode ser observado na Figura 3.5, o modelo do ambiente se mantém sem alterações. Porém, o modelo do agente muda, com as cinco funções a seguir passando a constituí-lo:

- Função de observação: modela a observação $Ao_{ij}(t)$ que pode ser realizada por um agente i, da ação $A_j(t)$ executada por um agente j. Esta função, definida para o agente i, é $P(Ao_{ij}(t)|A_j(t))$. É importante observar que a observação realizada pelo agente i é uma figura abstrata da ação realmente executada pelo agente j e depende muito da sua forma de modelar o mundo.
- Função de opinião: modela a opinião $Op_{ij}(t)$ que um agente i poderá ter da ação observada por um agente j ($Ao_{ij}(t)$), como sendo uma função do estado interno $S_i(t)$, da ação observada $Ao_{ij}(t)$, da ação $A_i(t)$ executada pelo próprio agente e da recompensa $R_i(t)$ obtida pelo agente. Estes dois últimos elementos são encapsulados no elemento conhecido como valor de referência ($Rv_i(t)$). Assim, esta função é definida como sendo $Op_{ij}(t) = f(S_i(t), R_i(t), A_i(t), Ao_{ij}(t))$.
- Função de Influência: modela o quanto um agente i é influenciado pelas opiniões dos outros agentes dentro do sistema ($IV_i(t)$). Este valor é calculado em função das opiniões ($Op_{ji}(t)\ \forall j \in \{1..n\} - i$) que os outros agentes têm sobre a ação $A_i(t)$ executada pelo agente, sabendo que a mesma o levou ao estado $S_i(t)$. Assim $IV_i = f(S_i(t), A_i(t), Op_{ji}(t)\ \forall j \in \{1..n\} - i)$
- Função de transição de estados: modela o estado interno do agente i dentro do sistema $S_i(t)$ como sendo uma função do estado interno anterior $S_i(t-1)$, da observação feita por ele do estado do mundo $Y_i(t)$, da recompensa obtida pelo agente $R_i(t)$, da ação $A_i(t)$ executada pelo mesmo quando estava no estado interno $S_i(t-1)$ e da influência $IV_i(t)$ que os outros agentes exercem sobre ele, isto é, $f(S_i(t-1), Y_i(t), R_i(t), A_i(t), IV_i(t))$.
- Função de seleção de ações: modela a ação $A_i(t+1)$ a ser escolhida pelo agente, conhecendo-se o estado interno do mesmo ($S_i(t)$), isto é, $A_i(t) = \pi(S_i(t))$.

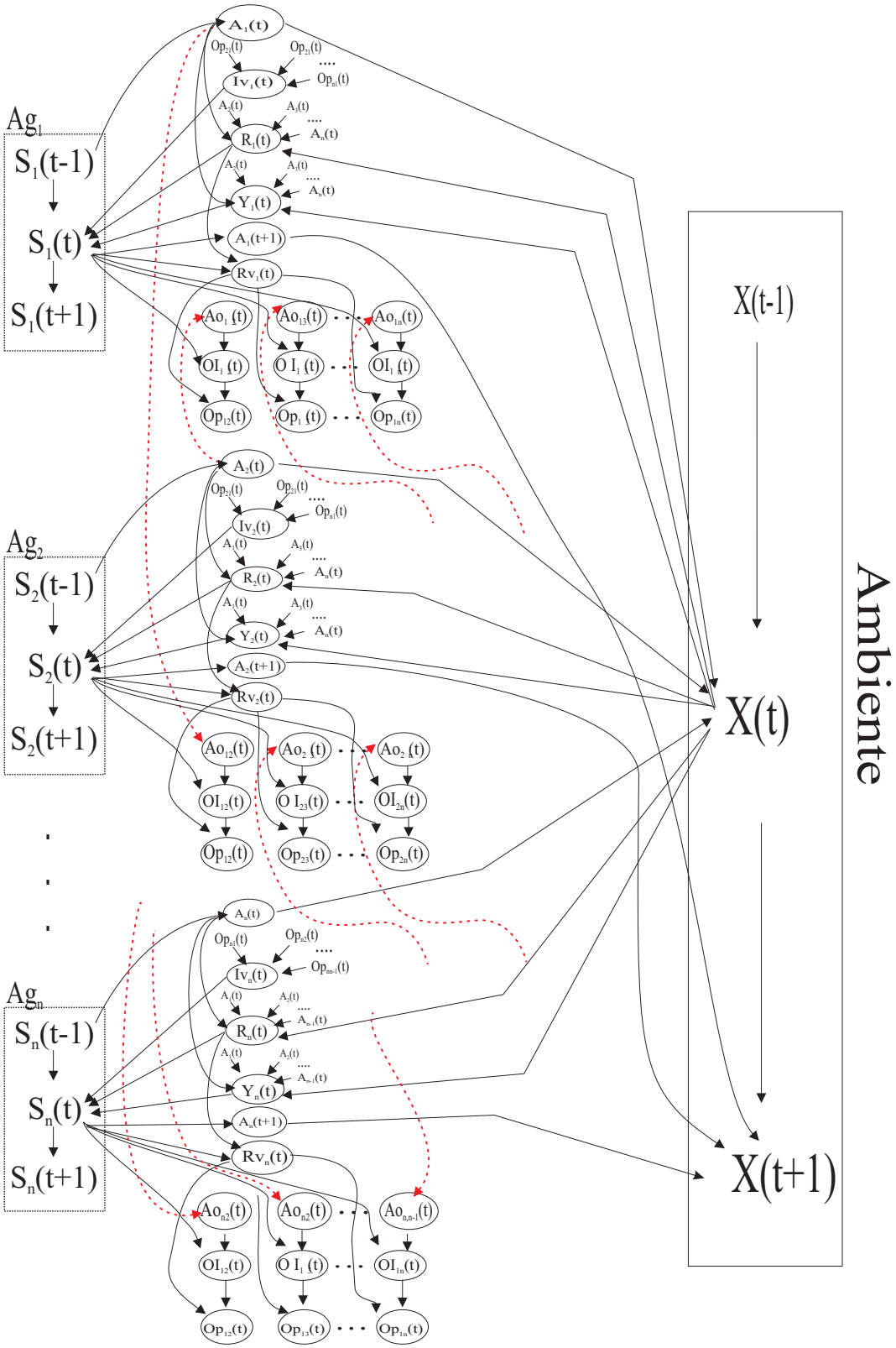


Figura 3.5: Aprendizado por Reforço com Valores de Influência

Neste novo modelo, podemos observar que a simplicidade da função de transição das de estados mantém-se para o agente, porém, já é considerado o fato que as ações dos colegas têm efeito direto nas recompensas do agente e na constituição do estado interno do mesmo.

3.2.2 O Paradigma

O modelo anterior é a base para a implementação do paradigma de aprendizado com valores de influência. A seguir, definimos os conceitos e equações necessárias para a construção de algoritmos baseados neste paradigma.

Definição 3.2 *O valor de influência para um agente i dentro de um grupo de N agentes é um valor numérico com a capacidade de alterar positiva ou negativamente as crenças que um agente possui de um determinado estado ou um par estado/ação. Este valor é definido pela seguinte equação:*

$$IV_i = \sum_{j \in \{1:N\}-i} \beta_i(j) * OP_j(i) \tag{3.3}$$

onde $\beta_i(j)$ é o coeficiente de influência do agente j sobre o agente i, $OP_j(i)$ é a opinião do agente j em relação às ações executadas pelo agente i.

O coeficiente de influência (β) determinará se um agente será ou não influenciado pela opinião de outros agentes. Já *OP* é um valor numérico que é calculado em função das recompensas que o agente obteve e das crenças que ele tem sobre os seus estados ou pares estado/ação. Devido a que, em aprendizado por reforço, o valor dos predicados de estado ou de um par estado-ação está diretamente relacionado às recompensas obtidas no passado, então a opinião de um agente será calculada em função deste valor e da recompensa obtida no momento da avaliação.

Definição 3.3 *A opinião de um agente j em relação às ações de um agente i, é definida como sendo um valor numérico da qualidade da solução oferecida pelo agente i do ponto de vista do agente j. Esta opinião é definida em função da avaliação do próprio agente sobre as recompensas obtidas em relação às crenças que ele tinha do mundo. Se esta avaliação é positiva, então a opinião (também positiva) irá decrescendo enquanto a ação que a gerou é repetida tendo como limite o valor zero. Por outro lado, se a avaliação é negativa, a opinião (também negativa) irá crescendo até um valor chamado valor de referência. A opinião é definida pela Equação 3.4:*

$$OP_j(i) = \begin{cases} RV_j * OI(s_j(t), a_i(t)) & \text{Se } RV_j < 0 \\ RV_j * (1 - OI(s_j(t), a_i(t))) & \text{Se } RV_j > 0 \\ 0 & \text{caso contrario} \end{cases} \tag{3.4}$$

onde RV_j representa a referência base das opiniões do agente e é definida por

$$RV_j = r_j(t+1) + V(s_j(t+1)) - V(s_j(t)) \tag{3.5}$$

para o caso de estar aprendendo os valores dos predicados de estados, e

$$RV_j = r_j(t+1) + \max_{a_j \in A_j} Q(s_j(t+1), a_j) - Q(s_j(t), a_j(t)) \tag{3.6}$$

para o caso de estar aprendendo os valores dos pares estado-ação. $OI(s_j(t), a_i(t))$ *é chamado de índice de ocorrência. O mesmo é calculado como sendo a percentagem do número de vezes que o agente i executou a ação* a_i *no estado do jogo no tempo t para o agente j (*$s_j(t)$*),* $V(s_j(t))$ *é o valor do estado do jogo no tempo t,* $Q(s_j(t), a_j)$ *é o valor do par estado-ação do agente j no tempo t e* A_j *é o conjunto de todas as ações que o agente j pode executar.*

3.3 IVQ-Learning

Após definir o modelo do agente interagindo com o ambiente e aprendendo com base na influência dos seus colegas, podemos definir as equações necessárias para a implementação do paradigma de aprendizado por reforço com valores de influência. Estas equações são as necessárias para o desenvolvimento de algoritmos baseados na nossa proposta, e recomendamos fortemente trabalhar com base em modificações dos algoritmos tradicionalmente usados em RL. Neste sentido, existem vários algoritmos que podem ser classificados como de aprendizado por reforço, sendo que uma boa revisão pode ser encontrada no livro de Sutton e Barto [Suton & Barto 1998].

Após realizar uma análise dos possíveis algoritmos a serem implementados, optamos por desenvolver uma versão modificada do algoritmo Q-learning [Watkins 1989] usando o paradigma proposto. Este algoritmo é um dos algoritmos de aprendizado por reforço mais usados na atualidade, tendo sido aplicado a problemas como aprendizado em robótica [Suh et al. 1997, Huber & Grupen 1997, Gonçalves et al. 1999, Gonçalves et al. 2000, Gu & Hu 2005], atribuição de canais em sistemas de comunicação móvel [Junhong & Haykin 1999], no problema do jogo dos blocos [Laurent & Piat 2001], na criação de estratégias de licitação de fornecedores de energia [Xiong et al. 2002], projeto de agentes inteligentes para administração de estoques [Hong & Park 2004], projeto de um sistema de guia de caminhos baseado em mapas eletrônicos [Zou et al. 2005], na navegação de robôs moveis [Barrios-Aranibar & Alsina 2004, Tanaka et al. 2007], no problema de conservação de energia e conforto em construções [Dalamagkidis et al. 2007], na alocação de recursos [Usaha & Barria 2007, Vengerov 2007], entre muitos outros.

Ainda, Q-learning e seus derivados também são aplicados em problemas usando sistemas multi-agentes. Por exemplo, o *fuzzy Q-learning* é aplicado em jogos multi-jogador repetitivos não cooperativos [Ishibuchi et al. 1997], uma versão hierárquica do Q-learning (HQL) é aplicada ao controle de movimentos e esquema de coordenação de um robô com seis pernas [Kirchner 1997], uma versão modular do Q-learning é aplicada a futebol de robôs [Park et al. 2001] e o próprio Q-learning é aplicado à tomada de decisões econômicas usando um grupo de agentes [Tesauro & Kephart 2002], entre outras aplicações que podem ser encontradas facilmente na literatura.

No algoritmo Q-learning tradicional, a modificação dos valores dos pares estado-ação

é realizada utilizando a Equação 3.7, dada por:

$$Q(s(t),a(t)) \leftarrow Q(s(t),a(t)) + \alpha(r(t+1) + \gamma \max_{a} Q(s(t+1),a) - Q(s(t),a(t))) \quad (3.7)$$

onde $Q(s(t),a(t))$ é o valor da ação $a(t)$ no estado $s(t)$, α é o coeficiente de aprendizado ($0 \leq \alpha \leq 1$), γ é o coeficiente de desconto ($0 \leq \gamma \leq 1$), $s(t+1)$ é o estado resultante da execução da ação $a(t)$ e $r(t+1)$ é a recompensa instantânea obtida ao executar a ação $a(t)$.

A versão que propomos, utilizando o paradigma de aprendizado com valores de influência (IVQ-learning), além de utilizar as Equações 3.3, 3.4 e 3.6, deve utilizar a Equação 3.8:

$$Q(s(t),a_i(t)) \leftarrow Q(s(t),a_i(t)) + \alpha(r_i(t+1) + \gamma \max_{a_i \in A_i} Q(s(t+1),a_i) - Q(s(t),a_i(t)) + IV_i) \quad (3.8)$$

onde $Q(s(t),a_i(t))$ é o valor da ação $a_i(t)$ executada pelo agente i, α é o coeficiente de aprendizado ($0 \leq \alpha \leq 1$), γ é o coeficiente de desconto ($0 \leq \gamma \leq 1$), $r_i(t+1)$ é a recompensa obtida pelo agente i e IV_i é o valor de influência do agente i.

A Equação 3.8 introduz a função de avaliação (IV_i), configurando-se então o Algoritmo 1). A definição desta função pode depender do tipo de aplicação, podendo ser determinada mesmo empiricamente, de forma bem simples. De fato, uma constante pode ser retornada pelos agentes, dependendo do reforço dado pela ação.

Algoritmo 1 Algoritmo IVQ-Learning para o agente i

Require: Inicializar $Q(s_i, a_i)$ de formar arbitrária
  for all episódios **do**
    Inicializar $s_i(0)$
    $t \leftarrow 0$
    repeat
      Escolher ação $a_i(t)$ no estado $s_i(t)$, usando uma política derivada de Q
      Executar ação $a_i(t)$, observar $r_i(t+1)$ e $s_i(t+1)$
      $RV_i = r_i(t+1) + \max_{a_i \in A_i} Q(s_i(t+1), a_i) - Q(s_i(t), a_i(t))$
      Observar ações $a_j(t)$ dos outros agentes no sistema ($j \in \{1:N\} - 1$).
      Atualizar $OI(s_i(t), a_j(t))$ para todos os outros agentes no sistema ($j \in \{1:N\} - 1$).
      for all $j =$ agente do sistema menos i **do**

$$OP_i(j) \leftarrow \begin{cases} RV_i * OI(s_i(t), a_j(t)) & \text{Se } RV_i < 0 \\ RV_i * (1 - OI(s_i(t), a_j(t))) & \text{Se } RV_i > 0 \\ 0 & \text{caso contrario} \end{cases}$$

      end for
      Observar opiniões $OP_j(i)$ dos outros agentes no sistema ($j \in \{1:N\} - 1$).
      Calcular $IV_i = \sum_{j \in \{1:N\} - i} \beta_i(j) * OP_j(i)$
      $Q(s(t), a_i(t)) \leftarrow Q(s(t), a_i(t)) + \alpha(r_i(t+1) +$
      $\gamma \max_{a_i \in A_i} Q(s(t+1), a_i) - Q(s(t), a_i(t)) + IV_i)$
      $t \leftarrow t+1$
    until $s_i(t)$ seja terminal
  end for

Capítulo 4

Implementações

No Capítulo 3, apresentamos a nossa proposta de aprendizado através de recompensas com valores de influência. Neste capítulo, descrevemos as implementações de problemas tradicionais encontrados na literatura, visando validar o método proposto, em comparação com os métodos tradicionais. Para tal, foram escolhidos dois jogos repetitivos, um jogo estocástico que exige coordenação por parte dos agentes, chamado neste capítulo simplesmente de jogo estocástico e um jogo estocástico que exige colaboração, chamado de jogo colaborativo. Inicialmente, fizemos testes nestes jogos usando 2 agentes apenas, estendendo a seguir para N agentes de tal forma que podemos ter uma análise mais conclusiva.

Ao implementar um sistema multi-agente utilizando o paradigma proposto neste trabalho, é importante considerar as diferenças com os outros paradigmas. Quando se usa o paradigma de aprendizado como um time, os agentes simplesmente executam ações conjuntas e o único sinal que recebem é a recompensa pelas mesmas. O mesmo acontece no paradigma de aprendizado independente. Porém, na visão dos agentes, estes recebem recompensas só pelas suas ações de forma individual. Já no caso do aprendizado de ações conjuntas, os agentes também devem receber a informação indicando qual a ação que cada agente, dentro do time, executou.

Na nossa proposta, os agentes deverão receber, além da recompensa do ambiente, a opinião que os outros agentes têm sobre as suas ações. Assim, ao implementar um sistema com este paradigma, o mesmo terá que ter a estrutura mostrada na figura 4.1.

Dependendo se a solução utilizando MAS está sendo implementada em jogos repetitivos, ou em jogos estocásticos, e dependendo também no número de agentes, as equações 3.3, 3.4, 3.6 e 3.8 podem ser utilizadas diretamente ou então com pequenas modificações como será mostrado neste Capítulo. Para o caso de aprendizado independente, utilizamos versões simplificadas do algoritmo Q-Learning que são denominadas nesta tese de *IQ-Learning*. Já os algoritmos implementados para o paradigma de aprendizado de ações conjuntas serão chamados de *JAQ-Learning* e são também derivados do algoritmo Q-Learning.

Por outro lado, todos os algoritmos derivados do Q-Learning deverão utilizar uma determinada política de seleção de ações. Existem três tipos de políticas: as gulosas, as ε-gulosas e as *softmax*. A política escolhida deve garantir a exploração de novas soluções (*exploration*) e a exploração do conhecimento adquirido (*explotation*). Neste sentido, a

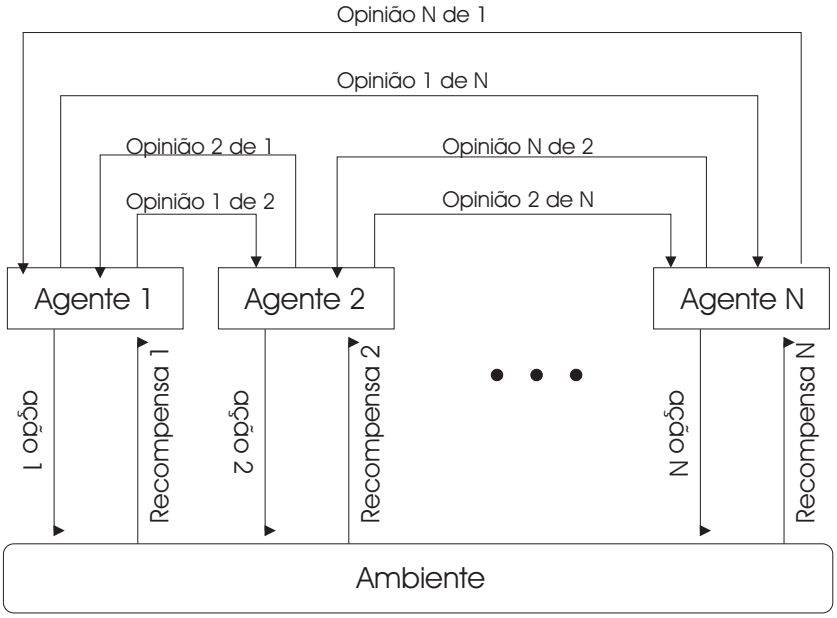


Figura 4.1: Sistema Multi-Agente Implementado com IVRL

melhor das três é a política *softmax* e é a que foi usada nos testes realizados durante o desenvolvimento da tese.

Uma política *softmax* muito popular é a baseada na equação de Boltzman:

$$Pr(a) = \frac{e^{Q(a)/T}}{\sum_{a'} e^{Q(a')/T}} \tag{4.1}$$

onde T é o parâmetro de temperatura, que pode diminuir com o tempo, com o que a exploração do conhecimento adquirido irá aumentar. Isto pode ser feito de forma que a convergência seja garantida [Singh et al. 2000].

4.1 Jogos Repetitivos

Os jogos repetitivos consistem numa série de interações entre dois ou mais agentes onde depois de cada interação, cada agente deve receber uma recompensa ou uma punição. O reforço para as interações são independentes de interações prévias [Panait & Luke 2005]. Para poder fazer uma análise da nossa proposta, foram implementados dois jogos repetitivos: *Penalty game* e *climbing game* propostos por Claus e Boutilier [Claus & Boutilier 1998]. Estes jogos são amplamente usados na literatura para testar algoritmos de aprendizado por reforço em sistemas multi-agente tais como os propostos por [Lauer & Riedmiller 2000, Wang & Sandholm 2002, Kapetanakis & Kudenko 2004]. Este jogo tem um equilíbrio de Nash sub-ótimo que foi fixado em 2 por Claus e Boutilier, porém nos testes desenvolvidos para esta tese, esse valor se tornou variável com o intuito de avaliar o rendimento dos paradigmas sob diversas condições.

4.1.1 2 Agentes

Para o caso de 2 agentes, a Figura 4.2 mostra a matriz de recompensas para o *penalty game*. Nesta matriz de recompensas, k é um valor negativo e representa uma penalidade. O valor de *so* é positivo (menor do que 10) e representa um equilíbrio sub-ótimo. Observa-se que existem dois pontos de equilíbrio de Nash ótimo e um ponto sub-ótimo. Neste contexto, se o agente A escolhe a ação a_1 esperando que o agente B escolha a ação b_1 com o objetivo de receber a máxima recompensa (10), mas o agente B escolhe b_3 com a esperança que o agente A escolha a_3, também com o objetivo de obter a máxima recompensa, ambos visando alcançar o equilíbrio de Nash ótimo, então estas escolhas levariam a um resultado não desejado por ambos os agentes que será a penalidade (k), o que pode lesionar o processo de aprendizado de ambos. Neste jogo, as ações a_2 e b_2 são escolhas seguras para os agentes desde que permitam alcançar uma recompensa de 0 ou *so*, dependendo da escolha do outro agente.

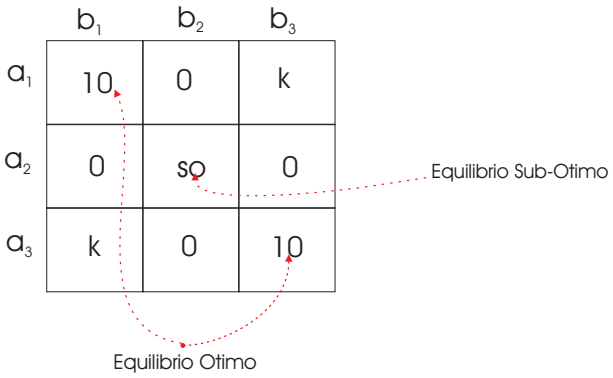


Figura 4.2: Matriz de Recompensas para o *Penalty Game* com 2 agentes

A Figura 4.3 mostra a matriz de recompensas para o caso do *climbing game*. Analisando a matriz, percebe-se que pode ser muito difícil os dois agentes convergirem para uma ação conjunta ótima, devido a poderem receber uma recompensa negativa, se não existir coordenação entre ambos.

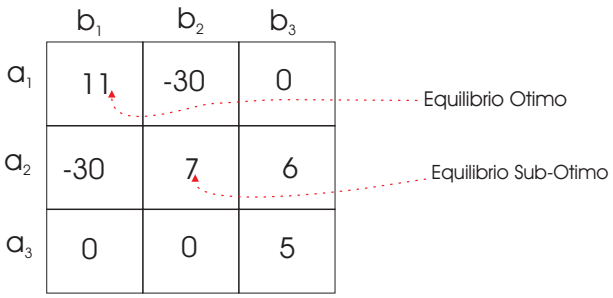


Figura 4.3: Matriz de Recompensas para o *Climbing Game* com 2 agentes

Por exemplo, se o agente A joga a_1 e o agente B escolhe b_2 então ambos receberam a recompensa negativa -30 incorporando isto no processo de aprendizado, ocasionando

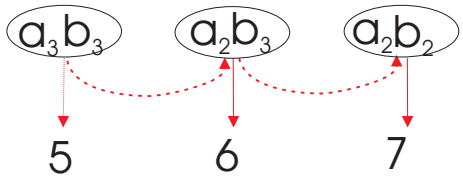


Figura 4.4: Busca do sub-ótimo no *Climbing Game* com 2 agentes

conseqüências negativas para ambos os agentes. Logo, o agente A poderá escolher a_3 e o agente B poderá escolher b_3, devido essas ações não ocasionarem nenhuma punição para ambos. Naturalmente o agente A, no futuro, poderá escolher a_2, o que o levaria a uma melhor recompensa. No mesmo caso, o agente B, no futuro, poderá escolher b_2, pela recompensa que Obteria com esta ação. Isto é ilustrado na figura 4.4. É importante observar que, teoricamente, quanto mais gulosa for a política de seleção de ações dos agentes neste jogo, maior será a probabilidade de cair num equilíbrio sub-ótimo.

Estes dois jogos foram usados para testar e comparar as versões para jogos repetitivos de 2 agentes dos três paradigmas avaliados: IQ-learning para o caso do paradigma de aprendizado independente, JAQ-learning para o caso do paradigma de aprendizado de ações conjuntas e IVQ-learning para o paradigma proposto nesta tese. Estes três algoritmos são explicados a seguir.

IQ-Learning

Para o caso do paradigma de aprendizado independente aplicado a jogos repetitivos com 2 agentes, utilizamos a versão do IQ-Learning proposta por Claus e Boutilier [Claus & Boutilier 1998] que é definido pela seguinte equação:

$$Q(a) \leftarrow Q(a) + \alpha(r - Q(a)) \tag{4.2}$$

onde $Q(a)$ é o valor da ação a no único estado, α é o coeficiente de aprendizado ($0 \leq \alpha \leq 1$) e r é a recompensa obtida pelo agente após executar a ação a. O algoritmo 2 foi o implementado para o IQ-Learning de 2 agentes.

Algoritmo 2 IQ-Learning para o agente A em jogos repetitivos de 2 agentes

Require: Inicializar $Q(a)$ de formar arbitrária
  for all jogos **do**
    Escolher ação a usando uma política derivada de Q
    Executar ação a e observar r_A
    $Q(a) \leftarrow Q(a) + \alpha(r_A - Q(a))$
  end for

JAQ-Learning

A idéia básica no aprendizado de ações conjuntas é que os agentes não calculam o valor de suas ações mas sim calculam o valor de suas ações quando combinadas com

ações de outros agentes. Para avaliar este paradigma em comparação com nossa proposta implementou-se uma das formas simples deste baseada no *Q-learning* denominado *JAQ-Learning* proposto em [Claus & Boutilier 1998].

No *JAQ-Learning*, o valor da ação conjunta (a,b) para o agente A é calculado pela equação 4.3 onde a é a ação do agente A e b é a ação do agente B, $Q(a,b)$ é o valor da ação conjunta (a,b) para o agente A, r é a recompensa obtida pelo agente A depois de que este executa a ação a e o agente B executa a ação b. α é o coeficiente de aprendizado.

$$Q(a,b) \leftarrow Q(a,b) + \alpha(r_A - Q(a,b)) \tag{4.3}$$

Porém no *JAQ-Learning*, o agente decide entre suas ações e não entre as ações conjuntas, pelo que usa um valor esperado da ação que inclui informação sobre as ações conjuntas e o modelo atual do outro agente. Este valor é calculado pela Equação 4.4, onde a é a ação do agente A, $EV(a)$ é o valor esperado da ação a, b é a ação do agente B, B é o conjunto de ações do agente B e Pr_b é a probabilidade do outro agente escolher a ação b. O algoritmo exposto 3 foi implementado para o JAQ-Learning com dois agentes, o qual é comparado com nossa proposta.

$$EV(a) \leftarrow \sum_{b \in B} Q(b \cup a) \times Pr_b \tag{4.4}$$

Algoritmo 3 JAQ-Learning para o agente A em jogos repetitivos de 2 agentes

Require: Inicializar $Q(a,b)$ de formar arbitrária
  for all jogos **do**
    Calcula o valor esperado para todas as ações possíveis segundo:
    $EV(a) \leftarrow \sum_{b \in B} Q(b \cup a) \times Pr_b$
    Escolher ação a usando uma política derivada de EV
    Executar ação a e observar r_A
    Observa a ação b do agente B
    Calcula $Q(a,b) \leftarrow Q(a,b) + \alpha(r_A - Q(a,b))$
  end for

IVQ-Learning

Para implementar o algoritmo IVQ-Learning para jogos repetitivos de 2 agentes, fizemos uma simplificação da equação 3.8 obtendo a Equação 4.5 para o agente A. Uma equação análoga é implementada no agente B.

$$Q(a) \leftarrow Q(a) + \alpha(r_A - Q(a)) + \beta(OP_B(A)) \tag{4.5}$$

onde $Q(a)$ é o valor da ação a do agente A, α é um constante de aprendizado ($0 \leq \alpha \leq 1$), β é o coeficiente de influência da opinião do outro agente (B) sobre o valor da ação a ($0 \leq \beta \leq 1$) e $OP_B(A)$ é a opinião que o outro agente têm da ação a do agente A.

Assim, a opinião de um agente em relação às ações do outro deverá ser calculada por:

$$OP_B(A) \leftarrow \begin{cases} (r_B - Q(b)) * OI(a) & \text{se } (r_B - Q(b)) < 0 \\ (r_B - Q(b)) * (1 - OI(a)) & \text{se } (r_B - Q(b)) > 0 \\ 0 & \text{caso contrario} \end{cases}$$

onde r_B é a recompensa obtida pelo agente B, $Q(b)$ é o valor da ação b para o agente B e $OI(a)$ é o índice de ocorrência da ação a do agente A observada pelo agente B.

Utilizando as versões simplificadas das equações 3.3, 3.4 e 3.8, o modelo de aprendizado para o agente A ocorre usando o Algoritmo 4. O algoritmo para o agente B é análogo.

Algoritmo 4 IVQ-Learning para o agente A em jogos repetitivos de 2 agentes

Require: Inicializar $Q(a)$ de formar arbitrária
  for all jogos **do**
    Escolher ação a usando uma política derivada de Q
    Executar ação a e observar r_A
    Observar ação b do agente B

$$OP_A(B) \leftarrow \begin{cases} (r_A - Q(a)) * OI(b) & \text{se } (r_A - Q(a)) < 0 \\ (r_A - Q(a)) * (1 - OI(b)) & \text{se } (r_A - Q(a)) > 0 \\ 0 & \text{caso contrario} \end{cases}$$

    Observar $OP_B(A)$
    $Q(a) \leftarrow Q(a) + \alpha(r_A - Q(a)) + \beta(OP_B(A))$
  end for

4.1.2 Implementando os algoritmos para N agentes

Posto que os jogos repetitivos representam ambientes de teste simplificados, onde é possível verificar as propriedades dos algoritmos de aprendizado por reforço aplicados a sistemas multi-agente, nesta tese estendemos os problemas propostos por Claus e Boutilier para serem aplicados em problemas de N agentes. A Figura 4.5 mostra a matriz de recompensas para 3 agentes no *Penalty Game*. Similar ao que ocorre no jogo de 2 agentes, para 3 três agentes existem dois pontos de equilíbrio ótimos: $\{(a_1, b_1, c_1), (a_3, b_3, c_3)\}$ e um ponto sub-ótimo (a_2, b_2, c_2).

Quando generalizado para 4 ou mais agentes, a visualização da matriz de recompensas se torna difícil, porém as generalizações devem respeitar as seguintes regras:

1. Todos os agentes deverão ganhar uma recompensa 10 se todos escolherem ao mesmo tempo sua primeira ação ou sua terceira ação.
2. Se todos os agentes escolherem ao mesmo tempo sua segunda ação, a recompensa será o valor *so* com a restrição que $0 \leq so < 10$.
3. Se o conjunto de ações escolhidas por todos os agentes dentro do sistema inclui unicamente a primeira ou terceira ação de cada um deles e a situação não se encaixa dentro da regra 1 os agentes devem receber como recompensa o valor k ($-\infty < k \leq 0$).

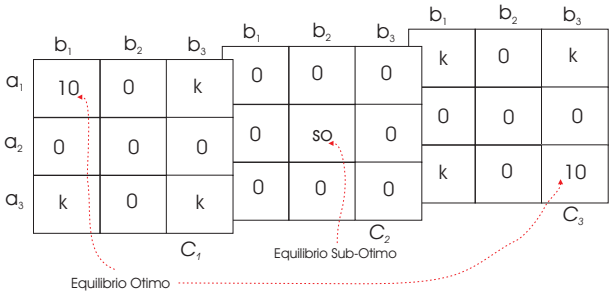


Figura 4.5: Matriz de Recompensas para o *Penalty Game* com 3 agentes

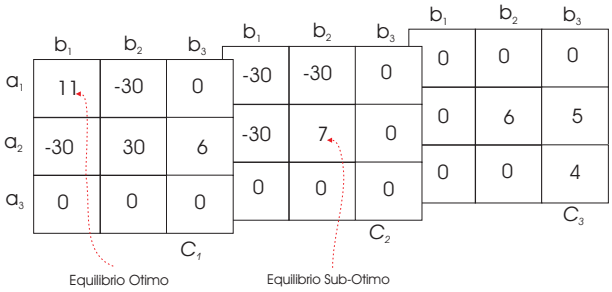


Figura 4.6: Matriz de Recompensas para o *Climbing Game* com 3 agentes

4. Se o conjunto de ações escolhidas não se encaixa em nenhuma das regras anteriores, a recompensa deverá ser 0.

O *climbing game* também foi generalizado para seu uso em problemas de N agentes. Assim, a figura 4.6 mostra sua matriz de recompensas para sistemas com três agentes. Como é visto o equilíbrio de Nash ótimo é (a_1, b_1, c_1) e o equilíbrios de Nash sub-ótimo é $(a_2,\ b_2,\ c_2)$. A Figura 4.7 mostra o caminho sub-ótimo.

Como pode ser observado, o jogo conservou as mesmas características que ele tinha para jogos com 2 agentes. Podemos verificar que existe uma tendência natural dos agentes chegarem ao equilíbrio sub-ótimo, ao passo que existe uma dificuldade de escolher o ótimo já que qualquer erro de coordenação pode acarretar uma penalidade de -30. Assim, inicialmente todos os agentes estarão tentados a escolher as ações a_3, b_3 e c_3 respectivamente por não existir probabilidade de serem penalizados com a mesma. Após isto, o agente A poderá perceber que, com a escolha feita por seus colegas, ele pode mudar tranqüilamente sua escolha para a_2 e ainda todos irão receber uma recompensa maior (5). Estando no ponto em que as escolhas são a_2, b_3 e c_3, o agente B poderá perceber que, se ele mudar sua escolha para b_2, a situação irá melhorar chegando ao conjunto de

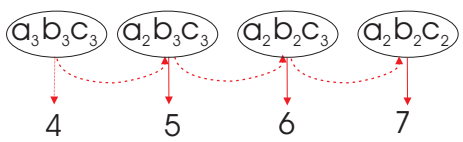


Figura 4.7: Busca do sub-ótimo para o *Climbing Game* com 3 agentes

escolhas a_2, b_2 e c_3. Finalmente, será a vez do agente C perceber a vantagem de mudar para c_2 fazendo com que o jogo chegue a um equilíbrio de Nash, porém o equilíbrio será sub-ótimo.

No caso de jogos com 4 ou mais agentes, o mesmo deve respeitar as seguintes regras:

1. Deve existir um equilíbrio ótimo quando todos os agentes escolhem sua primeira ação, devendo ser a recompensa neste caso a maior dentro de toda a matriz de recompensas.
2. Se todos os agentes escolherem a sua segunda ação, deverá ser dada uma recompensa positiva menor que a recompensa dada pela regra 1, porém não deverá existir outra recompensa igual ou maior a ela no jogo.
3. Se o conjunto de ações escolhidas por todos os agentes dentro do sistema inclui unicamente a primeira ou segunda ação de cada um deles, e a situação não se encaixa dentro da regra 1 nem dentro da regra 2, os agentes devem receber uma punição, sendo a mesma um valor negativo.
4. A partir da recompensa dada pela regra 2, deve se construir uma escada de recompensas para as escolhas que gradativamente levem ao ponto em que todos os agentes escolheram a sua terceira ação. Por exemplo, se o i-ésimo agente é chamado de ai e a escolha da sua j-ésima ação é representada como ai_j, se a recompensa dada pela regra 2 foi r, então a seguinte seria uma possível seqüência de recompensas e as situações correspondentes:
   - $a1_2, a2_2, ..., an-1_2, an_2 = r$
   - $a1_2, a2_2, ..., an-1_2, an_3 = r-1$
   - $a1_2, a2_2, ..., an-1_3, an_3 = r-2$

     .

     .

     .

   - $a1_2, a2_3, ..., an-1_3, an_3 = r-n+1$
   - $a1_3, a2_3, ..., an-1_3, an_3 = r-n$
5. Se a situação não se encaixar em nenhuma das regras anteriores, a recompensa deverá ser 0.

Como pode ser observado, na generalização deste jogo não se consideram valores fixos porque se o número de agentes for maior do que 7 e continuarmos com os valores definidos por Claus e Boutilier, chegaríamos num ponto de teríamos penalidades ao invés de recompensas positivas e a principal característica do jogo (a escada de recompensas) será perdida. Ainda, foi observado em testes de laboratório que a escolha dos valores das recompensas vai influenciar diretamente na capacidade de convergência dos algoritmos. Por exemplo, para os valores definidos originalmente para o problema aplicados a jogos com 3 e 4 agentes a capacidade de convergência dos algoritmos se vê muito reduzida. Mas, como o objetivo é o de testar os algoritmos sob diversas condições e as condições do jogo para 3 e 4 agentes podem considerar-se extremas (e.g. Em testes iniciais, nenhum algoritmo convergia) se escolheu continuar com esses valores.

IQ-Learning

O algoritmo implementado para o caso do paradigma de aprendizado independente em jogos repetitivos com N agentes é o mesmo implementado para o caso de 2 agentes (Algoritmo 2). A única ressalva é que ao falarmos do i-ésimo agente, a recompensa deverá ser chamada de r_i.

JAQ-Learning

Para o paradigma de aprendizado de ações conjuntas, o algoritmo 3 sofre mudanças no sentido que ao invés de o agente modelar duplas como ações conjuntas, cada ação conjunta terá N componentes, onde N é o número de agentes dentro do jogo. Assim, a equação 4.3 para o i-ésimo agente será definida pela equação 4.6.

$$Q_i(a_1,...,a_N) \leftarrow Q_i(a_1,...,a_N) + \alpha(r_i - Q_i(a_1,...,a_N)) \tag{4.6}$$

onde $Q_i(a_1,...,a_N)$ representa o valor da ação conjunta $(a_1,...,a_N)$ para o agente i, r_i é a recompensa ganha pelo agente i e a_i é a ação escolhida pelo agente i.

Já a equação 4.4, fica mais complexa desde que ao invés de considerar só a probabilidade de um outro agente escolher uma ação, deverá ser considerada a probabilidade conjunta de todas as combinações das ações dos outros agentes. Então, a mesma se transforma na equação 4.7.

$$EV(a_i^j) \leftarrow \sum_{a_{-i} \in A_{-i}} Q(a_{-i} \cup a_i^j) * \prod_{k \neq i} Pr(a_k^l) \tag{4.7}$$

onde $EV(a_i^j)$ é o valor esperado da j-ésima ação do i-ésimo agente, a_{-i} representa uma ação qualquer de um agente que não seja o agente i, A_{-i} representa o conjunto de ações dos agentes diferentes de i, $Q(a_{-i} \cup a_i^j)$ representa o valor da ação conjunta formada pela j-ésima ação do i-ésimo agente e as ações dos outros agentes, e $Pr(a_k^l)$ representa a probabilidade de um agente k escolher a ação l que formou parte da ação conjunta avaliada por Q.

Com estas modificações, o algoritmo para o i-ésimo agente de um sistema aprendendo pelo paradigma de ações conjuntas é o mostrado no algoritmo 5

Algoritmo 5 JAQ-Learning para o agente i em jogos repetitivos de N agentes

Require: Inicializar $Q_i(a_1,...,a_N)$ de formar arbitrária
  for all jogos **do**
    Calcula o valor esperado para todas as ações do agente segundo:
    $EV(a_i^j) \leftarrow \sum_{a_{-i} \in A_{-i}} Q(a_{-i} \cup a_i^j) * \prod_{k \neq i} Pr(a_k^l)$
    Escolher ação a_i usando uma política derivada de EV
    Executar ação a_i e observar r_i
    Observa as ações dos outros agentes dentro do sistema (a_{-i})
    Calcula $Q_i(a_1,...,a_N) \leftarrow Q_i(a_1,...,a_N) + \alpha(r_i - Q_i(a_1,...,a_N))$
  end for

IVQ-Learning

Para a implementação da nossa proposta com N agentes em jogos repetitivos, generalizamos a Equação 4.5 obtendo a Equação 4.8:

$$Q(a_i) \leftarrow Q(a_i) + \alpha(r_i - Q(a_i) + IV_i) \tag{4.8}$$

onde $Q(a_i)$ é o valor da ação a_i do agente i, α é a o coeficiente de aprendizado ($0 \leq \alpha \leq 1$) e IV_i é o valor de influência do agente i definido por:

$$IV_i = \sum_{j \in \{1:N\}-i} \beta_i(j) * OP_j(i) \tag{4.9}$$

onde β é o coeficiente de influência da opinião dos outros agentes ($0 \leq \beta \leq 1$) e $OP_j(i)$ é a opinião que um outro agente j têm da ação a_i do agente i.

A opinião de um agente i em relação às ações de um outro j deverá ser calculada por:

$$OP_i(j) \leftarrow \begin{cases} (r_i - Q(a_i)) * OI(a_j) & \text{Se } (r_i - Q(a_i)) < 0 \\ (r_i - Q(a_i)) * (1 - OI(a_j(t))) & \text{Se } (r_i - Q(a_i)) > 0 \\ 0 & \text{caso contrario} \end{cases} \tag{4.10}$$

onde $OI(a_j)$ é o índice de ocorrência da ação a_j do agente j observada pelo agente i, r_i é a recompensa obtida pelo agente i, $Q(a_i)$ é o valor da ação a para o agente i. Então o algoritmo 6 foi implementado para o agente i em jogos repetitivos com N agentes.

Algoritmo 6 IVQ-Learning para o agente i em jogos repetitivos de N agentes

Require: Inicializar $Q(a_i)$ de formar arbitrária
  for all jogos **do**
    Escolher ação a_i usando uma política derivada de Q
    Executar ação a_i e observar r_i
    Observar ações dos outros agentes dentro do sistema
    $$OP_i(j) \leftarrow \begin{cases} (r_i - Q(a_i)) * OI(a_j) & \text{Se } (r_i - Q(a_i)) < 0 \\ (r_i - Q(a_i)) * (1 - OI(a_j(t))) & \text{Se } (r_i - Q(a_i)) > 0 \\ 0 & \text{caso contrario} \end{cases}$$
    Observar $OP_j(i)$ de todos os agentes j dentro do sistema em relação à ação executada por i.
    $$IV_i = \sum_{j \in \{1:N\}-i} \beta_i(j) * OP_j(i)$$
    $$Q(a_i) \leftarrow Q(a_i) + \alpha(r_i - Q(a_i) + IV_i)$$
  end for

4.2 Jogos Estocásticos

Para poder testar o paradigma proposto neste trabalho em jogos estocásticos, criamos um jogo chamado *mundo grade*. O objetivo do jogo é a coordenação entre os agentes.

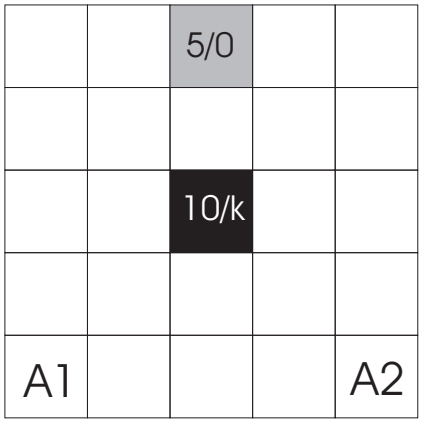


Figura 4.8: Jogo do *Mundo Grade* para testar a coordenação entre dois agentes

Isto significa que, neste jogo (Figura 4.8), ambos os agentes terão que coordenar suas ações para poder obter recompensas positivas. Qualquer falha na coordenação causaria penalidades para os dois.

O jogo começa com o agente um ($A1$) na posição $(5,1)$ e o agente dois ($A2$) na posição $(5,5)$. Os agentes têm que chegar às posições $(1,3)$ e $(3,3)$ para poder terminar o jogo. Se os dois chegarem ao mesmo tempo, ganharão recompensas positivas. Quando chegarem juntos na posição $(1,3)$ receberão 5 pontos cada e quanto chegarem ao mesmo tempo na posição $(3,3)$ cada um receberá 10 pontos. Porém, se só um deles chegar na posição $(3,3)$ os dois serão punidos com uma penalidade k. No outro caso, se só um dos dois agentes chegar na posição $(1,3)$ eles não serão punidos mas também não serão recompensados.

O jogo termina quando pelo menos um dos agentes chega na posição $(1,3)$ ou na $(3,3)$. O jogo também termina quando a ação de um dos agentes leva ele para fora da grade. Os agentes têm quatro possíveis ações: direita, esquerda, acima e abaixo.

Note que este jogo têm vários equilíbrios de Nash, que são todas as estratégias que levam os agentes a ganhar 5 ou 10 pontos. Porém, os equilíbrios ótimos são aqueles onde os agentes obtenham 10 pontos em quatro ações.

Do mesmo modo que foi realizado com os jogos repetitivos, implementamos os dois paradigmas tradicionais e o paradigma proposto nesta tese para jogos estocásticos. Os algoritmos implementados para os dois paradigmas tradicionais são explicados nas subseções seguintes, e no caso do IVQ-learning, o algoritmo 1 foi o implementado para sua aplicação em jogos estocásticos.

4.2.1 IQ-Learning

Para os jogos estocásticos, o algoritmo IQ-Learning é baseado na Equação 4.11, dada por:

$$Q(s_i(t),a_i(t)) \leftarrow Q(s_i(t),a_i(t)) + \alpha(r_i(t) + \gamma \max_{a_i} Q(s_i(t+1),a_i) - Q(s_i(t),a_i(t))) \quad (4.11)$$

onde $Q(s_i(t), a_i(t))$ é o valor da ação $a_i(t)$ no estado $s_i(t)$, α é o coeficiente de aprendizado ($0 \leq \alpha \leq 1$), γ é o coeficiente de desconto ($0 \leq \gamma \leq 1$), $s(t+1)$ é o estado resultante da execução da ação $a(t)$ e r é a recompensa instantânea obtida ao executar a ação $a(t)$. Baseado nesta Equação, o Algoritmo 7 implementa este paradigma para o agente i de um jogo estocástico de N agentes.

