멀티 에이전트 강화학습 기술 동향

A Survey on Recent Advances in Multi-Agent Reinforcement Learning

유병현 (B.H. Yoo, bhyoo@etri.re.kr) 복합지능연구실 연구원
데브라니 데비 (D.D. Ningombam, nddevi@etri.re.kr) 정보전략부 Post-Doc.
김현우 (H.W. Kim, kimhw@etri.re.kr) 복합지능연구실 책임연구원
송화전 (H.J. Song, songhj@etri.re.kr) 복합지능연구실 책임연구원
박경문 (G.-M. Park, gmpark@etri.re.kr) 복합지능연구실 연구원
이성원 (S. Yi, sungyi@etri.re.kr) 정보전략부 책임연구원/부장

ABSTRACT

Several multi-agent reinforcement learning (MARL) algorithms have achieved overwhelming results in recent years. They have demonstrated their potential in solving complex problems in the field of real-time strategy online games, robotics, and autonomous vehicles. However these algorithms face many challenges when dealing with massive problem spaces in sparse reward environments. Based on the centralized training and decentralized execution (CTDE) architecture, the MARL algorithms discussed in the literature aim to solve the current challenges by formulating novel concepts of inter-agent modeling, credit assignment, multi-agent communication, and the exploration-exploitation dilemma. The fundamental objective of this paper is to deliver a comprehensive survey of existing MARL algorithms based on the problem statements rather than on the technologies. We also discuss several experimental frameworks to provide insight into the use of these algorithms and to motivate some promising directions for future research.

KEYWORDS 멀티 에이전트 강화학습(MARL), 에이전트 간 관계 모델링, 에이전트 간 신뢰할당, 에이전트 간 통신, 탐색-이용 딜레마, MARL 실험 환경

Ⅰ. 서론

하나의 에이전트가 주어진 환경에서 자신의 보상을 최대화하는 행동 또는 행동순서를 학습하는 강화학습은 딥러닝의 출현 이후 Deep Q-network[1]를 시작으로 다양한 연구들이 진행되어 왔다[2-4]. 그러나 강화학습이 실제로 적용되는 로봇, 자동차, 전장 등 다양한 분야의 특성을 고려할

* DOI: https://doi.org/10.22648/ETRI.2020.J.350614
* 본 연구는 한국전자통신연구원 연구운영지원사업의 일환으로 수행되었음[20ZS1100, 자율성장형 복합인공지능 원천기술 연구, 19YE1400, 멀티 에이전트 환경에서 인간-에이전트 협업기술 선행연구 및 개발환경 구축].

때 다수의 에이전트에 대한 고려는 필수적이다. 따라서 다수의 에이전트가 협업 또는 경쟁하는 환경에서의 문제를 강화학습을 통해 해결하려는 멀티 에이전트 강화학습(MARL: Multi-Agent Reinforcement Learning)과 관련된 연구가 주목을 받고 있다.

멀티 에이전트 강화학습을 기존의 싱글 에이전트 강화학습과 유사하게 완전 중앙집중형(Fully Centralized) 방식으로 접근하게 되면 공동 행동 공간(Joint Action Space)이 기하급수적으로 늘어나기 때문에 현실적으로 최적에 가까운 해를 찾기 어려워진다. 반면에 완전 분산형(Fully Decentralized) 방식을 이용하게 되면, 멀티 에이전트 강화학습을 통해 얻길 원하는 협업 또는 경쟁적 행동에 대해 학습하기 어렵다. 이러한 이유로 대다수의 연구에서는 알고리즘의 훈련(Training) 과정 중에는 모든 에이전트의 관측 정보를 이용하고, 실행 과정에서는 각 에이전트 자신의 관측 정보만을 이용하여 실행하는 방식인 중앙집중형 훈련-분산형 실행(CTDE: Centralized Training and Decentralized Execution) 메커니즘을 기반으로 멀티 에이전트 강화학습 알고리즘을 구성한다.

이러한 환경을 바탕으로 최근 멀티 에이전트 강화학습에 대한 몇몇 문헌 조사가 발표되었다[5,6]. 참고문헌 [5]는 멀티 에이전트 강화학습의 이론적 배경이 되는 싱글 에이전트 강화학습, 가치 함수기반 방법(Value Function Based Method)과 정책 경사기반 방법(Policy Gradient Based Method) 등을 설명하고 이를 바탕으로 최근의 연구를 에이전트의 관측 조건을 기준으로 마르코프/통계적 게임과 확장형 게임으로 분류하여 설명한다. 또한 참고문헌 [6]에서는 최근 멀티 에이전트 강화학습 연구를 독립학습, 완전 관측 가능한 Critic, 가치 함수 분해(Value Factorization), 합의, 통신학습 등으로 분류하여 설명하였다.

그러나 최근 연구동향을 고려할 때 기술 또는 이론 중심의 관점으로 하나의 연구를 분류하기는 매우 어렵다. 이는 최근 논문들이 이러한 기술 및 이론을 바탕으로 하되 다양한 기술을 복합적으로 적용하여 문제에 특화된 해결방안을 제시하기 때문이다.

본 고에서는 사전연구와는 달리 기술이나 이론적 기반보다는 각 연구에서 해결하고자 하는 문제를 중심으로 멀티 에이전트 강화학습 알고리즘을 분류하여 기술하고자 한다. 우선, Ⅱ장 1절에서는 다른 에이전트의 상태와 행동 등의 정보를 이용하여 에이전트 간 영향을 모델링하는 대표적인 연구를 소개한다. 2절에서는 에이전트의 행동이 전체 문제 해결에 얼마나 기여하는지를 다루는 에이전트 간 신뢰할당에 대한 연구를 다룬다. 3절에서는 실행 시 에이전트가 자신의 정보만 이용해야 하는 제약 극복을 위해 사용되는 에이전트 간의 통신 문제를 다루고, 마지막으로 강화학습의 고전적인 문제인 탐색-이용 딜레마를 MARL 환경에서 접근한 연구들을 소개한다. Ⅲ장에서는 멀티 에이전트 강화학습의 비교 및 검증을 위해 널리 사용되고 있는 실험 환경에 대해 추가로 기술한다.

