https://doi.org/10.7840/kics.2024.49.1.12

온라인 커리큘럼 강화학습 기반 무인항공기를 활용한 재난 상황 네트워크 복구 연구

이 동 수◆, 어 제 연*, 권 민 혜°

Online Curriculum Reinforcement Learning Based UAV Training for Disaster Network Recovery

Dongsu Lee◆, Jeyeon Eo*, Minhae Kwon°

요 약

최근 무인 항공기(Unmanned Aerial Vehicle; UAV)를 적용해 무선 네트워크를 구축하는 플라잉 애드혹 네트워크(Flying Ad-hoc Network, FANET)에 관한 연구가 활발히 진행되고 있다. FANET은 2차원 공간 상에 장애물이 위치하는 경우에도 3차원 공간에 릴레이 노드를 배치하여 애드혹 네트워크 구축이 가능하다는 장점이 있어 재난 또는 위기 상황에서의 비상 네트워크로 활용할 수 있다. 본 논문은 애드혹 네트워크 구축 이후, 부분적 노드 동작 불능으로 소스 노드와 목적 노드 간의 연결이 불가할 때를 대비한 부분 복구 알고리즘을 제안한다. 이때, 중앙 시스템의 제어 없이 무인항공기가 자율적으로 다양한 상황에 의사결정이 가능하도록 심층강화학습(deep reinforcement learning) 알고리즘을 고려한다. 목표 달성 상황에서만 보상이 주어지는 지연되며 희소할 수 있는 문제를 해결하기 위해 커리큘럼 방식과 순환 신경망의 적용을 고려한다. 또한, 제안된 의사결정 모델의 경우 완벽한 상태 정보를 알지 못한 채 부분 관측 만으로 행동을 결정할 수 있도록 설계 하였다. 모의 실험을 통해 TD3(Twin Delayed Deep Deterministic Policy Gradient)로 학습된 무인 항공기가 다양한 시나리오에서 성공적으로 이동하여 FANET을 재구성할 수 있음을 확인

키워드 : 커리큘럼 학습, 심층 강화학습, 플라잉 애드혹 네트워크
Key Words : Curriculum learning, deep reinforcement learning, flying ad-hoc network

ABSTRACT

Extensive research has been conducted on the development of a flying ad-hoc network (FANET) utilizing unmanned aerial vehicles (UAVs) for the purpose of establishing a wireless network. The advantage of FANET lies in its ability to construct a network in a three-dimensional space, thereby offering a valuable solution even in situations where obstacles are confined to a two-dimensional plane. This unique characteristic of FANET enables it to be a promising solution for reconstructing networks in disaster or emergency scenarios. This paper presents a novel algorithm for reconstructing an ad-hoc network using UAVs, specifically addressing

※ 이 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원(No. 2021-0-00739, 분산/협력 AI 기반 5G+ 네트워크 데이터 분석 기능 및 제어 기술 개발)과 한국연구재단의 지원(RS-2023-00278812)을 받아 수행된 연구임
◆ First Author : Soongsil University Department of Intelligent Semiconductors, movementwater@soongsil.ac.kr, 학생회원
* Soongsil University Department of Intelligent Semiconductors, aircleaner@soongsil.ac.kr, 학생회원
° Corresponding Author : Soongsil University Department of Intelligent Semiconductors and School of Electronic Engineering, minhae@ssu.ac.kr, 종신회원
논문번호 : 202306-120-B-SE, Received June 9, 2023; Revised August 29, 2023; Accepted September 7, 2023

scenarios involving malfunctioning relay nodes that hinder the delivery of source data to the intended destination. To overcome the challenges associated with decentralized control, an autonomous decision-making solution is proposed for UAVs based on deep reinforcement learning. Furthermore, curriculum learning and recurrent neural networks are employed to tackle issues related to sparse and delayed rewards. To enhance practicality, the algorithm enables UAVs to make decisions based on partial and incomplete information about the surrounding environment. Experimental results demonstrate the effectiveness of the proposed algorithm in successfully reconstructing ad-hoc networks across diverse scenarios.

Ⅰ. 서 론

드론과 같은 무인 항공 이동 수단 기술의 발달에 따라 무선 네트워크 구축도 함께 주목을 받고 있다. 이와 같은 네트워크는 모바일 애드혹 네트워크(Mobile Ad-hoc Network; MANET)[1]의 한 종류인 FANET에 속한다[2]. FANET은 무인 항공기를 릴레이 노드로 포함하여 3차원 공간 상에서 소스 노드부터 목적 노드까지 데이터를 전송하는 무선 애드혹 네트워크이다. 이때, 애드혹 네트워크는 노드 간 송신 면적이 중첩되어 최소 2개 홉(hop)으로 데이터 송신이 가능할 때 형성 된다[3]. FANET은 무인 항공기를 노드로 사용해 장소에 구애 받지 않는다는 장점이 있으므로, 통신 기반 시설이 부재한 지역에서도 네트워크 구축이 가능하다. 따라서, 군사 상황이나 화재 및 산사태와 같은 재난으로 기반 시설이 붕괴된 상황에도 적용할 수 있다[4, 5]. 하지만, 무선 애드혹 네트워크는 이를 구성하고 있는 다수의 노드 중 하나라도 문제가 생길 경우 전체 네트워크의 연결이 불가능하다는 단점이 존재한다. 본 논문은 이러한 위기 상황에서 무인 항공기가 자율적으로 문제 노드의 위치를 찾아 대체할 수 있도록 하는 네트워크 복구 방안을 제시한다.

효율적인 네트워크 복구 방안 설계를 위해서는 자율적인 분산형 의사 결정 방식의 도입이 필요하다. FANET의 경우 노드들의 이동 가능 범위가 3차원이며 기상상태, 지형지물을 비롯한 장애물 등 고려해야 할 변수가 다양하다는 특징이 있다. 따라서, 모든 경우의 수를 포함해 네트워크 복구 방안을 설계하는 기존의 중앙 제어 방식은 네트워크를 구성하는 노드의 수가 증가하거나 고려해야 할 환경 조건이 많은 경우 매우 비효율적이거나 불가능할 수 있다. 반면, 분산형 의사 결정 방식은 각 네트워크 노드가 주변 노드와의 연결만을 고려하기 때문에 낮은 복잡도로 네트워크 복구 방안 설계가 가능하다.

