

## IMPLEMENTANDO UM MECANISMO DE NEGOCIAÇÃO INTEGRATIVA: DIFICULDADES E RESULTADOS

**Edmilson Lucena Néri**

Departamento de Polícia Federal  
Espírito Santo – ES

**Márcio Leal de Melo Dahia \***

Centro de Informática  
Universidade Federal de Pernambuco (UFPE)  
Recife – PE  
[mdahia@uol.com.br](mailto:mdahia@uol.com.br)

\* *Corresponding author/autor para quem as correspondências devem ser encaminhadas*

*Recebido em 10/2003; aceito em 08/2004*  
*Received October 2003; accepted August 2004*

### Resumo

Este artigo relata a experiência na formalização, em linguagem de programação, dos processos de decisão de compras e de negociação colaborativa, através de funções multiatributos de valor e pelas idéias de Howard Raiffa, encontradas no seu livro, *The Art & Science of Negotiation*, de 1982. De um caso específico, foi possível apontar algumas dificuldades de implementação, bem como apresentar alguns caminhos para a generalização formal desses processos. Com a popularização de sistemas multiagentes, formalizações do conhecimento sobre decisão e negociação são indispensáveis para uma mais segura, quiçá mais eficiente, delegação dessas tarefas a programas de computador.

**Palavras-chave:** negociação; decisão; compras.

### Abstract

This article illustrates an experiment in formalization, using programming language, of purchasing decision and collaborative negotiation processes, by multiattribute worth functions and Howard Raiffa's ideas exposed on his book *The Art & Science of Negotiation*, published in 1982. From a case of study, it was possible to point out some difficulties in implementation, as well as presenting some solutions and ways to achieve formal generalization of these processes. With the popularization of multiagents systems, formalization of the knowledge involved in decision and negotiation are indispensable for a more secure and, perhaps, a more efficient delegation of these tasks to computer programs.

**Keywords:** negotiation; decision; procurement.

## 1. Introdução

Recentes estudos e práticas comerciais indicam um movimento de transferência da competência de algumas tomadas de decisão para programas de computador, como visto em, por exemplo, Conway & Koehler (2000) ou em Yuan (2002). Negociação pode ser definida como uma forma de tomada de decisão envolvendo duas ou mais partes que não podem tomar decisões independentemente e são requisitadas a fazer concessões para chegarem a um acordo (Kersten *et al.*, 1991). Assim sendo, a negociação também pode ser delegada a programas.

Os programas que são delegados para realizar, autonomamente, tarefas em lugar das pessoas são conhecidos por agentes de *software*. A afirmação anterior está longe de ser unanimidade na comunidade acadêmica, pois há divergências sobre a própria definição de agente de *software* (Turban, 2000; Franklin & Graesser, 1996). Não obstante, aplicações que utilizam agentes de *software* já vêm sendo desenvolvidas. Muitas delas, para o comércio (Pattie *et al.*, 1999; Preist *et al.*, 2001; Chavez & Maes, 1996).

A formalização do conhecimento sobre tais tarefas deve ser, entretanto, anterior à delegação da decisão ou da negociação a programas. Das idéias à implementação das teorias, há várias etapas para serem cumpridas. Em um primeiro momento, implementações da teoria para casos específicos podem ser feitas. Depois do aprendizado, a generalização pode ser mais facilmente percebida. Desta forma, este texto apresenta uma formalização para um caso particular, muito embora já sugira os caminhos para formalizações genéricas de decisão e negociação.

Raiffa (1982) propôs, em palavras, um procedimento para negociação integrativa ou colaborativa. Para que o mesmo seja factível, os negociadores têm que estar raciocinando, mesmo que inconscientemente, através de funções de utilidade ou de valor. Outra exigência é que tais funções tenham, no mínimo, dois atributos de decisão.

Este artigo relata a experiência da formalização de mecanismos de decisão e negociação para compras de suprimentos de tecnologia da informação. A entidade compradora, inicialmente, fará uma avaliação da melhor opção de compra, através de uma função multiatributo de valor. Os atributos ou variáveis-chave da decisão são: preço, prazo de pagamento, quantidade e qualidade. Após escolher a melhor opção de compra – a que tiver um maior retorno da função – o comprador iniciará um processo de negociação integrativa com o fornecedor vencedor da concorrência.

Este trabalho é parte integrante de um projeto de um sistema multiagente para compras (Néri, 2002; Néri & Hoppen, 2003). A discussão aqui presente, todavia, independe da associação aos demais temas do projeto. O importante aqui é a percepção das oportunidades advindas da automação do processo decisório e de negociação, bem como discutir os problemas relacionados à implementação de alguns conceitos clássicos desses processos. A experiência aqui relatada é um passo para o surgimento de formalizações generalizadas para as idéias de Raiffa, conseqüentemente, para a criação de uma poderosa ferramenta analítica de negociação.

Na próxima seção, “Ordenando as preferências”, serão apresentados conceitos pertinentes à construção do mecanismo de decisão. Também serão úteis para o entendimento da seção “Negociação integrativa proposta por Raiffa”, onde será detalhada a teoria por trás do processo de negociação implementado. A seção 4 traz os detalhes de implementação dos mecanismos de decisão e negociação, ressaltando as adversidades encontradas para adaptar a teoria, genérica, a um caso específico. A metodologia seguida nos testes e as discussões sobre os mesmos são a essência da seção 5, “Os testes”. As “Considerações finais” formam a seção 6, onde se discutem os resultados alcançados, as limitações e as sugestões para futuros trabalhos.

## 2. Ordenando as Preferências

A primeira etapa para um mecanismo automatizado de negociação é a criação de um procedimento para ordenar as possíveis opções de decisão. Para que a ferramenta de negociação apresentada na seção seguinte seja viável, esse procedimento usará funções de utilidade ou de valor. Aqui, cabe uma diferenciação entre os dois conceitos.

Ambos especificam uma estrutura única de preferência. A função de valor, entretanto, é um caso particular de função de utilidade. A particularidade é que a primeira (valor) está inserida em um ambiente de certeza, enquanto a segunda (utilidade) também pode tratar problemas que envolvam loterias. Em Keeney & Raiffa (1976, p.220) há uma breve discussão a respeito da nomenclatura utilizada para diferenciar as duas. Assim, tudo que aqui for explicado em relação às funções de utilidade, implicitamente, vale para as funções de valor. Adianta-se que na implementação feita, não há um ambiente de incerteza. Logo, são todas funções de valor.

Uma decisão pode ser ponderada por apenas um atributo. Contudo, situações onde se tem apenas um atributo não são comuns. O mais fácil de se encontrar são problemas que exigem a avaliação de mais de um atributo. Por exemplo, as variáveis-chave de decisão de compra podem formar, cada uma delas, um atributo, quais sejam: qualidade, quantidade correta, tempo de entrega, fonte de suprimento e preço (Baily *et al.*, 2000). Para estabelecer uma ordem de preferência dentre possíveis muitas alternativas, ter-se-ia que utilizar funções de utilidade multiatributos (caso deseje-se utilizar funções de utilidade como ferramenta de decisão).

Seja  $A$  uma alternativa constituída por um conjunto de atributos  $\{X_1, X_2, \dots, X_n\}$ , onde  $n$  é o número de atributos. Uma função para representar a utilidade  $u$  de tal alternativa teria a forma:

$$u(x_1, x_2, \dots, x_n) = f[u_1(x_1), u_2(x_2), \dots, u_n(x_n)]$$

Onde  $x_i$  é uma quantidade específica de  $X_i$ ,  $f$  é uma função escalar e  $u_i$  é uma função de utilidade sobre  $X_i$ .

Dependendo da natureza do problema, envolvendo incerteza ou não, e das relações de dependência entre os atributos, a função de utilidade assumirá algumas diferentes formas.

