KIPS Trans. Softw. and Data Eng.
Vol.10, No.4 pp.143~152
ISSN: 2287-5905 (Print), ISSN: 2734-0503 (Online)

https://doi.org/10.3745/KTSDE.2021.10.4.143

C-COMA: A Continual Reinforcement Learning Model for Dynamic Multiagent Environments

Kyueyeol Jung† · Incheol Kim††

ABSTRACT

It is very important to learn behavioral policies that allow multiple agents to work together organically for common goals in various real-world applications. In this multi-agent reinforcement learning (MARL) environment, most existing studies have adopted centralized training with decentralized execution (CTDE) methods as in effect standard frameworks. However, this multi-agent reinforcement learning method is difficult to effectively cope with in a dynamic environment in which new environmental changes that are not experienced during training time may constantly occur in real life situations. In order to effectively cope with this dynamic environment, this paper proposes a novel multi-agent reinforcement learning system, C-COMA. C-COMA is a continual learning model that assumes actual situations from the beginning and continuously learns the cooperative behavior policies of agents without dividing the training time and execution time of the agents separately. In this paper, we demonstrate the effectiveness and excellence of the proposed model C-COMA by implementing a dynamic mini-game based on Starcraft II, a representative real-time strategy game, and conducting various experiments using this environment.

Keywords : Multiagent Reinforcement Learning, Dynamic Environment, Continual Learning, Starcraft II

C-COMA: 동적 다중 에이전트 환경을 위한 지속적인 강화 학습 모델

정 규 열† · 김 인 철††

요 약

다양한 실세계 응용 분야들에서 공동의 목표를 위해 여러 에이전트들이 상호 유기적으로 협력할 수 있는 행동 정책을 배우는 것은 매우 중요하다. 이러한 다중 에이전트 강화 학습(MARL) 환경에서 기존의 연구들은 대부분 중앙-집중형 훈련과 분산형 실행(CTDE) 방식을 사실상 표준 프레임워크로 채택해왔다. 하지만 이러한 다중 에이전트 강화 학습 방식은 훈련 시간 동안에는 경험하지 못한 새로운 환경 변화가 실전 상황에서 끊임없이 발생할 수 있는 동적 환경에서는 효과적으로 대처하기 어렵다. 이러한 동적 환경에 효과적으로 대응하기 위해, 본 논문에서는 새로운 다중 에이전트 강화 학습 체계인 C-COMA를 제안한다. C-COMA는 에이전트들의 훈련 시간과 실행 시간을 따로 나누지 않고, 처음부터 실전 상황을 가정하고 지속적으로 에이전트들의 협력적 행동 정책을 학습해나가는 지속 학습 모델이다. 본 논문에서는 대표적인 실시간 전략게임인 StarcraftII를 토대로 동적 미니게임을 구현하고 이 환경을 이용한 다양한 실험들을 수행함으로써, 제안 모델인 C-COMA의 효과와 우수성을 입증한다.

키워드 : 다중 에어전트 강화 학습, 동적 환경, 지속 학습, Starcraft II

1. 서 론

일반적으로 현실 세계는 여러 자율 에이전트들이 공존하는 다중 에이전트 환경이다. 예를 들어 교통 신호등 제어, 자율 주행 차량의 제어, 다수의 플레이어가 활동하는 비디오 게임 등 다중 에이전트 환경으로 구성되어 있다. 이러한 다중 에이전트 환경에서 각 에이전트는 경우에 따라서 어떤 때에는 서로 유기적으로 협력해야 하고, 어떤 때로 서로 경쟁하거나 적대적으로 행동해야 한다. 단일 에이전트의 경우와 마찬가지로, 다중 에이전트 행동 정책 학습을 위해서도 그동안 많은 심층 강화 학습(deep reinforcement learning) 기술들이 소개되었으며, 바둑이나 atari 비디오 게임 등과 같은 비교적 고난도 작업들에서도 큰 성공을 보여주고 있다. 하지만 다중 에이전트 환경은 다음과 같은 요소들 때문에 아직도 효율적

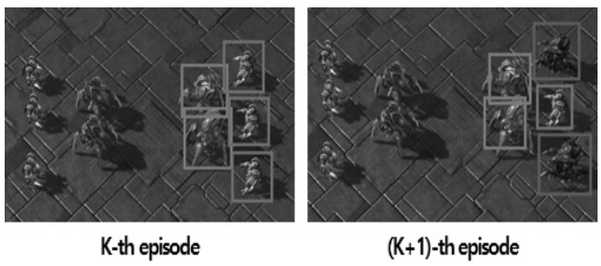
※ 정보통신기획평가원/정보통신방송 기술개발사업/클라우드에 연결된 개별 로봇 및 로봇그룹의 작업 계획 기술 개발 / 2020-0-00096.
※ 이 논문은 2020년 한국정보처리학회 추계학술발표대회의 우수논문으로 "동적 환경에서의 지속적인 다중 에이전트 강화 학습"의 제목으로 발표된 논문을 확장한 것임.
† 준 회 원 : 경기대학교 컴퓨터과학과 석사과정
†† 종신회원 : 경기대학교 컴퓨터과학과 교수
Manuscript Received : December 14, 2020
Accepted : December 25, 2020
* Corresponding Author : Incheol Kim(kic@kyonggi.ac.kr)

으로 행동 정책을 학습하기 어려운 경우가 많다. 첫째 대부분의 다중 에이전트 환경에서는 각 에이전트에게 환경에 대한 완전한 상태 정보(complete state information)는 주어지지 않고, 다만 부분적이고 불완전한 관측 정보(partial, incomplete observation)만 주어진다. 또한 많은 다중 에이전트 환경에서는 에이전트들 간의 통신(communication)이 전혀 불가능하거나 혹은 매우 제한적으로만 가능하다. 따라서 효율적인 팀워크 활동을 위해 필수적으로 요구되는 팀원들 간의 실시간 정보 교환(information exchange) 및 행위 조율(behavior coordination)이 어려워진다.

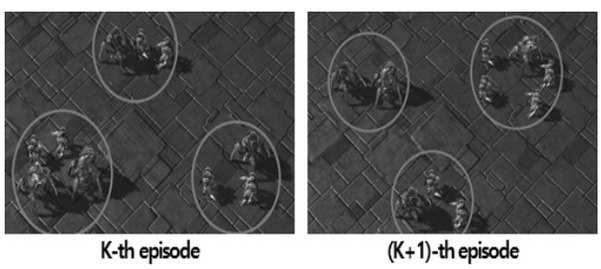
이러한 문제를 극복하고 에이전트 각각이 유기적으로 협동하여 팀 차원의 효율적인 행동을 수행하기 위해 그동안 다양한 다중 에이전트 강화 학습(Multi Agent Reinforcement Learning, MARL)기법들이 소개되었다. 특히 최근에는 실시간 전략게임인 StarcraftⅡ 환경을 다루는 다중 에이전트 챌린지(StarCraft Multi-Agent Challenge, SMAC)[1]를 중심으로 중앙 집중형 훈련과 분산형 실행(Centralized Training with Decentralized Execution, CTDE) 체제가 널리 보급되고 있다. 이러한 다중 에이전트 강화 학습의 대표 모델들로는 행동자-비평가(actor-critic) 구조에 기반한 COMA[2]와 Q 학습에 기반한 VDN[3], QMIX[4] 등이 있다. 이 학습 모델들의 가장 큰 특징은 중앙의 조정자 도움을 받아 각 에이전트가 최적의 Q 함수나 개별 행동 정책을 학습하고 나면, 실행 중에는 더 이상 학습을 진행하지 않는다. 다시 말해 학습 시와는 달리 수행 시에는 환경으로부터 추가적인 피드백과 이에 기초한 각 에이전트의 가치 함수(value function)나 정책 네트워크(policy network)의 변화는 발생하지 않는다.

