

탐색(Exploration)과 이용(Exploitation)의 상반관계의 균형에 관한 연구

(Balancing the Tradeoffs Between Exploration and Exploitation)

박 선 주 [†]

(Sunju Park)

요 약 전자상거래상에서 경매가 활발해짐에 따라 경매용 에이전트와 경매 에이전트의 비딩 스트레티지 개발에 관한 연구가 중요한 관심의 초점이 되고 있다. 특히, 미세한 스트레티지가 알려져 있지 않은 복잡한 경매 환경에서의 에이전트 스트레티지 개발은 실용적인 의미를 가지고 있다. 이 논문은 최적의 스트레티지가 존재하지 않는 연속이중경매(Continuous Double Auction, CDA) 환경에서 사용할 수 있는 “적응성 스트레티지”를 소개한다. 적응성 스트레티지는 현재 알려져 있는 P-스트레티지에 실시간 적응력을 추가하는 것을 주 아이디어로 한다. 적응성 스트레티지는 여러 종류의 알려진 스트레티지들 중 이제까지 좋은 성능을 보여준 스트레티지를 계속 사용하려는 탐색(exploration)과 바뀌어졌을지도 모르는 새로운 환경에 적합한 스트레티지를 찾아내려는 이용(exploitation)간의 균형을 피하며, 이를 각 스트레티지의 기대이득과 실행횟수사이의 상반관계를 고려하는 휴리스틱 탐색 함수를 이용하여 결정한다. 실험분석의 결과, 적응성 스트레티지는 (1) P-스트레티지가 잘 작동하지 않는 환경에선 P-스트레티지보다 높은 이득을, (2) P-스트레티지가 다른 종류의 단순한 스트레티지를 앞서는 환경에서는 P-스트레티지와 비슷한 이득을 보인다.

키워드 : 경매, 연속이중경매, 경매 에이전트, 비딩 스트레티지, 에이전트 개발

Abstract As auctions become popular, developing good agent bidding strategies has been an important focus in agent-based electronic commerce research. Especially for the continuous double auctions where no single dominant strategy is known, the agent bidding strategy has practical significance. This paper introduces an adaptive agent strategy for the continuous double auction. The central idea is to let the agent figure out at run time when the sophisticated strategy (called the p-strategy) is beneficial and when a simpler strategy is better. Balance between exploration and exploitation is achieved by using a heuristic exploration function that trades off the expected profits and the number of tries of each strategy. We have experimentally evaluated the performance of the adaptive strategy in a wide variety of environments. The experiment results indicate that the adaptive strategy outperforms the plain p-strategy when the p-strategy performs poorly, while it performs similar to the p-strategy when the p-strategy dominates the other simple strategies.

Key words : Auctions, Continuous Double Auctions, Auction Agents, Bidding Strategy, Agent Development

1. 서 론

최근 온라인 경매가 보편화함에 따라 에이전트를 이용한 경매참가에 대한 연구가 활발해지고 있다. 특히 연속이중경매(Continuous Double Auction, CDA)에서의 효과

적인 에이전트 비딩 스트레티지는 매우 중요한 관심사이다.

CDA는 다음과 같은 특징을 가지고 있다. 첫째, 어느 한쪽에서 비드를 하는 일반적인 경매와는 달리 판매자들과 구매자들이 양방향에서 모두 비드를 할 수 있다. 둘째, 매수비드(buy bid)와 매도오퍼(sell offer)간의 거래가 (매수입찰가가 매도입찰가보다 작거나 같으면) 끊임없이 이루어지며, 거래가 이루어지지 않은 매수비드나 매도오퍼는 큐에 대기하고 있다가 새로 들어오는 오퍼나 비드

^{*} 이 논문은 연세대학교 경영연구소의 연구비 지원에 의하여 연구되었음

[†] 종신회원 : 연세대학교 경영학과 교수
boxenju@yonsei.ac.kr

논문접수 : 2005년 3월 22일

심사완료 : 2005년 9월 20일

와의 거래가 가능한지를 보도록 되어있다. CDA는 현대 금융시장의 대부분을 이루는 중요한 경매 방법이기도 하나, 그 다이내믹스와 복잡성 때문에 전통적인 게임이론으로는 모델이 어려운 점이 있다. 따라서 아직까지 CDA에서의 우수한 스트레티지는 알려져 있지 않다.

이 논문은 CDA 환경에서 사용할 수 있는 새로운 에이전트 스트레티지를 소개한다. 이전의 연구에서 우리는 P-스트레티지를 개발하고 다른 스트레티지에 비교한 그의 우수성을 실험을 통해 보인 바 있다[1]. 그러나, P-스트레티지는 모든 경우에 항상 우수한 스트레티지는 아니었다. 첫째, P-스트레티지를 쓰는 에이전트가 대부분의 경우에 좋은 이득을 올리는 것은 사실이지만, 경쟁이 치열하거나 다른 에이전트들도 P-스트레티지를 쓰는 경우에는 P-스트레티지를 쓰는 것이 반드시 유리하지는 않았다. 둘째, P-스트레티지에서 사용하는 확률 모델링은 상당한 양의 계산을 요구하므로 그로 인한 심사숙고(deliberation) 오버헤드가 P-스트레티지의 장점을 감소시킬 수 있었다. 따라서, 우리는 대부분의 CDA 경매환경에서 P-스트레티지가 우수한 것을 인정은 하나, 에이전트에게 경매 환경의 변화에 따라 어떤 스트레티지를 사용할지를 실시간에 결정할 수 있는 권한을 주어야 한다고 주장한다.

새로 개발된 스트레티지의 주 아이디어는 에이전트가 언제 P-스트레티지를 쓸지, 언제 쓰지 않을지를 실시간에 결정하도록 하는 데 있다. 우리는 이러한 하이브리드 스트레티지를 “적응성 스트레티지”(Adaptive Strategy)라고 부른다. 실제로 에이전트가 모든 가능한 경매 상황에 적합한 스트레티지를 미리 알아둔다는 것은 불가능하고, 또한 현재의 상황을 완벽하게 알 수 없을 때도 있다. 그러므로 에이전트에게 실시간 적응력을 부가하는 것은 유용하고 실용적인 방법이다. 적응성 스트레티지는 언제 P-스트레티지를 사용할지 안할지를 실시간에 적응성있게 결정함으로써, 경매 진행 상황의 다이내믹스와 불확실성을 모델하는 P-스트레티지의 장점을 취하면서도 복잡한 확률 모델링이 가지는 계산의 오버헤드를 피할 수 있다.

이 논문은 다음과 같이 구성되어 있다. 2장은 관련연구를 설명한다. 3장에서는 간단하게 P-스트레티지를 소개하여 적응성 스트레티지 개발의 당위성에 대한 동기를 부여한다. 4장은 적응성 스트레티지의 탐색 알고리즘을 소개한다. 5장에서는 적응성 스트레티지의 성능을 실험을 통해 분석한다. 마지막 장에서 결론을 맺는다.

2. 관련 연구

엄밀히 말하자면 경매에이전트를 개발하는 것은 경매의 “설계”에 해당하지는 않는다. 이는 경매설계자가 경

매에 참가하는 에이전트들의 비딩스트레티지를 통제/관리하지 못하기 때문이다. 그러나 최근들어 경매가 활발해짐에 따라 에이전트 개발은 (특히 최적의 스트레티지가 존재하지 않는 CDA 환경에 적합한 에이전트 개발은) 하나의 중요한 연구분야로 각광받고 있다. 시장의 변화에 빠르게 대응할 수 있는 효율적인 CDA 방식은 현재 대부분의 금융 시장에서 사용되고 있으며, 이러한 환경에서의 에이전트 개발은 중요한 연구과제이다.

