

# 시간제약 환경에서 다중 속성을 이용한 확장된 협상 에이전트

## (An Extended Negotiation Agent Using Multi-Issues under Time-Constraint Environment)

김 현 식 <sup>†</sup> 양 성 봉 <sup>\*\*</sup>  
(Hyun-Shik Kim) (Sung-Bong Yang)

**요 약** 인터넷은 이미 우리 생활의 한 부분이 되었으며, 많은 생활 환경들이 인터넷을 중심으로 변화되고 있으며 그 중 가장 두드러진 분야가 전자상거래이다. 그러나 현재의 전자상거래는 판매자가 일방적으로 제시하는 조건에 구매자는 단순히 거래를 하거나, 제시된 조건을 비교하여 보다 더 좋은 조건을 찾는 수준에 머물러 있다. 향후 전자상거래는 어느 한쪽의 일방적인 제시가 아닌 서로의 이익을 최대화 할 수 있는 방안 중의 하나인 협상(negotiation)이 필수적이며, 또한 전자상거래가 전통적인 상거래 행위를 대신 할 수 있도록 하기 위해 하나의 속성이 아닌 더 많은 협상 대상을 요구하게 된다.

본 논문의 목적은 다중 속성을 이용한 협상을 통해 두 에이전트 사이의 균형 있는 이익과, 높은 협상 성사율을 이룰 수 있는 시간제약 환경에서의 협상 모델을 제안한다. 협상 모델은 두 에이전트가 각각의 협상 시간을 가지고 다중 속성에 대해 두가지 속성 값 변경 전략(교차적, 동시 전략)과 서로의 제안 값 선택 전략을 제시하고, 제안된 동일한 형태의 협상 메커니즘별로 두 에이전트간의 협상 모델을 평가한다. 또한 제안된 협상 모델의 비교를 위하여 또 다른 협상 모델을 구현하여 비교 하였다. 제안된 협상 모델에서 동적인 conceder 협상 방법과 선형적 협상 방법이 두 에이전트간의 균형 있는 이익과 높은 협상 성사율을 보여 주었고, 전체적으로 교차적 속성 값 변경 전략을 사용 하였을 때 두 에이전트의 유틸리티 값의 합이 높은 것으로 나타났다.

**키워드** : 전자 상거래, 협상, 다중 속성, 교차적 전략, 동시 전략

**Abstract** The Internet has become a part of our life and has changed our living environments dramatically. However, Electronic Commerce (EC) stays currently at a level that a buyer deals only with an one-sided condition (price) proposed by a seller or compares the proposed condition to find a better one. As the agent technology makes progress, EC requires negotiation that is not one-sided but is a way to maximize profits of both seller and buyer. Moreover, negotiation in EC should consider multiple issues rather than a single issue to replace the traditional commerce.

In this paper we propose a negotiation model, which guarantees balanced profits between two agents through negotiation using multi-issues, and makes a deal successfully under time-constraint environment. The proposed negotiation model suggests strategies(alternative and simultaneous strategy) that change the value of issues with each agent has time for negotiation. In this paper, we also suggest a strategy that proposes an offer to each other. We compare the proposed negotiation model with another negotiation model.

The experimental results show that dynamic conceder tactic and linear tactic showed balanced profits and a high percentage of deals for negotiation between two agents, and the sum of utilities of two agents is high, when the alternative strategy is used.

**Key words** : Electronic commerce, negotiation, multi issues, alternative strategy, simultaneous strategy

<sup>†</sup> 정 회 원 : 연세대학교 컴퓨터과학과  
roknkim@hanmail.net

<sup>\*\*</sup> 비 회 원 : 연세대학교 컴퓨터산업공학부 부교수  
yang@cs.yonsei.ac.kr

논문접수 : 2002년 9월 30일  
심사완료 : 2003년 9월 4일

### 1. 서 론

현재의 전자상거래는 판매자가 일방적으로 제시하는 조건(가격)에 구매자는 단순히 거래를 하거나, 제시된

가격을 비교하여 보다 좋은 조건을 찾는 수준에 머물러 있으나, 향후 전자상거래는 어느 한쪽의 일방적인 제사가 아닌 서로의 이익을 최대화 할 수 있는 방안 중의 하나인 협상(negotiation)이 필수적이다. 협상이란 서로에게 좋은 이익을 얻을 수 있게 하기 위한 의사결정의 단계로, 상호작용(interaction), 절충(tradeoff), 양보(concession)를 통하여 문제를 해결하여 서로에게 이익이 되도록 하기 위한 하나의 과정이다. [1]에서 협상의 유형을 경쟁적(competitive) 협상과 협조적(cooperative) 협상 두 유형으로 분류 하고 있다. 경쟁적 협상은 어느 한 쪽이 일방적으로 이기거나 지는, 다시 말해 이익이 아니면 손해를 가져오는 협상을 말하고, 협조적 협상은 협상을 수행하는 주체가 서로 양보를 통해 서로에게 이익이 될 수 있도록 하는 협상을 말한다. 또한 에이전트를 사용한 대표적인 협상시스템의 예가 MIT Media Lab의 Kasbah이다[2]. Kasbah는 웹을 기반 한 멀티-에이전트(multi-agent) 시스템으로, 사용된 협상 전략은 시간에 따른 가격 변화, 즉 세 가지의 협상전략(linear, quadratic, exponential)을 사용하지만, 실제 제품을 사고 팔 때 소비자는 제품의 가격만을 고려하는 것이 아니라, 여러 가지의 특성을 고려하기 때문에 현실성이 부족하다. 전자상거래는 전통적인 상거래와는 현실적으로 차이가 있으며, 그 중 가장 큰 차이점은 바로 얼굴 없는(faceless) 상거래, 즉, 구매자나 판매자는 인터넷을 통하여 거래를 하기 때문에 서로에 대한 신뢰성이 필요하며, 또한 전자상거래가 전통적인 상거래 행위를 대신 할 수 있도록 하기 위해서는 협상의 대상이 가격 뿐만이 아닌 더 많은 협상 대상을 요구하게 된다.

본 논문은 다중 속성을 이용한 협상을 통해 두 에이전트 사이의 균형 있는 이익과 높은 협상 성사율을 이룰 수 있는 방법을 찾기 위해 시간제약 환경에서의 협상 모델을 제안하며, 협상에서 고려되는 다중 속성들에 대해 각 에이전트들은 두가지의 속성 값 변경 전략과, 서로의 제안 값 선택 전략을 제시하게 되고, 제안된 동일한 형태의 협상 메커니즘별로 두 에이전트간의 협상 결과와, 제안된 협상 모델과의 비교를 위해 [3]에서 제시한 협상 방법과 비교를 하였다.

이 논문의 구성은, 제2장에서 관련연구로 하나의 속성만을 사용하여 협상을 진행하는 모델을 살펴보고, 3장에서 다중속성을 이용한 협상 모델을 제시하고, 4장에서 제안한 협상 모델을 통한 실험 및 결과를, 그리고 마지막으로 결론 순으로 구성되어 있다.

## 2. 관련연구

이 장에서는 먼저 쌍방간 협상(bilateral negotiation)에서 유틸리티 함수를 먼저 살펴보자. 두 에이전트 구매

자(b)와 판매자(s)를  $i \in (b, s)$ 로, 협상에 사용될 속성을 벡터  $X = X_1, X_2, \dots, X_n$ 로 나타내고, 각 속성은  $X_j \in [min_j, max_j] (1 \leq j \leq n)$ 의 값과, 각 속성의 중요도를 나타내는 가중치(weights)를 가진다. 에이전트  $i$ 의 각 속성

에 대한 가중치가  $\sum_{1 \leq j \leq n} w_j^i = 1$  로 정규화 될 때, 에이전트들의 유틸리티 함수를 다음과 같이 정의한다[4].