이러한 CTDE 기반의 다중 에이전트 강화 학습 모델들은 에피소드마다 아군 에이전트와 적군의 개체 수, 구성, 등장 위치 등등의 요소들이 고정되어있는 정적 환경(static environment)에서는 좋은 성능을 발휘할 수 있다. 하지만 Fig. 1의 (a)와 같이 에피소드마다 적군의 구성이 달라지거나, Fig. 1의 (b)와 같이 적군의 등장 위치가 변동되는 동적 변화가 학습 시간 이후에도 계속 발생할 수 있는 동적 환경에서는 큰 효과를 기대하기 어렵다. 따라서 경험하지 못한 새로운 환경 변화가 계속 발생하는 동적 환경에서도 이 변화에 적응하면서 효율적으로 팀 단위의 협력적 행동 정책을 학습할 수 있는 새로운 다중 에이전트 강화 학습 모델의 개발이 필요하다.

본 논문에서는 경험하지 못한 상황이 계속해서 발생하는 동적인 환경에서도 좋은 성능을 발휘하기 위한 새로운 지속적인 다중 에이전트 강화 학습 모델 C-COMA(Continual COMA)를 제안한다. 이 모델은 행동자-비평가(actor-critic) 구조를 기반으로 비-사실적 추론(counterfactual reasoning)이 가능한 COMA[2]를 지속 학습(continual learning)이 가능하도록 확장한 모델이다. 즉 C-COMA는 에이전트들의 훈련 시간과 실행 시간을 따로 나누지 않고, 처음부터 실전 상황을 가정하고 지속적으로 에이전트들의 협력적 행동 정책을 학습해나가는 지속 학습 모델이다. 본 논문에서는 대표적인



(a) Changes in team composition



(b) Changes in combat position

Fig. 1. Dynamic Changes in a StarcraftⅡ Mini Game

실시간 전략게임인 StarcraftⅡ를 토대로 동적 미니게임을 구현하고 이 환경을 이용한 다양한 실험들을 수행함으로써, 제안 모델인 C-COMA의 효과와 우수성을 입증한다.

2. 관련 연구

초창기의 다중 에이전트 강화학습의 접근법들은 개별 에이전트가 독립적으로 자신의 가치 함수 및 정책을 학습하도록 제안되었다. 일반적으로 Q 학습 기반의 독립적인 Q 학습(Independent Q-Learning, IQL)[5]의 경우 각 에이전트가 독립적으로 Q 학습[6]을 진행하며 추가로 DQN[7] 기반의 심층 강화학습으로 확장한 기법[8]도 제안되었다. 다시 말해 IQL은 단순히 분산형 실행을 수행한다. 따라서 에이전트들을 동시에 학습 및 탐색으로 발생 되는 불안정성(non-stationarity) 때문에 효과적인 수행이 어렵다. 이러한 불안정성을 해결하기 위하여 과거 경험에 대하여 안정성에 집중한 기법[9]이 소개되었다. 그러나 이 기법 또한 여전히 각 에이전트들이 제한된 관측 능력을 토대로 서로 독립적으로 행동 정책을 학습하기 때문에, 팀원들끼리 조율화된 협동 행동들을 보여줄 수 있는 다중 에이전트 행동 정책을 얻기가 쉽지 않다

이와 반대로 에이전트들의 독립성을 완전히 배제하여 중앙-집중형 훈련(centralized training)기법의 접근법들도 제안되었다. 중앙-집중형 훈련은 에이전트 사이의 협동을 자연스럽게 조절하여 불안정성을 개선하는 방법이다. 그러나 중앙-집중형 훈련은 에이전트 개별의 행동 공간의 크기 및 개체 수가 많아지게 되면 복잡성이 증대되어 효과적인 수행이 어렵다. 대표적으로 고전적인 중앙-집중형 훈련 방식은 협력 그래프(coordination graphs)[10]를 이용하는 방법이다. 협력

그래프 기반의 접근법은 에이전트 사이의 조건부 독립(conditional independencies)으로 형성된 보상을 누적하여 팀 차원의 전체 보상으로 이용한다. 이와 달리 테이블(tabular methods)을 이용한 Q 학습을 이용하여 협력이 필요할 때만 협력 그래프를 이용하여 학습하는 기법도[11] 존재한다. 이러한 방법들은 에이전트 사이의 정보가 미리 설정되어 있어야 하나 현실적으로 설정하기 매우 어렵다. 이를 해결하기 위해 중앙-집중형 훈련 기법을 유지하며 에이전트 사이의 소통(communication)을 허용하는 접근법도 제안되었다. 다시 말해 이러한 접근법들은 실행 중에 에이전트 사이의 소통을 좀 더 자주 진행하는 편이다. 소통을 진행하기 위하여 에이전트 사이의 정보교환을 위한 중앙-집중형 훈련을 위한 신경망을 사용하는 기법[12]이 제안되었다. 또한 행동자-비평가 형태의 에이전트 간의 소통을 위한 기법[13]도 제안되었다. 이 기법은 양방향 순환 신경망(bidirectional RNN)을 이용하여 정보 교환을 진행한다. 그리고 소통을 허용하는 기법들은 에이전트 개별에 대한 보상을 추정해야 한다.

그러나 에이전트 개별에 대한 보상을 추정하는 것은 매우 어려운 일이다. 따라서 에이전트의 독립성을 일정 부분적으로 허용해야 한다. 따라서 앞선 접근법들과 달리 에이전트의 독립성을 일정 부분적으로 허용하는 중앙-집중형 훈련과 분산형 실행(CTDE)이 개발되었다. CTDE 방식의 접근법은 Q 학습을 기반으로 하는 접근법들과 행동자-비평가 기반의 접근법들로 구분된다. 전자의 경우는 Q 학습 기반으로 하여 전체의 상태 가치함수(state value function)를 개별 에이전트의 상태 가치함수로 분해하여 평가하는 기법이다. 이러한 Q 학습 기반의 기법들은 먼저 에이전트들의 상태 가치함수를 전달받아 전체의 상태 가치함수를 선형(linear)[3] 또는 비선형(non-linear)[4] 통합한다. 이후 분해(factorization)하여 에이전트의 정책을 평가한다. 그러나 이런 방법은 복잡하고 어려운 환경에서는 잘 작동하지 않는다. 이와 달리 행동자-비평가 기반의 접근법인 COMA[2]의 경우는 분산형 실행을 하는 에이전트를 학습시키기 위해 중앙-비평가를 사용한다. 이는 에이전트 개인에 대한 비-사실적 추론(counterfactual reasoning)에 기반한 이득 함수(advantage function)를 산출하여 협력적 다중 에이전트에 환경에 적합한 정책을 수행하도록 도와준다. 한편 에이전트마다 비평가를 할당한 중앙-집중형 방법[14]도 제안되었다. 이 기법은 에이전트의 개체 수가 늘어나도 괜찮지만, 중앙-집중형 이득 값을 약화한다. 마지막으로 중앙-비평가가 에이전트마다 할당되어 있으며 연속적인 행동 공간(continuous action space)에서 수행된 활성 정책 기법[15]도 존재한다. 그러나 이 기법은 지역 최저점(local minima)에 쉽게 빠지게 된다.