Algoritmo 7 IQ-Learning para o agente i em jogos estocásticos de N agentes

Require: Inicializar $Q(s_i(t), a_i(t))$ de formar arbitrária
  for all episodios **do**
    $t \leftarrow 0$
    repeat
      Escolher ação $a_i(t)$ usando uma política derivada de Q
      Executar ação $a_i(t)$ e observar $r_i(t)$
      $Q(s_i(t), a_i(t)) \leftarrow Q(s_i(t), a_i(t)) + \alpha(r_i(t) + \gamma \max_{a_i} Q(s_i(t+1), a_i) - Q(s_i(t), a_i(t)))$
      $t \leftarrow t+1$
    until $s_i(t)$ seja terminal
  end for

4.2.2 JAQ-Learning

No caso do paradigma de aprendizado de ações conjuntas aplicado a jogos estocásticos, o valor das ações conjuntas é modificado utilizando a equação 4.12:

$$Q(s_i(t), a_1(t), ..., a_N(t)) \leftarrow Q(s_i(t), a_1(t), ..., a_N(t)) +$$

$$\alpha(r_i(t) + \gamma \max_{a_1,...,a_N} Q(s_i(t+1), a_1, ..., a_N) - Q(s_i(t), a_1(t), ..., a_N(t))) \quad (4.12)$$

onde $a_i(t)$ é a ação executada pelo agente i no tempo t, $Q(s_i(t), a_1(t), ..., a_N(t))$ é o valor da ação conjunta $(a_1(t), ..., a_N(t))$ para o agente i no estado $s_i(t)$, $r_i(t)$ é a recompensa obtida pelo agente após ele executar a ação $a_i(t)$ e os outros agentes no sistema executar as suas correspondentes ações, α é o coeficiente de aprendizado ($0 \leq \alpha \leq 1$) e γ é o coeficiente de desconto ($0 \leq \gamma \leq 1$).

Para calcular o valor esperado de suas ações o agente usa a Equação 4.13, dada por:

$$EV(s_i(t), a_i^j) \leftarrow \sum_{a_{-i} \in A_{-i}} Q(s_i(t), a_{-i} \cup a_i^j) * \prod_{k \neq i} Pr(t, a_k^l) \quad (4.13)$$

onde $EV(s_i(t), a_i^j)$ é o valor esperado da j-ésima ação no estado $s_i(t)$ do i-ésimo agente, a_{-i} representa uma ação qualquer de um agente que não seja o agente i, A_{-i} representa o conjunto de ações dos agentes diferentes de i, $Q(s_i(t), a_{-i} \cup a_i^j)$ representa o valor, no estado $s_i(t)$, da ação conjunta formada pela j-ésima ação do i-ésimo agente e as ações dos outros agentes, e $Pr(t, a_k^l)$ representa a probabilidade, no tempo t, de um agente k escolher a ação l que formou parte da ação conjunta avaliada por Q.

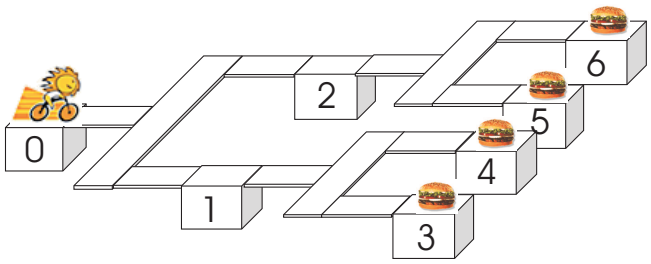


Figura 4.9: Jogo para verificar a capacidade de colaboração entre agentes.

A Algoritmo 8 implementa o paradigma de ações conjuntas num jogo estocástico, para o *i*-ésimo agente de um sistema com N agentes.

Algoritmo 8 JAQ-Learning para o agente i em jogos estocásticos de N agentes

Require: Inicializar $Q(s_i(t), a_1(t), ..., a_N(t))$ de formar arbitrária
  for all episodios **do**
    $t \leftarrow 0$
    repeat
      Calcula o valor esperado para todas as ações do agente segundo:
      $EV(s_i(t), a_i^j) \leftarrow \sum_{a_{-i} \in A_{-i}} Q(s_i(t), a_{-i} \cup a_i^j) * \prod_{k \neq i} Pr(t, a_k^l)$
      Escolher ação $a_i(t)$ usando uma política derivada de EV
      Executar ação $a_i(t)$ e observar $r_i(t)$
      Observa as ações dos outros agentes dentro do sistema (a_{-i})
      Calcula $Q(s_i(t), a_1(t), ..., a_N(t)) \leftarrow Q(s_i(t), a_1(t), ..., a_N(t)) +$
      $\alpha(r_i(t) + \gamma \max_{a_1,...,a_N} Q(s_i(t+1), a_1, ..., a_N) - Q(s_i(t), a_1(t), ..., a_N(t)))$
      $t \leftarrow t + 1$
    until $s_i(t)$ seja terminal
  end for

4.3 Jogos Colaborativos

Para poder testar a capacidade de colaboração e auto-organização (alocação autônoma de tarefas) num grupo de agentes usando aprendizado por reforço, criamos o jogo de busca de provisões mostrado na Figura 4.9. Neste jogo, chamado de "jogo das pontes", um time de agentes terá que procurar comida no ambiente e comê-la. Quando a comida no ambiente é esgotada, então, o jogo acaba. Inicialmente, os agentes não sabem que têm que pegar a comida para ganhar o jogo. Então, este jogo inclui duas tarefas de aprendizado. A primeira é aprender que comer é bom para os agentes (vai lhes dar boas recompensas) e a segunda é aprender as posições das fontes de comida para poder comê-la e ganhar o jogo.

O jogo é composto por sete plataformas unidas por pontes. A plataforma inicial (plataforma 0) têm acesso às plataformas 1 e 2 através de duas pontes. Finalmente, as plataformas 1 e 2 são unidas a outras duas plataformas cada (plataformas 3 e 4 com a plataforma 1 e plataformas 5 e 6 com a plataforma 2). Ainda, como pode se notar na Figura 4.9, a comida foi colocada nas plataformas $3, 4, 5$ e 6.

Os agentes devem começar o jogo na plataforma 0 como mostrado na figura 4.9 e ele pode realizar três ações em cada posição dentro do jogo (direita, esquerda, voltar). Quando o agente executa a ação "direita", ele entra dentro da ponte (se existir) e anda até chegar na plataforma localizada à direita da plataforma atual. O efeito análogo tem a ação "esquerda". Já a ação "voltar"faz com que o agente ande até a plataforma que se encontra antes da plataforma atual. Por exemplo, se um agente está na plataforma 1 e ele executa a ação "direita", ele irá andar até a plataforma 4, já se ele executa a ação "esquerda", ele andará até a plataforma 3; finalmente, se ele escolher a ação "voltar", ele vai andar até a plataforma 0. Se uma ação não pode ser executada porque a ponte não existe, então o agente vai ficar na plataforma atual.

No paradigma de aprendizado por reforço, é importante modelar o estado do ambiente e a forma como os agentes irão receber as recompensas durante o jogo. Neste jogo, o estado é formado pelas posições de cada um dos agentes no mundo e por quatro indicadores, mostrando a existência ou não comida em cada uma das quatro plataformas objetivo. É importante observar que o estado neste modelo não inclui as posições das plataformas objetivo, isto significa que os agentes terão de procurá-la durante o jogo. Também, quando um agente chega numa fonte de comida, todos os agentes no jogo irão receber uma recompensa positiva igual a 1. Em outros casos os agentes não irão receber recompensa. É esperado que os agentes possam aprender o caminho para chegar à comida partindo de qualquer posição dentro do jogo. Notamos que este modelo só garante que o agente vai aprender a encontrar as fontes de comida e não a forma como isto deve ser explorada (a estratégia). Assim, esta estratégia será responsabilidade do algoritmo de aprendizado.

Em relação aos algoritmos implementados para este jogo, foram usados os algoritmos 1, 7 e 8, já que os jogos colaborativos são também jogos estocásticos.

No próximo Capítulo, realizaremos uma análise dos resultados obtidos com as implementações realizadas.

Capítulo 5

Resultados Experimentais

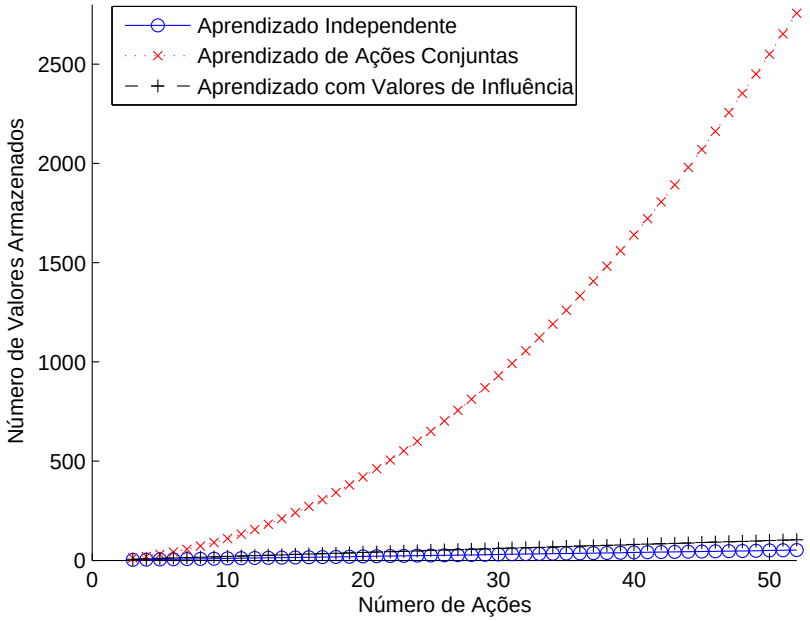
Os algoritmos implementados no presente trabalho, descritos no Capítulo 4, foram testados e avaliados com o intuito de analisar seu rendimento e seu comportamento em relação à mudanças realizadas nos seus parâmetros ou nas condições do ambiente.

Neste capítulo, mostramos os resultados obtidos nesses experimentos realizados. Inicialmente, são mostrados os resultados obtidos para os dois jogos repetitivos implementados, os quais mostram claramente o rendimento e comportamento de cada um dos algoritmos implementados. Logo, são mostrados os resultados com jogos estocásticos que exigem coordenação e jogos estocásticos que exigem colaboração.

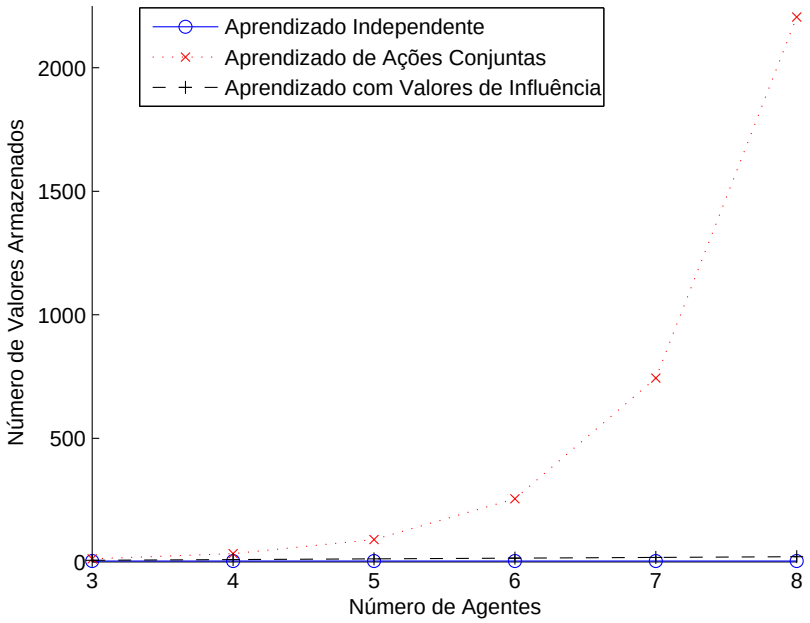
5.1 Coordenação em jogos repetitivos

Ao implementar o paradigma proposto neste trabalho juntamente com os outros dois paradigmas, observamos que os aprendizes independentes só têm que armazenar os valores de suas próprias ações; os aprendizes de ações conjuntas têm que armazenar os valores das ações conjuntas e a probabilidade dos outros agentes executarem cada uma das suas ações; e finalmente, os aprendizes com valores de influência têm que armazenar os valores de suas próprias ações e o índice de ocorrência das vezes que os outros agentes executaram cada uma das suas ações. Isto quer dizer que o número de valores a serem armazenados cresce se o número de ações e/ou agentes aumenta. A Figura 5.1 mostra como estes valores estourados aumenta: A Figura 5.1(a) mostra como estes valores armazenadas aumentam para um jogo repetitivo com 2 agentes e um número variável de ações e a Figura 5.1(b) mostra como esta quantidade aumenta para um jogo repetitivo com 3 ações e um número variável de agentes. Como pode ser observado, o aprendizado independente e o aprendizado com valores de influência têm uma vantagem sobre o aprendizado de ações conjuntas.

Como dito anteriormente, os jogos repetitivos são ambientes de teste simplificados que permitem avaliar o rendimento dos algoritmos de aprendizado por reforço sob diversas situações que incluem desde a mudança dos seus parâmetros até mudanças na estrutura do ambiente (e.g. mudança nas recompensas). Assim, após analisar as necessidades de armazenamento de cada um dos paradigmas propostos, a continuação serão analisados os resultados obtidos para jogos repetitivos com 2, 3 e 4 agentes. Se fez esta análise com o intuito de conhecer a capacidade de convergência dos nosso algoritmos, verificar a capaci-



(a) Dois Agentes



(b) Três Ações

Figura 5.1: Capacidade de Armazenamento Necessária em Jogos Repetitivos

dade de convergência dos paradigmas tradicionais relatados em trabalhos anteriores, e ao mesmo tempo determinar qual dos algoritmos tem uma maior capacidade de convergência a equilibrios de Nash ótimos.

5.1.1 Jogos com 2 agentes

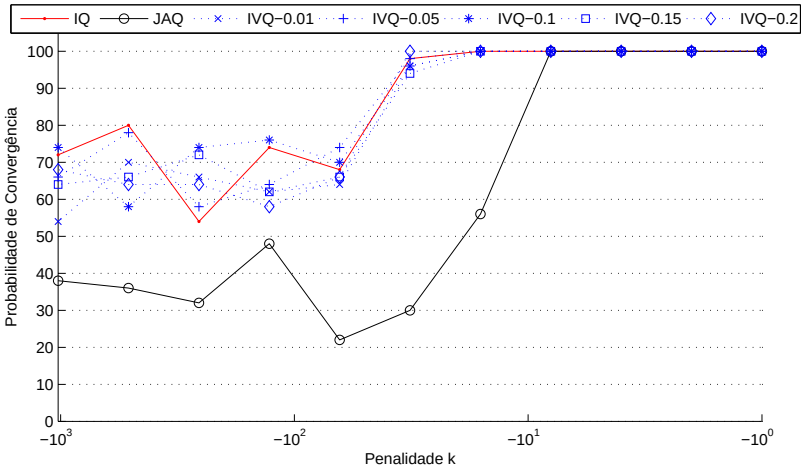
Os experimentos e resultados para jogos repetitivos com 2 agentes mostrados neste documento foram obtidos ao testar implementações dos algoritmos feitas na linguagem C++ e de forma totalmente paralela. Os mesmos têm uma correlação direta com os resultados para jogos com 3 e 4 agentes por terem sido todos testados sob as mesmas condições (mesmos parâmetros e valores das penalidades e recompensas sub-ótimas), na mesma plataforma e com a mesma política de seleção de ações (*softmax*). Porém, durante o desenvolvimento da tese, especificamente nos testes iniciais da nossa proposta, se fizeram outros experimentos envolvendo os mesmos jogos repetitivos com 2 agentes (implementados em Matlab de forma seqüencial), os quais foram publicados num congresso internacional [Barrios-Aranibar & Gonçalves 2007*a*]. Nesses resultados, os quais foram feitos usando uma política *softmax* gulosa (se dava maior prioridade às melhores ações porém não chegava a ser nem $\varepsilon-$Gulosa nem Gulosa), foi mostrado que para o jogo *Penalty Game*, o algoritmo com maior capacidade de convergência era o JAQ-Learning, seguido por nossa proposta e finalmente o algoritmo IQ-Learning. Já para o caso do jogo *Climbing Game* se obteve que o único algoritmo com probabilidade –diferente de zero– de convergir ao equilíbrio de Nash ótimo era o IVQ-Learning, proposto nesta tese. Estes resultados inicias encorajaram a continuação desta tese e a realização de novos testes envolvendo uma política totalmente *Softmax* já que este tipo de políticas teoricamente melhora a probabilidade de convergência dos algoritmos de aprendizado por reforço. Neste sentido, a continuação se mostram os resultados obtidos para os dois jogos repetitivos implementados. Ainda, os parâmetros fixos usados para todos os algoritmos foram $\alpha = 0.1$ e a temperatura inicial da equação de *Boltzman* foi $T(0) = 16$.

Penalty Game

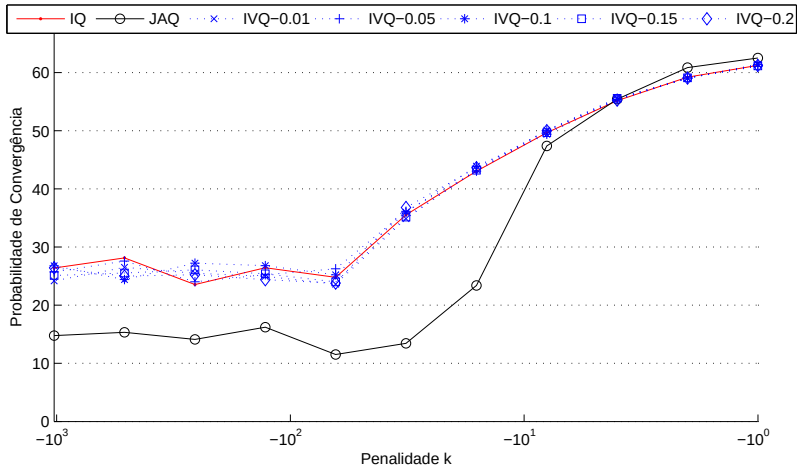
Os experimentos realizados no jogo *Penalty Game* possuem as seguintes características:

1. Valor variável da recompensa sub-ótima, $so \in \{2,3,4,5,6\}$.
2. Valor variável da penalidade, $k \in \{-1,-2,-4,-8,-16,-32,-64,-128,-256,-512,-1024\}$.
3. Valor da temperatura na equação de *Boltzman* decai a cada jogo baseado em $T(t) = T(0) * decT^t$. Assim, se definiu um valor variável do decaimento da temperatura, $decT \in \{0.999, 0.9995, 0.9998, 0.9999\}$.
4. Se testaram 5 versões do algoritmo $IVQ-Learning$ baseado num valor variável do coeficiente de influência, $\beta \in \{0.01, 0.05, 0.1, 0.15, 0.2\}$.
5. Cada versão do algoritmo implementado se treinou 50 vezes durante 10000 iterações, fazendo assim um total de 11000 programas IQ-Learning, 11000 programas JAQ-Learning e 55000 programas IVQ-Learning avaliados.

Para a análise do rendimento de cada paradigma, após o treinamento, foi calculada a probabilidade de cada programa convergir para os dois equilíbrios de Nash ótimos. Este cálculo foi feito de duas formas, a primeira considerando uma política de seleção de ações gulosa a partir dos valores aprendidos durante o treinamento, e a segunda considerando una política *softmax* onde a temperatura será definida por $T = (\max(Q) - \min(Q))/2$.



(a) Greedy

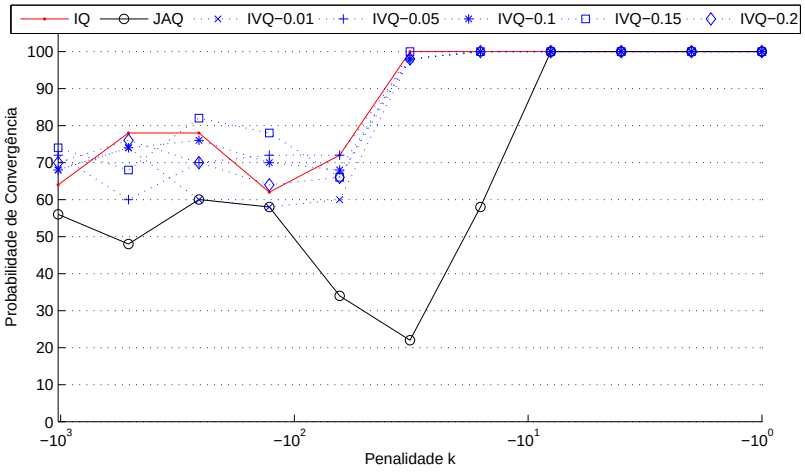


(b) Softmax

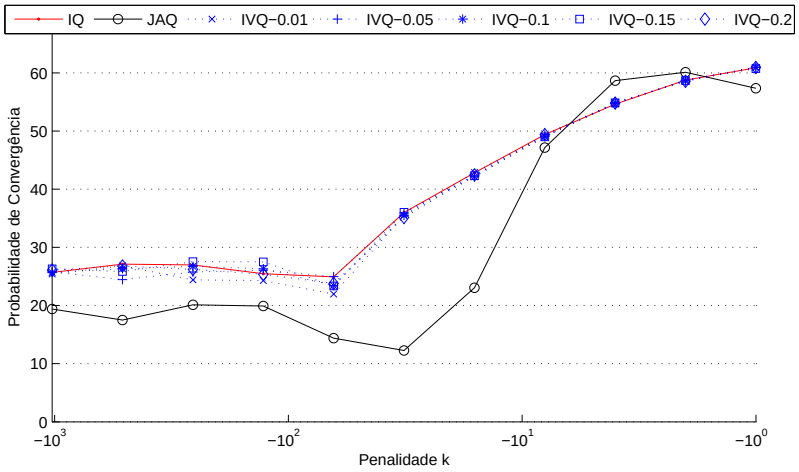
Figura 5.2: Convergência no *Penalty Game* com 2 agentes, $T = 16(0.999^t)$ e $so = 2$

Como pode ser observado na Figura 5.2, os algoritmos IQ-Learning e as 5 versões do IVQ-Learning têm melhores probabilidades de convergência que o JAQ-Learning. Estes resultados correspondem ao jogo com um equilíbrio de Nash sub-ótimo $so = 2$. Ainda, podemos observar que o comportamento é similar para ambos os tipos de estratégia de escolha de ações a partir dos valores de aprendidos, Q no caso do IQ-Learning e o IVQ-Learning, e EV para o JAQ-Learning. Também pode ser observado que a penalidade k

exerce uma influência negativa na capacidade de convergência de todos os algoritmos, porém que é mais influenciado é o algoritmo JAQ-Learning.



(a) Greedy

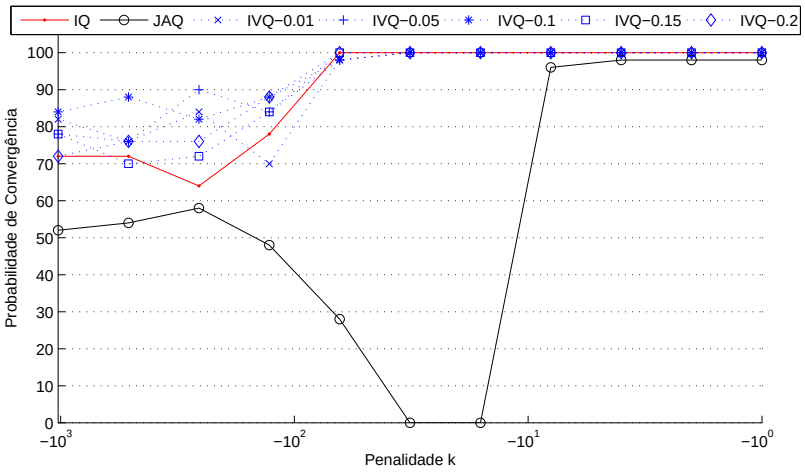


(b) Softmax

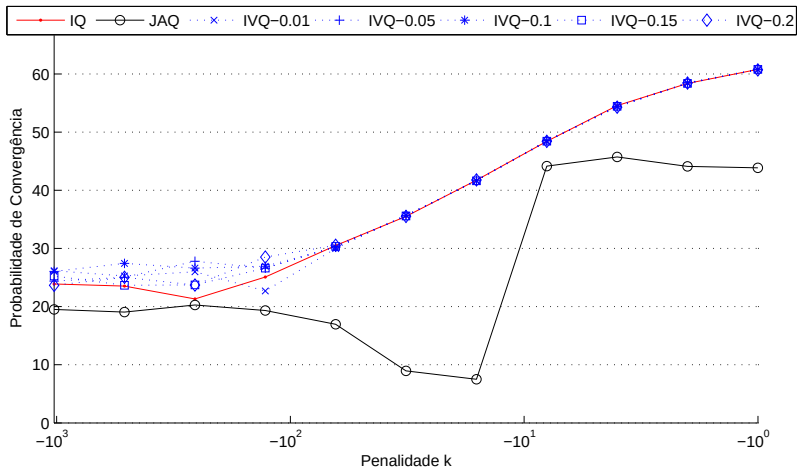
Figura 5.3: Convergência no *Penalty Game* com 2 agentes, $T = 16(0.9995^t)$ e $so = 2$

Quando aumentamos a exploração de novas ações (desaceleração no decaimento da temperatura T), o rendimento de todos os algoritmos melhora. Isto pode ser observado ao comparar as Figuras 5.2 e 5.3. Nestas Figuras, vemos que, ao aumentar a penalidade dos agentes, a probabilidade de convergência nos algoritmos IQ-Learning e IVQ-Learning chega a níveis próximos do 50% quando os algoritmos trabalham com $T = 16(0.999^t)$. Porém, estes mesmos algoritmos treinados com $T = 16(0.9995^t)$ têm uma probabilidade de convergência sempre maior que 60. Já o JAQ-Learning também se favoreceu com esta mudança na relação *exploration/explotation* da estratégia de seleção de ações, por exemplo se considerarmos valores da penalidade menores a 16 (ponto mais crítico deste

algoritmo), para $T = 16(0.999^t)$ a maior probabilidade de convergência alcançada foi inferior e próxima de 50 já para $T = 16(0.9995^t)$ a mesma foi 60%.



(a) Greedy

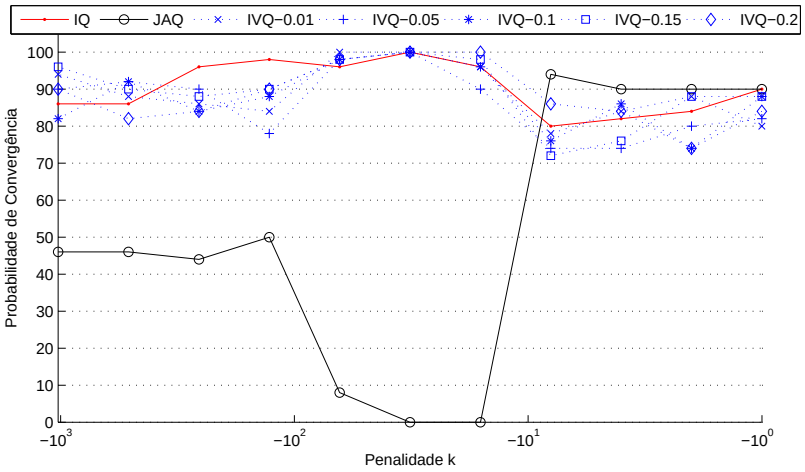


(b) Softmax

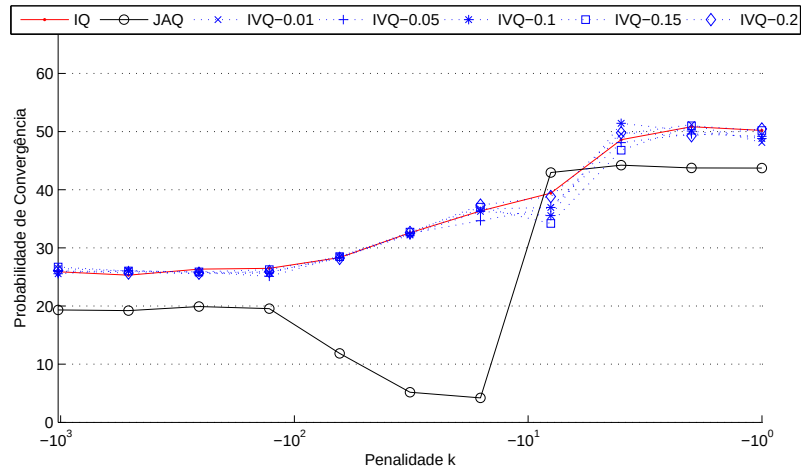
Figura 5.4: Convergência no *Penalty Game* com 2 agentes, $T = 16(0.9998^t)$ e $so = 2$

Seguindo com a análise, quando $T = 16(0.9995^t)$, a situação anterior descrita sobre a probabilidade de convergência dos algoritmos muda ligeiramente (Figura 5.4). Primeiro, em relação aos dois algoritmos com maior probabilidade de convergência, vemos que, como no caso anterior, esta mudança favoreceu a sua probabilidade de convergência. Porém, a diferença, em relação aos dois valores do decaimento de T anteriores, é que agora o mais favorecido foi o algoritmo IVQ-Learning em todas as suas versões, chegando a abrir uma vantagem de até 25 pontos percentuais sobre o IQ-Learning (vide penalidade $k = -256$). Ao analisar o comportamento de do JAQ-Learning, diferentemente da melhoria obtida para $T = 16(0.9995^t)$, o valor de $T = 16(0.9998^t)$ fez com que sua proba-

bilidade de convergência diminuísse chegando a níveis críticos para $k = -16$ e $k = -32$. Estes resultados trazem consigo uma idéia de que a relação *exploration/explotation* ótima, é diferente para cada paradigma.



(a) Greedy



(b) Softmax

Figura 5.5: Convergência no *Penalty Game* com 2 agentes, $T = 16(0.9999^t)$ e $so = 2$

Um outro resultado bastante interessante é mostrado na Figura 5.5. Nela, observamos que a probabilidade de convergência para valores de k próximos de zero diminui passando da perfeição (100%) a níveis inferiores a 90%. Porém, para penalidades muito grandes (valores de k muito negativos) a probabilidade de convergência continuou a aumentar chegando a níveis acima dos 90%. Ainda, o maior favorecido com esta nova estratégia de seleção de ações foi a algoritmo IQ-Learning que alcançou seus melhores níveis para estes valores de k, conseguindo pela primeira vez uma vantagem sobre todas as versões do IVQ-Learning (aproximadamente 5%). Para o algoritmo JAQ-Learning, novamente

vemos que este valor do decaimento da temperatura o prejudicou, fazendo com que sua probabilidade de convergência diminua ainda mais.

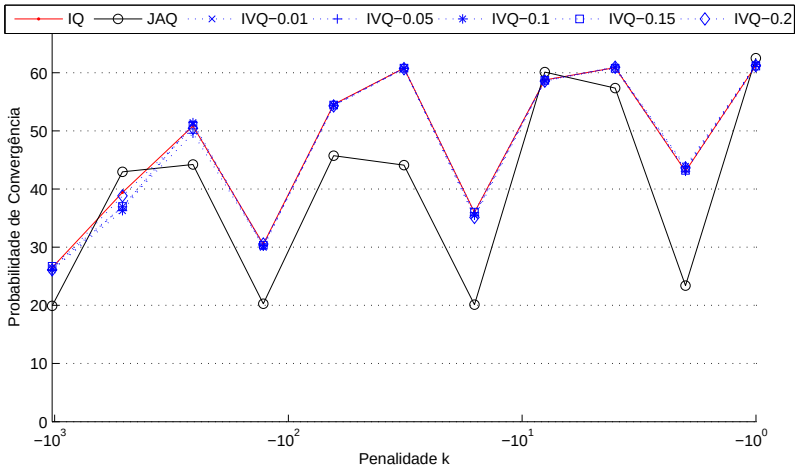


Figura 5.6: Melhores algoritmos no *Penalty Game* com 2 agentes para $so = 2$

Após estes resultados iniciais, observamos que, para todas as mudanças dos parâmetros, sempre existiu uma superioridade do IVQ-Learning sobre os outros algoritmos, chegando, em alguns casos, a ter uma diferença de até 25% na probabilidade de convergência, usando uma política gulosa, e de até 5% para uma política *softmax*. Porém, consideramos importante analisar as melhores versões de cada algoritmo, já que a influência dos parâmetros escolhidos não é a mesma para todos eles. Assim, a Figura 5.6 mostra uma comparação das melhores versões para cada algoritmo para cada uma das penalidades k. O programa considerado para $k = -2$ não será necessariamente o mesmo considerado para $k = -32$ por exemplo, isto se tratando do mesmo tipo de algoritmo. Um fato interessante a ser observado nesta figura é que as melhores versões dos algoritmos IQ-Learning e IVQ-Learning aplicadas ao jogo *Penalty Game*, com equilibrio sub-ótimo $so = 2$ são comparáveis. Isto se quisermos usar o algoritmo IQ-Learning e fazer uma análise de qual o melhor valor do decaimento da temperatura T. Porém, para este caso, recomendamos o algoritmo IVQ-Learning. Primeiro porque ele foi, na maioria das vezes, superior ao IQ-Learning e os seus melhores comportamentos foram no mínimo o mesmo que o melhor do IQ-Learning. Outro aspecto interessante, é que, no global, a desvantagem apresentada pelo algoritmo JAQ-Learning não é tão grande quando parecia ser ao analisarmos independentemente cada resultado obtido.

Como dito anteriormente, a análise feita na tentativa de ser o mais completa possível considerou um valor variável para a recompensa no equilíbrio sub-ótimo $so \in \{2,3,4,5,6\}$. Isto, com o intuito de analisar se a distância entre o equilíbrio ótimo e sub-ótimo terá influência relevante no rendimento dos diferentes paradigmas. Isto é, se considerarmos os equilíbrios de Nash dentro de um jogo como sendo picos de montanhas e os outros pontos como sendo vales. Ao diminuir a diferença entre o ótimo e o sub-ótimo, virtualmente estamos aumentando a profundidade dos vales (desde a perspectiva dos equilíbrios) e

teoricamente a dificuldade de chegar ao equilíbrio ótimo deve aumentar. Então, se compararmos, por exemplo, os resultados obtidos para $so = 2$ e para $so = 3$, vemos que o comportamento dos algoritmos é similar. Porém, é importante analisar estes resultados em detalhe para poder encontrar padrões na influência destes valores exercida sobre cada um dos algoritmos. Todas figuras correspondentes a estes resultados, podem ser analisadas no apêndice A.

Comparando as Figuras 5.2 e A.1 vemos que, aparentemente, os algoritmos, mesmo tendo um comportamento similar quando $so = 3$, apresentam um comportamento mais estável, isto é a curva de convergência é mais suave e ainda se mantém a tendência de diminuir de forma mais suave que quando $so = 2$. Também é possível observar que, para esta nova configuração do jogo e usando uma política gulosa, o algoritmo IQ-Learning diminuiu seus níveis de convergência para valores inferiores a 70% enquanto que o algoritmo $IVQ-Learning$ manteve seus melhores níveis próximo dos 80%. Ainda, a quantidade de representantes na faixa $[70\%, 80\%]$ para $k \leq 64$ aumentou de 8 quando $so = 2$ para 12 quando $so = 3$. Para o algoritmo JAQ-Learning, observamos que houve uma diminuição bem definida da capacidade de convergência. O comportamento relatado para a política gulosa é o mesmo para a política *softmax*, porém neste caso, as diferenças são menos marcadas posto que os índices de convergência também são menores.

Ao comparar as Figuras 5.3 e A.2, vemos que as observações feitas para $T = 16(0.999^t)$ são válidas também para $T = 16(0.99995^t)$. Porém, desta vez, não se observa melhora na capacidade de convergência ao trabalhar com estratégias de seleções de ações que dão maior prioridade à exploração de novas estratégias. Isto significa que, ao passar de $T = 16(0.999^t)$ para $T = 16(0.99995^t)$, diferentemente do acontecido quando $so = 2$, não houve melhora na probabilidade de convergência em nenhum dos três paradigmas testados.

Finalmente, se compararmos os melhores resultados para cada uma das implementações, os obtidos para $so = \{3, 4, 5, 6\}$ são os mesmos que para $so = 2$, ainda, a vantagem dos algoritmos IQ-learning e IVQ-learning sobre o JAQ-learning continua a existir e todos os algoritmos apresentam o mesmo comportamento.

Também, observamos que a distância entre o equilíbrio ótimo e sub-ótimo tem influência direta no rendimento dos diferentes paradigmas, porém a maior influência negativa é do JAQ-learning. Analisando estes resultados e os resultados preliminares explicados no inicio desta seção, podemos concluir que o JAQ-learning é mais eficiente na procura do equilibrio ótimo para políticas gulosas já que ao treinarmos com políticas *softmax*, e devido ao fato de que um agente aprendendo com o algoritmo JAQ-learning trabalha com o histórico das ações dos seus colegas, o agente perderá a capacidade de explorar as melhores estratégias já aprendidas, fazendo-se necessária uma mudança na forma de modelar o colega.

Em relação aos algoritmos IVQ-learning e IQ-learning, analisando também o conjunto de resultados obtidos (resultados preliminares e resultados analisados neste documento), vemos que, quando treinado com políticas Gulosas, o algoritmo IVQ-learning tem maior capacidade de convergência para o equilíbrio ótimo que o algoritmo IQ-learning, Porém, ao aumentar a capacidade de exploração de novas estratégias de solução (usando políticas *softmax* com temperatura decaindo no tempo), aumenta também a capacidade do último

de convergir ao equilíbrio ótimo, podendo chegar a valores comparáveis com os obtidos pelo IVQ-learning. Mas, é importante observar que se analisados ambos com os mesmos parâmetros (em todos os resultados obtidos durante o desenvolvimento da tese), o algoritmo IVQ-learning é superior.

Em relação ao parâmetro β do algoritmo IVQ-learning, observamos que são comparáveis no problema do *penalty game* com dois agentes, não sendo possível encontrar uma relação entre a capacidade de convergência e o valor deste parâmetro.

Climbing Game

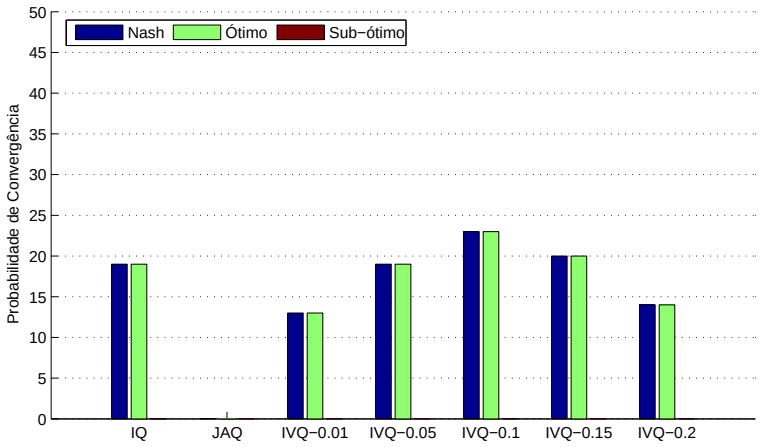
Os experimentos realizados no jogo *Climbing Game* têm as seguintes características:

1. Valor da temperatura na equação de *Boltzman* decai a cada jogo baseado em $T(t) = T(0) * decT^t$. Assim, se definiu um valor variável do decaimento da temperatura, $decT \in \{0.999, 0.9991, 0.9992, 0.9993, 0.9994, 0.9995, 0.9996, 0.9997, 0.9998, 0.9999\}$.
2. Foram testadas 5 versões do algoritmo $IVQ-Learning$ baseado num valor variável do coeficiente de influência, $\beta \in \{0.01, 0.05, 0.1, 0.15, 0.2\}$.
3. Cada versão implementada do algoritmo foi treinada 100 vezes durante 10000 iterações, fazendo assim um total de 1000 programas IQ-Learning, 1000 programas JAQ-Learning e 5000 programas IVQ-Learning avaliados neste jogo.

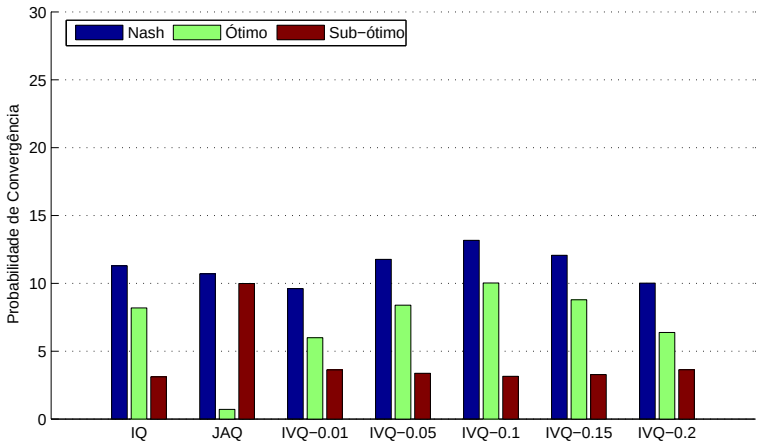
Da mesma maneira que foi feita para o *Penalty Game*, para a análise do rendimento de cada paradigma, após o treinamento, calculamos a probabilidade de cada programa convergir para os dois equilíbrios de Nash ótimos. Este cálculo foi feito de duas formas, a primeira considerando uma política de seleção de ações gulosa a partir dos valores aprendidos durante o treinamento, e a segunda considerando una política *softmax* onde a temperatura foi definida por $T = (\max(Q) - \min(Q))/2$.

Assim, observando a Figura 5.7, que mostra a probabilidade de convergência para os 7 algoritmos implementados no jogo *Climbing Game* com um valor do decaimento da temperatura de 0.999 ($T = 16(0.999^t)$). O primeiro aspecto importante em relação ao resultado obtido, é que, diferentemente ao resultado publicado por Claus e Boutelier [Claus & Boutilier 1998] e aos resultados iniciais obtidos durante o desenvolvimento desta tese, o algoritmo IQ-learning tem probabilidade diferente de zero de convergir ao equilibrio de Nash ótimo neste jogo. Este resultado leva a concluir que quanto mais gulosa a estratégia usada durante o aprendizado o algoritmo IQ-learning irá perder a possibilidade de convergir ao equilíbrio de Nash ótimo para o jogo Climbing Game. Por outro lado, é importante observar que neste jogo, durante todos os testes realizados,o algoritmo JAQ-learning jamais convergiu para o equilíbrio ótimo ao ser testado usando uma política gulosa para os testes. E, quando avaliado usando uma política *softmax*, a probabilidade de convergência é muito pequena.

Em relação aos resultados obtidos para o paradigma proposto nesta tese, especificamente o algoritmo IVQ-learning, os mesmos reforçam os resultados obtidos nos testes preliminares da tese dando a idéia de que seja com políticas gulosas ou com políticas *softmax* a nossa proposta tem a capacidade de convergir para o equilíbrio ótimo. Ainda, mesmo a política escolhida para o treinamento afetar de alguma forma a capacidade de



(a) Greedy



(b) Softmax

Figura 5.7: Convergência no *Climbing Game* com 2 agentes, $T = 16(0.999^t)$

convergência, nos experimentos realizados durante o desenvolvimento da tese, a probabilidade de convergência jamais chegou a ser zero como aconteceu com os paradigmas tradicionais.

Então, nos resultados mostrados na Figura 5.7, podemos observar que três das cinco versões da nossa proposta têm maior probabilidade de convergir para o equilíbrio de Nash ótimo neste jogo que o algoritmo IQ-learning. Ainda, a probabilidade de convergência da nossa proposta forma uma parábola invertida, dando a idéia de que existe um valor ótimo para o parâmetro β, e que no caso específico do jogo *Climbing Game* e o algoritmo sendo treinado com $T = 16(0.999^t)$, o mesmo seria 0.1. Observe ainda, que ao considerarmos a política *softmax* no calculo da probabilidade de convergência, a probabilidade do JAQ-learning de convergir a um equilíbrio de Nash chega a ser próxima das probabilidades do IQ-learning e do IVQ-learning, só que diferentemente destes algoritmos, o mesmo apresenta uma maior probabilidade de convergência ao equilibrio sub-ótimo que ao equilíbrio ótimo.

Ao aumentar o valor do parâmetro de decaimento da temperatura para os algoritmos para $T = 16(0.9991^t)$, quer dizer que a política de seleção de ações dará maior prioridade a exploração de novas ações, vemos que o comportamento dos algoritmos se mantém. No caso da política gulosa, o IQ-learning e o IVQ-learning só tem probabilidade de convergir para o equilíbrio de Nash ótimo e no caso da política *SoftMax*, a probabilidade de convergir para o sub-ótimo existe, porém existirá uma maior probabilidade de convergência para o ótimo. Ainda, pode se observar que, no geral, a probabilidade de convergência dos algoritmos diminuiu quando se passou de $T = 16(0.999^t)$ para $T = 16(0.9991^t)$.

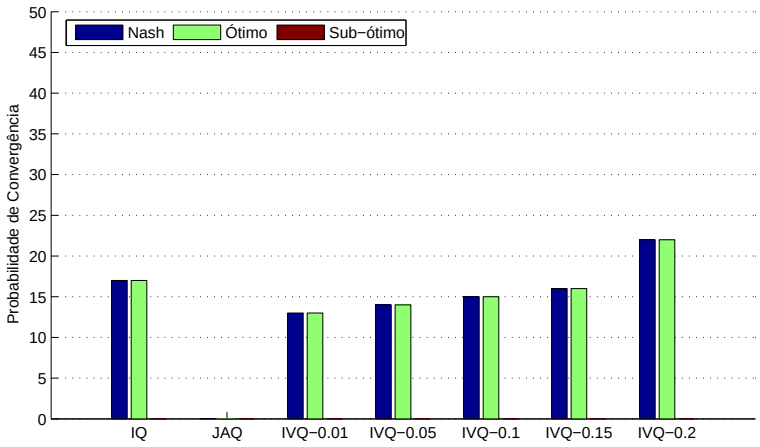
Já para $T = 16(0.9992^t)$, como mostrado na Figura A.21 do apêndice A, as probabilidades de convergência continuaram a diminuir, porém o comportamento em forma de parábola invertida voltou, mostrando que a melhor opção para estes parâmetros é o IVQ-learning com $\beta = 0.1$ (IVQ-0.1) e o mesmo apresenta melhores resultados que o IQ-learning.

Quando se avaliou os algoritmos usando $T = 16(0.9993^t)$, observou-se que novamente as probabilidades de convergência dos algoritmos aumentaram e que desta vez o melhor valor do parâmetro β foi 0.01.