$$V^i(X) = \sum_{1 \leq j \leq n} w_j^i V_j^i(X_j)$$

### 2.1 NDF(Negotiation Decision Function)

NDF[5]는 하나의 속성(single issue)을 사용하는 협상 모델로서 협상 프로토콜은 교차적(alternative) 제안 프로토콜[6]을 사용한다. 협상 프로토콜에서 에이전트  $i \in (b, s)$ 와, 속성 $[j]$ , 그리고 속성 $[j]$ 에 대한 값은  $[min_j^i, max_j^i]$ 으로 두 에이전트  $b, s$ 가 수용할 수 있는 값의 범위를 나타낸다. 두 에이전트가 수용할 수 있는 영역,  $[min_j^b, max_j^b] \cap [min_j^s, max_j^s]$ 이 이들 두 에이전트의 협상 영역(zone of agreement)이 된다.  $t_{max}^i$ 를 에이전트  $i$ 의 협상 한계 시간이라 하고,  $x_{b \rightarrow s}^i[j]$ 를 시간  $t$ 에서 에이전트  $b$ 가  $s$ 에게 제시한 속성 $[j]$ 의 값이라고 하면, 시간  $t$ 에서 에이전트  $s$ 가 취할 액션(action)은 다음과 같다.

$$A^s(t, x_{b \rightarrow s}^i) = \begin{cases} \text{reject} & \text{if } t > t_{max}^s \\ \text{accept} & \text{if } V^s(x_{b \rightarrow s}^i) \geq V^s(x_{s \rightarrow b}^i) \\ x_{s \rightarrow b}^i & \text{otherwise} \end{cases}$$

여기서  $t_{max}^s$ 는 에이전트  $s$ 에게 주어진 최대의 협상시간을 나타내며,  $x_{s \rightarrow b}^i$ 는 에이전트  $s$ 가 에이전트  $b$ 의 제안 값을 평가한 결과 만족하지 못할 때 새로 제안하게 될 값으로 다음에 오는 방법으로 생성하게 된다.

### 2.2 하나의 속성을 사용한 Offer의 생성

[5]에서 두 에이전트 간의 제안 값을 생성하기 위해 사용한 방법은 남아 있는 협상 시간에 따라 속성의 값을 바꾸는 방법을 사용하고 있다. 결국 다음 협상 시점에서 제안하게 될 값을 결정하는데 사용된 중요한 요인은 시간( $t$ )으로, 남아 있는 협상 시간에 따라 속성들의 값을 바꾸게 되고, 위에서 정의한 에이전트들의 협상 최대 시간  $t_{max}^i$ 으로 모델화되어 지는데, 에이전트  $b$ 에 의해 제안될 값은 각 속성이 가진 최대값에서 최소값을 뺀 구간( $max_j^b - min_j^b$ )에  $a_j^b$ 를 곱하여 정의한다. 모델의 정의를 보면, 시간  $t (0 \leq t \leq t_{max}^b)$ 에서 속성 $[j]$ 에 대하여 에이전트  $b$ 가 에이전트  $s$ 에게 제시할 값은 시간에 따른 함수  $a_j^b$ 로 표현된다.

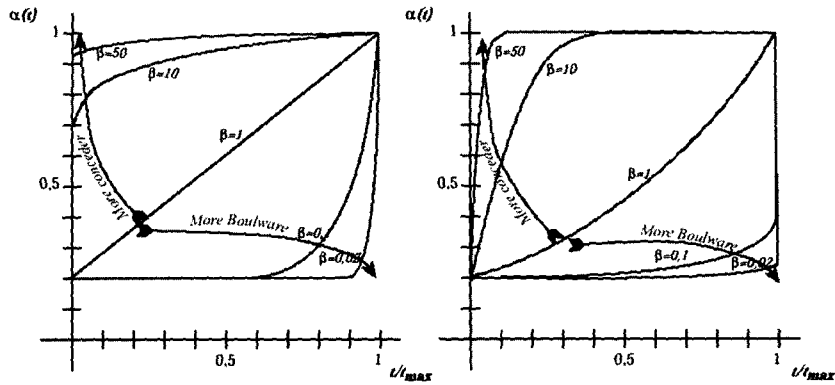


그림 1  $\alpha(t)$  값 변경과 Polynomial(왼쪽), Exponential(오른쪽) 함수

$$x_{b \rightarrow j}^t[j] = \begin{cases} \min_j^b + \alpha_j^b(t)(\max_j^b - \min_j^b) & (\text{if } V_j^b \text{ is decreasing}) \\ \min_j^b + (1 - \alpha_j^b(t))(\max_j^b - \min_j^b) & (\text{if } V_j^b \text{ is increasing}) \end{cases}$$

각 에이전트들이 제안하게 될 값은  $\alpha_j^b(t)$ 의 값에 따라 바뀌게 되고,  $\alpha$ 의 값은 반드시  $0 \leq \alpha_j^b(t) \leq 1$ 사이,  $\alpha_j^b(0) = k_j^b$ ,  $\alpha_j^b(t_{max}^b) = 1$ 이며,  $\alpha_j^b(t)$ 은 다음과 같은 두 가지의 함수로 정의 된다.

Polynomial .  $\alpha_j^b(t) = k_j^b + (1 - k_j^b) \left( \frac{\min(t, t_{max}^b)}{t_{max}^b} \right)^\beta$

Exponential .  $\alpha_j^b(t) = e^{(1 - \frac{\min(t, t_{max}^b)}{t_{max}^b})^\beta \ln k_j^b}$

여기서  $k_j^b$ 는 에이전트  $b$ 의 속성  $j$ 에 대한 초기 값을 결정하기 위한 상수이며, 이들 함수에 대한  $\beta$ 의 값이 무한하기 때문에 명확하게 양보의 형태에서 차이를 보이는 방법으로 세 가지 형태를 분류하고 있다.

(i) Boulware 협상 방법 :  $\beta < 1$ 의 값을 가진 exponential 또는 polynomial 함수로써, 초기 제안한 값 또는 초기값과 비슷한 값이  $t_{max}$ 이전까지 지속되며, 최대 시간( $t_{max}$ )에서 에이전트 자신의 최대 수용 값(reservation value)까지 양보한다.

(ii) Conceder 협상 방법 :  $\beta > 1$ 의 값을 가진 exponential 또는 polynomial 함수로써, 시간이 경과 할수록 제안될 값이 빠르게 최대 수용 값에 도달하게 되고, 그 이후 같은 값 또는 비슷한 값이  $t_{max}$ 까지 지속된다.  $\beta = 1$ 이면 선형적으로 증가된다.

2.3 다중 속성을 이용한 Offer의 생성

다중 속성을 사용하여 협상을 진행하는 방법은 하나의 속성만을 사용하는 협상 방법과는 다른 제안 생성 방법을 사용하게 되는데, [3]에서는 각 속성들의 최대 값과 최소 값 사이의 구간을 정하여 다음 협상을 위한 제안 값들의 생성을 정의하고 있다. 에이전트  $b$ 의 각 속

성  $[j]$ 에 대한 값은

$$X_j^b \in \{X_{ji}^b \mid 1 \leq i \leq I_j^b\}$$

으로 구간내의 값을 가지며, 여기서  $I_j^b$ 를 에이전트  $b$ 의 속성  $[j]$ 에 대한 구간 값이라 하면, 다중속성을 가진 에이전트  $b$ 는  $M$ 개의  $J$ -튜플(tuple)로 이루어진 협상을 위한 제안 값들을 얻을 수 있다.

$$M^b = \prod_{j=1}^J I_j^b = I_1^b \times I_2^b \times \dots \times I_J^b$$

그러나 이 방법은 협상을 시작하기 전에 각 속성들에 대한 구간( $I_j^b$ )을 미리 정의하여야만 한다. 이러한 다중 속성의 특징을 이용하여 3장에서 다중 속성을 사용하여 협상을 진행하면서 다음 협상 시점에서 제안하게 될 값을 생성하는 방법을 제안한다.

3. 다중 속성(Multi-issue)을 사용하는 협상 모델

3.1 다중 속성 협상 모델 정의

다중 속성을 이용한 협상 모델 역시 교차적인 협상 프로토콜을 사용하게 되고, 에이전트  $i \in (b, s)$ 에서  $b$ 를 구매자,  $s$ 를 판매자라 하고,  $J$ 개의 다중 속성은 곱집합(Cartesian product) 영역으로 정의되며, 각 속성에 대하여 에이전트가 제안할 값( $X$ )은 다음과 같다.

$$X = X_1, X_2, \dots, X_J, X_j = \{X_j \mid X_j \in [\min_j^i, \max_j^i], 1 \leq j \leq J\}$$

본 논문에서는 두개의 속성( $X_1$ =가격,  $X_2$ =보증)을 고려하여 협상을 진행하게 되며, 에이전트들은 각각의 협상 시간을 가지고 있으며,  $X_{b \rightarrow s}^t[X_1, X_2]$ 를 시간  $t$ 에서 에이전트  $b$ 가  $s$ 에게 제시한 속성  $[X_1]$ 과  $[X_2]$ 에 대한 값으로 정의한다.

