시간 제약적인 환경에서 컴파일 기법을 사용한 실질적이며 유연한 의사결정 방법

(Practical and Flexible Decision-Making Using Compilation in Time-Critical Environments)

노 상 욱 †

(Sanguk Noh)

요 약 여러 에이전트가 존재하는(multiagent) 환경에서 자율적인 에이전트들은 복잡하고 불확실한 환경뿐만 아니라 다른 에이전트들도 고려하여 자신의 결정을 수행하여야 하기 때문에 제한된 시간 내에 의사결정(decision-making)을 완료한다는 것은 실질적으로 불가능하다. 이러한 문제점을 극복하고 긴급한 상황에서 최적의 행동을 수행하기 위하여 자율적인 에이전트들에게 다양하고 유연한 의사결정 방법들을 제공한다. 이 방법들은 의사결정의 질적인 수준과 의사결정 소요시간을 고려하여 실질적인 에이전트의 의사결정을 가능하도록 한다. 유연한 의사결정 방법의 하나로 컴파일된 규칙의 사용을 제안하며, 자율적인 에이전트는 복잡한 실시간 환경에서 가능한 행동의 범위를 제한하기 위하여 조건-행동 규칙을 사용한다. 지대공 방어 환경에서 주어진 상황의 긴박한 정도에 따라 이에 적절한 행동을 자율적으로 수행하는 유연한 에이전트를 실험적으로 보인다.

키워드 : 에이전트 모델링, 의사결정, 적응력을 갖춘 에이전트, 시간 제약적인 환경, 지대공 방어 시스템

Abstract To perform rational decision-making, autonomous agents need considerable computational resources. When other agents are present in the environment, these demands are even more severe. In these settings, it may be difficult for the agent to decide what to do in an acceptable time in multiagent situations that involve many agents. These problems motivate us to investigate ways in which the agents can be equipped with flexible decision-making procedures that enable them to function in a variety of situations in which decision-making time is important. The flexible decision-making methods explicitly consider a tradeoff between decision quality and computation time. Our framework limits resources used for agent deliberation and produces results that are not necessarily optimal, but provide autonomous agents with the best decision under time pressure. We validate our framework with experiments in a simulated anti-air defense domain. The experiments show that compiled rules reduce computation time while offering good performance.

Key words : agent modeling, decision-making, adaptive agents, time-critical domains, anti-air defense

1. 서 론

복잡하고 불확실한 환경에서 자율적인 에이전트가 어떤 행동을 수행해야 하는가에 대한 이성적인 의사결정(rational decisions)은 많은 비용을 동반한다[1-6]. 예를 들어, 여러 행동들에 대한 예상 유틸리티(expected utility)를 계산하고 이 중에서 최대 유틸리티 값을 가지는 행동을 따르는 지능형 에이전트(intelligent agent)를 구현하는 것은 매우 복잡한 과정이며 최적의 행동을 얻기 위하여 많은 시간을 소모하여야 한다. 또한, 여러 에이전트들이 존재하는 환경에서 자율적인 에이전트들은 복잡하고 불확실한 환경뿐만 아니라 다른 에이전트들도 고려하여 자신의 결정을 수행해야 하기 때문에 제한된 시간 내에 의사결정을 완료한다는 것은 실질적으로 불가능하다[7-11]. 이러한 문제점을 극복하고 긴급한 상황에서 최선의 행동을 수행하기 위하여 자율적인 에이전트들에게 다양하고 유연한 의사결정 방법들을 제공한다. 이 방법들은 의사결정의 질적인 수준과 의사결정 소요

· 이 논문은 2002년도 한국학술진흥재단(KRF-2002-041-D00465), 2003년도 가톨릭대학교 교비연구비의 지원에 의하여 연구되었음

† 종신회원 : 가톨릭대학교 컴퓨터정보공학부 교수
sunoh@catholic.ac.kr

논문접수 : 2003년 6월 24일
심사완료 : 2003년 9월 9일

시간을 고려하여 실질적인 에이전트의 의사결정을 가능하도록 한다.

본 논문에서 제안하는 유연한 의사결정 방법론(flexible decision-making procedures)은 단순화(simplification)와 근사화(approximation) 기법으로 구성된다.

- 단순화 기법은 이용 가능한 정보의 일부를 무시하거나 이러한 정보를 처리하는 과정을 줄이는 것이다. 이 때, 최적의 의사결정은 얻을 수 있으나 이러한 기법이 특정한 상황에서 적용 가능한 것인지에 대한 보장은 없다.
- 근사화 기법은 의사결정을 위해서 고려하는 정보의 범위를 제한하는 방식이다. 이 방식을 이용하여 얻은 행동은 최적(optimal solution)이 아닐 수도 있다.