앞서 제안된 CTDE 기법은 아군과 적군의 구성과 개체 수가 에피소드마다 고정된 환경(static environment)에서 수행되었다. 이와 달리 에이전트의 구성과 개체 수가 에피소드마다 변형되는 동적 환경(dynamic environment)에 대응하기 위하여 기존의 QMIX를 확장한 기법이 AI-QMIX[16] 기법이다. AI-QMIX는 멀티 헤드 어텐션(multi-head attention)을 도입하여 특정 상황에 집중하도록 유도하였다. 또한 이매지네이션(imagination) 기법을 도입하여 비슷한 상황이 발생할 때 과거의 학습 기억을 상상하여 활용하도록 접근하였다. 그러나 AI-QMIX는 경험하지 못한 상황이 발생하는 동적 환경에서 수행되지는 않았다.

대부분의 논문은 실시간 전략게임인 StarcraftⅡ의 유닛을 직접 일일이 제어(micromanagement) [1]를 위한 목적으로 제안되었다. 또 한 에피소드마다 아군과 적군의 개체 수와 구성이 변형되지 않은 고정된 정적 환경(static environment)에서 진행되었다. 다시 말해 과거의 접근법들은 경험하지 못한 상황이 발생하지 않는 환경에서 수행되었다. 그러나 본 논문에서는 경험하지 못한 상황이 발생하는 상황에 대응하기 위하여 동적 환경으로 수정하였다. 첫 번째 수정 사항은 적군의 구성을 에피소드마다 달라지도록 변형하였고 두 번째 수정사항은 적군의 등장 위치를 변형하였다. 이러한 변형으로 인해 개별 에이전트들이 학습 시 경험하지 못하는 상황을 발생시켰다.

3. 문제 정형화

본 논문에서는 부분 관측 정보를 이용하여 진행하는 기법인 Dec-POMDP(Decentralized Partially Observable Markov Decision Process)를 기초로, n 명의 에이전트들로 구성된 다중 에이전트 강화 학습 문제는 $G=\langle U, P, r, O, n, \gamma\rangle$와 같은 튜플(tuple)로 정의한다. 여기서 O는 에이전트들의 협동 관측 공간(joint observation space)이며, $O=\{O_1, O_2, \cdots, O_n\}$와 같이 개별 에이전트 i의 관측 공간 O_i의 조합으로 정의한다. 개별 에이전트 i의 부분관측(partial observation)은 $o_t \in O$이다. 매 시간 반복 마다 각 에이전트는 행동 $u^a \in U$을 선택하며, 이 행동들이 모여 하나의 다중 에이전트 협동 행동(joint action) $\boldsymbol{u} \in \boldsymbol{U} \equiv U^n$을 구성한다. 한편, 관측 전이 함수(observation transition function)는 $P(o^{'}|o, \boldsymbol{u}) : O \times U \times O \rightarrow [0, 1]$과 같이 정의한다. 에이전트들은 협동 행동에 대한 결과로서 동일한 보상 함수$r(o^{'}, \boldsymbol{u}) : O \times U \rightarrow R$를 공유하며, 보상에 대한 감가율(discount factor)은 $\gamma \in [0, 1]$이다. 각 에이전트 a는 과거 자신의 관측과 행동 정보인 $\tau^a \in T \equiv (O \times U)^*$를 가지고 있으며, 각 에이전트 a의 확률적 행동 정책(stochastic policy)은 $\pi^a(u^a|\tau^a) : T \times U \rightarrow [0, 1]$로 정의한다. 다중 에이전트들이 협동 행동들을 수행함으로써 환경으로부터 받는 감가된 보상의 합(discounted return)은 $R_t = \sum_{l=0}^{\infty} \gamma^l r_{t+l}$이다. 따라서 튜플 $G=\langle U, P, r, O, n, \gamma\rangle$로 정의하는 다중 에이전트 강화 학습(Mutltiagent Reinforcement Learning, MARL) 문제는 감가된 보상의 합 $R_t = \sum_{l=0}^{\infty} \gamma^l r_{t+l}$이 최대가 될 수 있는 각 에이전트 a의 확률적 행동 정책 $\pi^a(u^a|\tau^a) : T \times U \rightarrow [0, 1]$을 학습하는 것이다. 또한, 이러한 다중 에이전트 강화학습 문제에서 상

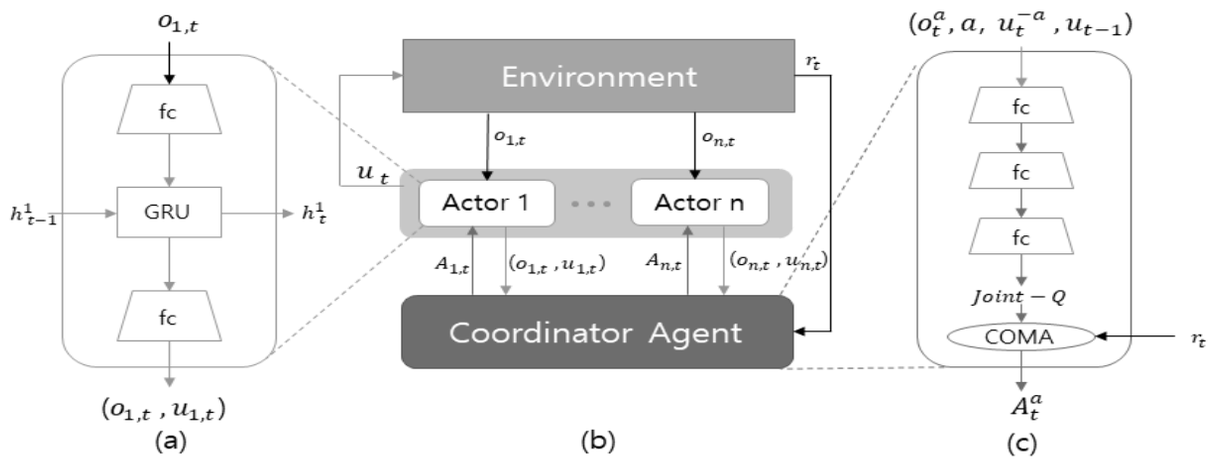


Fig. 2. Architecture of C-COMA

태 가치 함수는 $V^{\pi}(s_t) = E_{s_{t+1:\infty}, \mathbf{u}_{t:\infty}}[R_t|s_t]$로, 행동 가치 함수(action value function)는 $Q^{\pi}(s_t, \mathbf{u_t}) = E_{s_{t+1:\infty}, \mathbf{u}_{t:\infty}}[R_t|s_t, \mathbf{u_t}]$, 이득 함수(advantage function)는 $A^{\pi}(s_t, \mathbf{u_t}) = Q^{\pi}(s_t, \mathbf{u_t}) - V^{\pi}(s_t)$로 정의한다.

4. 지속적인 다중 에이전트 강화 학습

4.1 C-COMA 학습 모델

다중 에이전트 문제에서 가장 중요한 요소는 팀 보상을 통하여 각 에이전트 개인의 기여도를 적절히 부여하는 것이다. 더욱이 동적 환경에서는 복잡성이 증가하게 되어 문제가 어려워진다. 이를 효과적으로 해결하기 위하여 본 논문에서는 CTDE 기법의 대표적인 COMA[2] 기법을 지속학습의 목적에 맞게 변형하여 적용하였다. 왜냐하면, 다른 CTDE 기법들과 다르게 비-사실적 추론을 도입하여 팀 보상에 대한 행동 에이전트들의 기여도를 보다 효율적으로 산출한다. 이 기여도를 이용하여 에이전트의 정책을 팀 차원의 협력에 맞추도록 조정한다.