자율적 의사 결정 방식 중 하나인 강화학습은 학습 주체가 환경과 직접 상호작용하며 능동적으로 데이터를 수집하며, 이를 기반으로 최적의 행동 양식을 찾는 기계학습의 일종이다[6]. 학습의 주체인 개체는 주어진 환경 내 상태 정보를 바탕으로 최대의 보상을 획득할 수 있는 최적의 행동 양식인 정책(policy) 학습을 목적으로 한다. 개체는 환경 내 존재하는 다양한 변수에 대해 시행착오를 통해 학습을 수행한다. 이러한 특징에도 불구하고, 목적을 달성하는 경우에만 보상이 주어지는 지연 보상(delayed reward) 문제의 경우 정책 학습에는 어려움이 따른다[7]. 이러한 어려움은 전체 궤적에서의 모든 행동 중 목표를 달성하는 경우 외에는 보상 신호를 통한 피드백이 이루어지지 않기 때문이다.

본 연구에서는 자율적인 분산형 의사결정 방법인 강화학습을 기반으로 무인 항공기를 학습시켜 네트워크를 복구하는 방안을 제안하고자 한다. 이때, 무인 항공기의 효율적인 정책 학습을 위해 두가지 요소를 고려한다. 먼저 목표에 달성 하기 쉬운 문제를 시작으로 달성 하기 어려운 목표 문제로 문제의 범위를 확장하며 학습하는 커리큘럼 방식을 이용한다[8]. 또한, 궤적 전체에서 보상을 받지 못한 행동에 공헌(contribution)을 부여하기 위해 시계열 정보를 고려할 수 있는 순환신경망(Recurrent Neural Network; RNN)을 이용한다. 결과적으로, 무인 항공기는 안정적인 네트워크 연결 상태 유지가 가능하면서 적은 에너지를 사용하는 이동 경로를 고려하여 정책을 학습하는 것을 목표로 한다. 특히, 연속적인 행동 값으로 무인 항공기 제어가 가능하도록 TD3[9]에 기반한 네트워크 복구 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 선행 연구를 소개한다. Ⅲ장은 부분적 관측 마르코프 결정과정 모델을 기반으로 제안한 문제 상황을 정의한다. 또한, 적용한 커리큘럼 기반의 심층 강화학습 알고리즘에 대해 설명한다. Ⅳ장과 Ⅴ장에서는 시뮬레이션을 통해 제안한 알고리즘의 성능을 보인다.

Ⅱ. 선행 연구

2.1 플라잉 애드혹 네트워크

FANET은 무인 항공기를 적용시켜 3차원의 네트워크 형성 범위를 가진 MANET의 일종이다. 무인 항공기는 최고 속도, 배터리 용량, 데이터 저장 공간, 안테나의 각도가 제한되어 있으므로 제약 조건 내에서 네트워크를 안정적으로 구축해야 한다는 한계가 존재한다[10]. 그러나 정찰, 감시, 구조와 같은 다양한 임무를 효율적으로 수행할 수 있으며[11] 무인 항공기 편대의 수 조절을 통해 FANET의 크기를 결정할 수 있다는 장점이 있다. FANET은 현재 군사, 민간 환경에 널리 적용될 수 있도록 다양한 연구가 진행되고 있다[12].

2.2 심층 강화학습

심층 강화학습은 모델 기반(model-based) 강화학습과 모델이 없는(model-free) 강화학습으로 분류된다. 모델 기반 강화학습은 주어진 환경 모델을 기반으로 학습하는 방식으로, 모델이 없는 강화학습보다 학습 소요 시간이 짧다는 장점이 있다[13]. 그러나 환경 모델을 정확히 아는 것은 대부분의 공학 문제에서 불가능하기 때문에 모델이 없는 강화학습이 더욱 널리 적용된다[14,15].

모델이 없는 강화학습은 가치 기반(value-based) 학습과 정책 기반(policy-based) 학습으로 구분될 수 있다. 가치 기반 학습 방식으로는 Q-learning이 대표적이다. 이는 주어진 상태에서 선택한 행동에 대한 가치를 계산하는 함수인 Q-함수를 통해 Q-함수의 결과값을 최대화 시키는 행동을 학습한다. 최근에는 Q-learning에 인공 신경망을 결합한 DQN (Deep Q-learning Network)이 다수 적용되고 있다[16].

DQN은 상태 공간이 큰 문제에 적용 가능하다는 장점이 있으나, Q-함수 근사화 과정에서 과대 추정 문제가 발생한다는 단점이 있다. 이를 보완한 알고리즘이 DDQN (Double DQN)으로, DDQN은 2개의 인공 신경망의 결과 중 작은 값을 선택하여 과대 추정 문제를 방지한다[17]. 그러나 Q-함수의 변화에 따라 정책이 크게 변동될 수 있으며 이산적인 행동 공간을 가진 문제에만 적용 가능하다는 한계를 가진다.

정책 기반 학습은 정책을 파라미터화 하여 직접적으로 정책을 근사해 학습하는 방식이다. 대표적인 알고리즘으로는 REINFORCEMENT 알고리즘이 있다[6]. 정책 기반 학습은 정책의 분산이 크고, 많은 경우 국소 최적(local optimum)에 수렴된다는 단점이 존재한다.

가치 기반 학습과 정책 기반 학습을 절충하고자 제안된 액터-크리틱(actor-critic) 방식은 정책을 근사하는 액터(actor) 네트워크와 가치 함수를 근사하는 크리틱(critic) 네트워크를 사용한다. 액터 네트워크는 정책을 결정하며, 크리틱 네트워크는 액터가 선택한 행동을 가치 함수를 기반으로 평가한다. 따라서 액터-크리틱 알고리즘은 두 개의 네트워크를 동시에 학습시키게 된다. 액터-크리틱 방식은 정책 업데이트와 정책 평가가 각기 다른 네트워크에서 진행되므로 안정적인 학습이 가능하다는 장점을 가진다. 대표적인 알고리즘으로는 DDPG (Deep Deterministic Policy Gradient)와 TD3등이 있다[18,9].