여러 에이전트들이 존재하는 환경에서 이성적인 의사결정을 위하여 재귀모델링방법(Recursive Modeling Method)[12,13]을 사용한다. 재귀모델링방법에 의해 구현된 에이전트들은 다른 에이전트들의 선택을 예상하면서 자신을 위한 최적의 행동을 결정한다. 다른 에이전트들의 예상된 행동은 각 에이전트들이 가지고 있는 다른 에이전트들에 대한 모델에 기초하여 얻어진다. 또한, 다른 에이전트들도 각 에이전트들을 모델링 한다는 사실로부터 재귀적 모델의 계층구조를 형성하며, 이를 재귀적모델구조(Recursive Model Structure)라고 부른다. 각 에이전트들이 다른 에이전트들에 대하여 유한한 정보를 가지고 있다고 가정하면, 유한한 깊이의 재귀적모델구조를 형성한다. 이러한 재귀적모델구조는 다이나믹 프로그래밍(dynamic programming) 기법을 이용하여 풀 수 있으나[14-17], 에이전트의 개수가 증가하면 최악의 경우에 모델구조의 크기는 기하급수적으로 증가한다.

재귀모델링방법은 의사결정에 대한 지식을 재귀적모델구조를 이용하여 나타내며, 이 방법론 내에서 단순화 기법은 최적화된 행동을 얻기 위하여 재귀적모델구조를 풀어가는 계산과정의 일부분을 제거한다. 예를 들어, 에이전트가 다른 에이전트의 어떠한 행동에도 불구하고 모든 행동보다 바람직한 행동 - 우월행동(dominant action)으로 불린다 - 을 결정할 수 있다면, 다른 에이전트가 무엇을 할 것인지에 대하여 계산할 필요가 없으며 재귀적모델구조에서 해당 부분은 안전하게 제거될 수 있을 것이다. 단순화 기법에서 제공하는 모든 방법은 심각한 과부하 없이 의사결정에 필요한 계산 시간을 줄일 수 있다.

반면에 근사화 기법은 자율적인 에이전트에게 주어진 상황의 실시간 제약 하에서 제한된 이성적인 행동(bounded rational behavior)을 제공하는 것이다. 본 논문은 근사화 기법의 하나로 에이전트가 고려하는 행동들의 개수를 제한하는 컴파일된 규칙(compiled rules)을 사용한다. 이 규칙들은 결정이론(decision theory)에 의해 결정된 추론결과를 조건-행동 규칙으로 컴파일(compile) 함으로써 생성된다. 컴파일 과정은 멀티에이전트 환경에서의 일정한 규칙성을 찾아내며, 이러한 규칙에 해당하는 행동들 만을 고려 함으로서 긴급한 상황에서 의사결정을 위하여 필요한 시간을 줄일 수 있게 된다. 주어진 의사결정 상황에서 정형화된 규칙을 얻기 위하여 기계학습(machine learning) 알고리즘을 사용하며, 입력으로는 재귀모델링방법에 의해 얻어진 최적의 행동을 사용한다. 다양한 컴파일 기법과 결정이론적인 추론방식을 조합하여 시간 제약성을 충족시킬 수 있는 다양한 의사결정 방법을 제공한다. 예를 들어, 반응이 빠른(reactive) 조건-행동 규칙들은 정확하지만 의사결정 시간이 오래 걸리는 결정이론적인 방식에 전처리 단계로 작동하게 된다. 제안하는 방법론에서 컴파일된 규칙은 최적의 행동이 아닐 수 있는 행동들을 일차로 제거하고, 결정이론적인 방식에 다른 행동보다 바람직한 행동만을 전달하게 된다.

본 논문은 2장에서 제안하는 방법과 관련한 연구들을 비교하며, 3장에서 컴파일 과정과 컴파일된 규칙을 어떻게 사용하는가를 설명한다. 4장에서 멀티에이전트 환경에 대한 예제로 지대공 방어 환경을 소개한다. 5장에서는 시간제약적인 지대공 방어환경에서 제안하는 방법론을 수행하는 에이전트의 성취정도를 측정하며, 결론에서 실험결과와 앞으로의 연구방향에 대하여 논의한다.

2. 관련연구

복잡한 실시간 환경에서 최적의 의사결정을 생성하기에 충분한 시간이 없을 경우에 자율적인 시스템을 작동할 수 있도록 하는 연구는 다양하게 이루어져 왔다. 실시간 문제 해결기법에서의 전통적인 접근방식은 미리 정의된 규약(protocol)에 의존한 조건-행동 추론방식이다[18,19]. 이러한 조건-행동 추론방식은 주어진 조건에 빠르게 반응하지만, 문제 영역이 다양하거나 문제의 재설계가 필요할 경우에 이러한 방식을 따르는 에이전트들은 아무런 해법을 가지지 못하는 단점이 있다. 특히, 시스템의 조건-행동에 대한 규약 자체를 실시간 이전에 정의하지 못한다면, 조건-행동 추론방식은 유용하지 않으며 이러한 방식을 사용하는 에이전트들은 어떠한 의사결정도 수행하지 못할 것이다. 반면에, 본 논문을 통하여 제안하는 접근방식은 미리 정해진 조건-행동 추론규칙에 의존하는 것이 아니라, 가능한 행동들에 대한 정보를 바탕으로 최선의 행동을 실시간에 계산하여 선택한다. 따라서, 미리 예측이 불가능한 동적인 환경에서도 제안하는 에이전트는 최선의 행동을 결정할 수 있다.