Ⅱ. 멀티 에이전트 강화학습 알고리즘

1. 에이전트 간의 관계 모델링

멀티 에이전트 강화학습에서는 싱글 에이전트 강화학습과는 달리, 협업 또는 경쟁에 대한 다수의 에이전트의 최적 행동을 찾아야 한다. 그래서 다른 에이전트들에 대한 영향이나 에이전트 간의 관계가 정책이나 행동-가치 함수(Action-Value Function)에 반영되어야 한다. 다른 에이전트에 대한 영향을 멀티 에이전트 강화학습에서 본격적으로 고려하기 시작한 알고리즘으로는 MADDPG

(Multi-Agent Deep Deterministic Policy Gradient)[7]가 있다. MADDPG는 싱글 에이전트 강화학습에서의 대표적인 정책 경사기반 방법인 DDPG(Deep Deterministic Policy Gradient)[4] 알고리즘을 기반으로 하는 알고리즘으로서 다른 에이전트의 상태와 행동을 직접 행동-가치 함수의 입력으로 사용하여, 다른 에이전트의 영향을 고려하는 정책을 찾는 알고리즘이다. 다른 에이전트의 상태와 행동을 기반으로 에이전트의 영향을 직접 고려하기 때문에 협업 또는 경쟁적인 시나리오에서 모두 적용이 가능하며, 두 상황이 조합되어 있는 시나리오에서도 적용이 가능하다. MADDPG는 DDPG 대비 멀티 에이전트 환경에서 더 높은 성능의 정책을 찾는 것에 성공하였다. 그러나 MADDPG는 모든 에이전트의 상태와 행동을 입력으로 사용하기 때문에 에이전트의 수가 증가하거나 에이전트의 상태와 행동의 수가 늘어나면 정책 경사를 구하기 위한 목적함수가 기하급수적으로 복잡해진다. 이러한 문제를 해결하기 위해 MF-RL(Mean Field Reinforcement Learning)[8] 알고리즘이 제시되었다. MF-RL 알고리즘은 특정 에이전트의 정책을 계산할 때 해당 에이전트의 주변 범위 내의 에이전트들에 대한 평균 행동을 계산하여, 해당 에이전트의 정책 경사의 목적함수의 입력 또는 행동-가치 함수의 입력으로 사용하는 알고리즘이다. 다른 에이전트의 영향을 주변 에이전트의 평균 행동이라는 하나의 인자로 축약해서 표현하였기 때문에, 에이전트의 수가 많더라도 에이전트 간의 영향의 범위가 넓지 않고, 행동의 복잡도가 크지 않은 환경에서는 우수한 정책 또는 행동-가치 함수를 찾을 수 있으나, 에이전트 간의 영향의 범위가 넓은 시나리오에서는 적용에 한계가 있다.

MAAC(Multi-Actor-Attention-Critic)[9] 알고리즘은 인공 신경망 형태로 구성된 어텐션 메커니즘을 이용하여 상황별로 에이전트 간의 중요도를 계산하여 정책 경사의 목적함수의 입력으로 사용한 알고리즘이며, 싱글 에이전트 강화학습 알고리즘 중 Soft Actor-Critic[10] 알고리즘을 기반으로 구성되었다. 에이전트 간의 관계를 어텐션을 기반으로 추정하고 계산하기 때문에 협업 또는 경쟁적 시나리오에서 모두 적용이 가능하였고, MADDPG와 같이 다른 에이전트의 상태와 행동을 직접 입력으로 사용하는 것 대비 효율적으로 에이전트 간의 관계를 모델링할 수 있다. 또한, 다른 에이전트의 영향을 어텐션으로 표현하기 때문에 동일 환경에서 에이전트 수가 늘어날 때도 성능의 큰 저하 없이 주어진 문제를 해결할 수 있었다. 어텐션 자체가 에이전트 간의 중요도를 나타낼 수 있는 지표이기 때문에 학습이 진행된 이후에는 어텐션을 학습의 결과를 분석하는 용도로도 사용할 수 있다는 장점 또한 존재한다. MAAC 알고리즘이 특별한 가정 없이 일반적인 상황에서 어텐션을 적용한 알고리즘이었다면, HAMA(Hierarchical graph Attention-based Multi-agent Actor-critic)[11] 알고리즘은 에이전트의 특성에 따라 그룹을 구성하여 그래프 어텐션 메커니즘을 기반으로 그룹별 어텐션을 계산하고 정책 경사에 적용한 알고리즘이다. 에이전트별로 특성이 명확히 구분되어 그룹을 만들기 쉬운 환경에서는 HAMA 알고리즘이 기존 알고리즘 대비 높은 성능을 보였고, 그룹별로 어텐션을 계산하기 때문에 그룹 내의 에이전트 수가 늘어나더라도 성능 저하 없이 알고리즘이 작동하였다.

2. 에이전트 간의 신뢰할당

다수의 에이전트가 운용될 때, 환경으로부터 에이전트마다 별도의 보상을 부여받는 때도 있지만, 문제에 따라서는 모든 에이전트의 행동에 대해 하

나의 공동 보상(Team Reward)만 제공되기도 한다. 이때, 환경으로부터 얻어진 보상이 어떤 에이전트의 기여에 의한 것인지를 명확히 구분하고 정량화할 수 있으면 에이전트마다 보상에 대한 기여를 정확히 분배할 수 있다. 에이전트에 대한 기여도를 분배하는 이러한 문제를 신뢰할당(Credit Assignment) 문제라고 한다.

COMA(COunterfactual Multi-Agent policy gradient)[12]는 멀티 에이전트 환경에서의 신뢰할당 문제를 해결하기 위해 새로운 형태의 이득 함수(Advantage Function)를 제안하였다. 기존의 이득 함수는 행동-가치 함수에서 상태-가치 함수(State-Value Function)와의 차이를 통해 계산되나, COMA에서는 상태-가치 함수 대신 다른 모든 에이전트의 행동이 고정된 상태에서 특정 에이전트의 행동에 대한 행동-가치 함수의 평균값을 이용해 이득 함수를 계산하게 된다. 각각의 에이전트의 행동에 대한 행동-가치 함수의 평균값은 그 행동에 대한 기준값이 되며, 특정 행동을 취할 때 기준값 대비 얼마나 좋은 행동인지에 대한 정도를 판단할 수 있게 하는 역할을 한다. 정책 경사기반 방법인 COMA와는 다르게 가치 함수기반 방법에서도 에이전트 간의 신뢰할당 문제를 해결하려는 시도가 있었다.