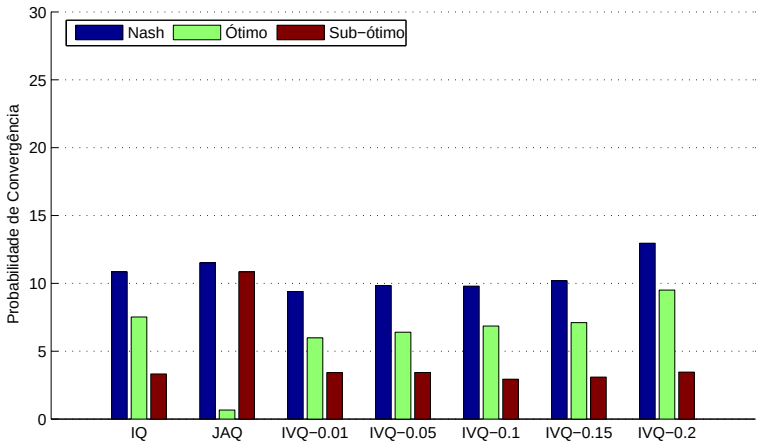
Já, quando se fizeram os experimentos com $T = 16(0.9994^t)$, a probabilidade de convergência continuou a aumentar, e desta vez se voltou a observar que três versões da nossa proposta se tiveram melhor rendimento que o algoritmo IQ-learning.

Em relação aos resultados para $T = 16(0.9995^t)$, se observou que a diferença entre a melhor versão do algoritmo IVQ-learning (IVQ-0.1) e o IQ-learning aumentou. Neste sentido, a probabilidade de convergência do primeiro foi quase o dobro da probabilidade do segundo.

Estes resultados mantiveram um padrão, para valores da temperatura entre $T = 16(0.999^t)$ e $T = 16(0.9997^t)$, porém quando se testou com $T = 16(0.9998^t)$, observou-se uma mudança radical, com estes valores a probabilidade de convergência a um equilibrio de Nash aumentou, porém existe uma grande probabilidade de convergir ao equilíbrio sub-ótimo. Neste ponto, é importante lembrar que no jogo *penalty Game* o melhor resultado foi obtido para $T = 16(0.9998^t)$, assim, isto da uma idéia de que a dificuldade de chegar ao equilibrio de Nash ótimo no *Penalty Game* é comparável à dificuldade de chegar ao equi-



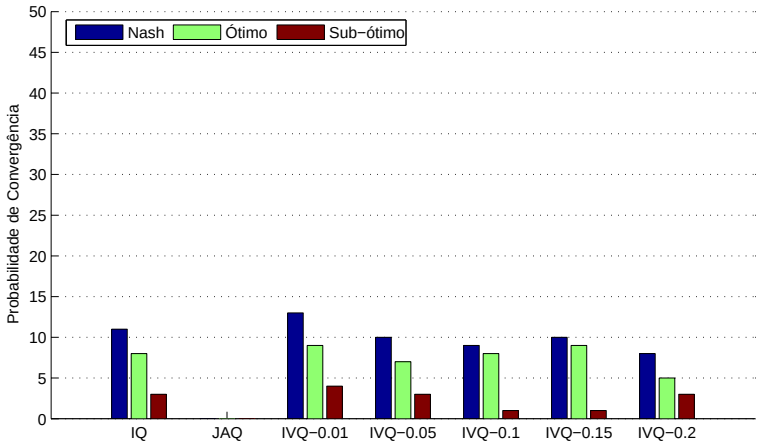
(a) Greedy



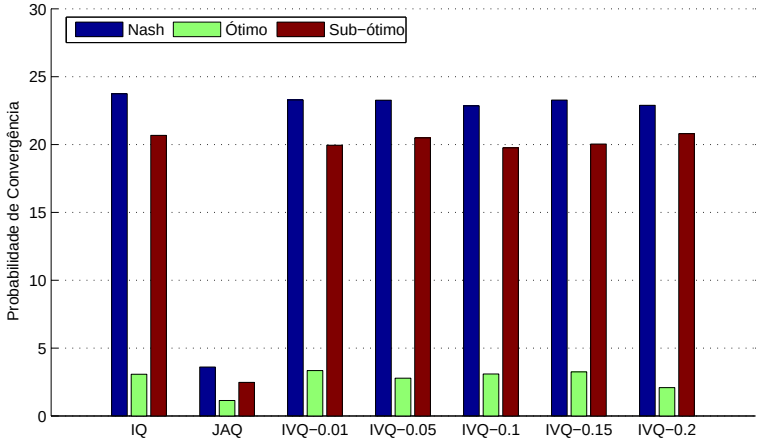
(b) Softmax

Figura 5.8: Convergência no *Climbing Game* com 2 agentes, $T = 16(0.9991^t)$

líbrio de Nash sub-ótimo no *Climbing Game* e que a melhor política de seleção de ações é a baseada no valor de $T = 16(0.9998^t)$.



(a) Greedy



(b) Softmax

Figura 5.9: Convergência no *Climbing Game* com 2 agentes, $T = 16(0.9999^t)$

Do mesmo jeito que aconteceu com os algoritmos aplicados ao *Penalty Game*, no *Climbing Game*, quando se treinou com $T = 16(0.9999^t)$ se obtiveram os piores resultados e por tanto os piores índices de convergência. Estes resultados podem ser verificados comparando a Figura 5.9 com as anteriores.

Já falando do tempo de convergência, e como este se vê influenciado pelo parâmetro T da equação de Boltzman usada para implementar a política de seleção de ações, observou-se que todos os algoritmos tiveram o mesmo comportamento. Isto é, os tempos de convergência necessários para todos ele foi praticamente o mesmo e foi aumentando a medida que o fator de decaimento da temperatura aumentava também. Este comporta-

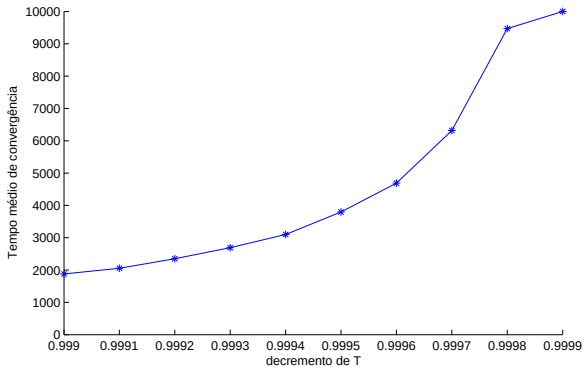


Figura 5.10: Tempo de Convergência ao ótimo no *Climbing Game* com 2 agentes

mento é mostrado na Figura 5.10.

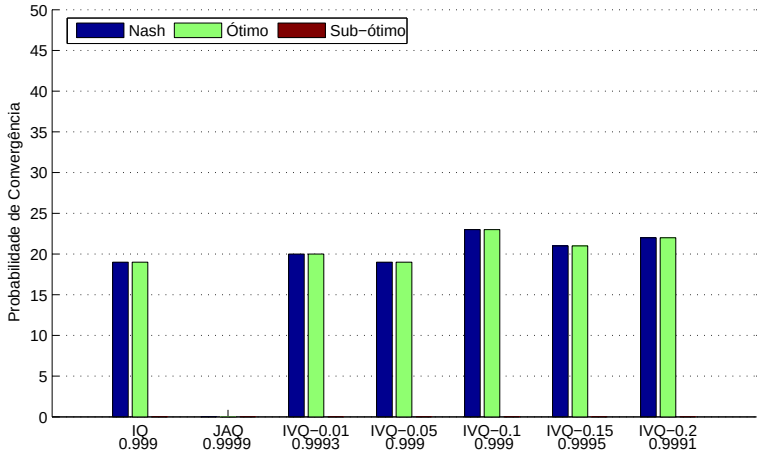
Finalmente, em relação ao *Climbing Game* com dois agentes, do mesmo jeito que se fez com o *Penalty Game*, se comparou as melhores versões de cada um dos paradigmas (desde o ponto de vista da temperatura de equação de Boltzman). Os resultados destas comparações são mostrados nas Figuras 5.11 e 5.12.

Na Figura 5.11 é mostrada a comparação desde o ponto de vista da convergência ao equilíbrio de Nash ótimo. Isto é, se consideraram como melhores aqueles algoritmos que tiveram maior probabilidade de convergência ao equilíbrio de Nash ótimo se m importar sua capacidade de convergência a qualquer equilíbrio de Nash. Observe que, todas as versões do paradigma IVRL proposto nesta tese tiveram melhor desempenho que o algoritmo IQ-learning, tanto considerando uma política *greedy* na hora da avaliação como uma política *softmax*.

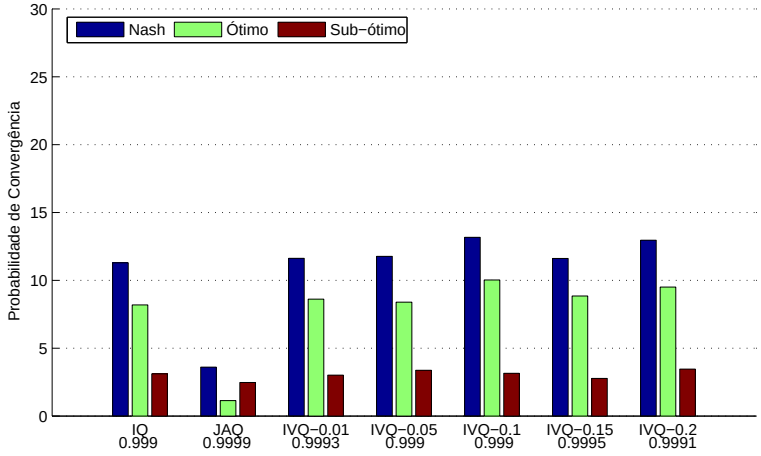
Já na Figura 5.12 são mostrados os algoritmos que tiveram a maior probabilidade de convergência a qualquer equilíbrio de Nash. É importante observar aqui que o melhor algoritmo nestes testes foi um algoritmo IVQ-learning, especificamente foi o treinado com $\beta = 0.1$. O mesmo garante a maior convergência a qualquer equilíbrio de Nash ($T = 16(0.9998^t)$) e também ao equilíbrio de Nash ótimo $T = 16(0.999^t)$

5.1.2 Jogos com 3 agentes

Como dito anteriormente, Além dos testes com 2 agentes, se realizaram também experimentos usando 3 agentes, isto para conhecer a capacidade de convergência de nossa proposta quando testada em ambientes mais complexos e com mais agentes em comparação com os paradigmas tradicionais. Assim, os resultados obtidos para os jogos *Penalty Game* e *Climbing Game* para 3 são analisados a continuação, e, as figuras correspondentes a esta análise são mostradas no apêndice A.

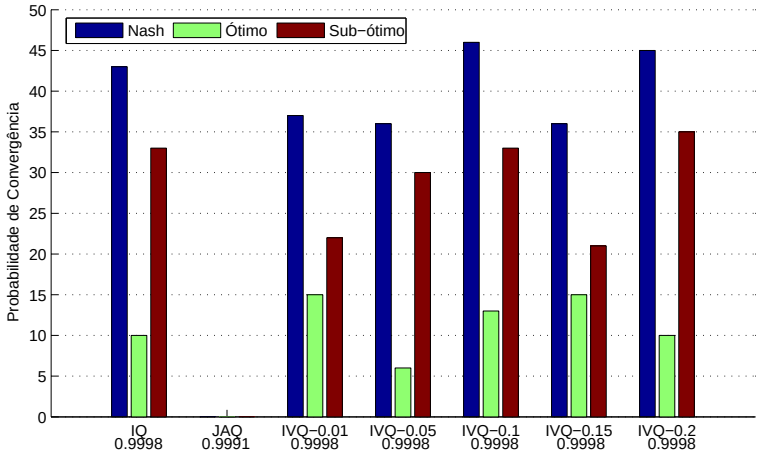


(a) Greedy

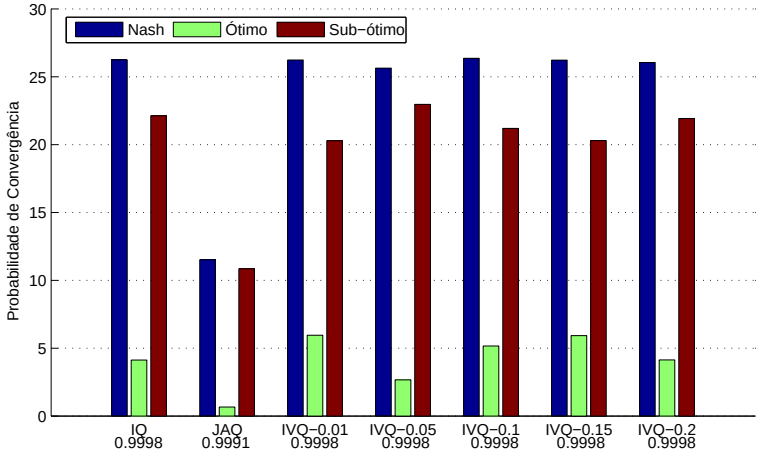


(b) Softmax

Figura 5.11: Melhores algoritmos no *Climbing Game* com 2 agentes (equilíbrio ótimo)



(a) Greedy



(b) Softmax

Figura 5.12: Melhores algoritmos no *Climbing Game* com 2 agentes (equilíbrio de Nash)

Penalty Game

Para os testes realizados no jogo *Penalty Game* com 3 agentes, foram usados exatamente os mesmos critérios, parâmetros e variações usados nos testes com 2 agentes. Durante a compilação dos resultados observou-se que o comportamento dos algoritmos, em relação à variação do parâmetro *so* do jogo, era o mesmo que nos jogos com 2 agentes. Isto é, ao mudar o valor de *so* a diferença de probabilidades de convergência dos algoritmos não mudava significativamente e a mesma ia diminuindo para valores de *so* próximos da recompensa ótima (10).

A primeira observação a ser feita e relação aos resultados obtidos para $T = 16(0.999^t)$ e $so = 2$, é que após inserir mais um agente no jogo, a probabilidade de convergência diminuiu; e o principal prejudicado foi o algoritmo JAQ-learning que perdeu totalmente a capacidade de convergir para o ótimo quando $k \leq -4$ para a política gulosa; e chegou a níveis muito próximos de zero quando $k \leq -16$ para a política softmax.

Uma segunda observação é que se abriu uma diferença entre o IQ-learning e as diferentes versões do IVQ-learning. Neste sentido pode se observar que sempre existe pelo menos uma versão do paradigma proposto nesta tese que têm maior probabilidade de convergir para o equilíbrio de Nash ótimo. Chegando a ser está diferença de até aproximadamente 20 pontos percentuais.

Em relação aos algoritmos usando $T = 16(0.9995^t)$, podemos observar que, diferentemente dos resultados anteriores, o algoritmo IQ-learning e as diferentes versões do IVQ-learning melhoraram sua probabilidade de convergência em relação à que tinham quando testados com 2 agentes. O JAQ-learning novamente perdeu sua capacidade de convergência. Ainda, novamente, houve uma diferença notada entre a capacidade de convergência do IVQ-learning e o IQ-learning, sendo que o IVQ-learning passou a ter melhores resultados.

A situação anterior continua a se apresentar para $T = 16(0.9998^t)$. Ainda, é importante observar que a probabilidade de convergência da nossa proposta foi sempre maior que 90%.

Um fato muito interessante é que, do mesmo jeito que aconteceu para 2 agentes, ao treinar com $T = 16(0.9999^t)$ os algoritmos diminuíram sua capacidade de convergência. Porém, podemos observar que a diferença entre o IVQ-learning e o IQ-learning se fez mais estável, mantendo quase para todas as versões do primeiro uma diferença de aproximadamente 10% com o segundo.

Do mesmo modo que foi realizado com 2 agentes, fizemos um gráfico das probabilidades de convergência ao compilar a lista das melhores versões dos algoritmos, em relação à sua capacidade de convergência para o equilíbrio de Nash. É importante observar que, embora o comportamento de todos os algoritmos continua a ser o mesmo, a nossa proposta já obtém melhores valores que o IQ-learning, chegando esta diferença a um máximo de aproximadamente 5%.

Climbing Game

Do mesmo modo no *Penalty Game*, neste jogo para 3 agentes foram usados os mesmos critérios e parâmetros usados para sua versão de 2 agentes. Os resultados obtidos refletem

uma característica particular deste jogo, a sua complexidade. Ela aumenta à medida que aumentamos o número de agentes, pois a quantidade de punições que afetam diretamente a escolha das ações que representam o equilíbrio ótimo e sub-ótimo, neste jogo, cresce exponencialmente. Assim podemos verificar que os algoritmos, ao serem testados neste jogo nas suas versões de três e quatro agentes, foram submetidos a situações extremas que permitem ressaltar suas diferenças ou fixar suas semelhanças. As figuras correspondentes a este análise também podem ser encontradas no apêndice A.

Nos resultados para três agentes, é possível observar que a capacidade de convergência dos algoritmos IQ-learning e JAQ learning é bastante diminuída. Por exemplo, do ponto de vista da política gulosa, o algoritmo só converge ao equilíbrio ótimo quando $T = 16(0.9999^t)$ e ainda com uma probabilidade igual a 3%. Isto significa que, dos 1000 programas implementados para este algoritmo, só 3 convergiram para a solução ótima. No caso do algoritmo IVQ-learning, a probabilidade de convergência é também bastante afetada. Embora sempre haja pelo menos uma versão deste algoritmo que consiga convergir para o equilíbrio ótimo, podemos calcular que, dos 5000 programas implementados para esta proposta, só 102 conseguiram convergir ao equilíbrio ótimo.

Outro detalhe muito importante observado ao analisar estes resultados é que, novamente, quando $T = 16(0.9998^t)$ (Figura A.41), a probabilidade de convergência ao equilíbrio sub-ótimo tomou valores diferentes de zero. Porém, é importante destacar uma diferença com os valores obtidos para 2 agentes. O IVQ-learning foi muito superior aos outros algoritmos chegando quase a duplicar sua capacidade de convergência (IVQ-0.2 Vs. IQ).

Em relação ao tempo de convergência necessário para atingir o equilíbrio ótimo, é importante destacar que estes tempos estão diretamente relacionados ao ponto em que o valor da temperatura T atinge um valor igual o inferior a 2. Também, o cálculo do tempo de convergência foi feito tomando como parâmetro a variação que os valores aos quais as funções Q dos agentes convergiram teve no tempo. Isto porque, diferentemente de muitos processos de decisão Markovianos, em jogos estocásticos e também na sua versão simplificada (jogos repetitivos) é virtualmente impossível saber quais os valores reais aos que deveriam convergir as funções Q dos agentes dentro do jogo.

Em relação à comparação feita as melhores versões dos algoritmos para o jogo *Climbing Game* de 3 agentes, podemos ver que os resultados obtidos são parecidos aos obtidos com 2 agentes, podendo destacar as características que se repetem:

- A probabilidade de convergência ao equilíbrio de Nash ótimo é maior para valores menores do decaimento da temperatura na equação de Boltzman.
- As melhores versões dos programas IVQ-learning são superiores à melhor versão do IQ-Learning.
- Geralmente as melhores versões dos algoritmos desde o ponto de vista da sua capacidade de convergência a qualquer equilíbrio de Nash dentro do jogo são aquelas onde $T = 16(0.9998^t)$.
- Quando a capacidade de convergir ao equilíbrio sub-ótimo aumenta, conseqüentemente, a probabilidade de convergir ao equilíbrio ótimo diminui ou desaparece.

5.1.3 Jogos com 4 agentes

Para finalizar a análise realizada aos diferentes paradigmas, usando jogos repetitivos, se analisa a continuação os resultados obtidos para jogos com 4 agentes os quais são apresentados no apêndice A. É importante observar que os parâmetros foram os mesmos definidos para 2 e 3 agentes.

Penalty Game

Com respeito aos resultados obtidos com o jogo *Penalty Game* com 4 agentes, é importante notar que os resultados obtidos para $so \in \{3,4,5,6\}$ são análogos aos obtidos com 2 e 3 agentes.

Em geral, podemos considerar que os resultados mantiveram-se na mesma linha, quer dizer, validando as conclusões preliminares obtidas dos resultados com 2 e 3 agentes. Porém, um aspecto importante é a instabilidade observada na probabilidade de convergência para o equilíbrio de Nash ótimo. Uma possível explicação para este fato é que quanto maior a quantidade de agentes maior o tempo necessário para convergir e conseqüentemente maiores deveriam ser o número de iterações durante o treinamento e talvez também devesse ser maior a temperatura inicial na equação de Boltzman usada para implementar a política de seleção de ações *SoftMax*.

Climbing Game

Após analisar os resultados obtidos para o jogo *Climbing Game* com 4 agentes, novamente os resultados preliminares foram confirmados. É possível, então, concluir que o algoritmo IVQ-Learning é o que melhor se comportou neste jogo mostrando maior capacidade de atingir o equilíbrio ótimo num jogo tão complexo para os agentes. Ainda, nossa proposta também se revelou como a melhor opção para atingir um equilíbrio de Nash, mesmo sendo sub-ótimo.

Um outro aspecto importante é a relação ao balanceamento entre exploração e explotação (*exploration/explotation*) na política de seleção de ações usada durante o treinamento. Observamos a mesma influencia em todos os algoritmos, porém, o mais influenciável é o JAQ-Learning. Isso acontece porque este tipo de algoritmo é muito dependente do modelo formalizado para os outros agentes. Ao priorizar a exploração, o modelo construído dos outros agentes pode fazer com que eles escolham de forma equiprovável todas as suas ações, evidentemente prejudicando sua capacidade de aprendizado.

Finalmente, é importante destacar que a vantagem de usar nosso paradigma com 4 ou mais agentes, adquire mais força. Isto ocorre porque os algoritmos implementados seguindo a nossa proposta são melhores, com relação à sua convergência para o equilíbrio de Nash ótimo e para qualquer equilíbrio de Nash.

5.2 Coordenação em jogos estocásticos

Nos experimentos envolvendo jogos estocásticos, testamos quatro versões de algoritmos baseadas no Q-Learning. As duas primeiras implementam o paradigma de aprendi-

zado independente, a terceira o paradigma de aprendizado de ações conjuntas e a última o paradigma proposto nesta tese de doutorado. Esses testes tiveram como objetivo avaliar a capacidade de convergência quando os agentes aprendem através de recompensas demoradas. Consideramos que o comportamento dos paradigmas foi extensivamente avaliado (em detalhes) na Seção anterior, que possibilitou analisar como os diferentes paradigmas se comportam em relação às mudanças de parâmetros, estratégia e do modelo do ambiente. Então, aqui é simplesmente necessário verificar se os algoritmos continuam sendo aplicáveis, caso o problema se torne mais complexo.

O primeiro algoritmo (Aprendizado Independente A) considera que o estado do agente é simplesmente sua posição, já o outros três algoritmos (Aprendizado Independente B, aprendizado de ações conjuntas e aprendizado com valores de influência) consideram o estado como sendo as posições dos dois agentes.

Nos testes, cada algoritmo foi executado três vezes para cada valor de penalidade k $(0 \leq k \leq 15)$ e usando cinco diferentes valores de decaimento da temperatura T da política softmax $(0.99t, 0.995t, 0.999t, 0.9995t, 0.9999t)$. Cada estratégia resultante (960 estratégias, sendo 3 para cada algoritmo com penalidade k e um certo índice de decaimento da temperatura T) foi testada 1000 vezes.

Foram escolhidos diferentes penalidades k para testar a capacidade de cada algoritmo de convergir ao equilíbrio ótimo. Já os diferentes índices de decaimento da temperatura T foram escolhidos para avaliar a influência da exploração de novas soluções e a exploração do conhecimento adquirido.

Calculamos a média estatística do percentual de vezes que a posição $(3,3)$ foi atingida pelos agentes ao mesmo tempo, usando cada tipo de algoritmo com uma certa penalidade k e um certo índice de decaimento da temperatura. A Figura 5.13 mostra a probabilidade de chegar na posição $(3,3)$ com $\alpha = 1$, $\lambda = 0.1$, $\beta = 0.1$ e $T = 0.99t$ utilizando os quatro algoritmos. Nesta Figura, podemos observar que o algoritmo que considera o estado como sendo só a posição do agente que está aprendendo (Aprendizado Independente A) não tem condições de chegar na posição $(3,3)$. Porém, como é mostrado na mesma Figura, ele pode chegar na posição $(1,3)$. Isto acontece porque a posição $(1,3)$ não tem penalidades.

Também observamos que, para este problema, o algoritmo JAQ-Learning tem a menor probabilidade de convergência para a posição $(3,3)$. Este comportamento se repete para os outros índices de decaimento da temperatura (Figuras 5.14 a 5.15).

Com estes experimentos, podemos observar que o algoritmo de aprendizado independente B e o algoritmo de aprendizado com valores de influência têm quase o mesmo comportamento. Porém, quando a exploração de novas soluções aumenta, a probabilidade de atingir o equilíbrio ótimo diminui para o aprendizado independente B e aumenta par o aprendizado com valores de influência. Também pode ser observado que em alguns momentos o algoritmo de aprendizado de ações conjuntas não converge para um equilíbrio de Nash e em outros momentos converge para políticas que não deixam os agentes terminarem o jogo.

Como mostrado na Figura 5.15, quando a exploração de novas soluções aumenta, o algoritmo de aprendizado independente B perde a capacidade de convergir para as posições $(1,3)$ e $(3,3)$.

Nas Figuras 5.13 e 5.15, pode ser visto que, quanto maior for a exploração de novas

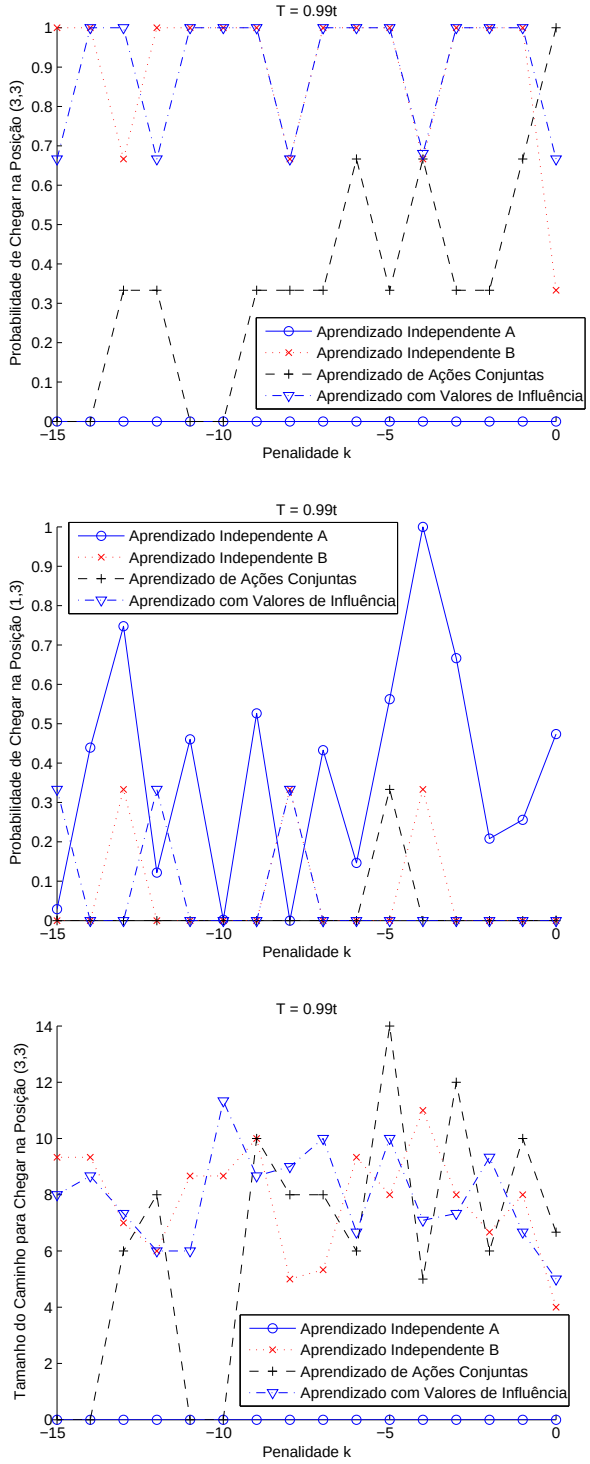


Figura 5.13: Rendimento dos algoritmos com $\alpha = 1$, $\lambda = 0.1$, $\beta = 0.1$ e $T = 0.99t$

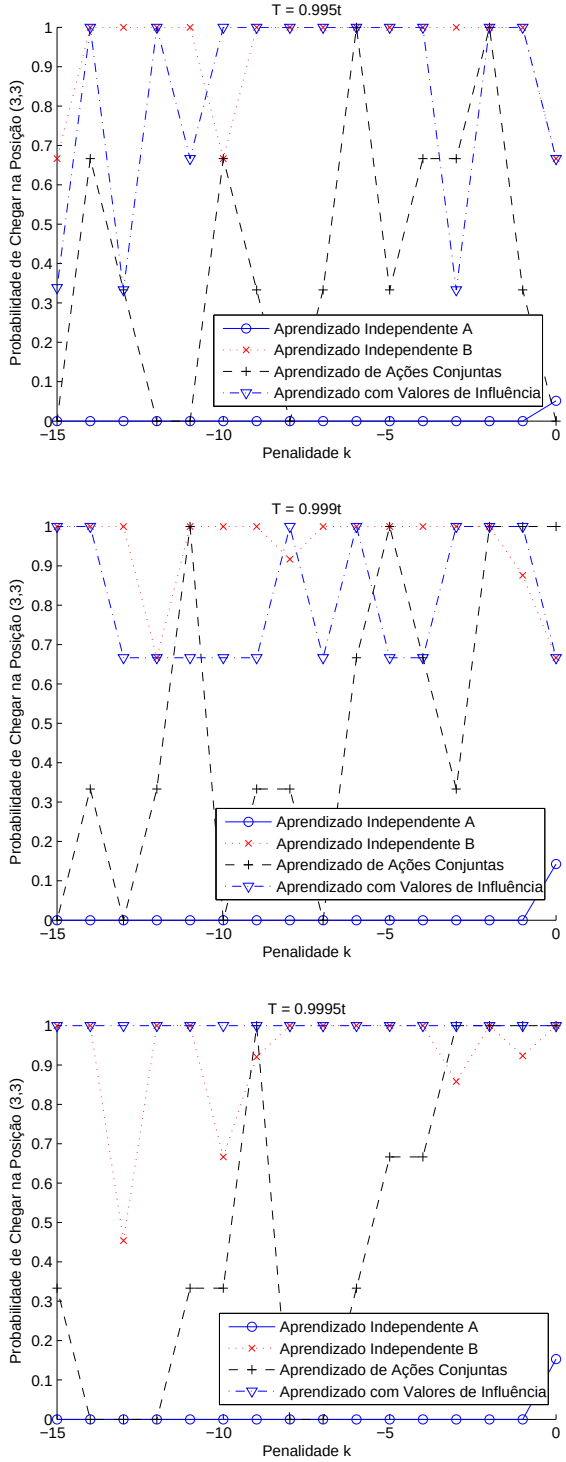


Figura 5.14: Probabilidade de chegar na posição $(3,3)$ para $\alpha = 1$, $\lambda = 0.1$, $\beta = 0.1$

soluções, menor será o tamanho do caminho para chegar na posição $(3,3)$. Isto mostra que a probabilidade de se chegar ao equilíbrio ótimo aumenta. É importante observar que o algoritmo IVQ-Learning tem a maior probabilidade de convergir para o equilíbrio ótimo. É possível chegar a esta conclusão observando conjuntamente a probabilidade de chegar na posição $(3,3)$ e a média do número de passos necessários para chegar na mesma.

5.3 Colaboração em jogos estocásticos

Finalmente, baseados na conjectura de que o paradigma proposto nesta tese teria capacidades de auto-organização, fizemos testes para avaliar a existência ou não desta propriedade, que, a nosso conhecimento, jamais foi reportada em algoritmos de aprendizado por reforço.

Inicialmente, testamos se um único agente aprendendo por reforço tem a capacidade de aprender a resolver esta categoria de jogos. Assim, implementamos o algoritmo Q-learning tradicional com um agente, que foi treinado durante 2000 épocas, com parâmetros $\alpha = 0.1$, $\gamma = 0.1$ e $T = 0.3$. Após a etapa de treinamento, testamos o conhecimento do agente em relação ao jogo. Com este intuito, usamos uma política gulosa para os testes (a ação com o maior valor de Q é a escolhida em cada estado) e a estratégia resultante é mostrada na Figura 5.16.

A Figura 5.16 mostra a estratégia do agente na forma de um grafo direcionado onde os vértices representam as plataformas dentro do jogo e as arestas representam o tempo discreto em que o agente passou de uma plataforma a outra. Como observado nesta Figura, o agente evoluiu para o comportamento ótimo dentro do jogo (completando o jogo em 10 tempos discretos), estes resultados mostram que o modelo usado no jogo é apropriado para resolver o problema usando algoritmos de aprendizado por reforço.

Assim, para poder testar a capacidade dos algoritmos de aprendizado por reforço, nos diferentes paradigmas, de convergir para uma estratégia colaborativa ótima, o mesmo problema foi implementado usando dois agentes. Notamos que, neste jogo, quando se usa dois agentes, a estratégia ótima deve levar só 4 tempos discretos para que os agentes terminem o jogo. Isto porque, quando ambos agentes iniciam o jogo na plataforma 0, um deles deverá tomar a ação "direita"e o outro "esquerda", subdividindo automaticamente o problema em dois sub-problemas e conseqüentemente cada agente exploraria só duas fontes de comida (figure 5.17).

Assim, o IQ-Learning, JAQ-Learning e o IVQ-Learning foram implementados para resolver este problema em 20000 épocas de treinamento. Também, cada algoritmo foi treinado 10 vezes com 3 valores diferentes da constante $\alpha \in \{0.05, 0.1, 0.15\}$. Para o coeficiente de influência foram usados 5 valores diferentes com o intuito de verificar sua influência, isto é $\beta \in \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$. Isto significa que implementamos 8 algoritmos ao todo, sendo cada um deles treinado 10 vezes. O parâmetro γ foi fixado para todos os algoritmos em 0.05.

A Figura 5.18 mostra a comparação desses oito algoritmos do ponto de vista do número de passos que os agentes treinados levaram para resolver o jogo das pontes. Este valor foi calculado considerando a média de 100 testes realizados para cada algoritmo implementado e para cada parâmetro α. Como dito anteriormente, na estratégia ótima,

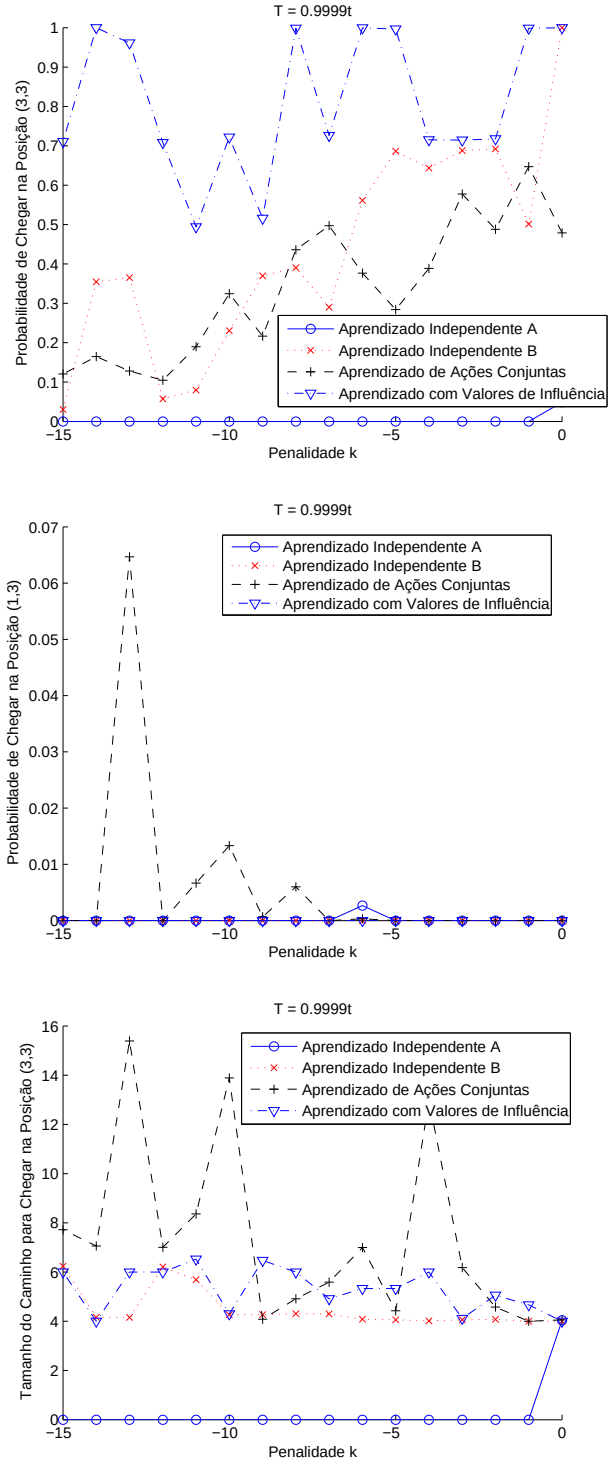


Figura 5.15: Rendimento dos algoritmos com $\alpha = 1$, $\lambda = 0.1$, $\beta = 0.1$ e $T = 0.9999t$

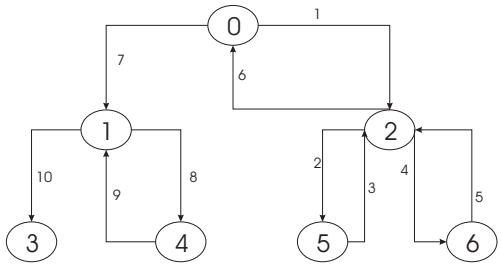
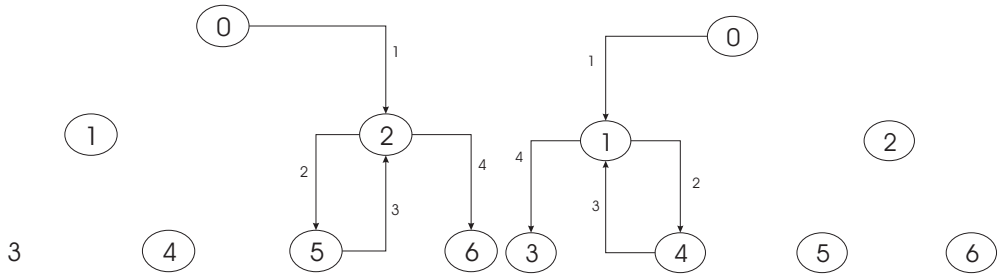


Figura 5.16: Estratégia de um Agente Aprendendo por Reforço no Jogo das Pontes



(a) Agente 1 (b) Agente 2

Figura 5.17: Estratégia Ótima de 2 Agentes no Jogo das Pontes

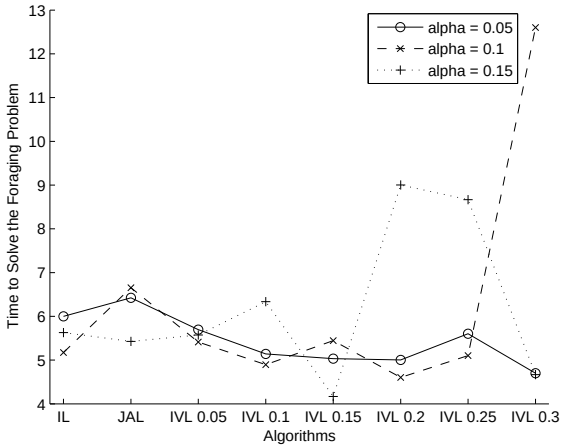


Figura 5.18: Tempo para 2 Agentes Resolver o Jogo das Pontes

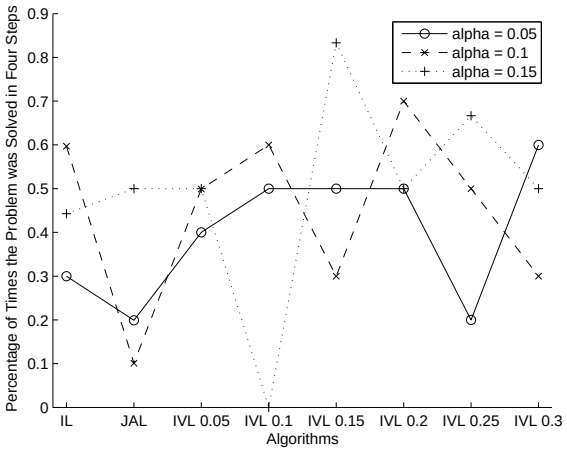
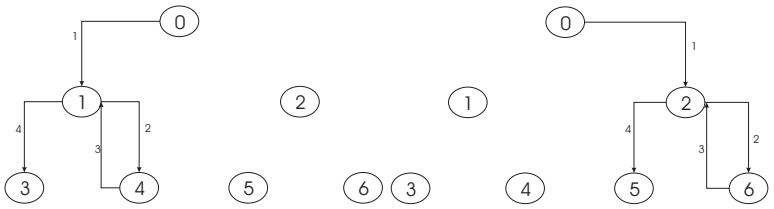


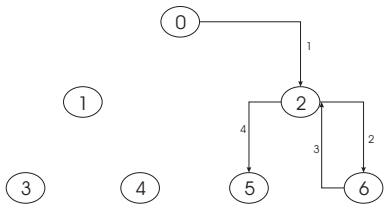
Figura 5.19: Probabilidade de Resolver o Jogo das Pontes em 4 passos

o número de passos deve ser quatro. Neste sentido, observamos que o paradigma proposto nesta tese de doutorado, com parâmetros $\beta = 0.15$ e $\alpha = 0.15$, obteve o melhor rendimento.

A Figura 5.18 mostra a média do número de passos que os agentes implementados com cada um dos algoritmos avaliados levou para resolver o problema. Convém ressaltar que, em alguns testes, os agentes convergiram para a estratégia e em outros testes não. A Figura 5.19 mostra a probabilidade de convergência para o ótimo, ou seja, a porcentagem do número de vezes que cada algoritmo conseguiu convergir para a solução ótima. Novamente, podemos observar que o algoritmo implementado usando o paradigma proposto nesta tese teve o melhor rendimento (IVQ-Learning com $\beta = 0.15$ e $\alpha = 0.15$).

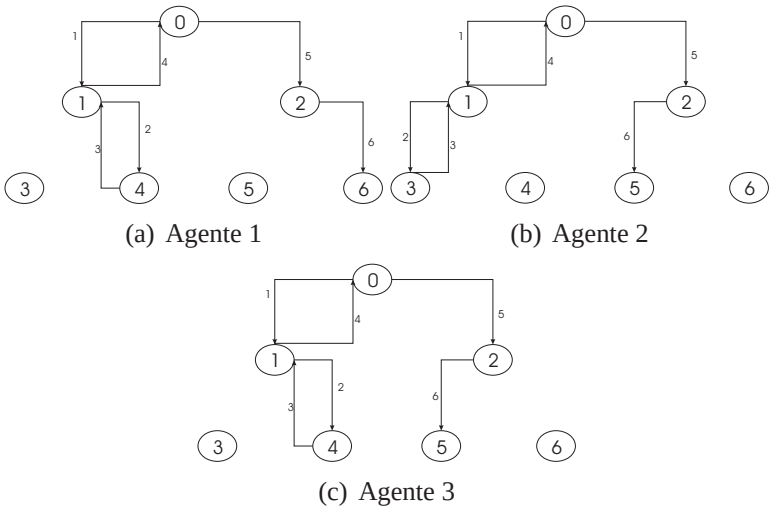


(a) Agente 1 (b) Agente 2



(c) Agente 3

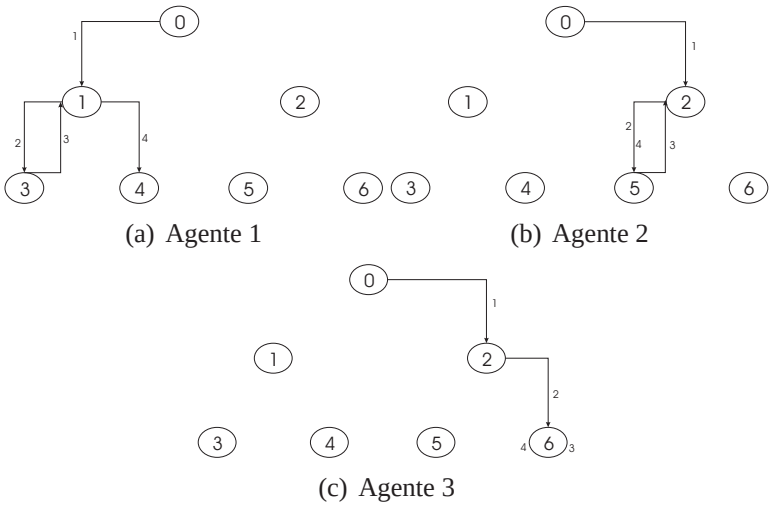
Figura 5.20: Estratégia do IQ-learning: 3 agentes, $\alpha = 0.1$, $\gamma = 0.1$ e $T = 0.3$



(a) Agente 1 (b) Agente 2

(c) Agente 3

Figura 5.21: Estratégia do JAQ-learning: 3 agentes, $\alpha = 0.1$, $\gamma = 0.1$ e $T = 0.3$



(a) Agente 1 (b) Agente 2

(c) Agente 3

Figura 5.22: Estratégia do IVQ-learning: 3 agentes, $\alpha = 0.1$, $\gamma = 0.1$, $\beta = 0.1$ e $T = 0.3$

Para poder analisar o rendimento da nossa proposta com três agentes, em comparação com os paradigmas tradicionais, todos foram treinados com os parâmetros $\alpha = 0.1$, $\gamma = 0.1$, $\beta = 0.2$ e $T = 0.3$. As Figuras 5.20, 5.21 e 5.22 mostram as estratégias gulosas obtidas usando os valores aprendidos pelos agentes após 75000 épocas de treinamento.

Como mostrado na Figura 5.20, os agentes implementando aprendizado independente não conseguiram colaborar entre eles para resolver a tarefa. A inclusão de um terceiro agente para este paradigma mostrou-se desnecessária. Pode ser observado que os agentes 2 e 3 fazem exatamente o mesmo, o que leva a concluir que a presença de um deles é desnecessária. Por outro lado, os agentes que aprenderam usando o paradigma de aprendizado de ações conjuntas convergiram para uma estratégia não ótima (Figura 5.21). Os agentes usando este paradigma precisam de 6 passos para resolver a tarefa. Novamente, a inclusão do terceiro agente gerou uma redundância no sistema tornando o mesmo desnecessário.

Finalmente, os algoritmos implementados usando o paradigma proposto nesta tese, convergiram para a estratégia de solução colaborativa ótima para o caso de três agentes colaborando no jogo das pontes. Como mostrado na Figura 5.22, inicialmente, os agentes, automaticamente, dividem o problema em dois sub-problemas, o primeiro para ser resolvido pelo primeiro agente, e o segundo pelos outros dois agentes. Logo a seguir, os agentes 2 e 3 dividem sua sub-tarefa em duas sub-subtarefas. Assim, como pode ser observado, os agentes que usaram o IVQ-learning realmente estão colaborando e dividindo automaticamente as tarefas dentro do jogo.