다른 접근 방식으로는 Bratman[20] 등이 제안한 수

단-목적 추론(means-end reasoning)과 결정이론적인 추론방식을 포함하는 에이전트 구조이다. 자원제한적인 에이전트를 위하여 에이전트의 지식(Beliefs), 목표(Desires), 의도(Intentions) - BDI - 가 의사결정과정을 규정하는 플랜 (plan)의 형성과 수행과정에 관여함을 보였다. 또한, Rao[21,22] 등은 BDI 구조에 의해 제공되는 응용 프로그램을 개발하였고, 항공관제시스템(OASIS: Agent-Oriented Air-Traffic Management System at Sydney Airport)에 적용하였다. 이러한 방법론은 추론방식을 결합하여 사용하는 이론적인 근거를 제공하였지만, 동적인 환경에서 어떻게 규칙들을 생성하여야 하는가에 대한 방법을 고려하지 않았다. 이러한 접근방식에서 규칙의 생성은 전적으로 시스템 설계자에 의존할 것이다.

멀티에이전트 분야에서 실질적인 의사결정을 위한 다양한 연구가 이루어져 왔다. Lesser[23]는 여러 에이전트가 존재하는 복잡한 환경에서 도메인에 독립적인 에이전트 구조를 제안하였다. 이러한 구조는 자원의 가용성에 적응하는 방법, 의사결정 수준과 자원의 양에 대한 적절한 균형, 에이전트들 사이의 복잡한 상호관계를 어떻게 해결하는가에 대한 이론적인 배경을 제공한다. 앞으로의 연구를 통하여 Lesser 등은 제안한 소프트웨어 구조의 성분들을 구현하여야 할 것이며, 실질적인 멀티에이전트 응용 환경에서 평가되어야 할 것이다. Durfee[24]는 에이전트들이 재귀모델링방법을 의사결정방법으로 사용할 경우에 실질적인 방법론을 제안하였다. 이 방법론은 유틸리티 계산을 간단히 하는 것, 가능한 선택들을 제한하는 것, 여러 개의 에이전트들을 하나로 취급하는 방법 등을 포함한다. Durfee의 방법론은 본 논문에서 제안하는 방법론과 자율적인 에이전트가 재귀모델링 방법을 사용한다는 점에서 공통점이 있으나, 에이전트의 의사결정을 제한하는 방식이 다르며, 특히 의사결정 시간과 의사결정 수준에 대한 고려는 본 논문에서 제안하는 접근방식과 Durfee의 방식을 비교할 때 확실한 차이점을 나타낸다.

3. 컴파일 방법을 사용한 의사결정 규칙의 생성

이 장에서는 실시간 멀티에이전트 환경에서 자율적인 에이전트에 의해 사용되는 유연한 의사결정 방법론의 근사화 기법중의 하나인 컴파일 방법에 대한 이론적인 배경을 설명한다. 이 장에서는 다음과 같은 표기법을 사용한다.

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

특정한 시간 t에 주어진 환경과 다른 에이전트에 대한 지식(또는 정보) E를 사용하여 에이전트 n_i가 획득한 최적의 행동 α에 대한 예상 유틸리티(expected utility: EU)는 다음과 같이 구한다.

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

위의 식 (1)에서,

- $P(s_k|E,t,a_i^j)$는 주어진 지식 E를 가지고 획득된 행동 a_i^j가 시간 t에 수행되어진 후 특정한 상태 s_k에 도달할 확률이며,
- $U(s_k)$는 상태 s_k의 유틸리티 값을 나타낸다.

유연한 의사결정에 대한 접근 방법은 컴파일 방식과 결정이론 방식의 결합으로 나타나는 에이전트의 제한된 이성(bounded rationality)을 추구하며, 결정이론 방식은 식 (1)에서 나타난 바와 같이 재귀모델구조에 포함된 정보에 의존하여 계산된 예상 유틸리티 값을 최대화하는 원칙을 따른다.

컴파일 과정을 정형화하기 위하여 조건(condition) 함수를 정의한다. 조건 함수는 상태 집합 S로부터 에이전트에 의하여 행해지는 관찰[1] 들의 집합 O로의 함수이다. 예를 들어, 각각의 $s_k \in S$에 대하여 $condition(s_k) = o_k$로 표시되며, o_k는 집합 O의 관찰할 수 있는 인스턴스이다. 또한, 컴파일을 위하여 에이전트가 사용하는 기계학습 알고리즘의 집합을 $L=\{l_1, l_2, \ldots\}$이라고 하자.

주어진 학습방법 $l_m \in L$에 대하여 컴파일된 의사결정 과정은 다음과 같은 규칙들의 집합을 생성한다.

$$\omega_l : O \times A_{n_i} \mapsto \{True, False\} \quad (2)$$

식 (2)는 하나의 행동이 임의의 상태에 대한 조건이 만족할 경우에 추천되든지(True) 혹은 추천되지 않음(False)을 결정한다. 예를 들어, $\omega_{lm}(condition(s_k), a_i^j) = True$는 학습 알고리즘 l_m에 의해 얻어진 규칙들의 집합이 조건 $condition(s_k)$에서 행동 a_i^j를 추천함을 의미한다. 이와 같이, 다양한 학습 알고리즘을 이용하여 결정이론을 따르는 행동들의 모델을 컴파일할 수 있으며, 이러한 결과는 여러 가지 ω_{lm}의 함수로 나타난다.