CDA에서의 에이전트 비딩 스트레티지 개발에 관한 연구는 크게 두가지로 나누어 볼 수 있다. 첫째, 에이전트를 이용하여 경매 방식의 속성을 이해하는 것에 목적을 둔 연구들이 있다[2]. 즉 경매에 참가하는 사람들의 행동양식을 체계적으로 변화시킨다는 것은 어려우므로, 대신 소프트웨어 에이전트들을 개발하고 이들의 행동양식들을 변화시켜 경매 진행에서의 특성이나 결과를 분석 조사하는 방법이다. 소프트웨어 에이전트를 이용한 실험은 경매의 규칙/규정의 영향과 개개인의 행동양식의 영향을 따로 분리시켜 연구하는데 특히 유용하다. 예를 들어, 이중경매(double auction)는 다른 경매 방식보다 잉여이득 추출(surplus extraction)의 효율성이 뛰어난데, 이 효율성이 경매방식의 차이에서 오는 것인지 참가자들의 행동양식에서 오는 것인지 또는 두 요인의 상호작용에서 오는 것인지를, 에이전트를 이용하여 조사할 수 있다. 이를 위해 Gode와 Sunder는 (경매 참가자들의 전략적인 행동 양식과는 다른) 매우 단순한 에이전트를 개발하였다. 그들의 실험결과에 따르면 단순한 에이전트들만이 참가하는 경매도 사람이 참여하는 경매와 비슷한 효율성을 얻었고, 따라서 이중경매의 효율성은 경매방식에 따른 것이라는 결론을 낼 수 있었다[3]. 이러한 에이전트를 이용한 마켓 시뮬레이션 연구는 다음의 논문들에서 찾아 볼 수 있다[4-9].

두번째 연구방식은 자치적인(autonomous) 에이전트들의 개발과 그들이 참여할 수 있는 경매시스템의 구현에 초점을 맞추고 있다. 그 예로 에이전트들이 비행기표를 살 수 있는 경매시스템이나, 에이전트들이 참가하는 이중경매 토너먼트의 구현 등을 들 수가 있다[6,10]. 한가지 놀라운 사실은 이중경매 토너먼트에서 우승한 에이전트는 매도입찰가와 매수입찰가의 차이(spread)가 일정값 이하로 떨어질 때까지 기다렸다가 마지막에 그보다 조금 좋은 가격을 비드하여 거래를 가로채는 이른바 “뒤에서 기다리기”(waiting in the background) 스트레티지를 사용하는 간단한 휴리스틱 에이전트라는 것이었다(조용히 기다리다가 마지막에 비드하여 거래를 가로채는 방식의 스트레티지는 최근 eBay와 같은 실제 경매에서도 많이 관측되었다).

앞에서 언급했듯이 Gode와 Sunder는 단순한 에이전트

를 이용하여 경매의 효율성을 조사하였다[3]. 이들의 에이전트는 원가에 무작위로 선택한 이윤을 더하여 비드하는 방식의 간단한 스트레티지를 쓰고 있는데, 이 스트레티지는 그 후 P-스트레티지를 포함한 여러 비딩 스트레티지 개발시에 벤치마크로 사용되고 있다.

진화(evolutionary) 알고리즘도 에이전트 스트레티지 개발에 사용되었다[11-13]. 예를 들어, 원가에 무작위로 이윤을 더해서 비드하는 ZI(zero-intelligence) 스트레티지를 기본으로 하되, 바로 적전의 비드가 거래에 성공했을 때는 이윤의 폭을 올리는 방식의 진화형 휴리스틱을 가미한 ZIP(zero-intelligence plus) 에이전트가 개발되었다[11]. ZIP 에이전트를 이용하여 균형 가격(equilibrium price) 바깥에서 거래되는 경우를 제거한 효율적인 이중 경매 메카니즘이 개발되기도 하였다[14].

Tesauro와 Das는 ZIP와 GD 스트레티지에 기반한 두 종류의 새로운 휴리스틱 스트레티지를 개발하고, 기존의 두 스트레티지와 변형된 두 스트레티지를 실험비교한 결과 변형된 GD 스트레티지가 가장 잘 작동하는 것을 보여주었다[15]. 다른 연구에서는 퍼지 논리를 이용한 비딩 스트레티지를 개발하고 CDA환경에서의 성능을 실험을 통해 보여주었다[16,17].

학습(learning)에 바탕을 둔 휴리스틱 에이전트의 개발도 활발하다. Hu와 Wellman은 강화 학습(reinforcement learning)을 통해 경매 참가자들을 모델하는 에이전트를 개발하였다[18,19]. 이들의 실험에 의하면 지나치게 단순화된 학습을 하는 에이전트는 나쁜 결과를 내었고, 또한 학습 에이전트들이 가지고 있는 상대 에이전트들에 대한 초기 선입견이 경매의 결과에 큰 영향을 미치는 것도 알 수 있었다. 한편, 경매에 참여하는 경쟁자의 속생각을 지나치게¹⁾ 모델링하는 것이 오히려 나쁠 수도 있다는 결과도 있다[20]. 어떤 연구에서는 간단한 학습을 하는 적응성 에이전트가 복잡한 생각과 모델을 하는 사람들만큼 좋은 결과를 보이기도 하였다[12]. 이러한 연구 결과들은 다른 에이전트들을 명시적으로(explicitly) 모델하지 않는 비딩 스트레티지를 만들고자 하는 우리의 방법에 근거를 제시하였다.

모든 휴리스틱 비딩 스트레티지들은 경우에 따라서는 잘 작동하지 않는다는 단점이 있다[21]. 예를 들어, 간단한 휴리스틱 스트레티지인 '뒤에서 기다리기'는 다른 에이전트들이 같은 스트레티지를 쓰지 않을 경우에만 잘 작동한다(모든 에이전트들이 상대가 비드하기만을 기다린다면

어떻게 되겠는가?). 마찬가지로 P-스트레티지의 우수성도 다수의 에이전트들이 P-스트레티지를 쓰면 감소하는 것을 볼 수 있었다[1]. 그러나 P-스트레티지에 적응성을 부과하면 이러한 단점을 어느 정도 막을 수 있다. 즉 적응성 스트레티지는 변화된 경매환경에 맞는 스트레티지를 에이전트가 실시간에 선택할 수 있도록 한다[22,23].

3. P-스트레티지의 재검점

현재 CDA 도메인에서 가장 고급화되고 복잡한 에이전트 스트레티지중 하나는 P-스트레티지이다. 그동안의 에이전트 비딩 스트레티지 개발 연구가 오랫동안 '상대방'을 모델링하는 것에 초점을 맞추고 있었다면, P-스트레티지의 주요 아이디어는 경매 '환경'의 다이내믹스를 마르코프 체인으로 모델링하지는 것에 있다. CDA와 같은 복잡한 환경에서는 에이전트가 일생동안 만나는 여러 타입의 (자주 바뀌는) 상대 에이전트들을 모델링한다는 것 자체에 어려움이 있다. 따라서 P-스트레티지는 이러한 문제점을 해결하기 위해 개개인의 상대 에이전트를 모델하는 대신, 경매 프로세스의 다이내믹스를 모델링하는데 초점을 두고 있다.

P-스트레티지가 CDA를 마르코프 체인을 이용하여 모델링한다는 것은 경매과정의 랜덤 프로세스라고 가정하는 것이다. 에이전트들이 전략적/계략적으로(strategically) 생각하고 활동하는 경매의 현장을 랜덤 프로세스로 모델링한다는 것은 어쩌면 매우 나쁜 결정으로 보일 수도 있다. 그러나 여러가지 에이전트 스트레티지들의 성능을 비교 실험한 결과, P-스트레티지가 (항상 승리하는 스트레티지는 아니더라도) 많은 경우에 가장 좋은 이득을 내는 스트레티지라는 것을 알 수 있었다.