Analisando os resultados obtidos para jogos colaborativos, após observar que um dos algoritmos IVQ-learning tem probabilidade de convergência ao ótimo de aproximadamente 90% e que em testes com 3 agentes o único algoritmo que conseguiu executar um comportamento colaborativo foi o IVQ-learning, concluímos que esta característica faz parte dos algoritmos implementados com o paradigma IVRL. Ainda, tal como em todo algoritmo de aprendizado de máquina, especialmente naqueles que trabalham com colaboração, a escolha dos parâmetros é relevante na qualidade da solução colaborativa obtida pelo algoritmo IVQ-learning.

Capítulo 6

Conclusões

Propusemos um novo paradigma para aprendizado em sistemas multi-agentes visando resolver problemas cooperativos, inspirado nas interações sociais entre pessoas, especificamente na idéia de que as opiniões de um agente podem mudar o comportamento dos outros. Introduzimos este novo paradigma como uma solução para um dos problemas ainda em aberto na área de aprendizado em sistemas multi-agentes, que é enunciado da seguinte forma: "a co-adaptação dos aprendizes uns com os outros é uma violação a uma suposição fundamental da maioria das técnicas de aprendizado de máquina, e então são necessários novos algoritmos para tratar deste problema"[Panait & Luke 2005].

Como mostrado no capítulo 3, existe um problema conceitual ao aplicar aprendizado por reforço em sistemas multi-agentes, causado pela diferença existente entre o modelo do ambiente e o modelo do agente. Este problema é resolvido parcialmente pelo paradigma de aprendizado de ações conjuntas. Esta solução é considerada parcial justamente porque a mesma inclui as informações necessárias no modelo do agente para torná-lo coerente ao modelo do ambiente em relação às ações. Porém, ela negligencia o fato de que a evolução dos agentes não depende unicamente das ações dos outros agentes mas também do estado interno deles. Isto significa que, a todo instante, os estados internos de todos os agentes são correlacionados, indicando que eles estão se adaptando, mutuamente e, ao mesmo tempo, ao ambiente. Assim, no paradigma proposto nesta tese incluímos, dentro do estado interno de cada agente, o estado interno, as ações e a recompensa obtida de todos os colegas. Isso faz de nossa proposta a primeira solução completa para o problema conceitual relatado acima e que, ao mesmo tempo, consegue resolver um dos principais problemas em aberto da área de aprendizado em sistemas multi-agente.

Um fato importante que valida a conclusão anterior é que, embora as ações de cada agente sejam produto do seu estado interno, elas não são totalmente dependentes deste estado interno, já que a política de seleção de ações escolhida pode fazer com que a escolha fuja totalmente das crenças atuais do agente (do seu estado interno).

Como visto, o paradigma proposto nesta tese inspira-se nas teorias sobre interações sociais de Levy Vigotsky. Especificamente, no fato de que, quando as pessoas interagem, elas comunicam umas às outras o que acham das ações das outras, seja através de crítica direta ou de elogios. Esses protestos ou elogios podem *influenciar* o comportamento de uma pessoa. Nesse sentido, quando outras pessoas protestam, uma pessoa tentará evitar as ações que causaram esses protestos e, quando outras pessoas tecem elogios, a pessoa tentará repetir estas ações mais vezes.

Assim, criamos um modelo deste paradigma para aprendizado por reforço e o implementamos como uma generalização do algoritmo Q-learning. Basicamente, na equação de aprendizado, acrescentamos informações sobre o estado interno dos outros agentes.

Após os experimentos envolvendo o algoritmo desenvolvido, notamos que o paradigma apresentado nesta tese, embora seja a melhor solução teórica que encontramos para aplicar aprendizado em sistemas multi-agentes, na prática ele ainda pode ser aprimorado. Mais estritamente, essa melhoria pode se dar na definição das equações que o implementam, de forma que o torne mais robusto e, consequentemente, substancialmente superior a outros algoritmos que implementam paradigmas tradicionais.

Nesta tese, implementamos a versão básica do paradigma onde a opinião é fundamentada na recompensa recebida pelo agente e nas suas crenças do mundo (valor de $V(s)$ ou $Q(s,a)$). E a influência é baseada nas opiniões multiplicadas por um fator constante. Esta versão se mostrou experimentalmente superior às versões básicas dos paradigmas tradicionais, porém, é possível melhorar os resultados obtidos mudando algumas características dos algoritmos como por exemplo, pode se modificar os critérios de opinião ao acrescentar esquecimento nos agentes, ou se pode modificar como os agentes são influenciados ao fazer com que o fator de influência (β) seja uma função dependente do conhecimento que se tem dos outros agentes, e assim por diante.

Os algoritmos implementados usando como base o algoritmo Q-learning e os conceitos propostos para o paradigma IVRL, se mostraram, nos problemas propostos, com uma capacidade de convergência para equilíbrios de Nash e especialmente para equilíbrios de Nash ótimos, bem maior que a capacidade de convergência dos paradigmas tradicionais. Ainda, observamos uma propriedade muito importante dos algoritmos que implementam o paradigma proposto. Esta propriedade que até o momento não havia sido observada em algoritmos de aprendizado por reforço, é a capacidade de auto-organização dos agentes que o implementam. Ainda, esta propriedade dá aos agentes a capacidade de se auto-alocar tarefas de uma forma coordenada. Portanto, nosso modelo brinda a possibilidade de eliminar a fase de alocação de tarefas bastante usada nas diferentes implementações de aprendizado em MAS.

Em relação ao tempo e capacidade de processamento deste paradigma, é importante observar que, o número de valores a serem armazenados cresce se o número de ações e/ou agentes aumenta. Então, ao compará-lo com os paradigmas tradicionais observamos que, enquanto para o paradigma de aprendizado de ações conjuntas o aumento nos requerimentos de espaço de armazenamento cresce exponencialmente, para o paradigma de aprendizado de ações independentes e para o paradigma proposto nesta tese o crescimento é linear. Observamos ainda que o tempo requerido para o processamento do aprendizado dos algoritmos propostos nesta tese e dos algoritmos de aprendizado independente são comparáveis, porém, o tempo de processamento cresce muito para os algoritmos de aprendizado de ações conjuntas.

Já em relação aos parâmetros de treinamento, observamos que, para políticas do tipo $softmax$, é importante trabalhar com um fator de decaimento da temperatura no tempo. E que, para o caso da nossa proposta, este valor influencia diretamente na capacidade de aprendizado. Porém, os algoritmos de aprendizado de ações conjuntas são muito mais sensíveis às mudanças nos parâmetros e às condições estabelecidas para os algoritmos e

problemas. Assim, nos problemas escolhidas para realização dos testes, observamos que determinados valores podem levar à perda total da capacidade de convergência.

Importante observar que após implementar, testar e comparar as versões do Q-learning para os paradigmas tradicionais e para o paradigma IVRL , tanto em jogos repetitivos como estocásticos, reconhecemos as principais características destes paradigmas, suas diferenças, assim como sua a influência que os parâmetros e a estrutura do ambiente têm sobre eles. Por exemplo, podemos observar que a distância entre o equilíbrio ótimo e subótimo possui influência direta no rendimento dos diferentes paradigmas, porém a maior influência negativa é a o JAQ-learning. Ainda, para políticas gulosas, o JAQ-Qlearning é mais eficiente na procura do equilíbrio ótimo já que ao treinarmos com políticas $softmax$, e devido ao fato de que um agente aprendendo com o algoritmo JAQ-learning trabalha com o histórico das ações dos seus colegas, o agente perderá a capacidade de explorar as melhores estratégias já aprendidas, fazendo-se necessária uma mudança na forma de modelar o outro.

Já uma característica dos algoritmos IVQ-learning e IQ-learning, é que, quando treinado com políticas Gulosas, o algoritmo IVQ-learning tem maior capacidade de convergência para o equilíbrio ótimo que o algoritmo IQ-learning. Porém, ao aumentar a capacidade de exploração de novas estratégias de solução (usando políticas softmax com temperatura decaindo no tempo), aumenta também a capacidade do último de convergir ao equilíbrio ótimo, podendo chegar a valores comparáveis com os obtidos pelo IVQ-learning. Porém, é importante observar que se analisados com os mesmos parâmetros (em todos os resultados obtidos durante o desenvolvimento da tese) o algoritmo IVQ-learning é superior.

Também importantes foram as extensões feitas aos jogos Repetitivos, por Claus e Boutilier, para serem aplicados em problemas de N agentes. Essas extensões permitiram fazer uma análise acurada dos paradigmas tradicionais e do paradigma proposto sem requerer entrar na complexidade existente em jogos estocásticos com N agentes. Este tipo de teste é muito importante porque, pela complexidade inerente a problemas em MAS, quanto mais simplificado o modelo de testes, melhor será a avaliação dos algoritmos.

Igualmente é importante observar que, com a criação deste novo paradigma, o conjunto de aplicações de aprendizado por reforço em MAS foi ampliado. Ou seja, além da possibilidade de aplicar estes algoritmos nos problemas tradicionais de aprendizado em MAS, como por exemplo coordenação de tarefas em sistemas multi-robôs, é possível aplicar aprendizado por reforço nos problemas essencialmente colaborativos, tais como os tradicionais ambientes de teste dos algoritmos inspirados biologicamente (caixeiro viajante, busca de provisões etc).

Finalmente, nosso trabalho deixa claro que o paradigma apresentado é teórica e experimentalmente superior aos paradigmas tradicionais. Isto porque embora em alguns testes realizados os resultados dos paradigmas tradicionais sejam comparáveis aos resultados do paradigma proposto, o último possui características como a auto-organização e o fato de ser pouco influenciável (se comparado com os paradigmas tradicionais) às mudanças dos parâmetros de treinamento e à relação *exploration/explotation*. Em todos os testes realizados, mesmo mudando a estratégia de seleção de ações, os parâmetros e o modelo do ambiente, a versão do algoritmo IVQ-learning obteve sempre melhores resultados.

6.1 Trabalhos futuros

Durante o desenvolvimento da tese, fizemos um estudo completo, incluindo todas as etapas de validação do paradigma proposto (desenvolvimento do modelo matemático, implementação e experimentação). Porém, embora o nosso algoritmo se encontre bem definido atualmente e tenha melhor rendimento se comparado aos paradigmas tradicionais, alguns trabalhos podem ainda ser realizados no sentido de melhorá-lo ainda mais. Exemplos de alguns desses trabalhos futuros, são mostrados na figura 6.1 e descritos a seguir.

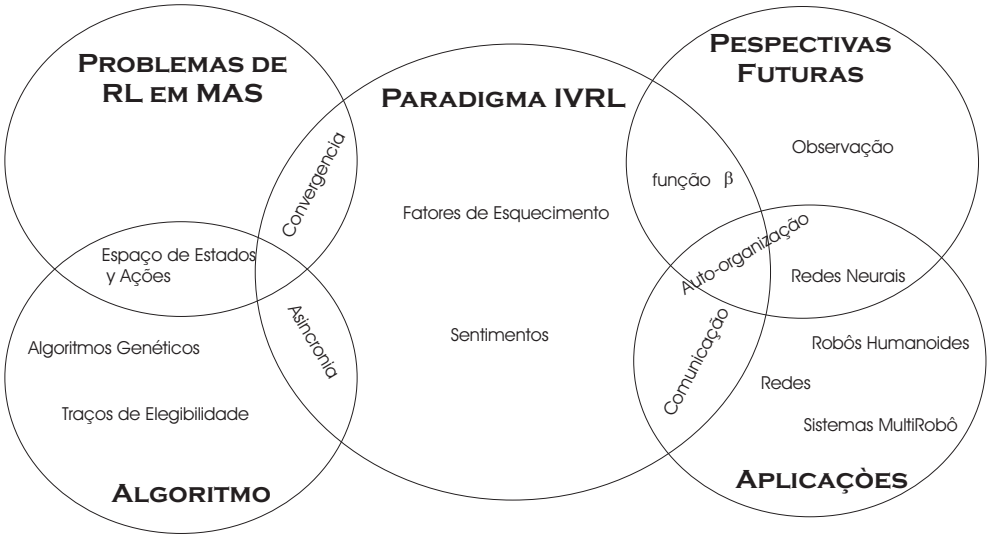


Figura 6.1: Perspectivas

1. Os algoritmos de aprendizado por reforço possuem uma restrição bem conhecida, relacionada ao modelo de espaços de estados e ações contínuos ou virtualmente infinitos. Torna-se cara a construção de uma tabela para armazenar os valores para tais situações ou, às vezes, chega a ser de implementação impossível. O paradigma que propomos não está alheio a este problema. Assim, uma possível extensão deste trabalho é o desenvolvimento de algoritmos que apliquem o paradigma IVRL e, ao mesmo tempo, tentem lidar com o problema mencionado.
2. Em relação à capacidade de convergência, nesta tese mostramos, experimentalmente, que o paradigma proposto tem capacidade maior de convergir para equilíbrios de Nash e para equilíbrios de Nash ótimos do que os paradigmas tradicionais. Porém, faz-se necessário uma melhor formalização desta probabilidade maior de convergência, o que pode ser também realizado a título de trabalho futuro. Conjeturamos que o uso de modelos da teoria de controle possa ajudar nesta formalização, sendo este considerado um trabalho matemático razoável (fora do escopo desta tese).
3. Em relação a outros algoritmos possíveis de serem implementados, falta ainda implementar e testar o paradigma proposto com algoritmos genéticos e com algorit-

mos baseados em traços de elegibilidade (*elegibility trace*). Esta idéia surge do fato de que os algoritmos genéticos são classificados como algoritmos baseados em recompensas e porque os traços de elegibilidade representam, em aprendizado por reforço, uma possibilidade de garantir melhores taxas de convergência ao reunir em um único algoritmo características de aprendizado das diferenças temporais e de Monte Carlo.

4. Ainda, uma das principais propriedades do paradigma proposto nesta tese é que ele suporta implementação de forma assíncrona. Como os algoritmos não foram implementados desta forma, outra idéia de possíveis melhorias é desenvolver algoritmos que trabalhem assincronamente ou então modificar os já propostos para trabalhar assim.
5. Em relação ao próprio paradigma proposto nesta tese, sugerimos o estudo de determinadas mudanças, como por exemplo, acrescentar "sentimentos"aos agentes, ou fatores de esquecimento das ações executadas pelos agentes. Estas modificações são importantes também porque existe a probabilidade que se tornem ferramentas chave para que nosso paradigma melhore suas capacidades de auto-organização.
6. Outra característica importante do nosso paradigma é que definimos o parâmetro β como se este fosse uma função. Porém, na implementação, optamos por usar uma constante, devido à complexidade que encontramos ao tentar definir a forma matemática dessa função. Sugerimos então, verificar a validade de se usar funções para modelar este valor. Isto pode ser útil para modificar ao longo do tempo a influência que o agente pode receber de cada um dos seus colegas e também para fazer com que esta influência seja diferente para cada um dos outros agentes.
7. Ainda, outra das principais características da nossa proposta é o fato de que a ação da qual se opina não seja em essência a mesma ação executada realmente pelo colega. Este modelo irá depender do estado interno de cada agente. Então, é importante a criação de técnicas de observação de outros agentes com o intuito de determinar que ações eles estejam executando para, logo, poder opinar. E também para permitir criar programas gerais onde as ações não necessariamente teriam que ser definidas pelos projetistas dos agentes. Ou ainda, para permitir a junção de agentes, implementados por diferentes projetistas, no mesmo sistema.
8. É importante continuar avaliando as capacidades de auto-organização do paradigma IVRL. Principalmente, o mesmo deve ser testado nos ambientes usados pela comunidade que trabalha nesta área. Por exemplo, testá-lo nas mesmas plataformas em que os sistemas inspirados biologicamente são testados.
9. Embora os nossos testes tenham sido realizados com agentes implementados de forma totalmente distribuída, com comunicação via *sockets*, é importante analisar a robustez e o nível de tolerância da nossa proposta a problemas de comunicação. Uma infinidade de problemas pode surgir daqui, principalmente considerando situações em que o sistema perde um ou mais agentes.
10. Além disso, todos os testes usados nesta tese são testes teóricos, considerados como problemas de laboratório. Então, faz-se necessária a implementação deste paradigma em problemas reais como sistemas multi-robôs, por exemplo, incluindo robôs humanóides e redes de computadores. De fato, trabalho neste sentido está

sendo realizado por outro pesquisador do Laboratório Natalnet.

11. Finalmente, devido às características próprias do paradigma proposto nesta tese, conjeturarmos que a criação de uma rede neural, onde cada neurônio implemente algoritmos de aprendizado por reforço com valores de influência, seria um grande salto no estado da arte em aprendizado de máquina.

Referências Bibliográficas

Aardt, D. Van & T. Marwala (2005), A study in a hybrid centralised-swarm agent community, *em* 'ICCC 2005 - IEEE 3rd International Conference on Computational Cybernetics - Proceedings', Vol. 2005, pp. 169–174.

Aguilar-Ponce, R., A. Kumar, J. L. Tecpanecatl-Xihuitl & M. Bayoumi (2007), 'A network of sensor-based framework for automated visual surveillance', *Journal of Network and Computer Applications* **30**(3), 1244–1271.

Anthony, A. & T. Jannett (2006), 'A framework for using agents in distributed sensor networks (non-refereed)'.

Banerjee, D. & S. Sen (2007), 'Reaching pareto-optimality in prisoner's dilemma using conditional joint action learning', *Autonomous Agents and Multi-Agent Systems* **15**(1), 91–108.

Barrios-Aranibar, D. & P. J. Alsina (2004), Reinforcement learning-based path planning for autonomous robots, *em* 'I ENRI - Encontro Nacional de Robótica Inteligente no XXIV Congresso da Sociedade Brasileira de Computação', Salvador, BA, Brazil.

Barrios-Aranibar, Dennis & Luiz Marcos Garcia Gonçalves (2007*a*), Learning coordination in multi-agent systems using influence value reinforcement learning, *em* 'Seventh International Conference on Intelligent Systems Design and Applications (ISDA 07)', IEEE Press.

Barrios-Aranibar, Dennis & Luiz Marcos Garcia Gonçalves (2007*b*), Learning from delayed rewards using influence values applied to coordination in multi-agent systems, *em* 'VIII SBAI - Simpósio Brasileiro de Automação Inteligente', Flotianópolis, SC, Brazil.

Barrios-Aranibar, Dennis & Luiz Marcos Garcia Gonçalves (2007*c*), Learning to collaborate from delayed rewards in foraging like environments, *em* 'VI Jornadas Peruanas de Computación - JPC 2007', Trujillo, Peru.

Barrios-Aranibar, Dennis & Luiz Marcos Garcia Gonçalves (2007*d*), Learning to reach optimal equilibrium by influence of other agents opinion, *em* 'Hybrid Intelligent Systems, 2007. HIS 2007. 7th International Conference on', Kaiserslautern, Alemanha, pp. 198 – 203.

Barrios-Aranibar, Dennis & Luiz Marcos Garcia Gonçalves (2008), Aprendizado por reforço com valores de influência para sistemas multi-agentes, *em* 'IV Workshop on MSc Dissertation and PhD Thesis in Artificial Intelligence (WTDIA)', Salvador, Brasil.

Barrios-Aranibar, Dennis & Luiz Marcos Garcia Gonçalves (2009), *Theory and Novel Applications of Machine Learning*, ed.Viena : Vedran Kordic, Austria, chapter Influence Value Q-Learning: A Reinforcement Learning Algorithm for Multi Agent Systems.

Barrios-Aranibar, Dennis, Luiz Marcos Garcia Gonçalves & Pablo Javier Alsina (2008), *Frontiers in Evolutionary Robotics*, ed.Viena : Aleksandar Lazinica, Austria, chapter Learning by Experience and by Imitation in Multi-Robot Systems.

Barrios-Aranibar, Dennis & Pablo Javier Alsina (2006), Learning strategies for coordination of multi robot systems: A robot soccer application, *em* 'XVI Congresso Brasileiro de Automática (CBA 2006)', Salvador, BA, Brazil.

Barrios-Aranibar, Dennis & Pablo Javier Alsina (2007), *Mobile Robots: The Evolutionary Approach*, Vol. 50/2007 de *Studies in Computational Intelligence*, Springer Berlin/Heidelberg, chapter Imitation Learning: An Application in a Micro Robot Soccer Game, pp. 201–219.

Barrios-Aranibar, Dennis, Viviane Gurgel, Aquiles Burlamaqui, Luiz Marcos Garcia Gonçalves, Marcela Santos, Gianna R. Araujo, Valber C. Roza & Rafaella A. Nascimento (2006), Technological inclusion using robots, *em* 'III ENRI - Encontro Nacional de Robôtica Inteligente no XXVI Congresso da Sociedade Brasileira de Computação', Campo Grande, MS, Brazil.

Barrios-Aranibar, Dennis, Viviane Gurgel, Marcela Santos, Gianna R. Araujo, Valber C. Roza, Rafaella A. Nascimento, Alzira Ferreira da Silva, Alynara Silva & Luiz Marcos Garcia Gonçalves (2006), Roboeduc: A software for teaching robotics to technological excluded children using lego prototypes, *em* '3rd IEEE Latin American Robotics Symposium - LARS 2006', Santiago, Chile.

Beni, G. (2005), From swarm intelligence to swarm robotics, *em* E.Sahin & W.Spears, eds., 'Proceedings of the SAB 2004 Workshop on Swarm Robotics', Lecture Notes in Computer Science 3342, Santa Monica, CA, USA, pp. 1–9.

Bonabeau, E., M. Dorigo & G. Theraulaz (1999), *Swarm intelligence: From natural to artificial systems*, Santa Fe Institute Studies in the Sciences of Complexity, Oxford University Press, New York, NY. 306 pp.

Botelho, Silvia & Rachid Alami (2000), Robots that cooperatively enhance their plans, *em* 'Proc. of 5th International Symposium on Distributed Autonomous Robotic Systems (DARS 2000). Lecture Notes in Computer Science', Springer Verlag.

Camazine, S., J.-L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz & E. Bonabeau (2003), *Selforganization in biological systems*, Princeton University Press, Princeton, NJ. 560 pp.

Carrascosa, C., J. Bajo, V. Julian, J. M. Corchado & V. Botti (2008), 'Hybrid multi-agent architecture as a real-time problem-solving model', *Expert Systems with Applications* **34**(1), 2–17.

Chalkiadakis, Georgios & Craig Boutilier (2003), Coordination in multiagent reinforcement learning: a bayesian approach, *em* 'Proceedings of the second international joint conference on Autonomous agents and multiagent systems', Melbourne, Australia.

Chen, D., B. Jeng, W. Lee & C. Chuang (2008), 'An agent-based model for consumer-to-business electronic commerce', *Expert Systems with Applications* **34**(1), 469–481.

Claus, Caroline & Craig Boutilier (1998), The dynamics of reinforcement learning in cooperative multiagent systems, *em* 'Proceedings of the 15th National Conference on Artificial Intelligence -AAAI-98', AAAI Press., Menlo Park, CA, pp. 746–752.

Dalamagkidis, K., D. Kolokotsa, K. Kalaitzakis & G. S. Stavrakakis (2007), 'Reinforcement learning for energy conservation and comfort in buildings', *Building and Environment* **42**(7), 2686–2698.

de Oliveira, Marta Kohl (1997), *Vygotsky: Aprendizado e Desenvolvimento: um Processo Sócio-Histórico*, Scipione, São Paulo.

de Silva, Alzira Ferreira, Alynara Silva, Luiz Marcos Garcia Gonçalves, Ana Maria Guimaraes Guerreiro & Dennis Barrios-Aranibar (2008), Utilização da teoria de vygotsky em robótica educativa, *em* 'IX Congreso Iberoamericano De Informatica Educativa RIBIE 2008', Caracas, Venezuela.

Dimitriadis, S., K. Marias & S. C. Orphanoudakis (2007), 'A multi-agent platform for content-based image retrieval', *Multimedia Tools and Applications* **33**(1), 57–72.

Fenwick, J. W., P. M. Newman & J. J. Leonard (2002), Cooperative concurrent mapping and localization, *em* 'Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on', Vol. 2, pp. 1810–1817.

Fernández, J. L., D. P. Losada & R. Sanz (2008), Enhancing building security systems with autonomous robots, *em* '2008 IEEE International Conference on Technologies for Practical Robot Applications, TePRA', pp. 19–24.

Filar, Jerzy & Koos Vrieze (1997), *Competitive Markov Decision Processes*, Springer-Verlag New York, Inc., 175 Fifth Avenue, New York, NY 10010, USA.

Fisher, M., R. H. Bordini, B. Hirsch & P. Torroni (2007), 'Computational logics and agents: A road map of current technologies and future trends', *Computational Intelligence* **23**(1), 61–91.

Goldman, Claudia V. & Jeffrey S. Rosenschein (1996), Mutually supervised learning in multiagent systems, *em* G.Weiß & S.Sen, eds., 'Adaptation and Learning in Multi–Agent Systems', Springer-Verlag: Heidelberg, Germany, Berlin, pp. 85–96.

Gonçalves, Luiz M. G., Gilson A. Giraldi, Antonio A. F. Oliveira & Roderic A. Grupen (1999), Learning policies for attention control, *em* 'IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA '99)', Monterey CA, USA.

Gonçalves, Luiz M. G., Roderic A. Grupen, Antonio A. Oliveira, D. Wheeler & A. Fagg (2000), 'Tracing patterns and attention: Humanoid robot cognition', *The Intelligent Systems and their Applications* **15**(4), 70–77.

Gu, D. & H. Hu (2005), 'Teaching robots to plan through q-learning', *Robotica* **23**, 139–147.

Guo, R., M. Wu, J. Peng, J. Peng & W. . Cao (2007), 'New q learning algorithm for multi-agent systems', *Zidonghua Xuebao/Acta Automatica Sinica* **33**(4), 367–372.

Gurgel, Viviane, Marcela Santos, Gianna R. Araujo, Valber C. Roza, Rafaella A. Nascimento, Dennis Barrios-Aranibar & Luiz Marcos Garcia Gonçalves (2006), The problem of robotic medical trash collector, *em* 'III ENRI - Encontro Nacional de Robotica Inteligente no XXVI Congresso da Sociedade Brasileira de Computação', Campo Grande, MS, Brazil.

Hayat, S. A. & M. Niazi (2005), Multi agent foraging - taking a step further q-learning with search, *em* 'Emerging Technologies, 2005. Proceedings of the IEEE Symposium on', pp. 215 – 220.

He, Yulan, Siu Cheung Hui & Yongxiang Sim (2006), *Information Retrieval Technology*, Vol. 4182/2006 de *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, capítulo A Novel Ant-Based Clustering Approach for Document Clustering, pp. 537 – 544.

Hendler, J. A. (1996), 'Intelligent agents: Where ai meets information technology', *IEEE Experts* pp. 20–23.

Hong, J. W. Lee E. & J. Park (2004), A q-learning based approach to design of intelligent stock trading agents, *em* 'Engineering Management Conference, 2004. Proceedings. 2004 IEEE International', Vol. 3, pp. 1289–1292.

Huber, Manfred & Roderic A. Grupen (1997), Learning to coordinate controllers - reinforcement learning on a control basis, *em* 'Proc. of XV International Joint Conference on Artificial Intelligence', AAAI Press, Menlo Park, CA, USA, pp. 1366–1371.

Huntsberger, T., P. Pirjanian, A. Trebi-Ollennu, H. Das Nayar, H. Aghazarian, A. J. Ganino, M. Garrett, S. S. Joshi & P. S. Schenker (2003), 'Campout: a control architecture for tightly coupled coordination of multirobot systems for planetary surface exploration', *Systems, Man and Cybernetics, Part A, IEEE Transactions on* **33**(5), 550–559.

Iba, H. (1999), Evolving multiple agents by genetic programming, *em* U.-M.L. Spector, W. Langdom & P.Angeline, eds., 'Advances in Genetic Programming', Vol. 3, The MIT Press, Cambridge, MA, pp. 447–466.

Ishibuchi, H., T. Nakashima, H. Miyamoto & Chi-Hyon Oh (1997), Fuzzy q-learning for a multi-player non-cooperative repeated game, *em* 'Fuzzy Systems, 1997., Proceedings of the Sixth IEEE International Conference on', Vol. 3, pp. 1573–1579.

Jars, Isabelle, Nadia Kabachi & Michel Lamure (2004), Proposal for a vygotsky's theory based approach for learning in mas, *em* 'AOTP: The AAAI-04 Workshop on Agent Organizations: Theory and Practice', San Jose, California. http://www.cs.uu.nl/ virginia/aotp/papers/AOTP04IJars.Pdf.

Junhong, N. & S. Haykin (1999), 'A q-learning-based dynamic channel assignment technique for mobile communication systems', *Vehicular Technology, IEEE Transactions on* **48**(5), 1676–1687.

Kapetanakis, S. & D. Kudenko (2002), Reinforcement learning of coordination in cooperative multi-agent systems, *em* 'Proceedings of the National Conference on Artificial Intelligence', pp. 326–331.

Kapetanakis, S. & D. Kudenko (2004), Reinforcement learning of coordination in heterogeneous cooperative multi-agent systems, *em* 'Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2004', Vol. 3, pp. 1258–1259.

Kapetanakis, S., D. Kudenko & M. J. A. Strens (2003), 'Reinforcement learning approaches to coordination in cooperative multi-agent systems', *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)* **2636**, 18–32.

Kirchner, F. (1997), Q-learning of complex behaviours on a six-legged walking machine, *em* 'Advanced Mobile Robots, 1997. Proceedings., Second EUROMICRO workshop on', pp. 51 – 58.

Kok, Jelle R. & Nikos Vlassis (2004), Sparse cooperative q-learning, *em* 'Proceedings of the twenty-first international conference on Machine Learning', Banff, Alberta, Canada, p. 61.

Kononen, Ville (2004), 'Asymmetric multiagent reinforcement learning', *Web Intelligence and Agent System* **2**(2), 105 – 121.

Lau, H. Y. K. & V. W. K. Wong (2007), 'An immunity approach to strategic behavioral control', *Engineering Applications of Artificial Intelligence* **20**(3), 289–306.

Lauer, Martin & Martin Riedmiller (2000), An algorithm for distributed reinforcement learning in cooperative multi-agent systems, *em* 'In Proceedings of the Seventeenth International Conference on Machine Learning', Morgan Kaufmann, pp. 535–542.

Lauer, Martin & Martin Riedmiller (2004), Reinforcement learning for stochastic cooperative multi-agent systems, *em* 'AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems', IEEE Computer Society, Washington, DC, USA, pp. 1516–1517.

Laurent, G. & E. Piat (2001), Parallel q-learning for a block-pushing problem, *em* 'Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on', Vol. 1, pp. 286–291.

Li, P., X. Huang, M. Wang & X. Zeng (2008), Multiple mobile robots map building based on dsmt, *em* '2008 IEEE International Conference on Robotics, Automation and Mechatronics, RAM 2008', pp. 509–514.

Liu, Yanfei & K. M. Passino (2004), 'Stable social foraging swarms in a noisy environment', *Automatic Control, IEEE Transactions on* **49**(1), 30 – 44.

Maes, P. (1994), 'Modeling adaptive autonomus agents', *Artificil Life Journal* **1**(1), 135 – 162.

Martinoli, Alcherio, Kjerstin Easton & William Agassounon (2004), 'Modeling swarm robotic systems: a case study in collaborative distributed manipulation', *The International Journal of Robotics Research* **23**(4), 415–436.

Mataric, M. J. (1995), 'Designing and understanding adaptive group behavior', *Adaptive Behavior 4:1, Dec 1995, 51-80* **4**(1), 51–80.

Mataric, M. J. (1997), 'Reinforcement learning in the multi-robot domain', *Autonomous Robots* **4**(1), 73–83.

Mes, M., M. van der Heijden & A. van Harten (2007), 'Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems', *European Journal of Operational Research* **181**(1), 59–75.

Mohamad, M. M., M. Dunnigan & N. Taylor (2006), Foraging ant motion planning for articulated robots, *em* 'SICE-ICASE, 2006. International Joint Conference', pp. 4285 – 4290.

Murphy, Kevin (2000), A survey of pomdp solution techniques, Relatório técnico.

Nash, John F., Jr. (1950*a*), 'The bargaining problem', *Econometrica* **18**(2), 155–162.

Nash, John F., Jr. (1950*b*), 'Equilibrium points in n-person games', *Proceedings of the National Academy of Sciences of the United States of America* **36**(1), 48–49.

Nash, John F., Jr. (1953), 'Two-person cooperative games', *Econometrica* **21**(1), 128–140.

Nassiri-Mofakham, F., M. A. Nematbakhsh, N. Ghasem-Aghaee & A. Baraani-Dastjerdi (2009), 'A heuristic personality-based bilateral multi-issue bargaining model in electronic commerce', *International Journal of Human Computer Studies* **67**(1), 1–35.

Noreils, Fabrice R. (1993), 'Toward a robot architecture integrating cooperation between mobile robots: Application to indoor environment', *The International Journal of Robotics Research* **12**(2), 79–98.

Oliveira, D. De & A. L. C. Bazzan (2006), 'Traffic lights control with adaptive group formation based on swarm intelligence', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **4150 LNCS**, 520–521.

Pan, W., L. Liu, Z. Cai & B. Chen (2008), An approach to cooperative multi-robot map building in complex environments, *em* 'Proceedings of the 9th International Conference for Young Computer Scientists, ICYCS 2008', pp. 1733–1737.

Panait, Liviu & Sean Luke (2005), 'Cooperative multi-agent learning: The state of the art', *Autonomous Agents and Multi-Agent Systems* **11**(3), 387–434.

Park, K. H., Y. J. Kim & J. H. Kim (2001), 'Modular q-learning based multi-agent cooperation for robot soccer', *Robotics and Autonomous Systems* **5**(2), 109–122.

Rekleitis, Ioannis, Gregory Dudek & Evangelos Milios (2001), 'Multi-robot collaboration for robust exploration', *Annals of Mathematics and Artificial Intelligence* **31**(1-4), 7–40.

Ribeiro, R., F. Enembreck & A. L. Koerich (2006), 'A hybrid learning strategy for discovery of policies of action', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **4140 LNAI**, 268–277.

Rocha, R., J. Dias & A. Carvalho (2005), 'Cooperative multi-robot systems: A study of vision-based 3-d mapping using information theory', *Robotics and Autonomous Systems* **53**(3-4), 282–311.

Rooker, M. N. & A. Birk (2005), 'Combining exploration and ad-hoc networking in robocup rescue', *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)* **3276**, 236–246.

Russel, S. & P. Norvig (1995), *Artificial Intelligence: A Mordern Approach*, Prentice Hall.

Ryde, J. & Huosheng Hu (2005), Fast circular landmark detection for cooperative localisation and mapping, *em* 'Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on', pp. 2745–2750.

Salustowicz, R. P., M. A. Wiering & J. Schmidhuber (1998), 'Learning team strategies: Soccer case studies', *Machine Learning* **12**(4), 263–282.

Schmickl, T., R. Thenius, C. Moeslinger, G. Radspieler, S. Kernbach, M. Szymanski & K. Crailsheim (2009), 'Get in touch: Cooperative decision making based on robot-to-robot collisions', *Autonomous Agents and Multi-Agent Systems* **18**(1), 133–155.

Sen, Sandip & Mahendra Sekaran (1996), Multiagent coordination with learning classifier systems, *em* G.WeiB & S.Sen, eds., 'Proceedings of the IJCAI Workshop on Adaption and Learning in Multi-Agent Systems', Vol. 1042, Springer Verlag, pp. 218–233.

Sen, Sandip, Mahendra Sekaran & John Hale (1994), Learning to coordinate without sharing information, *em* 'Proceedings of the National Conference on Artificial Intelligence', Vol. 1, pp. 426–431.

Shen, J. & Z. Wu (2008), Service-oriented organization management and coordination control of mas, *em* 'Proceedings - International Conference on Intelligent Computation Technology and Automation, ICICTA 2008', Vol. 1, pp. 479–483.

Shoham, Y., R. Powers & T. Grenager (2007), 'If multi-agent learning is the answer, what is the question?', *Artificial Intelligence* **171**(7), 365–377.

Singh, Satinder P., Tommi Jaakkola, Michael L. Littman & Csaba Szepesvári (2000), 'Convergence results for single-step on-policy reinforcement-learning algorithms', *Machine Learning* **38**(3), 287–308.

Ŝniezyński, B . & J. Koźlak (2006), 'Learning in a multi-agent system as a mean for effective resource management', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **3993 LNCS - III**, 703–710.

Suematsu, N. & A. Hayashi (2002), A multiagent reinforcement learning algorithm using extended optimal response, *em* 'Proceedings of the International Conference on Autonomous Agents', Vol. 2, pp. 370–377.

Suh, I. H., J. H. Kim & S. R. Oh (1997), Region-based q-learning for intelligent robot systems, *em* 'Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on', pp. 172–178.

Sujan, V. A. & S. Dubowsky (2002), Visually built task models for robot teams in unstructured environments, *em* 'Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on', Vol. 2, pp. 1782–1787.

Suton, Richard S. & Andrew G. Barto (1998), *Reinforcement Learning: an introduction*, número 2 *em* 'Adaptative Computation and Machine Learning Series', The MIT Press.

´t Hoen, P. J., K. Tuyls, L. Panait, S. Luke & J. A. La Poutré (2006), An overview of cooperative and competitive multiagent learning, *em* 'Learning and Adaptation in Multi-Agent Systems', Springer LNAI, pp. 1–50.

Tanaka, T., K. Nishida & T. Kurita (2007), 'Navigation of mobile robot using location map of place cells and reinforcement learning', *Systems and Computers in Japan* **38**(7), 65–75.

Tesauro, G. & J. O. Kephart (2002), 'Pricing in agent economies using multi-agent q-learning', *Autonomous Agents and Multi-Agent Systems* **5**(3), 289–304.

Toksari, M. Duran (2007), 'Ant colony optimization approach to estimate energy demand of turkey', *Energy Policy* **35**(8), 3984–3990.

Tumer, K., A. K. Agogino & D. H. Wolpert (2002), Learning sequences of actions in collectives of autonomous agents, *em* 'Proceedings of the International Conference on Autonomous Agents', Vol. 2, pp. 378–385.

Usaha, W. & J. A. Barria (2007), 'Reinforcement learning for resource allocation in leo satellite networks', *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **37**(3), 515–527.

Vengerov, D. (2007), 'A reinforcement learning approach to dynamic resource allocation', *Engineering Applications of Artificial Intelligence* **20**(3), 383–390.

Walker, Reginald L. (2007), 'Purposive behavior of honeybees as the basis of an experimental search engine', *Journal Soft Computing - A Fusion of Foundations, Methodologies and Applications* **11**(8), 697 – 716.

Wang, J. & L. Gasser (2002), 'Mutual online concept learning for multiple agents', *Proceedings of the International Conference on Autonomous Agents* **2**, 362–369.

Wang, Xiaofeng & Tuomas Sandholm (2002), Reinforcement learning to play an optimal nash equilibrium in team markov games, *em* 'in Advances in Neural Information Processing Systems', MIT Press, pp. 1571–1578.

Watkins, C. (1989), Learning from Delayed Rewards, Thesis, University of Cambidge, England.

Wei, L. L., X. Zhang & F. Jin (2008), Fuzzy knowledge for agent oriented knowledge management in e-commerce, *em* 'Proceedings - 5th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2008', Vol. 5, pp. 175–179.

Wooldridge, Michael & Nicholas R. Jennings (1995), 'Intelligent agents: Theory and practice', *The Knowledge Engineering Review* **10**(2), 115–152.

Xiong, G., T. Hashiyama & S. Okuma (2002), An electricity supplier bidding strategy through q-learning, *em* '(2002) Proceedings of the IEEE Power Engineering Society Transmission and Distribution Conference', Vol. 3, pp. 1516–1521.

Yeung, David W. K. & Leon A. Petrosyan (2006), *Cooperative Stochastic Differential Games*, Springer Series in Operations Research and Financial Engineering, Springer Science+Business Media, Inc., 233 Spring Street, New York, NY 10013, USA.

Yun, Xie, Yang Yi-min & Xia Yi-min (2005), Cooperative map building of multi-robot based on grey fusion, *em* 'Proceedings of the IEEE International Conference on Industrial Technology', pp. 353–358.

Zhang, D., G. Xie, J. Yu & L. Wang (2007), 'Adaptive task assignment for multiple mobile robots via swarm intelligence approach', *Robotics and Autonomous Systems* **55**(7), 572–588.

Zhang, G. ., Y. Wang & C. W. De Silva (2008), Multi-sensor gripper positioning in unstructured urban environments using neural networks, *em* 'Proceedings of the IEEE International Conference on Automation and Logistics, ICAL 2008', pp. 1474–1479.

Zou, L., J. Xu & L. Zhu (2005), Designing a dynamic path guidance system based on electronic maps by using q-learning, *em* 'Proceedings SPIE, International Conference on Space Information Technology', pp. 5985, 59855A.

Apêndice A

Coordenação em Jogos Repetitivos

Neste apêndice serão mostrados todos os resultados obtidos ao testar nossa proposta em jogos repetitivos com 2, 3 e 4 agentes. Estes resultados são muitos correlatos.

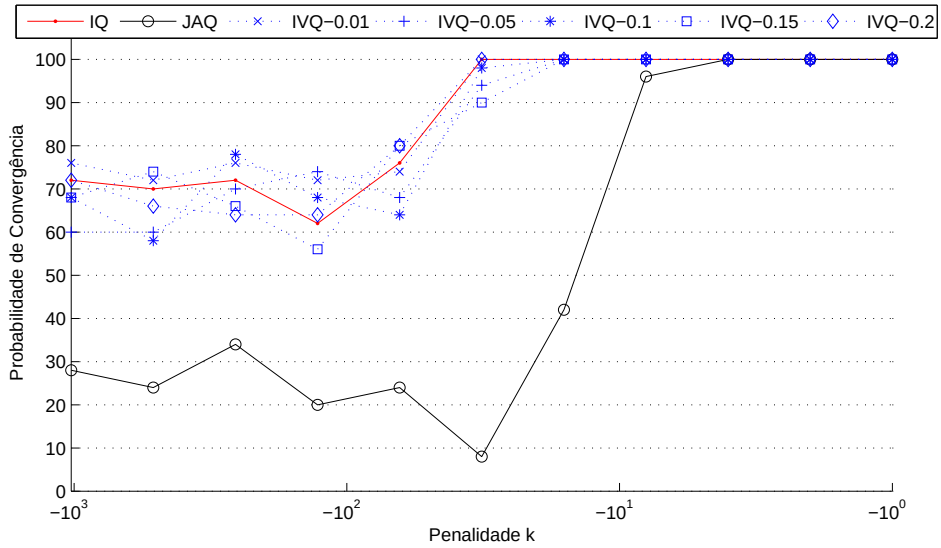
Como dito anteriormente, para o penalty game, a análise feita neste trabalho, na tentativa de ser o mais completa possível, considerou um valor variável para a recompensa no equilíbrio sub-ótimo $so \in \{2,3,4,5,6\}$. Isto, com o intuito de analisar se a distância entre o equilíbrio ótimo e sub-ótimo terá influência relevante no rendimento dos diferentes paradigmas.

Observou-se que ao mudar o valor de *so* a diferença de probabilidades de convergência dos algoritmos não muda significativamente e a mesma vai diminuindo para valores de *so* próximos da recompensa ótima (10). Assim, os gráficos correspondentes a $so \in \{3,4,5,6\}$ só serão mostrados para os jogos repetitivos com 2 agentes. Já, para o caso de jogos com 3 e 4 agentes estes foram omitidos; e só serão mostrados os gráficos para $so = 2$.

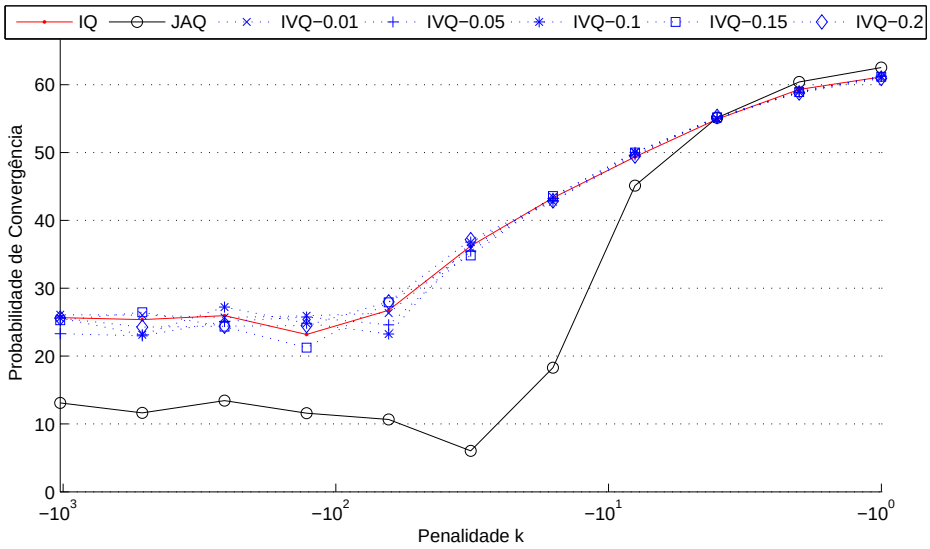
Em relação ao Climbing Game, quando testado, usou-se valores variáveis de T ($T \in \{16(0.9991^t), 16(0.9992^t), ..., 16(0.9999^t)\}$). Desse modo, para o jogo com 2 agentes, serão mostrados os resultados obtidos para valores de T entre $16(0.9992^t)$ e $16(0.9999^t)$. Além disso, como dito no capítulo de resultados, os algoritmos, ao serem testados neste jogo nas suas versões de três e quatro agentes, foram submetidos a situações extremas que permitem ressaltar suas diferenças ou fixar suas semelhanças. Todos os resultados obtidos para 3 e 4 agentes são mostrados nas seções correspondentes a este jogo.