조건-행동 규칙에 의해 반응이 빠른 방식과 결정이론적인 방식은 주어진 환경에 따라 다양한 형태로 결합될 수 있다. 에이전트가 주어진 환경에서 충분한 시간을 가지고 의사결정을 할 수 있으면, 각 에이전트는 예상 유

1) 이러한 관찰은 에이전트가 보이는 관심과 관련한 파라미터 만을 포함한 즉각적인 상태의 요약으로 해석할 수 있다. 예를 들어, 블록 도메인(blocks-world domain) 에서 현재 상태의 요약은 에이전트가 블록을 잡으려고 할 경우, 에이전트의 위치와 블록의 위치, 그리고 블록과 관련한 파라미터들을 말한다.

틸리티를 최대화하는 결정이론적인 방식을 이용하여 최적의 의사결정을 수행하려고 할 것이다. 반면에, 시간제약적인 환경에서 에이전트의 의사결정은 이용 가능한 계산시간에 의하여 제한 받는다. 결정이론적인 방식을 적용하기 전에, 자율적이며 환경에 적응하는 에이전트는 모든 가능한 행동으로부터 바람직한 행동을 분리하기 위하여, 그리고 의사결정 시간을 줄이기 위하여 컴파일된 규칙의 집합을 사용하게 된다. 따라서, 주어진 상황의 특정한 관찰인 o_k와 기계학습 알고리즘 l_m을 통하여 컴파일된 규칙을 에이전트 n_i가 사용하는 경우, 모든 가능한 행동으로부터 바람직한 행동들의 집합은 다음과 같이 결정된다.

$$A'_{n_i} = \{a_i^j \mid a_i^j \in A_{n_i} \text{ and } \omega_{l_m}(o_k, a_i^j) = True\} \quad (3)$$

이때, $A'_{ni} \subseteq A_{ni}$이다. 직관적으로 A'_{ni}는 에이전트 n_i가 주어진 상황에서 고려해야만 하는 바람직한 행동들의 집합이다. 시간제약적인 환경에서 자율적인 에이전트는 컴파일된 규칙을 현재 상태에 적용하여 최선의 행동을 얻기 위한 후보 행동들의 집합을 생성한다. 구체적으로, 주어진 상황 $s_k \in S$에서 행동들의 집합이 $A_{ni} = \{a_i^1, a_i^2, a_i^3\}$일 때, $\omega_{lm}(o_k, a_i^1) = True$, $\omega_{lm}(o_k, a_i^2) = False$, $\omega_{lm}(o_k, a_i^3) = True$라고 가정하면, a_i^1, a_i^3이 바람직한 행동이며, a_i^2는 최선의 행동을 얻기 위한 후보 행동들의 집합에 포함되지 않는다. 이와 같이 미리 여과된 행동들 중에서 예상 유틸리티 값을 최대화하는 행동을 결정하며 결과적으로 최선의 행동은 다음과 같이 얻어진다.

$$\arg\max_{a_i^j \in A'_{n_i}} EU(a_i^j \mid E, t) \quad (4)$$

의사결정 방식을 결합함으로써 에이전트는 추론시간을 절약할 수 있으며, 에이전트의 실질적인 의사결정을 가능하게 한다. 에이전트의 이성적인 행동들에 대한 컴파일 과정은 자율적인 에이전트로 하여금 주어진 환경에서의 정형성을 이용할 수 있도록 하며, 복잡한 환경에서 더욱 유연하고 적응성이 높도록 하며, 의사결정 과정의 고유한 복잡도를 줄인다.

4. 멀티에이전트 도메인: 지대공 방어 시스템

지대공 방어 시스템 모델은 그림 1에 나타난 것과 같이 일련의 공격해오는 목표물들과 상호 협력하여 목표물을 제거하는 방어 에이전트들로 구성되어 있다. 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

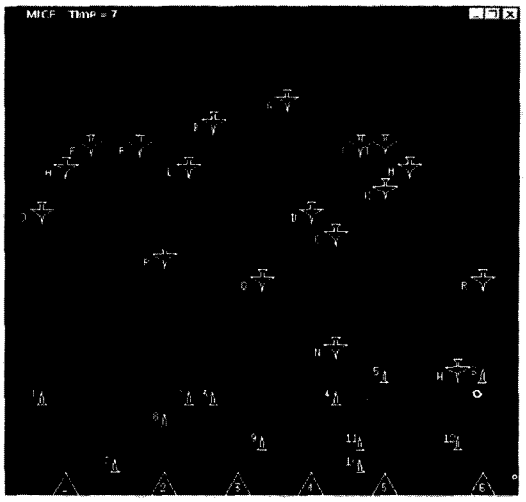

그림 1 복잡한 지대공 방어 시스템 시나리오

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 확률[2] $P(H_{ij})$은 목표물 j의 진행방향과 방어 에이전트 n_i가 목표물을 바라보는 직선방향 사이의 각 γ_{ij}에 의존한다[25]. 요격 확률은 다음과 같다.

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

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

5. 실험 및 평가

이 장에서의 실험은 어떻게 에이전트 사이의 상호협력과 유연한 의사결정이 이루어 질 수 있는가를 구체적인 지대공 방어 환경에 적용함으로써 평가하고자 한다. 복잡한 지대공 방어환경에서 구현한 에이전트의 성능을 분석하며, 학습을 통하여 컴파일된 규칙이 최적이 아닌 행동들을 제거할 경우의 에이전트 의사결정 시간과 성취도를 측정한다.