이에 기존 COMA의 전체적인 구조는 Fig. 2와 같이 두 가지 구성 요소로 이루어져 있다. 첫 번째는 Fig. 2의 (a) 와 같이 실질적으로 행동을 취하는 행동 에이전트(actor agent)와 두 번째는 Fig. 2의 (c) 와 같이 행동 에이전트들이 전달해주는 정보를 모두 받아 조율 역할을 하는 조정자 에이전트(coordinator agent)로 이루어져 있다. 행동 에이전트의 역할은 환경으로부터 부분 관측된 정보와 과거 정보를 이용하여 정책을 결정하고 정보를 조정자 에이전트에게 전달한다. 조정자 에이전트는 행동 에이전트들로부터 정보를 받아서 팀 차원의 관점으로 가치를 학습하고 행동 에이전트들이 개인이 아닌 팀 관점에 맞는 효율적인 행동을 수행하도록 조정한다. 이러한 조정에 의하여 행동 에이전트의 학습이 진행된다.

동적인 환경에 대응하기 위한 지속 학습을 위하여 COMA에 몇 가지 사항을 변경하였다. 첫 번째로 학습과 수행에 관계없이 계속 학습하도록 구조 변경하였다. 구체적으로 일반적인 CTDE 기법에서 학습 시에는 행동 에이전트 Fig. 2의 (a) 와 조정자 에이전트 Fig. 2의 (c) 모두 사용된다. 그러나 수행 시에는 조정자 에이전트는 사라지고 행동 에이전트만 남는다. 따라서 행동 에이전트들은 본인들의 학습된 정책을 통하여 수행만 하게 된다. 이렇게 되면 학습 시에 경험하지 못한 상황에 대처 능력이 떨어진다. 이에 따라 학습 시와 같이 수행 시에도 조정자 에이전트의 도움을 받아 새로운 환경에서도 지속해서 피드백이 가능하도록 변형하였다. 이 때 기존에 학습한 경험을 어느 정도 유지하며 새로운 경험에 적응해야 한다. 따라서 조정자 에이전트와 행동 에이전트 모두 과거의 학습 기록을 어느 정도 기억하기 위하여 지속학습을 적용하였다. 학습 중에는 Fig. 2의 (a), (c)에서 초록색 화살표와 같이 역전파 과정을 모두 적용한다. 즉, 학습을 진행할 때는 모든 신경망의 파라미터들을 수정한다. 이와 반대로 수행 중에는 붉은색 화살표와 같이 파라미터 일부분만 역전파를 진행하여 수정한다. 이 방법에 따라서 행동 에이전트와 조정자 에이전트의 신경망이 학습된다.

4.2 조정자 에이전트

조정자 에이전트는 환경에 등장하지 않는 가상의 에이전트이며 환경의 모든 정보를 수집하여 행동 에이전트들의 정책을 제어한다. 만일 행동 에이전트끼리만 정책을 수립한다면 팀 차원의 협동에 따른 정책을 수립하기 어려울 것이다. 그 이유는 본인들의 부분관측 정보만 볼 수 있을 뿐 전체 정보를 보지 못하기 때문이다. 따라서 전체 정보를 볼 수 있는 조정자 에이전트를 추가하여 행동 에이전트들이 팀을 위한 협동에 맞는 정책을 수립하도록 도움을 준다.

조정자 에이전트는 Fig. 2의 (b), (c)같이 행동 에이전트들로부터 부분관측 정보와 정책을 전달받아 행동 에이전트들이 팀 관점에 맞추어 정책을 결정하도록 유도한다. 다시 말해 조정자 에이전트는 행동 에이전트와 달리 전체 정보를 볼 수 있다. 그러나 행동 에이전트들은 전체 정보 중 본인이 볼 수 있

는 일부 정보만을 토대로 행동을 한다. 그러므로 팀 관점이 아닌 개인의 관점에서 행동을 수행할 수밖에 없다. 따라서 조정자 에이전트는 팀 전체 관점에서 가치를 높일 수 있는 각 행동 에이전트의 정책을 학습하도록 유도하는 역할을 한다. 이를 위하여 조정자 에이전트는 행동 에이전트들의 협동 행동에 대해 팀 차원에서 평가할 수 있는 Q 함수를 학습한다. 팀 차원의 협동 행동 가치 함수 Q를 학습하기 위하여, 조정자 에이전트는 Equation (1)의 $y^{(\lambda)}$ 와 같이 목표 네트워크(target network)를 설정한다. 이후 이 목표 네트워크는 신경망에 최적인 TD(λ) 기법을 통하여 값을 산출한다. TD(λ) 기법은 $G_t^{(n)} = \sum_{l=1}^{n} \gamma^{l-1} r_{t+l} + \gamma^n f^{co}(\bullet_{t+n}, \theta^{co})$ 같은 n 단계 보상(n-step return)을 이용한다. 이후 조정자 에이전트 $f^{co}(\bullet, \theta^{co})$는 활성 정책(on-policy) 방법으로 Q 함수 네트워크를 업데이트한다. 조정자 에이전트의 파라미터 θ^{co}는 Equation (1) 과 같이 매 시간마다 목표 네트워크의 결과값과 평균 제곱 오차(MSE)로 손실 함수 $L_t(\theta^{co})$를 정의할 수 있다. 이를 이용하여 경사 하강법(gradient descent)의 기법으로 조정자 에이전트는 보상을 학습한다.

$$
\begin{aligned}
&L_t(\theta^{co}) = (y^{(\lambda)} - f^{co}(\bullet_t, \theta^{co}))^2 \\
&where,\ y^{(\lambda)} = (1-\lambda)\sum_{n=1}^{\infty} \lambda^{n-1} G_t^{(n)} \\
&\theta^{co} = \theta^{co} - \nabla_{\theta^{co}} L_t(\theta^{co})
\end{aligned}
\tag{1}
$$

이 과정을 통하여 학습된 조정자 에이전트는 각 행동 에이전트의 이득 값을 부여한다. 일반적으로 이득 값은 Equation (2) 와 같이 TD-error 기법을 사용한다.

$$
g = \nabla_{\theta^a} \log \pi(u|\tau_t^a)(r + \gamma V(o_{t+1}) - V(o_t)) \tag{2}
$$

그러나 이 방법은 팀 차원의 보상만 추론할 수 있고 에이전트 각각의 기여도를 측정하기는 어렵다. 이를 위하여 비-사실적 추론 기법[2]으로 이득 값을 산출한다. 이 기법은 전체 팀 보상에서 각 에이전트의 보상 차(difference reward) $D^a = r(s, \boldsymbol{u}) - r(s, (\boldsymbol{u}^{-a}, c^a))$의 개념에서 출발한다. 다시 말해 행동 에이전트 a가 기본적 행동(default action) c^a를 하였을 때 전체 보상에서 개별 보상의 차이를 나타낸다. 이 기법은 상당히 강력한 방법이나 모든 가능성과 경우의 수를 고려해야 한다. 따라서 연산의 과정이 상당히 복잡해지며 비효율적이다. 따라서 Equation (3) 과 같이 다른 에이전트들의 행동은 고정하고 본인의 행동에 따라서만 확률을 고려하는 비-사실적 추론(counterfactual reasoning)기법으로 이득 값을 산출한다.

$$
A^a(o, \boldsymbol{u}) = Q(o, \boldsymbol{u}) - \sum_{u'^a} \pi^a(u'^a | \tau^a) Q(o, (\boldsymbol{u}^{-a}, u'^a)) \tag{3}
$$

그러나 이 자체로도 신경망에서 사용한다면 연산량이 상당히 커진다. 또 한 협동의 공간(joint-action space)도 $|U|^n$ 만큼 커지게 되고 효율적인 학습을 어렵게 한다. 따라서 representation 기법을 도입하였다. 구체적으로 Fig. 2의 (c) 와 같이 다른 에이전트의 행동 $\boldsymbol{u}_t^{-a}$을 신경망의 입력값으로 사용한다. 결과적으로 비-사실적 추론에 의한 이득(advantage by counterfactual reasoning) 값이 각 행동 에이전트들에게 효율적으로 전달이 된다. 또 한 협동의 공간(joint-action space)도 $|U|^n$에서 $|U|$로 감소하게 된다. 결론적으로 조정자 에이전트는 행동 에이전트가 전달해준 정보들을 활용해 팀 차원의 협동 행동 가치 함수 Q를 학습한다. 이를 토대로 팀원들인 각 행동 에이전트의 정책을 팀 관점에서 가치를 최대한 높일 수 있도록 조정하는 임무를 수행한다.