정책을 직접 개선해나가는 정책 경사기반 방법과는 다르게, 가치기반 방법에서는 행동-가치 함수를 추정하고 이로부터 최적 정책(Optimal Policy)을 도출하게 되는데, 공동 보상만 존재하는 멀티 에이전트 환경에서는 학습 단계에서 모든 에이전트의 관측 정보와 행동을 입력으로 하는 공동 행동-가치 함수(Joint Action-Value Function)를 계산할 수는 있으나 실행 단계에서는 각각의 에이전트는 자신의 관측 범위 내에 있는 정보만을 이용할 수 있어 공동 행동-가치 함수를 사용할 수 없다. 이러한 이유로 가치기반 방법에서는 각 에이전트의 행동-가치 함수의 역할을 하는 유틸리티(Utility) 함수를 구성하고, 유틸리티 함수와 공동 행동-가치 함수 간에 식 (1)을 만족하도록 함수의 형태를 구성한다.

$$\arg\max_{\boldsymbol{u}} Q_{\mathrm{jt}}(\boldsymbol{\tau}, \boldsymbol{u}) = \begin{pmatrix} \arg\max_{u^1} Q_1(\tau^1, u^1) \\ \cdots \\ \arg\max_{u^i} Q_i(\tau^i, u^i) \\ \cdots \\ \arg\max_{u^n} Q_n(\tau^n, u^n) \end{pmatrix} \quad (1)$$

u는 행동, τ는 행동과 관측 정보의 집합, Q_{jt}는 공동 행동-가치 함수, Q_i은 i번째 에이전트의 유틸리티 함수, n은 에이전트의 수를 의미한다. 공동 행동-가치 함수를 위의 조건을 만족하도록 유틸리티 함수로 분해하는 것을 가치 함수 분해라 한다. 가치 함수 분해가 성공적으로 수행되면 실행 단계에서도 에이전트는 자신의 유틸리티 함수만을 이용해 공동 행동-가치 함수가 높은 행동을 찾을 수 있다. 이러한 논리로 멀티 에이전트 환경에서의 가치 함수기반 방법에서는 식 (1)을 만족하는 가치 함수 분해 방법에 관한 연구가 많이 진행되었으며, 대표적인 연구로는 VDN, QMIX, QTRAN[13-15] 알고리즘이 있다.

VDN(Value Decomposition Networks)[13]에서는 각 에이전트의 유틸리티 함수의 합으로 공동 행동-가치 함수를 구성하여 식 (1)을 만족하게 했다. VDN에 비해 더 일반적인 형태로 유틸리티 함수와 공동 행동-가치 함수의 관계를 구성하기 위해 QMIX[14]에서는 유틸리티 함수를 입력으로 하는 단층의 인공 신경망으로 공동 행동-가치 함수를 계산하는 Mixing network를 제안하였다. 단순히 단층의 인공 신경망으로 Mixing network를 구성하면 식 (1)을 만족할 수 없으므로 QMIX에서는 식 (1)에

대한 충분조건으로서 식 (2)와 같이 공동 행동–가치 함수는 유틸리티 함수에 대한 단조 증가 함수여야 한다는 제약 조건을 적용하였다.

$$\frac{\partial Q_{\mathrm{jt}}}{\partial Q_i} \geq 0 \tag{2}$$

이 제약 조건을 만족하도록 Mixing network를 구성하기 위해 QMIX에서는 Mixing network의 가중치를 0 또는 양수로 설정하고 학습을 수행한다. QMIX는 VDN에 비해 복잡한 시나리오의 문제에서도 더 높은 성능을 보임을 확인하였다.

QMIX에서의 단조 증가에 대한 제약 조건은 쉽게 적용할 수 있지만, 공동 행동–가치 함수가 유틸리티 함수에 대해 단조 증가하지 않는 문제를 해결하기 위해서는 더 일반적인 해 공간(Solution Space)에서 해를 찾는 알고리즘이 필요하다. 예를 들어, 하나의 에이전트가 자신의 유틸리티 함수가 감소하는 행동을 선택하는 것이 모든 에이전트 관점에서 유리한 경우에는 QMIX로 최적의 정책을 얻지 못한다. 이렇게 더 일반적인 형태의 문제를 해결하기 위해서 QTRAN[15] 알고리즘이 제안되었다. QTRAN 알고리즘은 다음과 같은 또 다른 제약 조건인 식 (3)을 제안하고 해당 조건이 식 (1)에 대한 충분조건임을 증명함으로써 더 일반적인 형태의 문제를 해결할 수 있는 알고리즘을 제안하였다.

$$\begin{aligned}
&\sum_{i=1}^{n} Q_i(\tau_i, u_i) - Q_{\mathrm{jt}}(\boldsymbol{\tau}, \boldsymbol{u}) + V_{\mathrm{jt}}(\boldsymbol{\tau}) = \begin{cases} 0 & \boldsymbol{u} = \overline{\boldsymbol{u}} \\ \geq 0 & \boldsymbol{u} \neq \overline{\boldsymbol{u}} \end{cases} \\
&V_{\mathrm{jt}}(\boldsymbol{\tau}) = \max_{\boldsymbol{u}} Q_{\mathrm{jt}}(\boldsymbol{\tau}, \boldsymbol{u}) - \sum_{i=1}^{n} Q_i(\tau_i, \overline{u}_i)
\end{aligned} \tag{3}$$

V_{jt}는 공동 가치 함수(Joint Value Function), $\overline{u}$는 최적 행동이다. 위의 조건을 알고리즘상에 구현하기 위해서 식 (3)의 등호, 부등호 조건을 추가적인 손실 함수로 반영하였다. 단조 증가 제약 조건으로 풀기 어려운 문제에 대해 실제로 QTRAN은 더 우수한 성능을 보이는 것이 논문상에서 검증되었다.

에이전트 간의 신뢰할당 문제를 풀기 위한 또 다른 방법으로는 에이전트마다 개별적인 보상을 주되, 이 값을 학습을 통해 생성하여 부여하는 LIIR(Learning Individual Intrinsic Reward)[16] 알고리즘이 있다. LIIR은 정책 경사기반의 방법으로서 정책과 Critic에 대한 인공 신경망 외에 각 에이전트에 대한 개별 보상을 위한 별도의 인공 신경망을 구성하였다. 여기서 개별 보상은 환경으로부터 주어지는 외적 보상(Extrinsic Reward)이 아닌, 에이전트가 스스로 생성하는 내적 보상(Intrinsic Reward)이다. 개별 보상을 위한 인공 신경망으로부터 에이전트의 내적 보상을 계산하고, 에이전트는 환경으로부터 얻어진 외적인 공동 보상과 개별적으로 생성한 내적 보상의 합으로 전체 보상을 계산하게 된다. 이때 두 보상 간의 가중치는 별도로 설정되어야 한다. LIIR 알고리즘은 이중 수준 최적화(Bi–level Optimization)를 수행하며, 외적 보상과 내적 보상의 합을 최대로 하는 정책의 파라미터를 일차적으로 찾고, 그렇게 얻어진 정책의 파라미터를 기반으로 외적 보상만을 최대로 하는 개별 보상을 위한 인공 신경망의 파라미터를 최적화한다. 내적 보상의 형태로 에이전트마다 개별적으로 보상을 부여하기 때문에 높은 성능의 정책을 찾을 수 있고, 기존의 알고리즘으로 찾기 어려운 다양한 패턴의 정책을 찾을 수 있다는 장점이 있다는 것이 논문상에서 검증되었다.