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

지대공 방어 시스템은 MICE 시뮬레이터[26]를 이용하여 Common Lisp으로 작성되었으며 LINUX 상에서 수행된다. 시뮬레이션 환경에서 각각의 시나리오를 위하여 목표물의 위치와 탄두의 크기는 임의로 생성되었으며, 위치 좌표는 0에서 10까지, 그리고 탄두 크기는 100에서 300까지의 값을 가진다.

실험을 위하여 다음과 같은 환경을 이용하였다.

- 2 vs. 2: 두 개의 방어 에이전트와 두개의 공격해 오는 목표물이 존재한다. 이러한 환경은 컴파일 과정을 위하여 오프라인에 사용된다.
- 6 vs. 18: 6개의 방어 에이전트와 18개의 공격해 오는 목표물이 있으며, 이 환경은 에이전트의 성능을 테스트하기 위한 복잡한 시나리오로 사용된다.

실험에서 에이전트의 성취도(performance)는 방어지역에서 제거된 전체 위협(total threat removed)으로 나타낸다. 제거된 위협(또는 피해 (Damage))은 공격해 오는 목표물의 탄두에 대한 예상 제거 합으로 정의한다. 이와 같이 특정한 목표물이 요격을 위하여 선택되면, 요격확률과 탄두의 크기를 곱한 만큼 제거된 전체 위협에 포함된다. 반면에 목표물이 선택되지 않으면, 제거된 전체 위협에 포함되지 않는다.

지대공 방어 환경에서 컴파일된 규칙을 생성하기 위하여 naive Bayesian classifier[27], C4.5[28], CN2[27], FOIL[29] 등의 4가지 학습 알고리즘을 사용하였다. 학습 알고리즘을 위한 입력 데이타로 재귀모델링방법에 의하여 얻은 의사결정 결과를 사용하였다. naive Bayesian classifier는 학습 결과를 주어진 클래스에서 각각의 속성값이 발생하는 확률로 나타낸다. 실험에서 클래스는 목표물을 선택하는 것(Select Target)과 선택하지 않는 것(Don't Select Target)으로 이루어 진다. C4.5는 결과를 의사결정 나무(decision tree)로 나타내며, CN2의 결과는 순서화된 조건-행동 규칙의 집합이다. FOIL의 학습결과는 Horn clause의 형태로 속성간의 관계를 나타낸다. 이와 같은 학습 알고리즘을 이용하여 얻은 지대공 방어 에이전트의 컴파일된 규칙과 각 에이전트의 의사결정 능력을 비교한다.

5.1 학습된 조건-행동 규칙

실험에서 각 에이전트의 방어 수행능력이 다르며, 다른 에이전트에 대한 제한된 지식을 가지고 있다고 가정하였다. 즉 방어 수행능력은 표 1에서 나타난 바와 같이 에이전트들이 지대공 방어 환경에서 감지하는 상황에 대한 속성에 의존한다. 이러한 속성은 탄두의 크기, 목표물의 속도, 목표물과 방어 에이전트 사이의 각도, 목표물과 방어 에이전트 사이의 거리, 에이전트 자신의 요격 능력 - 장거리와 단거리 미사일을 보유한 경우, 장거리 미사일만을 보유한 경우, 단거리 미사일만을 보유한

표 1 지대공 방어 환경에서의 속성

속성	자료 유형	값의 범위
목표물 탄두의 크기 (size)	Numeric	100 - 300
목표물 속도 (speed)	Nominal	slow, mid, fast
목표물과 방어 에이전트의 각도 (angle)	Numeric	0 - 90
목표물과 방어 에이전트의 거리 (distance)	Numeric	0 - 20
요격 능력 (capacity)	Nominal	Both, long, short, incap.
모든 공격수단 보유 확률	Numeric	0 - 1
장거리 공격무기 보유 확률	Numeric	0 - 1
단거리 공격무기 보유 확률	Numeric	0 - 1
작전수행 능력이 없을 확률	Numeric	0 - 1

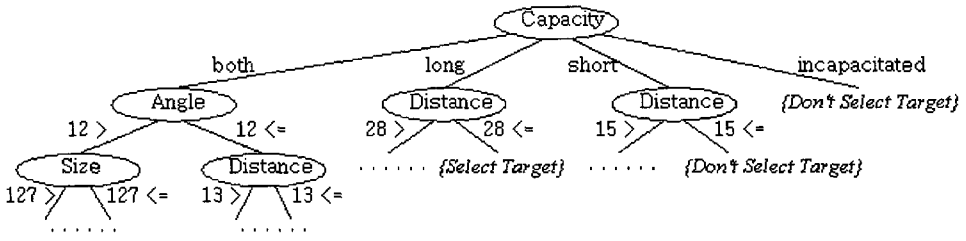

그림 2 C4.5를 통하여 획득한 의사결정 나무

경우, 그리고 작전수행 능력이 없는 경우 - 및 다른 에이전트들의 요격 능력[3]에 대한 모델과 연관된 확률 등을 포함한다. 실험에서는 속성 값들이 표 1의 범위 내에서 임의로 생성되었다.

표 1에 있는 속성들을 기초로 하여 방어 에이전트들이 고려하는 목표물은 두 개의 클래스로 분류된다: {Select Target, Don't Select Target}. 예를 들어, C4.5를 통하여 획득한 의사결정 나무는 그림 2와 같다.