3.2 다중 속성에서의 Offer 생성 제안

다중 속성을 사용하여 협상을 진행하는 방법은 하나의 속성을 사용하는 협상 방법과는 다른 제안 생성 방법을 사용하게 되는데, 2.2절에서 두 에이전트  $b, s$ 가 각각의 속성에 대한 값( $x_{b \rightarrow s}^b[t]$ )을 바꾸는 방법을 동시에 사용하게 된다. 각 에이전트의 속성은 그 특성에 따라  $V_j^b$ 의 값이 낮아지거나 높아질 수도 있다. 예를 들어, 속성  $X_1 = Price$ ,  $X_2 = Warranty$ 라고 하면, 구매자는 가능한 낮은 가격과 오랜 기간의 보증을 원할 것이다. 이처럼 구매자와 판매자는 서로 반대의 관심을 보이기 때문에 시간  $t$ 가 진행됨에 따라 에이전트의  $b$ 의  $[X_1]$ 의 값은 증가되고,  $[X_2]$ 의 값은 감소된다. 시간  $t$ 에서 에이전트  $b$ 가  $s$ 에게 제시할 속성 값 $[X_1, X_2]$ 은 다음과 같다.

$$X_{b \rightarrow s}^b[X_1, X_2] = \left\{ \begin{array}{l} X_1 = \min_{x_1}^b + \alpha_{x_1}^b(t)(\max_{x_1}^b - \min_{x_1}^b) \quad (\text{for price}) \\ X_2 = \min_{x_2}^b + (1 - \alpha_{x_2}^b(t))(\max_{x_2}^b - \min_{x_2}^b) \quad (\text{for warranty}) \end{array} \right\}$$

에이전트  $s$ 에 대해서는 위 식의 반대 적용으로 속성들의 값을 정의할 수 있으며, 중요한 점은 속성들에 대한 값을 바꾸는 방법이다. 즉, 각 에이전트가 협상 중 동시에 두 속성의 값을 변경 할 것인지, 아니면 두 속성 중 하나만 변경 할 것인지를 결정하는 문제와, 어느 속성의 값을 먼저 변경 할 것인지를 고려하여야 하는데, 본 논문에서는 두 가지의 방법을 제시한다.

첫번째 방법은, 교차적 속성 값 변경 방법으로 시간  $t$ 에서 두 속성에 대하여 동시에 속성의 값을 변경하지 않고 두 속성 중 하나의 속성 값만을 변경하는 방법이고, 두번째 방법은 동시 속성 값 변경으로서 시간  $t$ 에서 동시에 두 속성의 값을 변경하는 방법이다.

(i) 교차적 속성 값 변경 : 이 방법은 시간  $t$ 에서 두 속성 중 어느 하나의 속성만을 변경하는 방법이다. 두 에이전트는 각각의 협상 제한 시간과, 각 속성에 대한 가중치를 가지고 있으므로, 이 방법은 상당히 현실적인 방법이다. 왜냐하면 구매자와 판매자는 각 속성에 대하여 서로 반대의 관심을 보여주기 때문에 자신에게 큰 이익을 주는 속성의 값은 가능하면 바꾸지 않으려 할 것이고, 결국 두 에이전트는 가중치가 낮은 속성의 값을 먼저 변경하여 상대방에게 제시하게 된다. 시간( $t=1$ )에서 에이전트  $b$ 가  $s$ 에게 각 속성  $X_1$ 과  $X_2$ 의 값을 제시한 후, 다음 시간( $t=2$ )에서 에이전트  $b$ 가 에이전트  $s$ 에게 다시 제안할 값은 다음과 같은 방법으로 모델 되어진다.

$$X_{b \rightarrow s}^b[X_1, X_2] = \begin{cases} X_1 = X_1^{t-1}, & (n \geq 2) \\ X_2 = \min_{x_2}^b + (1 - \alpha_{x_2}^b(t))(\max_{x_2}^b - \min_{x_2}^b) & \text{if } X_1\text{'s weights} > X_2\text{'s weights} \end{cases}$$

여기서  $t_{n-1}$ 는 현재 시간( $t_n$ )에서 한 단계 이전 시간에 제시한 속성  $X_1$ 에 대한 값을 나타낸다. 이 방법은 시간

이 경과함에 따라 속성  $X_1$ 의 값과  $X_2$ 의 값이 서로 바뀌는 방법으로 제시 되어지고, 또한 각 협상 단계마다 시간( $t$ )이 경과함에 따라 생성되는 제안의 수( $N$ )를 다음과 같이 표현할 수 있다.

$$N_{offer}^t = \begin{cases} n-k+1 & (\text{if } n=1,3,5,\dots,t_{max}^b, \text{ then } k=1,2,3,\dots,n) \text{ only } n \text{ is odd} \\ n-1 & \text{Not change if } n \text{ is even} \end{cases}$$

여기서,  $N$ 은 시간  $t$ 에서 생성된 제안의 수이며,  $n$ 은 현재 시간( $t=1$ )을 한 단계로 보면  $n$ 이 홀수일 때 제안의 수가 바뀌게 되고, 짝수일 때는 바뀌지 않게 된다.

(ii) 동시 속성 값 변경 : 이 방법은 하나의 속성만을 사용하여 속성의 값을 바꾸는 방법과 유사하다. 왜냐하면 시간이 경과함에 따라 두 속성의 값을 동시에 변경하기 때문이며, 또한 생성되는 제안의 수도 다음과 같이 각 단계마다 다른 수( $N$ )의 제안을 생성하게 된다.

$$N_{offer}^t = 2n - 1 \quad (n = 1,2,3,\dots, t_{max}^b)$$

**3.3 속성 값 변경 전략과 협상 프로토콜**

협상 전략은 무엇보다도 가능한 한 자신에게 높은 이익( $V^i=1$ )이 되는 전략을 선택하게 될 것이다. 그러나 저속적으로 자신에게 유리한 전략을 택한다면 합의에 도달할 수 없으므로, 협상이 성사가 되지 않았을 경우 서로가 자신의 이익을 낮추어 양보를 하여야만 한다. 양보의 형태는 크게 3가지의 형태(Boulware, Conceder, Linear)로 구분하여 제시하고 있다. 하나의 속성을 사용할 경우 시간( $t$ )에서 거래가 성사되지 않으면, 다음 시간( $t'$ )에는 반드시 속성에 대하여 서로 양보를 하여 협상을 계속한다. 이처럼 하나의 속성을 사용하는 협상에서 협상을 위한 대상(가격)이 하나만 존재 하기 때문에 반드시 속성에 대한 값을 높이든지(buyer) 아니면 낮춤(seller)으로써 양보를 하게 된다. 그러나 여러 개의 속성을 사용할 경우 3.2절에서 언급한 것처럼 다음 협상을 위한 값을 제시할 때 양보를 하게 되지만 반드시 모든 속성들에 대해 양보를 하는 것은 아니다.

**3.3.1 교차적 속성 값 변경 전략(AS)과 협상 프로토콜**

각 에이전트는 자신들의 이익을 가능한 한 유지하기 위해 속성에 대한 값을 바꾸지 않으려 하지만, 협상을 계속 진행하기 위해서는 양보를 하여야 한다. 이때 각 속성들에 대한 값을 바꾸어야 하며, 또한 어느 속성 값을 먼저 바꿀 것인지를 결정 한 후 에이전트는 시간  $t$ 에서 자신의 제안 값을 상대방에게 제시하게 되고, 이 제안된 값이 상대방에 의하여 거절(reject)되면 다음 협상 단계에서 제안될 값은 이전에 양보를 한 속성의 값은 그대로 유지하고, 나머지 속성에 대해 양보를 하는 방법을 취한다. 표 1은 각 협상 단계에서 교차적 속성 값 변경 전략을 통한 속성들의 값 변경과 제시 가능한

표 1 교차적 속성 값 변경 전략을 통한 값 변경과 선택 가능한 제안의 수

$X_{b \rightarrow s}^t$ [ $X_1, X_2$ ]	n=1	n=2	n=3	n=4	n=5	...	$n=t_{max}^b$
[ $X_1$ ]	150	150	160	160	170	...	$max^b[X_1]$
[ $X_2$ ]	15	14	14	13	13	...	$min^b[X_2]$
생성된 제안(수)	[150,15] (1)	[150,14] (1)	[160,15] [160,14] (2)	[150,13] [160,13] (2)	[170,15] [170,14] [170,13] (3)	...	... (n-k+1)개 또는 ( $t_{n-1}$ )개

제안의 수를 나타낸 것이다.