2.3 커리큘럼 강화학습

기존의 강화학습의 경우 다양한 도메인 및 여러 업무가 결합되어 있거나 희소한 보상이 주어지는 환경에서의 정책 학습에 어려움을 겪고 있다. 이와 같은 문제를 해결하기 위해 많은 기존에는 미세 조정(fine-tuning) 등의 방식을 고려하였지만 뚜렷한 진전은 보이지 않았다. 보다 최근에는 이와 같은 작업들을 완전 초기 상태에서 배우는 것은 복잡하기 때문에 문제 달성을 위해 필요한 지식을 순차적으로 제공하는 커리큘럼 강화학습 방식이 제안되었다[8]. 커리큘럼 강화학습의 핵심은 업무를 세분화하여 달성하기 쉬운 업무부터 개체가 학습할 수 있도록 인간 지도자가 커리큘럼을 정하는 것에 있다. 구체적으로, 다중 개체 환경의 경우 학습의 증가에 따라 개체의 수를 증가시켜 문제를 해결하는 방법[19], 복잡한 공간의 탈출을 위해 주기적으로 하위 목표를 설정하는 방법[20], 지형 지물과 같은 공간 구조를 점진적으로 복잡하게 설정하는 방법[21,22] 등이 포함될 수 있다. 본 연구는 개체가 정확한 손실 지점을 찾은 후 해당 지점에 도달해야만 보상을 얻을 수 있다. 이와 같은 희소 보상 문제를 해결하기 위해 개체의 행동 반경을 커리큘럼 방식을 통해 점차 확장하여 효율적으로 문제를 해결하고자 한다.

2.4 강화학습 기반 플라잉 애드혹 네트워크 구축

FANET을 구성하는 기기들은 3차원의 이동성을 가지고 있어 네트워크의 연결성 유지가 어렵다. 또한, 각 기기는 일반적으로 배터리를 기반으로 동작하므로 효율적인 에너지 사용이 필수적이다. 이에 따라 네트워크를 안정적으로 유지하는 동시에 에너지를 효율적으로 사용하기 위한 Q-learning 기반 모바일 애드혹 네트워크 구축 알고리즘이 제시되었다[23]. 해당 연구에서는 랜덤 이동 모델을 따라 움직이는 모바일 노드가 최소한의 홉 개수로 네트워크를 유지하는 방향으

로 송신 전력을 학습함을 보였다. 또한, 한정된 전력량 문제를 해결하고자 에너지 하베스트 노드(energy harvest node)를 가정한 모델 기반 강화학습 알고리즘도 제시되었다[24]. 이는 인접 노드와의 거리와 인접 노드의 배터리 상태를 상태 정보로 정의해 축적한 에너지, 현재 배터리 수준을 기반으로 에너지 효율적인 네트워크를 구축한다.

심층 강화학습을 기반으로 FANET의 QoS (Quality of Service)를 만족시키기 위해 무인 항공기의 위치를 결정하고 자원을 할당하는 연구가 진행되었다[25]. 해당 연구는 지연시간과 에너지 소모 두 가지의 QoS를 모두 만족시키는 FANET을 구축하기 위해 두 개의 별도 정책을 학습하였다. 또한, 심층 순환 그래프 신경망(deep recurrent graph network)을 이용해 무인 항공기 편대의 위치 결정 전략에 관한 연구도 이루어졌다[26]. 해당 연구는 각 무인 항공기가 공평하게 통신 지원 범위를 나누기 위한 목적으로 진행되었으나, 모든 무인 항공기의 높이가 고정되어 있어 FANET의 3차원적 특징을 고려하지 못하였다는 한계를 가진다. 본 논문에서는 무인 항공기의 고도 역시 조정할 수 있도록 함으로써 보다 실제적인 시나리오에 적용 가능한 알고리즘을 제안한다.

Ⅲ. 커리큘럼 강화학습 기반 애드혹 네트워크 부분 복구 알고리즘

3.1 고려하는 FANET 환경

본 연구에서는 기존에 구축된 무선 애드혹 네트워크의 일부 노드가 동작 불능 상태가 된 상황을 고려한다. 이때, 문제 노드의 정확한 위치 파악은 불가하며, 문제 지역으로의 접근이 어려운 경우를 가정한다. 이와 같은 문제 상황에서 무인 항공기를 활용하여 문제 노드를 대체 함으로써 네트워크의 부분 복구를 목적으로 한다. 고려하는 FANET 환경에 대한 예시 이미지는 그림 1을 통해 확인할 수 있다. FANET 환경은 $N \times N \times N$ 크기의 3차원 환경이다.

FANET을 구성하는 노드는 집합 $I \in \{s, d, a, r\}$을 통해 나타낼 수 있다. 집합에 속한 요소는 각각 정보를 수집하는 소스 노드 e_s, 데이터 센터로 데이터를 전달하는 목적 노드 e_d, 소스 노드와 목적 노드 사이의 연결을 담당하는 M개의 보조 노드 $e_{a_1}, \cdots, e_{a_M}$, 그리고 학습의 주체인 릴레이 노드 e_r을 의미한다. 여기서 보조 노드는 재구축 이전부터 FANET을 구축하고 있던 기존 노드이며, 동작 불능이 된 보조 노드는 전체 보조 노드 중 하나인 단일 개체 강화학습 문제를 고려한다. 이와 같은 상황에서 릴레이 노드 e_r은 변화하는 고장 노드의 위치와 해당 위치에서 필요로 하는 데이터 송신 범위 반지름을 학습하여 애드혹 네트워크를 재구축한다.