3. 에이전트 간의 통신

CTDE 환경에서 각 에이전트는 학습 시 공동 행동–가치 함수를 학습하고 실행 시에는 자신의 관측 정보에만 의존하여 독립적으로 실행된다. 따라

서 다른 에이전트의 정책과 행동을 고려한 최적의 행동을 하기 어렵다. 이때 대안으로 자주 거론되는 해법이 에이전트 간의 통신이다.

멀티 에이전트 강화학습에 있어 통신 문제는 크게 다음의 두 가지 문제로 나누어 생각해 볼 수 있다. 하나는 주어진 네트워크 환경에서 가장 효율적인 통신방식을 찾는 문제이고, 다른 하나는 통신 자체를 하나의 행동으로 보았을 때 각 에이전트가 효율적인 통신 프로토콜을 학습할 수 있는가에 대한 문제이다.

가. 통신방식의 성능을 높일 수 있는가?

멀티 에이전트 강화학습에서 에이전트 간의 통신을 이용한 초기 연구로는 참고문헌 [17]이 있다. 참고문헌 [17]에서 에이전트들은 학습 시 그리고 실행 시에 서로 메시지를 주고받는다. 따라서 에이전트는 자신의 관측 정보와 다른 에이전트들로부터 수신한 메시지를 이용하여 학습하고 이에 따라 행동한다. 그러나 메시지를 보내는 에이전트들의 수가 증가함에 따라 각 에이전트가 수용해야 하는 신경망의 크기가 함께 증가하게 되고 신경망이 커지게 되면 학습의 효율성이 떨어지는 문제가 발생한다.

참고문헌 [18]은 통신 문제를 공유 메모리 디바이스를 통해 단순화한다. 메모리 디바이스는 에이전트가 디바이스와 상호작용함에 따라 점진적으로 에이전트들의 상태를 학습하고 이를 에이전트 간에 공유하는 데 사용된다. 에이전트는 행동을 취하기 전에 메모리 디바이스에서 다른 에이전트가 남긴 메시지를 가져오고 해석한다. 그 다음, 에이전트는 메모리 디바이스를 갱신함으로써 자신의 상태를 다른 에이전트들과 공유한다. 이 연구는 MADDPG기반의 메모리 디바이스를 통해 각 에이전트가 동시에 그리고 종단 간(End-to-End) 학습 가능한 통신 프로토콜을 제안했으며 다양한 환경에서 발생하는 에이전트 간 통신형태를 분석하고 시각화했다는 점에 의의가 있다. 그러나 실제상황에서 공유 디바이스와 같은 환경이 제공될 수 있는지, 에이전트 수가 증가함에 따라 동기화가 깨질 경우에는 성능에 미치는 영향 등은 추후 연구의 대상으로 남을 것으로 보인다.

참고문헌 [17]에서와 같이, 에이전트가 실행 시 가장 정확한 행동을 선택하기 위해서는 다른 모든 에이전트의 상태와 행동을 통신을 통해 지속해서 수신하여야 한다. 그러나 공유되는 정보량의 증가는 학습 자체를 어렵게 할 뿐 아니라, 정보공유를 위한 통신에도 오버헤드가 될 수밖에 없다. 참고문헌 [19]는 모든 에이전트의 상태 전체를 공유하는 대신 공유되는 정보의 양을 최소화하여 네트워크 부담을 줄이되 각 에이전트의 성능은 높이고자 한다. 이를 위해 에이전트는 자신의 행동-가치가 모호한 경우에만 정보를 공유하는 방법을 취한다. 이를 위해 에이전트는 행동-가치의 분산값이 작은 경우, 즉 가장 큰 두 행동-가치의 차가 작은 경우, 다른 에이전트에게 정보를 요청한다. 요청을 받은 에이전트는 자신의 행동-가치의 분산값이 큰 경우, 즉 자신의 행동이 명확한 경우에만 응답한다. 이처럼 공유하는 정보의 양을 제어함으로써 통신의 오버헤드를 최소화할 뿐만 아니라, 불필요하거나 잡음(Noise)이 될 수 있는 정보를 제거함으로써 학습을 효율을 높인다.

최근 발표된 연구에서는 MADDPG와 유사한 구조에 message Coordinator 네트워크를 추가함으로써 에이전트 간 통신에 소요되는 리소스 문제를 해결하고자 했다. 참고문헌 [20]에서 에이전트는 각자의 actorNet과 메시지 생성 네트워크를 가지나 criticNet을 함께 공유한다. 이때 통신채널 역할을 하는 Message CoordinatiorNet(MCN)이 에이전트 사

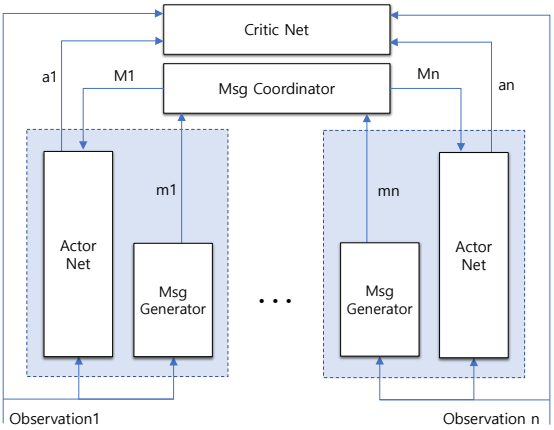

그림 1 Message Pruning 시스템 구조[20]

이에 공유된다(그림 1 참조). 에이전트는 MCN에 메시지를 보내고 MCN은 모든 로컬 메시지를 바탕으로 각 에이전트에 전송할 메시지를 추출한다. 따라서 각 에이전트는 자신의 관측 정보와 글로벌 메시지를 바탕으로 행동을 수행하게 된다. 전송되는 메시지의 양의 최소화에는 메시지 pruning 방법을 사용한다. 에이전트는 각자가 생성한 메시지와 함께 확률정보를 생성하고 여기에 인디케이터 함수를 적용함으로써 메시지를 생성하거나 삭제함으로써 전송하는 메시지의 양을 제어한다. 실험 시나리오에 따라 다소 차이가 있으나 제안된 방법은 매우 작은 보상값의 변화 범위 내에서 전체 네트워크 메시지를 약 30~70%까지 줄여주는 효과가 있는 것으로 보고된다.

각 에이전트가 독립적이고 분리된 환경에서 무선통신을 하는 경우, 대역폭(Bandwidth)과 medium access는 중요한 제한요소로 작용한다. 대역폭은 전송하는 정보의 양을 제한하고 medium access는 여러 에이전트 중 동시에 통신할 수 있는 에이전트의 수를 제한하게 된다. 이러한 제약 극복을 위해 참고문헌 [21]은 CSMA(Carrier Sense Multiple Access)기반 통신방식을 제안한다. 이 연구는 k개의 에이전트를 선택하고 선택된 에이전트만이 주어진 대역폭을 통해 메시지를 broadcast할 수 있게 함으로써 에이전트가 높은 누적 보상(Cumulative Reward)을 받을 수 있도록 하는 데 그 목적이 있다.

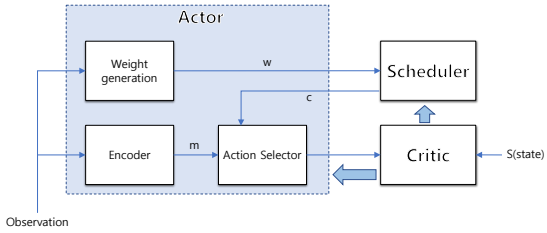

그림 2 SchedNet 시스템 구조[21]

이를 위해 제안된 Schednet은 actor, scheduler, critic 네트워크로 구성된다(그림 2 참조). Actor 네트워크는 각 에이전트들의 actor네트워크의 집합으로 다시 message encoder, action selector, weight generator로 구성된다. Message encoder는 압축된 메시지를 생성하고, weight generator는 에이전트별 관측을 기반으로 weight를 생성하며, Scheduler는 weight를 기반으로 top(k) 또는 softmax(k)를 이용하여 스케줄링 프로파일을 생성하는 역할을 한다.

나. 통신 프로토콜 자체를 학습할 수 있는가?

참고문헌 [22]는 각 에이전트가 언제 누구와 통신해야 하는지를 결정하는 통신 프로토콜 자체를 학습하는 데 그 목적이 있다. 일반적으로 에이전트가 하는 행동은 환경에 대한 것과 통신에 대한 것으로 나눌 수 있다. 이때, 통신에 대한 행동은 다른 에이전트가 볼 수는 있으나 보상이나 환경에 영향을 주지 않는다. 에이전트가 통신 프로토콜을 학습하는 가장 직관적인 방법으로는 에이전트가 환경과 통신의 행동에 대해 별도의 네트워크를 가지고 각각의 행동–가치값을 최대화하도록 학습하는 방법이다. 그리고 한 에이전트가 선택한 행

동을 다른 에이전트의 다음 입력으로 사용으로써 에이전트들은 상호작용하게 된다. 이러한 접근법을 RIAL(Reinforced Inter-Agent Learning)이라고 한다. RIAL은 통신 프로토콜을 학습하는 데는 유용하나 에이전트 간 통신 행동에 대한 직접적인(일반적인 인간의 대화에 필수적인 요소) 피드백을 반영할 수 없는 단점이 있다. 이를 위해 제안된 기술이 DIAL(Differentiable Inter-Agent Learning)이다.

DIAL은 RIAL에서의 파라미터 공유를 넘어 통신을 통해 에이전트 간에 경사(Gradient)값을 직접 전달한다. 즉, 학습 시 통신 행동 대신에 한 에이전트의 네트워크 결과값이 다른 에이전트의 입력으로 전달된다. 경사값이 에이전트 사이에서 전달됨으로써 에이전트들은 더 풍부한 피드백을 주고받게 되고, 이로 인해 시행착오에 의존해서 학습해야 하는 학습량을 줄일 수 있다. 이러한 아이디어는 협력적, 경쟁적, 혹은 이러한 관계가 혼재하는 다양한 환경에서 효율적인 것으로 보였으나 하나의 채널만을 사용하여 통신함으로써 에이전트가 통신을 참여할지 말지를 결정할 수 없는 문제는 향후 추가 연구의 대상이 될 것으로 보인다.

다른 연구에서는 사회적 영향을 내적 동기로 활용하여 통신 프로토콜을 학습한다. 참고문헌 [23]에서 각 에이전트가 통신 정책을 학습하는 데 사용되는 보상은 공동 보상과 사회적 영향력 보상의 합으로 이루어지는데, 사회적 영향력 보상은 통신으로 전달된 메시지가 다른 에이전트의 행동에 얼마나 영향을 주었는가로 결정된다. 즉, 한 에이전트의 행동이 다른 에이전트의 사회적 영향에 의해 영향을 받으면 해당 에이전트는 추가 보상을 받는다. 따라서 이 방식에서 받게 되는 보상은 공동 보상만을 고려하는 방식보다 항상 크게 되어 보다 좋은 학습곡선을 찾는 데 도움이 될 수 있다.

참고문헌 [24]는 Communication channel을 통해서 에이전트들이 협상의 성공 여부만을 보상으로 하여 협상 정책을 (강화학습을 통해) 학습할 수 있음을 보였다. 이 연구에 사용된 협상의 목표는 3개 유형의 항목을 에이전트가 협상을 통해 나누는 데 있다. 이때, 각 에이전트는 임의로 생성되는 항목별 유틸리티 함수를 받게 되며 자신의 유틸리티가 최대가 되도록 협상을 진행하게 된다. 협상을 위한 통신에는 에이전트의 행동과 직접 관련된 채널인 "제안 채널"과 전달한 메시지 내용의 진실 여부가 보장되지 않는 cheap talk이 사용된다. 연구의 결과는 제안 채널을 이용하여 에이전트들이 협상에 성공할 수 있으며, cheap talk을 이용하는 경우 에이전트들이 내시 균형(Nash Equilibrium)에 이를 수 있음을 보여준다. 특히, cheap talk의 경우 에이전트 간의 최적 협상의 편차를 줄여주는 효과가 있어 안정적인 결과를 얻는 데도 효과가 있음을 보여준다.

이어지는 연구에서는 "통신의 발생"을 화자(Speaker)와 청자(Listener) 사이의 상호작용으로 본다[25]. 통신의 자발적 발생을 위해서 이 연구에서는 positive signaling과 positive listening의 모델링을 위해 speaker loss와 listener loss를 사용한다. Speaker loss는 화자의 관점에서 다양한 상황에 따른 메시지를 만드는 데 사용되고, listener loss는 청자가 다른 메시지에 다르게 행동하도록 하기 위함이다. 즉, 화자의 메시지가 청자의 정책에 영향을 주고 청자의 다른 행동은 화자가 통신을 학습하도록 도움을 준다. 실험을 통해 이러한 loss의 정의가 통신을 쉽게 하며 효과적인 통신 프로토콜을 학습하는 데 도움이 됨을 밝혔다.

모든 에이전트의 "직접 통신"을 가정한 이제까지의 연구와는 다르게 참고문헌 [26]은 모든 에이전트가 특정 네트워크상에 있고 에이전트는 직접 이웃인 에이전트들만을 대상으로 통신할 수 있는 환경을 가정한다. 각 에이전트의 관측 정보는 주위

에이전트가 보내는 메시지에 의해 확장되고 메시지를 받은 에이전트는 받은 메시지의 품질을 경사값을 통해 피드백한다. 이렇게 함으로써 에이전트가 이웃들에게 메시지를 보내도록 유인하고 이 메시지를 통해 이웃들은 그들의 보상을 최대로 함으로써 각 에이전트가 협력하도록 한다. 이 연구는 네트워크 환경에 있어서 제약을 고려하였으나 간단한 메시지 피드백 메커니즘만을 이용해 에이전트가 스스로 통신을 학습할 수 있도록 했다는 점에서 의미가 있다.

4. 탐색-이용 딜레마

강화학습에서는 에이전트가 현재 상태에 대해 어떤 행동을 선택하여 다음 상태로 진행할 것인지를 정해야 한다. 행동 선택 시 기존에 에이전트가 학습한 정보를 기준으로 보상이 높을 행동을 이용(Exploitation)할지, 새로운 행동을 탐색(Exploration)할지 정하는 것은 어려운 문제이며, 탐색에 집중하는 경우 수렴 속도가 과하게 늦을 수 있고, 이용에 집중하는 경우 최적의 정책을 찾지 못할 수 있으므로 적절한 범위 안에서 탐색과 이용의 균형을 유지해야 한다. 이러한 문제를 탐색-이용 딜레마(Explore-Exploit Dilemma)라고 한다. 멀티 에이전트 강화학습에서는 다수의 에이전트가 존재하고 에이전트 간의 상호작용에 따라 해 공간이 상당히 넓어지기 때문에 이러한 탐색-이용 딜레마의 문제를 해결하기 위한 효율적 방안에 대한 중요성이 더 증대된다. 멀티 에이전트 강화학습에서의 탐색-이용 딜레마 문제를 해결하기 위해 EITI(Exploration via Information-Theoretic Influence)와 EDTI(Exploration via Decision-Theoretic Influence)[27] 알고리즘이 제안된 바 있다. EITI는 하나의 에이전트가 다른 에이전트의 행동 선택에 대해 미치는 영향을 상호의존 정보(Mutual Information)를 통해 계산하여 두 에이전트 간의 영향을 정량화하고, 이를 내적 보상의 형태로 적용한 알고리즘이다. EDTI는 하나의 에이전트가 다른 에이전트의 행동-가치 함수에 미치는 영향을 상호 의존 정보의 형태에 따라 계산하여 EITI와 마찬가지로 내적 보상의 형태로 적용한 알고리즘이다. 행동-가치 함수에는 행동 선택에 대한 정보가 포함되기 때문에 EDTI는 EITI에서 표현하는 에이전트 간의 정보에 대한 영향을 포함하며 동시에 보상에 대한 영향까지 고려하기 때문에 일반적으로 EDTI의 성능이 EITI보다 더 우수하다.

멀티 에이전트 강화학습에서의 탐색-이용 딜레마를 해결하기 위한 또 다른 연구로는 CTEDD (Centralized Training and Exploration with Decentralized execution via policy Distillation)[28] 알고리즘이 있다. 이 연구에서는 모든 에이전트의 관측 정보를 입력으로 받는 중앙집중형(Centralized) 인공 신경망을 구성하고, Policy distillation을 통해 각 에이전트가 각자 실행 가능한 정책을 구성할 수 있도록 인공 신경망을 구성하였다. 모든 에이전트에 대해 인공 신경망을 구성하기 때문에 에이전트의 행동을 탐색할 때도 중앙집중형 방식으로 진행되고, 이러한 탐색 방식은 에이전트마다 탐색을 수행하는 것에 비해 효과적인 탐색이 가능해진다.

탐색-이용 딜레마에 대한 문제를 해결하기 위한 또 다른 연구로는 가치 함수기반 방법에 대해 더 효과적인 탐색을 수행하고자 하는 MAVEN (Multi-Agent Variational Exploration)[29] 알고리즘이 제안된 바 있다. MAVEN에서는 기존의 QMIX 알고리즘에서의 제한적인 탐색을 극복하기 위해 잠재 변수(Latent Variable)를 기반으로 하는 계층적 정책(Hierarchical Policy)을 제안하고, 정책의 다양한 탐색을 위해 상호 의존 정보를 사용한 추가적인 손

실 함수를 이용해 다양한 상태에 대한 방문이 가능한 효과적인 탐색 방식을 제안하였다. MAVEN은 기존의 가치 함수기반 방법에서 해결하기 어려웠던 복잡하고 탐색 공간이 상당히 큰 문제를 해결함으로써 유의미한 결과를 보였다.

III. 실험 환경

상당수의 멀티 에이전트 강화학습 연구에서는 알고리즘 검증 시 연구의 방향성을 잘 보여줄 수 있는 형태의 환경을 새롭게 설계하고 그 환경 내에서 검증을 수행한다. 그러나 알고리즘 간의 공평한 성능 비교나 제안하는 알고리즘의 우수성을 명확히 보여주기 위해서는 공통으로 사용하는 검증 환경이 필요하며, 실제로 다양한 연구 기관에서 멀티 에이전트 강화학습용 검증 환경을 제시하고 있다. 본 절에서는 앞서 기술한 최신 멀티 에이전트 강화학습 알고리즘들이 검증되고 있는 주요 환경에 관해 기술하고자 한다.

1. Multi-agent particle environment

Multi-agent particle environment는 Open AI gym[30]을 기반으로 구성된 멀티 에이전트 강화학습용 검증 환경이다. 에이전트는 기본적인 물리 법칙하에 작동하는 2차원의 입자(Particle) 형태로 구성되며, 에이전트가 관측하는 정보는 연속적이고 행동은 이산적으로 설정되어 있다. 단순한 에이전트의 형태 대비 다양한 기능과 시나리오 설정이 가능하고, 협업과 경쟁 또는 그 두 형태가 융합된 시나리오 설정이 가능하며, 사용자가 원하는 시나리오를 직접 구성할 수도 있어 다양한 연구 논문에서 알고리즘 검증 시 활용하고 있다. 그림 3은 참고문헌 [7]을 참조하여 Multi-agent particle environment