위의 표에서처럼  $b$ 가 먼저 시간  $t_1$ 에서 협상을 시작하면, 초기 제안 값  $X_{b \rightarrow s}^1[X_1, X_2]$ 를 에이전트  $s$ 에게 제시하고,  $s$ 는  $b$ 가 제안한 각 속성들의 값이 자신의 협상 구간 내에 존재하는지를 먼저 비교하여 존재하지 않으면  $b$ 의 제안을 거절하고, 그렇지 않으면  $b$ 가 제안한 값을 평가하여  $V^s[X_1, X_2]_{b \rightarrow s}^1 \geq V^s[X_1, X_2]_{s \rightarrow b}^1$ 이면 수락(accept), 그렇지 않으면 거절하고,  $s$ 는  $b$ 에게  $X_{s \rightarrow b}^1[X_1, X_2]$ 의 대응 제안(counter offer)을 한다. 시간  $t_2$ 에서  $b$ 는  $s$ 가 제안한 값이  $V^b[X_1, X_2]_{s \rightarrow b}^1 < V^b[X_1, X_2]_{b \rightarrow s}^1$  인지를 평가한다.  $t_2$ 에서의 제안 가능한 속성들의 값은  $t_1$ 부터 현재 시간까지의 속성 값들의 조합으로 만들어지며, 현재 시간 이전에 생성 또는 상대방에게 제시된 적이 있는 제안은 제외된다. 만약  $t_2$ 에서  $V^b[X_1, X_2]_{s \rightarrow b}^1 < V^b[X_1, X_2]_{b \rightarrow s}^1$  이면,  $b$ 가  $s$ 에게 제시할 대응 제안은 여러 가능한 제안들 중 가장 유틸리티 값이 낮은 것을 제안함으로써 양보를 하게 된다. 이는 시간 제약 환경에서 협상을 진행하기 때문에 가장 낮은 값을 갖는 제안을 선택하였다. 시간  $t_n$ 에서 새로이 생성된 가능한 제안의 수는  $n=1,3,5, \dots, t_{max}^b$ 일 때  $k=1,2,3, \dots, n$ 이면, 즉,  $n$ 이 홀수일 때만 제안의 수가 바뀌고,  $n$ 이 짝수이면 현재 시간( $t_n$ )의 한 단계 이전 시간( $t_{n-1}$ ) 때의 수를 유지한다.

$$AS \ X_{b \rightarrow s}^{t+1} = \begin{cases} \text{quit, if } t > t_{max}^b \\ \text{reject, when offer not exist within acceptable domain or,} \\ V^b[X_1, X_2]_{s \rightarrow b}^t < V^b[X_1, X_2]_{b \rightarrow s}^{t+1} \\ \text{accept, when offer exists within acceptable domain and,} \\ V^b[X_1, X_2]_{s \rightarrow b}^t \geq \text{all possible } (n-k+1) \ V^b[X_1, X_2]_{b \rightarrow s}^{t+1} \\ X_{s \rightarrow b}^{t+1}(X_1, X_2) \ \text{otherwise propose one of the lowest } b\text{'s offers} \end{cases}$$

3.3.2 동시 속성 값 변경 전략(SS)과 협상 프로토콜

동시에 속성의 값을 변경하는 전략은, 위에서 보여준 교차적 속성 값 변경 협상 프로토콜과 유사하지만, 다음 협상 시간에 제시할 수 있는 제안의 생성이 다르다. 즉, 매 시간 마다 생성되는 제안의 수는  $2n-1$ 개가 생성이 된다. 3.3.1절에서처럼 제안 값의 생성은 이전 속성들의 값과 현재 변경된 속성 값들과의 조합으로 생성되고, 양

보의 형태는 위와 동일하다.

표 2 동시 속성 값 변경 전략을 통한 값 변경과 선택 가능한 제안의 수

$X_{b \rightarrow s}^t$ , [ $X_1, X_2$ ]	n=1	n=2	n=3	...	$n=t_{max}^b$
[ $X_1$ ]	150	160	170	...	$max^b[X_1]$
[ $X_2$ ]	15	14	13	...	$min^b[X_2]$
생성된 제안(수)	[150,15] (1)	[150,14] [160,15] [160,14] (3)	[150,13] [160,13] [170,15] [170,14] [170,13] (5)	...	... (2n-1)개

$$SS \ X_{b \rightarrow s}^{t+1} = \begin{cases} \text{quit, if } t > t_{max}^b \\ \text{reject, when offer not exist within acceptable domain or,} \\ V^b[X_1, X_2]_{s \rightarrow b}^t < V^b[X_1, X_2]_{b \rightarrow s}^{t+1} \\ \text{accept, when offer exists within acceptable domain and,} \\ V^b[X_1, X_2]_{s \rightarrow b}^t \geq \text{all possible } (2n-1) \ V^b[X_1, X_2]_{b \rightarrow s}^{t+1} \\ X_{s \rightarrow b}^{t+1}(X_1, X_2) \ \text{otherwise propose one of the lowest } b\text{'s offers} \end{cases}$$

3.4 각 속성들의 값 변경 과정

이 절에서는 각 속성들의 값이 어떻게 변경되는지 그 과정을 보여준다. 속성들의 값을 바꾸기 위한 환경변수는 각 에이전트의 속성  $[j]_{1 \leq j \leq n}$ 에 대하여  $[t_{max}^i, k_{X_j}^i, \min_{X_j}^i, \max_{X_j}^i]$ 으로 속성의 수가 늘어날수록 변수는 증가 하게 된다. 또한 하나 이상의 속성을 사용할 경우 각 속성에 대해서 서로 다른  $\beta$  값을 설정하는 2가지 방법을 제안한다. 첫 번째 방법은, 각 속성들에 대해 서로 다른  $\beta$  값을 갖게 하는 방법으로 각 속성들이 가지고 있는 가중치를  $\beta$  값으로 사용한다. 여러 개의 속성을 사용하는 협상 환경에서 각 속성들은 가중치를 가지고 있으며, 중요한 속성 일수록 큰 값을 가지고 있다. 다시 말하면 3.3절에서 언급 하였듯이 두 에이전트는 자신에게 유리한 전략을 취하기 때문에 가중치가 큰 속성의 값은 바꾸지 않으려 할 것이다. 각 속성  $X_1$ 과  $X_2$ 에 대해 아래와

같은  $\beta$  값을 가지게 되며, 만약 각 속성의 가중치가  $X_1 > X_2$  라면,  $X_1$ 에 대한  $\beta$  값은  $X_2$ 의  $\beta$  값보다 더 작은 값을 가지게 되고, 속성  $X_2$  보다 양보의 속도가 늦어지게 된다.

$$\beta^i = \begin{cases} X_1 = 1 - \text{issue } X_1\text{'s Weights} \\ X_2 = 1 - \text{issue } X_2\text{'s Weights} \end{cases}$$

두번째는, 각 속성들에 대한  $\beta$  값을 동적인 값을 갖게 하는 것이다. 즉, 시간 제약 환경에서 협상 요인으로 남은 시간을 사용하므로 시간이 경과 할수록 에이전트의 최대 수용 값에 도달하게 된다. 여기에 동적인  $\beta$  값을 사용하여 속성들의 값을 변경하는 방법으로, 에이전트  $i$ 의  $\beta$  값은 현재까지 상대 에이전트와 진행한 협상의 총 횟수(round)를  $\beta$  값으로 사용한다.

$$\beta^i = \sum R_{b \leftrightarrow s}^i$$

여기서  $\sum R_{b \leftrightarrow s}^i$ 는 에이전트  $b$ 와 에이전트  $s$ 사이 현재 시간까지 진행된 협상 총 횟수이다. 이 방법은 초기에 각 속성들의 값이 큰 쪽으로 변경이 이루어지게 되고, 시간이 경과할수록  $\beta$  값은 커지지만 협상 시간이 줄어들기 때문에 속성들의 값은 낮은 쪽으로 변경하게 된다.

