

실시간 환경에서 긴급한 정도의 계산을 통한 자율적인 에이전트의 유연한 의사결정

(Flexible Decision-Making for Autonomous Agents Through Computation of Urgency in Time-Critical Domains)

노 상 욱 ^{*}
(Sanguk Noh)

요 약 자율적인 에이전트들은 이성적인 의사결정을 위하여 상당한 양의 계산자원을 필요로 하며, 실시간 환경에서 항상 최적의 행동을 수행하는 완벽하게 이성적인 에이전트(rational agent)의 구현은 실질적으로 가능하지 않다. 이러한 실시간 문제 해결기법에서의 전통적인 접근 방식은 미리 정의된 규약에 의존한 조건-행동 추론 방식이다. 조건-행동 추론 방식은 주어진 상황에 빠르게 반응하지만, 문제 영역이 다양하거나 문제의 재설계가 필요한 경우에는 아무런 해법을 갖지 못한다. 따라서 이러한 문제점을 해결하기 위해 본 논문에서는 주어진 행동들의 유틸리티를 실시간에 계산하고, 긴급한 정도(urgency)를 측정하여 상황이 긴급할 경우에는 더 이상의 계산을 중단하고 즉각적인 행동을 취하며, 반면에 상황이 긴급하지 않을 경우에는 최선의 의사결정을 위하여 추가적인 정보를 고려하여 더 바람직한 행동을 결정하는 방법론을 제안한다. 제안한 방법론의 평가를 위하여 시간 제약적인 환경에서 최선의 의사결정을 수행하는 실질적이며 유연한 에이전트를 구현하고자 한다.

키워드 : 의사결정, 자율적인 에이전트, 긴급한 정도, 지대공 방어 시스템

Abstract Autonomous agents need considerable computational resources to perform rational decision-making. The complexity of decision-making becomes prohibitive when large number of agents are present and when decisions have to be made under time pressure. One of approaches in time-critical domains is to respond to an observed condition with a predefined action. Although such a system may be able to react very quickly to environmental conditions, predefined plans are of less value if a situation changes and re-planning is needed. In this paper we investigate strategies intended to tame the computational burden by using off-line computation in conjunction with on-line reasoning. We use performance profiles computed off-line and the notion of urgency (i.e., the value of time) computed on-line to choose the amount of information to be included during on-line deliberation. This method can adjust to various levels of real-time demands, but incurs some overhead associated with iterative deepening. We test our framework with experiments in a simulated anti-air defense domain. The experiments show that the off-line performance profiles and the on-line computation of urgency are effective in time-critical situations.

Key words : decision-making, autonomous agents, urgency, anti-air defense system

1. 서 론

실시간 환경에서 항상 최적의 행동을 수행하는 완벽하게 이성적인 에이전트(rational agent)의 구현은 실질적으로 가능하지 않다. 이러한 실시간 문제 해결기법에

서의 전통적인 접근방식은 미리 정의된 규약에 의존한 조건-행동 추론방식이다[1,2]. 조건-행동 추론방식은 주어진 조건에 빠르게 반응하지만, 문제 영역이 다양하거나 문제의 재설계가 필요한 경우에는 아무런 해법을 가지지 못하는 단점이 있다. 특히, 시스템이 조건-행동에 대한 규약 자체를 실시간 이전에 정의하지 못한다면, 이러한 방식을 사용하는 에이전트들은 동적인 환경에서 어떠한 의사결정도 수행하지 못할 것이다. 이러한 문제점을 해결하기 위해 실시간 내에 가장 적절한 행동을

* 이 논문은 2002년도 가톨릭대학교 정착연구비, 2003년도 한국학술진흥재단 (KRF-2003-041-D20465)의 지원에 의하여 연구 되었음

^{*} 종신회원 : 가톨릭대학교 컴퓨터정보공학부 교수
sunoh@catholic.ac.kr

논문접수 : 2004년 3월 9일
심사완료 : 2004년 6월 17일

계산하는 제한적 최적화(bounded optimality) 이론이 제안되어 왔다[3-5].

일반적으로 실시간 환경에서 자율적인 에이전트는 많은 양의 정보 또는 자원을 처리할 수 없기 때문에, 의사결정(decision-making)을 위하여 유용한 정보를 충분히 검토하지 못하는 경우가 발생 한다. 이와 같이 에이전트가 계산에 필요한 자원들을 모두 포함하여 의사결정을 하지 못할 경우, 어떻게 행동 할 지에 대한 결정이 요구된다. 따라서 시간 제약적인 환경에서 긴급한 정도(urgency)에 따라 적절하게 행동하기 위하여, 추가적인 정보를 고려하기 위해 사용하는 계산시간에 대한 비용(cost of computation)과 추가 정보를 고려하여 얻은 이득(benefit of computation)에 대한 비교가 필요하다. 복잡한 실시간 환경에서 효율적으로 행동하는 자율적인 에이전트를 구현하기 위해서, 본 논문에서는 주어진 상황에 대한 긴급한 정도를 계산하기 위한 방법론을 제안한다. 구체적으로 제안한 방법론의 평가를 위하여, 지대공 방어 환경에서 긴급한 정도에 따라 유연한 의사결정을 수행하는 자율적인 에이전트를 구현한다.