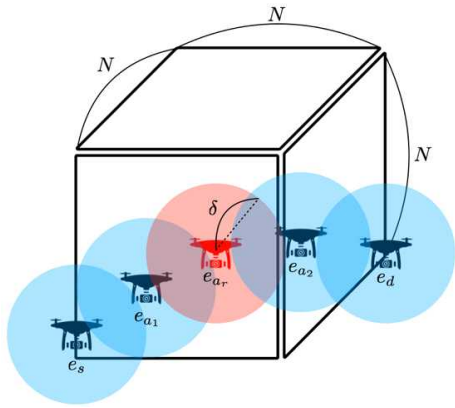

그림 1. 문제 상황 예시
Fig. 1. Illustrative diagram of problem formulation

본 논문에서는 소스 노드와 목적 노드가 충분히 멀리 떨어져 있어 직접적인 데이터 전송이 불가한 상황을 가정한다. 릴레이 노드는 전체 네트워크 내에서 이동이 가능하며, 데이터 송신 범위 반지름 및 관측 가능 거리 δ는 고정된다. 시간 t에서 소스 및 목적 노드는 위치가 각각 (x^s, y^s, z^s) 및 (x^d, y^d, z^d)로 고정되어 있다.

보조 노드와 릴레이 노드는 이중 통신(full duplex)을 가정하며 데이터를 동시에 송수신할 수 있다. 노드 e_i와 인접 노드 e_j 사이의 (x, y, z) 축 별 거리는 각각 $l_{x,t}(i,j), l_{y,t}(i,j)\ l_{z,t}(i,j)$로 정의하며, $l_{axis,t}(i,j) = |axis_t^i - axis_t^j\ |_{axis\in\{x,y,z\}}$이다. 송신 및 관찰 가능 반지름 δ 내에 노드 e_j가 존재하는 경우 $(l_{x,t}(i,j)^2 + l_{y,t}(i,j)^2 + l_{z,t}(i,j)^2)^{1/2} \leq \delta$을 만족한다. 이와 같은 상황에서 노드 e_i는 노드 e_j에 인접하다고 정의하며, 노드 e_i는 노드 e_j에게 데이터 송신이 가능하다. 무인 항공기의 한정된 메모리 크기를 반영하여 릴레이 노드가 관찰 및 통신을 유지할 수 있는 최대 인접 노드의 수는 N_{obs}로 한정한다.

3.2 부분적 관측 마르코프 의사 결정 모델

정의된 문제를 강화학습을 통해 해결하기 위해 FANET을 구성하는 릴레이 노드의 의사결정과정을 부분적 관측 마르코프 결정 과정(Partially Observable Markov Decision Process; POMDP)으로 모델링할 수 있다. 학습 주체인 릴레이 노드는 개체(agent)로서

정의된다. 개체는 환경에 관한 모든 정보를 습득하는 것은 현실적으로 불가능하다. 따라서, 개체는 전체 상태 정보 중 부분적 관측 정보를 기반으로 학습을 수행한다. 개체는 환경과 상호작용하며 관찰 정보에 대한 행동을 수행하고, 이에 따른 보상 신호를 기반으로 누적 미래 보상을 최대로 하는 정책을 찾는 것을 목표로 한다.

POMDP는 튜플 <S, O, A, Ω, T, R, γ> 로 표현 가능하다. S는 유한한 상태 집합(finite state space), O는 유한한 관찰 집합(finite observation space), A는 유한한 행동 집합(finite action space), $T(s_{t+1}|s_t, a_t)$는 상태전이 확률(state transition probability), $\Omega(o_t|s_t)$는 관측전이 확률(observation transition probability), R은 주어진 상태에서 행동을 선택했을 때의 보상(reward), γ는 미래 보상 가치에 대한 감가율(discount factor)을 의미한다.

3.2.1 상태 정보(state)

상태 $s_t \in S$는 시간 t에서의 네트워크 내 모든 정보를 의미하며, 아래와 같이 정의한다.

$$\boldsymbol{s}_t = [\ \boldsymbol{I}, \boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{z}_t\ ].$$

s_t는 시간 t에서 노드 e_i의 역할 $I \in \{s, d, a, r\}$, 노드 e_i의 3차원 공간에서의 위치로 구성된 행렬이다. 여기서, x_t, y_t, z_t는 시간 t에서의 모든 노드의 위치 정보를 의미한다.

$$\begin{aligned}\boldsymbol{x}_t &= [x^s, x^d, x_t^r, x^{a_1}, \cdots, x^{a_M}]^T \\ \boldsymbol{y}_t &= [y^s, y^d, y_t^r, y^{a_1}, \cdots, y^{a_M}]^T \\ \boldsymbol{z}_t &= [z^s, z^d, z_t^r, z^{a_1}, \cdots, z^{a_M}]^T\end{aligned}$$

3.2.2 관측 정보(observation)

관측 정보 $o_t \in O$는 시간 t에서 릴레이 노드와 인접 노드 e_j 사이의 상대 거리 $l_{x,t}(r, j)$, $l_{y,t}(r, j)$, $l_{z,t}(r, j)$와 인접 노드의 종류 $I \in \{s, d, a\}$, 릴레이 노드 자신의 위치 (x_t^r, y_t^r, z_t^r)로 정의한다. 이때, 릴레이 노드가 관찰할 수 있는 인접 노드의 수는 최대 N_{obs}이므로, 관측 정보의 크기는 $o_t \in \mathbb{R}^{3(N_{obs}+1)}$이다. o_t는 식 (1)에 나타낸다.

$$\boldsymbol{o}_t = \begin{bmatrix} l_{x,t}(r, I_1) & \cdots & l_{x,t}(r, I_{N_{obs}}) & x_t^r \\ l_{y,t}(r, I_1) & \cdots & l_{y,t}(r, I_{N_{obs}}) & y_t^r \\ l_{z,t}(r, I_1) & \cdots & l_{z,t}(r, I_{N_{obs}}) & z_t^r \\ I_1 & \cdots & I_{N_{obs}} & \delta_t^r \end{bmatrix} \quad (1)$$

3.2.3 행동(action)

행동 $\boldsymbol{a}_t \in A$은 릴레이 노드의 이동 거리 벡터 값 $(\Delta x_t^r, \Delta y_t^r, \Delta z_t^r)$으로 정의한다.

$$\boldsymbol{a}_t = \left[\Delta x_t^r, \Delta y_t^r{}_t, \Delta z_t^r\right]^T.$$

이때 각 원소는 행동의 최소값인 A_{min}과 행동의 최대값인 A_{max} 사이의 값을 가진다.

$$\left(A_{\min} \le \Delta x_t^r, \quad \Delta y_t^r{}_t, \Delta z_t^r \le A_{\max}\ \right)$$

3.2.4 보상 함수(reward function)

보상 함수 $R(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})$는 개체의 학습 목표에 따른 보상항과 처벌항의 선형 결합으로 다음과 같이 정의한다.

$$R(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}) = \eta_1 \mathcal{R}_1(\boldsymbol{s}_{t+1}) + \eta_2 \mathcal{R}_2(\boldsymbol{a}_t)$$

여기서 η_n은 각 항의 가중치를 결정하며, $\mathcal{R}_n$은 n번째 보상항을 의미한다.

먼저, $\mathcal{R}_1$은 행동을 수행한 다음 시점에 파괴된 네트워크가 복구되었는지 여부에 따라 다음과 같이 주어진다.

$$\mathcal{R}_1(\boldsymbol{s}_{t+1}) = \begin{cases} 1, & \text{network is recovered} \\ 0, & \text{otherwise} \end{cases}$$

즉, 네트워크가 정상적으로 복구되는 경우 보상이 주어지며, 그렇지 않은 경우에는 해당 항은 작동하지 않는다.

이어서, $\mathcal{R}_2$는 개체의 움직임에 따른 처벌을 부여하는 항이다. 해당 항은, 최적의 이동 경로를 통해 네트워크를 복구한 다음 복구 위치에서 움직임을 최소화한 상태로 네트워크를 유지할 수 있도록 유도한다. 해당 식은 다음과 같이 정의한다.

$$\mathcal{R}_2(\boldsymbol{a}_t) = \sqrt{\Delta x_t^r + \Delta y_t^r + \Delta z_t^r}$$

3.3 TD3 기반 POMDP 솔루션

본 연구에서는 연속적인 상태 및 행동 공간에서의 정책 및 가치함수 최적화를 위해 액터 크리틱(actor-critic) 기반의 결정론적 알고리즘인 TD3를 적용한다.

TD3 알고리즘의 경우 2개의 크리틱 네트워크로 Q

함수 값을 계산한 다음, 둘 중 작은 값을 네트워크 업데이트에 이용하는 clipped double Q-learning 방식이 적용된다. 해당 기법을 통해 가치 함수 최적화에서 문제가 되었던 기법을 통해 가치 함수 최적화에서 문제가 되었던 과대추정(overestimation) 문제를 완화한다. 이를 기반으로 두개의 크리틱 네트워크 목적함수는 다음과 같이 정의된다.

$$L(\theta_i) = \mathbb{E}\left[\left(Q_{\theta_i}(\boldsymbol{o}_t, \boldsymbol{a}_t) - y_t\right)^2\right] \quad (2)$$

여기서 y_t는 temporal difference (TD) target으로 다음과 같이 정의된다.

$$y_t = r_t + \gamma \min_{i=1,2} Q_{\theta_i'}(\boldsymbol{o}_{t+1}, \tilde{\boldsymbol{a}})$$

여기서 $\tilde{\boldsymbol{a}}$는 정책에서 샘플링된 행동 $\boldsymbol{a}_{t+1}$~$\pi_\phi(\boldsymbol{o}_{t+1})$에 smoothing noise ϵ~$clip(\mathcal{N}(0, \tilde{\sigma}), -c, c)$를 더해준 값으로 상수 c를 최대 및 최소로 고려하여 정의된다. 이어서, 정책을 근사하는 액터 네트워크는 다음과 같은 목적함수를 갖는다.

$$L(\phi) = \mathbb{E}\left[Q_{\theta_1}\left(\boldsymbol{o}_t, \pi_\phi(\boldsymbol{o}_t)\right)\right] \quad (3)$$

3.4 커리큘럼 기반 FANET 복구 알고리즘

본 논문에서 해결하고자 하는 문제에서 개체는 네트워크를 복구하지 않는 경우 긍정적인 보상 신호를 획득하지 못한다. 학습의 핵심이 될 수 있는 보상 신호의 희소성은 정책 및 가치 함수의 최적화 문제를 어렵게 만든다. 게다가, 우연하게 네트워크를 복구하는 궤적을 경험하였다고 하더라도 목표에 달성한 순간을 제외한 관측-행동 쌍에 대한 평가는 제대로 수행되지 않는다는 문제점이 있다. 이와 같은 문제를 해결하기 위해 작업의 난이도를 낮은 수준에서 높은 수준으로 확장하는 커리큘럼 방식을 고려(그림 2)하며, 성공적인 목표 달성을 수행한 궤적 전체에 보상 정보를 간접적으로 할당하기 위해 시계열 정보를 고려할 수 있는 순환 신경망 구조를 고려한다.

그림 2. 커리큘럼 학습에서의 문제 영역 확장
Fig. 2. Expansion of task domain in curriculum learning

먼저 알고리즘 1은 학습 과정이 진행됨에 따라 개체의 학습을 위한 커리큘럼 방식에 대한 정보를 포함한다. 총 L개의 커리큘럼이 존재할 때 최대 보상을 받을 수 있는 최소 환경 범위인 $N_r \times N_r \times N_r$부터 최대 환경 범위인 $N \times N \times N$까지 순차적으로 확장하게 된다. 이때 각 환경에서의 학습 기간은 전체 학습 기간인 K를 균등하게 L로 나누어 고려한다.