5.2 컴파일된 규칙을 이용한 성취정도

학습에 의해 제안된 가설이 적절하게 생성되었는가에 대한 검증을 위하여 몇 개의 학습예제(training examples) 집단을 준비하고, 학습예제의 수가 증가함에 따라 결과적으로 얻어진 성취도에 대한 학습 곡선(learning curve)을 그려본다. 2개의 방어 에이전트와 2개의 목표물이 있는 지대공 방어 환경에서 충분한 학습예제의 수는 250개로 확인 되었다. 결과적으로 컴파일된 규칙을 이용하여 새로운 250개의 시나리오에 적용하여 시험한 에이전트의 평균 성취도(제거된 전체 위협)와 의사결정을 위하여 사용한 시간(초)에 대한 값은 표 2에 나타난다.

표 2에 나타난 성취도를 표준변이분석법(standard analysis of variance: ANOVA)을 사용하여 비교하였다. ANOVA 방법에 의하여 구해진 값인 f=8.94는 표준측정값인 3.32(= $f_{.01,\ 4,\ \infty}$) 보다 크므로 5개의 학습 알고리즘을 이용하여 컴파일된 규칙을 사용하는 에이전트 팀의 성능은 99%의 신뢰도로 각각 다르다고 분석할 수 있다. 예상한 바와 같이 결정이론 방식을 따르는 재귀모델링방법을 이용하여 의사결정을 수행하는 에이전트의 성취도가 가장 좋았다. naive Bayesian classifier는 주어진 두개의 클래스를 위한 확률을 계산하기 때문에 방어 에이전트로 하여금 계산된 확률 중에서 가장 큰 값을 선택하도록 한다. 결과적으로, Bayesian classifier를 이용한 에이전트의 성취도는 비교적 높은 반면 계산시간은 빠른 편이었다. 순전히 반응이 빠른 의사결정 방법은 주어진 환경에서 목표물에 대한 자료가 유사할 경우에 유일하게 목표물을 선택하지 못하게 되므로, 여러 개의 목표물이 존재할 경우에 임의로 선택하게 하였다. C4.5를 사용한 에이전트의 성취도는 FOIL 또는 CN2 방식에 비하여 더 좋은 성능을 보였으며 동시에 적은 계산시간을 사용하였다.

재귀모델링방법이 가장 긴 수행시간을 필요로 하였으며 반면에 C4.5는 가장 짧은 수행시간이 소요되었다. 표준변이분석법을 통하여 구해진 값인 f=15.592는 표준측정값인 3.32(= $f_{.01,\ 4,\ \infty}$) 보다 크므로 다섯 개의 알고리즘들이 필요로 한 수행시간의 차이가 99%의 신뢰도로 유효하였음을 알 수 있었다. 방어 에이전트들이 의사결정을 수행할 경우에 입력 값을 조건-행동 규칙들 중의 하나와 비교하고 조건을 만족하면 두 개의 클래스 중의 하나로 분류하게 된다. FOIL과 CN2로부터 획득된 학습결과는 순차적 비교로 나타나며, C4.5는 의사결정 나무구조로 나타난다. 이러한 차이 때문에 C4.5 보다 FOIL과 CN2의 수행시간이 더 오래 걸린 것으로 판단된다. 의사결정 나무구조의 장점은 입력과 관련이 없는 노드를 고려하지 않음으로써 전체 비교회수를 줄일 수 있다는 것이다.

표 2 두개의 방어 에이전트와 두개의 목표물이 있는 지대공 방어 환경에서의 성취도와 의사결정 소요시간

의사결정 방법	성취도	의사결정 소요시간
결정이론 방식		
재귀모델링방법(RMM)	200.6 ± 97.9	0.0062 ± 0.0048
반응이 빠른 방식		
Naïve Bayes	170.5 ± 105.9	0.0027 ± 0.0044
C4.5	158.4 ± 102.2	0.0019 ± 0.0039
FOIL	155.9 ± 104.8	0.0024 ± 0.0042
CN2	154.3 ± 100.5	0.0025 ± 0.0043
ANOVA	8.935	15.592

적응성(adaptive)을 갖춘 결정이론을 따르는 에이전트의 효율성을 측정하기 위하여, 그림 1과 같은 복잡한 지대공 방어 환경을 가지고 실험하였다. 이러한 방어 환경에는 6개의 방어 에이전트와 18개의 공격해 오는 목표물이 존재한다. 구현한 에이전트는 의사결정 소요시간을 줄이기 위하여 각 에이전트와 가장 근접한 에이전트의 행동을 모델링한다. 표 2에 나타난 바와 같이 naive Bayesian classifier와 C4.5가 반응이 빠른 방식에서 가장 좋은 성취도를 보였기 때문에 두 가지 알고리즘을 에이전트의 바람직한 행동들의 집합 A'_{ni}을 얻기 위하여 사용한다(식 (3) 참조). 학습을 훈련하기 위한 자료로 13,800 튜플(tuples)을 생성하였으며 이중 10,200개의 튜플은 {Don't Select Target} 클래스로 3,600개의 튜플은 {Select Target} 클래스로 구성하였다. 결과적으로 복잡한 지대공 방어환경에서 얻은 평균 성취도(제거된 전체 위협)와 계산시간(초)은 표 3에 정리하였다.