본 논문에서 제안하는 접근방식은 실시간 분산 시스템에서 사용되는 전통적인 방식들과 비교할 때 확실한 차이점을 나타낸다. 기존의 접근 방식[6]은 시스템 설계자에 의해 소프트웨어 에이전트의 이성적인 행동들을 위하여 고려되는 계산자원(computational resources)이 결정되었으며, 에이전트의 의사결정에 필요한 시간의 중요성을 나타내는 방법이 존재하지 않았다. 따라서 기존의 시스템은 계산자원의 지나친 단순화 때문에 최적의 결정을 할 수 없으며, 긴급한 정도에 대한 정보가 없기 때문에 종료시점 이전에 의사결정을 수행하지 못할 수도 있다. 가장 중요한 본 논문의 독창성은 에이전트가 처한 상황의 긴급한 정도 또는 시간의 값(the value of time)을 계산하며, 동적인 도메인에서 긴급한 정도에 따라 빠르게 대처할 수 있도록 한다는 것이다.

다음 장에서는 본 논문에서 제안하는 접근방식과 유사한 접근방식과의 관련연구에 대하여 정리하며, 3장에서는 긴급한 정도를 정의하고 이를 측정하는 방법을 정형화하며, 4장에서는 시간이 중요한 도메인으로써 지대공 방어 시스템을 소개한다. 5장에서는 제안한 방법론이 지대공 방어 도메인에서 어떻게 구현되었는가를 구체적으로 설명하고 실험결과를 보여준다. 결론에서는 시간이 중요한 경우의 자율적인 에이전트의 의사결정에 대한 기여도를 정리하며, 앞으로의 연구방향을 제시한다.

2. 관련 연구

주어진 실시간 환경에서 최선의 의사결정을 수행하기 위하여, 적절한 계산자원을 긴급한 정도에 따라 할당하는 추론능력에 대한 해법으로 제한적 최적화(bounded optimality) 이론이 제안되어왔다[3-5]. 이러한 연구들의 대부분은 의사결정에 소요되는 시간을 고려한 의사결정의 질적인 수준을 표현하고자 노력하였다. 즉, 의사결정에서 얻어지는 성취도의 실질적인 값은 얻어진 이득으로부터 의사결정을 위하여 소비한 계산비용(computational cost)을 제외한 값이 된다는 것이다. 그러나, 대부분의 관련 연구에서는 계산비용의 정형화 대신에 주어진 환경에서 미리 정해진 벌칙(penalty) 값으로 가정하였으며, 본 논문에서는 이러한 계산비용을 결정이론(decision theory)적인 입장에서 예상 유틸리티 값으로 나타내고자 한다.

시간 제약적인 환경에서 제한적 최적화 이론의 구현을 위하여는 긴급한 상황에 대한 실질적인 표현 뿐만 아니라 어떠한 시점에서도 에이전트의 의사결정을 수행할 수 있는 부분적인 해결책(partial results)을 제공할 수 있어야 한다[3,7]. 부분적인 해결책을 제공하는 대부분의 연구들은 고려하는 행동들의 개수를 제한하는 플래닝 기법(planning techniques)을 사용하지만, 본 논문에서는 에이전트가 고려하는 계산자원에 따라 다른 질적인 수준을 나타내는 행동을 결정하는 방법을 제안한다. 다시 말하면, 기존의 접근 방식과 본 논문에서 제안하는 접근 방식 모두 공통적으로 에이전트에게 어떠한 시점에서도 환경에 반응할 수 있는 부분적인 해결책을 제공할지만, 각각을 생성하는 방식에는 차이가 있다. 우리의 접근 방식은 에이전트가 상호협력하는 멀티 에이전트 환경(multi-agent domains)에서 각각의 에이전트가 의사결정을 위하여 고려하는 모델링 공간의 깊이와 너비를 반복적으로 증가 시키면서 부분적인 해결책을 생성하며, 이러한 결과를 성취정도 프로파일(performance profile)[4,7,8]의 형태로 저장한다.

3. 긴급한 정도의 계산

긴급한 상황에서 에이전트의 의사결정이 이루어져야 할 경우 시간의 값(value of time)을 나타내는 것은 매우 중요하다. 시간의 중요한 정도를 나타내려는 연구는 인공지능 연구 분야[7]에서 수행되어 왔지만, 본 논문에서는 결정이론(decision theory)적인 입장에서 새롭게 정의하고자 한다.

정리 1. 긴급한 정도(urgency)는 에이전트가 주어진 상황에서 아무런 행동을 하지 않을 경우 시간의 경과에 따라 에이전트가 잃어버리는 예상 유틸리티(expected

1) 긴급한 정도는 주어진 상황에서 시간의 중요한 가치(value of time)로 해석된다. 경제학적인 측면에서는 기회비용(opportunity cost)으로도 불린다.

utility) 값으로 정의한다.

예를 들어, 지대공 방어 환경(그림 1 참조)[9]에서 방어 에이전트가 공격해오는 비행체를 격추시키려는 노력을 하지 않는다면, 시간이 경과함에 따라 근접하는 비행체를 제거할 확률이 적어지기 때문에 방어지역에 가해지는 위협이 증가할 확률이 높아지게 된다. 이와 같이, 긴급한 정도 - 시간의 값 - 는 에이전트가 아무런 행동을 하지 않을 경우 손실하는 예상 유틸리티(expected utility: EU)[3]의 시간적인 비율로 정의할 수 있다.

예상 유틸리티의 계산을 위하여, 이 장에서는 다음과 같은 표기법을 사용한다.

- 에이전트의 집합: $N = \{n_1, n_2, \dots\}$.
- 에이전트 n_i , $n_i \in N$ 의 행동들에 대한 집합: $A_{n_i} = \{a_i^1, a_i^2, \dots\}$.
- 가능한 상태들의 집합: $S = \{s_1, s_2, \dots\}$.