A.1 Coordenação em jogos repetitivos com 2 agentes

A.1.1 Penalty Game

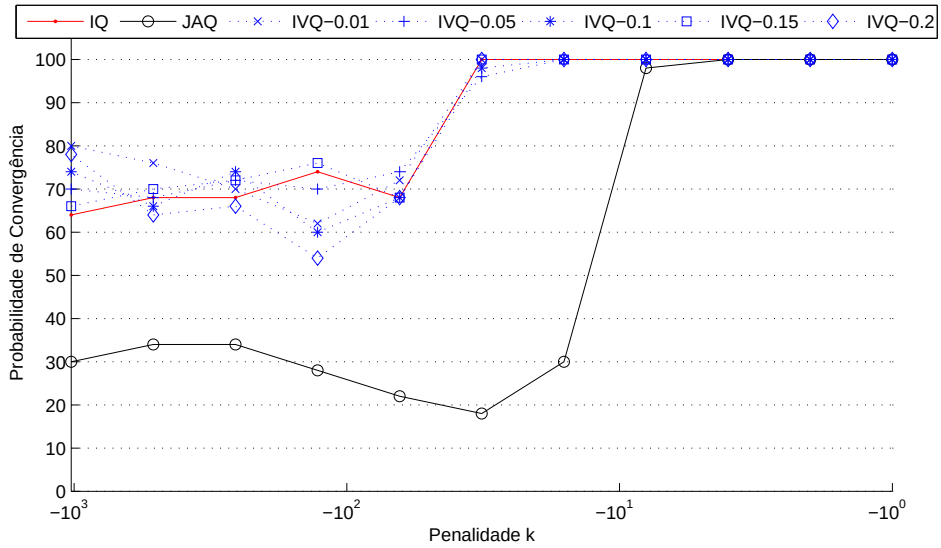


(a) Greedy

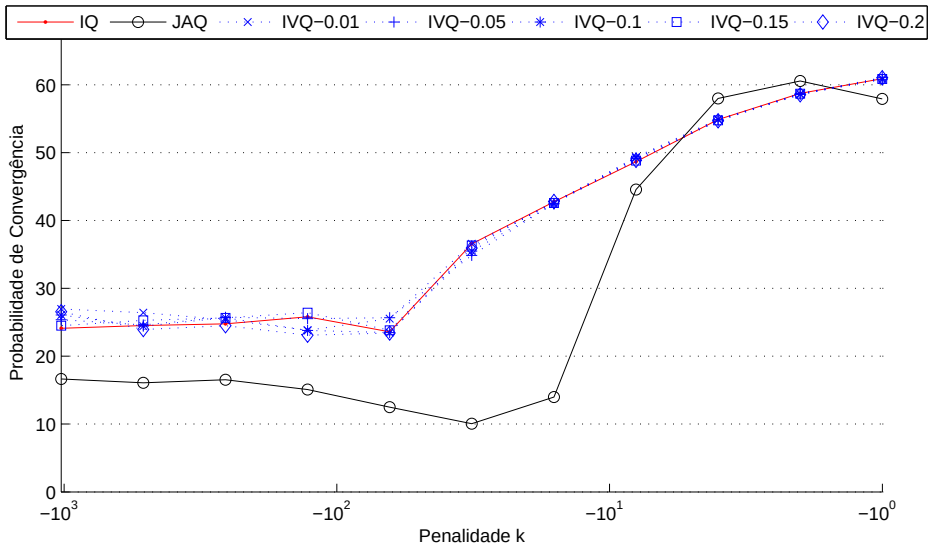


(b) Softmax

Figura A.1: Convergência no *Penalty Game* com 2 agentes, $T = 16(0.999^t)$ e $so = 3$

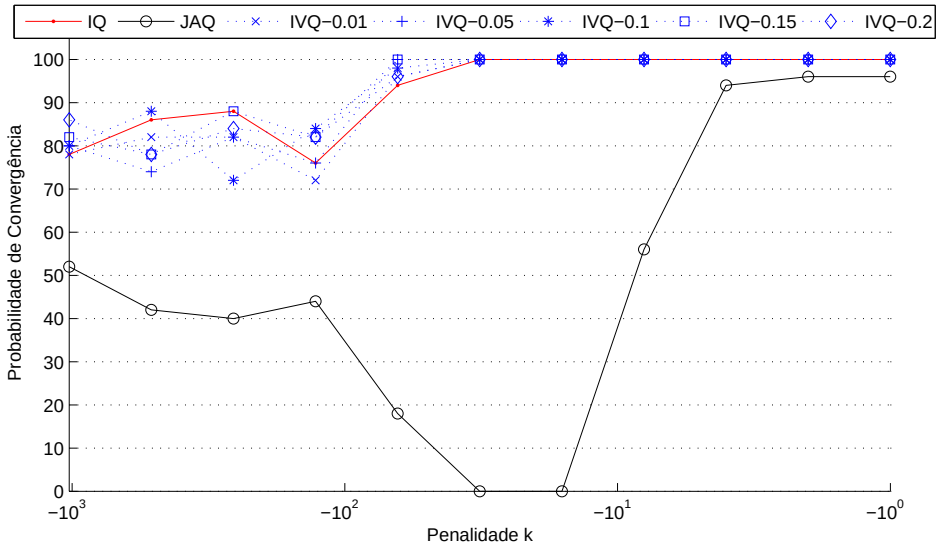


(a) Greedy

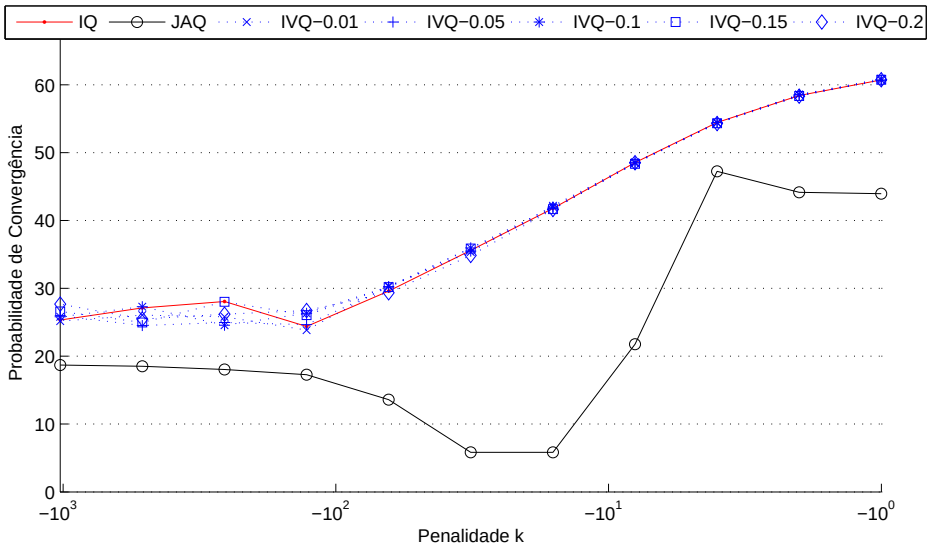


(b) Softmax

Figura A.2: Convergência no *Penalty Game* com 2 agentes, $T = 16(0.9995^t)$ e $so = 3$

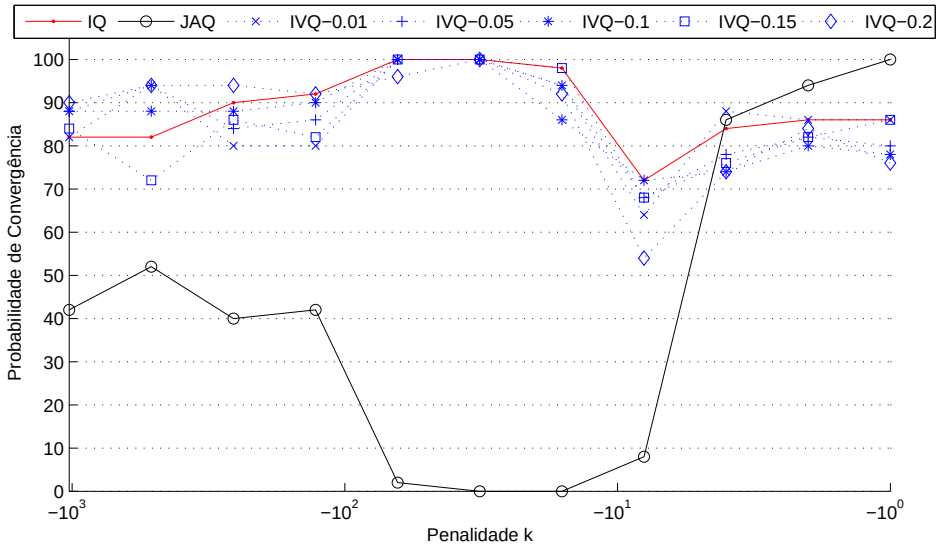


(a) Greedy

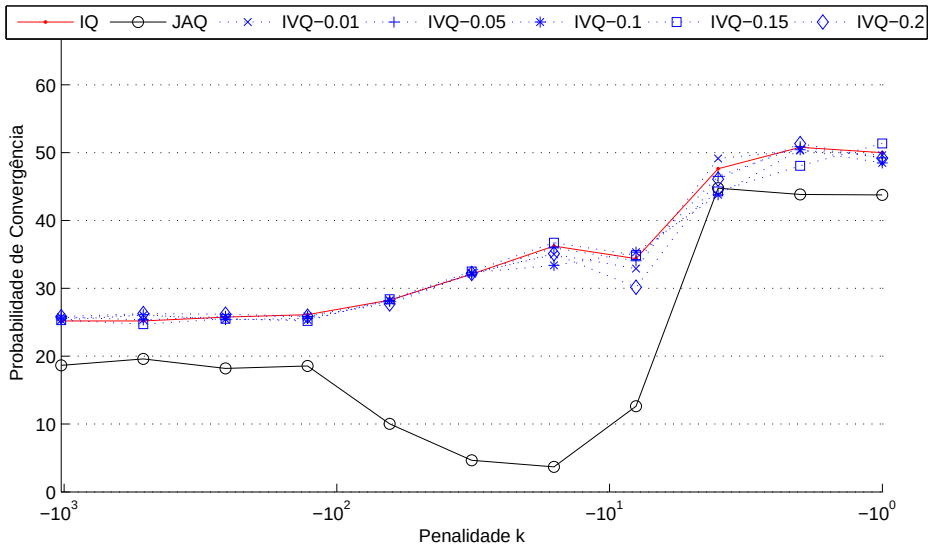


(b) Softmax

Figura A.3: Convergência no *Penalty Game* com 2 agentes, $T = 16(0.9998^t)$ e $so = 3$



(a) Greedy



(b) Softmax

Figura A.4: Convergência no *Penalty Game* com 2 agentes, $T = 16(0.9999^t)$ e $so = 3$

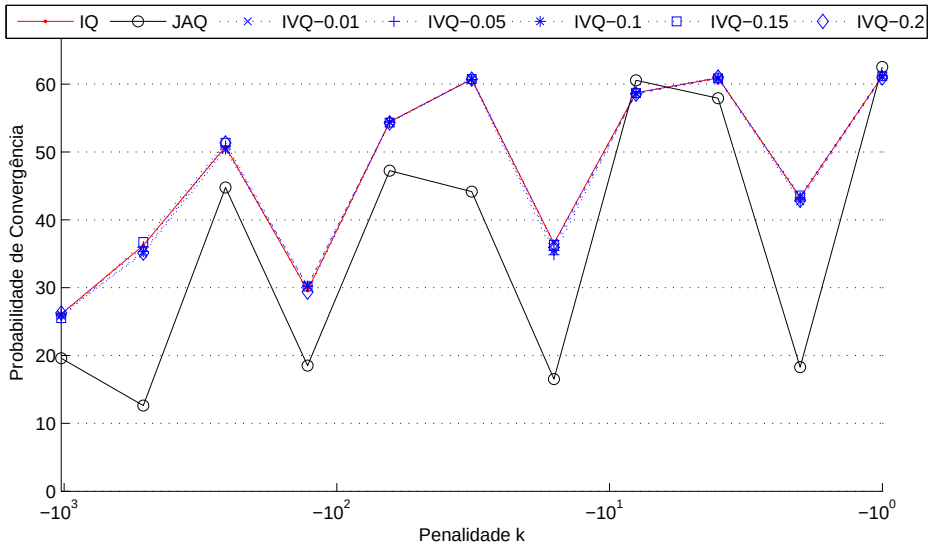
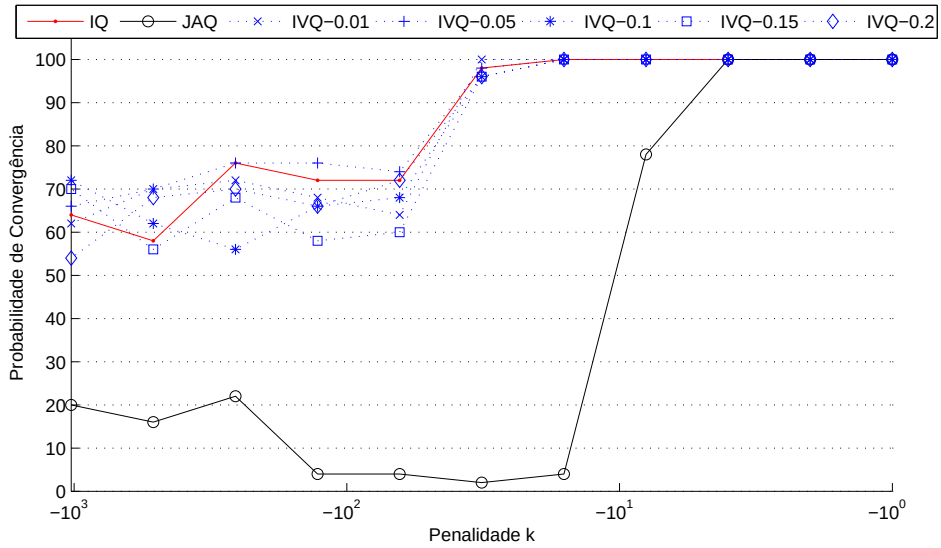
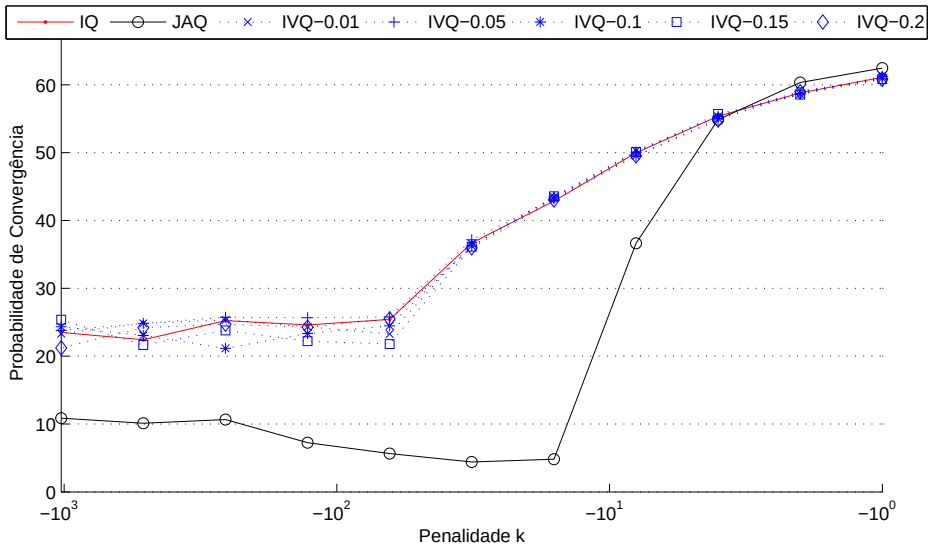


Figura A.5: Melhores algoritmos no *Penalty Game* com 2 agentes para $so = 3$

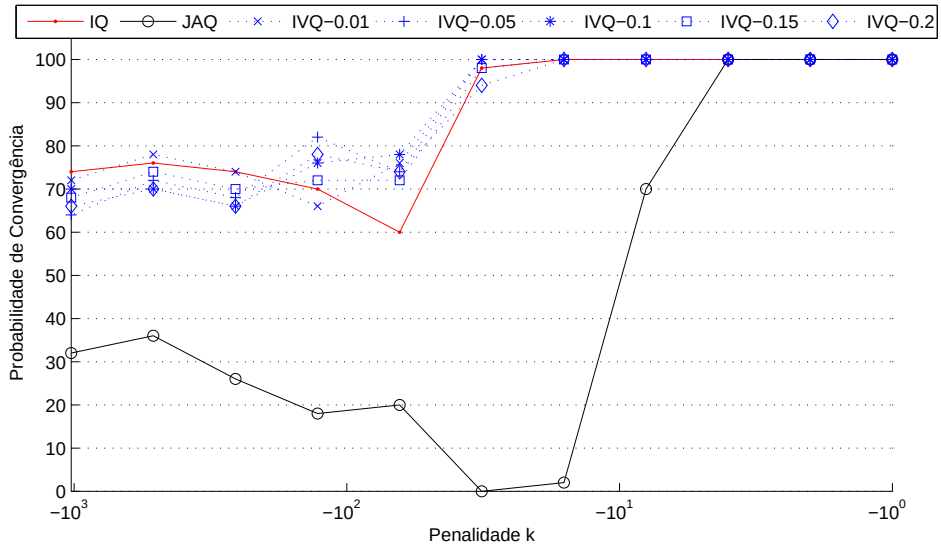


(a) Greedy

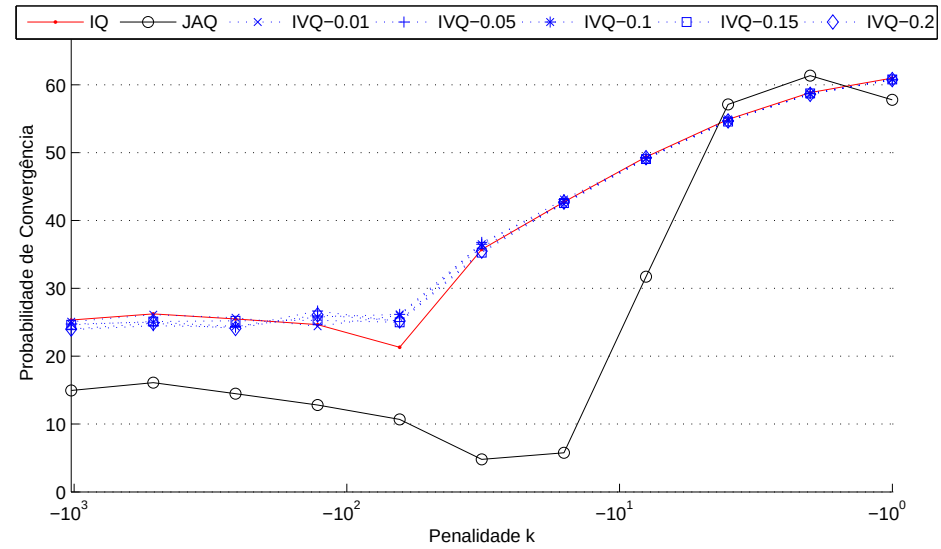


(b) Softmax

Figura A.6: Convergência no *Penalty Game* com 2 agentes, $T = 16(0.999^t)$ e $so = 4$

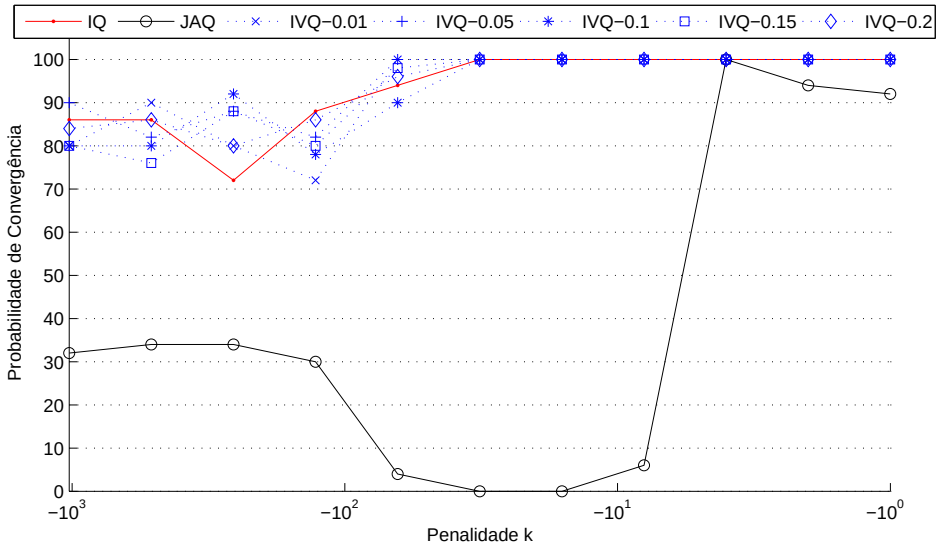


(a) Greedy

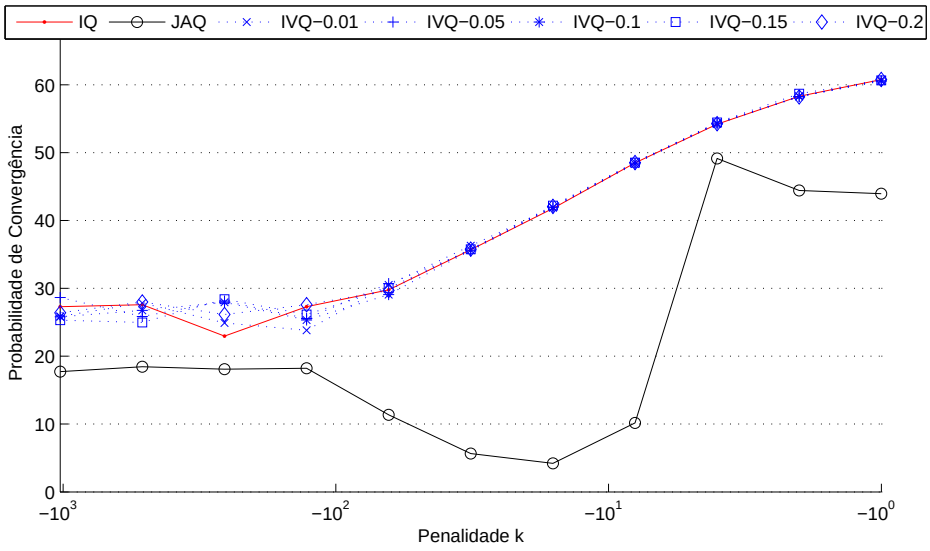


(b) Softmax

Figura A.7: Convergência no *Penalty Game* com 2 agentes, $T = 16(0.9995^t)$ e $so = 4$

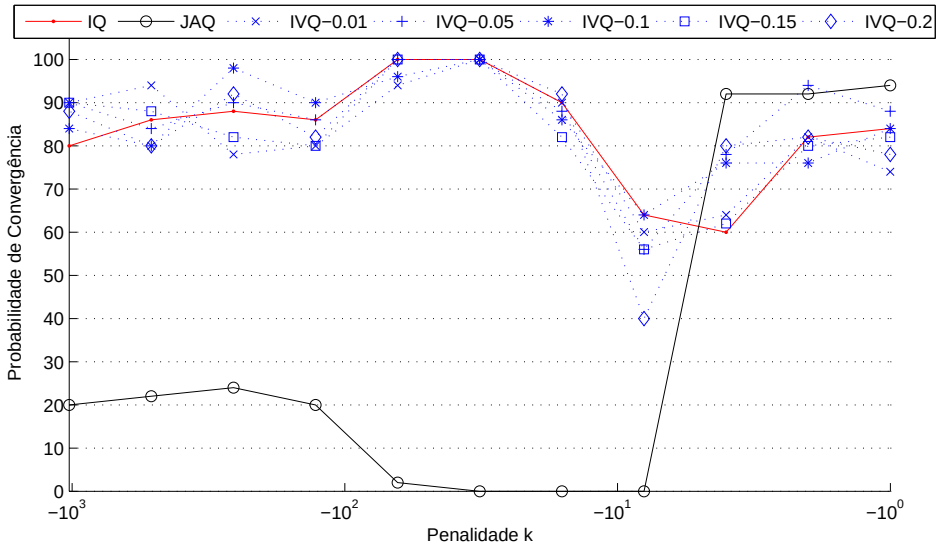


(a) Greedy

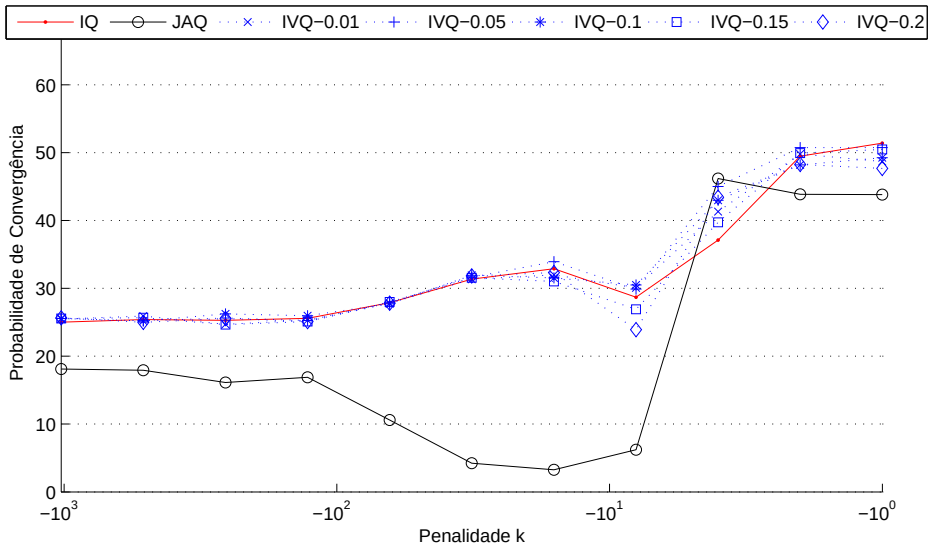


(b) Softmax

Figura A.8: Convergência no *Penalty Game* com 2 agentes, $T = 16(0.9998^t)$ e $so = 4$



(a) Greedy



(b) Softmax

Figura A.9: Convergência no *Penalty Game* com 2 agentes, $T = 16(0.9999^t)$ e $so = 4$

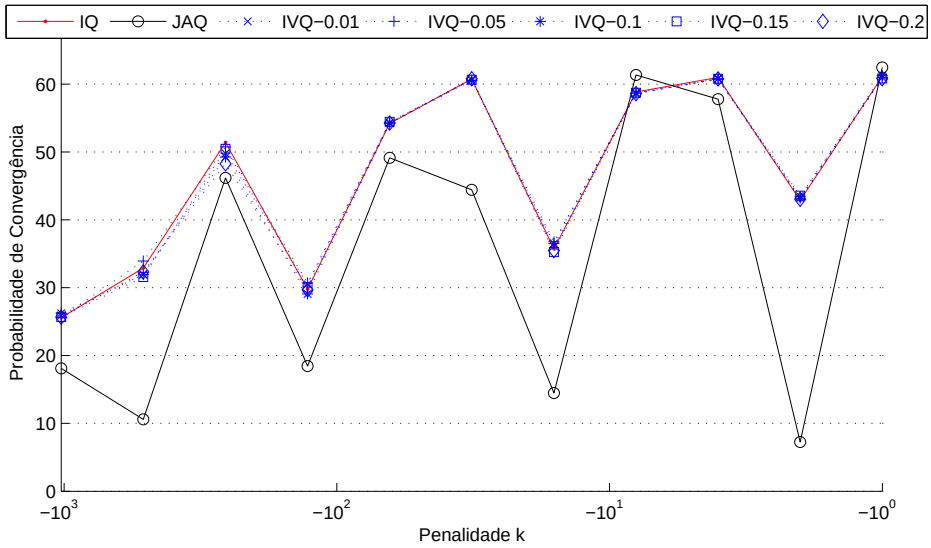
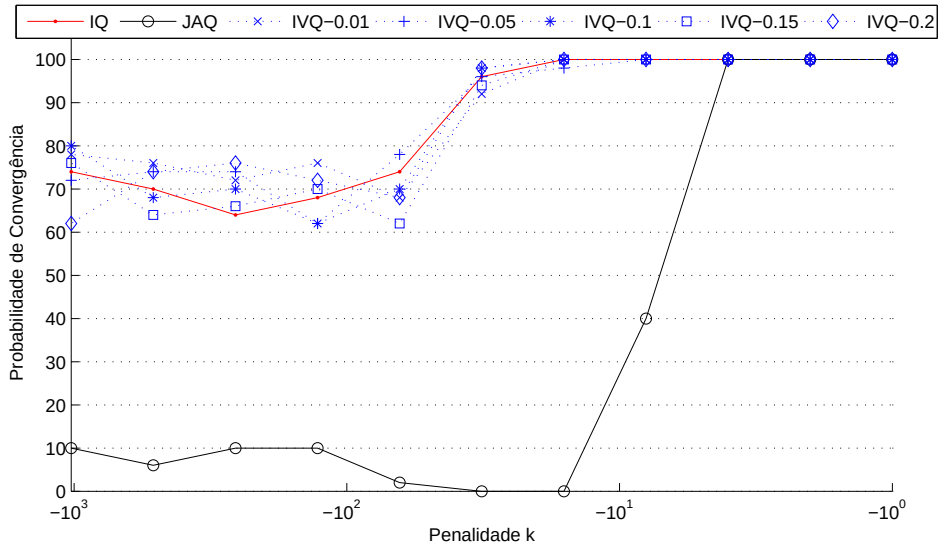
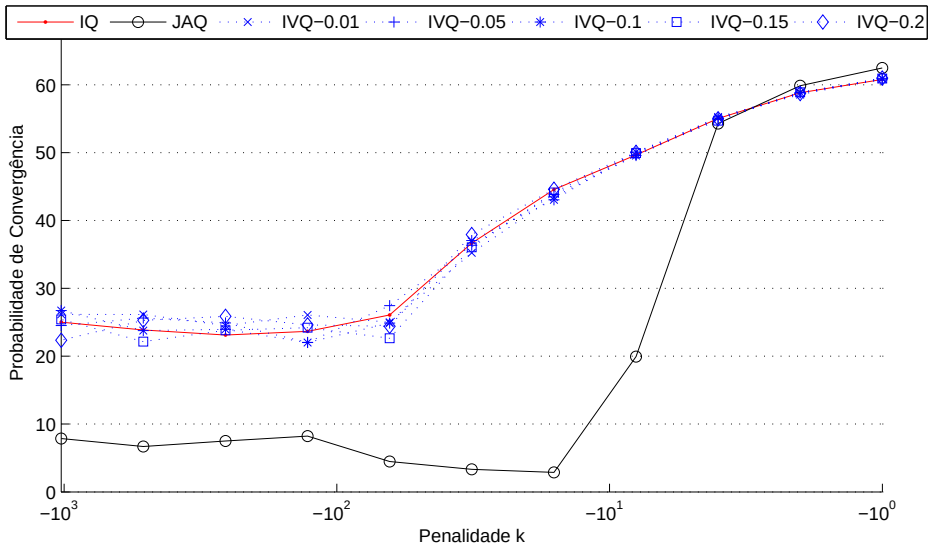


Figura A.10: Melhores algoritmos no *Penalty Game* com 2 agentes para $so = 4$

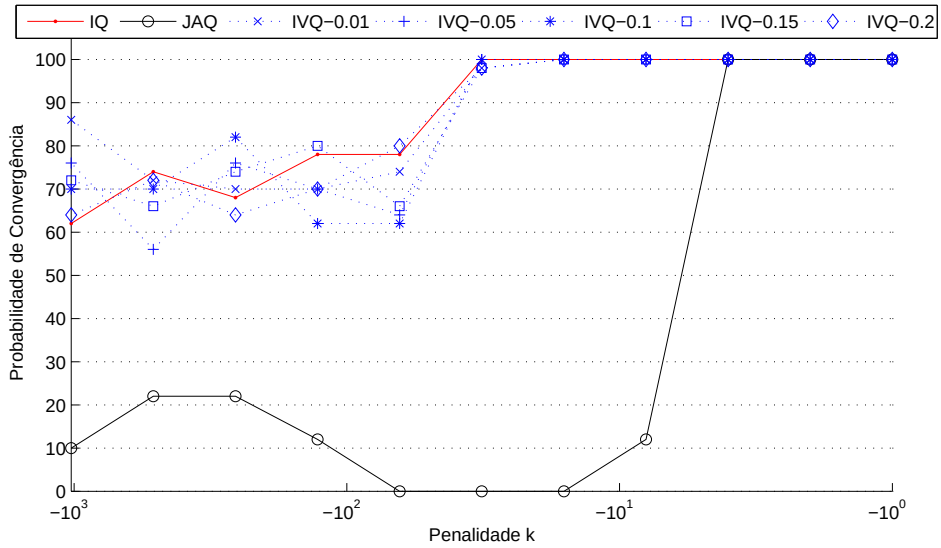


(a) Greedy

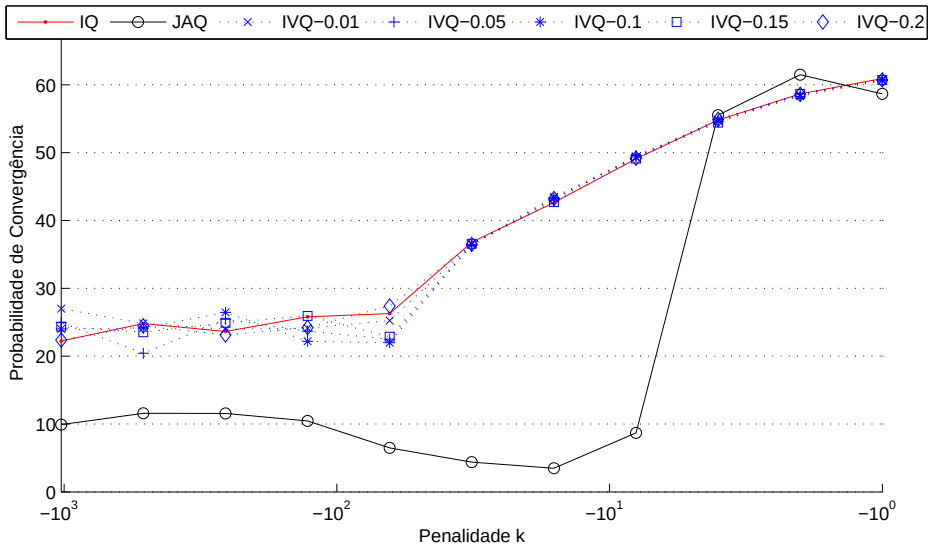


(b) Softmax

Figura A.11: Convergência no *Penalty Game* com 2 agentes, $T = 16(0.999^t)$ e $so = 5$

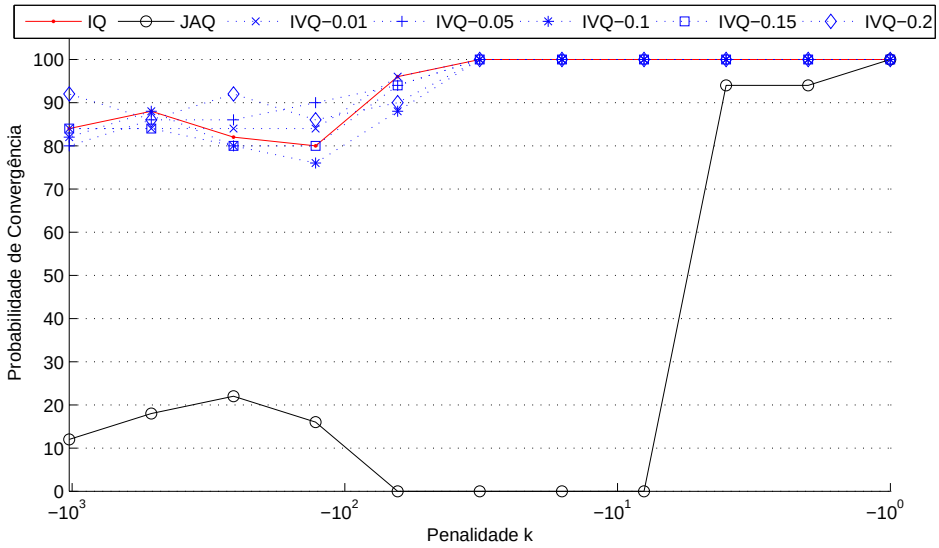


(a) Greedy

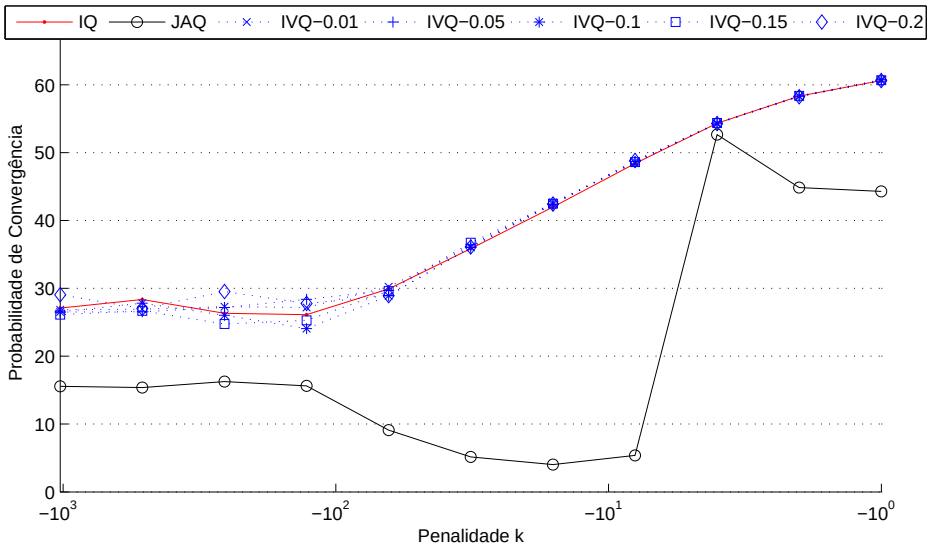


(b) Softmax

Figura A.12: Convergência no *Penalty Game* com 2 agentes, $T = 16(0.9995^t)$ e $so = 5$

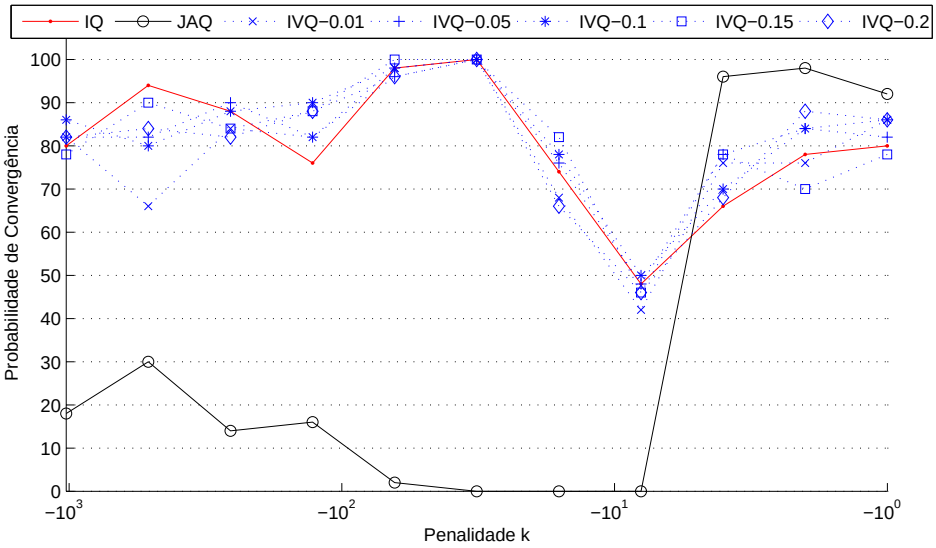


(a) Greedy

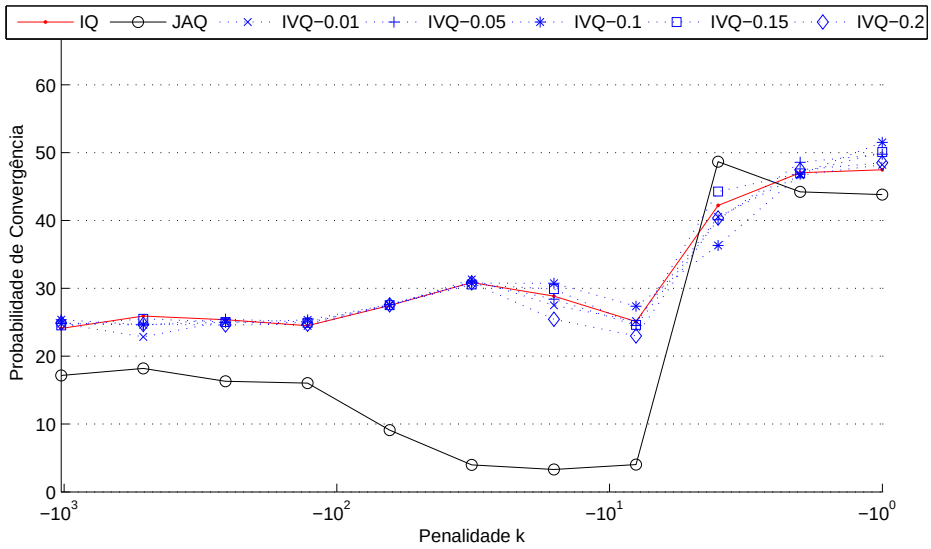


(b) Softmax

Figura A.13: Convergência no *Penalty Game* com 2 agentes, $T = 16(0.9998^t)$ e $so = 5$



(a) Greedy



(b) Softmax

Figura A.14: Convergência no *Penalty Game* com 2 agentes, $T = 16(0.9999^t)$ e $so = 5$

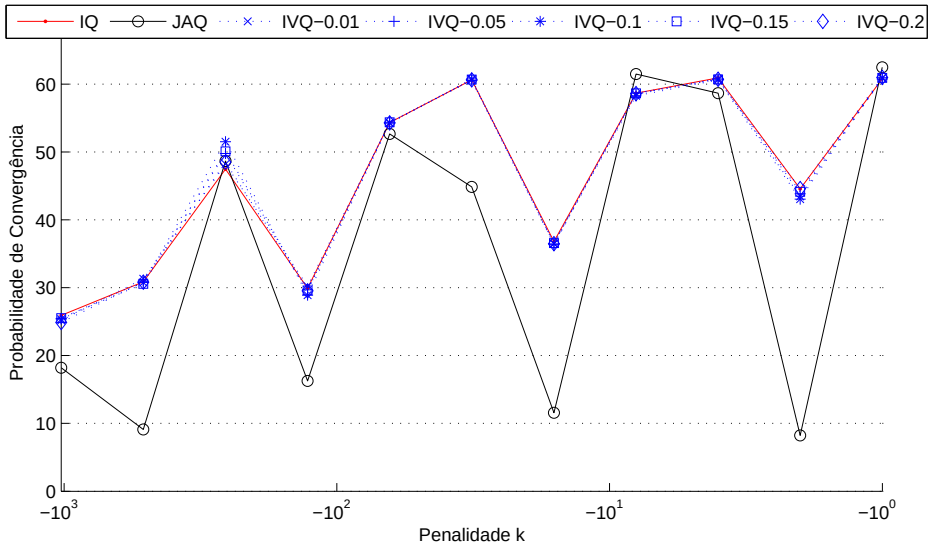
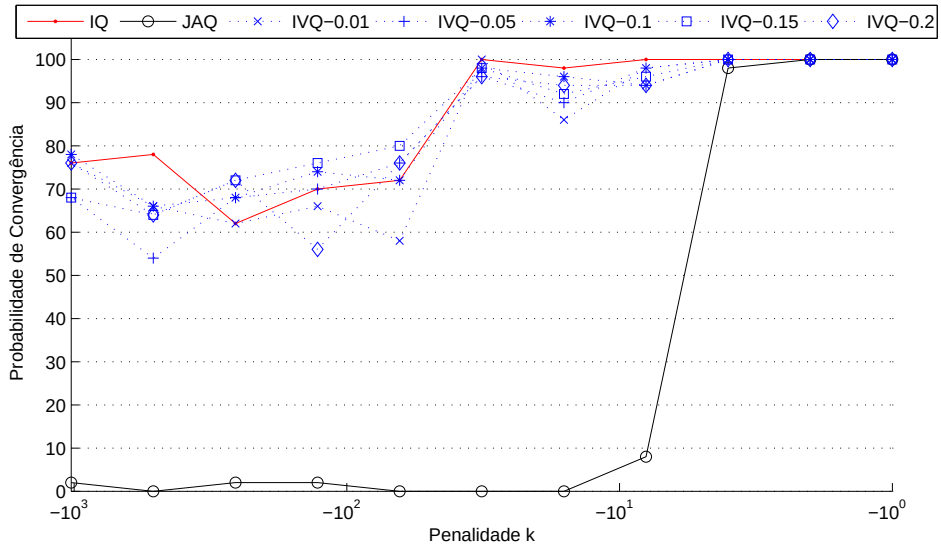
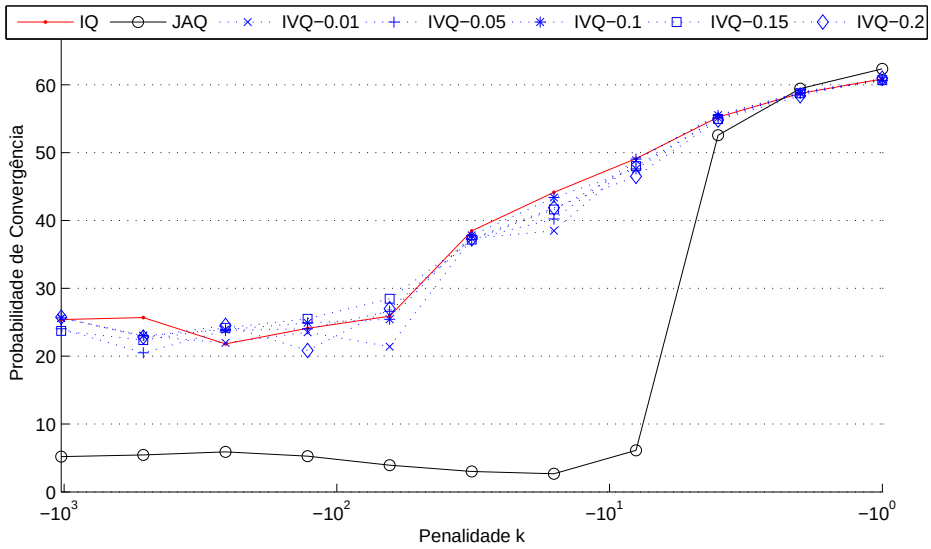


Figura A.15: Melhores algoritmos no *Penalty Game* com 2 agentes para $so = 5$

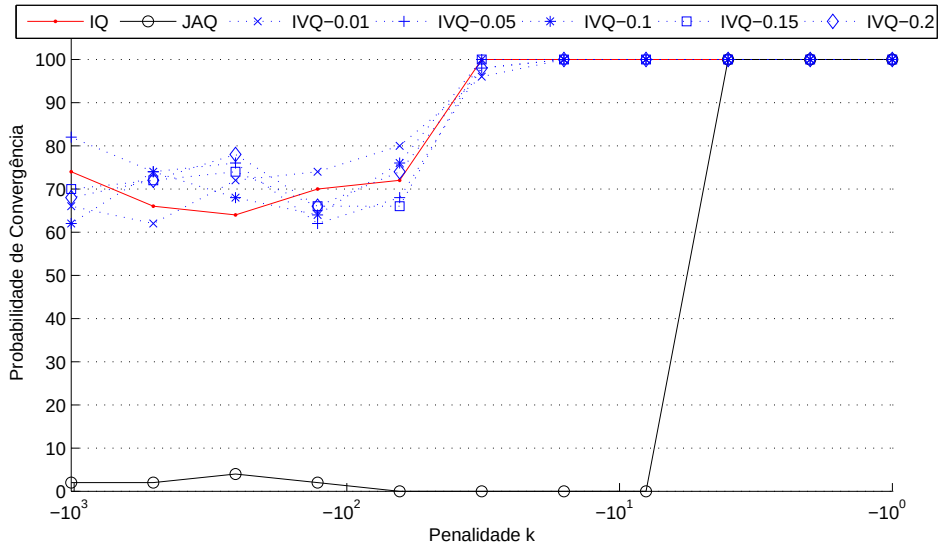


(a) Greedy

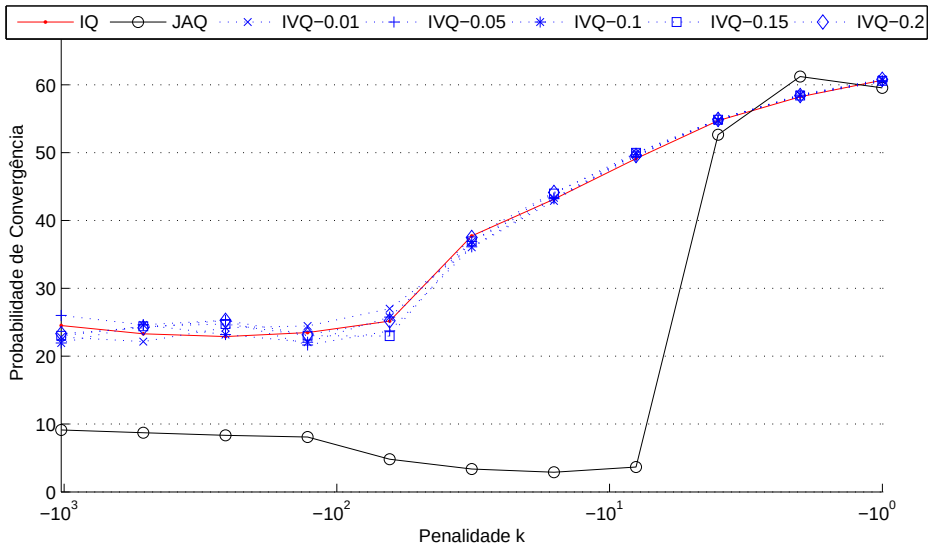


(b) Softmax

Figura A.16: Convergência no *Penalty Game* com 2 agentes, $T = 16(0.999^t)$ e $so = 6$