표 3은 200개의 시나리오에서 수행한 에이전트의 평균 성취도와 수행시간을 나타낸다. 본 실험에서 세가지 유형의 에이전트 성취도에 대한 결과를 정리하였다. 결정이론방식만을 이용하여 의사결정을 수행하는 유형, 반응이 빠른 방식을 이용하여 바람직하지 않은 선택들을 미리 제거한 후 결정이론 방식을 적용하는 유형, 및 반응이 빠른 방식만을 이용하여 의사결정을 수행하는 유형의 세가지이다. 오직 결정이론방식을 이용하는 재귀모

3) 표 1에서는 2개의 방어 에이전트에 대한 파라미터를 정리하였다. 에이전트 집합의 일반화를 위하여는 부가적인 확률을 필요로 한다.

표 3 복잡한 지대공 방어 환경에서의 성취도와 의사결정 소요시간

의사결정 방법	성취도	의사결정 소요시간
재귀모델링방법 (RMM)	2458.5 ± 261.8	18.455 ± 1.708
Bayes-RMM	2389.1 ± 259.7	2.075 ± 2.229
C4.5-RMM	2361.1 ± 233.2	1.004 ± 0.653
Bayes	2164.9 ± 267.4	0.791 ± 0.043
C4.5	1885.5 ± 236.3	0.127 ± 0.009

델링방법의 성취도가 가장 좋았다. 두 가지 의사결정 방식을 혼합한 Bayes-RMM과 C4.5-RMM은 RMM 에이전트의 성취도에 각각 97.2%와 96.0%를 달성하였다. 적응성이 있으며 결정이론 방식과 혼합한 유형의 에이전트의 의사결정 소요시간은 급격히 감소하였다. 실험에서 반응이 빠른 방식에 의하여 제거된 후, 바람직한 행동들의 후보로 남은 행동의 개수는 평균 7(Bayes) - 9(C4.5) 개로 측정되었다. 또한, Bayes-RMM의 평균 의사결정 시간은 2.075였으며, C4.5-RMM 에이전트의 평균 의사결정 시간은 1.004초로 나타났다. 이러한 결과는 반응이 빠른 방식과 결정이론적인 방식이 혼합한 형태의 에이전트는 의사결정 시간이 짧은 반면 상대적으로 높은 수준의 의사결정을 가능하게 하였다. 또한, naive Bayesian classifier는 각각의 클래스를 위한 부가적인 확률을 포함하고 있기 때문에 C4.5 보다 더 좋은 성취도를 나타낼 수 있었다. C4.5만을 사용한 에이전트의 의사결정은 바람직하지 않은 행동들을 제거한 후에 임의로 목표물을 선택하였기 때문에 5가지 유형에서 가장 낮은 성취도를 나타내었다.

6. 결 론

본 논문에서는 여러개의 에이전트가 존재하는 환경에서 각각의 에이전트들이 실질적인 시간 내에 최선의 의사결정을 수행할 수 있는 유연한 의사결정 방법론을 제안하였으며, 시간 제약적인 멀티 에이전트 환경에서 제안한 방법론이 어떻게 사용될 수 있는가에 대한 실험을 수행하였다. 유연한 의사결정 방법론이 긴급한 상황에서 자율적인 에이전트의 실질적인 의사결정을 가능하게 하는 유용한 방법임을 입증하였다. 수행한 연구결과를 정리하고 앞으로의 연구방향을 논의하고자 한다.

6.1 연구결과 및 토의

시간 제약적인 환경에서 일관성 있는 조건-행동 규칙을 축적(컴파일)하기 위하여 귀납학습방법을 이용하였다. 컴파일된 규칙은 에이전트의 규범으로 정형화할 수 있었으며, 학습 결과는 하나의 계층구조를 형성하며 이러한 계층구조는 에이전트의 경험으로 축적되었다. 결과적으로 조건-행동 규칙에 의해 추천된 행동의 집합이 재귀모델링알고리즘의 입력이 되며 바람직한 행동들 중에서 에이전트가 최선의 행동을 선택할 수 있었다. 자율적인 에이전트가 컴파일된 지식의 조건-행동 규칙과 결정이론적인(decision-theoretic) 알고리즘을 혼합하여 사용 함으로써, 실시간 추론 시간을 줄이며 실질적인 의사결정이 가능하도록 하였다.

유연한 의사결정 방법론을 지대공 방어환경에서 평가하였다. 지대공 방어환경은 실시간성을 요구하는 분야이며, 시간에 적절히 반응하지 않을 경우에는 상호 협력하는 방어 에이전트들에게 심각한 피해를 가져온다. 방어 에이전트들이 컴파일된 규칙을 사용함으로써 이러한 재난을 극복할 수 있었으며, 긴급한 상황의 지대공 방어환경에서 제한된 의사결정을 가능하게 하였다. 제안한 방법론을 가지는 에이전트 구조의 가장 중요한 특징은 오프라인(offline)과 온라인(online) 모듈들을 선택적으로 사용하여 다양한 상황에서 적절한 의사결정을 수행할 수 있다는 것이다.