특정한 시간 t 에 주어진 환경과 다른 에이전트에 대한 지식(또는 정보) E 를 사용하여 에이전트 n_i 가 획득한 최적의 행동 a 에 대한 예상 유틸리티(expected utility: EU)는 다음과 같이 구한다.

$$EU(a^E, t) = \max_{a_i^j \in A_{n_i}} \sum_k P(s_k | E, t, a_i^j) U(s_k) \quad (1)$$

위의 식 (1)에서,

- $P(s_k | E, t, a_i^j)$ 는 주어진 지식 E 를 가지고 획득된 행동 a_i^j 가 시간 t 에 수행되어진 후 특정한 상태 s_k 에 도달할 확률이며,
- $U(s_k)$ 는 상태 s_k 의 유틸리티 값을 나타낸다.

에이전트가 의사결정을 위하여 고려할 정보의 양, 의사결정에 사용한 계산시간, 그리고 결과적으로 얻은 의사결정의 질적인 수준에 대한 상관관계를 나타내기 위하여, 각기 다른 양의 정보를 이용하여 최적의 행동을 결정하는 예상 유틸리티 값을 비교한다. 주어진 특정한 시간에 최적의 행동을 결정한 에이전트는 즉각적으로 행동을 수행하거나 조금 더 바람직한 행동을 얻기 위해 정보를 추가하여 의사결정을 계속 진행할 수 있다. 현재 이용 가능한 정보의 양을 E 라 하고 추가적으로 고려할 정보를 e 라 하면 시간 $t+t'$ 에 수행할 최적의 행동은 a 이며, 추가정보를 처리하여 더 바람직한 행동을 얻으려 할지 또는 현재 시점의 최적의 행동을 수행할지의 결정은 다음과 같이 표시된다.

$$EU(a^{E+e}, t+t') > EU(a^E, t) \quad (2)$$

위의 식에서 더 많은 정보(e)를 고려하여 의사결정의 질적인 수준은 높지만 그 만큼의 정보를 처리하기 위하여 지연된 행동에 대한 예상 유틸리티($EU(a^{E+e}, t+t')$)가 현재 최적의 행동에 대한 예상 유틸리티($EU(a^E, t)$)보다 크면, 에이전트는 최적의 의사결정을 위하여 더 많은

계산 시간을 소모할 것이다. 이러한 결정은 a^{E+e} 의 계산 전에 실질적으로 이루어져야 한다. 본 논문에서 제안하는 접근방식은 실시간 이전에 성취정도 프로파일(performance profile)[4,7,8]을 생성하고, 이를 바탕으로 현재 최적의 행동을 수행할지 또는 계속 생각할지를 결정하게 된다. 성취정도 프로파일을 이용하여 현재시점 t 에서의 더 많은 정보(e)를 고려하여 얻은 행동에 대한 예상 유틸리티를 예측할 수 있다. 다시 말하면, 시간 $t+t'$ 에서의 실질적인 a^{E+e} 의 예상 유틸리티를 계산하는 것에 대한 의미가 없기 때문에, 현재 시점 t 에서의 예측 가능한 a^{E+e} 를 다음과 같이 예측 값으로 사용한다.

$$EU(a^{E+e}, t+t') = EU(a^{E+e}, t) - [EU(a^E, t) - EU(a^E, t+t')] \quad (3)$$

이와 같이, 지연된 행동 a^{E+e} 는 현재 최적의 행동 a^E 와 같은 비율로 시간이 경과함에 따라 예상 유틸리티 값이 감소한다고 가정한다. 시간 $t+t'$ 에서의 최적의 행동 a^{E+e} 는 현재 시점 t 에서 성취정도 프로파일로부터 예측 가능한 행동 a^{E+e} 로부터 현재 최적의 행동 a^E 를 지연시키기 때문에 감해지는 예상 유틸리티 값으로 해석한다. 식 (3)을 식 (2)에 적용하여 정리하면 다음을 얻는다.

$$EU(a^{E+e}, t) - EU(a^E, t) > EU(a^E, t) - EU(a^E, t+t') \quad (4)$$

식 (4)의 부등호 왼쪽 부분은 현재 이용 가능한 정보(E)를 이용하여 획득한 최적의 행동과 추가적으로 고려할 정보(e)를 이용하여 획득한 최적의 행동에 대한 각각의 예상 유틸리티 값의 차로 표현되는 계산 이득(benefit of computation)을 나타낸다. 즉, 추가적인 정보를 활용하여 지속적인 계산을 수행할 경우에 얻어질 수 있는 의사결정 수준의 향상을 표시하며, 각각의 예상 유틸리티 값은 미리 측정해 놓은 성취정도 프로파일로부터 얻어진다. 또한, 식 (4)의 부등호 오른쪽 부분은 긴급한 정도에 대한 시간의 값(value of time)이 감소하는 것을 나타낸다.²⁾ 다시 말하면, 현재까지의 정보를 이용하여 얻은 최적의 행동을 수행하지 않고, 추가적인 정보를 처리하기 위하여 계산하는 동안, 그 행동을 미루기 때문에 발생하는 예상 유틸리티 값의 손실을 의미한다. 이러한 예상 유틸리티 값의 손실은 계산 비용(cost of computation)을 말하며, 에이전트가 처한 상황에 의존하여 실시간에 계산된다.