4.3 행동 에이전트

팀원들인 행동 에이전트는 실질적으로 환경에 놓여 자신이 수행할 행동 정책을 학습하고, 이에 따라 행동을 수행하는 역할을 담당한다. 각 행동 에이전트 i는 제한된 부분 관측 정보 $o_{i,t}$와 과거에 자신이 수행한 행동 이력 정보에 의존해 수행할 행동 $u_{i,t}$을 결정해야 한다. 그리고 이러한 각 행동 에이전트의 행동 $u_{i,t}$는 다른 에이전트들의 행동과 조화를 이루어 팀 차원의 이득을 극대화할 수 있어야 한다. 따라서 본 제안 모델에서는 Fig. 2의 (a)와 같이 게이트 순환 신경망(Gated Recurrent Units, GRU)을 기초로 각 행동 에이전트의 정책 네트워크를 구성하였다. 그리고 각 행동 에이전트의 부분 관측과 행동을 중앙의 조정자 에이전트에게 보내 Fig. 2의 (b) 와 같이 팀 차원의 가치 Q와 이에 따른 각 에이전트의 이득 $A_{i,t}$을 따져보고 이를 반영하여 행동 정책 π_θ을 갱신하도록 한다. Equation (4)는 이득 A에 기초한 경사 상승법(gradient ascent)을 적용하여 각 행동 에이전트의 정책 네트워크 파라미터 θ을 갱신하는 식을 나타낸다.

$$
\begin{aligned}
&\nabla_\theta J(\theta) = E_\pi[\sum_a \nabla_{\theta_a} \log \pi_{\theta_a}(u^a|\tau^a) A^a(o,u)] \\
&\theta = \theta + \alpha \nabla_\theta J(\theta)
\end{aligned}
\tag{4}
$$

5. 구현 및 실험

5.1 실험 환경과 모델 학습

본 논문의 제안 모델은 Windows 10에서 Python 딥러닝 라이브러리인 PyTorch를 이용하여 구현하였다. 모델의 학습 및 평가를 위하여 StarcraftⅡ의 미니게임을 이용한 다중 에이전트 학습환경(SMAC) [3]을 목적에 맞게 변형하여 실험을 수행하였다. SMAC은 실시간 전략게임인 StarcraftⅡ를 이용하여 다중 에이전트의 연구를 위해 제작된 미니게임을 제공한다. StarcraftⅡ의 경우는 두 가지 게임 특성이 있는데 하나는 자원관리 및 건물관리(macromanagement)이고 나머지 하나는 유닛을 직접 일일이 제어(micromanagement)하는 것이다. 본 논문에서는 후자 기반의 미니게임을 사용하

여 실험을 진행한다. 또 한 에이전트들의 부분관측 정보는 게임 환경에서 직접적으로 주어진다. 본 논문에서는 학습 환경과 수행 환경에서 실험하였고 모두 적군과 아군의 개체 수는 동일하게 설정하였다. 학습 환경은 수행 환경은 기존 SMAC에서 제공하는 미니게임을 아군은 변형이 없고 적군의 등장 위치 및 구성을 변형한 환경으로 수정하였다. 해당 환경은 StarcraftⅡ의 지도 편집기(map editor)를 이용하여 수정하였다. 등장 위치는 편집기 상에 임의로 배치했으며 구성 변경은 지도 편집기의 트리거 기능을 이용하여 구현하였다.

여기서 학습 환경은 위치 및 구성 일부분만 변경된다. 반대로 수행 환경은 학습 환경에서 경험한 상황에 더 추가로 등장 위치와 구성 변형을 하도록 수정하였다. 구체적으로 아군의 유닛은 추적자 두 마리와 광전사 세 마리로 구성되며 적군의 구성은 총 세 가지 유닛으로 광전사, 추적자, 히드라로 구성된다. 총 다섯 마리로써 추적자 두 마리는 고정이며 나머지 세 마리는 광전사와 히드라로 무작위 구성된다. 단 학습 환경에서는 히드라가 한 마리 또는 등장 안 할 수도 있다. 반대로 실행환경에서는 광전사와 히드라의 구성이 에피소드 단위로 무작위 변형된다. 다시 말해 근거리 공격을 하는 광전사가 원거리 공격을 하는 히드라로 변형되면서 문제는 더욱 어려워지며 학습으로 일부는 경험했던 상황이 발생하나 대부분은 경험하지 못했던 상황이 발생하게 된다. 또 한 적군의 경우 학습 기반의 인공지능이 아닌 StarcraftⅡ 내부의 게임 인공지능이며 난이도는 1부터 10까지 설정할 수 있고 본 실험은 난이도 7에서 수행하였다.

모델 학습을 위하여 조정자 에이전트의 레이어 수(number of layers)는 3이며 일반적인 학습과 달리 지속학습을 위하여 신경망 일부만 업데이트하도록 수정하였다. 행동 에이전트와 조정자 에이전트의 학습율(learning rate)은 모두 0.0005로 설정하였다. 보상을 학습하기 위한 TD(λ)의 값은 0.8로 설정하였으며, 목표 네트워크는 200회 반복마다 갱신하였다. 학습 중 행동 선택은 ε-greedy 방법을 이용하였으며, ϵ의 초기 값은 1로 설정하였다. 추후 50K 반복마다 ϵ값을 감쇄시키며 최소 0.05까지 감쇄된다. 마지막으로 경험에 대한 버퍼 크기(buffer size)는 5000으로 설정하였다. 실험과정 모두 총 8개의 프로세스로 병렬적으로 실험을 수행하였다. 실험은 64GB의 메인 메모리와 Geforce RTX 2080 TI 2개를 포함한 컴퓨터 환경에서 수행되었다.

5.2 성능 평가 실험

본 논문에서는 제안 모델인 C-COMA의 효과와 우수성을 입증하기 위해 StarcraftⅡ 동적 미니게임 환경을 이용한 다양한 실험을 수행한다. 첫 번째 실험은 본 논문에서 제안하는 지속적인 다중 에이전트 강화 학습 모델인 C-COMA가 지속학습인 불가능한 기존 모델들보다 동적 환경에서 우수성을 입증하는 실험이다. 이 실험을 위해 C-COMA를 비 지속 학습 모델들인 COMA[2], QMIX[4], VDN[3] 등과 비교하였다. 이들은 모두 중앙의 조정자가 팀 단위의 가치 평가를 담당하는 모델들로서, VDN[3]는 각 에이전트의 개별 Q 함숫값의 선형 조합으로, QMIX[4]는 비-선형 함수로 각각 팀 전체의 가치함수인 Q를 표현한다. COMA[2]의 경우는 행동자-비평가(actor-critic) 강화 학습을 기초로, 중앙의 조정자는 팀 단위의 가치 함수인 Q를 학습하고, 개별 에이전트는 자신의 지역 행동 정책(local policy)을 학습하는 역할을 담당한다. 대신 중앙의 조정자가 에이전트들의 협동 행동(joint action)에 관한 팀 전체 가치를 평가할 때,비-사실적 추론(counterfactual reasoning)을 적용한다. 이 실험에서 모델의 성능 평가 척도로는 총 30개의 에피소드 동안의 평균 승률(win rate)을 사용하였다. 에피소드 단위별로 변화가 일어나는 학습 환경에서 5M만큼의 학습 주기까지 진행 시킨 후, 수행 환경에서 1M 주기까지 테스트 실험을 진행하였다. 단 200K 주기까지는 학습 환경과 동일한 상황들이 발생 되지만, 그 이후로는 이전에 경험하지 못한 새로운 상황들이 동적으로 발생하도록 설정하였다.