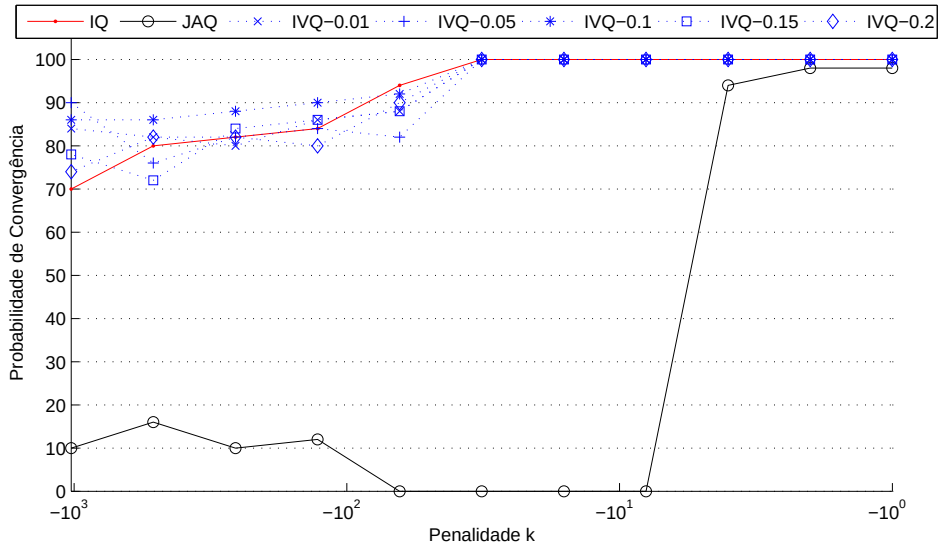


(a) Greedy

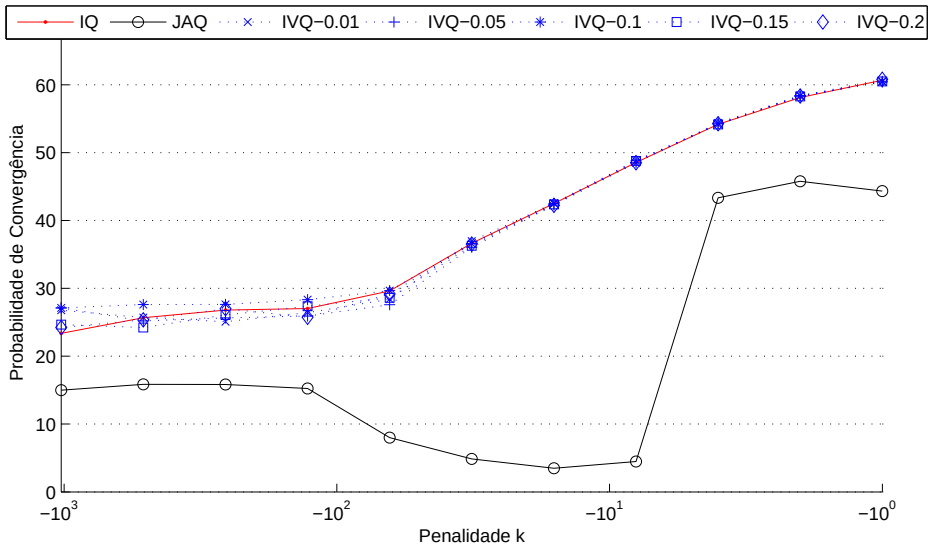


(b) Softmax

Figura A.17: Convergência no *Penalty Game* com 2 agentes, $T = 16(0.9995^t)$ e $so = 6$

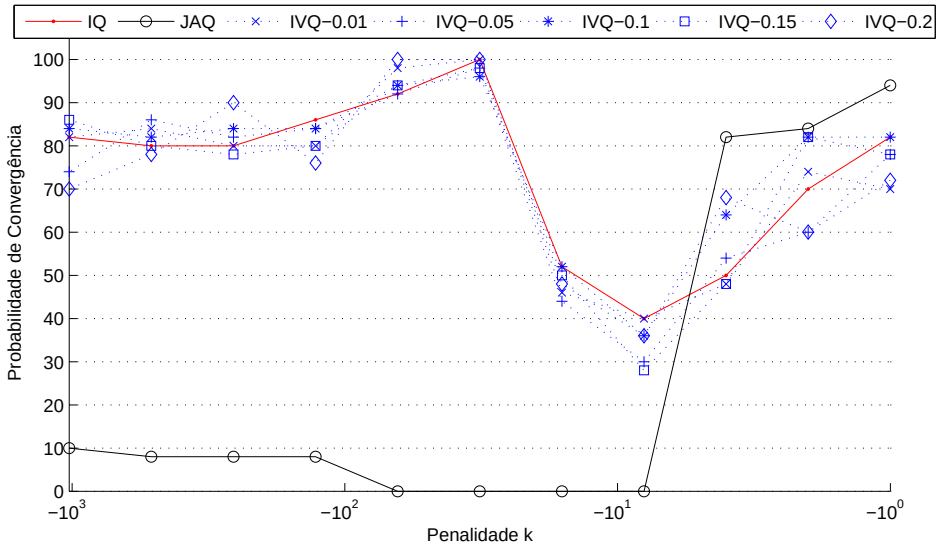


(a) Greedy

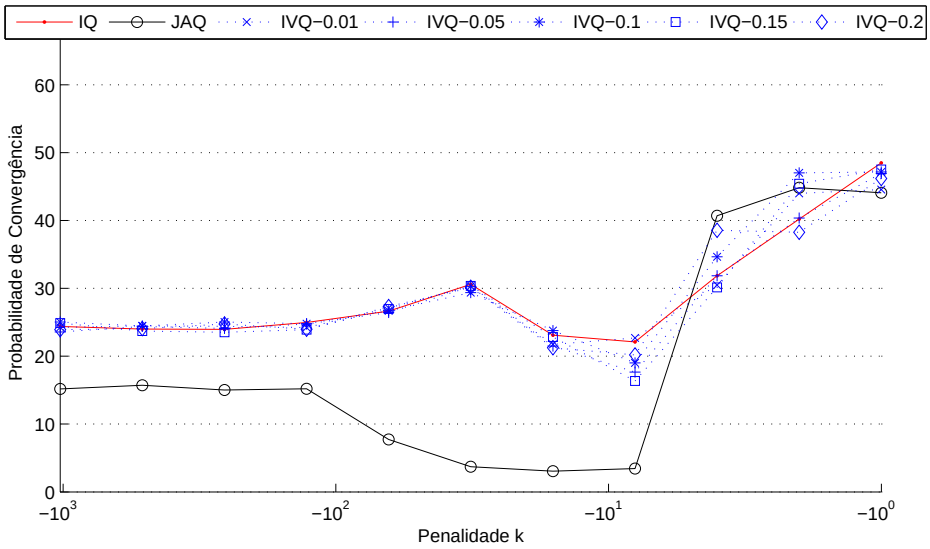


(b) Softmax

Figura A.18: Convergência no *Penalty Game* com 2 agentes, $T = 16(0.9998^t)$ e $so = 6$



(a) Greedy



(b) Softmax

Figura A.19: Convergência no *Penalty Game* com 2 agentes, $T = 16(0.9999^t)$ e $so = 6$

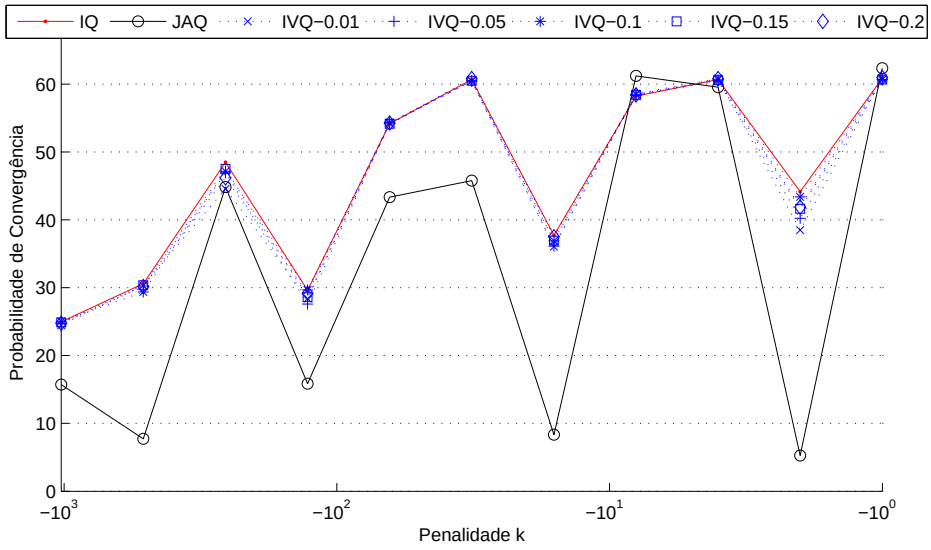
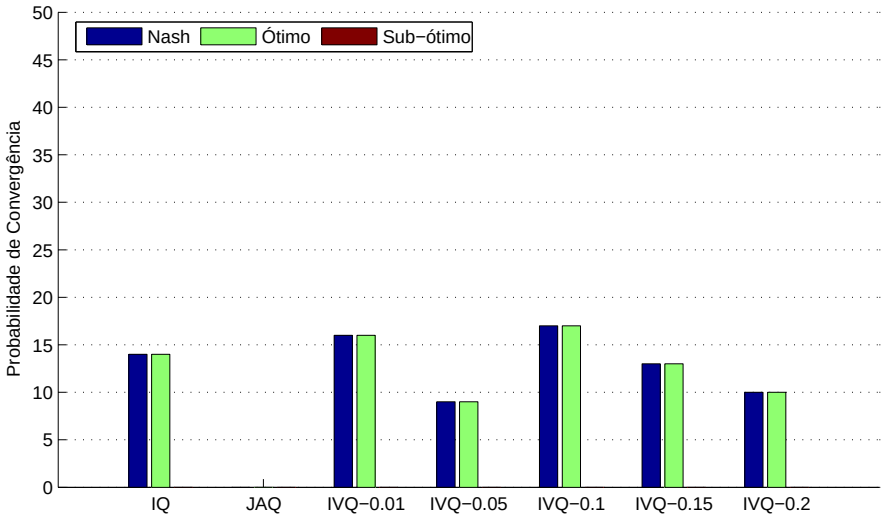
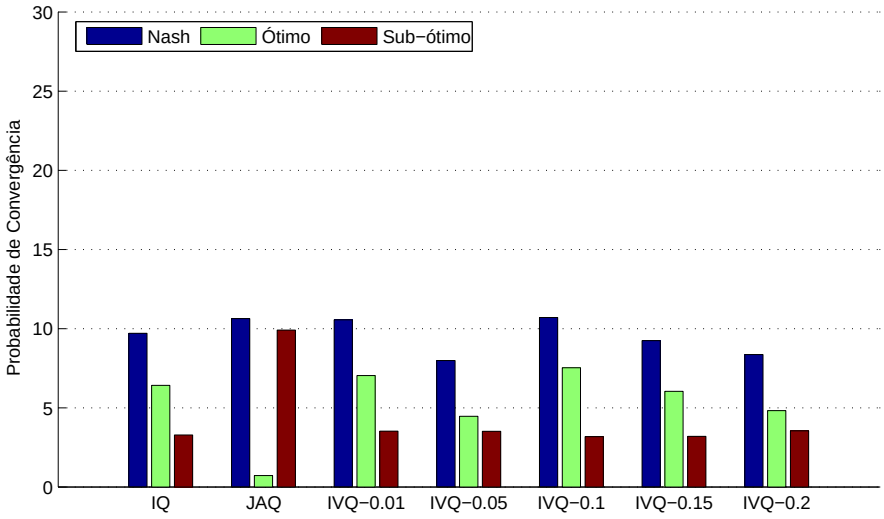


Figura A.20: Melhores algoritmos no *Penalty Game* com 2 agentes para $so = 6$

A.1.2 Climbing Game

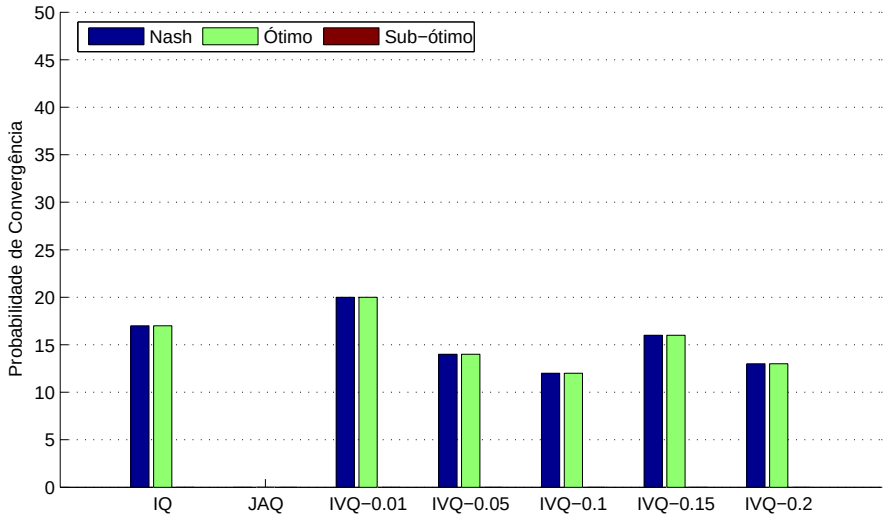


(a) Greedy

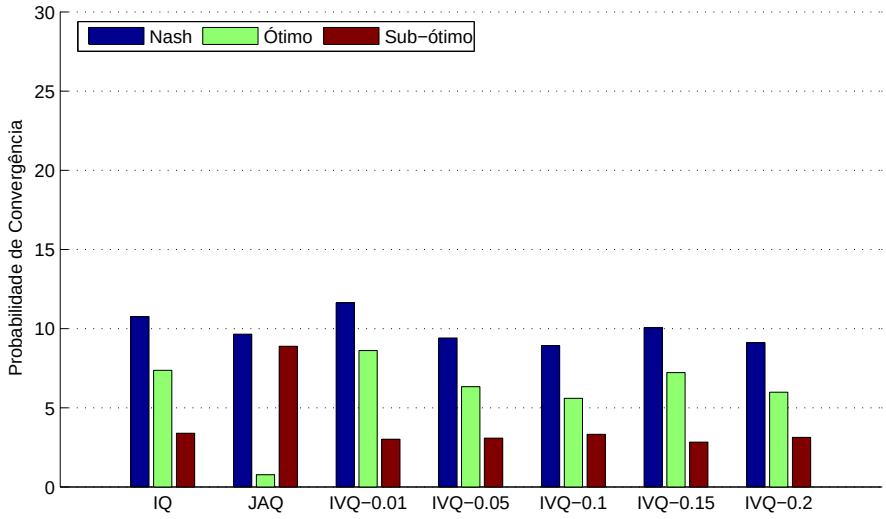


(b) Softmax

Figura A.21: Convergência no *Climbing Game* com 2 agentes, $T = 16(0.9992^t)$

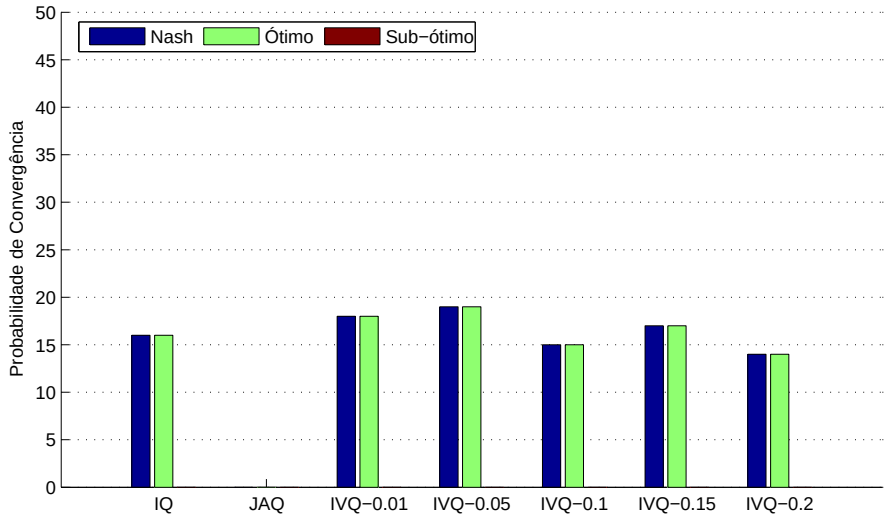


(a) Greedy

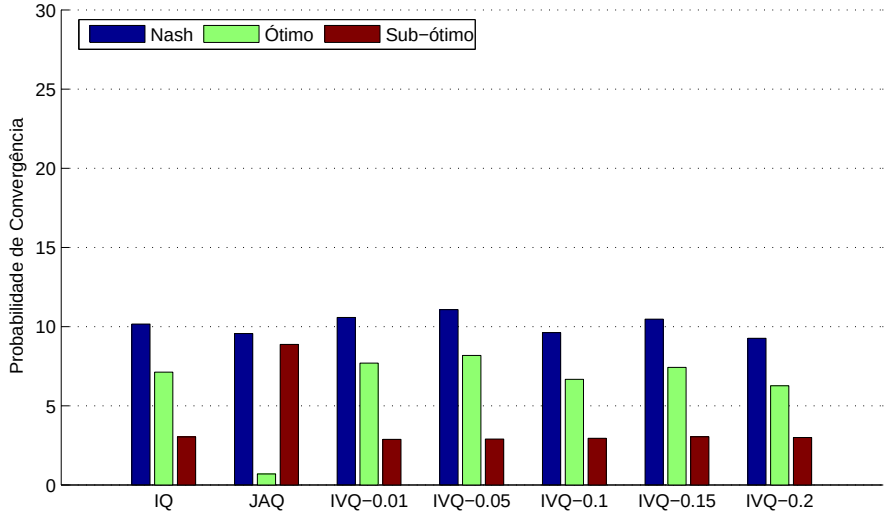


(b) Softmax

Figura A.22: Convergência no *Climbing Game* com 2 agentes, $T = 16(0.9993^t)$

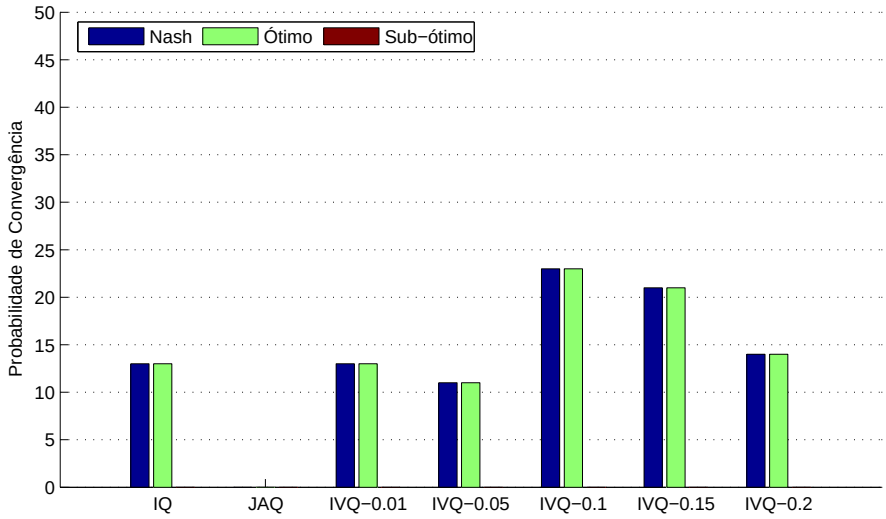


(a) Greedy

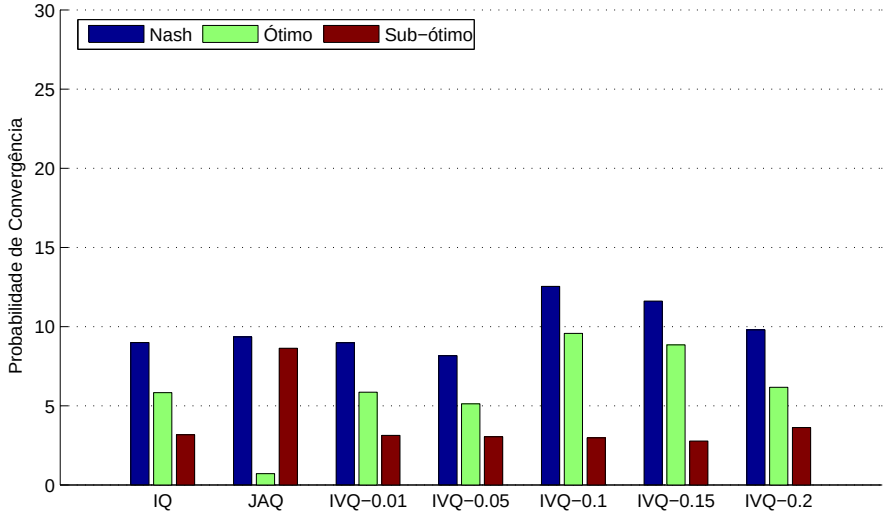


(b) Softmax

Figura A.23: Convergência no *Climbing Game* com 2 agentes, $T = 16(0.9994^t)$

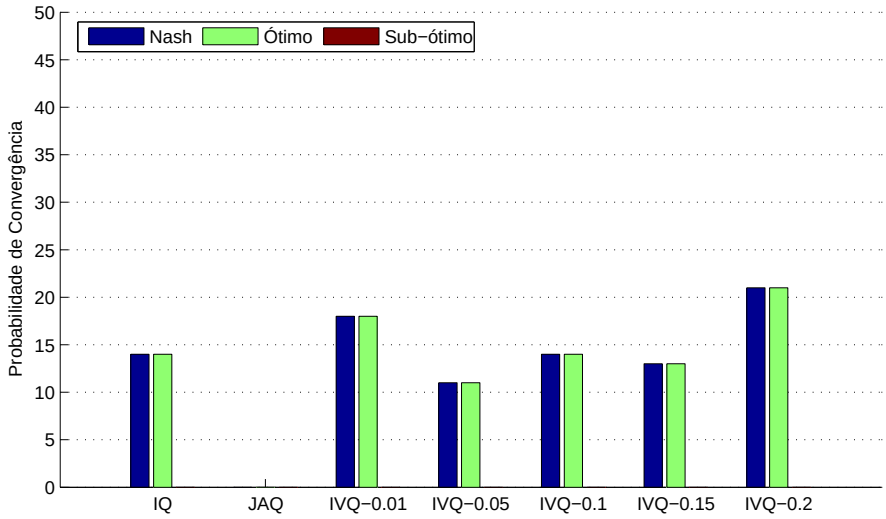


(a) Greedy

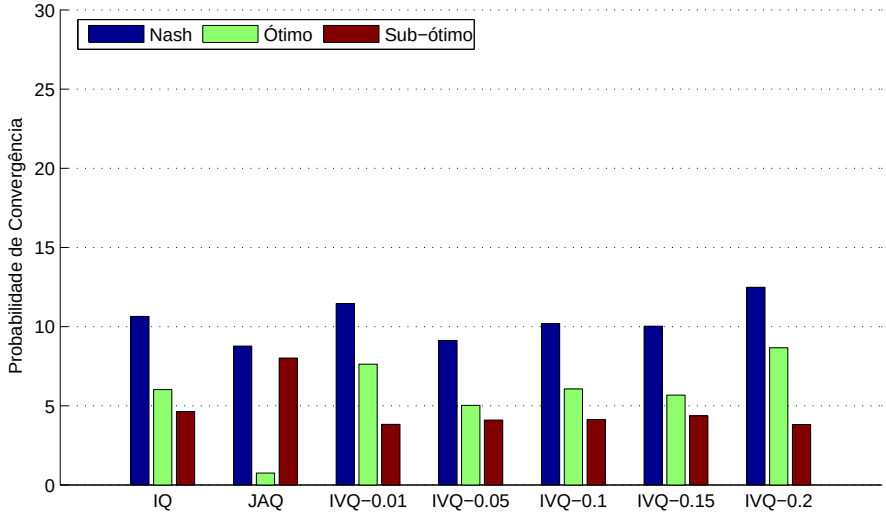


(b) Softmax

Figura A.24: Convergência no *Climbing Game* com 2 agentes, $T = 16(0.9995^t)$

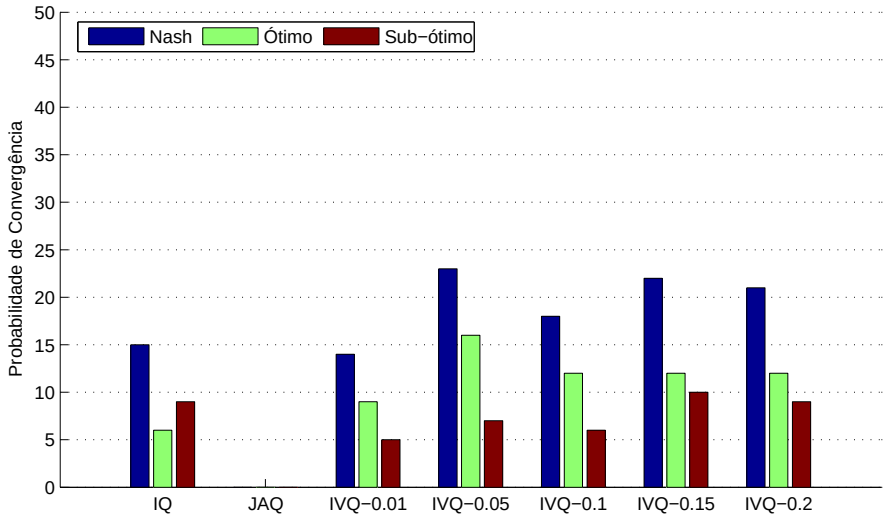


(a) Greedy

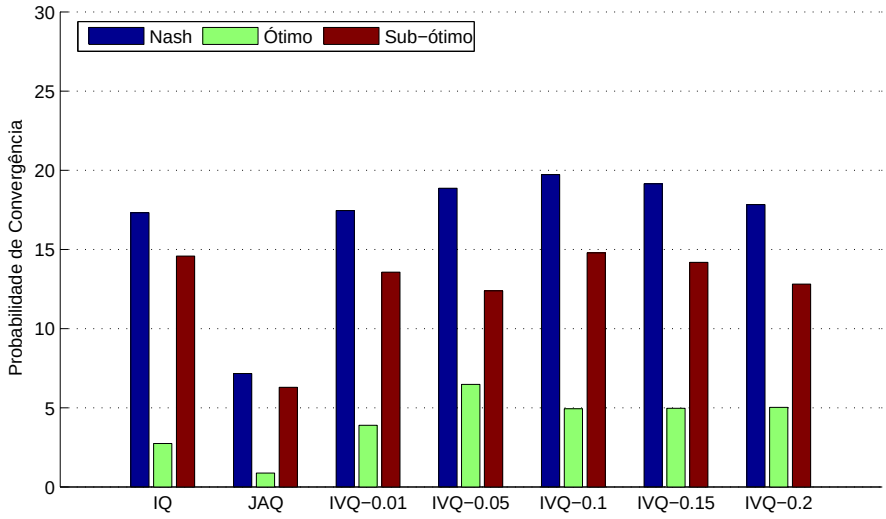


(b) Softmax

Figura A.25: Convergência no *Climbing Game* com 2 agentes, $T = 16(0.9996^t)$

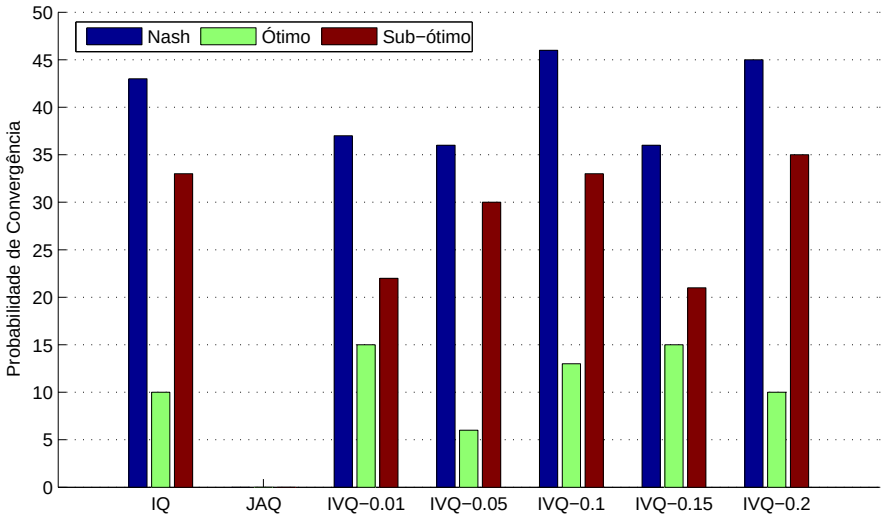


(a) Greedy

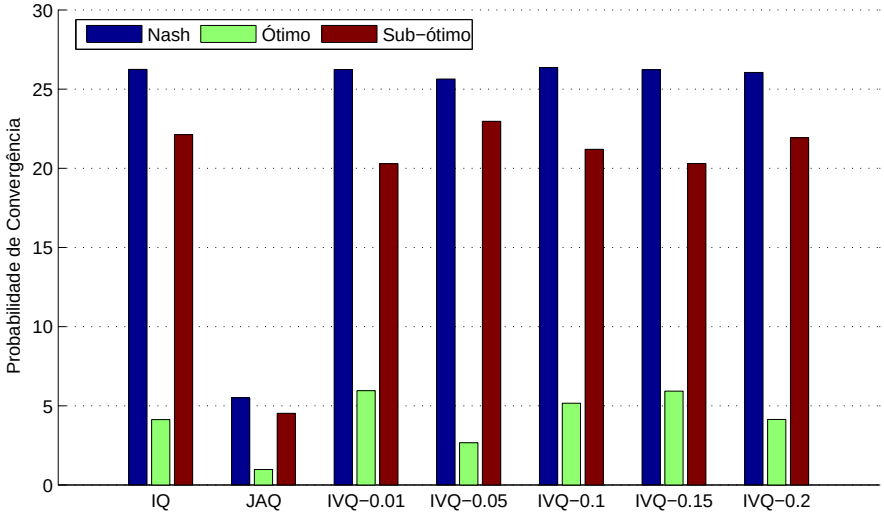


(b) Softmax

Figura A.26: Convergência no *Climbing Game* com 2 agentes, $T = 16(0.9997^t)$



(a) Greedy

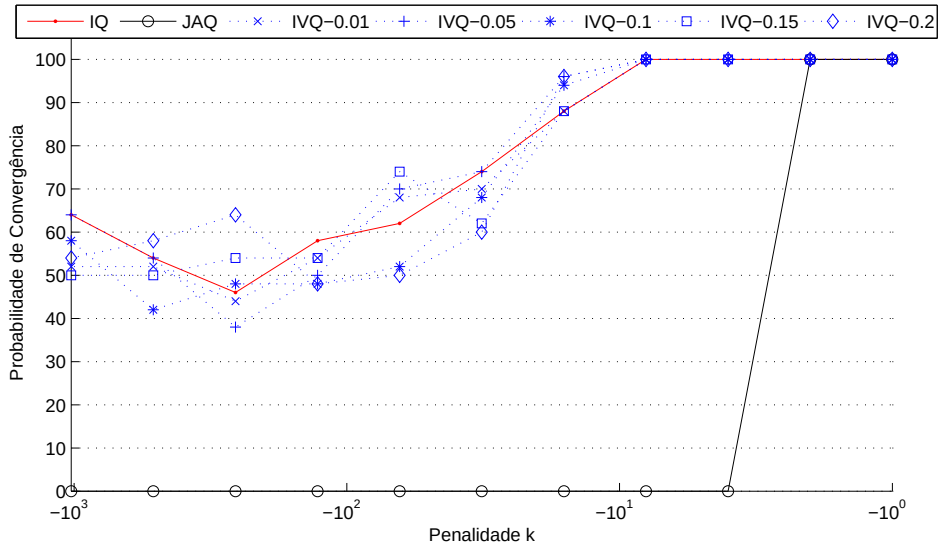


(b) Softmax

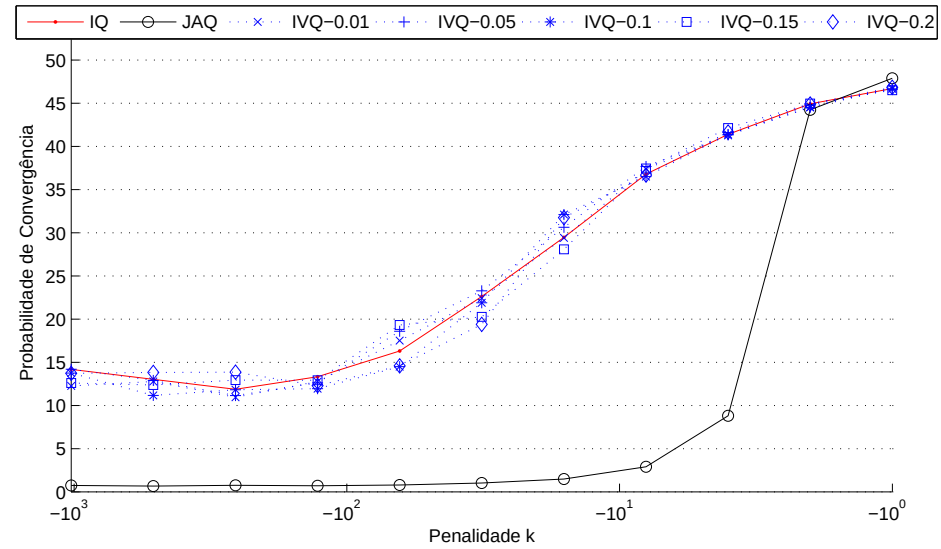
Figura A.27: Convergência no *Climbing Game* com 2 agentes, $T = 16(0.9998^t)$

A.2 Coordenação em jogos repetitivos com 3 agentes

A.2.1 Penalty Game

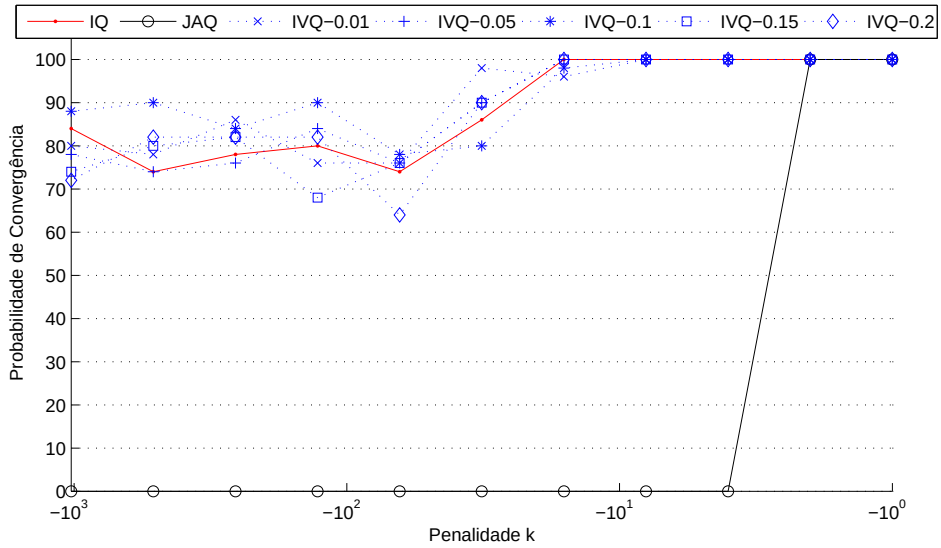


(a) Greedy

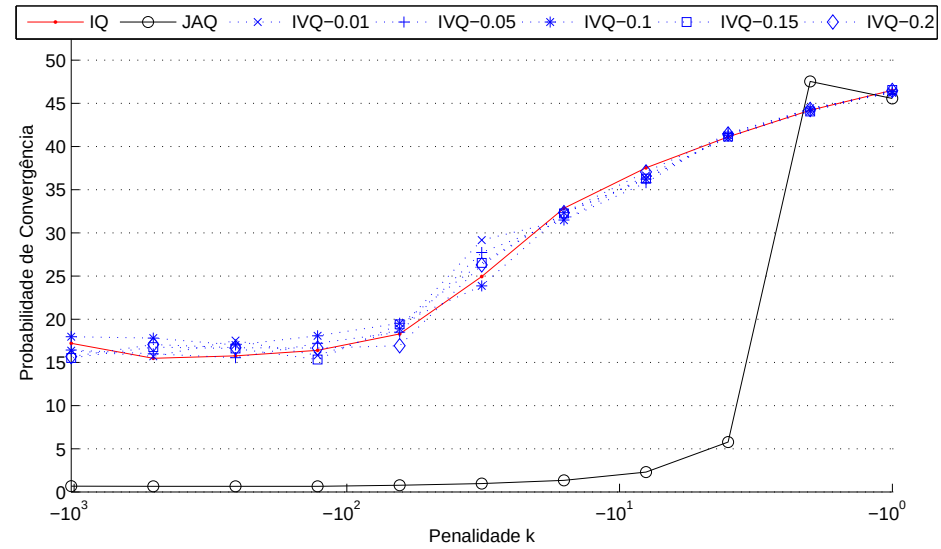


(b) Softmax

Figura A.28: Convergência no *Penalty Game* com 3 agentes, $T = 16(0.999^t)$ e $so = 2$

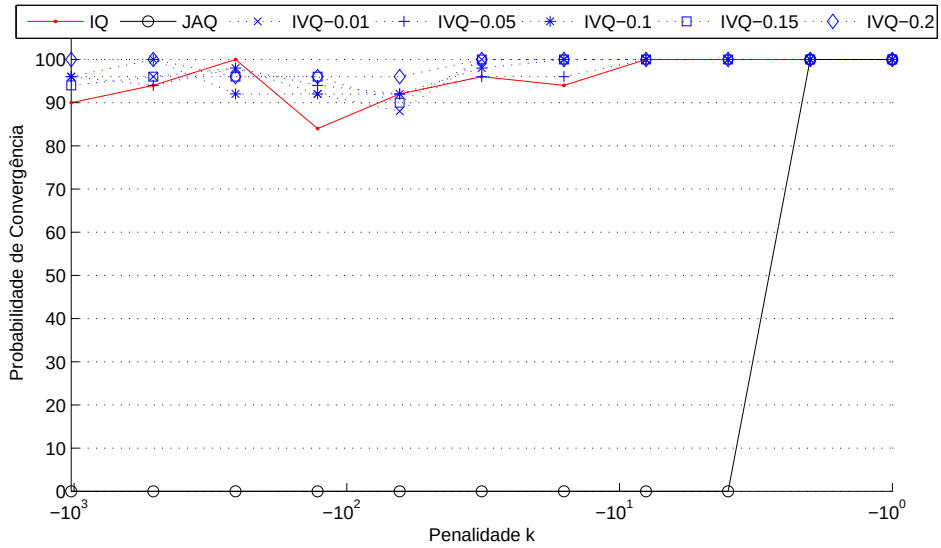


(a) Greedy

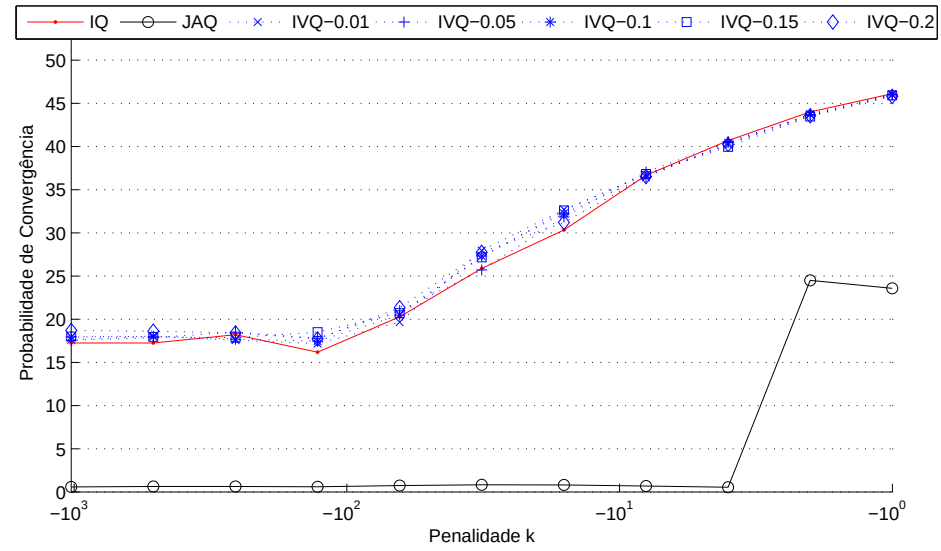


(b) Softmax

Figura A.29: Convergência no *Penalty Game* com 3 agentes, $T = 16(0.9995^t)$ e $so = 2$

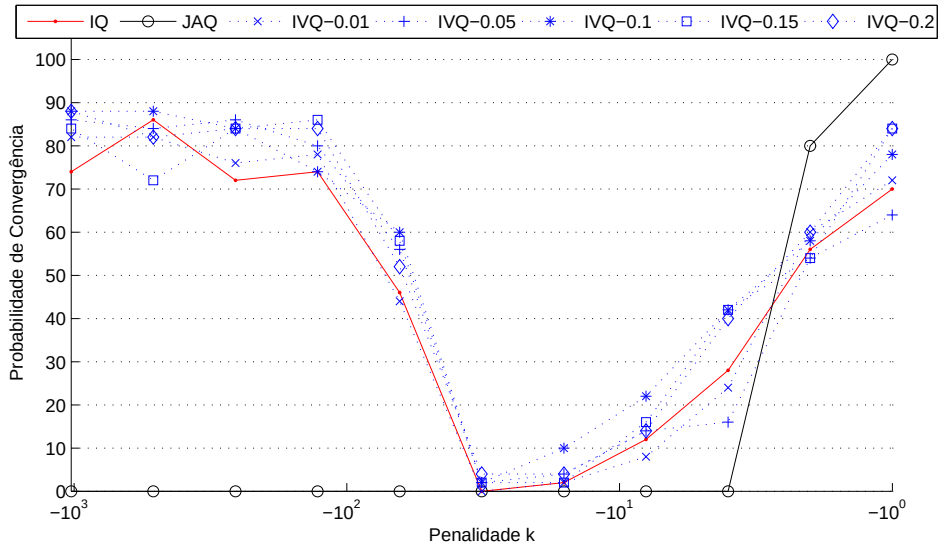


(a) Greedy

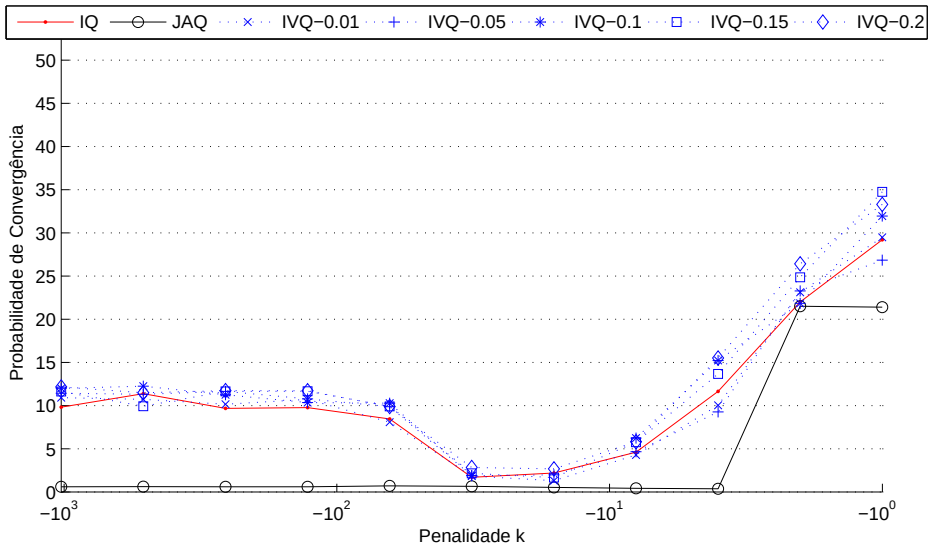


(b) Softmax

Figura A.30: Convergência no *Penalty Game* com 3 agentes, $T = 16(0.9998^t)$ e $so = 2$



(a) Greedy



(b) Softmax

Figura A.31: Convergência no *Penalty Game* com 3 agentes, $T = 16(0.9999^t)$ e $so = 2$