의사결정을 수행하는 에이전트는 성취정도에 대한 프로파일을 바탕으로 t' 만큼의 시간을 사용해 얻은 예상 유틸리티의 증가량이 t' 동안의 현재 예상 유틸리티에

2) 예상 유틸리티를 시간에 대하여 감소하는 비율로 나타내는 것도 가능하다. 여기서는 간단한 형태로 나타내고자 한다.

대한 감소량 보다 크면(식 (4)의 경우), 현재보다 더 좋은 성취도를 얻기 위해 t' 만큼의 시간을 사용하여 추가 정보를 얻어온다. 반면에, 에이전트가 t' 만큼의 시간을 사용해 얻은 예상 유틸리티의 증가량이 t' 동안의 현재 예상 유틸리티에 대한 감소량 보다 작으면(식 (4)의 반대 경우) - 다시 말해서, 계산에 의한 이득 보다 계산 비용이 더 크다고 판단하면 - 더 이상의 추가 정보를 얻지 않는다. 이 경우, 더 이상의 추가정보를 고려하지 않고, 현재까지의 정보를 바탕으로 이미 결정한 최적의 행동을 수행하게 된다. 일반적으로, 주어진 상황이 긴급하지 않으면 유틸리티의 시간에 대한 손실이 적을 것이고 에이전트는 더 바람직한 의사결정을 위하여 추가정보를 처리할 것이며, 반면에, 주어진 상황이 매우 긴급하다면 유틸리티의 시간에 대한 손실이 상대적으로 크기 때문에 현재 결정된 최적의 행동을 시간의 지체 없이 수행할 것이다.

4. 시간이 중요한 동적 환경

제한한 방법론을 적용하기 위하여 시간 제약적인 환경으로 지대공 방어 시스템을 모델화한다. 지대공 방어 시스템은 실시간 환경에서 시간의 긴급한 정도에 따라 적절한 행동이 요구되며, 긴급한 상황에서의 의사결정이 이루어지지 않은 경우에는 시스템의 손실 또는 적의 공격에 의한 비정상적인 동작을 가져온다.

지대공 방어 시스템(anti-air defense system) 모델은 그림 1에 나타난 것과 같이 일련의 공격해오는 목표물(target)들과 상호 협력하여 목표물을 제거하는 방어

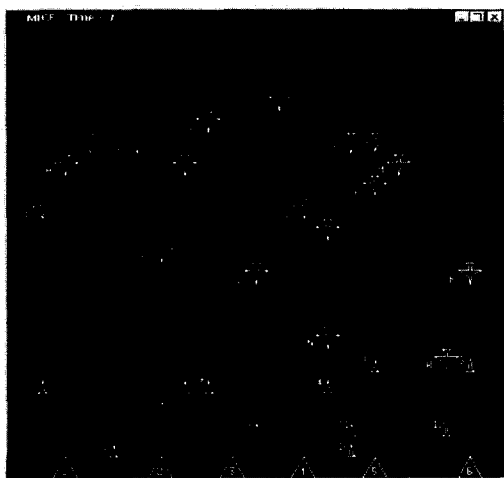


그림 1 복잡한 지대공 방어 시스템 시나리오. 화면 하단에 6개의 지대공 방어 에이전트와 A부터 R까지 18개의 공격해오는 목표물들로 구성되어 있다.

에이전트(defense agent)들로 구성되어 있다[9]. 방어 에이전트의 임무는 자신의 영역에 피해를 최소화하기 위하여 공격해오는 목표물들을 요격하는 것이다. 이때, 에이전트들은 상호 협력이 필요한 데 만약 다른 에이전트의 고려 없이 독자적인 의사결정을 하여 동일한 목표물을 제거하려 한다면 다른 위협들이 방어하는 지역에 도달할 것이고, 이러한 피해는 탄두의 크기에 비례하여 증가하게 된다.

방어 에이전트들은 특정한 방어 상황을 결정이론적인 형태로 분석한다. 즉, 에이전트가 고려하는 여러 개의 행동들 중에서 최적의 행동을 선택하기 위한 지침으로써 속성들을 확인한다. 첫째로, 각 목표물은 특정 값의 위협을 가진다. 이 값은 목표물의 탄두 크기를 나타낸다. 따라서 방어 에이전트들은 다른 위협보다 큰 위협을 먼저 제거하려고 할 것이다. 둘째로, 방어 에이전트들은 공격해오는 목표물들을 보다 효율적으로 요격할 수 있는 확률을 고려한다. 요격 확률³⁾ $P(H_{ij})$ 은 목표물 j 의 진행방향과 방어 에이전트 n_i 가 목표물을 바라보는 직선 방향 사이의 각 γ_{ij} 에 의존한다[10]. 요격 확률은 다음과 같다.

$$P(H_{ij}) = e^{-\mu \gamma_{ij}} \quad (5)$$

이때, μ 는 방어환경에 의존하는 특정한 상수로 가정하며, 실험에서 0.01로 사용한다.

5. 지능형 에이전트의 성취정도

다른 에이전트와 상호협력(coordination)하며 최적의 의사결정을 수행하는 에이전트는 그림 2와 같은 모델링 공간을 가진다[11,12]. 본 논문에서 제안하는 반복 알고리즘(iterative algorithm)은 에이전트 모델링 공간의 깊이와 너비를 반복적으로 증가시키면서 에이전트의 의사결정에 대한 질적 수준을 높인다.