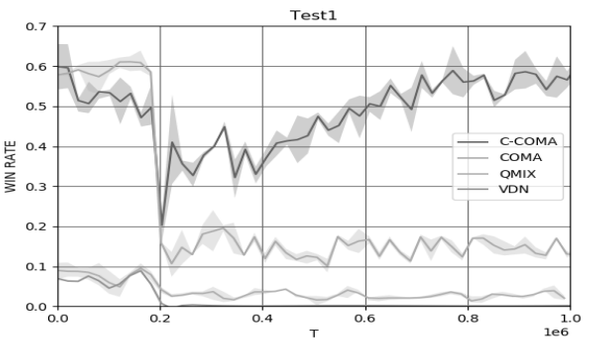


Fig. 3. Comparison with Non-continual Learning models: Win Rate

첫 번째 실험 결과는 Fig. 3에서 볼 수 있듯이, 지속학습을 진행 여부에 따라 성능의 차이가 상당하였음을 확인할 수 있다. 200K 반복을 기준으로 경험하지 않은 상황들이 발생하여 지속학습의 여부와 상관없이 모든 접근법들의 성능이 저하되었다. 지속 학습을 진행한 C-COMA의 경우는 이후 성능이 다시 향상되었으나 지속 학습을 진행하지 않은 접근법들은 저하된 성능이 회복되지 않았다. 이는 지속 학습을 진행한 경우 새로운 경험에 대하여 피드백 과정을 수행하기 때문이다. 반대로 비 지속 학습을 진행한 경우 새로운 경험에 효과적으로 대처하지 못하기 때문에 성능이 저하된 것으로 판단된다. 따라서 경험하지 못한 상황이 발생하는 동적 환경에서는 지속학습이 필요하다는 것을 알 수 있다. 그리고 지속학습을 진행하지 않는 경우 COMA의 경우가 나머지 QMIX와 VDN과 비교하여 성능이 우수하다. 그 이유는 팀 보상에 대한 행동 에이전트의 기여도를 비-사실적 추론을 도입한 COMA가 효과적으로 작동한 것으로 판단된다.

두 번째 실험은 COMA에 기초한 지속학습의 기능을 추가한 모델인 C-COMA가 다른 지속학습을 진행하는 모델들에 비해 우수성을 입증하는 실험이다. 이 실험에서는 제안 모델인 C-COMA를 QMIX 기반의 C-QMIX, VDN 기반의 C-VDN들과 성능을 비교한다.

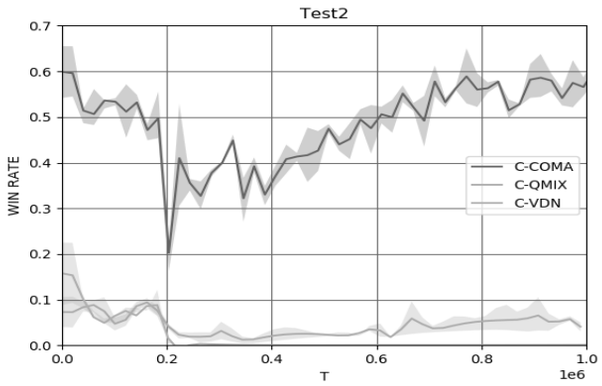


Fig. 4. Comparison with Continual Learning Models: Win Rate

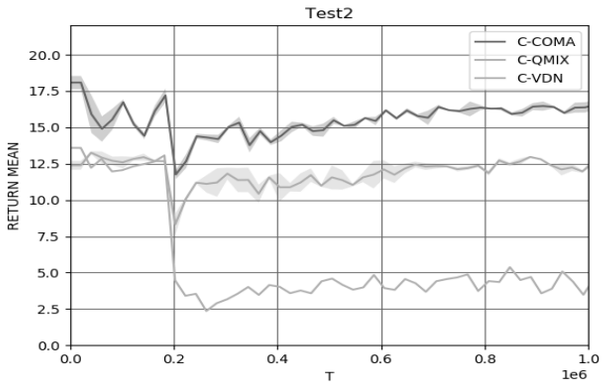


Fig. 5. Comparison with Continual Learning Models: Mean Return

두 번째 실험 결과는 Fig. 4에서 볼 수 있듯이, C-COMA의 경우 성능이 안정적으로 상승하였다. 반면에 C-QMIX의 경우, 승률이 10% 안팎에서 머물렀다. 또 한 C-VDN의 경우 성능 향상을 하지 못했다. 이는 조정자 에이전트의 역할이 중요함을 알 수 있는 결과이다. Q 학습 기반의 조정자들을 보면 C-QMIX의 성능이 C-VDN보다 약간 좋다는 것을 알 수 있다. 이는 행동 에이전트들의 Q 함수를 비선형으로 조합한 조정자 에이전트가 좀 더 효율적으로 행동 에이전트들을 제어한다고 판단된다. 그러나 C-COMA와 비교하면 성능의 차이가 상당하다. 이는 COMA처럼 비-사실적 추론이 행동 에이전트들의 행동을 팀 차원에 맞춰 수행하도록 효과적인 제어를 하였음을 알 수 있다.

승률 이외에도 정량적 평가를 위해 에피소드 마다 획득한 보상의 평균값(mean return)을 지표로 이용하여 지속학습을 진행하는 모델들을 비교하였다. 보상의 평균이 높으면 가치 있는 행동을 하였다고 볼 수 있다. 다시 말해 학습의 성능이 좋다고 말할 수 있다. 승률 실험과 동일하게 200K 실행 주기 까지는 경험하였던 상황이 발생한다. 그 이후에는 경험하지 못한 상황이 발생한다.

Fig. 5는 보상의 평균값의 추이를 나타낸다. C-VDN의 경우 경험하지 못하는 상황이 발생하였을 때 획득하는 보상이 감소하였고 더는 상승하지 않았다. 이와 반대로 C-QMIX의 경우는 획득 보상이 감소하였으나 얼마 지나지 않아 회복되었다. 이는 같은 Q 학습 기반의 접근법의 경우 행동 에이전트들의 Q 값을 선형으로 조합한 C-VDN 보다 비선형으로 조합한 C-QMIX가 더 좋은 보상을 획득했음을 알 수 있다. C-COMA의 경우도 마찬가지로 획득한 보상이 저하되었으나 곧 회복하여 상승하였다. 결국은 C-COMA가 C-QMIX보다 더 좋은 보상을 획득하였다. 이는 C-QMIX보다 행동가-비평가 기반에 비-사실적 추론 기반의 C-COMA가 더 효과적으로 학습을 수행한 것으로 판단된다.