한편, t-테스트를 통한 확률분석의 결과에 의하면 '모든 경우에 P-스트레티지를 꼭 써야 한다는 결론을 내릴 수는 없었다. 첫째, P-스트레티지의 성능은 상대방이 누구인지, 경매에서 더 얻어낼 수 있는 이득이 원래부터 존재했는지, 등등의 여러가지 요인에 의해 좌우된다. 예를 들어, P-스트레티지를 사용하는 판매자는 구매자가 많을 때에 그들로부터 이득을 더 많이 뽑아낼 수 있어 잘 작동한다. 하지만 경쟁이 심할 때는 더이상의 엑스트라 이득을 뽑아내기가 어려워지므로 다른 단순한 스트레티지들과 비슷한 정도의 성능을 보인다. 또한 P-스트레티지를 사용하는 에이전트들이 늘어나면 그 이득이 줄어드는 것을 볼 수 있었다.

둘째, 만약 실험 결과에서 P-스트레티지가 항상 우세하다고 나왔더라도, 실험은 실제 일어날 수 있는 경매상황의 일부분만을 분석하였으므로 그 결과를 (연구 분석하지 않았던) 모든 경매의 상황으로 확대 해석하는 것은 경계해야만 한다.

1) 경매에 참가할 때 우리는 상대가 어떻게 나올지를 예측하려고 애쓴다. 이러한 노력이 지나치게 되면 모델링이 끝이 없게 된다. 즉, 나는 상대를 모델하고, 상대방은 상대를 모델하는 나를 모델하고, 나는 상대를 모델하는 나를 모델하는 상대방을 모델하고, ... 식의 회귀적인(recursive) 모델링이 나올 수 있다.

따라서 경매 에이전트가 실시간에, 상황에 맞추어 어떤 스트레티지를 쓸지를 결정할 수 있도록 하는 것이 가장 좋다고 본다.

4. 적응성 스트레티지의 탐색 알고리즘

적응성 스트레티지의 장점은 언제 어떤 종류의 스트레티지를 쓸지를 실시간에 결정하는 데 있다.

먼저 P-스트레티지에 적응력을 가미하는 단순무식한 방법을 생각해 보도록 하자. 만약 각 경매 상황에 해당 하는 최적의 스트레티지를 알고 있다면, 우리는 에이전트에게 이러한 정보가 들어있는 의사 결정표(decision table)를 주어 에이전트가 각 상황에 맞는 최상의 스트레티지를 선택하도록 하면 된다. 그러나 이러한 방법은 두 가지 문제점을 가지고 있다. 첫째, 모든 가능한 경매 상황의 경우의 수가 너무 많아 미리 모든 상황을 다 조사해둔다는 것은 무리가 있다. 아무리 많은 경우를 미리 조사한다고 해도 실시간에는 또 다른 (새로운) 상황을 만날 수 있기 때문이다. 둘째, 더 중요한 문제점은 에이전트가 현재의 상황을 알 수 없을 경우가 있다는 것이다. 예를 들어, 에이전트는 상대 에이전트가 어떤 스트레티지를 쓰는지를 알 수 없는 경우가 많다.

이러한 이유 때문에 우리는 에이전트 스트레티지 개발에 탐색(exploration)과 이용(exploitation)의 기법을 채택한다. 즉, 예전에 가장 좋은 성능을 보였던 스트레티지를 계속 사용하려는 exploitation과 미래에 다른 종류의 스트레티지가 더 좋은 성능을 보일지를 확인해보려는 exploration을 조화시키는 것이 목적이다. 그리고 이것을 위해 각 스트레티지의 기대이익과 시도 횟수의 상관관계(tradeoff)를 고려하는 휴리스틱 탐색 함수를 개발하였다.

그림 1은 탐색과 이용의 원리에 기반하여 개발된 적응성 탐색 알고리즘을 보여준다. Q_i 는 Δ 크기의 관찰 윈도우(observation window) 내에서 스트레티지 i 가 얻은 비드당 평균 이익(average profit per try)을 표현하고, E_i 는 관찰 윈도우에서 스트레티지 i 가 시도된 정도를 퍼센트로 표현한다. 높은 Q_i 값은 스트레티지 i 가 평균적으로 높은 이익을 얻었다는 것을, 높은 E_i 값은 스트레티지 i 가 많이 사용되었다는 것을 말한다. 적응성 알고리즘의 기본 아이디어는 Δ 안에서 가장 좋은 성능을 보인 스트레티지를 사용하되, 상황이 바뀔 수 있으니 다른 스트레티지들도 가끔씩 사용해 보자는 데에 있다. 이를 위해, Q_i 가 클수록 또는 E_i 가 낮을수록 함수값이 큰 함수 $f(Q_i, E_i)$ 를 정의하고, 비드를 할 때 $f(Q_i, E_i)$ 가 가장 높은

Function Adaptive Strategy() returns an action

a_i : choices of actions (e.g., different bidding strategies)

t : current time

w_{lb}, w_{ub} : the lower & upper bound of the current observation window

Δ : the length of the observation window

E_i : action frequencies (%) during the current observation window

Q_i : action-values (average profit per try) during the current observation window

$f(Q_i, E_i)$: a heuristic exploration function that increases with Q_i and decreases with E_i

/* slide the current observation window */

$w_{ub} = t$;

$w_{lb} = t - \Delta$;

/* update the Q_i and E_i values */

for each a_i

 update E_i to the percentage of action a_i during $[w_{lb}, w_{ub}]$

 update Q_i to the average profit of action a_i per try (i.e., offer) during $[w_{lb}, w_{ub}]$

end

/* select a strategy with the highest f -value */

action = $\operatorname{argmax}_i (f(Q_i, E_i))$;

return action

그림 1 적응성 스트레티지의 탐색 알고리즘

스트레이지를 선택하도록 한다. 비드를 한 후에는 관찰 윈도우를 앞으로 이동시켜 현재의(변해졌을지도 모르는) 상황에 적합하지 않는 오래된 Q_i 와 E_i 의 값을 버린다.

그림 1의 알고리즘을 이용하는 적응성 스트레이지의 행동 양식은 (1) Q_i 와 E_i 의 값의 변경 방법, (2) $f(Q_i, E_i)$ 함수, (3) 관찰 윈도우 (Δ)의 길이라는 세가지 요소에 의해 좌우된다. 다음에서 적응성 알고리즘이 이 세가지 요소를 어떻게 사용하는지에 대해 자세히 알아보도록 하자.

첫째, 다음과 같은 등식으로 Q_i 와 E_i 의 값을 업데이트한다.

$E_i = n_i / \sum n_k$, 여기에서 n_i 는 Δ 의 구간 안에서 스트레이지 i 를 시도한 횟수이다.

$Q_i = (\sum \Delta q_i) / n_i$, 여기에서 q_i 는 스트레이지 i 가 경매 성공시에 얻은 이득을 말한다.

위와 같은 방법 외에도 최근의 경매성공에 더 중요치를 주는 대안도 가능하다. 예를 들어, $Q_i = (1-\rho) \times Q_i + \rho \times q_i$, $0 < \rho < 1$ 와 같은 방법도 있다. 이때 높은 ρ 값은 최근의 경매를 성공시킨 스트레이지에게 더 가중치를 주게 된다.

둘째, 탐색 함수를 사용함으로써 에이전트는 높은 이득을 내는 스트레이지를 계속 사용하려는 성향과, 변해간 상황에 따라 달라질 수도 있는 각 스트레이지의 성능을 탐구해 보려는 두가지 상반된 목적을 균형있게 이루고자 한다. 이를 위한 탐색함수, $f(Q_i, E_i)$,는 높은 이득에 대한 선호도(즉, 높은 Q_i 값)와 탐색에 대한 선호도(즉, 낮은 E_i 값)를 표현해야 한다. 우리는 다음과 같은 함수를 사용한다.

$$f(Q_i, E_i) = Q_i + a/E_i$$

이때 관찰 윈도우 안에서 각 스트레이지가 적어도 E 퍼센트 이상 실행되는 것을 보장하려면, a 의 값은 $Q \cdot E$ 보다 커야 한다. 여기에서 Q 는 적응성 스트레이지를 사용하는 에이전트의 최대가능 기대이득(maximum possible expected profit)을 나타낸다. Q 값은 이제까지의 거래기록을 토대로 정해지며, 경매 상황이 변함에 따라 그 값이 달라진다. 따라서 적응성 스트레이지에 사용하는 a 의 값도 자동적으로 변하도록 설계되어 있다.