이어서 알고리즘 2는 TD3 알고리즘을 기반으로 애드혹 네트워크의 부분적 복구를 위해 릴레이 노드의 학습 과정을 포함한다. 전체 학습 기간 K와 단일 에피소드를 구성하는 timestep T를 고려할 때 개체가 경험할 수 있는 전체 에피소드의 수를 계산할 수 있다. 매 스텝마다 릴레이 노드는 환경 상태를 관찰하고 관찰한 정보를 바탕으로 행동을 결정한다. 이때, 개체는 관찰 가능한 최대 주변 노드의 수 N_{obs}의 크기만큼 정보를 습득하게 된다. 시간 t에서 개체는 관찰 정보를 바탕으로 데이터 송신 범위 반지름 변화량 $\Delta\delta_t^r$와 이동거리 $(\Delta x_t^r, \Delta y_t^r, \Delta z_t^r)$를 결정하며, 탐색(exploration)을 위해 노이즈 $\mathcal{N}$가 더해진다. 임계값 적용 함수는 함수의 출력값 범위를 제한하는 함수이며, ϕ'은 타겟 액터 네트워크의 파라미터이다.

$$\boldsymbol{a}_{t+1} = \pi_{\phi'}(\boldsymbol{o}_{t+1}) + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma)$$

개체는 행동과 행동에 의해 변화된 상태 정보에 따라 보상을 받는다. 보상은 네트워크 처리율과 에너지 소비량이다. 이후 현재 관찰, 행동, 보상, 다음 관찰을 튜플 형태로 재현 메모리(replay buffer)에 저장한 뒤,

Algorithm 1. Environment expansion based on curriculum learning

Algorithm 1 Curriculum Learning

Require Training iterations K, The number of curriculum level L, Reward zone size N_r, Environment size N

1: **for** Curriculum level $l = 1$ **to** L **do**
2: **for** iteration $k = 1$ **to** $\frac{K}{L}$ **do**
3: **operate** Algorithm 2
4: Set environment size $\frac{(N-N_r)}{L} \times l$
5: **end for**

Algorithm 2. TD3 based ad-hoc network reconstruction algorithm

Algorithm 2 TD3-based Ad-hoc Network Reconstruction

Require entire node set $\boldsymbol{I}$, action set A, replay buffer $\mathcal{D}$, critic network parameters θ_1, θ_2 and an actor network parameter ϕ, target network parameters $\theta'_1, \theta'_2, \phi'$, target update frequency d, soft update ratio τ, batch size N

for episode = 0, 1, …, T
  Initialize relay node location x_t^r, y_t^r, z_t^r, transmission radius δ_t^r
  Update exploration noise σ_t
  for step = 0, 1, …, t_{end}
    Select action with exploration noise $\boldsymbol{a}_t \sim \pi_\phi(\boldsymbol{o}_t) + \epsilon_{exp}, \epsilon_{exp} \sim \mathcal{N}(0,\ \sigma_t)$
    Update state $\boldsymbol{s}_{t+1}$
    $(x_{t+1}^r, y_{t+1}^r, z_{t+1}^r) \leftarrow (x_t^r, y_t^r, z_t^r) + \left(\Delta x_t^r, \Delta y_{t_t}^r, \Delta z_t^r\right)$
    $\delta_{t+1}^r \leftarrow \delta_t^r + \Delta\delta_t^r$
    Broadcast data
    Get new observation $\boldsymbol{o}_{t+1}$, receive reward r_t
    Store transition $d = (\boldsymbol{o}_t, \boldsymbol{a}_t, r_t, \boldsymbol{o}_{t+1})$ in $\mathcal{D}$
    Sample N random batch of transitions from $\mathcal{D}$
    $y_t \leftarrow r_t + \gamma \min_{i=1,2} Q_{\theta'_i}(\boldsymbol{o}_{t+1}, \boldsymbol{a}_{t+1})$
    Update critics θ_i by eqn. (2)
    If t mod d **then**
      Update policy ϕ by eqn.(3)
      Update target networks:
      $\theta'_i \leftarrow \tau\theta_i + (1-\tau)\theta'_i$
      $\phi'_i \leftarrow \tau\phi_i + (1-\tau)\phi'_i$
    end for
end for

N개의 튜플을 랜덤하게 샘플링해 최적 정책을 찾기 위한 정책 네트워크의 파라미터 업데이트에 적용한다.

Ⅳ. 모의실험 결과

4.1 모의실험 설정

본 논문에서 제안한 알고리즘의 성능을 검증하고자 두 가지 시나리오를 고려한다. 모든 시나리오에서 보조 노드의 수 M= 2이며, 총 3개의 노드가 연결이 되어야 전체 네트워크를 복구할 수 있는 상황을 가정한다. 첫번째 시나리오는 네트워크를 복구를 위한 목표 지점이 매번 동일한 문제이며, 두번째 시나리오는 네트워크 복구를 위한 목표 지점이 3가지 중 하나로 랜덤하게 선택되는 문제이다. 네트워크의 크기 N= 10m로, 즉 3차원 공간의 크기는 10m × 10m × 10m로 정의된다. 이때 릴레이 노드의 비행상황을 고려하기 위해 2m 이하로의 비행은 불가능하다고 가정하여 z축의 이동 범위는 [2m, 10m]로 정의한다. 개체는 한번에 각 축을 기준으로 A= [-1m, 1m]의 범위에서 움직일 수 있으며, 소스 노드의 위치는 (0m, 0m, 0m), 목적 노드의 위치는 (10m, 10m, 0m)로 정의한다. 모든 노드의 관측가능 및 데이터 송신 가능 범위 δ = 5.5m로 고정되어 있다. 또한, 소스 및 목적 노드의 경우 위치가 고정되며, 보조 노드의 경우 각축으로 1m의 행동반경을 갖고 움직이는 경우와 고정된 경우에 대해 설정 가능하다. 시나리오 1과 시나리오 2에 대한 각 노드의 초기 배치와 관련된 설정은 표 1을 통해 확인할 수 있다.