다음은 본 논문에서 제안하는 지속적인 다중 에이전트 학습 모델인 C-COMA의 효과를 실제 사례를 통해 분석해보는 정성적 평가를 수행하였다. Fig. 6은 C-COMA의 지속 학습 효과를 잘 보여주는 예이다. 그림에서 원형의 노란색 점선은

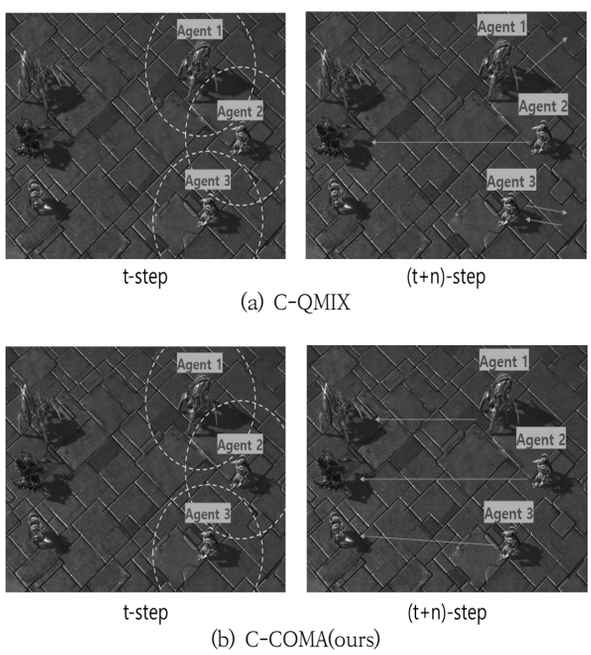


Fig. 6. Joint Action When Enemies are Out of Sight

행동 에이전트가 관측할 수 있는 범위이며 초록색 화살표는 이동을 나타낸다. 아군의 행동 에이전트들은 번호를 부여하였고, 차례대로 추적자, 광전사, 광전사이다.

적군의 등장 위치가 변하는 전투 환경에서는 Fig. 6과 같이 행동 에이전트들의 시야에서 적군이 탐지가 안 되는 경우도 발생한다. 이때 이상적인 행동은 행동 에이전트 자신이 공격할 수 있는 범위까지 적군에게 다가가야 한다. 특히 광전사는 근접 공격만 가능하기 때문에 적에게 가까이 접근해서 공격해야 한다. 이런 경우 C-COMA는 Fig. 6의 (b)와 같이 적군의 위치로 다가가서 전투를 진행하였다. 반대로 기존의 다른 다중 에이전트 강화 학습인 C-QMIX의 경우는 Fig. 6의 (a)와 같이 적군에게 다가가지 못하고 그 자리를 맴돌거나 일부만 적군에게 다가가는 경우가 많았다. 이러한 상황이 발생하는 경우 행동 에이전트들이 전투 상황 전체를 보지 못하기 때문에, 전체 상황을 볼 수 있는 조정자 에이전트의 역할이

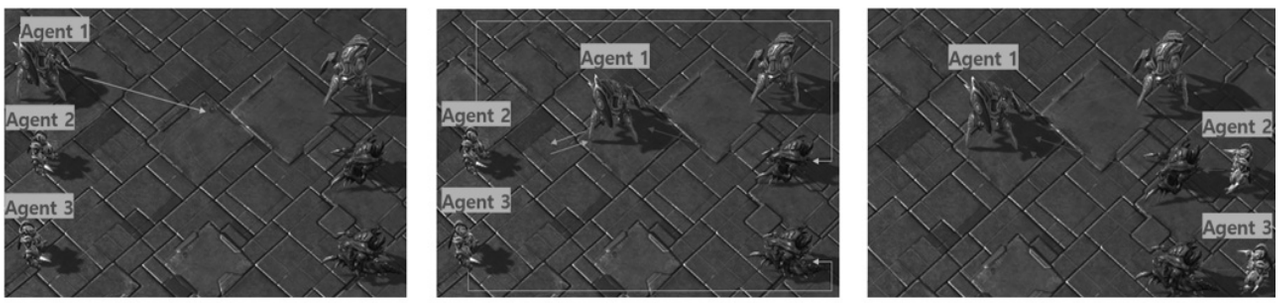


Fig. 7. Joint Action Learned by C-COMA When All Enemies are Distant

매우 중요하다. 이러한 상황이 발생하였을 때 Fig. 6의 (b)와 같이 C-COMA의 경우만 목적에 맞게 잘 행동하였다. 이는 기존의 다른 다중 에이전트 강화학습 모델들의 조정자 에이전트의 접근 방법에 대한 차이 때문에 발생한 것으로 판단된다.

Q 학습의 기반 모델인 C-VDN과 C-QMIX의 경우 행동 에이전트들의 Q 함숫값을 각자의 방법대로 통합한 후 행동 에이전트들의 정책과 상황에 대한 상관관계를 단순하게 분할하여 조정한다. 이 과정에서 행동 에이전트들은 누락된 정보를 받게 된다고 판단된다. 그러므로 정적 환경과 달리 동적 환경과 같이 복잡한 경우 Fig. 6의 (a)처럼 잘 작동하지 않았다. 반면에 C-COMA의 경우는 상황에 따라 다른 행동 에이전트들의 정책과 본인의 정책을 비교하며 가치평가를 하여 행동 에이전트의 정책에 도움을 준다. 행동 에이전트들이 앞선 C-QMIX와 달리 전체 정보를 온전히 전달 받을 수 있다. 이러한 특성의 결과로 행동 에이전트들이 적군의 위치를 탐지하지 못하고 경험해 보지 못한 동적 상황에서 조정자 에이전트가 효과적으로 대응하도록 유도하였다고 판단된다. 따라서 C-COMA의 조정자 에이전트가 VDN, QMIX 등을 토대로 한 다른 지속 다중 에이전트 강화 학습 모델들의 조정자 에이전트보다 팀 승리를 위한 협력적 행동을 하도록 행동 에이전트들의 정책을 더 효과적으로 수립하게 도움을 주었다고 판단된다. 그러나 적군으로부터 너무 멀리 있는 경우에는 C-COMA 역시 적군에게 접근하지 못하고 제자리를 맴돌았다. 이런 상황이 C-COMA의 승률이 일정 부분 이상 상승시키지 못하는 요인이었다. 향후에 이런 현상을 해결하기 위한 방법을 연구할 계획이다.

앞서 Fig. 6과 같이 적군의 등장 위치 변경되는 상황 이외에도 적군의 구성이 Fig. 7과 같이 변경되는 상황이 발생한다. 다음은 후자와 같이 적군의 구성이 변경되었을 때의 본 논문에서 제안하는 모델의 효과를 확인하기 위한 정성적 평가이다. 초록색 화살표는 유닛의 이동을 표현하며 붉은색 화살표는 공격을 표현한다. 또 한 Fig. 7의 (a), (b), (c)는 시간 순서대로의 흐름을 나타낸다. Fig. 7과 같이 적군의 유닛 중 근거리 공격을 하는 광전사가 모두 원거리 공격을 하는 히드라로 변경된다. 이와 반대로 아군은 근거리 공격을 하는 광전사 두 마리와 원거리 공격을 하는 추적자 한 마리로 이루어져 있다. 다시 말해 아군은 적군에 가까이 접근해서 공격을 수행해야 하므로 문제의 해결이 더 어려워진다.

이런 상황이 발생할 때 마찬가지로 모든 정보를 관측 가능한 조정자 에이전트의 역할이 매우 중요하다. 이런 경우 Q 학습의 기반 모델인 C-VDN과 C-QMIX은 관측결과 조정자 에이전트가 전략적인 수행을 하도록 행동 에이전트들을 잘 제어하지 못하였다. 반대로 C-COMA의 경우는 조정자 에이전트가 Fig. 7과 같이 전략적으로 수행하도록 행동 에이전트들을 지휘하였다. 본 실험에서는 두가지의 의미있는 상황이 포착 되었다.