최소한의 시도 횟수를 보장하기 위해, 시도 횟수의 퍼센트(E)를 쓰지 않고 시도 횟수의 절대값(N)을 사용하는 방법도 있을 수 있다. 높은 E 값이나 높은 N 값은 둘 다 적게 실행된 스트레이지를 더 사용하게 하는 효과를 가진다. 하지만 상대적인 퍼센트(E)를 이용하면 $f(Q_i, E_i)$ 의 값이 관찰 윈도우상에서 시도된 각 스트레이지의 절대 횟수와 무관하다. 즉, N 를 이용하면 최소 시도 '횟수'가 정확하게 얼마인지를 알아내야 하지만, E 를 사용하는 알고리즘은 그러한 걱정을 하지 않아도 되는 장점이 있다.

한편, E 를 사용하는 경우에는 자주경매에 참여하는

(즉, 비드를 자주 하는) 에이전트가 같은 관찰윈도우내에 비드를 적게 하는 에이전트보다 더 많은 '시간'을 탐색에 쓰도록 만든다. 따라서 같은 E 의 값은 (1) 낮은 비드 레이트의 에이전트는 탐색을 너무 적게 하도록, (2) 높은 비드 레이트의 에이전트는 탐색을 너무 많이 하도록 만들 수도 있다. 이러한 문제점을 해결하기 위해, 우리는 환경의 변화에 비해 비드 레이트가 너무 크거나 작을 때 E 의 값을 자동적으로 조절할 수 있도록 탐색함수를 설계하였다.

이제까지 탐색함수 $f(Q_i, E_i)$ 가 어떻게 각 스트레이지의 최소 시도횟수를 보장하면서도 높은 이득을 내는 스트레이지를 선택할 수 있는지를 살펴보았다. 그러나 위에 정의된 $f(Q_i, E_i)$ 는 심사숙고(deliberation) 오버헤드로 인한 경매성공 횟수의 감소를 고려하지 않고, 비드당 평균 이득만을 고려하므로 심사숙고 오버헤드를 완벽하게 모델하고 있지는 않다. 예를 들어, a_1 과 a_2 라는 두개의 스트레이지가 비슷한 정도의 이득을 내며, a_1 만이 심사숙고 오버헤드를 가지는 경우를 살펴보자. 현재의 탐색 함수는 심사숙고 오버헤드가 없는 a_2 가 더 좋은 스트레이지라는 것을 알지 못하고 두개의 스트레이지를 똑같은 정도로 시도해 볼 것이다. 탐색 함수에 다음과 같이 심사숙고 오버헤드에 의한 비드횟수의 감소를 표현하는 매개변수 (D_i)를 추가하면 이러한 경우가 생기는 것을 방지할 수 있다.

$$D_i = \text{deliberation delay} / (\text{deliberation delay} + \text{average offer interval}).$$

따라서 적응성 스트레이지에서 사용하는 탐색 함수는 다음과 같다.

$$f(Q_i, E_i, D_i) = (1 - D_i) \times Q_i + E_i$$

마지막으로, 관찰 윈도우의 길이, Δ ,를 선택하는 방법에 대해 알아보도록 한다. 적응성 알고리즘은 관찰 윈도우의 시작부근에서는 어느 스트레이지가 가장 좋은 성능을 보이는지 알아내기 위해 여러가지 스트레이지를 시도해 본다. 그 후에는 최소 시도횟수를 맞추기 위해 다른 스트레이지를 실행해 보는 경우를 제외하고는 가장 좋은 스트레이지를 계속 실행한다. 따라서 작은 Δ 값은 (Δ 안에서 모든 스트레이지를 적어도 E 퍼센트로 시도함으로써) 탐색을 더 자주 하도록 하여, 변화하는 환경에 빨리 대응할 수 있도록 한다. 그러나 최상의 스트레이지가 변하지 않는다면 작은 Δ 값은 열등한 스트레이지를 자주 시도함으로써 이득을 낮추는 위험도 있다. 최상의 스트레이지가 변하지 않는 상황에서는 높은 Δ 값을 사용하면 탐색을 적게하여 더 높은 이득을 올릴 수 있다. 그러나 환경이 급변한다면 높은 Δ 값을 이용하는 적응성 알고리즘은 (예전에는 우수했지만) 지금은 잘 작동하지 않는 스트레이지를 고집함으로써 이득을 줄일 수 있다.

그림 2는 적응성 알고리즘이 잘 작동하지 않는 상황을 과장하여 보여준다. 그림 2(a)에서 보여진 것처럼 최상의 스트레티지가 t_1 에서 변하는 상황에서, Δ 가 $t_0 \sim t_2$ 보다 길다고, 또 탐색이 주로 $t_0 \sim t_1$ 구간에서 이루어진다고 가정해보자. $t_0 \sim t_1$ 구간동안에는 적응성 스트레티지가 탐색을 위해 스트레티지 a_1 과 a_2 를 모두 시도하므로, t_1 까지의 총이익은 a_1 과 a_2 사이의 이익을 얻을 수 밖에 없다. 또한 $t_1 \sim t_2$ 구간에서는 탐색기간인 $t_0 \sim t_1$ 구간에서 a_2 보다 뛰어난 성능을 보였던 a_1 을 주로 시행하는 오류를 범한다. 결과적으로, 적응성 스트레티지의 총이익은 두 스트레티지보다 낮게 된다(그림 2(b) 참조).

위에서 본 바와 같이 최적의 Δ 값은 환경의 변화 특성에 따라 변한다. 최상의 스트레티지가 변하는 환경에서는 짧은 Δ 값이 유리하며, 환경의 변화가 적으면 긴 Δ 값이 유리하다. 만일 환경이 변하지 않더라도 최상의 스트레티지가 확실하지 않는 경우에는 짧은 Δ 가 유리하다. 그러나 경매환경을 미리 알 수 없으므로 Δ 값을 최적의 값으로 미리 정할 수가 없는 것이 사실이다. 따라서 적응성 알고리즘은 Δ 값을 실시간에 변화시킬 수 있도록 설계하였다. 즉, 낮은 Δ 값에서 시작하여 현재의 우수 스트레티지가 잘 작동하는 동안에는 Δ 를 미리 정해놓은 최대치(max- Δ)까지 점차적으로 증가시키고, 다른 스트레티지가 더 좋은 이익을 올리면 Δ 값을 감소시킨다. 이러한 다이내믹한 Δ 의 조절 메카니즘은 TCP의 슬라이딩 윈도우 컨트롤 기법을 응용한 것이다[24].

5. 적응성 스트레티지의 성능에 대한 실험 분석

적응성 스트레티지는 다른 스트레티지가 P-스트레티지보다 더 좋은 이익을 얻는 경우가 있다는 것을 인정하고, 이런 상황에서는 다른 스트레티지를 써서 (P-스트레티지만을 고집하는 에이전트보다) 좋은 성능을 보이는데 주안점을 두고 있다. 따라서 이번 장에서 우리는 실험을 통해 과연 적응성 스트레티지가 P-스트레티지보다

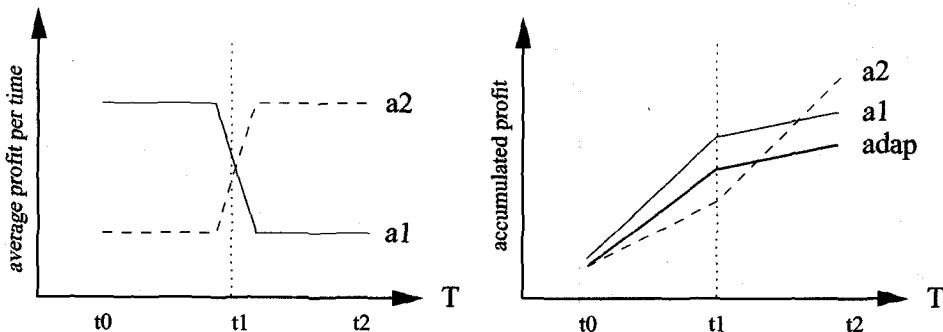
나은 결과를 낳는지를 알아보려 한다.