각 에이전트의 행동을 보기 위해 서로 다른 협상 제한 시간을 가지고 교차적 속성 값 변경 전략(AS)과, 동시 속성 값 변경 전략(SS)을 이용한 속성들의 값을 변경하는 과정,  $\beta$  값으로 각 속성들의 가중치 사용과, 매 협상에서 현재 시간까지의 협상 횟수를 사용 하였을 때, 속성들의 값이 바뀌게 되는 과정을 그림 2부터 그림 6까지 보여주고 있으며, 각 속성에 대한 상수 값  $K_j^i = 0.1$ 로 설정하였다.

그림 2부터 그림 6까지의 그림 중 (a),(b)는 속성  $X_1$ 에 대해 제안한 두 가지 변경 방법(AS,SS)을, 그림(c)는 교차적 속성 값 변경 전략(AS)을 사용하여  $t_{max}$ 에 도달하였을 때 속성  $X_2$ 의 값이 최대 수용 값에 도달하는 과정을 나타낸 것이다. 교차적 속성 값 변경 전략(AS)은  $t_{max}$ 에 따라 두 속성 중 하나만 최대 수용 값에 도달하게 되고, 동시 속성 값 변경 전략(SS)은  $t_{max}$ 에 도달했을 때 두 속성의 값 모두 최대 수용 값에 도달하게 된다. 그림에서 동시 속성 값 변경 전략(SS)을 사용하였을 때 속성  $X_2$ 에 대한 변경 과정은 생략하였다.

그림 2는 시간이 경과함에 따라 자신에게 유리한 값을  $t_{max}$ 이전까지 유지하면서 거의 양보를 하지 않으며, 그림 3은 시간이 경과함에 따라 선형적으로 각 속성들의 값이 변하기 때문에 각 에이전트 자신의 최대 수용

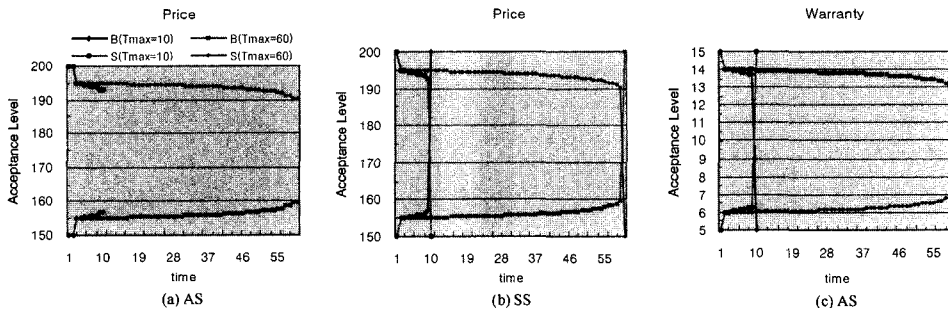


그림 2 Boulware,  $\beta=0.1$ ,  $\min^i[X_1]=150$ ,  $\max^i[X_1]=200$ ,  $\min^i[X_2]=5$ ,  $\max^i[X_2]=15$ ,  $t_{max}^i=10, 60$

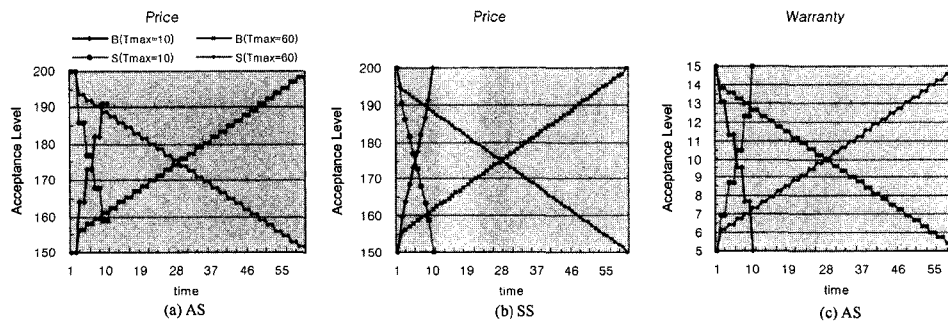


그림 3 Linear,  $\beta=1$

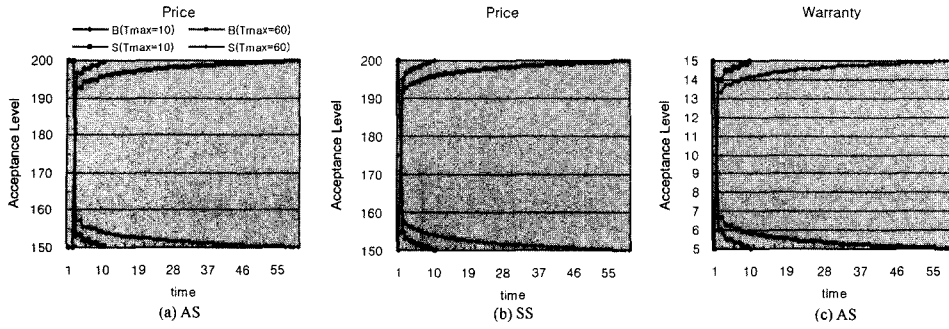


그림 4 Conceder,  $\beta=20$

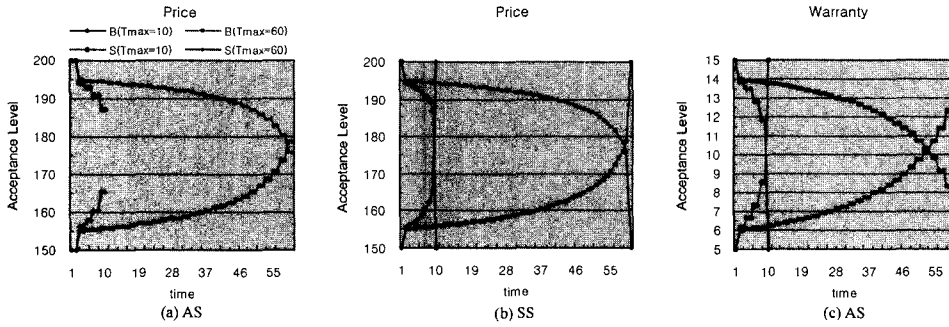


그림 5 Boulware,  $\beta=1-X_1^s$  (or  $X_2^s$ ) weights

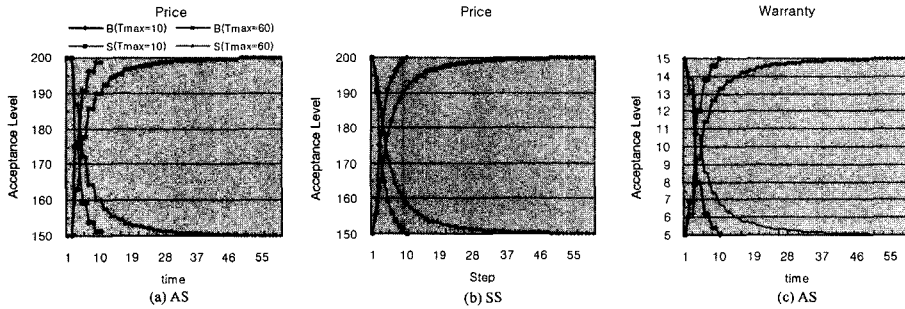


그림 6 Conceder,  $\beta = \sum R_{b+1}^k$

값까지 양보를 하는 형태를 보인다. 그림 4에서는  $\beta$  값이 크기 때문에 각 에이전트의 두번째에 제안될 값은 큰 쪽으로 양보를 하게 되고, 시간이 지남에 따라 양보의 폭이 작아지게 된다. 그림 5는  $\beta$  값으로 자신의 기중치를 사용한 경우이며, 이 경우 시간이 경과함에 따라 초기에는 작은 양보의 폭으로 속성들의 값을 바꾸게 되고,  $t_{max}$ 에 가까울수록 양보의 폭이 커진다. 그림 6은  $\beta$  값이 협상 진행에 따라 바뀌는 정수형의 값을 갖는 경우이다. 그림 6은 그림 5와는 반대로 시간이 경과 할수록 양보의 폭이 작아지는 형태를 보여준다.