첫 번째로 돋보이는 점은 Fig. 7의 (a)와 같이 아군의 1번 행동 에이전트인 추적자가 먼저 적군을 향해 접근하여 Fig. 7의 (b)의 위치로 이동한다. 이러한 행동의 이유는 조정자 에이전트가 적군의 공격 성질을 파악했기 때문으로 판단된다. 적군은 내장된 게임 인공지능(built-in game AI)으로서, 먼저 관측된 적에게 공격을 진행한다. 혹여 다른 적이 관측되어도 먼저 관측 및 공격 중인 적이 섬멸할 때까지 공격을 진행하고 관측된 다른 적은 무시한다. 이러한 성질을 조정자 에이전트가 간파하여 아군의 추적자에게 먼저 접근하도록 제어하여 적군 모두의 공격을 받아 희생하였다. 두 번째로 돋보이는 점은 적군은 공격 중인 목표물이 공격 범위를 벗어나면 현재 공격을 중지하고 목표물에 다가가서 공격을 진행한다. 다시 말해 적군의 공격 속도가 느려지게 되고, 이러한 성질 또한 조정자 에이전트가 간파하여 아군의 추적자가 Fig. 7의 (b)에 표현된 초록색 화살표와 같이 공격과 후퇴를 반복하도록 조정하였다. 이렇게 하면 적군이 연속적으로 공격하기 어려워진다.

앞선 두 가지의 행동 수행으로 인해 아군의 추적자가 희생 및 적군의 공격을 교란하는 동안 아군의 광전사들은 Fig. 7의 (b)의 초록색 화살표와 같이 적군의 히드라 뒤편으로 이동하여 Fig. 7의 (c)와 같은 상태로 변경된다. 이동 중에도 적군은 모두 추적자만 공격할 뿐 아군의 광전사는 무시한다. 이후 Fig. 7의 (c)와 같이 적군의 히드라 뒤편에서 공격을 진행하였다. 이후 적군의 히드라가 모두 섬멸되어 무력화되고 아군의 유닛들이 모두 적군의 추적자에게 집중공격을 진행하여 승리하였다. 따라서 제안 모델인 C-COMA의 경우 조정자 에이전트가 행동 에이전트들에게 전략적인 수행을 효과적으

로 유도하도록 역할을 하였다고 판단된다.

그러나 적군의 구성이 광전사와 히드라가 섞여 있는 경우, 다시 말해 근거리 공격을 하는 유닛과 원거리 공격을 하는 유닛이 섞여 있는 경우에는 아군의 행동 에이전트들이 효과적인 전투 수행을 하지 못했다. 이 부분 또한 승률 저하의 원인이었다. 따라서 이러한 현상을 해결하기 위해 향후 연구를 진행할 계획이다.

6. 결 론

본 논문에서는 동적 환경에 효과적으로 대응하기 위해, 새로운 다중 에이전트 강화 학습 체계인 C-COMA를 제안하였다. C-COMA는 에이전트들의 훈련 시간과 실행 시간을 따로 나누지 않고, 처음부터 실전 상황을 가정하고 지속적으로 에이전트들의 협력적 행동 정책을 학습해나가는 지속적인 다중 에이전트 강화 학습 체계이다. 본 논문에서는 대표적인 실시간 전략게임인 StarcraftⅡ를 토대로 동적 미니게임을 구현하고 이 환경을 이용한 다양한 실험들을 수행함으로써, 제안 모델인 C-COMA의 효과와 우수성을 입증하였다.

제안한 C-COMA는 승률이 두 가지 요인 때문에 0.6에서 더 오르지 않는 문제가 존재한다. 첫 번째는 적군이 시야 밖으로 너무 멀리 떨어진 경우이고 두 번째는 적군이 근거리 공격 유닛과 원거리 공격 유닛이 모두 섞여 있는 경우이다. 따라서 향후 승률을 더 높이기 위하여 추가적인 개선을 해볼 계획이다. 또 한 제시한 동적 환경에서 어떤 기준으로 행동 에이전트들이 정책을 수립했는지 현재로서는 설명할 방법이 없다. 따라서 동적인 다중 에이전트 환경에서 행동 에이전트들이 정책을 수행한 이유에 대해서도 모니터링이 가능한 방법을 찾아 분석할 계획이다.

References

[1] M. Samvelyan, T. Rashid, C. S. Witt, G. Farquhar, N. Nardelli, T. G. J. Rudner, C. M. Hung, P. H. S. Torr, J. N. Foerster, and S. Whiteson, "The StarCraft Multi-Agent Challenge," CoRR, abs/1902.04043, 2019.

[2] J. N. Foerster, G, Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[3] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, and T. Graepel, "Value-decomposition networks for cooperative multi-agent learning based on team reward," in *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2017.

[4] T. Rashid, M. Samvelyan, C. S. Witt, G. Farquhar, J. N. Foerster, and S. Whiteson, "Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp.4292-4301, 2018.

[5] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents." in *Proceedings of the Tenth International Conference on Machine Learning (ICML)*, pp.330-337, 1993.

[6] C. Watkins, "Learning from delayed rewards," Ph.D. Thesis, University of Cambridge England, 1989.

[7] V. Mnih, et al., "Human-level control through deep reinforcement learning," *Nature*, pp.529-533, 2015.

[8] A. Tampuu, et al., "Multiagent cooperation and competition with deep reinforcement learning," *PLoS ONE*, Vol.12, No.4, 2017.

[9] J. N. Foerster, et al., "Stabilising experience replay for deep multi-agent reinforcement learning," in *Proceedings of The 34th International Conference on Machine Learning (ICML)*, pp.1146-1155, 2017

[10] C. Guestrin, D. Koller, and R. Parr, "Multiagent planning with factored MDPs," *In Advances in Neural Information Processing Systems (NIPS)*, MIT Press, pp.1523-1530, 2002.

[11] J. R. Kok and N. Vlassis, "Collaborative multiagent reinforcement learning by payoff propagation," *Journal of Machine Learning Research*, pp.1789-1828, 2006.

[12] S. Sukhbaatar, R. Fergus, A. Szlam, and R. Fergus, "Learning multiagent communication with backpropagation," *In Advances in Neural Information Processing Systems (NIPS)*, pp.2244-2252, 2016.

[13] P. Peng, et al., "Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play StarCraft combat games," *In Advances in Neural Information Processing Systems (NIPS)*, 2017.

[14] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Springer, pp.66-83, 2017.

[15] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *In Advances in Neural Information Processing Systems (NIPS)*, pp.6382-6393, 2017.

[16] S. Iqbal, C. A, C. S. Witt, B. Penget, W. Böhmer, S. Whiteson, and F. Sha, "AI-QMIX: Attention and imagination for dynamic multi-agent reinforcement learning," *arXiv: 2006.04222*, 2020.

정 규 열

https://orcid.org/0000-0003-1768-5985
e-mail : raptorjung@gmail.com
2019년 경기대학교 컴퓨터과학과(학사)
2020년~현 재 경기대학교 컴퓨터과학과 석사과정
관심분야 : 멀티에이전트 강화학습, 게임인공지능지능, 로봇지능

김 인 철

https://orcid.org/0000-0002-5754-133X
e-mail : kic@kyonggi.ac.kr
1985년 서울대학교 수학과(이학사)
1987년 서울대학교 전산과학과(이학석사)
1995년 서울대학교 전산과학과(이학박사)
1996년~현 재 경기대학교 컴퓨터과학과 교수
관심분야 : 인공지능, 기계학습, 로봇지능