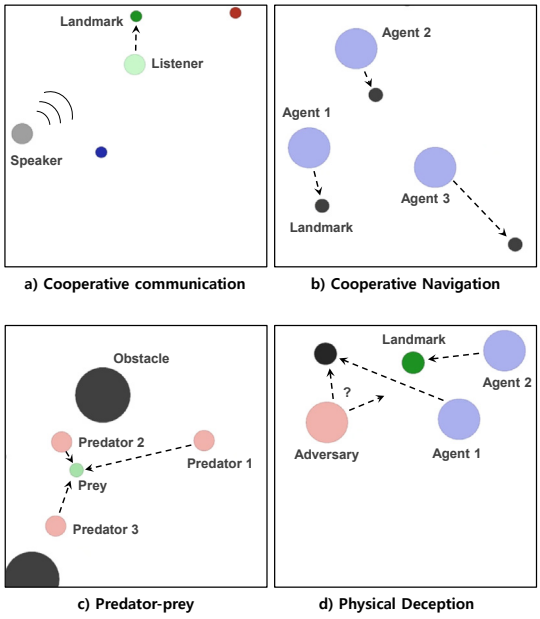

그림 3 Multi-agent particle environment에서의 멀티 에이전트 시나리오 예시[7]
(a) Cooperative communication, (b) Cooperative navigation, (c) Predator-prey, (d) Physical deception

에서의 시나리오 예시를 나타낸 것이며, 이 시나리오들은 실제로 다양한 MARL 논문의 검증에 활용된 바 있다.

그림 3의 시나리오 예시에 대해 간략히 기술하면, 그림 3(a)에서는 청자는 이동이 가능하나 정확한 목적지에 대한 정보를 모르는 상태이고, 화자는 이동이 불가능하나 목적지에 대한 정보를 알고 있어, 두 에이전트 간의 협업을 통해 청자가 화자로부터 정보를 전달받아 정해진 목적지에 도달하는 시나리오이다. 두 에이전트 간의 기능이 다르기 때문에 이종(Heterogeneous)의 에이전트 간의 협업이 필요한 시나리오이다. 그림 3(b)에서는 다수의 에이전트가 다수의 랜드마크에 각각 충돌 없이 도달해야 하는 시나리오이며, 동종(Homogeneous)의 에이전트 간에 협업이 필요한 시나리오이다. 그림 3(c)는 predator-prey로 널리 알려진 시나리오로

서, predator가 prey를 추적하게 되며, predator 입장에서는 다른 predator와는 협업을, prey와는 경쟁해야 하는 상황이기 때문에 협업과 경쟁이 융합되어 있는 시나리오이다. 그림 3(d)는 목적지에 대한 정보를 아는 에이전트들과 목적지에 대한 정보를 모르는 상대 에이전트(Adversary)가 존재할 때, 상대 에이전트에게 정확한 정보를 주지 않으면서 목표 지점에 도달하는 시나리오이다. Predator-prey와 마찬가지로 에이전트 간에는 협업을, 상대 에이전트와는 경쟁해야 하는 시나리오이다. 그림 3(d)와 같은 시나리오의 경우, 단순히 이동 또는 추적하는 것 외에 상대 에이전트에게 잘못된 판단을 하도록 유도하는 비교적 고차원적 행위를 구현한 시나리오이며, Multi-agent particle environment의 경우 이러한 다양하고 고차원적 시나리오의 구성이 가능하다.

2. StarCraft multi-agent challenge

StarCraft multi-agent challenge(SMAC)[31]는 기존의 블리자드사의 StarCraft II 게임에서 소규모 유닛 간의 전투에 집중해 만든 학습 환경으로서 멀티 에이전트의 학습 환경으로 다양한 연구에서 활용되고 있다. 하나의 게임 유닛이 하나의 에이전트에 해당하며 아군에 해당하는 에이전트들은 협업을 통해 적군에 해당하는 에이전트를 죽이는 것을 목표로 한다. 모든 아군 에이전트들은 받은 피해와 적군에게 가한 피해의 조합으로 계산되는 공동의 보상을 받게 설정되어 있고, 에이전트는 제한적인 공격 가능 범위와 관측 가능 범위가 설정되어 있다. 적군 에이전트는 내장된 스크립트 기반의 인공지능을 기반으로 행동하며, 적군 에이전트와의 전투 상황에서의 승률을 토대로 성능을 검증한다. 대표적으로 알려진 게임 속 협업 전투기술로는 focus fire, kiting 등이 있다. Focus fire는 아군 에이전트들이 동시에 하나의 적군만을 공격하고 해당 적군을 물리친 이후에는 다른 적군 하나만을 공격하는 형태의 기술이다. Kiting은 적군 에이전트와의 거리를 유지하면서 공격하고 이동하는 것을 반복하는 공격 형태로 가능한 한 적은 피해를 받으며 공격할 수 있는 행동 방식이다. 학습이 원활하게 이루어진다면 앞서 기술한 협업 전투기술들을 학습 과정에서 자연스럽게 습득하게 될 것이다. 아군 에이전트 간의 동종/이종 여부와 적군 에이전트 간의 동종/이종 여부, 아군과 적군 에이전트 간의 대칭성 여부, 아군과 적군 에이전트의 수 등에 따라 난이도가 달라진다. 그림 4는 SMAC 환경의 구동 예시이다. SMAC 환경 역시 게임 편집기를 활용하면, 원하는 임의의 환경을 사용자가 생성할 수 있다.

그림 4 StarCraft multi-agent challenge 캡처 화면[31]

Ⅳ. 결론

본 고에서는 최근 주목받고 있는 멀티 에이전트 강화학습 연구에 대해 크게 에이전트 간의 관계 모델링, 신뢰할당, 통신, 탐색-이용 딜레마라는 네 가지 문제를 중심으로 기술하고, 그와 관련된 주요 실험 환경에 대해 살펴보았다. 최근까지도 멀티 에

이전트 강화학습과 관련된 다양한 알고리즘이 제안되고 있지만, 여전히 해결하지 못한 문제와 한계점들이 존재한다. 그렇기 때문에 본 고와 같이 문제별 분류를 통해 각 문제의 어려움과 그에 대한 해결방식에 대한 전반적인 멀티 에이전트 강화학습의 연구동향을 살펴보는 것은 멀티 에이전트 강화학습 연구를 진행하기 위한 단계로서 상당히 유의미한 작업이 될 것이다. 또한, 실제 사회, 산업에서의 적용을 고려할 때 멀티 에이전트 강화학습은 필수적으로 연구되어야 할 중요한 분야이기 때문에 향후 인공지능 분야에서의 핵심적인 기술로써 자리 잡을 것으로 기대된다.