반복 알고리즘은 다른 에이전트들의 모델링 없이 - 이것은 독립적인 의사결정을 수행하는 방식으로 불린다. - 자신의 의사결정 상황을 바탕으로 행동을 결정하는 것으로 출발한다(그림 2의 루트(root)인 에이전트 1의 의사결정 상황에 해당). 그림 2에서 에이전트 1은 자신의 의사결정을 위하여 다른 에이전트와 상호협력 할 수 있으며, 예를 들어, 에이전트 1은 에이전트 2를 모델링한다(그림 2의 에이전트 1의 의사결정 상황과 에이전트 1의 에이전트 2에 대한 모델로 구성). 에이전트 1은 자신의 의사결정을 위하여 다른 에이전트들을 고려하며, 계속하여 다른 에이전트들은 각각의 의사결정을 위하여 또다시 다른 에이전트들을 고려할 수 있다(그림 2의 에이

3) 요격 확률은 목표물을 격추시킬 확률(probability of kill)이며, 본 논문에서는 경험적인 요소에 기초하여 산출한다.

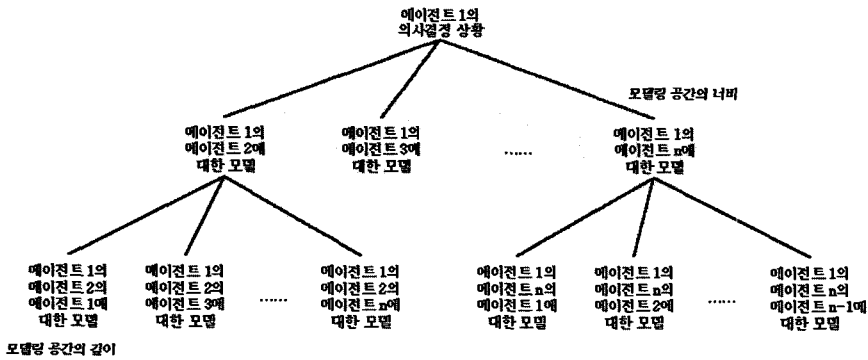


그림 2 에이전트 1의 의사결정을 위한 모델링 공간. 모델링 공간은 다른 에이전트를 고려하는 정도를 나타내는 모델링 공간의 너비와 다른 에이전트 또한 계속하여 모델링할 수 있음을 나타내는 모델링 공간의 깊이로 구성된다.

이전트 1의 의사결정 상황, 에이전트 1의 에이전트 2에 대한 모델, 및 에이전트 1의 에이전트 2의 에이전트 1에 대한 모델로 구성). 이와 같이 에이전트들은 반복 알고리즘을 이용하여, 에이전트 모델링 공간의 깊이 (the depth of modeling space)를 증가시키며, 결과적으로 각각의 의사결정에 대한 질적 수준을 향상시킨다. 이와 유사한 방식으로, 에이전트들의 상호협력을 고려하는 의사결정을 위하여 에이전트 모델링 공간의 또다른 축인 모델링 공간의 너비(the breadth of modeling space)를 확장시켜 나갈 수 있다. 에이전트 모델링 공간의 너비는 에이전트가 다른 에이전트를 고려하는 정도를 나타낸다. 그림 2에서 에이전트 1은 자신의 의사결정을 위하여, 예를 들어, 에이전트 2와 에이전트 3을 모델링 할 수 있다 (그림 2의 에이전트 1의 의사결정 상황, 에이전트 1의 에이전트 2에 대한 모델, 및 에이전트 1의 에이전트 3에 대한 모델로 구성). 이러한 모델링 과정은 각 에이전트의 의사결정을 위하여 계속 되어질 수 있으며, 반복 알고리즘을 이용하여 모델링 공간의 깊이와 너비를 반복적으로 증가시켜 나갈 수 있다.

이 장에서는 의사결정 소요시간이 의사결정의 질적인 수준에 영향을 미치는 지대공 방어환경에서, 반복 알고리즘을 사용하는 방어 에이전트의 모델링 공간을 증가시키면서 에이전트의 성취정도를 측정하고, 이에 대한 각 에이전트의 성취정도 프로파일(performance profile)을 작성한다. 성취정도 프로파일을 바탕으로, 에이전트는 실시간에 주어진 상황의 긴급한 정도를 계산하고, 이에 따라 적절한 모델링 정보의 양을 결정함으로써 유연한 의사결정을 수행하게 된다.

5.1 지대공 방어 에이전트의 성취정도 프로파일

반복 알고리즘을 사용하는 지대공 방어 에이전트의

성능을 오프라인에 측정하여 성취정도 프로파일의 형태로 저장한다. 성취정도 프로파일은 온라인에 에이전트가 처한 상황의 긴급한 정도를 계산하기 위하여 사용된다. 지대공 방어 에이전트의 목표(goal)는 4장에서 언급한 바와 같이 에이전트들이 방어하는 지역의 피해를 최소화하기 위하여 공격해오는 목표물들을 제거하는 것이다. 따라서, 지대공 방어 환경에서 에이전트의 성취정도는 방어지역에 가해지는 위협(threat) 또는 피해(damage)를 제거한 양으로 나타내었다. 실험에 사용한 지대공 방어 환경은 다음과 같다:

- 6개의 방어 에이전트와 6개의 공격해오는 목표물로 구성;
- 목표물의 위치는 좌표값 0부터 10까지 범위에서 생성;
- 목표물의 위협정도는 100부터 300에서 임의로 설정.