6.2 앞으로의 연구방향

본 논문에서 제안한 방법론을 다양한 시간 제약적인 환경에 적용하도록 할 것이다. 또한, 주어진 상황이 얼마나 긴급한지에 대한 예측이 가능할 수 있도록 긴급한 정도를 계산하는 것에 대한 이론을 연구할 것이다. 결과적으로 복잡한 분산 실시간 환경에서 시간의 긴급한 정도에 따라 적절한 수준의 의사결정을 수행하는 - 실질적인 의사결정이 가능한 - 자율적인 에이전트를 구현할 것이다.

앞으로의 지속적인 연구를 통하여 전시작전 분야 및 우주탐험 분야등과 같은 인간이 대처하기에 매우 위험한 상황에서 예측하지 못한 임무를 수행하는 유연한 의사결정 방법들을 연구할 것이며, 개발한 에이전트를 여러 가지 실시간 분산 시스템에 적용시키고자 한다.

참 고 문 헌

[1] Brooks, R.A., A robust layered control system for a mobile robot, IEEE Journal on Robotics and Automation, vol. RA-2, no. 1, pp. 14-23, Mar. 1986.

[2] Dean, T., Decision-theoretic control of inference for time-critical applications, Artificial Intelligence, pp. 1-28, Nov. 1990.

[3] Good, I.J., Twenty-seven principles of rationality, Foundations of Statistical Inference, V.P. Godambe and D.A. Sprott, Eds., pp. 108-141. Holt, Rinehart, and Winston, 1971.

[4] Horvitz, E.J., Reasoning about beliefs and actions under computational resource constraints, In Proceedings of the 1987 Workshop on Uncertainty in Artificial Intelligence, 1987.

[5] Horvitz, E.J., Cooper, G.F., and Heckerman, D.E., Reflection and action under scarce resources: Theoretical principles and theoretical study, In Proceedings of the 11th International Joint Conference on Artificial Intelligence, Detroit, Michigan, Aug. 1989, pp. 1121-1127.
[6] Kortenkamp, D., Schreckenghost, D., and Bonasso, R.P., Real-time autonomous control of space habitats, AAAI Spring Symposium, March 2000, pp. 38-45.
[7] Russell, S. J. and Wefald, E. H., Principles of Metareasoning, Journal of AI, Volume 49, pp. 361-395, 1991.
[8] Russell, S.J. and Subramanian, D., Provably bounded-optimal agents, Journal of Artificial Intelligence Research, Volume 2, pp. 575-609, 1995.
[9] Russell, S.J., Rationality and Intelligence, Artificial Intelligence, Volume 94, pp. 57-77, 1997.
[10] Simon, H.A., The Sciences of the Artificial, MIT Press, 1969.
[11] Zilberstein, S. and Russell, S.J., Optimal composition of real-time systems, Artificial Intelligence, vol. 82, no. 1, pp. 181-213, 1996.
[12] 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.
[13] 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.
[14] 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.
[15] Noh, S. and Gmytrasiewicz, P.J., Towards Flexible Multi-Agent Decision-Making Under Time Pressure, In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, pp. 492-498, Stockholm, Sweden, August 1999.
[16] Noh, S. and Gmytrasiewicz, P.J., Rational communicative behavior in anti-air defense, In Proceedings of the Third International Conference on Multi-Agent Systems, pp. 214-221, July 1998.
[17] Noh, S. and Gmytrasiewicz, P.J., Coordination and Belief Update in a Distributed Anti-Air Environment, In Proceedings of the 31st Hawaii International Conference on System Sciences, Vol. V, pp. 142-151, Hawaii, January 1998.
[18] 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.
[19] 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.
[20] Bratman, M.E., Israel, D.J. and Pollack, M.E., Plans and resource-bounded practical reasoning, Journal of Computational Intelligence, vol. 4, pp. 349-355, 1988.
[21] Rao, A.S. and Georgeff, M.P., An abstract architecture for rational agents, In Proceedings of the Knowledge Representation and Reasoning, pp. 439-449, 1992.
[22] 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.
[23] Lesser, V.R., Reflections on the nature of multi-agent coordination and its implications for an agent architecture, Autonomous Agents and Multi-Agent Systems, vol. 1, no. 1, pp. 89-111, 1998.
[24] Durfee, E.H., Practically coordinating, AI Magazine, vol. 20, no. 1, pp. 99-116, 1999.
[25] Macfadzean, R.H.M., Surface-Based Air Defense System Analysis, Artech House, 1992.
[26] Durfee, E.H. and Montgomery, T.A., MICE: A flexible testbed for intelligent coordination experiments, In Proceedings of the 1989 Distributed AI Workshop, pp. 25-40, Sept. 1989.
[27] Clark, P. and Niblett, T., The CN2 Induction Algorithm, Machine Learning Journal, Vol. 3, No. 4, pp. 261-283, 1989.
[28] Quinlan, J.R., C4.5 Programs for Machine Learning, Morgan Kaufmann, 1988.
[29] Cameron-Jones, R.M. and Quinlan, J.R., Efficient top-down induction of logic programs, SIGART Bulletin, vol. 5, no. 1, pp. 33-42, Jan. 1994.

노 상 욱

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 조교수. 2002년~현재 가톨릭대학교 컴퓨터정보공학부 조교수. 관심분야는 지능형 에이전트, 멀티 에이전트 시스템, 지식관리, 기계학습, 실시간 분산 시스템 등