어쩌면 간단하게 보이는 이 문제는 적응성 알고리즘들이 갖는 기본적인 성질 때문에 대답하기가 쉽지 않다. 적응성 알고리즘은 선택가능한 여러 스트레티지들을 시도하여 현재 상황에 가장 잘 작동하는 스트레티지를 사용한다. 다시 말해 적응성 스트레티지는 탐색을 위해 현재 잘 작동하지 않는 열등한 스트레티지를 시도해 보기 때문에 아무리 잘해도 최상의 스트레티지를 능가하지 못할 수 밖에 없다. 예를 들어, P-스트레티지가 항상 가장 좋은 스트레티지인 상황을 생각해 보자. 이때 적응성 스트레티지를 사용한다면 P-스트레티지보다 적은 이익을 올릴 것이다. 만일 경매 환경이 계속 바뀌어 하나의 최상의 스트레티지가 없는 경우라도 적응성 스트레티지가 P-스트레티지를 이기는 정도는 각 스트레티지의 성능이 얼마나 오르내리는지에 따라 결정된다. 그러나 어떠한 환경에서도 적응성 스트레티지는 최상의 스트레티지에 근접하는 성능을 보여야 한다.

이 장에서는 적응성 스트레티지를 세가지 측면에서 분석한다. 첫째, 여러 다양한 환경에서 적응성 알고리즘이 언제 어떤 방식으로 최상의 스트레티지를 선택하는지를 알아본다. 둘째, 스테틱 환경에서 적응성 스트레티지가 어떻게 작동하는지를 알아본다. 마지막으로 환경의 변화에 따라 최상의 스트레티지가 변할 때 적응성 스트레티지가 어떻게 작동하는지를 알아본다.

앞으로의 모든 실험에서 적응성 스트레티지가 P-스트레티지와 CP-스트레티지의 두가지 선택을 할 수 있다고 가정한다. 두개의 스트레티지만을 사용하여 실험 분석과 분석 결과의 이해가 용이하며, 이러한 분석 결과는 다수의 스트레티지가 있는 경우에도 적용된다.

CP-스트레티지는 청산가격(clearing price)을 이용하는 휴리스틱이다. 예를 들어, CP-스트레티지를 사용하는 판매자는 다음번 청산가격에 대한 정보를 입수하여, 그 청산가격이 자신의 원가(cost)보다 높으면 청산가격



(a) 스트레티지의 평균이익

(b) 스트레티지의 총이익

그림 2 최상의 스트레티지가 바뀌는 경우에 적응성 스트레티지의 성능

을 비드하고, 원가보다 낮으면 원가에 일정한 이윤(markup)을 더해서 비드한다(이 경우는 큐에서 다른 더 나은 비드가 들어오기를 기다리게 되는 셈이다). 청산가격을 비드한 경우에는 청산가격을 알아본 순간부터 비드할 때까지 사이에 청산가격이 변하지 않았다면(즉, 다른 비드가 들어오지 않았다면) 거래가 성립된다. 따라서 CP-스트레티지는 경매가 활발하지 않을 때 특히 잘 작동하는 휴리스틱이다. 하지만 새로 들어오는 매도비드와 매수오퍼들을 고려하지 않기 때문에 CP-스트레티지를 사용하면 앞으로 있을 더 좋은 거래를 놓칠 수도 있다.

5.1 적응성 스트레티지의 작동에 대한 이해

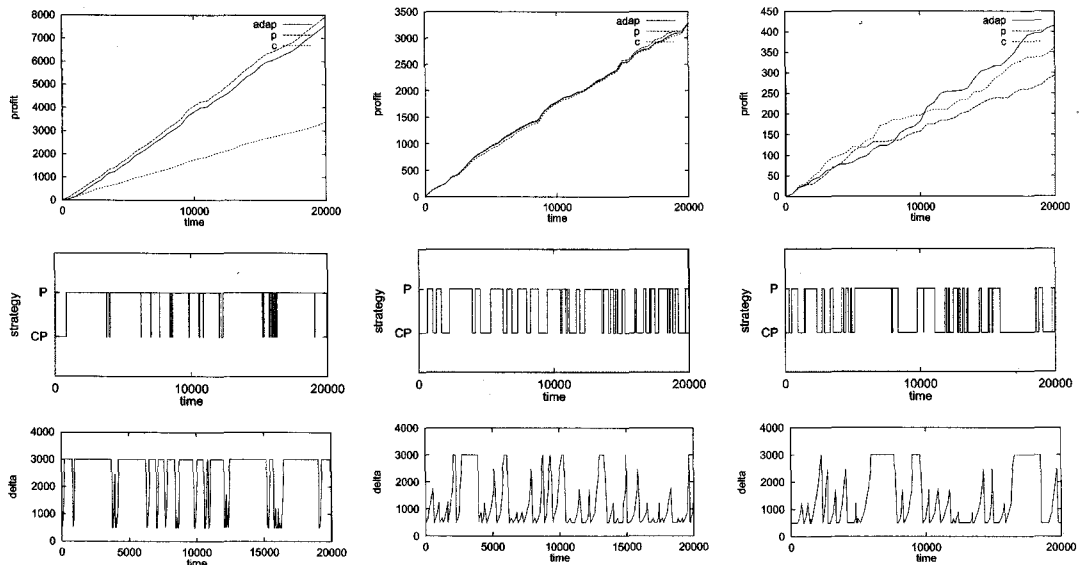
선택 가능한 스트레티지가 두 종류가 있다고 가정한다면, 경매 환경은 (1) 하나의 스트레티지가 다른 스트레티지보다 월등히 우세한 경우, (2) 두 스트레티지가 비슷하게 작동하는 경우, (3) 최상의 스트레티지가 계속 변화하는 경우의 세가지의 경우로 나눌 수 있다. 그림 3(a), (b), (c)는 이러한 세가지 경우에 적응성 알고리즘이 어떻게 작동하는지를 보여준다.

각각의 그림에서 첫번째 그림은 세 스트레티지의 누적된 총이익을 나타낸다. 두번째 그림은 적응성 알고리즘이 각 단위시간에 어느 스트레티지를 선택했는지를 보여준다. 마지막 그림은 관찰 윈도우의 길이(Δ)의 변화를 보여준다. 실험에서는 P-스트레티지와 기본 이윤(default markup)이 5인 CP-스트레티지의 두가지 대안 스트레티

지를 사용한다. 관찰 윈도우의 길이(Δ)는 500에서 3,000 사이에서 적응성있게 정해지며, 각 스트레티지의 최소 시도 퍼센트 (E)는 10%로 정한다.

그림 3(a)는 P-스트레티지가 최상의 스트레티지인 환경에서 적응성 알고리즘이 대부분의 경우에 P-스트레티지를 선택하는 것을 보여준다. 이때 Δ 값은 최대치인 3,000에 빨리 이르르고, 최대치에 근접한 값을 계속 유지하는 것을 알 수 있다. 또한 매 3,000 싸이클마다 탐색을 위해 CP-스트레티지와 P-스트레티지를 시도해 보는 것을 알 수 있다. 이러한 스트레티지 탐색 때문에 결과적으로 적응성 P-스트레티지는 P-스트레티지보다 약간 낮은 이득을 얻는 것을 볼 수 있다. 그림 3(a)의 두번째 그림에서 볼 수 있듯이 실험 초기에는 P-스트레티지가 최상의 스트레티지라는 정보가 없으므로 CP-스트레티지가 많이 선택된다. 또한 P-스트레티지의 성능이 오르내리는 16,000사이클 부근에서는 상대적으로 긴 탐색 기간이 있는 것을 알 수 있다.