#### 4. 실험 및 결과

##### 4.1 실험

실험의 환경은 3.4절에서의 각 변수들에 대하여 200개의 환경을 생성하여 협상을 실시 하였으며, 두 에이전트의 최대 협상 시간은  $t_{max} \in [30, 60]$  사이,  $t_{max}^b \geq t_{max}^s$ ,  $t_{max}^b < t_{max}^s$ 으로,  $k_j^i$ 의 값은 0.1로 설정하였으며 에이전트  $b$ 가 먼저 협상을 진행한다. 실험을 위한 협상 방법과  $\beta$  값의 범위는 그림 7과 같으며, 각각의 양보 형태에 대하여 두 가지의 속성 값 변경 전략(AS,SS)을 적용하여 각 환경에 대하여 실험하였다. 에이전트  $b$ 가 각 환경( $e$ )

기존양보형태	제안한 양보형태	$\beta$ 값의 범위	Strategy
Boulware	BASW, BSSW	1-weights	각 양보형태에 대하여 AS와 SS 제안 방법 적용
	BAS, BSS	$\beta \in [0.01, 0.2]$ $\beta \in [0.25, 0.4]$ $\beta \in [0.4, 0.7]$ $\beta \in [0.7, 0.99]$	
Conceder	CAS, CSS	$\beta \in [20, 40]$	
	CASR, CSSR	$\beta = \sum R_{b \rightarrow s}^c$	
Linear	LAS, LSS	$\beta = 1$	

- BA(S)SW : Boulware Alternative (Simultaneous) Strategy using Weights.
- CA(S)SR : Conceder Alternative (Simultaneous) Strategy using Round.
- LA(S)S : Linear Alternative (Simultaneous) Strategy.

그림 7  $\beta$ 값의 범위

에 대하여, 하나의 협상 방법  $l$ 를 선택했을 때, 에이전트  $s$ 는 에이전트  $b$ 와 동일한 형태의  $m$ 개의 협상 방법을 가지고 협상을 진행한다. 왜냐하면 서로 다른 협상 방법을 사용할 경우 각 속성 값에 대한 양보의 형태가 다르기 때문이다. 예를 들어, 에이전트  $b$ 는 boulware 방법을, 에이전트  $s$ 는 conceder 방법을 사용할 경우  $b$ 가  $s$ 보다 월등히 높은 유틸리티 값을 가지게 되며, 협상 성사율도 높아지기 때문이다.

4.2 결과

실험결과는 각 환경에 대하여 에이전트의 각 협상 방법별 결과에 대한 평균 유틸리티와, 평균 협상 성사비율을 계산하였다.  $e$ 개의 환경에 대하여 에이전트  $b$ 가  $l$ 형태의 협상 방법을 사용하고, 에이전트  $s$ 는 에이전트  $b$ 의 협상 방법과 같은  $m$ 개의 모든 협상 방법을 사용해 협상이 이루어진 평균 유틸리티를 계산하였다.

$$U_i^e [l, m] = \frac{\sum_{s=1}^n U_i^s [l, m]}{\text{The numbers of dealed}}$$

그림 8은  $e$ 개의 환경에 대하여, 그림(a)는 에이전트들이  $\beta$ 값으로 각 속성의 가중치를 가지고 boulware 협상 방법을 사용하여 협상한 결과에 대한 평균 유틸리티를

나타낸 것이다. 에이전트  $b$ 의 경우, AS를 선택 하였을 때 동시 SS를 선택 할 때보다 평균 유틸리티가 더 높은 값을 보여준다. 이는 3.2절에서 설명한 것처럼 자신에게 큰 이익을 주는 속성의 값은 가능한 바꾸지 않으려 하기 때문이다. 그림(b)는 협상이 이루어진 비율을 나타낸 것으로 협상이 이루어진 비율이 낮으며, 두 에이전트 모두 SS를 사용한 경우 협상이 이루어질 확률이 더 높은 것으로 나타났다. 이는 두 에이전트 모두 속성의 값을 매 협상 단계마다 바꾸어 서로 양보를 하기 때문이다.

그림 9는 conceder 협상 방법을 사용한 경우를 나타낸 것으로, 그림(a)에서 에이전트  $b$ 는 SS를, 에이전트  $s$ 는 AS를 사용할 경우, 에이전트  $b$ 의 평균 유틸리티가 가장 낮으며, 반대로 에이전트  $s$ 는 이 경우 평균 유틸리티가 가장 높다. 이는  $\beta$ 의 값이 클수록 다음 협상시간에 속성의 값이 큰 쪽으로 바뀌기 때문이다. 또한 두 에이전트의 평균 유틸리티가 비슷한 경우는 에이전트  $b$ 가 AS를, 에이전트  $s$ 는 SS를 사용할 경우 두 에이전트의 평균 유틸리티가 비슷한 값을 가지며, 에이전트  $b$ 의 경우 이때 가장 높은 유틸리티 값을 가지며, 이 경우를 제외한 나머지의 경우 두 에이전트 사이의 유틸리티 값의

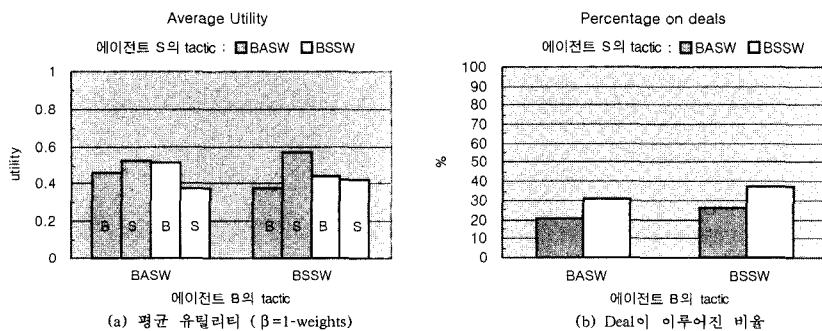


그림 8 BA(S)SW를 사용했을 때 평균 유틸리티와 협상 성사비율



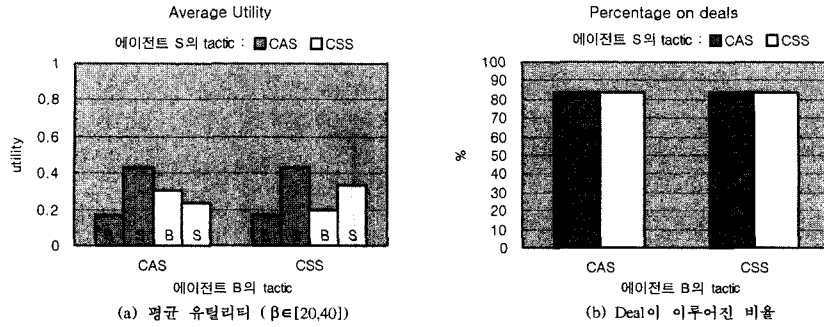


그림 9 CA(S)S를 사용했을 때 평균 유틸리티와 협상 성사비율

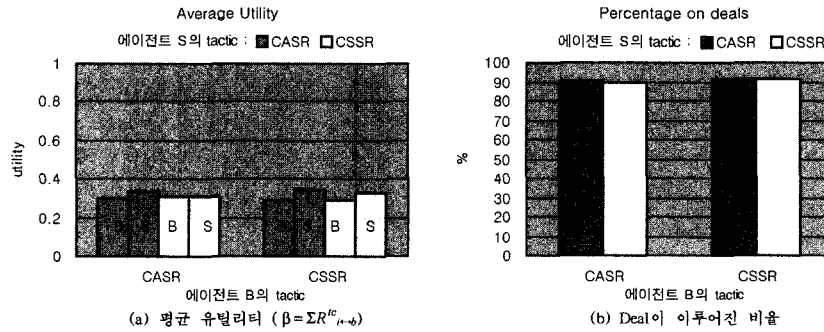


그림 10 CA(S)SR를 사용했을 때 평균 유틸리티와 협상 성사비율

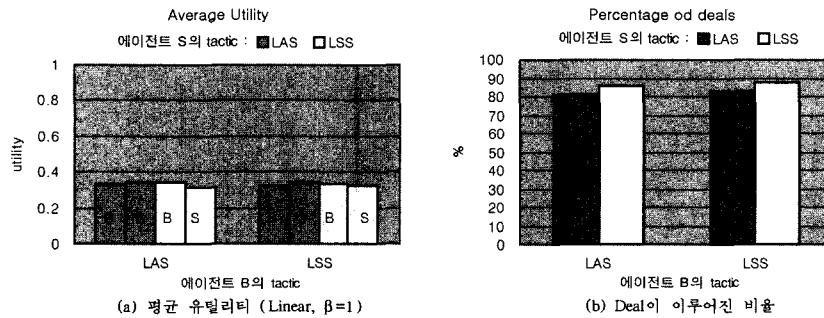


그림 11 LA(S)S를 사용했을 때 평균 유틸리티와 협상 성사비율

차이가 크고, 평균적으로 에이전트 s가 에이전트 b에 비해 높은 유틸리티 값을 갖는다. conceder 협상 방법을 사용할 경우 협상이 이루어질 비율은 그림(b)에서처럼 상당히 높게 나타났다. conceder 협상 방법은  $\beta$  값이 클수록 속성의 값은 큰 폭으로 바뀌므로, 두 에이전트 사이 유틸리티 값의 차이가 커지게 되고,  $\beta$  값이 작을수록 이 차이가 작아진다. 또한  $\beta$  값이 커질수록 각 에이전트들의 유틸리티 값이 작아진다.

그림 10은  $\beta$  값을 매 협상 때 마다 동적인  $\beta$  값( $\beta \geq 1$ )을 사용한 결과이다. conceder 협상 방법을 사용하였

을 때 두 에이전트의 유틸리티 값 차이가 크게 나는 것(그림 10)과는 달리, 그림 11에서는 두 에이전트 사이 유틸리티 값의 차이가 줄었다. 동적인  $\beta$  값을 사용한 협상은 일반적으로 어느 한 쪽이 큰 이익을 얻는 것 보다는 가능한 두 에이전트 모두가 균형 있는 이익을 얻을 수 있도록 한다. 이는 협상 초기에 거래 성사가 이루어지지 않으면 시간이 경과 할수록 양보의 폭이 작아지기 때문에 서로의 이익을 최대화 할 수 있기 때문이다. 그림(a)에서 두 에이전트 사이의 유틸리티 값의 차이가 그림 10의 (a)보다 작은 것을 볼 수 있다. 또한 그림(b)에

표 3 BA(S)S를 사용했을 때 평균 유틸리티와 협상 성사 비율( $\beta < 1$ )

$\beta$ 값 범위	S의 전략 B의 전략	BAS			BSS		
		Utility		성사율(%)	Utility		성사율(%)
		Buyer	Seller		Buyer	Seller	
[0.01,0.2]	BAS	0.3308	0.7970	7.5	0.5530	0.4372	12.5
	BSS	0.2968	0.7633	8.5	0.4606	0.4899	13.0
[0.25,0.4]	BAS	0.4422	0.5570	15.5	0.5347	0.3870	23.5
	BSS	0.3328	0.6055	19.5	0.4362	0.4413	28.5
[0.4,0.7]	BAS	0.4099	0.4628	34.5	0.4566	0.3688	41.5
	BSS	0.3678	0.4813	36.5	0.4136	0.3806	44.5
[0.7,0.99]	BAS	0.4087	0.4210	42.5	0.4365	0.3595	48.5
	BSS	0.3647	0.4382	47.0	0.3956	0.3719	53.0

서 협상이 이루어진 비율이 그림 10의 (b)보다 높게 나타난 것을 보여주고 있으며  $e$ 개의 환경 전체에 대해 높은 협상 성사율을 보여준다.

그림 11은 선형적 협상 방법을 사용한 것으로, 두 에이전트 사이의 평균 유틸리티 값의 차이는 동적인  $\beta$  값을 사용한 것보다 작고, 두 에이전트의 유틸리티 값이 더 높게 협상이 이루어진다. 그러나 선형적 협상 방법은 *boulware* 협상 방법을 제외한 나머지 협상 방법들에 비해 협상 성사 비율이 상대적으로 낮은 것으로 나타났으며, 두 에이전트 모두 다른 협상 방법들에 비해(*boulware* 협상 방법을 제외한) 가장 높은 유틸리티 값을 갖는다.

그림 8에서 각 속성들의 가중치를  $\beta$  값( $\beta < 1$ )으로 사용하여 협상한 결과를 이미 보였다. 각 속성의 가중치를  $\beta$  값으로 사용한 이유는  $\beta$  값을 결정하기 위해 제안한 것이다. 표 3은 *boulware* 협상 방법을 사용하여  $\beta < 1$  인 값을 AS와, SS으로 협상한 결과를 나타낸 것이다.  $\beta$  값의 범위는 그림 7에서 보여준 것처럼 크게 4가지의 범위로 설정하였다.

$\beta \in [0.01, 0.2]$ 의 값을 사용하였을 때의 협상 결과 두 에이전트 사이 유틸리티 값의 차이가 크게 나타났으며,  $e$ 개의 환경에 대해 협상 성사율이 아주 낮다. 왜냐하면 그림 2에서 보이는 것처럼 두 에이전트의 양보의 폭이  $\beta$  값에 의존하기 때문에 협상 시간이 지나도 에이전트들이 값을 바꾸지 않기 때문인 것을 알 수 있다.

표 3에서 보이는 것처럼 *boulware* 협상 방법은  $\beta$  값이 커질수록(1에 가까울수록) 두 에이전트의 평균 유틸리티는 비슷해지고, 협상 성사 비율은 높아지는 것을 볼 수 있다. 이는  $\beta$  값이 커질수록 에이전트는 속성들의 값을 크게 양보하게 되고, 이로 인해 협상이 이루어질 확률이 커지기 때문이다. 이전에 언급한 것처럼  $\beta$  값이 무수히 많이 존재하기 때문에  $\beta$  값을 결정하기가 쉽지 않다. 그러므로  $\beta$  값을 결정하기 위하여 그림 8에서 사용한 각 속성들의 중요도를 나타내는 가중치를 사용하는

것이 아주 낮은  $\beta$  값을 사용하는 것보다 협상이 이루어질 확률이 높다. 지금까지 보여준 여러 가지 협상 방법 중 두 에이전트의 협상 성사 비율이 가장 높은 방법은 동적인  $\beta$  값을 사용하는 방법이며, 두 에이전트 사이의 균형 있는 유틸리티 값으로 두 에이전트 사이 상호이익을 최대화 할 수 있는 방법이 선형적 협상 방법과, 동적인 *conceder* 협상 방법인 것으로 나타났다.

4.3 제안한 협상 모델 비교

4.2절에서 제안된 협상 모델의 실험 결과를 보였다. 표 4는 본 논문에서 제안한 협상 모델과 [3]에서 얻어진 협상 결과에 대한 비교를 나타낸 것이다. 비교는 제안한 협상 모델에서 사용한 동일한 각 환경들의 값을 사용하였으며, 각 에이전트의 최대 협상 시간( $t_{max}$ )을 각 속성들의 구간 값으로 사용하였다. 표 4에 나타난 것처럼 비교 모델은 두 에이전트 간의 균형 있는 유틸리티 값을 갖는 반면 협상 성사율이 낮다. 제안한 협상 모델 중 동적 *conceder* 협상 방법, 선형적 협상 방법이 두 에이전트 간의 균형 있는 유틸리티 값과 높은 협상 성사율을 보이는 것으로 나타났다.

5. 결론 및 향후 연구

본 논문에서는 향후 발전하게 될 전자상거래 분야에 서 상품의 다중 속성을 이용한 협상 모델을 제안하였다. 제안된 협상 모델들은 각 속성들의 값을 협상이 진행되는 동안 어떻게 변경하는지, 생성된 제안들 중 어느것을 제안할 것인지를 협상 형태별로 제시하였으며, 또한 이들 속성들의 값을 결정하기 위한  $\beta$  값의 범위를 구매자와 판매자의 관점에서 중요도를 나타내는 가중치를 사용하는 방법과, 동적인  $\beta$  값을 가지는 방법을 추가하여 실험을 하였다. 무수히 많이 존재하는  $\beta$  값 중 하나를 선택하는 것이 쉽지 않기 때문에, 각 속성에 대한 중요도를 나타내는 가중치를  $\beta$  값으로 사용한 것과 각 속성에 대해 서로 다른  $\beta$  값을 적용한 것 휴리스틱한 방법으로 볼 수 있다. 두 에이전트가 각각의 협상 시간을 가

지고 다중 속성에 대해 매 협상 단계마다 각 속성들의 값을 모두 변경하는 것이 아니라 각 속성들이 가지고 있는 가중치에 따라 낮은 가중치를 가진 속성들의 값을 먼저 변경하여 상대방에게 제안하는 교차적 속성 값 변경 전략과, 이와는 반대로 매 협상 단계마다 속성들의 값을 동시에 모두 변경하는 동시 속성 값 변경 전략과, 두가지 전략에 따라 각 협상 단계에서 생성되어 지는 제안 값들 중 어느것을 상대에게 제안할 것인지에 대한 서로의 이러한 속성들의 값 변경과, 생성된 제안 값 선택 전략을 이용하여 본 논문에서 각각의 협상결과를 보였다. 협상 결과는 두 가지 측면에서 평가를 실시하였으며, 또한 제안된 협상 모델의 비교를 위해 [3]을 이용한 협상 모델을 구현하여 두 모델 사이의 협상 결과를 비교하였다.

먼저 유틸리티 측면에서는  $\beta < 1$ 인 boulware 협상 방법이 높은 값을 보였으나, 협상이 이루어질 확률이 낮은 것으로 나타났다. conceder 협상 방법은 협상이 이루어질 확률이 높지만  $\beta$  값이 커질수록 유틸리티 측면에서 어느 한쪽에게만 이익을 크게 하는 불균형을 보인다. 선형적 협상 방법은 conceder 협상 방법 보다는 협상이 이루어질 확률이 다소 낮지만, boulware 협상 방법을 제외한 나머지 협상 방법에 비해 유틸리티 값이 높고, 두 에이전트 사이의 유틸리티 값의 차이가 적은 것으로

나타났다. 그리고 마지막으로 conceder 협상 방법 중 동적인  $\beta$  값을 사용한 경우, 다른 어느 협상 방법보다 협상이 이루어질 확률이 가장 높은 것으로 나타났으며, 선형적 협상 방법과 비슷한 정도의 유틸리티 값과 두 에이전트 사이의 유틸리티 값의 차이도 낮은 것으로 나타나 두 에이전트간 이익의 불균형을 해소하는 것으로 나타났다. 협상 성사율은 동적 conceder 협상 방법이 평균 91%, 선형적 협상 방법이 평균 84%의 성사율을 보여 주었다. 그리고 제안된 협상 모델과 비교 모델과의 협상 결과에서도 비교 모델은 두 에이전트 간의 균형 있는 이익을 갖지만, 협상 성사율에서 66%의 낮은 협상 성사율을 나타내었으며, 제안한 협상 모델과의 비교에서 역시 동적인  $\beta$  값을 사용한 conceder 협상 방법이 가장 높은 성사율을 보이며, 또한 두 에이전트간의 최적의 유틸리티 값을 두 에이전트 간의 이익의 합이 최대라고 했을 때 제안한 두가지 속성 값 변경 전략 중 이익의 합이 최대가 되는 것은 전체적으로 교차적 속성 값 변경 전략을 사용 하였을 때 두 에이전트의 유틸리티 값의 합이 높은 것으로 나타났다

본 논문에서 제안한 방법은 실험의 간결성을 위해 초기 속성들의 값을 결정하는 변수  $k_j^i$ 의 값을 모두 동일하게 설정하여 실험을 하였으나, 속성에 따라 서로 다른  $k_j^i$ 의 값 설정과 혼합된 속성 값 변경 전략, 그리고 교차

표 4 제안된 협상 모델과 비교 모델과의 평균 유틸리티와 협상 성사 비율

구 분	전 략	제안 모델		비교 모델			$\beta$ 값 범위			
		성사율(%)	성사율(%)	Utility		성사율(%)				
				Buyer	Seller					
Boulware 협상방법	B \ S	BASW	BSSW	0.4125	0.4146	66.0	1-weights			
	BASW	21.0	31.0							
	BSSW	26.0	37.0							
	B \ S	BAS	BSS				0.4125	0.4146	66.0	[0.01,0.2]
	BAS	7.5	12.5							
	BSS	8.5	13.0							
	BAS	15.5	23.5							
	BSS	19.5	28.5							
	BAS	34.5	41.5							
	BSS	36.5	44.5							
BAS	42.5	48.5								
BSS	47.0	53.0								
Conceder 협상방법	B \ S	CAS	CSS	0.4125	0.4146	66.0	[20,40]			
	CAS	84.0	84.0							
	CSS	84.0	84.0							
	B \ S	CASR	CSSR							
	CASR	90.5	90.0							
CSSR	91.5	91.0								
Linear 협상방법	B \ S	LAS	LSS	0.4125	0.4146	66.0	I			
	LAS	80.5	86.0							
	LSS	82.5	87.5							

적 속성 값 변경에서 1:1 비율의 속성 값 변경이 아닌 비대칭 교차적 속성 값 변경을 통한 협상 모델 연구가 필요하다.

**참 고 문 헌**

[1] Robert H. Guttman, Pattie Maes, "Cooperative vs. Competitive multi-agent negotiations in retail electronic commerce," *Proceedings of the Second International Workshop on Cooperative Information Agents*, pp.135-147, July, 1998.

[2] A. Chavez, P. Maes, "Kasbah: An Agent Marketplace for Buying and Selling Goods," *Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, April 1996.

[3] Lei Zhao, Wee-Keong Ng, Ee-Peng Lim, "Cooperative multi-attribute bilateral online negotiation for E-Commerce," *Database and Expert Systems Applications, Proceedings of the 12th International Workshop*, pp.703 - 707, 2001.

[4] Keeney, R. L. and Raiffa, H, *Decisions with Multiple Objectives*, Cambridge University Press, 1993.

[5] Peyman Faratin, Carles Sierra, Nick R. Jennings, "Negotiation Decision Functions for Autonomous Agents," *Int. Journal of Robotics and Autonomous System*. Vol 24, No. 3-4, pp.159-182. 1998.

[6] M.J. Osborne, A. Rubinstein, *A Course in Game Theory*. MIT Press, Cambridge MA, 2001.

[7] Noyda Matos Carles Sierra Nick R. Jennings, "Determining Successful Negotiation Strategies : An Evolutionary Approach," *Proceedings of the 3rd International Conference on Multi-Agent Systems*, pp.182-189, 1998.

[8] Robert H. Guttman, Alexandros G. Moukas, Pattie Maes, "Agents as mediators in electronic commerce," *International Journal of Electronic Markets*, Vol. 8, No. 1, February 1998.

[9] Robert H. Guttman, Pattie Maes, "Agent-mediated integrative negotiation for retail electronic commerce," *Proceedings of the Workshop on Agent Mediated Electronic Trading*, pp.70-90, May 1998.

[10] Carles Sierra, Penman Faratin, Nick R. Jennings, "A Service-Oriented Negotiation Model between Autonomous Agents," *Proceeding of the 8th European Workshop on Modeling Autonomous Agents in a Multi-Agent World*, pp.17-35, 1997.

[11] S. Shaheen Fatima, Michael Wooldridge, Nicholas R. Jennings, "Optimal Negotiation Strategies for Agents with Incomplete Information," *Proceeding of the 8th Int. Workshop on Agent Theories, Architectures and Languages*, pp.53-68, 2001.



**김 현 식**  
 1992년 금오공과대학 전자공학과(학사)  
 2003년 연세대학교 컴퓨터과학과(석사)  
 2003년~현재 해군 소령. 관심분야는 전자상거래, 소프트웨어 에이전트



**양 성 봉**  
 1981년 연세대학교 공학사. 1984년 Univ. of Oklahoma 컴퓨터과학 석사  
 1992년 Univ. of Oklahoma 컴퓨터과학 박사. 1993년~1994년 전주대학교 전자계산학과 전임강사. 1994년~현재 연세대학교 컴퓨터산업공학부 부교수. 관심분야는 전자상거래, 그래픽스, 인터넷 컴퓨팅