모든 실험의 학습 iteration 수 K= 3,000,000으로 고려하였으며, 단일 에피소드를 구성하는 timestep T = 40, 전체 커리큘럼 레벨은 L = 10으로 설정하였다. 최대 관측 가능 노드의 수 N_{obs} = 2, 감가율 γ = 0.99, 학습률(learning rate) a = 10^{-4}로 설정하였다.

본 논문에서는 ACK가 없는 브로드캐스트(broadcast) 상황에서의 최대 네트워크 처리율을 가정하였으며, 이는 50Mbps의 값으로 설정하였다[27].

4.2 비교 알고리즘

제안된 알고리즘이 효율적인 문제 해결을 수행하는지 객관적으로 확인하기 위해 본 논문에서는 총 4가지의 알고리즘과 비교를 수행한다. 비교를 위해 선택된 알고리즘은 DDPG-C[28], DDPG-OS[28], DDPG-COS[28], 그리고 TD3 이다. 네 가지 알고리즘 모두 대표적인 온라인 강화학습 알고리즘 중 하나인

표 1. 시나리오 별 노드 위치 설정
Table 1. Initial position of nodes per scenario

	Target Position	e_{a_1}	e_{a_2}	e_s	e_d
Scenario1	[5.95, 5.05, 3.3]	[3.0, 2.0, 2.7]	[8.9, 8.1, 3.9]	[0, 0, 0]	[10, 10, 10]
Scenario2	[5.95, 5.05, 3.3]	[3.0, 2.0, 2.7]	[8.9, 8.1, 3.9]	[0, 0, 0]	[10, 10, 10]
	[8.9, 8.1, 3.9]	[5.95, 5.05, 3.3]	[3.0, 2.0, 2.7]		
	[3.0, 2.0, 2.7]	[5.95, 5.05, 3.3]	[8.9, 8.1, 3.9]		

DDPG를 기반으로 한다는 특징을 갖고 있다. 먼저 DDPG-C는 actor 네트워크와 critic 네트워크 사이의 호환성 문제를 해결하기 위해 critic network의 목적함수 계산 $L(\theta) = \frac{1}{N}\sum_{i=1}^{N} rate_i\,(y_i - Q)^2$을 위해 정책 네트워크의 샘플 호환성 $\zeta = \exp\left(a - \pi_\phi(s)\right) + \exp\left(\pi_\phi(s) - \pi_{\phi'}(s)\right)$에 따라 업데이트를 진행하며, 해당 계수를 기반으로 샘플 별 업데이트 비율을 다음과 같이 고려한다.

$$rate_i = clip\left(\frac{\zeta_i}{\max G_t \sum_{n=1}^{N}\zeta_n}, 1-c, 1+c\right)$$

이어서 DDPG-OS의 경우 critic 네트워크의 과대추정 문제 해결을 고려하기 위해 TD-target 값을 clipping 하는 방법이다. 이때, TD-target의 최대 값은 가능한 최대 누적 보상 Max G_T이며, 최소 값은 음의 무한으로써 $y = clip(y, -\infty, \max G_t)$로 설정된다. DDPG-COS는 DDPG-C와 DDPG-OS의 방식을 모두 고려한 알고리즘이며, TD3는 본 논문의 3.3 절에서 자세하게 확인할 수 있다.

4.3 성능 검증

4.3.1 Scenario 1에서의 알고리즘 성능 분석

그림 3**A-B**는 Scenario 1에서 학습이 진행됨에 따라 보이는 hit ratio를 나타내며, 그림 3**A**의 경우 보조 노드가 고정된 상황을 그림 3**B**는 보조 노드가 움직일 수 있는 상황에서 진행된 실험이다. 각각의 실선은 평균값을 의미하며 음영은 2 표준편차 기반의 신뢰구간을 의미한다. 각 에피소드는 40 timestep으로 구성되어 있으며 hit ratio가 1에 가까울수록 높은 복구 및 유지 성능을 나타낸다. 그림 3A-B에서 확인할 수 있듯 단일 시나리오에서 가장 높은 성능을 보장하는 알고리즘은 본 논문에서 제안하는 LSTM 및 커리큘럼 학습 방법을 채택한 제안 방법이다. 그림 3A에서는 다른 4가지 알고리즘의 성능이 거의 비슷한 반면, 그림 3B에서는 TD3가 다른 DDPG기반의 알고리즘과 비교하여 우세한 성능을 보인다. 또한, 보조 노드의 이동성을 고려하는 경우 전반적으로 분산의 크기가 증가하는 모습을 확인할 수 있다. 이러한 결과는 모든 에피소드에서 복구를 위해 이동해야 할 목표 위치가 동일 하기 때문에 주변 상황이 아닌 해당 위치 값에 과적합 될 수 있기 때문이다. 즉, 해당 시나리오의 경우 개체가 [5.95, 5.05, 3.3] 위치로의 이동을 목표로 쉽게 문제를 해결할 수 있다.

여기서 모든 커리큘럼 학습 방식의 성능 곡선은 우하향 하는 형태를 나타낸다. 이는 학습 기간이 증가함에 따라 개체가 경험 가능한 환경의 범위가 확장되기 때문이다. 즉, 개체가 보상을 얻기 위한 목표 지점까지의 최소 스텝수가 증가하며, 새로운 공간에서의 경험을 기반으로 정책을 추가 학습하는 과정이 포함되기 때문이다.