### 2.1 Dominância

Uma alternativa  $A_1$  domina uma alternativa  $A_2$  se todos os níveis de seus atributos são maiores ou iguais aos níveis dos atributos de  $A_2$ , com, ao menos, um deles estritamente maior.

### 2.2 Curvas de indiferenças

Podem ser pensadas como um conjunto de alternativas (algumas, talvez, sendo hipotéticas) entre as quais o tomador de decisão não tem preferência (Clemen, 1996).

### 2.3 Fronteira eficiente

É o subconjunto composto pelas alternativas que não são dominadas por quaisquer outras do conjunto de alternativas possíveis. Se uma alternativa está contida na fronteira eficiente, não há outra alternativa melhor para o problema, há, no máximo, uma ou mais alternativas indiferentes em preferência.

### 3. Negociação Integrativa Proposta por Raiffa

Raiffa (1982, p.131) define “*integrative bargain*” como sendo a barganha (negociação) em que duas partes e várias questões estão envolvidas.

Negociação integrativa – também chamada de negociação colaborativa – vem sendo descrita como o processo de tomada de decisão para resolver um conflito envolvendo duas ou mais partes sobre objetivos múltiplos, mas não mutuamente exclusivos (Lewicki *et al.*, 1997). Ou ainda, na teoria dos jogos, é descrita como um jogo de soma não-zero, onde os valores ao longo das múltiplas dimensões podem ser deslocados em diferentes direções, possibilitando que todas as partes alcancem melhores resultados (Rosenschein & Zlotkin, 1994).

Para cada questão envolvida na negociação, que será encarada como um atributo de decisão, existem alguns níveis que o negociador deve estipular. Um deles é o nível de reserva, que é aquele mínimo, do qual não se pode baixar, ou máximo, do qual não se pode elevar – no caso do valor de um produto em negociação. Em uma negociação de compra e venda, o vendedor deve estabelecer um valor mínimo pelo qual ainda aceita vender. Já o comprador tem o seu preço de reserva, ou seja, o valor máximo que está disposto a pagar por aquele determinado produto em negociação. Esses níveis dos atributos são valores restritivos.

Outros níveis importantes dos atributos são os níveis desejados. Ainda no exemplo de compras, determinado comprador pode desejar receber o produto em negociação no dia seguinte, como ideal, mesmo tendo um prazo de reserva de sete dias, por exemplo. Já o vendedor pode desejar entregar em quinze dias, mas tem como reserva cinco. Em menos que isto, não tem como entregar.

Continuando o exemplo do prazo, é interessante notar que há como se chegar a um acordo, pois há o que se chama de *zona de acordo* entre o quinto e o sétimo dia – sendo um prazo viável para ambos: comprador e vendedor.

A existência de vários atributos de decisão – como no caso de decisão de compras – favorece o alcance de um acordo entre duas partes envolvidas, já que fica mais fácil encontrar uma forma de ceder em algum atributo em troca de benefício em um outro.

Com o uso de funções multiatributos de utilidade ou de valor, pode-se facilmente manipular o impacto de ceder em um atributo e ganhar em outro. A negociação pode fluir com maior rapidez e eficiência

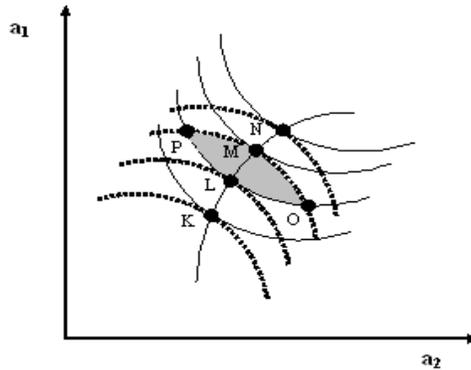
Após os negociadores terem chegado a um acordo, podem tentar melhorar seus desempenhos juntos. Não se trata mais de um jogo de soma-zero, mas algo como “aumentar o bolo para depois repartir”.

No gráfico da Figura 1, estão traçadas as curvas de indiferença de dois negociadores. As curvas com os traços cheios são as do *negociador 1*, já as tracejadas são as do *negociador 2*. Para o *negociador 1*, as curvas mais a nordeste são melhores, enquanto para o *negociador 2*, ao sudoeste.

Suponha-se o ponto *O* como representando o acordo em que se chegou. Note-se que para o *negociador 1*, qualquer ponto na curva *OLP* é indiferente em preferência ao ponto *O*. De forma análoga, para o *negociador 2*, os pontos da curva *OMP* são indiferentes.

A área hachurada no gráfico indica todos os pontos possíveis em que o acordo pode ser melhorado, ou tornar-se indiferente. A curva *KLMN* pode ser entendida como a fronteira

eficiente da negociação, que nesta ambiência é rotulada de *curva de contrato*. Nesta curva, não há como melhorar a negociação de forma colaborativa (“repartir o bolo para repartir”), volta-se a um processo de soma-zero.



**Figura 1** – Curvas de Indiferença e Curva de Contrato

O desafio, portanto, é depois de se ter chegado a um acordo *O* qualquer, melhorá-lo, idealmente, para algum ponto da curva de contrato. Uma forma possível é através de um processo iterativo e iterativo de *trade-offs* e concessões. Por exemplo, o *negociador 1* pode questionar o *negociador 2* se ele está interessado em ganhar um pouco do atributo  $a_1$  em troca de ceder um pouco do atributo  $a_2$ , que estão representados na Figura 1, no eixo das ordenadas e abscissas, respectivamente. Em outras palavras, cada negociador pode oferecer acordos indiferentes em preferência, de tal sorte que o seu parceiro tenha oportunidade de melhorar o acordo. De forma semelhante, após a escolha de um dos pontos, os papéis são invertidos. Isso deve ser feito até, idealmente, se atingir a curva de contrato.

#### 4. Os Mecanismos de Decisão e de Negociação

##### 4.1 O tomador de decisão e os negociadores

A adaptação das teorias acima expostas será feita para a decisão e negociação de compras, onde um comprador escolherá a melhor dentre as propostas de venda enviadas pelos diversos fornecedores, através de uma função de valor multiatributo. Os fornecedores, por sua vez, não tomarão nenhuma decisão em relação a qual o melhor comprador para quem poderia vender. Admite-se que se o vendedor enviou uma proposta de venda de um determinado produto, o mesmo honrará sua proposta.

Essa forma de compra é bem comum nas empresas. A implementação e testes dos mecanismos foram pensados para o caso de uma empresa real, no caso de compras e suprimentos de tecnologia da informação. As variáveis-chave de decisão de compra também foram definidas pelos responsáveis por tais compras. São elas: preço, prazo, quantidade e qualidade. Essa última será encarada de uma forma diferente, estática. Para cada produto de cada marca, será atribuído um valor numérico que representa a mensuração da qualidade do mesmo. Essa atribuição deve ser feita pela equipe de compras da empresa (Néri, 2002).

Após o comprador ter decidido qual a melhor proposta, o fornecedor da mesma irá ser convidado para negociar um acordo melhor. Para que isso seja possível, é necessário que o fornecedor também tenha sua função de valor. No caso do fornecedor, a função terá três atributos: preço, prazo e quantidade. A prerrogativa da negociação é que ela será do tipo integrativa, ou seja, nenhum dos dois negociadores perderá em negociar. O acordo já foi feito, mas tentar-se-á melhorá-lo. Em outras palavras, não pode haver diminuição do retorno das funções de valor dos negociadores. Só aumento.