용어해설

MARL(Multi-Agent Reinforcement Learning) 하나의 에이전트가 아닌 다수의 에이전트에 대한 강화학습 알고리즘

CTDE(Centralized Training and Decentralized Execution) 멀티 에이전트 강화학습에서의 주요 학습 방식으로, 학습 모델 훈련 시에는 모든 에이전트의 정보를 이용하고 실행 단계에서는 각 에이전트의 정보만을 이용하는 방식

신뢰할당(Credit Assignment) 강화학습에서 학습 과정 중에 얻게 된 보상을 분배하는 문제. 멀티 에이전트 강화학습에서는 에이전트 간의 기여도를 분배하는 문제로 해석

약어 정리

COMA	COunterfactual Multi-Agent policy gradient
CSMA	Carrier Sense Multiple Access
CTDE	Centralized Training and Decentralized Execution
CTEDD	Centralized Training and Exploration with Decentralized execution via policy Distillation
DDPG	Deep Deterministic Policy Gradient
DIAL	Differentiable Inter-Agent Learning
EDTI	Exploration via Decision-Theoretic Influence
EITI	Exploration via Information-Theoretic Influence
HAMA	Hierarchical graph Attention-based Multi-agent Actor-critic
LIIR	Learning Individual Intrinsic Reward
MAAC	Multi-Actor-Attention-Critic
MARL	Multi-Agent Reinforcement Learning
MAVEN	Multi-Agent Variational Exploration
MF-RL	Mean Field Reinforcement Learning
SMAC	StarCraft Multi-Agent Challenge
VDN	Value Decomposition Networks

참고문헌

[1] V. Mnih et al., "Playing atari with deep reinforcement learning," arXiv preprint, CoRR, 2013, arXiv: 1312.5602.

[2] J. Schulman et al., "Trust Region Policy Optimization," in Proc. Int Conf. Mach. Learn. (Lille, France), Feb. 2015, pp. 1889-1897.

[3] J. Schulman et al., "Proximal policy optimization algorithms," arXiv preprint, CoRR, 2017, arXiv: 1707.06347.

[4] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," in Int. Conf. Learn. Representations, 2016.

[5] K. Zhang, Z. Yang, and T. Basar, "Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms," arXiv preprint, CoRR, 2019, arXiv: 1911.10635v1.

[6] O. Jadid and D. Hajinezhad, "A review of cooperative multi-agent deep reinforcement learning," arXiv preprint, CoRR, 2019, arXiv: 1908.03963v3.

[7] R. Lowe et al., "Multi-agent actor-critic for mixed cooperative-competitive environments," in Advances in Neural Information Processing Systems, 2017, pp. 6379-6390.

[8] Y. Yang et al., "Mean field multi-agent reinforcement learning," in Proc. Int. conf. Mach. Learn. (Stockholm, Sweden), 2018.

[9] S. Iqbal and F. Sha, "Actor-attention-critic for multi-agent reinforcement learning," in Proc. Int. Conf. Mach. Lear. (Long Beach, CA, USA), 2019, pp. 2961-2970.

[10] T. Haarnoja et al., "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in Prco. Int. Conf. Mach. Learn. (Stockholm, Sweden), 2018, pp. 1861-1870.

[11] H. Ryu, H. Shin, and J. Park, "Multi-agent actor-critic with hierarchical graph attention network," in Proc. AAAI Conf.

Artif. Intell. (New York, USA), 2020, pp. 7236-7243.

[12] J. Foerster et al., "Counterfactual multi-agent policy gradients," in Proc. AAAI Conf. Artif. Intell. 2020.

[13] P. Sunehag et al., "Value-decomposition networks for cooperative multi-agent learning based on team reward," in Proc. Int. Conf. Auto. Agent. Multi. Syst. 2018, pp. 2085-2087.

[14] T. Rashid et al., "QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning," in Proc. Int. Conf. Mach. Learn. 2018.

[15] K. Son et al., "Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning," in Proc. Int. Conf. Mach. Learn. 2019.

[16] Y. Du et al., "LIIR: Learning Individual Intrinsic Reward in Multi-Agent Reinforcement Learning," in Proc. Adv. Neural Inform. Process. Syst. 2019, pp. 4403-4414.

[17] C. V. Goldman and S. Zilberstein, "Decentralized control of cooperative systems: Categorization and complexity analysis," J. Artif. Intelli. Res. vol. 22, 2004, pp. 143-174.

[18] E. Pesce and G. Montana, "Improving coordination in small-scale multi-agent deep reinforcement learning through memory-driven communication," Mach. Learn. vol. 109, 2020, doi: 10.1007/s10994-019-05864-5.

[19] S. Q. Zhang, Q. Zhang, and J. Lin, "Efficient communication in multi-agent reinforcement learning via variance based control," in Adv. Neural Inform. Process. Syst. 2019, pp. 3235-3244.

[20] H. Mao et al., "Learning agent communication under limited bandwidth by message rruning," arXiv preprint, CoRR, Dec. 2019, Accessed: Sep. 21, 2020. [Online]. Available: http://arxiv.org/abs/1912.05304.

[21] D. Kim et al., "Learning to schedule communication in multi-agent reinforcement learning," arXiv preprint, CoRR, Feb. 2019, Accessed: Sep. 10, 2020. [Online]. Available: http://arxiv.org/abs/1902.01554.

[22] J. Foerster et al., "Learning to communicate with deep multi-agent reinforcement learning," in Adv. Neural Inform. Process. Syst. 2016, pp. 2137-2145.

[23] N. Jaques et al., "Social influence as intrinsic motivation for multi-agent deep reinforcement learning," in Proc. Int. Conf. Mach. Learn. 2019, pp. 3040-3049.

[24] K. Cao et al., "Emergent communication through negotiation," arXiv preprint, CoRR, Apr. 2018, Accessed: Sep. 09, 2020. [Online]. Available: http://arxiv.org/abs/1804.03980.

[25] T. Eccles et al., "Biases for emergent communication in multi-agent reinforcement learning," in Adv. Neural Inform. Process. Syst. 2019, pp. 13111-13121.

[26] S. Gupta, R. Hazra, and A. Dukkipati, "Networked multi-agent reinforcement learning with emergent communication," In Proc. Int. Conf. Auton. Agents and Multiagent Syst. (Auckland, New Zealand), May 2020.

[27] T. Wang et al., "Influence-based multi-agent exploration," in Proc. Int. Conf. Learn. Representations, 2020.

[28] G. Chen, "A new framework for multi-agent reinforcement learning-centralized training and exploration with decentralized execution via policy distillation," in Proc. Int. Conf. Auton. Agents Multiagent Sys. 2019.

[29] A. Mahajan et al., "Maven: Multi-agent variational exploration," in Adv. Neural Inform. Process. Syst. 2019, pp. 7613-7624.

[30] G. Brockman et al., "Openai gym," arXiv preprint, CoRR, arXiv: 1606.01540.

[31] M. Samvelyan et al., "The starcraft multi-agent challenge," arXiv preprint, CoRR, 2019, arXiv: 1902.04043.