그림 3(b)처럼 두 스트레티지의 성능이 비슷할 경우는 Δ 값이 최대치에 이르는 경우가 거의 없는 것을 알 수 있다. 어느 스트레티지도 다른 스트레티지를 누르지 못하기 때문에 적응성 알고리즘은 두 스트레티지를 비슷한 정도로 시도해 본다. 즉, 적응성 알고리즘은 계속하여 탐색을 하고 있다. 최상의 스트레티지가 계속 변하는 그림 3(c)의 경우에는 적응성 알고리즘이 상황에 맞추어 탐색과 이용을 하는 것을 알 수 있다. 이때 6,000 싸이클의



(a) 하나의 스트레티지가 다른 스트레티지보다 우세한 경우 (b) 두 스트레티지가 비슷한 성능을 보이는 경우 (c) 최상의 스트레티지가 계속 변하는 경우

그림 3 세가지 환경에서의 적응성 스트레티지의 작동

부근에서는 P-스트레티지가, 16,000 싸이클의 부근에는 CP-스트레티지가 선택되는 것을 알 수 있다.

5.2 스택틱 환경에서의 적응성 스트레티지의 성능

이번 장에서는 참여하는 에이전트의 스트레티지가 변하지 않는 스택틱 환경에서의 적응성 스트레티지의 성능을 알아본다. 실험환경은 다음과 같다. 총 10명의 매수 에이전트와 10명의 매도 에이전트가 있으며, 그 중 하나의 매도 에이전트에게 비교대상인 적응성 스트레티지, P-스트레티지, CP-스트레티지를 쓰게 하여 각각의 실험 결과를 비교한다. 모든 매수 에이전트는 자신이 원하는 가격(reservation price)을 비드하며, 비교대상을 제외한 나머지 9명의 매도 에이전트는 아래와 같은 스트레티지를 쓴다.

- 실험 I: 9명의 매도 에이전트들은 자신의 원가를 그대로 비드한다.
- 실험 II: 5 에이전트는 P-스트레티지를 사용한 비드를 하고 나머지 4명은 자신의 원가를 비드한다.

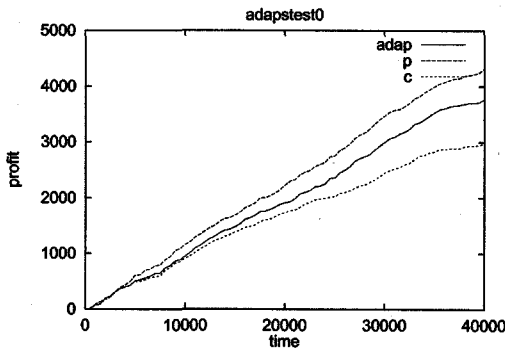
실험 I과 II의 결과는 그림 4와 그림 5에 각각 나와 있다. 각 그림에서 (a), (b), (c), (d)는 경매의 활발한 정도와 매수/매도비드의 도착률을 달리한 실험을 나타낸다.

(a)는 경매가 활발하지 않으며 매수와 매도 비드가 비슷한 정도로 도착하는 경우, (b)는 매수비드가 매도비드보다 적은 경우, (c)는 매수비드가 매도비드보다 많은 경우, (d) 경매가 활발하며 매수와 매도 비드가 비슷한 도착률을 보이는 경우를 나타낸다(자세하게 말하자면, (a), (b), (c), (d)는 매수와 매도 비드의 도착률이 각각 (0.1 & 0.1), (0.1 & 0.4), (0.4 & 0.1), (0.4 & 0.4)인 경우를 나타낸다).

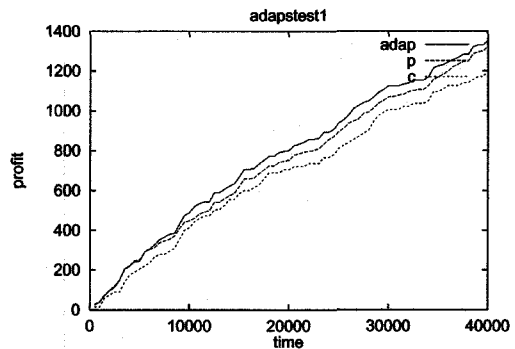
각각의 경매상황에 가장 적당한 max-4와 E를 선택할 수 있었다면 적응성 스트레티지의 성능이 많이 높아질 수 있다. 그러나 가장 좋은 값은 시행착오를 통해 알아낼 수밖에 없기 때문에, 실험에서는 max-4를 3,000으로 E를 10%로 통일하였다.

그림 4에 나타난 실험 I의 결과를 보면 (a)와 (d)에서는 P-스트레티지가 CP-스트레티지보다 월등히 나으며 (b)와 (c)에서는 비슷한 성능을 보인다. 예상했던대로 적응성 스트레티지는 (a)와 (d)에서는 P-스트레티지에 이어 두번째로 좋은 (second-best) 결과를 보여주며, (b)와 (c)에서는 세 스트레티지가 모두 비슷한 성능을 보인다.

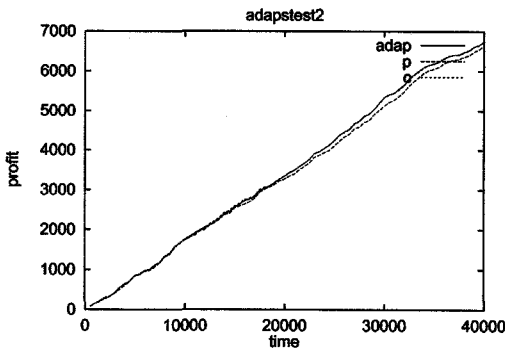
한편 그림 4의 (a)와 (d)의 경우를 자세히 비교해 보



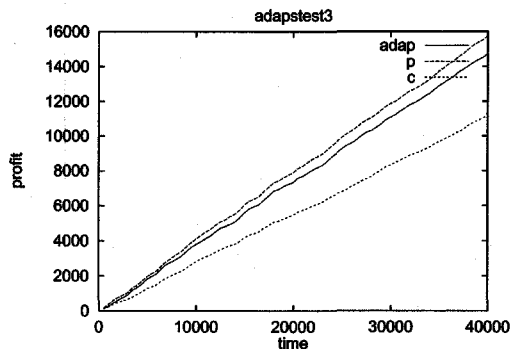
(a) 매수/매도의 도착률: (0.1 & 0.1)



(b) 매수/매도의 도착률: (0.1 & 0.4)



(c) 매수/매도의 도착률: (0.4 & 0.1)



(d) 매수/매도의 도착률: (0.4 & 0.4)

그림 4 실험 I의 결과도

면, 적응성 스트레티지가 (d)에서 더 P-스트레티지에 근접한 성능을 보였다는 차이점을 알 수 있다. 이는 P-스트레티지와 CP-스트레티지의 성능의 차이가 (a)보다 (d)에서 더 심하기 때문이다. 최상의 스트레티지가 확실히 구분되는 (d)에서는 Δ 값이 최고치에 더 오래 머물게 되고 따라서 더 나은 스트레티지인 P-스트레티지가 더 많이 선택된 때문이다.

그림 5는 다수의 P-스트레티지 에이전트가 존재하는 경매에 대한 실험(실험 II)의 결과이다. 경쟁하는 P-스트레티지 에이전트가 많아질수록 P-스트레티지의 성능이 떨어지므로, 그림 4에 비해 P-스트레티지와 CP-스트레티지간의 차이가 현저히 감소되었다. (b)와 (c)의 경우에는 CP-스트레티지가 P-스트레티지를 약간 앞서는 경우가 생겼다. 그러나 어떠한 경우에도 적응성 스트레티지는 최상의 스트레티지에 근접한 성능을 보이는 것을 알 수 있다.

5.3 다이내믹 환경에서의 적응성 스트레티지의 성능

이제까지 우리는 적응성 스트레티지가 여러 스테틱 환경에서 잘 작동함을 보았다. 이번 장에서는 변화하는 환경에서도 적응성 스트레티지가 좋은 성능을 보이는 지를

알아보고자 한다.