Para uma correta leitura das próximas seções, ficam estabelecidas as nomenclaturas Compradores e Fornecedores para, respectivamente, as classes das entidades que representam os compradores e os fornecedores. Por sua vez, as nomenclaturas Comprador e Fornecedor, no singular, representarão uma entidade qualquer das classes Compradores e Fornecedores, respectivamente. Bom lembrar que um Comprador ou um Fornecedor neste artigo significa um programa que compra ou fornece de forma autônoma, nessa ordem. Não há intervenção humana após a configuração desses *softwares*.

#### 4.2 Construindo as funções multiatributos: soluções viáveis

A decisão de compra obedecerá a dois tipos de critérios: eliminatórios (ou restritivos) e classificatórios (ou de ordenação). O Fornecedor que obedecer a todos os critérios restritivos e obtiver a melhor classificação será escolhido para fornecer o pedido de compra.

Quando se cria uma entidade Comprador, definem-se as faixas de valores que as variáveis de decisão (preço, prazo de entrega e quantidade) podem assumir, bem como, se há restrições sobre as marcas aceitáveis. Assim, por exemplo, determina-se que o valor para a variável prazo pode ser qualquer um compreendido entre 1 e 15 dias. Qualquer cotação que apresentar prazo distinto será eliminada. De forma semelhante, definem-se limites mínimos e máximos para as outras variáveis. Até a variável preço assume um valor mínimo, que é acreditado como sendo o menor valor possível de ser praticado, sem que se suspeite da integridade do Fornecedor.

Obedecidos todos os critérios restritivos, as cotações serão avaliadas através da soma ponderada de todas as funções de valor associadas a cada uma das variáveis de decisão (preço, prazo de entrega, quantidade e qualidade). A cotação que maximizar a função abaixo, que será chamada de Função Valor Total, será escolhida. Em caso de empate, ganha a cotação mais antiga.

$$\text{FuncaoValorTotal} = \sum_{i=1}^4 \mu_i F_i(x_i)$$

onde:

$i \in \{\text{preço, prazo de pagamento, quantidade, qualidade}\}$

$\mu_i \rightarrow$  peso relacionado à  $i$

$$\sum_{i=1}^4 \mu_i = 1$$

$F_i \rightarrow$  tipo de função relacionado a  $i$  (linear, exponencial, inversa, etc.)

$x_i \rightarrow$  valor numérico relacionado a  $i$

O conjunto dos pesos será informado na configuração do *software* Comprador. É uma etapa de fundamental importância e é realizada através do julgamento do criador do programa.

Para cada variável de decisão de compra, sua função de valor retornará valores entre 0 (pior) e 1 (melhor). A escolha desse intervalo é arbitrária, podendo ser qualquer outro intervalo contínuo (com valores pertencentes ao conjunto dos números reais), pois a estrutura de preferência poderia ser reescalada. Uma outra observação refere-se à forma da função valor total. No exemplo deste artigo, os atributos são independentes entre si. Se isto não fosse verdade, a função valor total assumiria outra forma, onde as interdependências seriam retratadas.

Keeney & Raiffa (1976) sugerem um método interativo e iterativo para a construção de uma função de utilidade, bem como uma função de valor. O tomador de decisão deve ser entrevistado para evidenciar alguns pontos da sua função. Inicialmente, procura-se descobrir os valores que devem retornar 0 e 1 (pior e melhor, respectivamente). Depois, questiona-se sobre o valor  $V$ , que deixaria o tomador de decisão igualmente satisfeito em sair do pior valor para  $V$  ou, de  $V$  para o melhor, ou seja, o valor que retorna 0,5. Repete-se o passo anterior para os dois novos intervalos, 0 a 0,5 e 0,5 a 1, e, assim, sucessivamente, até o entrevistador (quem estrutura a decisão) achar que tem pontos suficientes para descrever a função de preferência do tomador de decisão.

Poder-se-ia dotar o *software* com a tarefa de entrevistar o pessoal responsável pelas compras e pelas vendas (fornecedores) para montar as funções, de forma semelhante ao exposto acima. Acredita-se, contudo, que esse processo é lento e pode frustrar o usuário do sistema.

Uma solução alternativa é predefinir alguns comportamentos. Ao invés de buscar pontos intermediários, pede-se para o tomador de decisão escolher, dentre os comportamentos predefinidos, qual o mais próximo da sua “real” função de preferências. Como forma de parametrização, o tomador de decisão pode definir o grau de concavidade da sua curva.

Abordagens semelhantes são facilmente encontradas na literatura, por exemplo, para definir quão rápido um agente negociador cede (Chavez & Maes, 1996; Kang & Han, 2002). Clemen (1996, p.477) mostra, genericamente, como modelar atitudes de riscos (funções de utilidade), através de funções matemáticas (em especial, exponenciais e logarítmicas).

Na modelagem descrita neste trabalho, decidiu-se oferecer três tipos de comportamentos para descrever as funções dos tomadores de decisão: linear, exponencial e inversa. Questionam-se, para o tomador de decisão, os limites das funções (ou, em outras palavras, os valores pior e melhor), os tipos das mesmas e, para as exponenciais e inversas, os graus de concavidade, que representam a rapidez com que as funções associadas crescem ou decrescem, também conhecidos como parâmetros de forma.

A escolha pela função linear foi pela possibilidade do tomador de decisão indicar que a taxa crescimento (ou decrescimento) é constante. As exponenciais e inversas foram escolhidas, arbitrariamente, como duas funções que apresentam taxas de crescimento variáveis, tendo as primeiras, retornos sempre menores ou iguais a linear e, as do segundo tipo, sempre maiores ou iguais, para os graus de concavidades admitidos (pertencentes ao conjunto dos números naturais).

É importante salientar que as funções foram escolhidas para simular o uso dos mecanismos de decisão e de negociação. Não houve preocupações com reais representações das funções de valor de qualquer empresa.

Com os limites, têm-se dois pontos:  $(x_1, 0)$  e  $(x_2, 1)$ , onde,  $x_1$  é o pior valor esperado e  $x_2$  é o melhor. Para determinar uma função linear, não se necessita de mais nada. A função tem a seguinte forma, onde  $x$  é o valor da variável que se deseja conhecer seu retorno:

$$Linear(x) = \frac{1}{x_1 - x_2} (x_1 - x)$$

Ao contrário da função linear, funções exponenciais e inversas não são definidas com apenas dois pontos, pois várias delas podem passar pelos mesmos. Para a definição da função exponencial, os passos foram: expressar uma função exponencial “genérica”, substituir os dois pontos na expressão e resolver o sistema. Chamando o pior valor de  $x_1$ , o melhor de  $x_2$  e o grau de concavidade de  $\alpha$ , tem-se a seguinte função exponencial:

$$Exponencial(x) = Ce^{Kx^\alpha} - 1$$

onde:

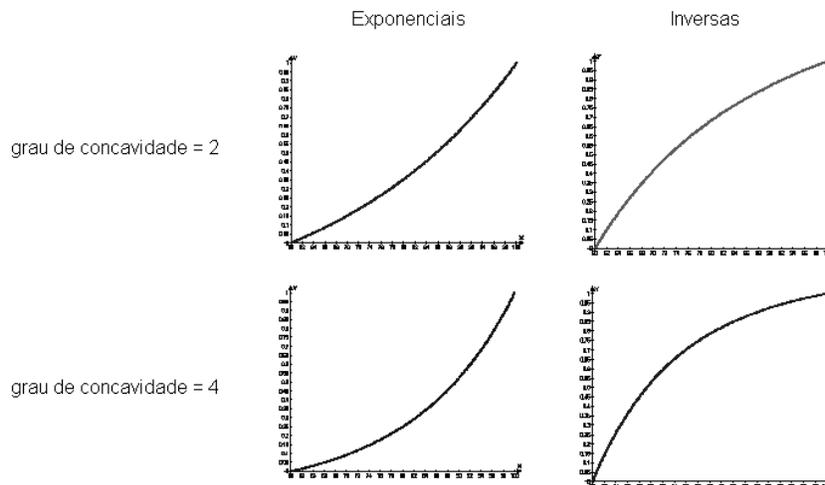
$$C = \frac{1}{e^{Kx_1^\alpha}}$$

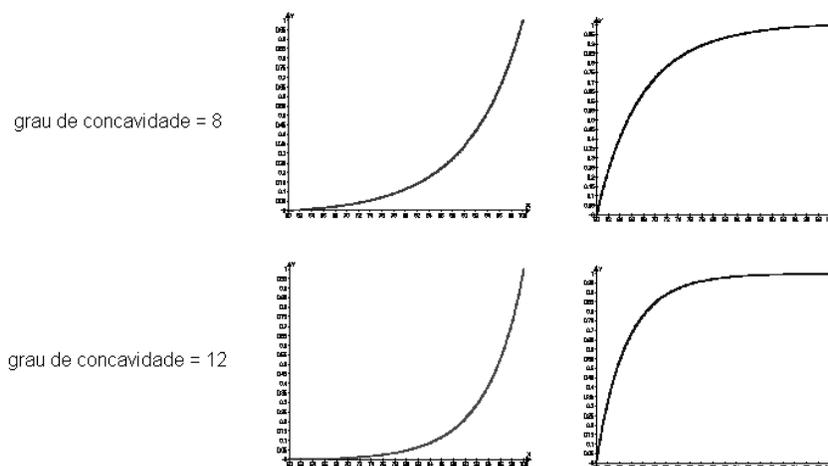
$$K = \frac{\ln(2)}{x_2^\alpha - x_1^\alpha}$$

De forma semelhante à exponencial, a inversa foi determinada, chegando-se a particular função inversa:

$$Inversa(x) = -\frac{x_1^\alpha x_2^\alpha}{x^\alpha (x_2^\alpha - x_1^\alpha)} + \frac{x_2^\alpha}{x_2^\alpha - x_1^\alpha}$$

A Figura 2 traz exemplos das funções exponenciais e inversas (todas crescentes), com diferentes graus de concavidade (2, 4, 8 e 12). A interface gráfica do sistema deve permitir que os criadores das entidades Comprador e Fornecedor visualizem os gráficos, podendo manipular o grau de concavidade. Os gráficos das funções lineares (retas) também devem ser mostrados.





**Figura 2** – Exemplos das Funções Exponenciais e Inversas

Assumiu-se que, para o Comprador, as funções para o preço, prazo de entrega e quantidade, são sempre não-crescentes (crescentes ou constantes).

O algoritmo para a negociação, apresentado na seção seguinte, exige que ambos os negociadores estejam munidos de funções de valor para expressar suas preferências. Assim, necessita-se construir funções duais às dos Compradores para os Fornecedores. Por exemplo, se as funções dos compradores são não-crescentes, as funções dos fornecedores serão assumidas como não-decrescentes.

Também será preciso, para o algoritmo de negociação, achar as inversas das funções apresentadas, pois, para este contexto em particular, será necessário encontrar o valor do preço, dada a sua função de valor. Assim, a seguir, apresentam-se as inversas das funções linear, exponencial e inversa, respectivamente:

$$Inv\_Linear(x) = x_1 - x(x_1 - x_2)$$

$$Inv\_Exponencial(x) = \left[ \frac{x_2^\alpha - x_1^\alpha}{\ln(2)} \ln(x+1) + x_1^\alpha \right]^{\frac{1}{\alpha}}$$

$$Inv\_Inversa(x) = \left( \frac{C}{x-K} \right)^\alpha$$

onde:

$$C = -Kx_1^\alpha$$

$$K = \frac{x_2^\alpha}{x_2^\alpha - x_1^\alpha}$$

### **4.3 Os limites das funções para diferentes tomadores de decisão: um problema para a negociação**

Na próxima seção, será apresentada a implementação do mecanismo de negociação, que se baseia nas funções de valor dos negociadores. Para haver negociação, é necessário que haja alguma interseção entre os domínios das funções. Isso, contudo, não é suficiente para um adequado funcionamento do mecanismo.

Suponha que o Fornecedor tenha os valores 1 e 300 como limites da sua função de valor associada a variável quantidade, por exemplo. Por sua vez, o Comprador deseja comprar 100 unidades, tolerando até, por exemplo, 120. Enquanto o intervalo do Fornecedor tem 300 elementos, o do Comprador tem apenas 21. Nesta configuração, caso se esteja avaliando o impacto na Função Valor Total de aumentar de 120 para 121 unidades, ter-se-ia um considerável impacto para o Comprador, enquanto, para o Fornecedor, o impacto seria desprezível, para quase todas as comportamentos possíveis da função de valor desse último.

Para evitar que ocorram essas incompatibilidades no tamanho dos domínios das funções, buscou-se amenizar o problema, através de uma etapa batizada de “handshake”, onde o Comprador informa quais seus limites máximos. Para a quantidade, em particular, o Fornecedor também saberá o limite mínimo, que é a quantidade cotada. Para o prazo, admite-se que o limite inferior para o Comprador seja igual a 1 dia. Se o Fornecedor tiver o mesmo prazo mínimo, os intervalos serão os mesmos. Senão, serão, ao menos, compatíveis. No preço, dificilmente os dois terão os mesmos intervalos, mas também serão intervalos compatíveis, viabilizando possíveis melhorias pelo algoritmo.

### **4.4 O algoritmo para a negociação colaborativa: passo a passo da solução proposta**

Raiffa (1982) apresentou um método para negociação colaborativa (ou integrativa), consistindo em um negociador apresentar proposições de dois pontos de indiferença ao ponto atual da negociação (como mostrado anteriormente): um ponto, um pouco “à direita”, e um outro, um pouco “à esquerda” na mesma curva de indiferença. O outro negociador poderia escolher um dos dois pontos (o de maior retorno de sua função de valor) para ser o novo ponto atual da negociação. O processo é repetido, sucessivas vezes, alternando o negociador proponente.

Para que fosse possível a utilização do método, ambos os negociadores necessitavam ter funções de valor (ou de utilidade) como forma de ordenar suas preferências, ou, ao menos, fazer inferência de pontos indiferentes, como se houvesse uma curva de indiferença.

No texto citado, não há uma formalização em notação matemática ou computacional desse método. A apresentação do mesmo foi feita para duas variáveis. Assim, duas coisas são necessárias para utilizá-lo na negociação integrativa de compras: formalizá-lo (um algoritmo) e expandi-lo para três variáveis (preço, prazo de entrega e quantidade). A variável qualidade não entra na negociação, pois a modelagem prevê que cada acordo chegado tenha uma marca já atribuída (Néri, 2002). Assim, ela só é usada no processo de decisão: escolher melhor cotação.

Vale lembrar que a negociação é feita apenas entre o comprador e o fornecedor vencedor da concorrência. Será desconsiderada, no mecanismo de negociação proposto, a discussão sobre o impacto da abertura dos valores de reserva de cada negociador.

A Figura 3 mostra o algoritmo de negociação em forma de fluxograma.