실험에서 각 방어 에이전트의 모델링 공간의 깊이는 1에서 4까지였으며, 모델링 공간의 너비는 2부터 6까지였다. 반복 알고리즘은 가장 간단한 모델링 공간인 2개의 다른 에이전트와 추론 깊이 1부터 시작하여, 점진적으로 5개의 다른 에이전트를 모두 고려하며 추론 깊이 4까지 포함하는 모델링 공간을 사용하였다. 이와 같은 지대공 방어환경에서 제한한 반복 알고리즘을 이용하여 측정된 방어 에이전트의 성취정도와 의사결정을 위하여 소요된 계산시간을 표 1에 정리하였다.⁴⁾ 표 1의 실험결과는 100개의 지대공 방어 시나리오에서 각각의 방어 에이전트의 평균 성취정도와 계산시간을 나타낸다. 지대공 방어 에이전트가 자신의 의사결정을 위하여 다른 에이전트를 모델링하는 경우의 성취정도에 대한 비교를 위하여, 임의로 목표물을 선택하는 에이전트(random

4) 지대공 방어 에이전트의 실험을 위하여 1.6 GHz Pentium IV single-CPU 장비를 사용하였다.

표 1 임의로 목표물을 선택하는 방식(random), 독립적인 의사결정을 수행하는 방식(independent), 및 반복 알고리즘을 사용하여 의사결정을 수행하는 에이전트의 성취정도와 계산시간

의사결정 방식	성취정도	계산시간
Random	516.2±126.6	0.013±0.007
Independent	527.9±175.3	0.017±0.005
2 with depth 1	809.7±105.1	0.040±0.015
2 with depth 2	816.7±108.9	0.056±0.020
2 with depth 3	841.4±82.1	0.058±0.021
2 with depth 4	859.9±57.9	0.062±0.043
3 with depth 1	906.2±98.3	0.364±0.047
3 with depth 2	1001.3±60.2	0.374±0.053
3 with depth 3	1050.8±46.3	0.381±0.051
3 with depth 4	1055.0±47.9	0.373±0.054
4 with depth 1	1055.2±48.9	3.036±0.380
4 with depth 2	1055.4±48.0	3.129±0.501
4 with depth 3	1055.7±50.8	3.168±0.556
4 with depth 4	1055.9±47.4	3.202±0.570
5 with depth 1	1057.2±48.0	32.533±11.064
5 with depth 2	1057.4±46.6	38.400±7.026
5 with depth 3	1057.7±46.3	37.675±6.813
5 with depth 4	1058.1±49.0	37.975±7.108
6 with depth 1	1060.6±45.3	144.776±97.834
6 with depth 2	1060.8±44.4	691.043±166.245
6 with depth 3	1061.2±44.3	689.943±164.606
6 with depth 4	1061.5±44.0	692.443±164.804

agent)와 다른 에이전트의 모델링 없이 독립적으로 의사결정을 수행하는 에이전트(independent agent)의 성취정도 또한 측정하였다.

임의로 목표물을 선택하는 에이전트는 가장 나쁜 성취정도(516.2)를 보였지만, 가장 적은 계산시간(0.013)을 사용하였다. 독립적으로 의사결정을 수행하는 에이전트의 성취정도(527.9)와 계산시간(0.017)은 임의로 목표물을 선택하는 에이전트와 유사하였으며, 이는 다른 에이전트의 고려없이 자신의 예상 유틸리티 값을 최대화 함으로써 상호협력에 실패하였기 때문으로 판단된다. 반면에, 하나의 다른 에이전트를 모델링하며 자신의 추론 깊이 1을 가지는 에이전트⁵⁾의 성취정도(809.7)는 독립적으로 의사결정을 수행하는 에이전트와 비교할 때 58%의 성능향상을 보였다. 따라서, 에이전트의 상호협력력을 위하여 에이전트간의 모델링이 필수적임을 알 수 있었다.

표 1에서 나타난 바와 같이, 에이전트가 의사결정을 위하여 고려하는 모델링 공간의 깊이와 너비가 증가됨에 따라 에이전트의 성취정도는 향상되었으며 이를 위한 의사결정 소요시간은 증가하였다. 상호협력력을 위하여 고려하는 에이전트의 수가 2와 3일 때, 성취도는 추론 깊이에 비례하여 점진적으로 향상되었다. 즉, 인접한 하

나의 에이전트를 고려할 경우 방어지역에 가해지는 위협을 제거한 정도는 809.7에서 859.9로 향상되었으며, 인접한 두개의 에이전트를 모델링 할 경우의 성취도는 906.2에서 1055.0으로 증가하였다. 그러나, 모델링을 위하여 더 많은 에이전트와 추론 깊이가 포함된다 하더라도 성취정도 결과는 거의 동일하였다. 세 개의 다른 에이전트를 고려하며 모델링 깊이가 1인 경우와 다섯 개의 다른 에이전트를 고려하며 모델링 깊이가 4인 경우의 에이전트 성취정도의 차이는 대략 6(1061.5 - 1055.2) 정도로 나타났다. 지대공 방어 환경에서 방어 에이전트는 인접한 두개의 에이전트와 추론 깊이가 4인 경우(이는 표 1에서 3 with depth 4에 해당)에 이미 최적의 성취도에 가까운 결과를 나타냄을 알 수 있었으며, 더 많은 정보가 모델링을 위하여 추가되어도 최적의 의사결정에 큰 영향을 주지 않았음을 결과적으로 보여준다.