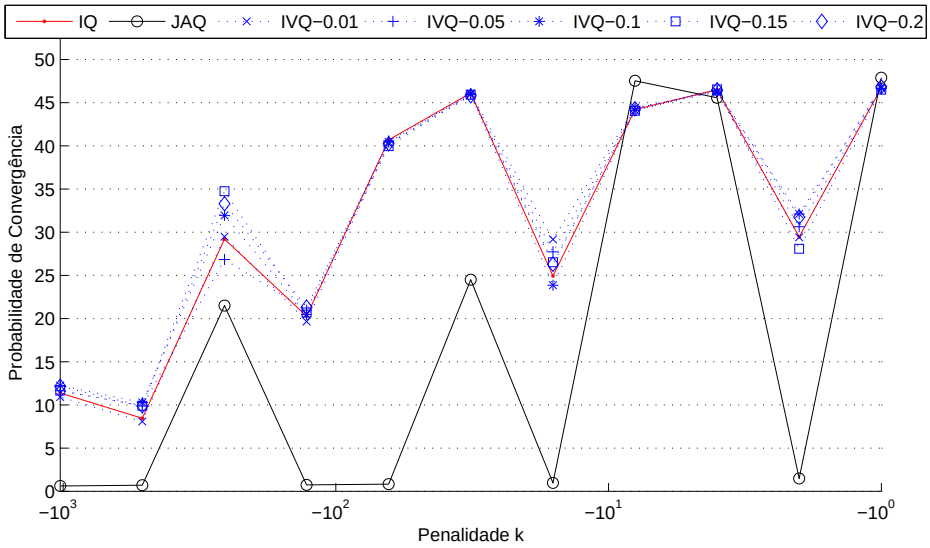
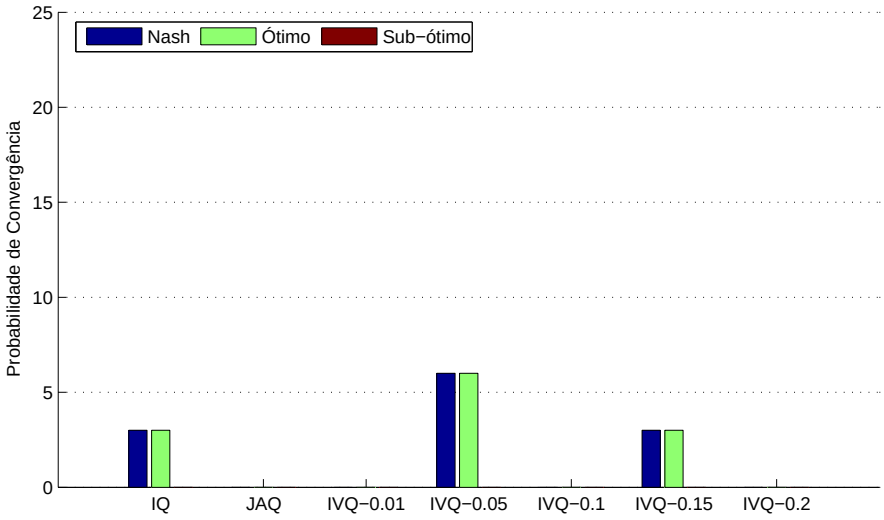
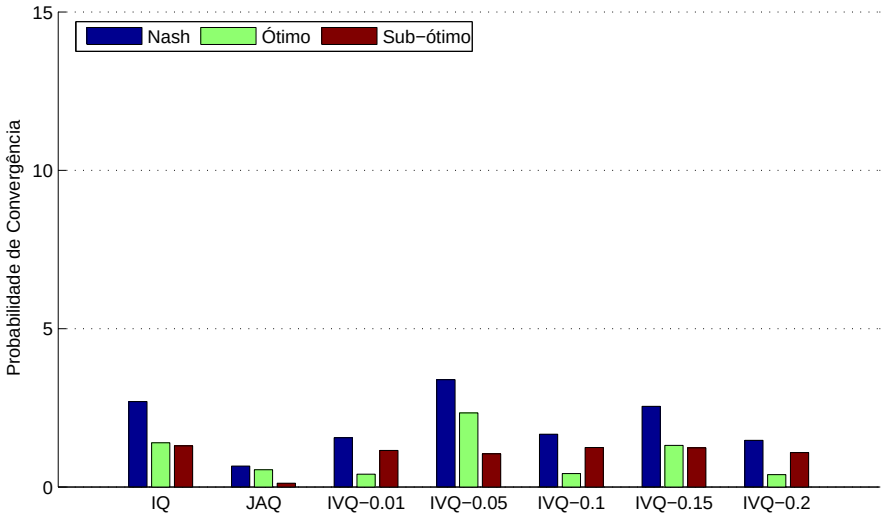


Figura A.32: Melhores algoritmos no *Penalty Game* com 3 agentes para $so = 2$

A.2.2 Climbing Game

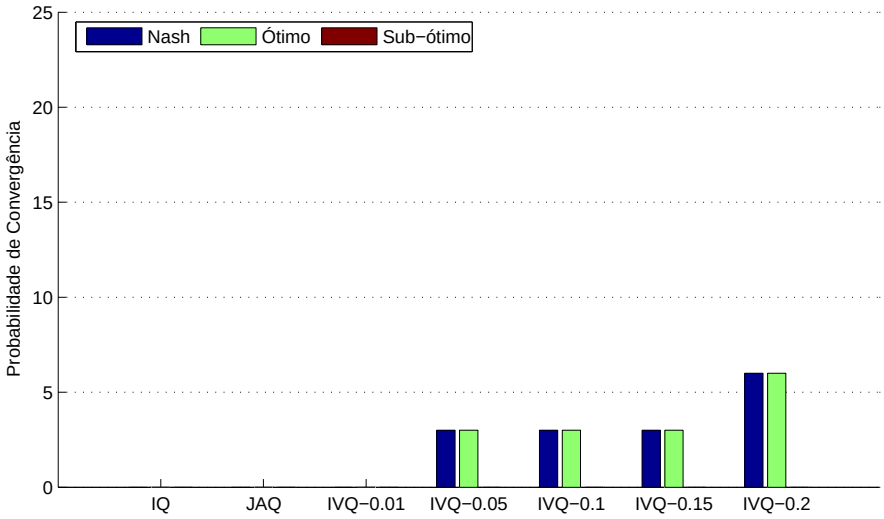


(a) Greedy

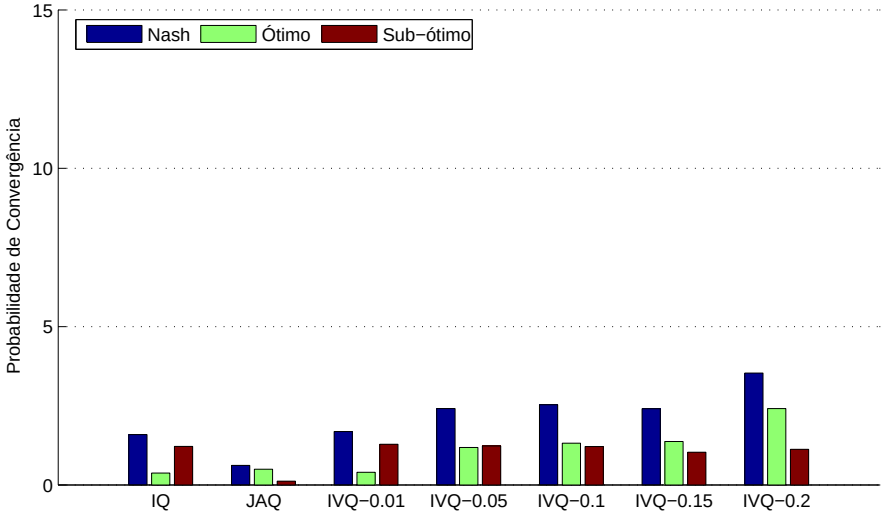


(b) Softmax

Figura A.33: Convergência no *Climbing Game* com 3 agentes, $T = 16(0.999^t)$

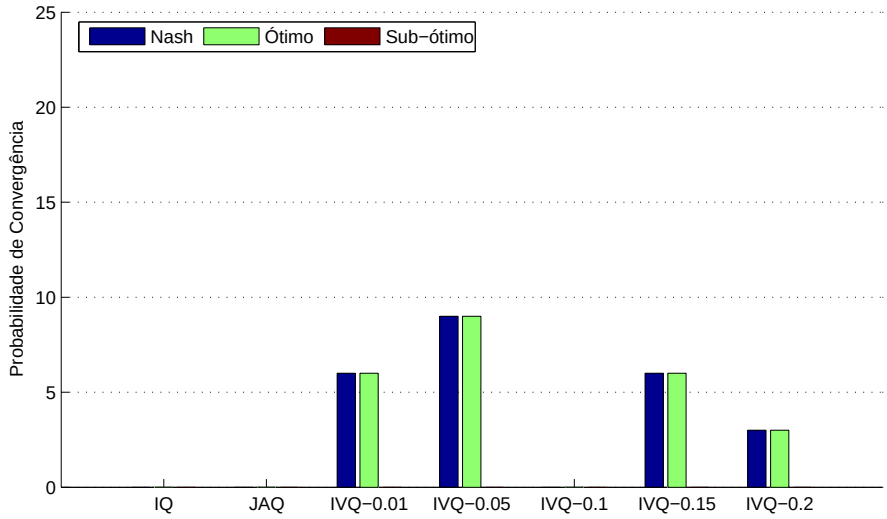


(a) Greedy

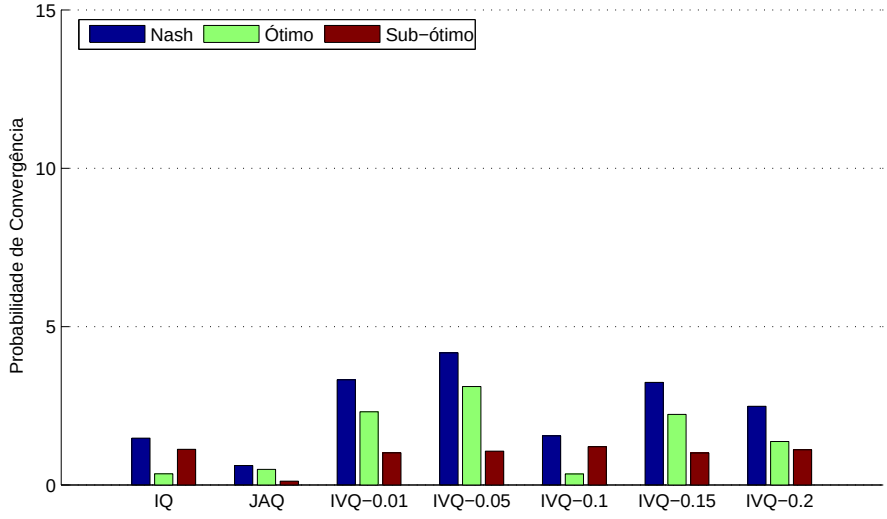


(b) Softmax

Figura A.34: Convergência no *Climbing Game* com 3 agentes, $T = 16(0.9991^t)$

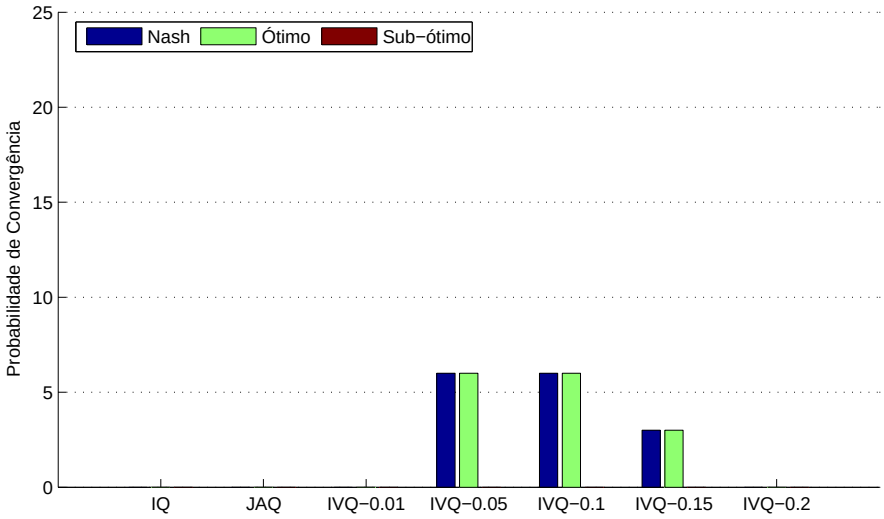


(a) Greedy

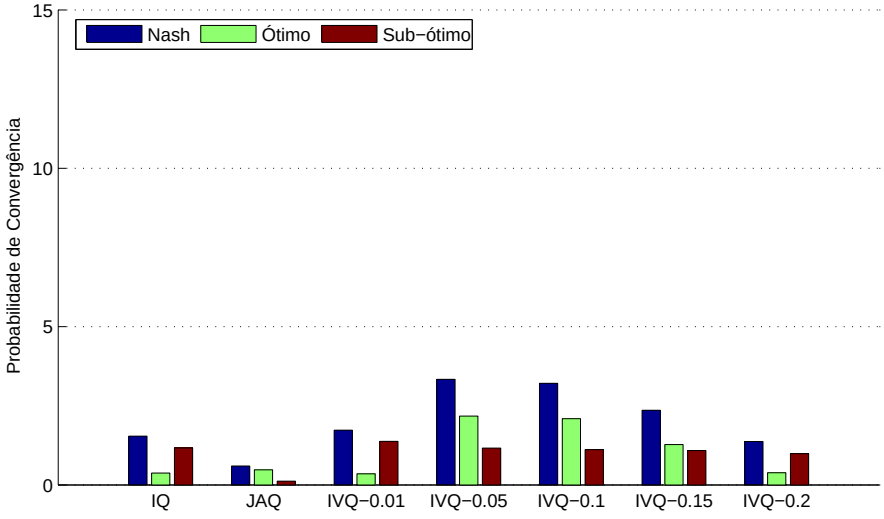


(b) Softmax

Figura A.35: Convergência no *Climbing Game* com 3 agentes, $T = 16(0.9992^t)$

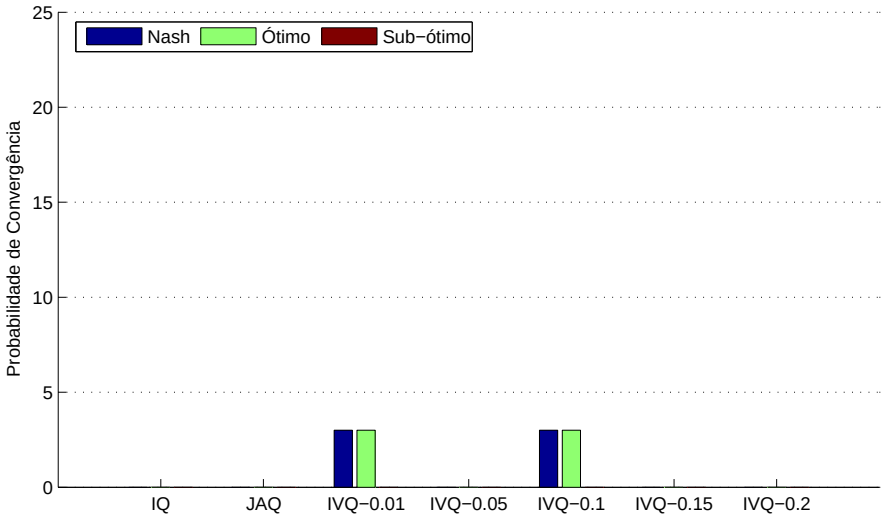


(a) Greedy

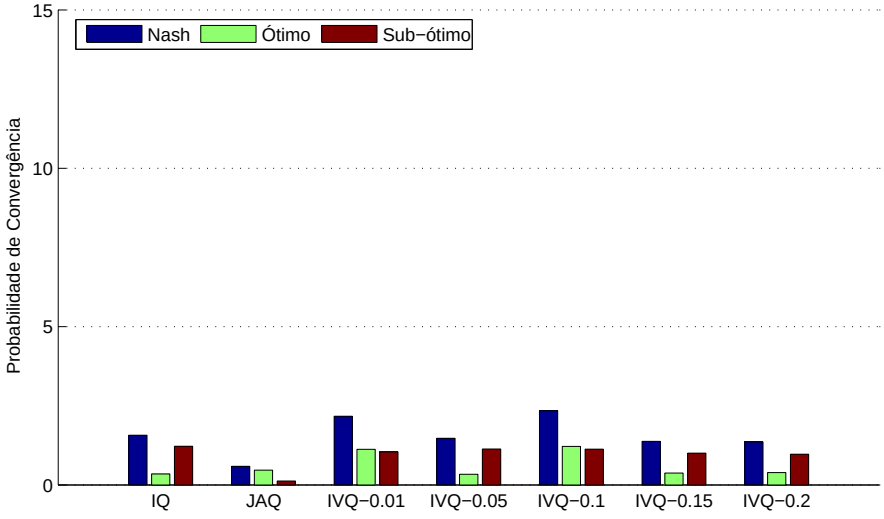


(b) Softmax

Figura A.36: Convergência no *Climbing Game* com 3 agentes, $T = 16(0.9993^t)$

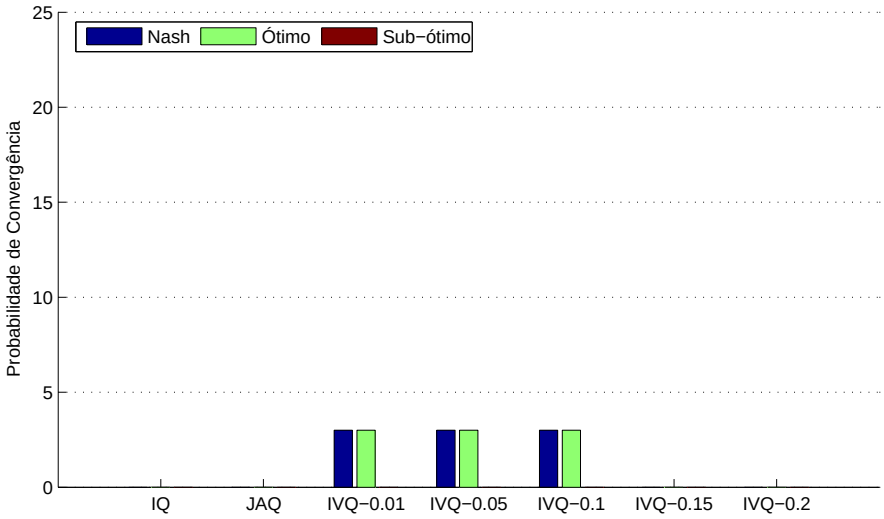


(a) Greedy

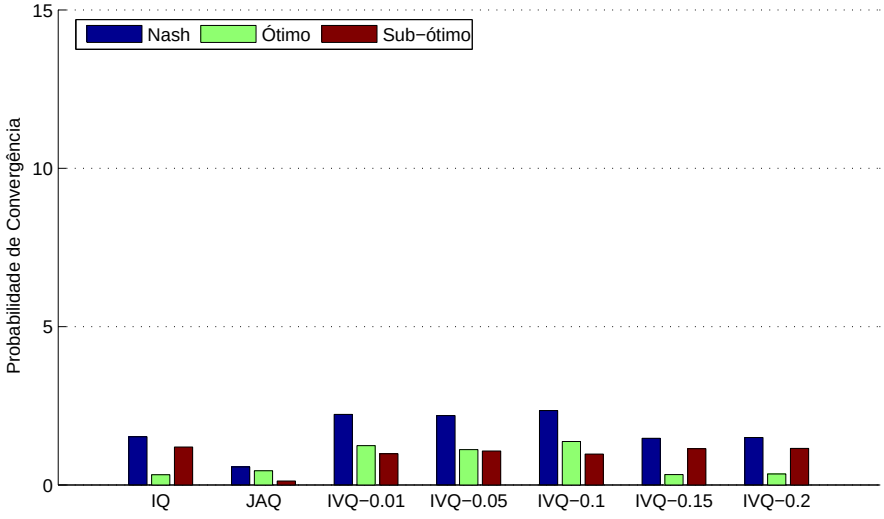


(b) Softmax

Figura A.37: Convergência no *Climbing Game* com 3 agentes, $T = 16(0.9994^t)$

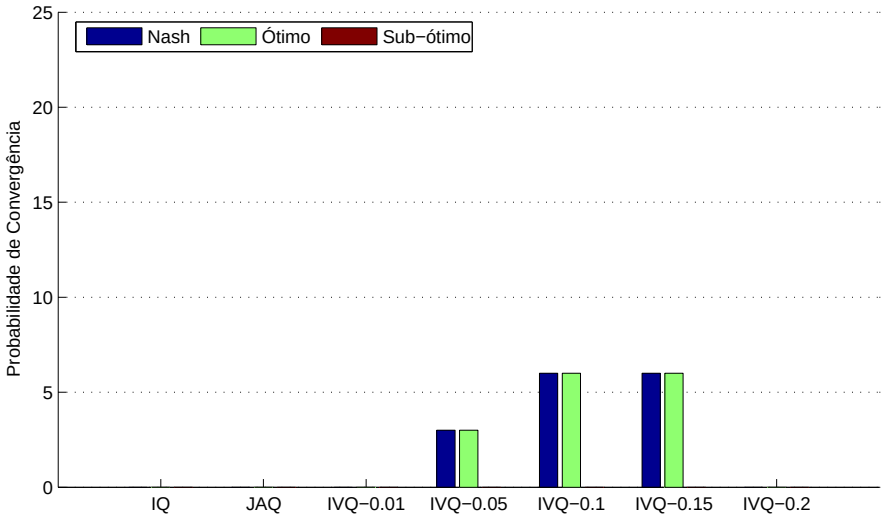


(a) Greedy

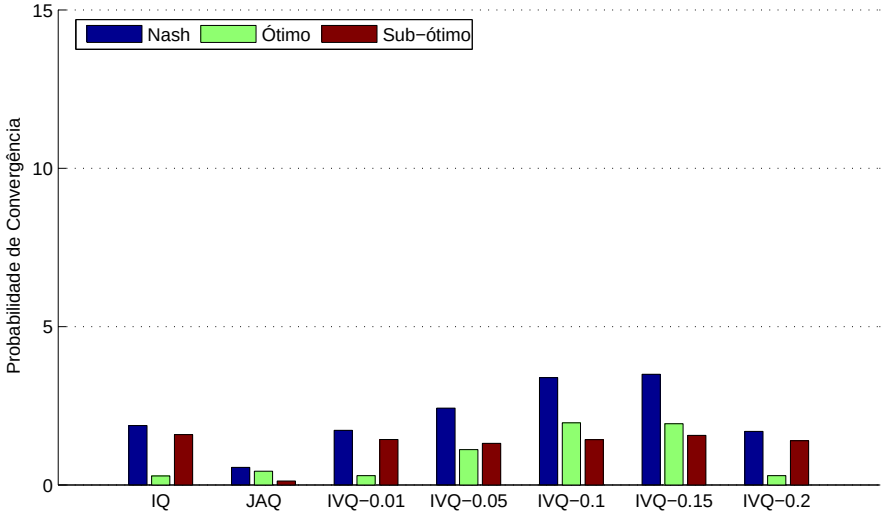


(b) Softmax

Figura A.38: Convergência no *Climbing Game* com 3 agentes, $T = 16(0.9995^t)$

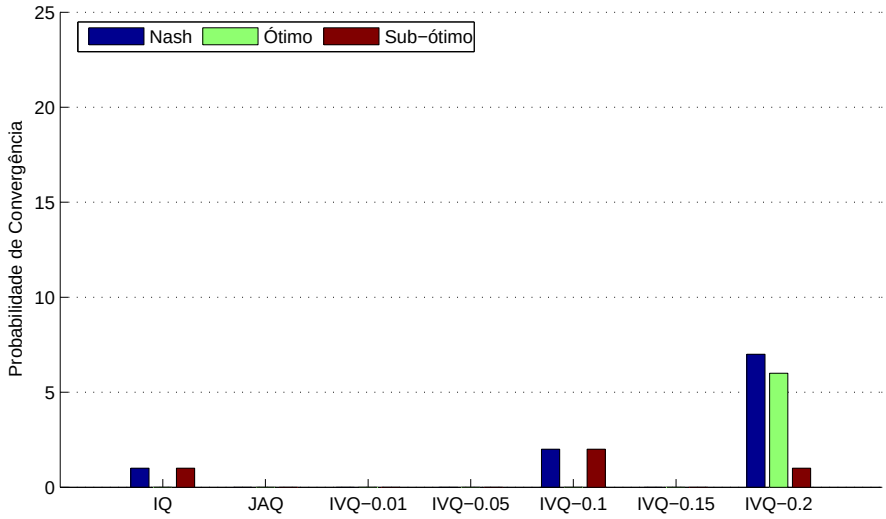


(a) Greedy

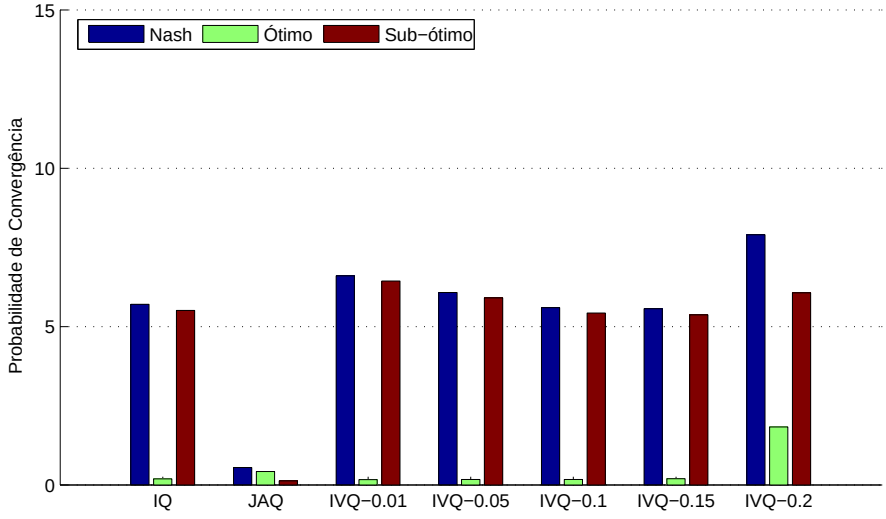


(b) Softmax

Figura A.39: Convergência no *Climbing Game* com 3 agentes, $T = 16(0.9996^t)$

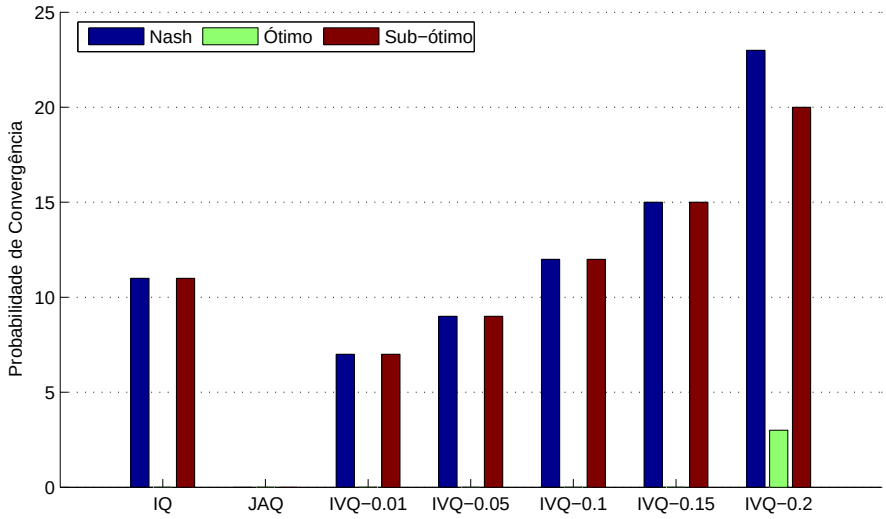


(a) Greedy

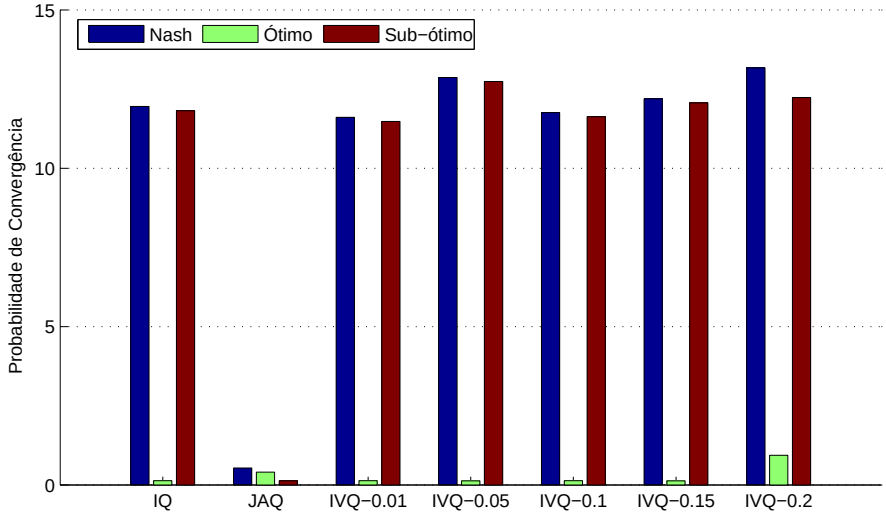


(b) Softmax

Figura A.40: Convergência no *Climbing Game* com 3 agentes, $T = 16(0.9997^t)$

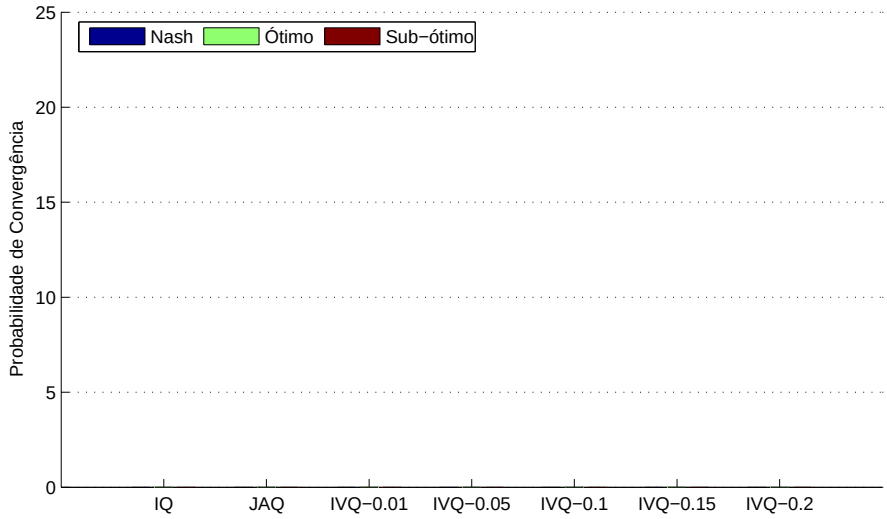


(a) Greedy

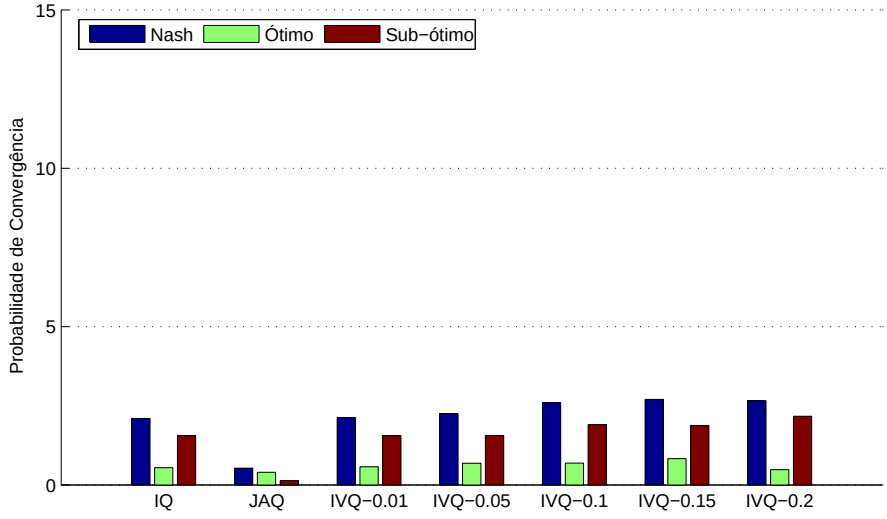


(b) Softmax

Figura A.41: Convergência no *Climbing Game* com 3 agentes, $T = 16(0.9998^t)$



(a) Greedy



(b) Softmax

Figura A.42: Convergência no *Climbing Game* com 3 agentes, $T = 16(0.9999^t)$

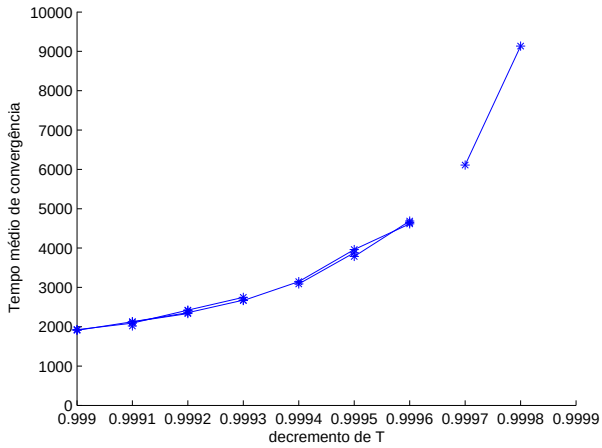
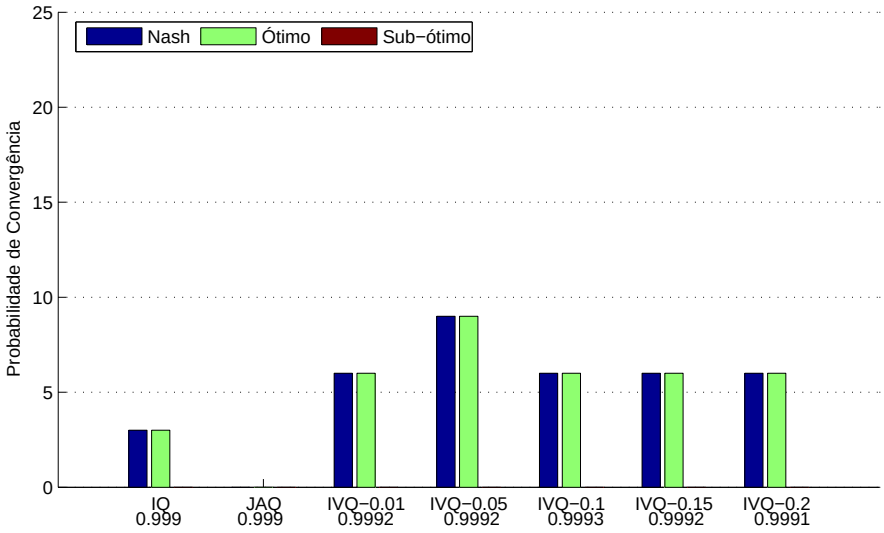
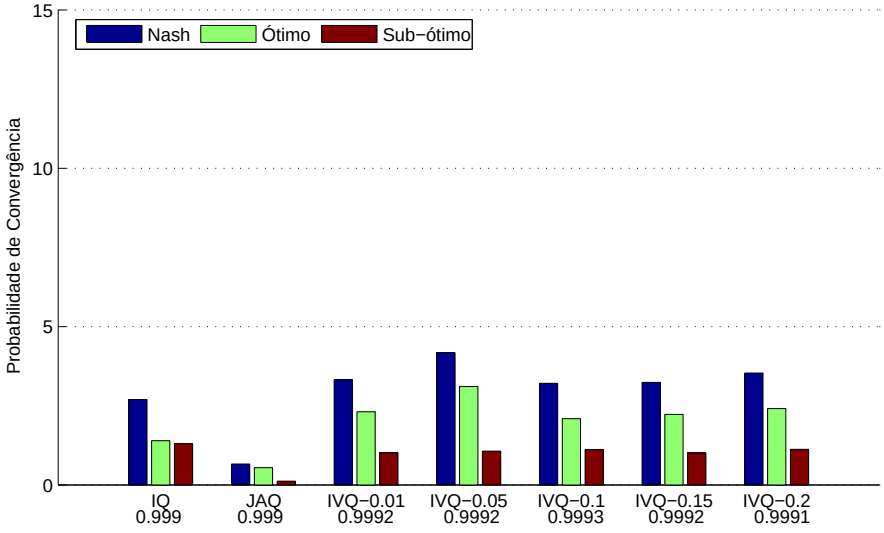


Figura A.43: Tempo de Convergência ao ótimo no *Climbing Game* com 3 agentes

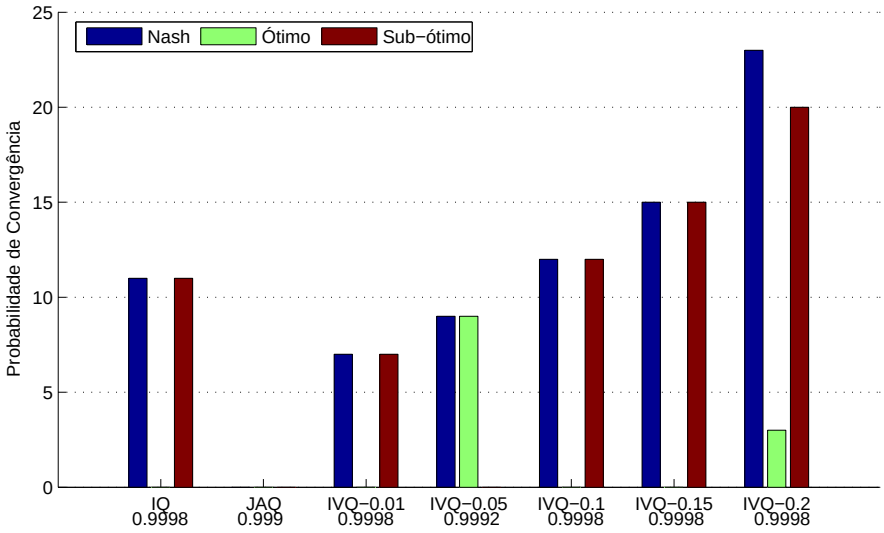


(a) Greedy

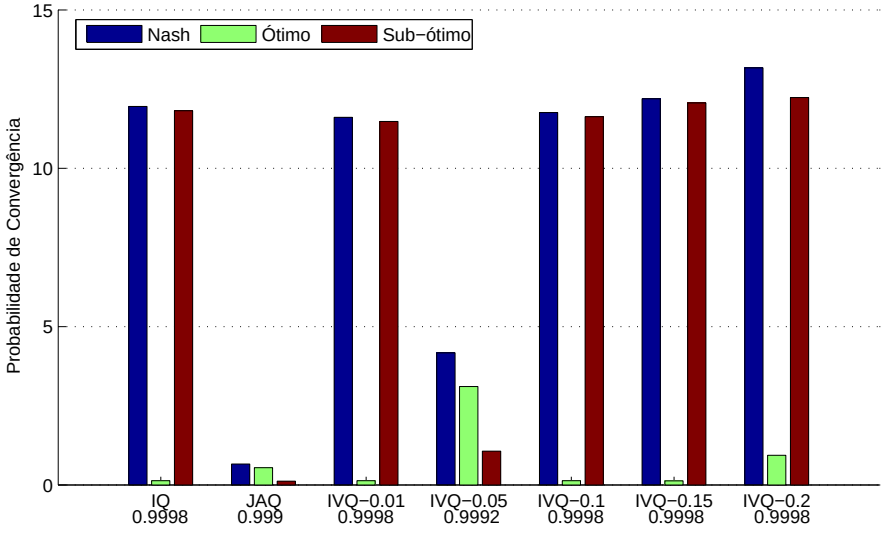


(b) Softmax

Figura A.44: Melhores algoritmos no *Climbing Game* com 3 agentes (equilíbrio ótimo)



(a) Greedy

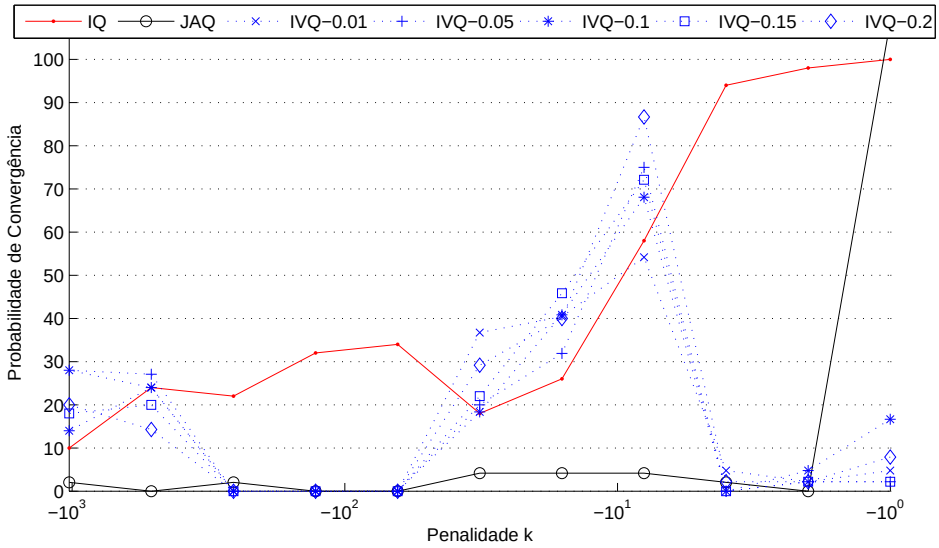


(b) Softmax

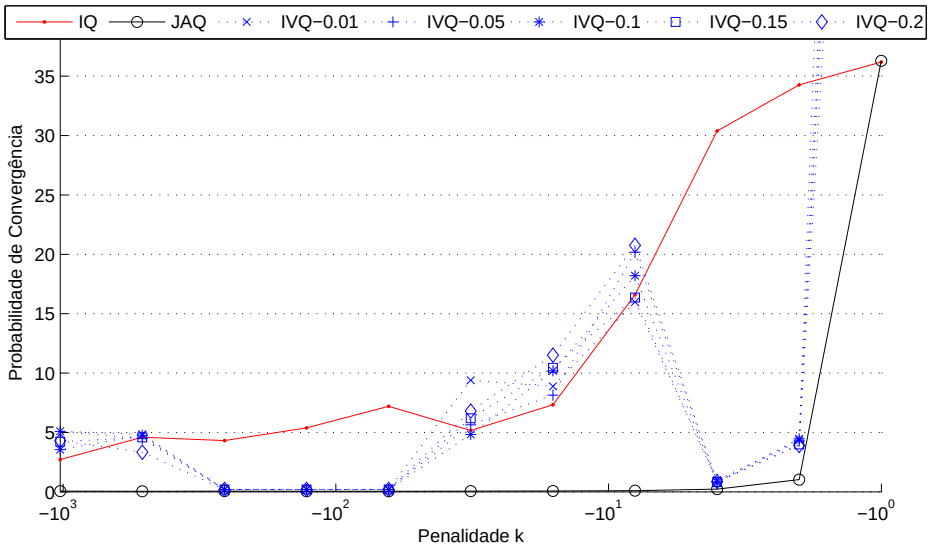
Figura A.45: Melhores algoritmos no *Climbing Game* com 3 agentes (equilíbrio de Nash)

A.3 Coordenação em jogos repetitivos com 4 agentes

A.3.1 Penalty Game

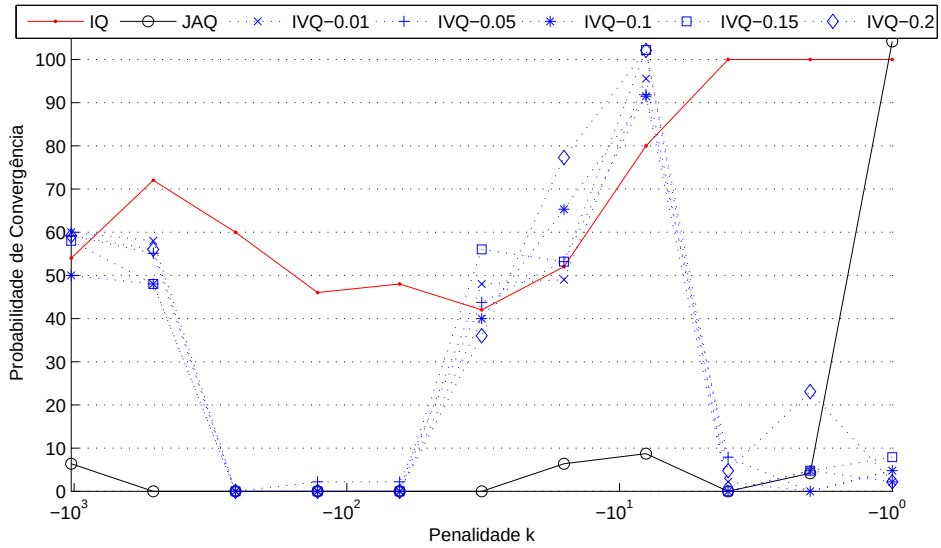


(a) Greedy

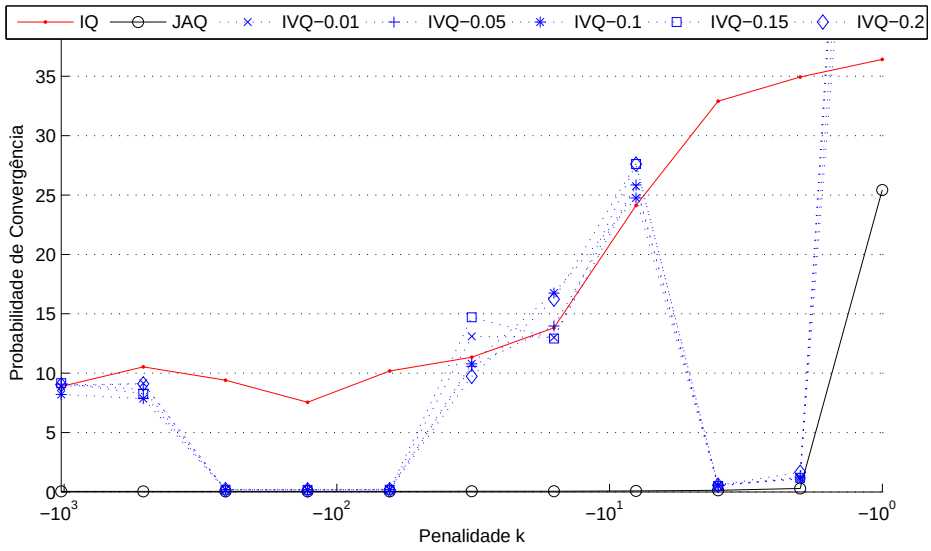


(b) Softmax

Figura A.46: Convergência no *Penalty Game* com 4 agentes, $T = 16(0.999^t)$ e $so = 2$