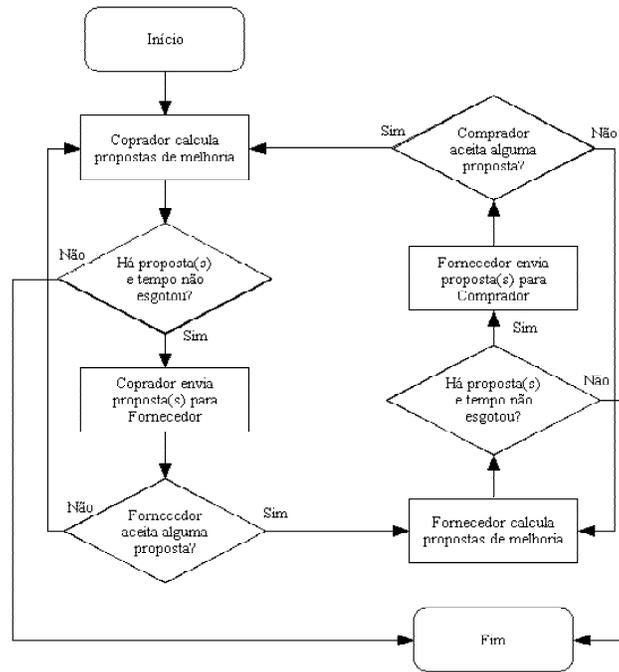


Figura 3 – Algoritmo de negociação

A seguir, a implementação, em Java, do algoritmo sugerido:

```

public void negociacao()
{
    Negociador proponente = comprador;
    Negociador tomadorDecisao = fornecedorEscolhido;
    Negociador aux;
    Proposta propostaAtual;
    Atributos atributosAtual;
    do
    {
        propostaAtual = proponente.enviaProposta();
        if (propostaAtual == null)
        {
            break;
        }
        atributosAtual = tomadorDecisao.escolheProposta(propostaAtual);
        if (atributosAtual != null)
        {
            tomadorDecisao.aceitaProposta(atributosAtual);
            proponente.aceitaProposta(atributosAtual);
            invertePapeis(tomadorDecisao, proponente);
        }
    }
    while (parada.satisfeito());
}

protected void invertePapeis(Negociador tomadorDecisao, Negociador proponente)
{
    Negociador aux = tomadorDecisao;
    tomadorDecisao = proponente;
    proponente = aux;
}
    
```

Figura 4 – Implementação do Algoritmo de Negociação

Na implementação acima, há uma classe chamada *Negociador* que se especializa em *Comprador* e em *Fornecedor*. Para futuras implementações em outros contextos, bastam novas especializações dessa classe.

A classe *Proposta* é uma lista de objetos da classe *Atributos*, que são, basicamente, conjuntos dos valores das variáveis em negociação (neste caso, preço, prazo de pagamento e quantidade). Esses objetos da classe *Atributos* representam, nesse instante, os possíveis acordos que um negociador oferece ao outro: são os acordos indiferentes em preferência. Assim, um objeto da classe *Proposta* é um conjunto de possíveis acordos, enviados pelo *proponente*, para apreciação do *tomadorDecisao*.

Os acordos indiferentes em preferência são gerados pela manipulação das variáveis discretas (prazo de pagamento e quantidade), somando-se ou subtraindo-se um fator (igual a 1) e, através da variável contínua preço, são feitos os ajustes, de forma a manter-se o mesmo retorno da Função Valor Total. A seguir, a implementação desse método (*enviaProposta*), que ainda precisa ser generalizado para a aplicação em outros contextos, com, por exemplo, *n* variáveis, discretas e contínuas.

```
public Proposta enviaProposta()
{
    int valor = 0;
    int numIteracoes = fatorIncremento*2;
    Proposta proposta = new Proposta();

    for (int i = 0; i <= numIteracoes; i++)
    {
        valor = i/2;
        int sinal = (int) Math.pow(-1, i);
        proposta.inserir(this.adicionaAtributos(valor, sinal*fatorIncremento));
        if (i != numIteracoes)
            proposta.inserir(this.adicionaAtributos(sinal*fatorIncremento, valor));
        else
            proposta.inserir(this.adicionaAtributos(-sinal*fatorIncremento, -valor));
        if (valor != 0) {
            proposta.inserir(this.adicionaAtributos(-valor, sinal*fatorIncremento));
            proposta.inserir(this.adicionaAtributos(sinal*fatorIncremento, -valor));
        } //if
    } //for

    this.removeAtributosProposta(proposta);
    this.fatorIncremento++;

    return proposta;
}
```

**Figura 5** – Método *enviaProposta()*

Para entender a implementação acima, é necessário entender os métodos *adicionaAtributos()* e *removeAtributosProposta()*. O primeiro, realizado para cada possível acordo, recebe dois parâmetros: incremento na quantidade e incremento no prazo, que podem assumir zero (0), “+*fatorIncremento*” ou “-*fatorIncremento*”. Como houve um aumento, ou uma diminuição, dos valores da variável quantidade, prazo, ou de ambas, mudou-se o retorno da Função Valor Total do negociador. Para que o retorno continue o mesmo, necessita-se de ajuste no preço.

O ajuste no preço é feito através de um método que calcula a diferença entre os retornos das Funções Valor Total do acordo atual e do possível acordo que será proposto. Soma-se essa diferença ao retorno da função preço do acordo atual. O resultado é o argumento da função inversa do preço, achando qual deve ser o novo valor para preço no possível acordo, de tal

sorte que a Função Valor Total continue a mesma, ou seja, o possível acordo continue indiferente em preferência ao atual, para o negociador proponente.

O valor de *fatorIncremento* é, inicialmente, igual a 1. Assim, em um primeira “rodada” de oferta de possíveis acordos (ou proposta), um negociador propõe oito possíveis acordos indiferentes em preferência. São os acordos mais próximos do acordo atual, se está se pensando em uma “zona matemática de acordo”.

Caso nenhum dos possíveis acordos seja aceito pelo negociador parceiro (*tomadorDecisao*), o negociador proponente irá buscar possíveis acordos um pouco mais distantes na “zona matemática de acordo”. Assim, o *fatorIncremento* é adicionado em uma unidade de incremento, definida como 1. Essa busca por acordos indiferentes é feita até o negociador *tomadorDecisao* aceitar algum novo acordo ou, no algoritmo de negociação, ter se chegado à condição de parada (*parada.satisfeito() == true*).

O método *removeAtributosProposta()* simplesmente elimina os possíveis acordos gerados que passaram dos limites das funções, ou seja, verifica se os valores para quantidade, prazo de pagamento e preço, são valores aceitos pelos critérios restritivos. Os possíveis acordos gerados que não satisfaçam os critérios de restrição são eliminados da lista *Proposta*.

Para entender a implementação do algoritmo de negociação, ainda são necessárias explicações sobre os métodos *escolheProposta()* e *aceitaProposta()*. O primeiro, que é realizado pelo negociador *tomadorDecisao*, calcula todos os possíveis acordos enviados pelo negociador proponente, em busca daquele que maximize sua Função Valor Total, que será o novo acordo. Entretanto, se nenhum dos possíveis acordos propostos aumenta a Função Valor Total do negociador *tomadorDecisao*, ele espera por uma próxima “rodada” de possíveis acordos oferecidos pelo negociador proponente.

O método *aceitaProposta()* é usado quando, no método anterior, acha-se um acordo melhor, que passa a ser o atual, recomeçando todo o processo, sendo que os negociadores invertem os papéis: vira *tomadorDecisao* quem era *proponente* e vice-versa. O *fatorIncremento* retorna a valer 1, como de início.

Na Figura 5, a palavra “*this*” é apenas um detalhe de implementação que serve para distinguir sobre qual negociador se está tratando.

#### 4.5 A implementação: outros detalhes

As implementações dos mecanismos foram feitas na linguagem de programação Java, em aproximadamente 1700 linhas, independentemente dos conceitos de sistemas multiagentes. Buscou-se, também, torná-las independentes em relação à aplicação no contexto de compras. Para adaptá-las em outras aplicações, não são necessários grandes esforços, mas apenas alguns ajustes.

Com a estrutura de classes programada, pode-se, com algumas poucas adaptações, utilizar os mecanismos de decisão e de negociação para a utilização em outros contextos, com “n” variáveis, além de incluir novos comportamentos (funções), como, por exemplo, parabólicos ou logarítmicos.

A interface gráfica não foi implementada. Todos os testes foram feitos no próprio ambiente de programação (Borland JBuilder 6.0), através da interface padrão (caracteres).

## 5. Os Testes

### 5.1 Metodologia da simulação

Para observar o comportamento dos mecanismos, escolheram-se três produtos para compra que constam em uma tabela de suprimentos de tecnologia da informação da empresa pesquisada. O critério de escolha foi a observação de produtos com características distintas de compra. São dados adaptados de uma empresa real, ligeiramente diferentes para poderem se tornar públicos (Néri, 2002). Todos os produtos da tabela podem ser encarados como commodities. Assim sendo, preço é praticamente o determinante da compra. Todavia, na empresa pesquisada, não são considerados commodities, pois a variável qualidade ainda representa um diferencial. Quantidade e prazo de entrega também têm importância, justificando o uso de funções multiatributos de valor para as tomadas de decisão e compra.

O preço sendo a variável mais importante da compra, com importância bem maior que as demais, limita bastante o uso do mecanismo de negociação. Mesmo podendo ter sido eleitos outros tipos de negociação, supostamente mais adequados, preferiu-se observar a negociação de compra de suprimentos por constituírem uma prática rotineira das empresas e que, uma vez automatizada, representaria uma boa queda em custos.

Para cada um dos produtos escolhidos, criou-se um cenário para testes. Em cada cenário, escolheram-se três hipotéticos fornecedores candidatos. Os valores cotados para as variáveis foram escolhidos, observando a coerência em relação a possíveis reais fornecedores. As funções, entretanto, foram definidas sem a preocupação de corresponder à realidade de qualquer empresa, servindo para simular a negociação. De forma semelhante, todos os outros “inputs” do sistema obedeceram à mesma sistemática.

Usou-se uma planilha Excel, configurada para, dados os valores das variáveis, seus pesos e tipos de funções associadas, exibir o retorno das Funções Valor Total. Todas as avaliações das cotações feitas pelo programa e pela planilha tiveram o mesmo retorno das funções, para ambos os negociadores. Os valores dos acordos chegados, após a negociação, também foram verificados pela planilha. Todos os valores das Funções Valor Total da implementação e da planilha foram coincidentes.

A seguir, serão apresentados os cenários de testes, através de três tabelas, cada. Na primeira, os “inputs” necessários para configurar o Comprador podem ser observados. Já na segunda, observam-se os “inputs” necessários para configurar os Fornecedores. Finalmente, a terceira traz uma comparação entre os acordos, antes e depois da negociação. As células das tabelas em destaque (fundo cinza) indicam que os valores foram passados ou calculados pelo Comprador.

### 5.2 Cenário de Testes 1

Para o primeiro cenário, foi escolhido um produto que tivesse variáveis com intervalos relativamente grandes, isto é, o preço, o prazo de entrega e a quantidade têm, potencialmente, boa margem de flexibilidade no momento da negociação. Nesse cenário, considerou-se que o prazo e a quantidade tinham importância expressiva na decisão. Coerentemente, o preço continua, todavia, sendo a mais importante das variáveis de decisão. A Tabela 1 mostra as configurações do Comprador.

**Tabela 1** – Configuração do Comprador no Cenário de Testes 1

<b>produto</b>	Cabos para conexões de Hubs 3Com, part number 3C16420				
<b>quantidade</b>	150				
<b>marca</b>	<b>qualidade (0-10)</b>				
-	-				
<b>variável</b>	<b>peso</b>	<b>função</b>	<b>fator</b>	<b>mínimo</b>	<b>máximo</b>
preço	0,60	inversa	4	80	120
prazo	0,20	exponencial	6	1	15
quantidade	0,20	exponencial	4	150	200
qualidade	-	-	-	-	-

A Tabela 2 apresenta a configuração dos três fornecedores, bem como a avaliação feita pelo Comprador das cotações enviadas pelos mesmos (última coluna). O número do Fornecedor eleito está destacado em negrito, bem como a avaliação da sua cotação, feita pelo Comprador. Somente com esse fornecedor que o Comprador realizará a negociação.

**Tabela 2** – Configuração e Avaliação dos Fornecedores no Cenário de Testes 1

fornecedor	variável	peso	função	fator	mínimo	máximo	cotação enviada	função valor total para comprador
1	preço	0,5	inversa	8	95	120	100	0,558545
	prazo	0,2	linear	-	3	15	3	
	quantidade	0,3	inversa	8	150	200	150	
	marca	-	-	-	-	-	-	
2	preço	0,5	inversa	6	92	120	98	<b>0,583959</b>
	prazo	0,15	exponencial	4	3	15	5	
	quantidade	0,35	inversa	8	150	200	150	
	marca	-	-	-	-	-	-	
3	preço	0,7	inversa	4	102	120	102	0,535240
	prazo	0,0	linear	-	1	15	1	
	quantidade	0,3	inversa	6	150	200	150	
	marca	-	-	-	-	-	-	

Assim como o Comprador, os fornecedores dão grande importância ao preço. A quantidade também representa, nesse cenário, importante papel, tendo, para os três fornecedores, pesos associados relativamente altos. Note-se que quaisquer pequenas alterações iniciais no preço ou na quantidade geram aumentos significativos nas Funções Valor Total dos fornecedores, pois os três têm funções inversas com altos graus de concavidade associadas a essas variáveis.

Dadas as configurações acima, a Tabela 3 mostra um comparativo entre os acordos anterior e posterior à aplicação do mecanismo de negociação. É importante lembrar que a função qualidade permanece a mesma, após a negociação, pois a marca continua a mesma, sempre. Em particular, nesse cenário de testes, não foram determinadas as possíveis marcas. Assim, a variável qualidade não influencia na decisão.

**Tabela 3** – Comparativo Antes/Após Negociação no Cenário de Testes 1

	variável	valores	função valor total do comprador	função valor total do comprador (sem qualidade)	função valor total do fornecedor 2
<b>antes da negociação</b>	preço	98	0,583959	0,583959	0,199073
	prazo	5			
	quantidade	150			
<b>depois da negociação</b>	preço	92,813953	0,605349	0,605349	0,249478
	prazo	5			
	quantidade	166			

Nesse teste, o algoritmo de negociação conseguiu melhorar o acordo, um pouco para o Comprador, e, significativamente para o Fornecedor.

Fazendo algumas variações nos “inputs” do cenário de testes, pode-se chegar a resultados diferentes. Por exemplo, se o Fornecedor 1 tivesse enviada o preço como 98, assim como fez o Fornecedor 2, ele seria o escolhido na concorrência, pois sua cotação teria sido avaliada pelo Comprador com, apenas, 0,0004 a mais de retorno da Função Valor Total, em relação ao Fornecedor 2. Contudo, após negociação, ter-se-ia chegado a um acordo cuja Função Valor Total do Comprador seria 0,007579 menor.

Alterações nas funções também, evidentemente, podem causar (e geralmente causam) modificações na escolha do fornecedor e/ou nos resultados da negociação. Testou-se, por exemplo, trocar o grau de concavidade da função preço do Fornecedor 2 de 6 para 4 e, depois, para 2. Chegou-se a acordos próximos, mas diferentes.

### 5.3 Cenário de Testes 2

Nos próximos cenários de testes, buscou-se acompanhar o comportamento do algoritmo para a compra de produtos que oferecem pequenas margens para negociação, ou, em outras palavras, que têm funções com pequenos intervalos de domínio. Como o algoritmo é fundamentado na cessão e ganho dos valores das variáveis, acredita-se que, nesses casos, as melhorias não serão grandes, se houver alguma.

No segundo cenário, o produto escolhido foi um Disco Rígido. A escolha da quantidade foi determinada como quatro unidades, podendo estender-se até seis. A Tabela 4 traz mais detalhes sobre a configuração do Comprador.