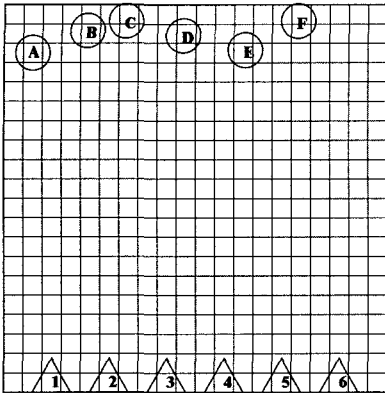
모델링 공간의 깊이와 너비가 증가됨에 따라 지대공 방어 에이전트는 더 많은 계산시간을 사용한다. 지대공 방어 시스템이 Lisp 프로그래밍 언어로 구현되었기 때문에 측정된 평균시간의 편차가 발생하였으나, 표 1에서 모델링 공간의 깊이별 계산시간을 정리하면 각각 다음과 같다: [0.040-0.062], [0.364-0.381], [3.036-3.202], [32.533-38.400]. 실험에서 가장 복잡한 모델링 공간을 사용하는 경우, 즉, 다섯 개의 다른 에이전트를 모델링하는 경우, 에이전트의 의사결정을 위한 계산시간은 144.776(추론깊이 1)이 소요되었으며, 추론 깊이가 증가할 경우 계산해야하는 각각의 의사결정 상황이 입체적인 육면체에 해당하고 다섯 개의 다른 육면체를 계산해야하기 때문에 계산시간은 691.043(추론깊이 2)으로 급격히 증가하였다.

5.2 지대공 방어 에이전트의 긴급한 정도에 대한 계산

3장에서 언급한 긴급한 정도의 계산을 여기서는 구체적인 지대공 방어 환경에 적용하고, 실시간에 적절한 모델링 정보의 양을 결정하는 에이전트의 예를 보인다.

그림 3과 같이 나타난 지대공 방어 환경을 고려하자. 화면하단에 6개의 방어 에이전트가 존재하며, 화면상단에 6개의 공격해오는 목표물이 존재한다. 목표물들은 1초에 2개의 격자를 이동한다고 가정하며, 이 예제에서는 지대공 방어 에이전트 1의 의사결정 상황을 보여준다. 에이전트 1은 임의로 6개의 목표물중 하나를 선택하며, 이를 위하여 0.013초가 소요된다(표 1 참조). 에이전트 1은 3장의 식 (4)에 서술된 바와 같이, 계산 이득과 계산 비용을 고려한다. 우선, 에이전트 1이 가명 임의로 목표물 D를 선택하였을 때의 계산 비용을 보인다. 현재 시간 t 에서, 에이전트 1의 목표물 D에 대한 요격 확률 (식 (5) 참조)은 다음과 같다.

5) 이 에이전트는: 표 1에서 2 with depth 1에 해당한다.



Targets: A B C D E F
Positions: (2,3) (5,2) (7,1) (10,2) (13,3) (16,1)
Batteries: 1 2 3 4 5 6
Positions: (3,20) (6,20) (9,20) (12,20) (15,20) (18,20)

그림 3 긴급한 정도의 계산을 위한 예제 시나리오. 6개의 목표물(A부터 F까지)과 6개의 지대공 방어 에이전트(1부터 6까지)가 존재한다.

$$\text{At time } t, P(H_{1D}) = 0.979008$$

1초 후에 목표물 D는 방어지역을 향하여 두 격자 아래로 가까이 오며, 방어 에이전트와 목표물 D사이의 각도는 21도에서 23도로 증가하며, 이때의 적중확률 또는 요격확률은 다음과 같이 감소한다.

$$\text{At time } t+1, P(H_{1D}) = 0.976680$$

목표물 D의 위협정도(threat)는 250으로 주어졌으며, 다음 단계의 전략 - 표 1에서 독립적인 의사결정을 수행하는 방식(Independent) - 을 위하여 0.017초의 계산 시간이 필요하므로, 이때의 계산 비용은 다음과 같이 구할 수 있다.

$$EU(\alpha^E, t) - EU(\alpha^E, t+0.017) = (0.979008 - 0.976680) \times 250 \times 0.017 = 0.009894$$

표 1에 나타난 성취정도 프로파일(performance profile)을 바탕으로, 계산 이득(식 (4)의 부등호 좌측 참조)은 다음과 같이 구한다.

$$EU(\alpha^{E+t}, t) - EU(\alpha^E, t) = 527.9 - 516.2 = 11.7$$

따라서, 이 예제 시나리오에서는 계산 이득(11.7)이 계산 비용(0.009894)보다 크므로 에이전트는 목표물 D를 적중시키려는 즉각적인 행동대신에 추가적인 정보를 처리하고자 할 것이다. 이러한 긴급한 정도의 계산은 계산 이득이 계산 비용보다 작게 될 때까지 계속될 것이다. 실질적으로 에이전트가 계산을 중지하고 결정된 행동을 수행한 시점은 세 개의 에이전트와 추론 깊이 3(3 with depth 3)으로 판명되었다. 그러나 주어진 상황에서 목표물이 더욱 빠르게 방어지역을 향하여 이동하자

나 계산하는 속도가 느렸다면, 의사결정을 위한 모델링 범위가 줄어들게 되며 더 빠른 의사결정이 필요했을 것이다. 일반적으로, 긴급한 정도를 계산하는데 영향을 미치는 요소들은 자료처리 능력과 주어진 상황의 긴급함을 결정하는 외부요인들을 포함한다.