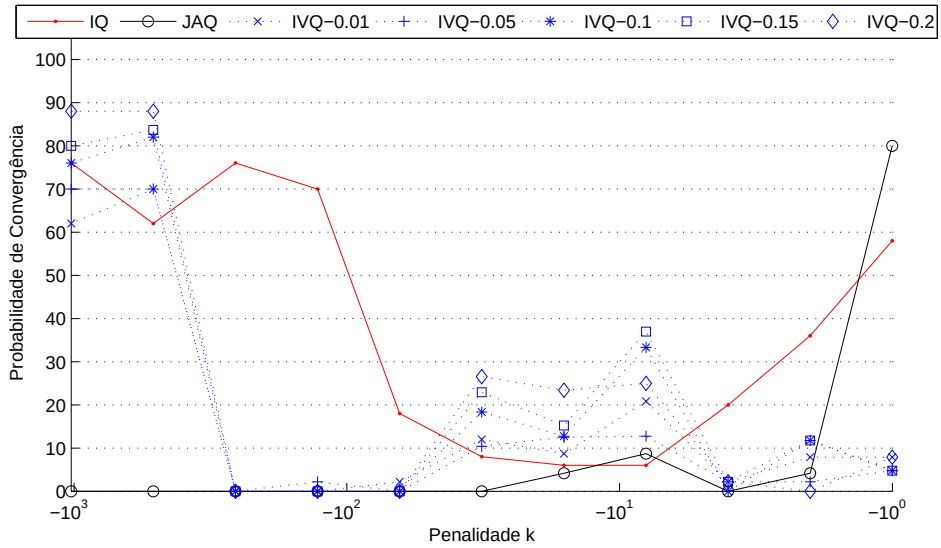


(a) Greedy

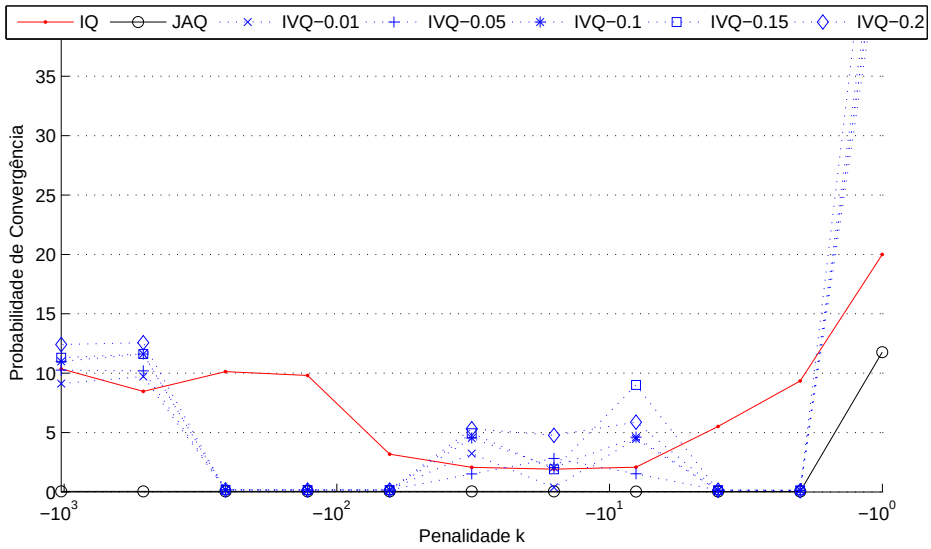


(b) Softmax

Figura A.47: Convergência no *Penalty Game* com 4 agentes, $T = 16(0.9995^t)$ e $so = 2$

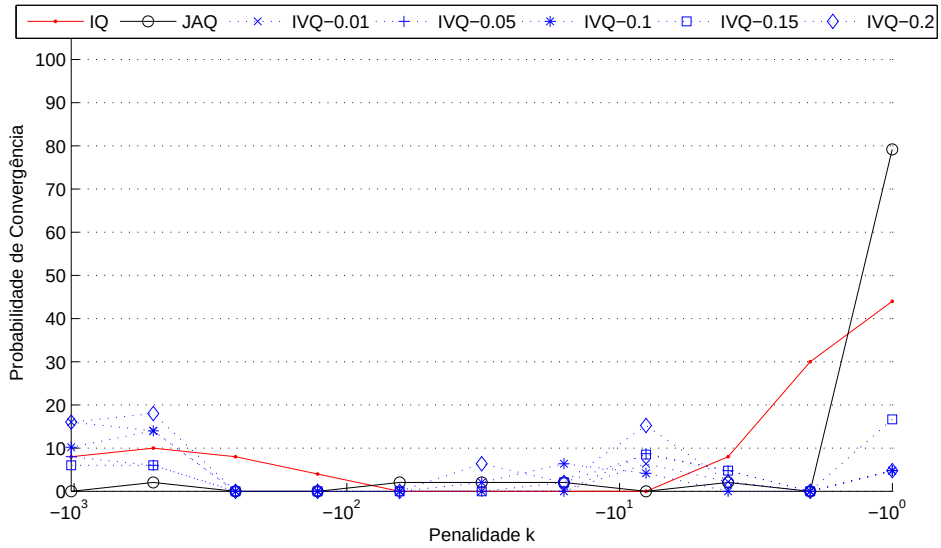


(a) Greedy

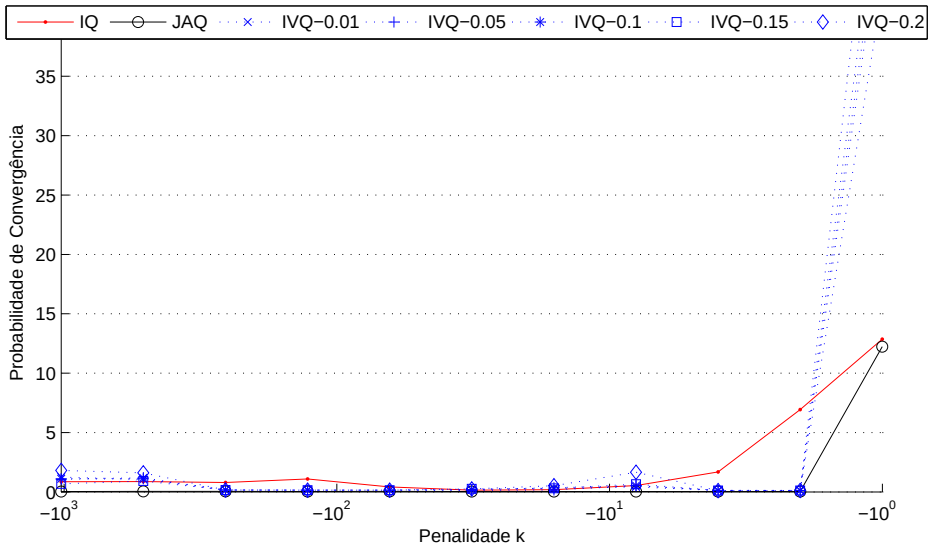


(b) Softmax

Figura A.48: Convergência no *Penalty Game* com 4 agentes, $T = 16(0.9998^t)$ e $so = 2$



(a) Greedy



(b) Softmax

Figura A.49: Convergência no *Penalty Game* com 4 agentes, $T = 16(0.9999^t)$ e $so = 2$

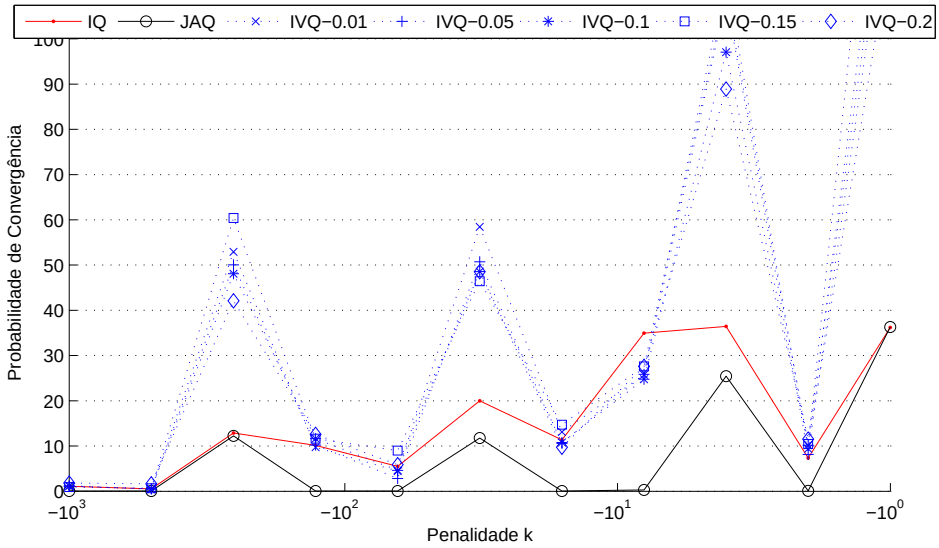
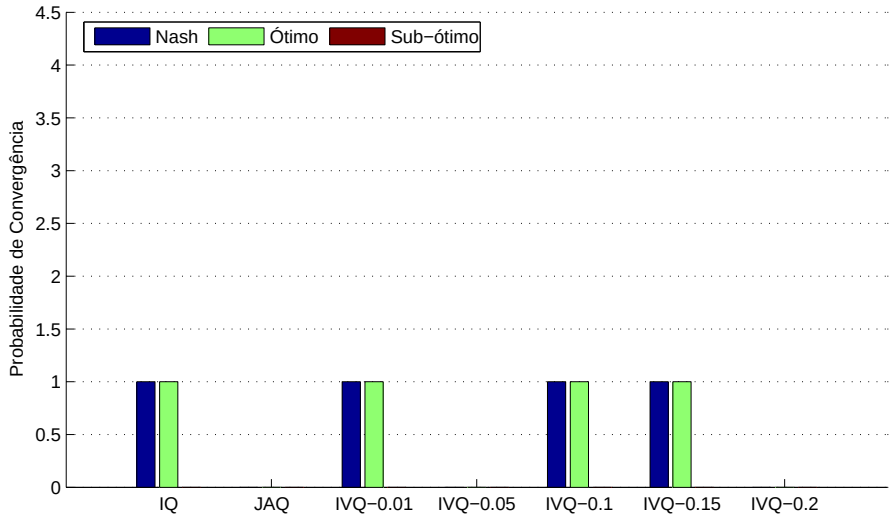
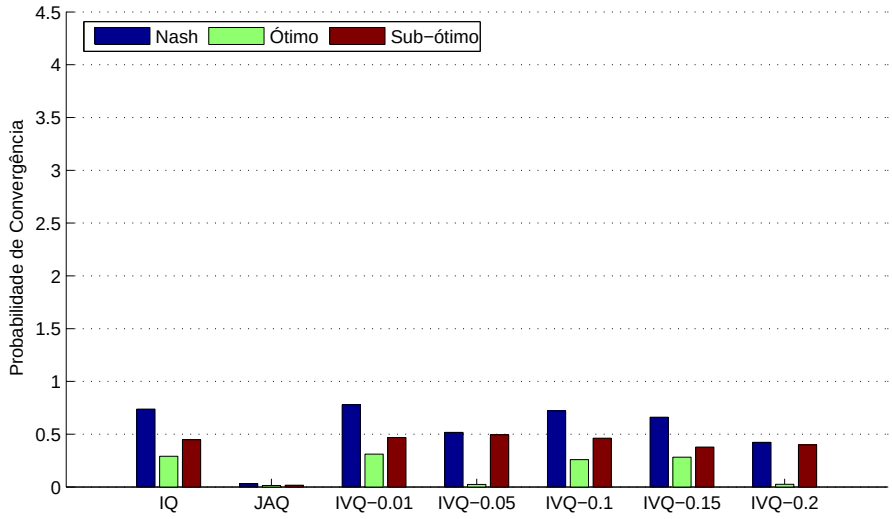


Figura A.50: Melhores algoritmos no *Penalty Game* com 4 agentes para $so = 2$

A.3.2 Climbing Game

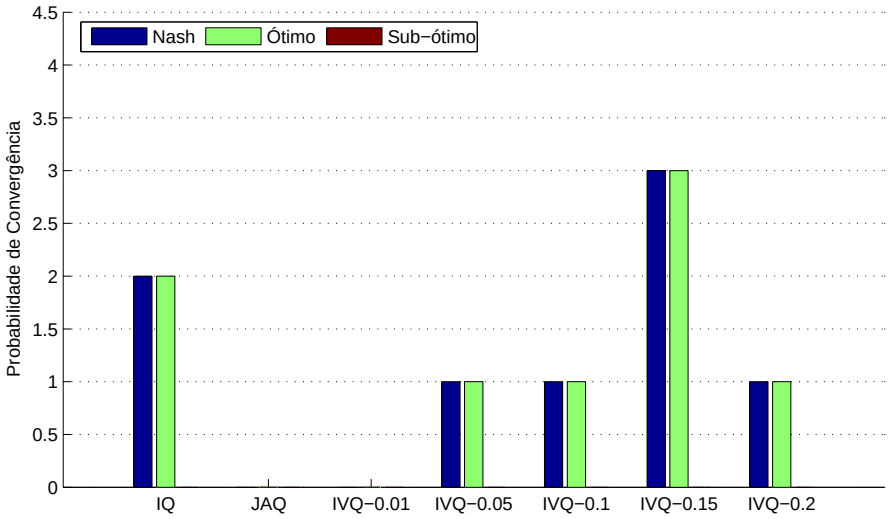


(a) Greedy

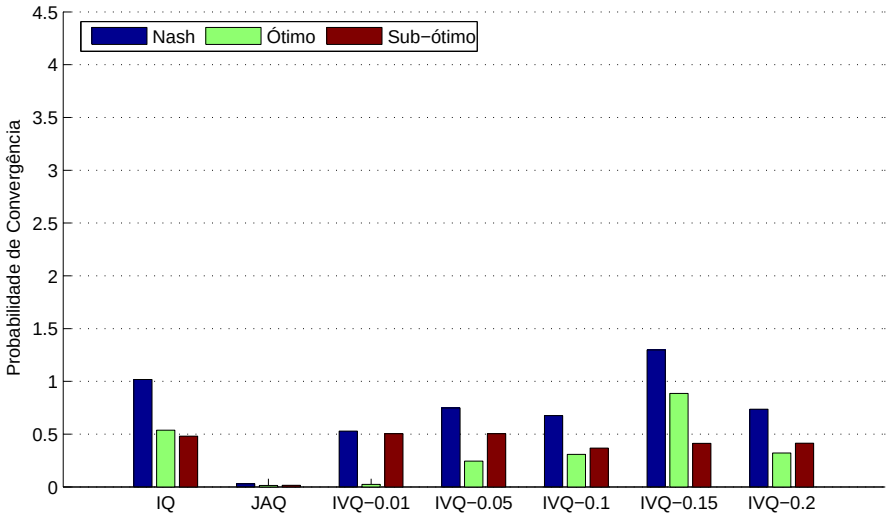


(b) Softmax

Figura A.51: Convergência no *Climbing Game* com 4 agentes, $T = 16(0.999^t)$

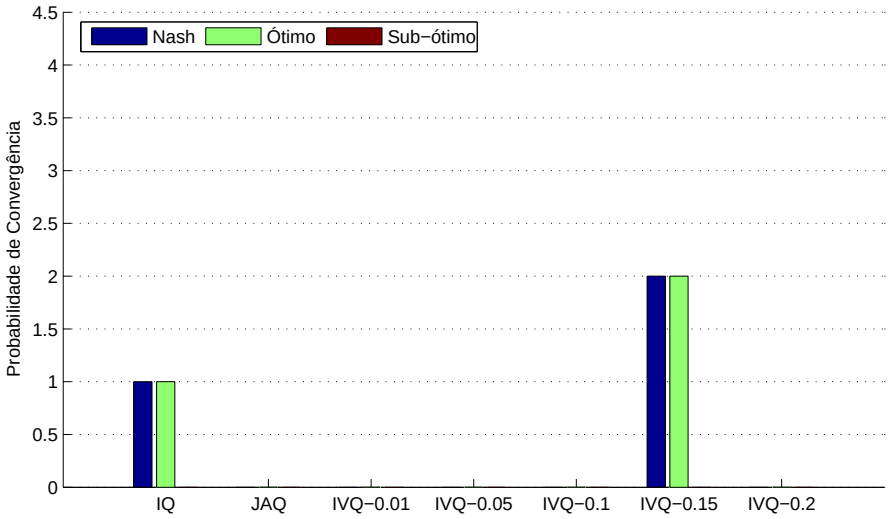


(a) Greedy

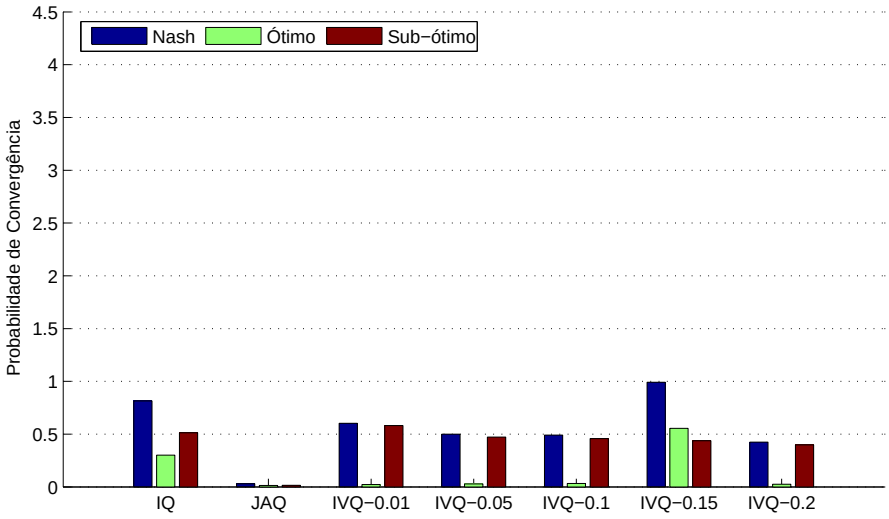


(b) Softmax

Figura A.52: Convergência no *Climbing Game* com 4 agentes, $T = 16(0.9991^t)$

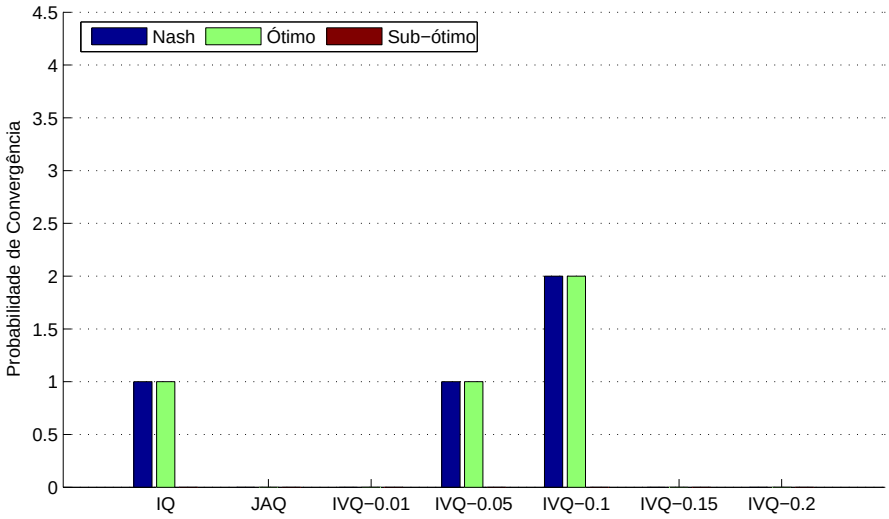


(a) Greedy

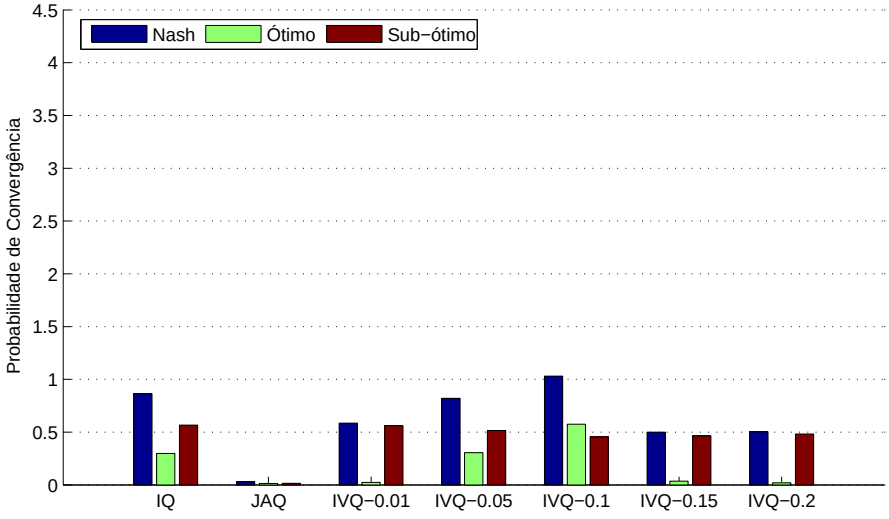


(b) Softmax

Figura A.53: Convergência no *Climbing Game* com 4 agentes, $T = 16(0.9992^t)$

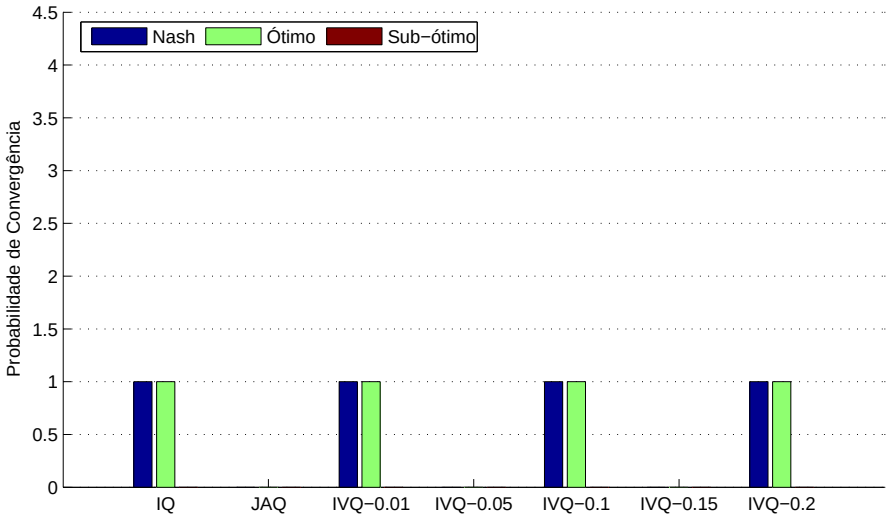


(a) Greedy

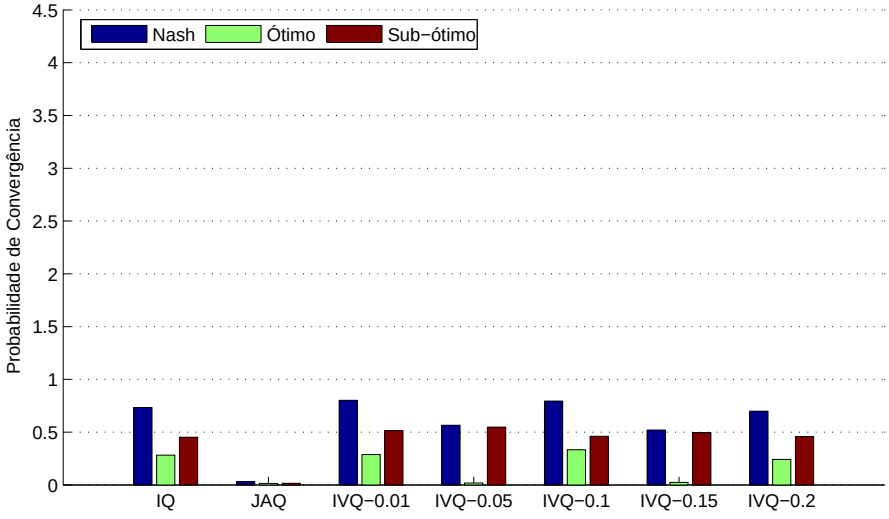


(b) Softmax

Figura A.54: Convergência no *Climbing Game* com 4 agentes, $T = 16(0.9993^t)$

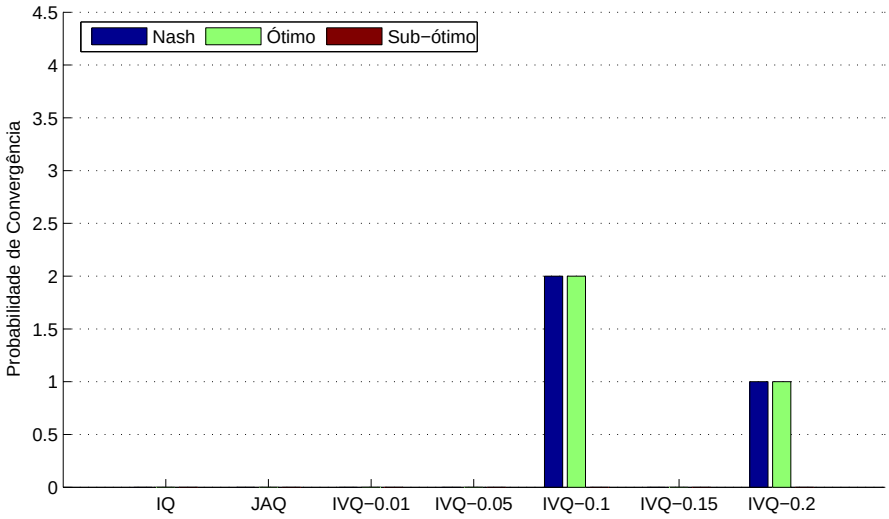


(a) Greedy

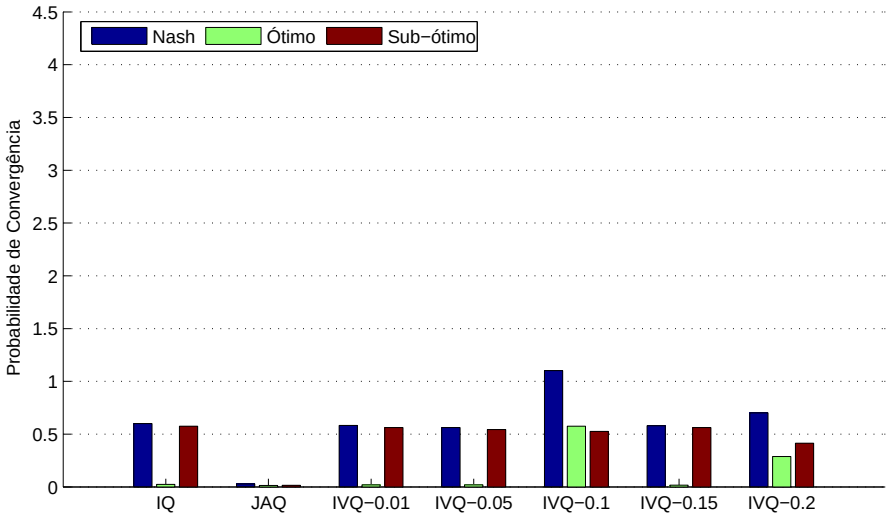


(b) Softmax

Figura A.55: Convergência no *Climbing Game* com 4 agentes, $T = 16(0.9994^t)$

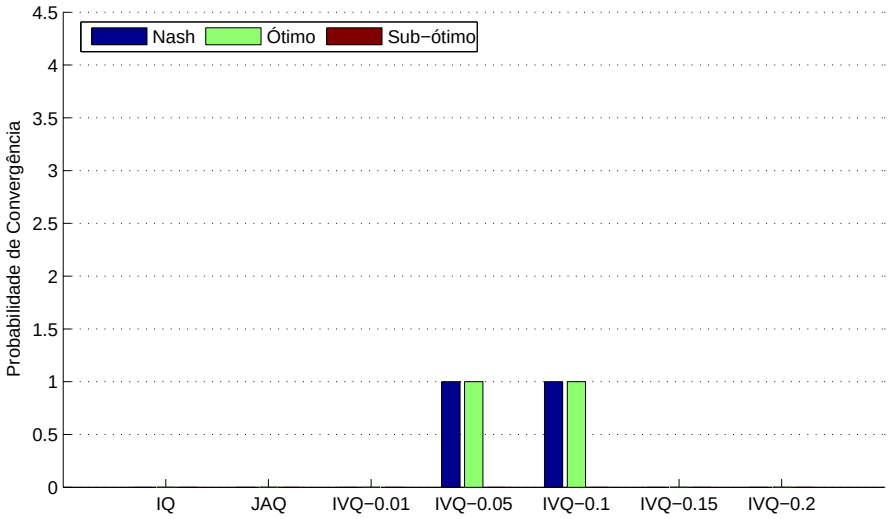


(a) Greedy

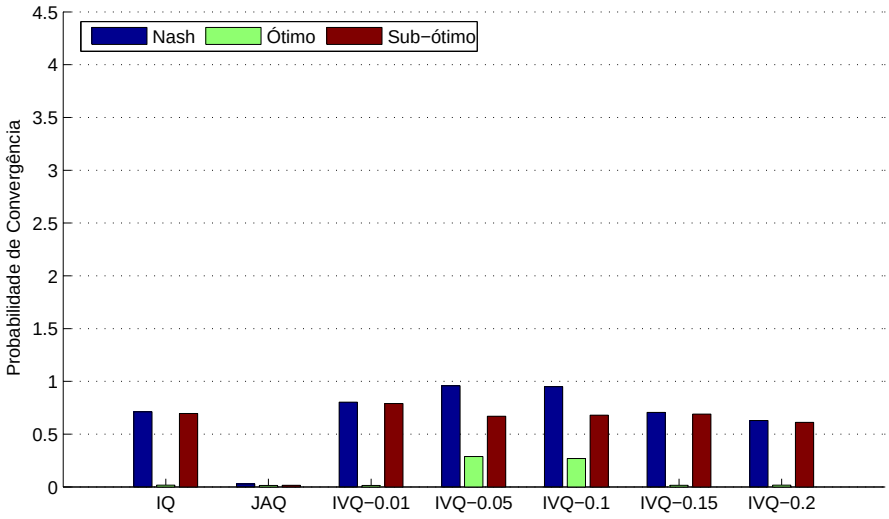


(b) Softmax

Figura A.56: Convergência no *Climbing Game* com 4 agentes, $T = 16(0.9995^t)$

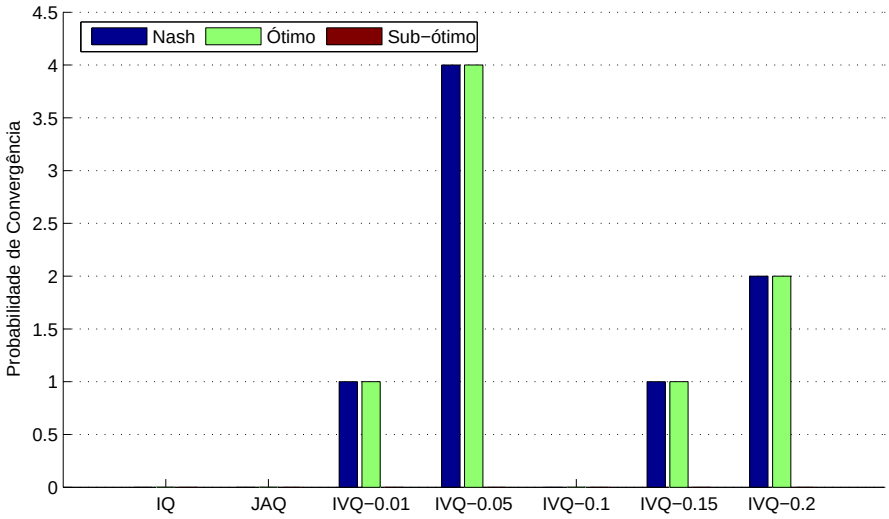


(a) Greedy

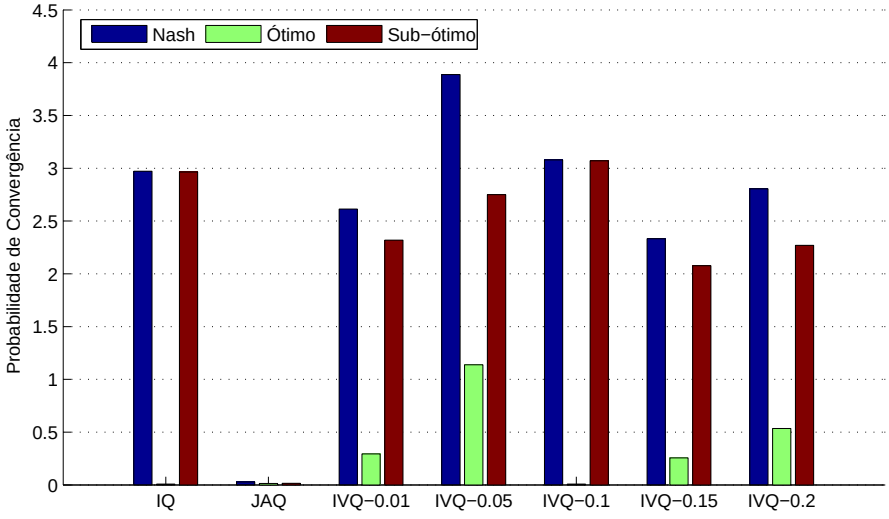


(b) Softmax

Figura A.57: Convergência no *Climbing Game* com 4 agentes, $T = 16(0.9996^t)$

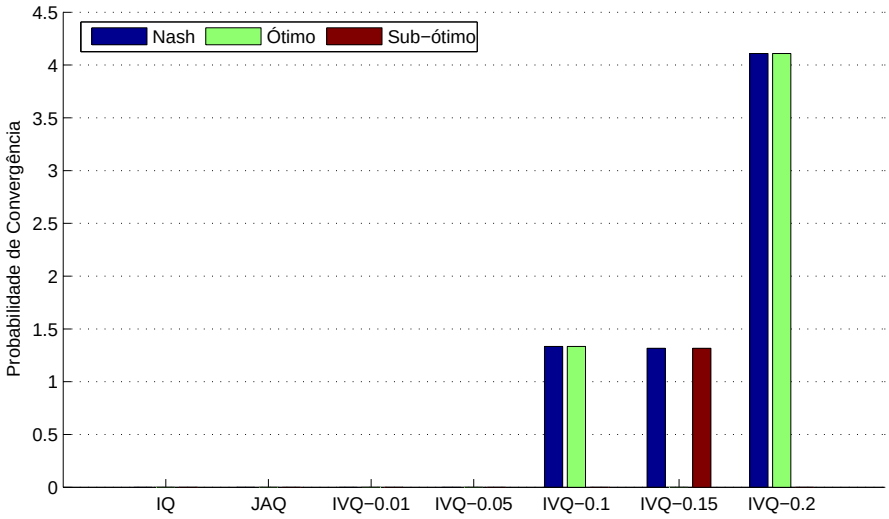


(a) Greedy

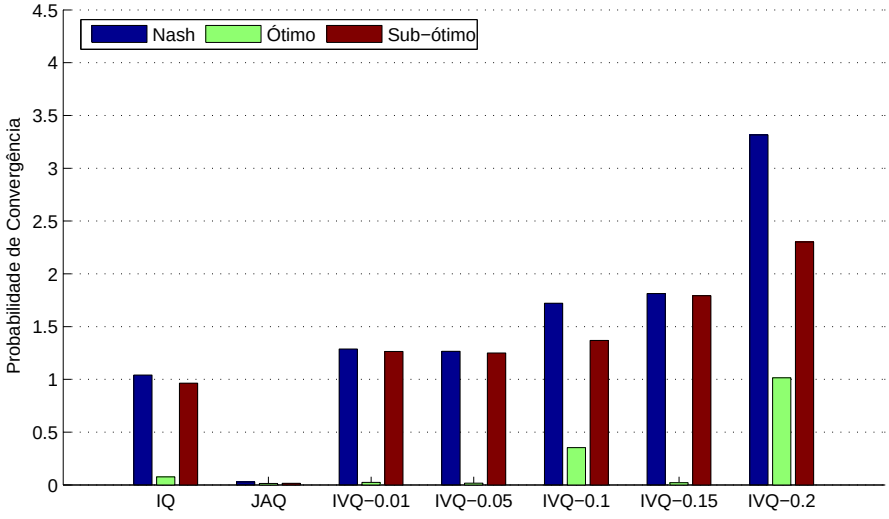


(b) Softmax

Figura A.58: Convergência no *Climbing Game* com 4 agentes, $T = 16(0.9997^t)$

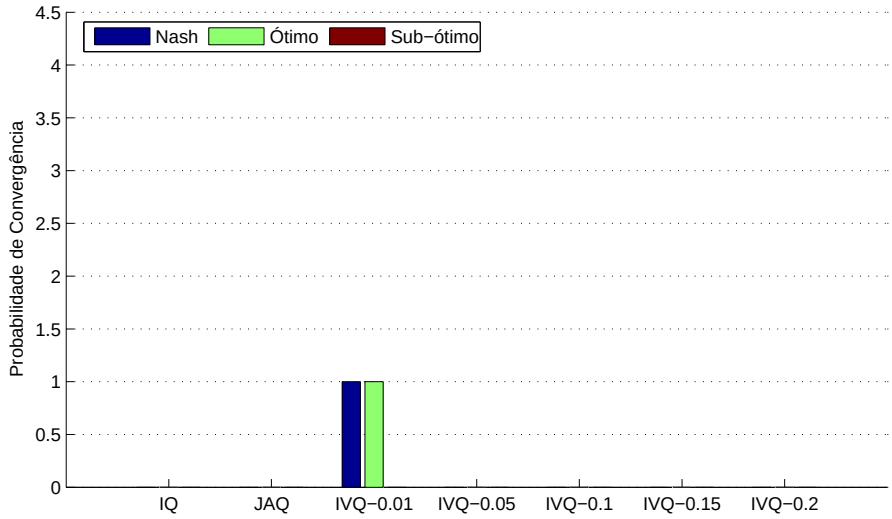


(a) Greedy

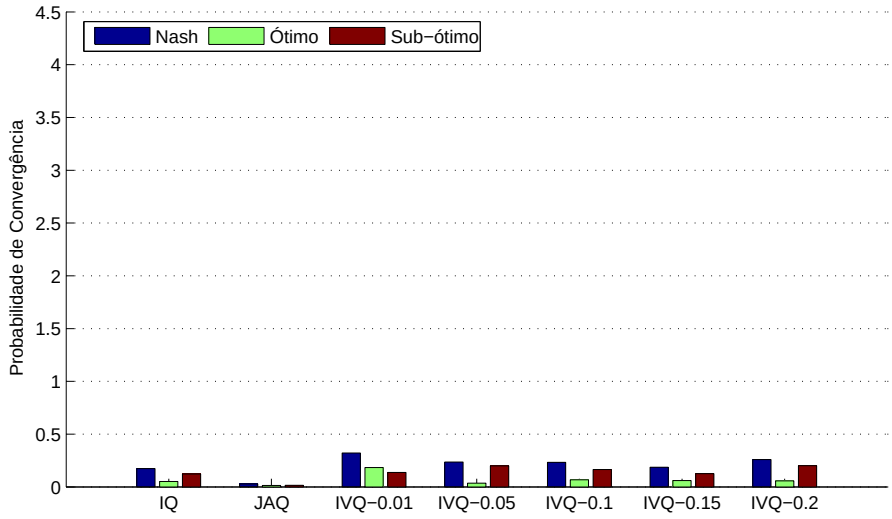


(b) Softmax

Figura A.59: Convergência no *Climbing Game* com 4 agentes, $T = 16(0.9998^t)$



(a) Greedy



(b) Softmax

Figura A.60: Convergência no *Climbing Game* com 4 agentes, $T = 16(0.9999^t)$

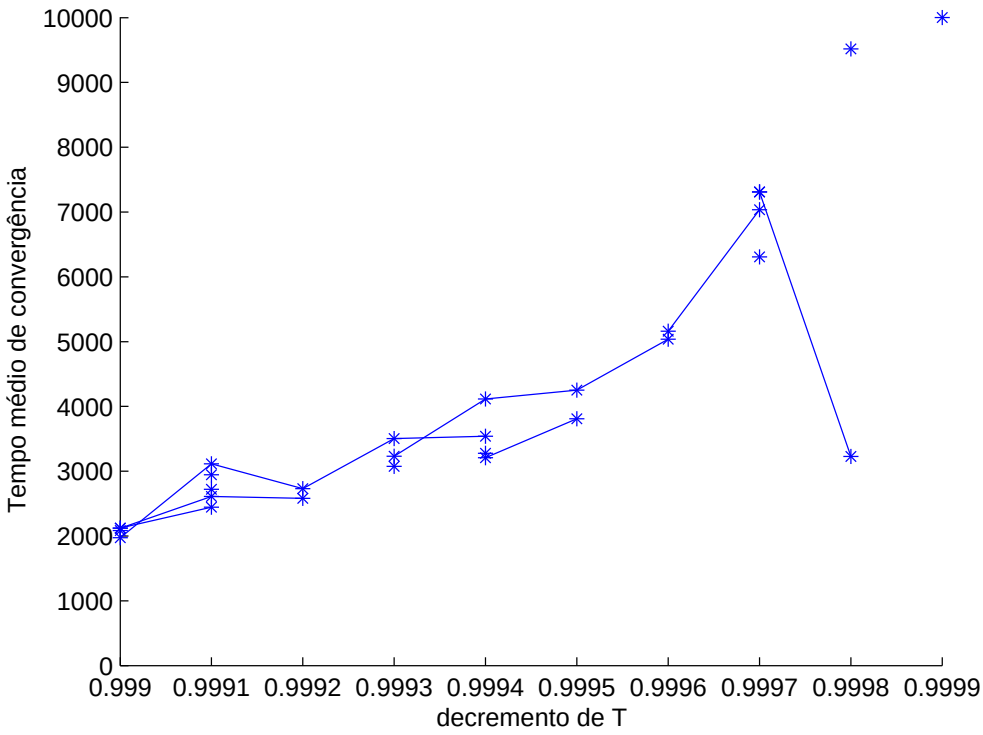
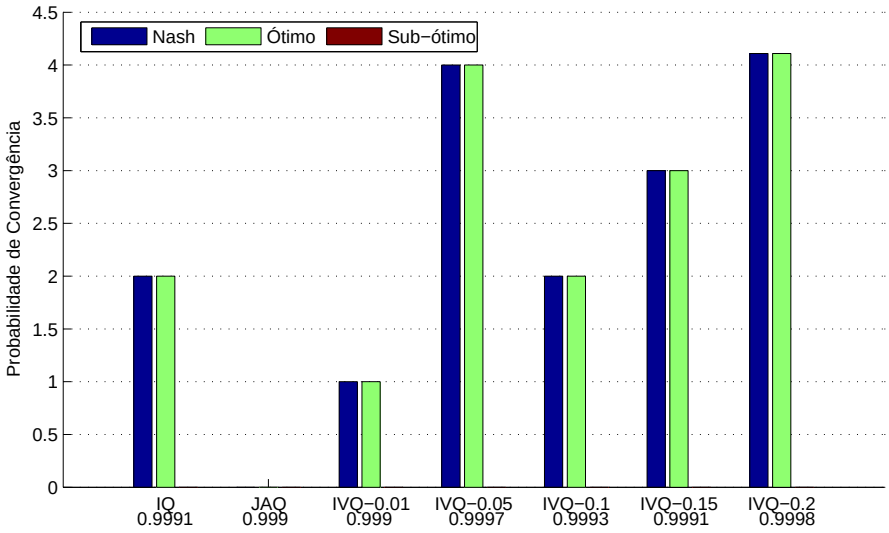
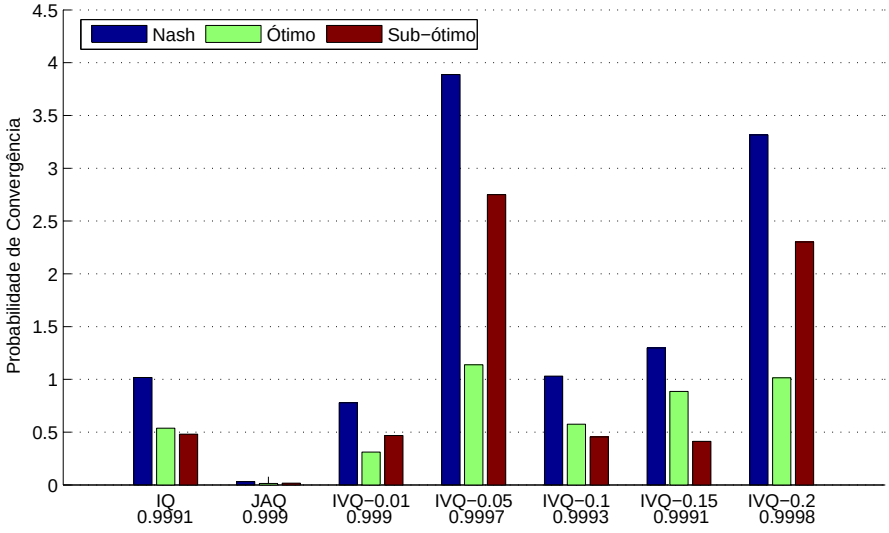


Figura A.61: Tempo de Convergência ao ótimo no *Climbing Game* com 4 agentes

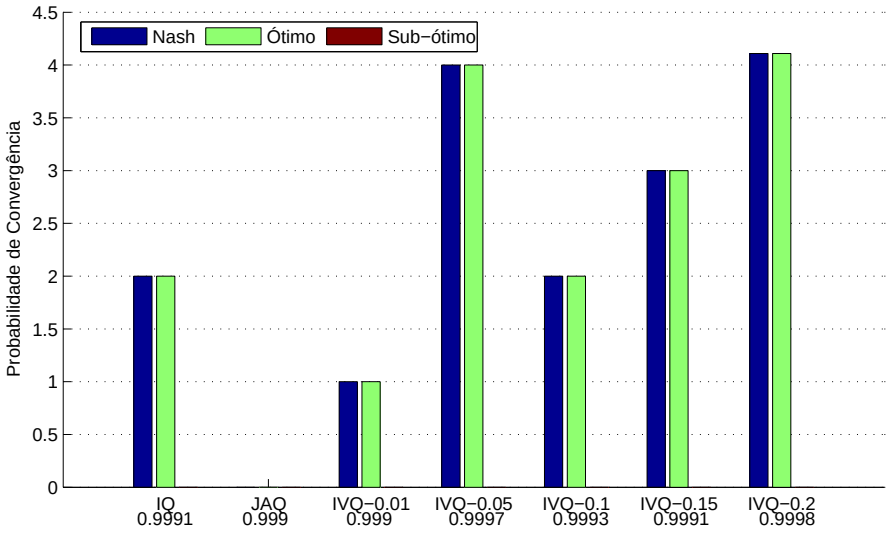


(a) Greedy

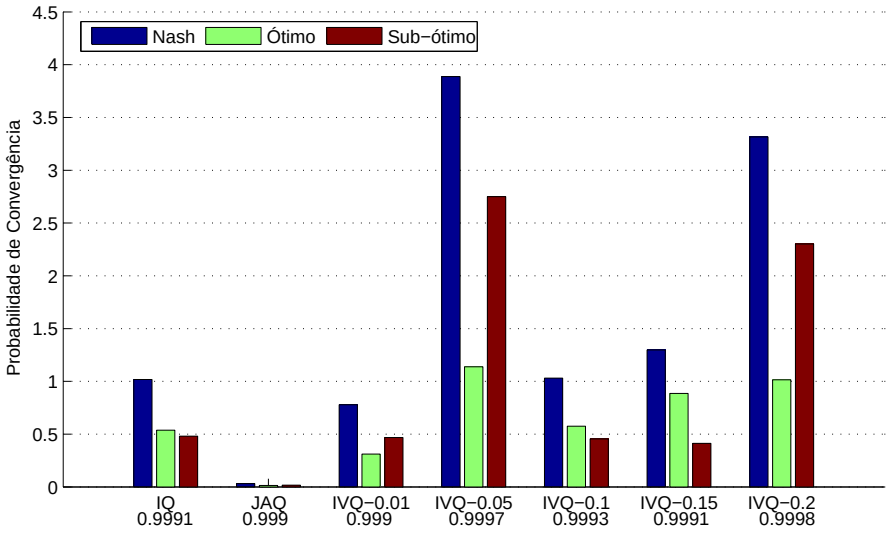


(b) Softmax

Figura A.62: Melhores algoritmos no *Climbing Game* com 4 agentes (equilíbrio ótimo)



(a) Greedy



(b) Softmax

Figura A.63: Melhores algoritmos no *Climbing Game* com 4 agentes (equilíbrio de Nash)

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo