

## **Um ambiente para desenvolvimento de negociadores eletrônicos inteligentes**

**Rômulo Nunes<sup>1</sup>, Alan Silva<sup>2</sup>, Ig Bittencourt<sup>2</sup>,  
Evandro Costa<sup>2</sup>, Camila Nunes<sup>2</sup>**

<sup>1</sup>Universidade Federal de Alagoas (UFAL)

Pólo Penedo - Av. Beira Rio S/N - Centro Histórico, Penedo, AL – Brasil

<sup>2</sup>Instituto de Computação – Universidade Federal de Alagoas (UFAL)

Campus A. C. Simões, BR 104 - Norte, Km 97, C. Universitária, Maceió, AL – Brasil

romulo@dsc.ufcg.edu.br {alan,ibert,evandro,cpbn}@tci.ufal.br

***Abstract.** Automated negotiation process is a hot research topic in AI applied to e-commerce. Particularly, one of the problems concerning the design of negotiation agents is to endow them with learning and adaptation capabilities in order to face the dynamism often required by negotiation processes. This work proposes an environment for building intelligent agents capable of negotiating and learning. Experiments were done in order to demonstrate the feasibility of the proposed environment. The experiments show that the agents learned during the interaction process according to the used techniques: Neural networks, Q-learning, Rule-based Reasoning and Case-based Reasoning.*

***Resumo.** A automatização do processo de negociação é um dos tópicos importantes e desafiadores na IA aplicada ao e-commerce. Particularmente, um dos problemas relacionados à concepção de agentes de software negociadores diz respeito a dotá-los da capacidade de aprender e se adaptar ao dinamismo normalmente requerido no processo de negociação. Neste trabalho, propõe-se um ambiente para construção de agentes inteligentes dotados das capacidades de negociar e aprender. Foram realizados experimentos para demonstrar a factibilidade da nossa proposta. Os experimentos mostraram a capacidade de aprendizagem dos agentes, segundo as técnicas empregadas: Redes Neurais, Q-learning, Raciocínio Baseado em Regras e Raciocínio Baseado em Casos.*

### **1. Introdução**

Muito se tem discutido sobre sistemas multiagentes no comércio eletrônico, onde o objetivo básico é transferir o trabalho do negociador humano à entidades de software capazes de tomar decisões baseadas em visões cognitivas semelhantes às do homem. Mas, existe uma outra característica humana interessante ao comércio eletrônico: a aprendizagem, que no mundo dos negócios é traduzida como experiência do negociador. As técnicas da Inteligência Artificial (IA) para aprendizagem podem ser aplicadas em vários pontos do processo de negociação, o que foi motivação para este trabalho. Desta forma, é proposto um sistema multiagente (SMA) que seja adaptável aos vários cenários de negociação, permitindo que os agentes possam alterar suas estratégias dinamicamente para serem flexíveis e adaptáveis conforme as variações do ambiente. Uma característica interessante

do SMA negociador é que suas ações são importantes para a aprendizagem e independem do modelo de tomada de decisão dos seus oponentes.

Este artigo está organizado da seguinte forma: na Seção 2 serão apresentados alguns trabalhos relacionados. Na Seção 3, é mostrada nossa proposta para aprendizagem de máquina. Na Seção 4, nós consideramos um cenário ilustrativo mostrando a nossa solução proposta. Na Seção 5, nós faremos uma discussão comparativa do trabalho mostrando sua relevância. Finalmente, a conclusão e direcionamentos para trabalhos futuros em potencial são apresentados na Seção 6.

## 2. Trabalhos Relacionados

No contexto do comércio eletrônico, a construção de softwares e ambientes para elaboração de negociadores já tem sido abordada em diversos trabalhos, destacando-se:

- o trabalho realizado por Bartolini et al. [Bartolini, Preist e Jennings 2005] em que enfocam a necessidade do tratamento das propostas de negócio por meio de regras de produção. Na implementação do arcabouço em [Badica et al. 2005] foram utilizados: o Jade (*Java Agent Development Framework*), que proporciona toda infraestrutura básica para agentes de software; e o JESS (*Java Expert System Shell*), utilizado com as regras que atuam nas mensagens de interação entre os negociadores e no controle de seus comportamentos.
- Em [Gimpel et al. 2003], o arcabouço PANDA (*Policy-driven Automated Negotiation Decision-making Approach*) utiliza regras e funções de utilidade para negociação de serviços e recursos web. Não existe a aprendizagem dinâmica e a principal fonte de conhecimento sobre a negociação está nas regras de negociação. Alguns trabalhos semelhantes que utilizam funções de utilidade e similaridade são vistos em [Faratin, Sierra e Jennings 2002, Fatima, Wooldridge e Jennings 2004].
- O trabalho em [Fatima, Wooldridge e Jennings 2005] utiliza técnicas da IA para solucionar problemas como a escolha da estratégia mais adequada, quando enviar ultimatoss, avaliação de utilidade das propostas, e até mesmo a consideração de negociações simultâneas.

Estes últimos trabalhos foram avanços originados a partir da necessidade da aproximação do modelo de negociação praticado na Web ao modelo praticado no mundo real. O trabalho proposto contribui simplificando a implementação das técnicas cognitivas em negociadores eletrônicos, utilizadas para representação da experiência e aprendizagem do negociador.

## 3. Modelo de Aprendizagem

A arquitetura do SMA negociador está representada na Figura 1, onde estão destacadas (com bordas mais grossas) as classes relacionadas à aprendizagem e tomada de decisão. O objetivo aqui é mostrar a estrutura do ambiente para criação de SMA Negociadores inteligentes, que poderão fazer uso de quaisquer técnicas da IA disponíveis.

No modelo proposto, técnicas de IA estão presentes para duas finalidades bem distintas:

**Tomada de decisão:** na escolha da melhor estratégia para cada situação, na elaboração de propostas interessantes, e até mesmo na escolha da melhor combinação de técnicas de

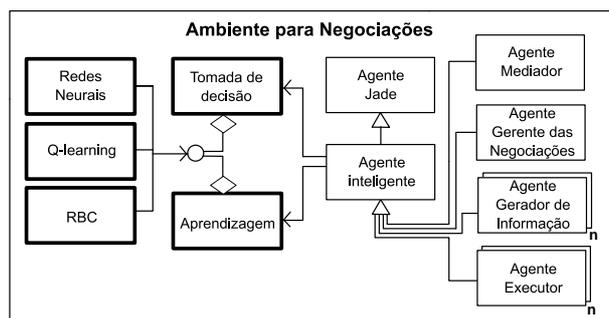


Figura 1. Arquitetura do SMA Negociador

negociação. A tomada de decisão então está presente como uma ferramenta disponível para os agentes da classe *Agente Inteligente*, que poderá se utilizar de qualquer técnica relacionada à tomada de decisão e aprendizagem: RBC, Raciocínio Baseado em Regras (RBR), Rede Neural, Aprendizagem por reforço etc.

Vendo a tomada de decisão como um mecanismo que considera fatos relacionados ao problema e toma uma decisão (ou fornece um indicativo para uma ação), com está ilustrado na Figura 2, pode-se considerar a utilização de qualquer técnica da IA, sendo que algumas necessitam de um treinamento prévio, outras necessitam de uma amostragem inicial, outras de um conjunto de regras de decisão. Esse fomento para o comportamento dessa “função de decisão” deverá ser fornecido à parte, antes da sua utilização.

**Aprendizagem:** é o que assegura a inserção de novas informações à base de dados do agente. Essa aprendizagem pode acontecer de duas formas: com o tempo, à medida que o agente recebe *feedbacks* relacionados as suas ações no ambiente; ou de forma reservada, quando acontece após o acúmulo de muitas informações sobre o comportamento a ser aprendido. A aprendizagem pode ser vista como um processo de ajuste de informações já conhecidas ou como a aquisição de uma nova informação/regra (Figura 3).

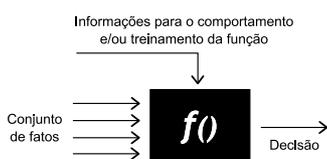


Figura 2. Tomada de decisão vista como uma função

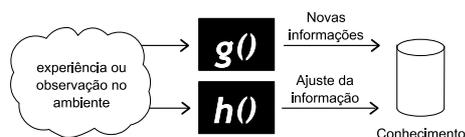


Figura 3. Processo de aprendizagem visto como uma função

Para facilitar o entendimento, a aprendizagem é tratada isoladamente, como se o processo de aprendizagem alimentasse uma base de dados que futuramente serão utilizados na tomada de decisão, mas não acontece sempre assim: o processo de aprendizagem pode também estar diretamente ligado à tomada de decisão, por exemplo com o Q-learning e RBC, onde se faz obrigatório a utilização do módulo de aprendizagem após cada tomada de decisão.

#### 4. Estudo de Caso

Foram realizados experimentos para demonstrar a factibilidade da nossa proposta, dos quais apenas um está descrito neste artigo. Aplicando o ambiente desenvolvido ao

domínio do Turismo, mais especificamente na venda de pacotes turísticos, foram elaborados dois negociadores diferentes: o primeiro, atua sempre como cliente, tendo um modelo baseado em uma função matemática (definida adiante) para a elaboração das propostas de negociação e instanciado com características baseadas em algumas regras de produção; o segundo negociador, atuando como vendedor, deverá implicitamente aprender sobre as regras utilizadas para elaboração das instâncias do primeiro negociador, e para isso utilizará algumas técnicas da IA, descritas ao longo desta seção.

Foi admitido, como simplificação para os experimentos, que cada pacote é constituído por  $P = \{Local, Transporte Ida, Transporte Volta, Estadia, Tempo Estadia, Valor\}$ , em que: *Local* representa o destino da viagem; *Transporte Ida* e *Transporte Volta* representam os meios de transporte para ida e volta, respectivamente; *Estadia* representa a qualidade/tipo do local de acomodação durante a estadia na cidade; *Tempo Estadia*, a duração da viagem no destino; e *Valor* representa o custo financeiro total para o pacote turístico.

#### 4.1. Definição dos Negociadores

Foram definidos dois negociadores para o experimento no ambiente. O primeiro SMA negociador não aprende e trabalha gerando propostas com dados matemáticos e regras definidas a priori. Para o segundo SMA, seus agentes internos utilizam técnicas de aprendizagem de máquina. Maiores detalhes sobre cada um desses agentes são apresentados a seguir:

##### 4.1.1. Negociador 1

No SMA Negociador 1, os agentes colaboradores assumirão as seguintes características:

**Agente Executor:** elabora propostas segundo a metodologia adotada pelo *agente gerenciador da negociação* e baseado no tempo. A função polinomial, utilizada para o cálculo do valor de um atributo  $x$  na elaboração de uma proposta do agente  $a$  para o agente  $b$ , é dada por Faratin [Faratin, Sierra e Jennings 2002] e está representada na Equação 1.

$$x_{a \rightarrow b}^t[j] = \begin{cases} \min_j^a + \alpha_j^a(t) (\max_j^a - \min_j^a) & \text{Se } V_j^a \text{ decresce} \\ \min_j^a + (1 - \alpha_j^a(t)) (\max_j^a - \min_j^a) & \text{Se } V_j^a \text{ cresce} \end{cases} \quad (1)$$

onde  $\alpha_j^a$  é a subfunção (Equação 2) que depende do tempo e sua imagem está no intervalo  $[0, 1]$ .  $V_j^a$  é a variação positiva ou negativa dos valores do atributo, que representa, por exemplo, se o agente está contra ou a favor do preço.

$$\alpha_j^a(t) = \kappa_j^a + (1 - \kappa_j^a) \left( \frac{\min(t, t_{max})}{t_{max}} \right)^{\frac{1}{\beta}} \quad (2)$$

O coeficiente  $\beta$  (onde  $\beta \in \mathbb{R}^+$ ), presente na Equação 2, representa um número infinito de estratégias e será determinado pelo Agente Gerenciador da Negociação.

**Agente Gerenciador da Negociação:** informará sempre a estratégia adequada para o momento na negociação, fornecendo valores para o índice  $\beta$  da Equação 2. Esse índice  $\beta$  pode ser visto como um representante da estratégia do negociador na elaboração das propostas, onde: valores próximos a zero caracterizam um negociador que não cede no início da negociação, mas se torna flexível quando o tempo da negociação está próximo

ao fim; valores de  $\beta$  maiores que 1, caracterizam um negociador que cede mais em suas propostas durante o início da negociação do que no fim [Faratin, Sierra e Jennings 2002].

O índice  $\beta$  é dado segundo algumas regras fixas de produção que dependem do tempo restante para negociação (*TempoRestante*), do tempo médio de uma interação (*TempoMedio*), informações do *agente monitor do ambiente* e se recebeu um *ultimato* ou não. Como as instâncias deste SMA devem se comportar de forma diferente segundo suas características, para cada par *profissão-motivo da viagem* foram elaboradas regras diferentes. As variáveis de decisão e características particulares das instâncias do Negociador 1, que são geradas aleatoriamente, variam conforme a Tabela 1.

Características e Preferências	Possíveis Variações
Profissão	professor, médico, engenheiro, advogado
Motivo da Viagem	férias, visita, trabalho
Limite Disponível (R\$)	1000, 5000, 10000, 20000
Prazo de Pagamento	entre 1 e 12 meses
Dias para Início da Viagem	entre 1 e 120 dias
Transporte Ida	trem, ônibus, avião, navio
Transporte Volta	trem, ônibus, avião, navio
Estadia	pousada, hotel 3 estrelas, hotel 4 estrelas, hotel 5 estrelas

Tabela 1. Opções para os campos da proposta para o Negociador 1

A instanciação dinâmica para este negociador obedece a algumas regras, as quais deverão ser implicitamente aprendidas ou mapeadas pelo Negociador 2 (Objetivo da simulação). Assim, a definição do perfil do negociador e suas preferências na negociação são feitas quando o mesmo é criado, permanecendo fixas durante o ciclo de vida da instância. As regras utilizadas na criação do negociador não foram projetadas para serem mutuamente excludentes, podendo uma mesma instância do negociador obedecer a mais de uma. Essas regras estão descritas a seguir:

1. Se *MotivoViagem* = [trabalho] Então *Estadia* = [hotel3Estrelas, hotel4Estrelas] e *TransporteIda* = [avião, onibus] e *TransporteVolta* = [avião, onibus] e *DiasInicioViagem* = [2–15]
2. Se *MotivoViagem* = [Ferias] Então *Estadia* = [pousada, hotel5Estrelas] e *TransporteIda* = [navio, avião, trem] e *DiasInicioViagem* = [30–90]
3. Se *Profissao* = [advogado, professor] Então *LimiteDisponivel* = [5000] e *TransporteIda* = [aviao, onibus] e *DiasInicioViagem* = [5–30]
4. Se *Profissao* = [medico, engenheiro] e *MotivoViagem* = [visita] Então *LimiteDisponivel* = [5000,10000,20000] e *Estadia* = [pousada, hotel simples] e *TransporteIda* = [avião]
5. Se *Profissao* = [medico, engenheiro] e *MotivoViagem* = [ferias] Então *LimiteDisponivel* = [10000,20000] e *Estadia* = [hotel4Estrelas, hotel5Estrelas] e *TransporteIda* = [navio, avião, trem] e *TransporteVolta* = [navio, avião, trem]

#### 4.1.2. Negociador 2

Seus agentes colaboradores assumirão as seguintes características:

**Agente Gerenciador da Negociação:** Informará sempre a estratégia com maior chance de sucesso, ou mais adequada para o momento na negociação, fornecendo valores para o índice  $\beta$  da Equação 2, assim como o Negociador 1. Porém, aqui será utilizado o algoritmo *Q-learning* para aprender sobre o melhor valor de  $\beta$  associado ao estado atual da negociação (tempo restante, tempo da negociação, informações dos agentes *Monitor do Ambiente* e *Catalogador de Perfis*).

**Agente Catalogador de Perfis:** Agrupa os usuários considerando vários aspectos utilizando Redes Neurais para a aprendizagem, com 16 neurônios na camada de saída e 5 na camada de entrada. A escolha desses números foi arbitrária, uma vez que não é objetivo do experimento a descoberta da melhor configuração possível, interessando apenas a caracterização do “aprender”. Para o treinamento da rede neural do tipo SOM (*Self-Organizing Maps*) foram utilizados os valores de identificação na instanciação do agente, que inclui as características *profissão* e *motivo da viagem*, em conjunto com informações de acordos relacionados em histórico. Após a aprendizagem este agente deve enviar uma informação de classificação para o Agente Gerenciador da Negociação, que a utiliza como variável que define o estado da negociação. Como será visto na subseção seguinte, foi utilizado um conjunto de treinamento com 2000 casos. O critério de parada para o treinamento foi cumprimento de 100 ciclos de treinamento com o mesmo conjunto.

**Agente Executor:** Elaborar propostas segundo a metodologia adotada pelo agente de estratégias, também conforme a função de Faratin [Faratin, Sierra e Jennings 2002], vista na Equação 1. São elaboradas várias propostas através de pequenas variações (entre 1% e 10%) do coeficiente  $\beta$  recebido. Porém, algumas dessas propostas vêm do algoritmo de RBC, que toma como base a experiência em histórico, utilizando algumas características ( $\beta$  & profissão & tempo restante & índice de variação das propostas vindas do negociador) que definem o estado do ambiente como chave de busca. Para avaliar as propostas recebidas, este agente utiliza a função utilidade definida em [Faratin, Sierra e Jennings 2002]. Se a uma nova proposta elaborada tiver utilidade inferior a proposta recebida, esta última é aceita.

A função que calcula a similaridade entre dois atributos  $r_i, s_i$  é dada por:  $sim(r_i, s_i) = 1 - |h_i(r_i) - h_i(s_i)| \in [0, 1]$ , onde  $h_i()$  é a função de similaridade para o  $i$ -ésimo atributo que retorna valores no intervalo  $[0, 1]$ .

## 4.2. Descrição e Resultados do Experimento

Com os negociadores descritos acima, inicialmente foram realizadas 2000 negociações, com instâncias diferentes do Negociador 1 e utilizando o mesmo Negociador 2 (para aquisição de experiência). Em um segundo momento, após o treinamento do negociador 2, mais 500 negociações para avaliação. Em todas as 2500 negociações, o tempo limite para fechamento do negócio foi de 1 min, e ocorreu uma por vez. Os testes foram realizados com 2 computadores, ambos com a seguinte configuração: AMD Athlon 2.08Ghz, 512RAM, sistema operacional Windows XP, em uma rede de barramento 10 Mbps (*MegaBits Per Second*). Os resultados são discutidos em seguida.

A priori, comparando os 2000 primeiros negociadores, criados aleatoriamente, com as 5 regras utilizadas na criação do Negociador 1, foi identificado que: 12% dos negociadores criados enquadraram-se na primeira regra; 9%, na segunda; 22%, na terceira; 7%, na quarta; e, 11%, na última regra (Os outros 39% não se enquadraram em nenhuma regra).

Na Figura 4, o gráfico da média das utilidades no tempo, ilustra também um ganho da diminuição do tempo médio de convergência. Na Figura 5, é mostrada uma aproximação média do número de acordos realizados por interação. A linha pontilhada representa o primeiro momento do experimento, onde a concentração dos acordos fechados está na quinta interação. O segundo momento, representado pela linha tracejada, tem

a maioria dos acordos realizados na terceira interação.

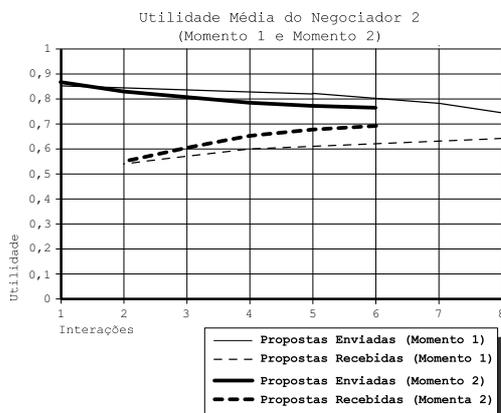


Figura 4. Comparação da convergência entre os cenários

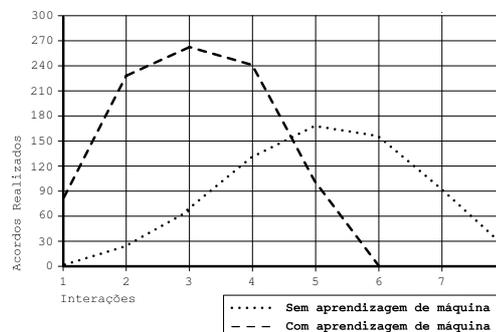


Figura 5. Acordos por ciclo de interação

As diferenças entre as duas partes da experiência trouxe uma aproximação para um negociador inteligente através do Negociador 2. Como o Negociador 1 não possuía um modelo de aprendizagem definido, e sempre eram tomadas novas instâncias dele a cada negociação, não foi observado nenhuma evolução. O objetivo da experiência, que era mostrar o ambiente e a aprendizagem influenciando significativamente o processo da negociação, foi alcançado. Uma pesquisa exaustiva à parte deveria ser feita, tanto para o ajuste das técnicas utilizadas, como para compará-las com outras, mas foge do escopo do experimento. Considerando as alterações proporcionadas, mesmo com um modelo de aprendizagem passível de aprimoramentos, pode-se considerar que as técnicas da IA contribuíram significativamente no experimento e portanto podem atuar também em um cenário real.