5.3 지대공 방어환경에서의 유연한 의사결정

시간 제약적인 환경에서 유연한 의사결정 방법론을 평가하기 위하여, 6개의 방어 에이전트와 6개의 공격해오는 목표물이 존재하는 시나리오에서 에이전트가 긴급한 정도를 계산하여 의사결정을 수행한 결과를 측정하였다. 표 1에 나타난 성취정도 프로파일을 바탕으로 100개의 지대공 방어 시나리오에서 긴급한 정도를 계산한 결과를 표 2에 나타내었다.

표 2 지대공 방어환경에서 유연한 의사결정을 위한 모델링 공간의 평균 너비와 평균 깊이

모델링 공간의 평균 너비	모델링 공간의 평균 깊이
3.000 ± 0.142	3.010 ± 0.225

표 2에서와 같이 방어 에이전트가 고려한 모델링 공간의 평균 너비는 3.000이었으며, 평균 깊이는 3.010이었다. 즉, 에이전트는 주어진 100개의 시간 제약적인 상황에서, 방어 에이전트의 자율적인 의사결정을 위하여 평균적으로 다른 두개의 에이전트를 모델링하였으며, 추론 깊이는 3을 약간 초과하였다. 지대공 방어 환경에서 임의로 생성한 목표물의 속도와 위협정도에서 에이전트는 계산 비용과 계산 이득을 비교하여 적절한 범위의 모델링 정보를 결정할 수 있었으며, 주어진 환경에서 최적의 의사결정을 수행할 수 있었다.

6. 결론

지대공 방어 시스템과 같은 시간 제약적인 환경에서, 자율적인 에이전트는 주어진 조건에서 최적의 의사결정을 필요로 한다. 본 논문에서 제안한 방법론은 의사결정 시간이 매우 중요한 경우에 여러 에이전트들이 상호협력하며 제한적인 최적화(bounded rationality)[4,5]를 달성할 수 있도록 하였다. 또한, 오프라인에 획득한 에이전트의 성취정도 프로파일(performance profile)을 바탕으로, 결정이론적인 측면에서 긴급한 정도를 정형화하였다. 에이전트가 행동할 것인지 더 생각할 것인지를 결정하기 위하여 성취정도 프로파일에 포함된 정보와 긴급한 정도의 계산을 결합한 식을 유도하였다.

제안한 방법론을 지대공 방어 시스템에 적용하였으며, 각각의 시뮬레이션 환경에서 자율적인 에이전트의 성취도를 측정하였다. 실험에서 각 에이전트는 시간 제약적인 환경에서 제한적인 최적화를 어떻게 달성하는가를

보여주었다. 본 논문을 통하여 자율적인 에이전트의 유연하며 실질적인 의사결정을 향한 목표에 한 걸음 다가서는 계기가 될 수 있기를 바란다. 앞으로 에이전트의 지식을 표현하는 일반적인 셸(shell)을 개발하여 자율적인 에이전트를 보완할 것이며, 제안한 방법론을 다양한 시간 제약적인 환경 - 예를 들면, 무선 이동 환경 등 - 에 적용할 것이다.

참 고 문 헌

- [1] Agre, P.E. and Chapman, D., Pengi: An implementation of a theory of activity, In Proceedings of the National Conference on Artificial Intelligence, Seattle, Washington, pp. 268-272, 1987.
- [2] Fox, J. and Krause, P., Symbolic decision theory and autonomous systems, In Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence, UCLA, California, pp. 103-110, July 1991.
- [3] Russell, S. J. and Wefald, E. H., Principles of Metareasoning, Journal of AI, Volume 49, pp. 361-395, 1991.
- [4] Russell, S.J. and Subramanian, D., Provably bounded-optimal agents, Journal of Artificial Intelligence Research, Volume 2, pp. 575-609, 1995.
- [5] Russell, S.J., Rationality and Intelligence, Artificial Intelligence, Volume 94, pp. 57-77, 1997.
- [6] Rao, A.S. and Georgeff, M.P., BDI agents: From theory to practice, In Proceedings of the 1st International Conference on Multiagent Systems, pp. 312-319, July 1995.
- [7] Dean, T., Decision-theoretic planning and markov decision processes, Technical Report, Brown University, Computer Science Department, Providence, RI, 1994.
- [8] Zilberstein, S. and Russell, S.J., Optimal composition of real-time systems, Artificial Intelligence, vol. 82, no. 1, pp. 181-213, 1996.
- [9] Noh, S. and Gmytrasiewicz, P.J., Uncertain Knowledge Representation and Communicative Behavior in Coordinated Defense, Lecture Notes in Artificial Intelligence 1916, Issues in Agent Communication, pp. 281-300, Springer, 2000.
- [10] Macfadzean, R.H.M., Surface-Based Air Defense System Analysis, Artech House, 1992.
- [11] Gmytrasiewicz, P.J. and Durfee, E.H., Rational coordination in multi-agent environments, Autonomous Agents and Multiagent Systems Journal, vol. 3, pp. 319-350, 2000.
- [12] Gmytrasiewicz, P.J., Noh, S., and Kellogg, T., Bayesian update of recursive agent models, User Modeling and User-Adapted Interaction: An International Journal, vol. 8, no. 1/2, pp. 49-69, 1998.



노 상 옥

1987년 서강대학교 이과대학 학사. 1989년 서강대학교 공과대학 전산학 석사. 1999년 University of Texas at Arlington 전산학 박사. 1989년~1995년 국방과학연구소 연구원. 2000년~2002년 University of Missouri-Rolla 조교수. 2002년~현재 가톨릭대학교 컴퓨터정보공학부 조교수. 관심분야는 지능형 에이전트, 멀티 에이전트 시스템, 지식관리, 기계학습, 실시간 분산 시스템 등