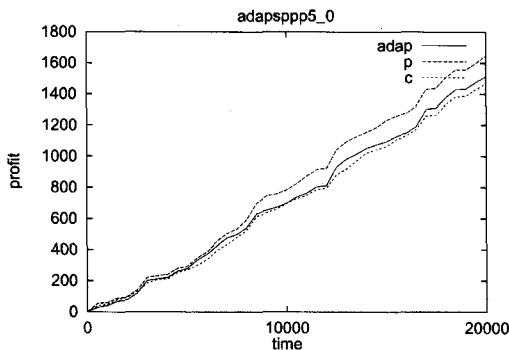
우리는 다음과 같은 세가지의 다이내믹 환경을 모델한다.

- 유형 1: P-스트레티지가 최상의 스트레티지인 두개의 스테틱 환경들로 구성된 경매 환경,
- 유형 2: CP-스트레티지가 최상의 스트레티지인 두개의 스테틱 환경들로 구성된 경매 환경,
- 유형 3: 각 상황에서의 최상의 스트레티지가 다른 경우.

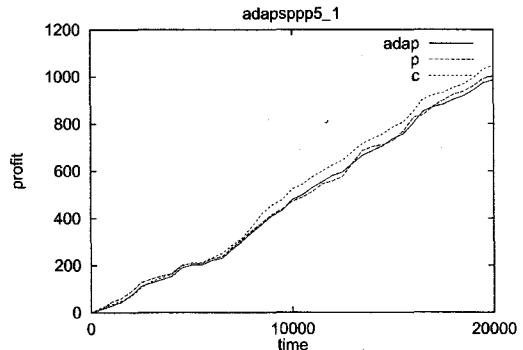
각 유형에 해당하는 실험의 결과는 그림 6, 그림 7, 그림 8에 있다. 각 유형에 대해 두 가지의 다른 환경을 가지고 실험을 했으며, 각 실험에서 두 스테틱 환경의 변화는 20,000 사이클에서 일어난다.

실험에 의하면 적응성 스트레티지는 별다른 성능의 저하없이 어떠한 환경하에서도 최상의 스트레티지와 비슷한 정도의 성능을 보인다. 전체적으로 볼때 유형3에서는 20,000 사이클 부근에서 가장 많이 선택되는 스트레티지가 변하게 되나, 이 변화는 그림만으로는 쉽게 판단하기 어렵다. 유형 1과 유형 2의 실험에서는 두개의 환경이 변화하는 20,000 사이클 부근에서 일시적으로 탐색 시도를 많이 한다.

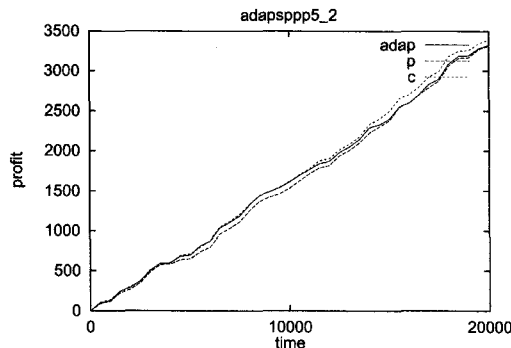
그림 9의 (a)와 (b)는 변화하는 환경에서의 스트레티지



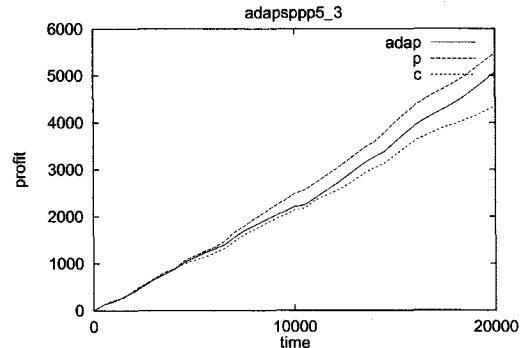
(a) 매수/매도의 도착률: (0.1 & 0.1)



(b) 매수/매도의 도착률: (0.1 & 0.4)



(c) 매수/매도의 도착률: (0.4 & 0.1)



(d) 매수/매도의 도착률: (0.4 & 0.4)

그림 5 실험II의 결과도

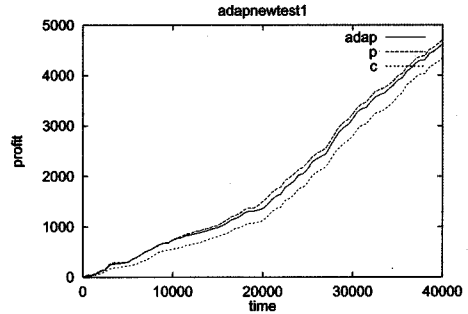
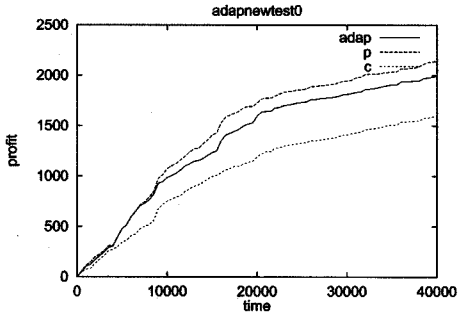


그림 6 유형 1의 다이내믹 환경에서의 결과

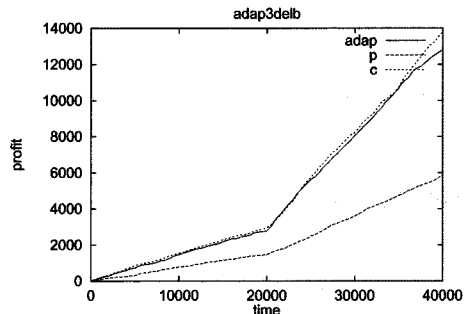
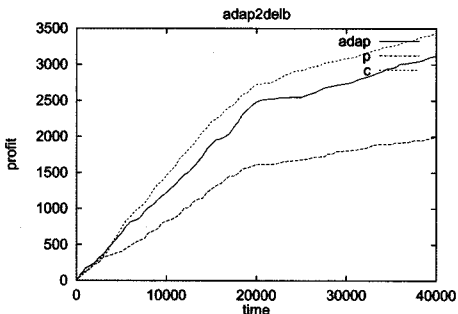


그림 7 유형 2의 다이내믹 환경에서의 결과

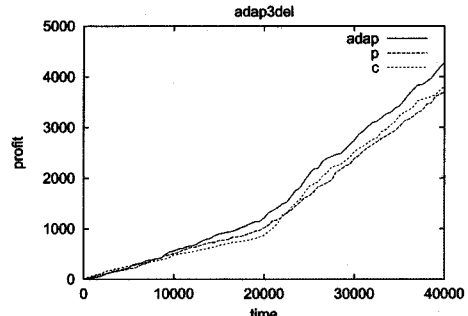
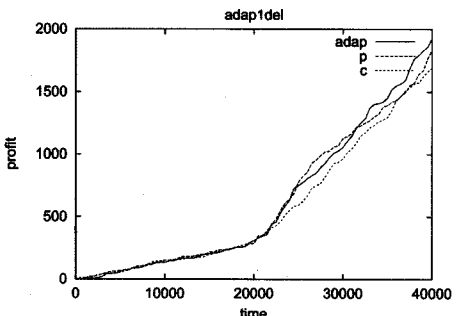
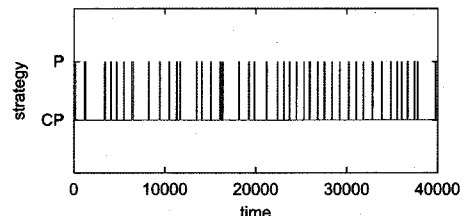
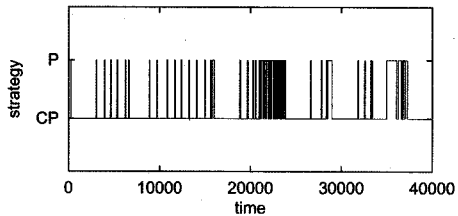


그림 8 유형 3의 다이내믹 환경에서의 결과



(a) CP와 P-스트레티지의 성능차이가 줄어들 때

(b) CP와 P-스트레티지의 성능차이가 늘어날 때

그림 9 변화하는 환경에서의 스트레티지 선택

선택을 보여준다. (두 경우 모두 CP-스트레티지가 우세한 경우인 것을 유의하자.) 그림 9(a)는 두번째 환경에서 CP-스트레티지와 P-스트레티지의 성능차이가 줄어들 때 경우에 탐색이 20,000 싸이클을 넘어서도 계속 이루어지

는 것을 보여준다. 한편 그림 9(b)는 20,000 싸이클 이후에 CP-스트레티지와 P-스트레티지의 성능 차이가 늘어나는 경우에는 계속해서 최상의 스트레티지인 CP-스트레티지를 선택하는 것을 알 수 있다.

6. 결론

P-스트레티지는 CDA환경의 다이내믹스를 확률모형으로 모델링하는 방법을 이용하여 대부분의 경제상황에서 다른 단순한 스트레티지보다 더 좋은 성능을 보여주고 있다. 그러나, 모형을 구축하고 그의 확률을 계산하는데 드는 적지않은 오버헤드는 P-스트레티지의 유용성을 떨어트린다. 따라서, 우리는 심사속고 오버헤드가 높거나 경쟁이 심해 P-스트레티지가 덜 효율적일때에 사용할 수 있는 적응성 스트레티지를 개발하였다.

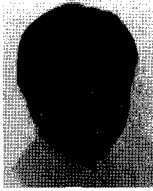
기본 아이디어는 에이전트에게 확률 모형이 언제 유용하고 언제 유용하지 않는지를 실시간에 결정할 수 있도록 하자는데 있다. 탐색과 이용의 균형은 기대 이득과 각 스트레티지의 실행횟수의 상반관계를 고려하는 휴리스틱 탐색 함수를 통해 얻는다. 여러 환경에서 적응성 스트레티지의 성능을 실험한 결과, 적응성 스트레티지는 (1) P-스트레티지가 잘 작동하지 않는 환경에선 P-스트레티지보다 높은 이득을, (2) P-스트레티지가 다른 단순한 스트레티지를 앞서는 환경에서는 P-스트레티지와 비슷한 이득을 보인다.

에이전트 비딩 스트레티지에 적응성을 가미한 것은 실용적인 의미가 크다. 첫째, 적응성 스트레티지는 상황의 변화에 잘 적응한다. 따라서 단순한 스트레티지가 더 복잡한 P-스트레티지와 비슷한 (또는 더 좋은) 성능을 보일때, 단순한 스트레티지를 사용함으로써 P-스트레티지만을 고집하는 에이전트보다 높은 이득을 얻을 수 있었다. 둘째, 적응성 알고리즘은 현재 존재하는 스트레티지보다 나은 스트레티지가 개발되면 그를 선택가능한 스트레티지중의 하나로 사용할 수 있도록 설계되어 있다. 이러한 확장성(expandability)은 최상의 스트레티지가 존재하지 않는 경매에 참여하는 에이전트 개발에서 주안점을 두어야 할 중요한 지표중의 하나이다.

참고 문헌

- [1] Park, S., E. H. Durfee, et al. "Use of Markov Chains to Design an Agent Bidding Strategy for Continuous Double Auctions," *Journal of Artificial Intelligence Research*, Vol. 22, 175-214, November, 2004.
- [2] Steiglitz, K., M. L. Honig, et al. "A Computational Market Model based on Individual Action," *Market-based Control: A Paradigm for Distributed Resource Allocation*. S. Clearwater. 1996.
- [3] Gode, D. K. and S. Sunder. "Lower Bounds for Efficiency of Surplus Extraction in Double Auctions," *The Double Auction Market: Institutions, Theories, and Evidence*. D. Friedman and J. Rust. Reading, MA, Addison-Wesley: 199-219. 1993.
- [4] Verkama, M., R. P. Hamalainen, et al. "Multi-Agent Interaction Processes: From Oligopoly Theory to Decentralized Artificial Intelligence," *Group Decision and Negotiation* 2: 137-159. 1992.
- [5] Roth, A. E. "On the Early History of Experimental Economics," *Journal of the History of Economic Thought*: 184-209. 1993.
- [6] Rust, J., J. Miller, et al. "Behavior of Trading Automata in a Computerized Double Auction Market," *The Double Auction Market*. D. Friedman and J. Rust: 155-198. 1993.
- [7] Kagel, J. and A. E. Roth, Eds. *Handbook of Experimental Economics*, Princeton University Press. 1995.
- [8] Roth, A. E. "Introduction to Experimental Economics," *Handbook of Experimental Economics*. J. Kagel and A. E. Roth, Princeton University Press: 3-109. 1995.
- [9] Kirchler, E., B. Maciejovsky, and M. Weber (Forthcoming). "Framing Effects, Selective Information and Market Behavior: An Experimental Approach," *Journal of Behavioral Finance*. 2005.
- [10] White, J. E. "Telescript Technology: The Foundation for the Electronic Marketplace," *White Paper*, General Magic. 1994.
- [11] Cliff, D. "Genetic Optimization of Adaptive Trading Agents for Double-Auction Markets," *Autonomous Agents '98 Workshop, Artificial Societies and Computational Markets*, Minneapolis/St.Paul. 1998.
- [12] Oliver, J. R. "On Artificial Agents for Negotiation in Electronic Commerce," *Dissertation, Wharton school of business*. Philadelphia, U of Pennsylvania. 1998.
- [13] Bye, A. "Applying Evolutionary Game Theory to Auction Mechanism Design," *Technical Report, HPL2002-321*, Hewlett-Packard Lab. 2002.
- [14] Priest, C. "Commodity Trading Using an Agent-Based Iterated Double Auction," *Technical Report: HPL-2003-238*, Hewlett-Packard Lab. 2003.
- [15] Tesauro, G. and R. Das. "High-performance bidding agents for the continuous double auction," *Proceedings of the 3rd ACM conference on Electronic Commerce*, 206-209, Tampa, Florida, USA, 2001.
- [16] He, M. and N. R. Jennings. "Designing a Successful Trading Agent: A Fuzzy Set Approach," *IEEE Transactions on Fuzzy Systems*, Vol 12, No. 3: 389-410. 2004.
- [17] Vytelingum, P., R. K. Dash, E. David, and N. R. Jennings. "A Risk-Based Bidding Strategy for Continuous Double Auctions," *European Conference on Artificial Intelligence*, 79-83. 2004.
- [18] Wellman, M. P. and J. Hu. "Conjectural Equilibrium in Multiagent Learning," *Machine Learning* 33: 179-200. 1998.
- [19] Hu, J. and M. P. Wellman. "Learning About Other Agents in a Dynamic Multiagent System," *Cognitive Systems Research* 2: 67-79, 2001.
- [20] Gmytrasiewicz, P. J. and E. H. Durfee. "Rational Communication in Multi-Agent Systems," *Autonomous Agents and Multi-Agent Systems Journal*, 4(3): 233-272. 2001.

- [21] Bartos, O. J. *Process and Outcome of Negotiations*, Columbia University Press. 1974.
- [22] Watkins, C. J. and P. Dayan. "Q-learning," *Machine Learning*, 8: 279-292. 1992.
- [23] Russell, S. and P. Norvig. *Artificial Intelligence: A Modern Approach*, Prentice Hall. 1995.
- [24] Tanenbaum, A. *Computer Networks*, Prentice Hall. 1996.



박 선 주

1989년 서울대학교 컴퓨터공학과(학사)
1991년 서울대학교 컴퓨터공학과(석사)
1999년 Univ. of Michigan, Ann Arbor,
CSE (Ph.D.). 1999년~2005년 Rutgers
University, MSIS Department (Assis-
tant Professor). 2005년~현재 연세대학
교 경영학과(조교수). 관심분야는 에이전트 시스템, 전자상
거래, 옵션, 인공지능, SCM.