## 5. Relevância do Trabalho

Refletindo sobre os modelos mostrados na Seção 2, percebe-se que já existem modelos bem consolidados que tratam sobre o protocolo e modelos de negociação, que dão suporte: a negociações com vários atributos, ultimato, comparação de acordos finalizados (através do estado de compromisso), modelo de decisão por análise estatística. Todas essas contribuições, fruto da evolução de várias pesquisas enfocando pontos específicos da negociação, em sua essência foram incorporadas neste projeto, adicionando-se a questão da aprendizagem de máquina.

Nos modelos baseados em funções determinísticas e estatísticas [Fatima, Wooldridge e Jennings 2004], a manipulação do conhecimento, contido nas informações em histórico, dá-se através de cálculos de probabilidades, estimativas, geração de funções de interpolação etc. Por sua vez, IA acrescenta o dinamismo do comportamento de um vendedor real, seja na tomada de decisão empírica, baseada na experiência, julgamento ou intuição do negociador, como na elaboração de propostas que se adequam ao estado do negociador no ambiente. O modelo proposto neste trabalho acrescenta ao agente negociador características de aprendizagem e tomada de decisão, baseadas nos conceitos e técnicas da IA.

Além disso, o ambiente traz a facilidade na construção de agentes capazes de aprender, diferenciando-se portanto dos trabalhos correlatos. A ferramenta em si também

é uma contribuição pois facilita novas implementações de agentes negociadores assim como seu ambiente de negociação.

## 6. Conclusão e Trabalhos Futuros

Embora existam modelos bem fundamentados, que tratam a negociação como um todo, não foi identificado em nenhum deles a possibilidade da combinação de várias técnicas para aprendizagem de máquina, o que, com o advento deste trabalho, cria a necessidade de experimentos específicos para identificar a melhor técnica para cada característica inteligente no negociador.

Vários trabalhos futuros podem ser apontados para o enriquecimento deste trabalho e da negociação automatizada. Um primeiro trabalho, considerado fundamental, seria um esforço em conjunto com outras disciplinas, como administração e economia, para o desenvolvimento do conhecimento sobre a negociação de produtos. Esse conhecimento poderia ser geral o bastante, de forma a ser utilizado em diversos domínios de negócios, ao mesmo tempo em que também poderia permitir o negociador fazer pequenos ajustes, a fim de se encaixar melhor em suas necessidades.

Como a proposta da plataforma é oferecer um suporte de decisão e aprendizagem, especificamente para os agentes negociadores, isto poderia ser ampliado, de forma a oferecer suporte de decisão para os funcionários de uma empresa. Também, de uma forma mais simples, a ferramenta também poderia prover entidades de apoio a decisão, por exemplo, para prover alertas a uma empresa quando determinadas condições no ambiente fossem encontradas.

## Referências

- BADICA, C. et al. Rule-based framework for automated negotiation: Initial implementation. In: ADI, A.; STOUTENBURG, S.; TABEL, S. (Ed.). *RuleML*. [S.l.]: Springer, 2005. (Lecture Notes in Computer Science, v. 3791), p. 193–198. ISBN 3-540-29922-X.
- BARTOLINI, C.; PREIST, C.; JENNINGS, N. R. A software framework for automated negotiation. In: *Software Engineering for Multi-Agent Systems III: Research Issues and Practical Applications*. [S.l.: s.n.], 2005. p. 213–235.
- FARATIN, P.; SIERRA, C.; JENNINGS, N. R. Using Similarity Criteria to Make Issue Trade-offs in Automated Negotiations. *Artificial Intelligence*, v. 142, n. 2, p. 205–237, 2002.
- FATIMA, S. S.; WOOLDRIDGE, M.; JENNINGS, N. R. An Agenda Based Framework for Multi-issues Negotiation. *Artificial Intelligence*, v. 152, n. 1, p. 1–45, January 2004.
- FATIMA, S. S.; WOOLDRIDGE, M.; JENNINGS, N. R. Bargaining with incomplete information. *Annals of Mathematics and Artificial Intelligence*, v. 44, n. 3, p. 207–232, 2005.
- GIMPEL, H. et al. Panda: Specifying policies for automated negotiations of service contracts. In: ORLOWSKA, M. E. et al. (Ed.). *ICSOC*. [S.l.]: Springer, 2003. (Lecture Notes in Computer Science, v. 2910), p. 287–302. ISBN 3-540-20681-7.