Note-se que, nesse teste, a qualidade está associada às marcas que podem ser compradas. Considerou-se a variável de decisão qualidade importante, pois a mesma recebeu um percentual expressivo na decisão (20%).

**Tabela 4** – Configuração do Comprador no Cenário de Testes 2

<b>produto</b>	Disco rígido de 36.4GB, SCSI para Desktop				
<b>quantidade</b>	4				
<b>marca</b>	<b>qualidade (0-10)</b>				
A	8				
B	9				
C	8				
D	10				
<b>variável</b>	<b>peso</b>	<b>função</b>	<b>fator</b>	<b>mínimo</b>	<b>máximo</b>
preço	0,5	inversa	8	1000	1500
prazo	0,1	exponencial	6	1	15
quantidade	0,2	exponencial	10	4	6
qualidade	0,2	peso*(qualidade/10)	-	-	-

A Tabela 5 apresenta os fornecedores que concorrem à compra. Cada um deles vende disco rígido de uma marca diferente do outro.

**Tabela 5** – Configuração e Avaliação dos Fornecedores no Cenário de Testes 2

fornecedor	variável	peso	função	fator	mínimo	máximo	cotação enviada	função valor total para comprador
1	preço	0,6	exponencial	8	1350	1500	1430	0,477647
	prazo	0,2	inversa	4	5	15	10	
	quantidade	0,2	exponencial	8	4	6	4	
	marca	B	-	-	-	-	-	
2	preço	0,7	inversa	6	1400	1500	1470	0,503371
	prazo	0,05	linear	-	4	15	5	
	quantidade	0,25	exponencial	2	4	6	4	
	marca	D	-	-	-	-	-	
3	preço	0,4	inversa	4	1300	1500	1350	0,486670
	prazo	0,2	exponencial	12	1	15	5	
	quantidade	0,4	inversa	6	4	6	4	
	marca	C	-	-	-	-	-	

Finalmente, a Tabela 6 expõe os dados comparativos entre os acordos anterior e posterior à negociação.

**Tabela 6** – Comparativo Antes/Após Negociação no Cenário de Testes 2

	variável	valores	função valor total do comprador	função valor total do comprador (sem qualidade)	função valor total do fornecedor 2
<b>antes da negociação</b>	preço	1470	0,503371	0,303371	0,528630
	prazo	5			
	quantidade	4			
<b>depois da negociação</b>	preço	1471,083401	0,503371	0,303371	0,530881
	prazo	4			
	quantidade	4			

Como esperado, não houve grandes mudanças entre os acordos. A mudança foi quase imperceptível, ocorrendo apenas uma discreta melhoria para o Fornecedor. Já o Comprador, manteve seu resultado da Função Valor Total. O que ocorreu aqui foi a aceitação, pelo Fornecedor, de um dos possíveis acordos oferecidos pelo Comprador, que, por sua vez, não conseguiu oferecer nenhuma proposta nova de acordo que fosse do interesse do Comprador. Observe-se que, além do pequeno intervalo possível da função da variável quantidade, havia também um pequeno intervalo para o prazo. Nesse cenário, o Comprador também dificultou a negociação quando definiu suas funções com altos graus de concavidade.

Algumas variações foram feitas nesse cenário, mas todas continuaram a indicar que a negociação não melhorava o acordo para o Comprador.

#### 5.4 Cenário de Testes 3

No cenário de testes 3, buscou-se analisar a compra de um produto com preços baixos. A variável de decisão preço tem um peso associado muito alto e um intervalo pequeno para negociação. Não há preocupação com a marca, conseqüentemente, com a qualidade. A quantidade apresenta um bom intervalo para negociação.

**Tabela 7** – Configuração do Comprador no Cenário de Testes 3

<b>produto</b>	Abraçadeiras de Nylon				
<b>quantidade</b>	100				
<b>marca</b>	<b>qualidade (0-10)</b>				
-	-				
<b>variável</b>	<b>peso</b>	<b>função</b>	<b>fator</b>	<b>mínimo</b>	<b>máximo</b>
preço	0,8	exponencial	8	0,10	0,17
prazo	0,1	linear	-	1	15
quantidade	0,1	exponencial	4	100	120
qualidade	0	-	-	-	-

A Tabela 8 apresenta quais as configurações para os fornecedores.

**Tabela 8** – Configuração e Avaliação dos Fornecedores no Cenário de Testes 3

fornecedor	variável	peso	função	fator	mínimo	máximo	cotação enviada	função valor total para comprador
1	preço	0,6	exponencial	8	0,14	0,17	0,15	0,648220
	prazo	0	linear	-	1	15	1	
	quantidade	0,4	exponencial	8	100	120	100	
	marca	-	-	-	-	-	-	
2	preço	0,7	inversa	6	0,13	0,17	0,14	0,764220
	prazo	0,15	linear	-	1	15	5	
	quantidade	0,15	inversa	8	100	120	100	
	marca	-	-	-	-	-	-	
3	preço	0,7	inversa	4	0,16	0,17	0,18	-
	prazo	0,05	exponencial	12	1	15	2	
	quantidade	0,25	inversa	6	100	120	100	
	marca	-	-	-	-	-	-	

O Fornecedor 3 não obedeceu aos critérios restritivos. Assim, foi eliminado da concorrência, por isso sua Função Valor Total não aparece na tabela. Observe-se também que o Fornecedor escolhido tem, assim como o Comprador, um peso alto associado à variável preço (70% de importância total da compra). O resultado do processo de negociação é apresentado na Tabela 9.

**Tabela 9** – Comparativo Antes/Depois Negociação no Cenário de Testes 3

	variável	valores	função valor total do comprador	função valor total do comprador (sem qualidade)	função valor total do fornecedor 2
antes da negociação	preço	0,14	0,764220	0,764220	0,356927
	prazo	5			
	quantidade	100			
depois da negociação	preço	0,140173	0,769966	0,769966	0,370842
	prazo	1			
	quantidade	104			

Houve um aumento ínfimo na Função Valor Total do Comprador e, um pequeno na do Fornecedor. Mesmo com pequenas possibilidades de melhorias, o processo de negociação fez um ajuste mínimo nas variáveis. Note-se que, na prática, não houve mudança na variável preço.

Além dos pequenos intervalos das funções, o que dificultou a melhoria do acordo pelo algoritmo de negociação foi o alto valor dado ao peso da variável preço. Quando se testou o cenário, com alterações nos pesos do comprador, sendo 0,6 para o preço e 0,2 para prazo e quantidade, houve melhorias significativas nos acordos antes e depois da negociação.

## 5.5 Observações finais sobre os testes

Pode-se observar que os resultados da negociação são extremamente sensíveis a todos os “inputs” de configuração. Assim, recomenda-se, em uma aplicação real, um extremo cuidado nas definições de todos esses “inputs”. Isto inclui a definição das funções que, em uma situação real, deve representar bem as estruturas de preferência dos tomadores de decisão. As funções aqui apresentadas e testadas serviram para demonstrar os mecanismos de decisão e negociação propostos, não sendo fundamentadas na realidade de nenhuma empresa.

Uma outra observação é que, como teste, na implementação da negociação, inverteu-se a ordem dos negociadores, ficando o Fornecedor responsável pela primeira proposta. Não houve diferenças significativas.

## 6. Considerações Finais

### 6.1 Resultados alcançados

- Em todos os testes, o algoritmo de negociação cumpriu seu papel: buscar possíveis melhorias mútuas. Em nenhum dos três cenários de testes realizados, houve uma diminuição da Função Valor Total de qualquer negociador. Essa característica é implícita ao fundamento do algoritmo e da negociação colaborativa, de uma forma mais geral. Portanto, a solução apresentada é uma representante das possíveis formas de negociação colaborativa. Também se pôde constatar que a idéia de Raiffa é passível de implementação para negociação de compras de suprimentos, apesar de, em princípio, não representarem bons tipos de negociação para tal idéia, como discutido anteriormente.
- A decisão de qual a melhor cotação recebida para o fornecimento dos produtos desejados foi estruturada através de uma soma ponderada de funções de valor. Para cada variável de decisão de compra, existiam um peso e uma função de valor associados. A negociação apresentou-se, neste contexto, como sendo uma excelente etapa para ajustes dos valores das variáveis que, nitidamente, poderiam ser melhorados. Nos testes, houve maiores melhorias para os fornecedores que para os compradores. Isso é explicado pelas diferenças nos domínios das funções dos negociadores. Os fornecedores, em geral, tiveram funções com domínios bem menores. Assim, um pequeno deslocamento no valor de uma variável representava um maior retorno da função associada para o Fornecedor, em relação ao Comprador;
- Mesmo sendo o Comprador o foco do sistema proposto, boas melhorias para o Fornecedor ajudam a manter bons relacionamentos com esse último, o que também traz vantagens para o primeiro.

### 6.2 Limitações

- As implementações dos mecanismos foram feitas para o contexto específico da compra de suprimentos de tecnologia da informação da empresa pesquisada (Néri, 2002). Os dados usados nas simulações não são reais, apesar de terem sido baseados em valores reais. Também não foram escolhidos aleatoriamente. Foram escolhidos valores que se imaginavam problemáticos e, nos três cenários, com características distintas entre si;
- A escolha das funções exponenciais, lineares e inversas, não representa, necessariamente, a realidade de nenhuma empresa. Foram funções arbitrárias para fins de simulação;

- A fase de “*handshake*” foi de fundamental importância para o funcionamento adequado do mecanismo de negociação. Sem essa fase, o Fornecedor, via de regra, estabeleceria domínios para suas funções bem maiores que faz o Comprador, inviabilizando a negociação. Todavia, quando o Comprador informa os limites máximos de suas funções, ele abre informações que, *a priori*, não deveriam ser expostas, pois podem levar ao uso malicioso das mesmas por parte dos fornecedores, como elevar, em conjunto, o preço dos produtos para próximo do valor máximo que o Comprador está disposto a pagar. Questões sobre a segurança das informações estão fora do escopo do trabalho;
- A negociação foi implementada para dois atributos discretos (prazo de pagamento e quantidade) e um contínuo (preço). Quando um negociador propõe acordos indiferentes em preferência, esses valores são criados através do incremento ou decremento dos atributos discretos. Para se manter a indiferença, ajustes são necessários na variável contínua. Assim, em negociações diferentes da apresentada, a criação dos acordos indiferentes em preferência pode exigir solução distinta.

### 6.3 Sugestões de melhorias para futuros trabalhos

- Variações foram feitas nas configurações dos Compradores e dos Fornecedores. Em alguns casos, como era esperado, a decisão muda. Mecanismos de decisão com análise de sensibilidade poderiam ser pensados em futuros trabalhos;
- Não se levaram em consideração, neste trabalho, aspectos de segurança da informação. Um deles, de particular interesse neste contexto, é deixar que seu parceiro de negociação descubra suas estruturas de preferência e seus valores de reserva. A forma possível para haver a negociação sem a abertura de tais valores é ter as entidades negociadoras pertencentes a uma terceira parte, confiável a ambos os parceiros;
- Em uma futura implementação da solução aqui proposta, aconselha-se criar funções sob medida para a empresa, o que pode ser feito de várias maneiras, como, por exemplo, através de entrevistas para descobrir algumas preferências e tentar associar alguma função matemática de acordo com o levantado;
- Os mecanismos de decisão e negociação propostos neste artigo podem ser adaptados facilmente para outros contextos. Como sugestão de futuros trabalhos, têm-se: generalização das classes e métodos utilizados que ainda não são genéricas, estudo aprofundado da complexidade computacional da solução, melhoria dos mecanismos (por exemplo, uso de heurísticas para a determinação das condições de parada ou técnicas de inteligência artificial para aprendizado de quais os melhores caminhos a seguir na “zona de acordo”);
- Outro desafiador futuro trabalho é generalizar a negociação para vários negociadores. Por exemplo, o Comprador poderia negociar com vários fornecedores, bem como esses últimos, com vários compradores;
- Aplicar as soluções aqui propostas para negociações diferentes. Acredita-se, que em alguns casos, a negociação melhorará bastante. Em decisão de compras de suprimentos, tanto os compradores quanto os fornecedores preocupam-se, principalmente, com os preços. São, muitas vezes, compras de commodities. Uma negociação entre sindicato e empresários ou compra e venda de uma empresa pode oferecer cenários onde as melhorias dos mecanismos serão bem mais visíveis.

### Referências Bibliográficas

- (1) Baily, P.; Farmer, D.; Jessop, D. & Jones, D. (2000). *Compras – Princípios e Administração*. São Paulo.
- (2) Chavez, A. & Maes, P. (1996). *Kasbah: An Agent Marketplace for Buying and Selling Goods*. First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, London.
- (3) Clemen, R.T. (1996). *Making hard decisions: an introduction to decision analysis*. 2<sup>nd</sup> ed. Duxbury Press, Belmont.
- (4) Conway, D.G. & Koehler, G.J. (2000). Interface agents: caveat mercator in electronic commerce. *Decision Support Systems*, **27**, 355-366.
- (5) Franklin, S. & Graesser, A. (1996). Is it an Agent, or just a Program? A taxonomy for Autonomous Agents. *Proceedings of the Third International Workshop on Agents Theories, Architectures, and Languages*, Spring, Verlag.
- (6) Kang, N. & Han, S. (2002). Agent-based e-market system for more fair and efficient transaction. *Decision Support Systems*, **34**, 157-165.
- (7) Keeney, R.L. & Raiffa, H. (1976). *Decisions with Multiple Objectives: preferences and value trade-offs*. John Wiley & Sons, New York.
- (8) Kersten, G.; Michalowisk, W.; Szpakowicz, S. & Koperczak, Z. (1991). Restructurable representations of negotiation. *Management Science*, **37**(10), 1269-1290.
- (9) Lewicki, R.; Saunders, D. & Minton, J. (1997). *Essentials of Negotiation*. Irwin.
- (10) Néri, E.L. (2002). *Agentes de Software nas Empresas: Um Estudo de Caso na Área de Compras*. Dissertação de Mestrado, UFRGS, Curso de Pós-Graduação em Administração. Porto Alegre.
- (11) Néri, E.L. & Hoppen, N. (2003). Modelagem de um Sistema Multiagente para Compras de Suprimentos de TI Usando Agent UML. *Anais do XXVII Encontro da ANPAD*, Atibaia.
- (12) Pattie, M.; Guttman, R.H. & Moukas, A.G. (1999). Agents that Buy and Sell: Transforming Commerce as we Know It. *Communications of the ACM*, **43**, 81-ff.
- (13) Preist, C.; Bye, A. & Bartolini, C. (2001). Economic Dynamics of Agents in Multiple Auctions. *Proceedings of the Fifth International Conference on Autonomous Agents (ACM AGENTS'01)*, Montreal, Quebec, Canadá, 545-551.
- (14) Raiffa, H. (1982). *The Art & Science of Negotiation*. Harvard University Press, Cambridge, Massachusetts.
- (15) Rosenschein, J. & Zlotkin, G. (1994). *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. MIT Press.
- (16) Turban, E.; Lee, J.K.; King, D. & Chung, M. (2000). *Electronic Commerce: A Managerial Perspective*. 1<sup>st</sup> ed. Prentice Hall, New Jersey.
- (17) Yuan, S. (2002). A personalized and integrative comparison-shopping engine and its applications. *Decision Support Systems*, **34**, 139-156.