4.3.2 Scenario 2에서의 알고리즘 성능 분석

그림 3A-B 는 Scenario 2에서 학습이 진행됨에 따라 보이는 hit ratio를 나타내며, 각 그림은 보조 노드가 고정된 상황 및 보조 노드가 움직일 수 있는 상황에서 진행된 실험을 의미한다. Scenario 2의 경우 Scenario 1과 달리 복구를 위한 목표 위치가 매 에피

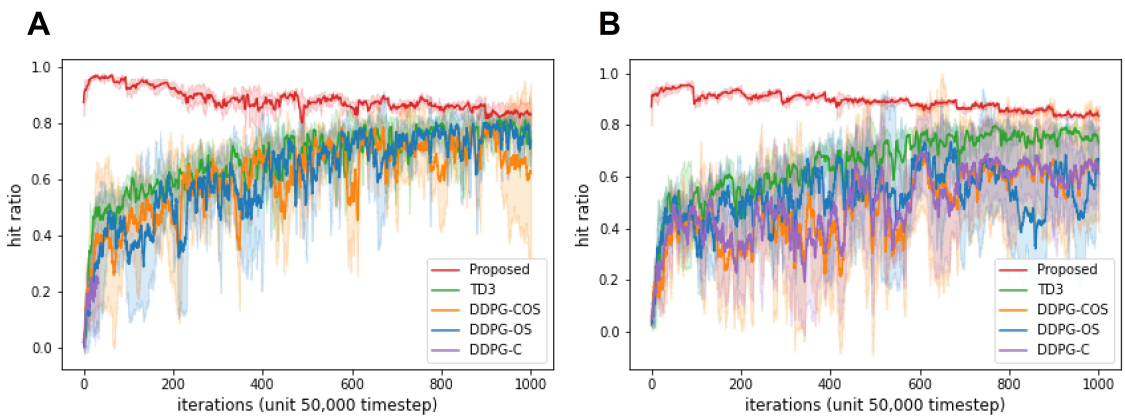

그림 3. Scenario 1에서 학습 진행에 따른 hit ratio. A: 보조 노드 고정. B: 보조 노드 이동 가능.
Fig. 3. Hit ratio over training time in Scenario 1. A: Assistance node is fixed. B: Assistant node is movable.

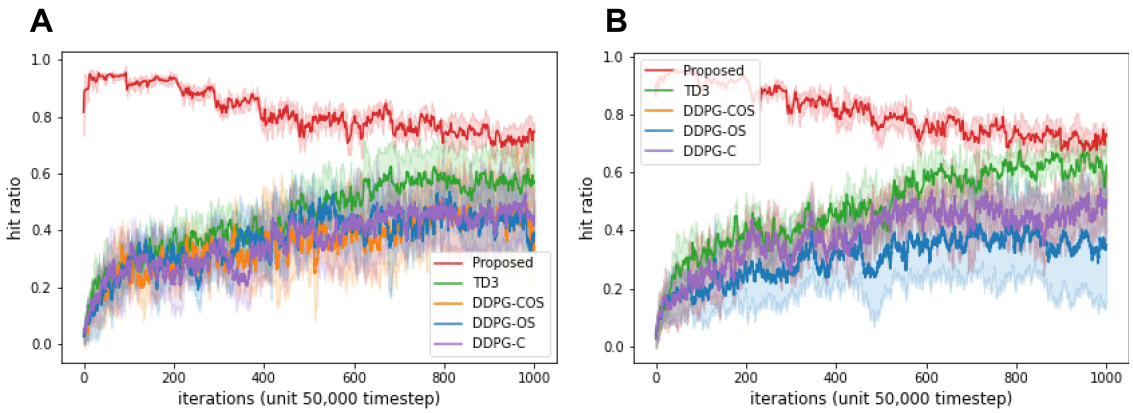

그림 4. Scenario 2에서 학습 진행에 따른 hit ratio. A: 보조 노드 고정. B: 보조 노드 이동 가능.
Fig. 4. Hit ratio over training time in Scenario 2. A: Assistance node is fixed. B: Assistant node is movable.

소드마다 무작위로 선택되기 때문에 특징 위치로의 최적화 방식을 통해서는 문제를 해결할 수 없다. 해당 시나리오에서 가장 높은 성능을 보장하는 알고리즘은 예상과 같이 LSTM 및 커리큘럼 방식이 고려된 본 연구에서 제안된 방식이다. 반면 다른 방식들의 경우 Scenario 1과 비교하여 상대적으로 낮은 성능을 내는 것을 확인할 수 있다. 또한, curriculum과 비교하여 더욱 높은 분산을 갖는 것을 확인할 수 있다. 이러한 양상은 특히 보조 노드가 이동가능한 4B에서 확인 가능하다. 또한, Scenario 1에서도 확인하였던 결과와 유사하게 보조 노드의 이동성이 없는 경우보다 이동성이 존재하는 경우에 성능이 보다 낮으며 신뢰 구간이 큰 것을 확인할 수 있다. 실험 결과를 통해 커리큘럼 방식은 보다 효율적이며 안정적으로 네트워크 복구를 위한 UAV의 학습을 수행 가능한 점을 보일 수 있다.

4.4 중앙 제어 방식과의 계산 복잡도 비교

중앙 제어 방식은 전통적으로 네트워크 내 모든 요소를 제어하기 위해 사용되어 왔으나, 네트워크 구성 요소의 수가 기하급수적으로 증가함에 따라 계산 복잡도의 한계에 다다르고 있다. 제안한 네트워크 부분 복구 알고리즘을 중앙 제어 방식으로 구현할 경우, 전체 네트워크인 환경에 대한 전수 조사가 필요하다. 중앙 제어 방식을 고려할 시 계산 복잡도는 탐색 공간에 비례하며, 이는 네트워크의 크기와 행동 공간의 곱으로 정의할 수 있다.

제안한 알고리즘은 중앙 제어 방식이 아닌 자율 제어 방식인 강화학습을 사용해 계산 복잡도가 인공 신경망의 계산 복잡도와 스텝 수에 비례하며, 이는 표 2에 나타냈다. 관측한 정보의 크기 $x = |o_t|$를 입력으로 행동 벡터 요소의 수 $y = |a_t|$가 출력이라고 할 때, 첫번째 레이어 W_1의 경우 LSTM 을 적용하는 경우 계산 복잡도는 $24x^2 + 24x$이며, MLP를 고려하는 경우 $2x^2$이다. 두번째 레이어 W_2의 계산 복잡도는 $8x^2$, 출력 레이어 W_{out}의 계산 복잡도는 $4xy$이다.

이어서 전체 환경의 크기가 $N \times N \times N$이고, 단위 길이 1을 q개로 양자화하면 중앙 제어 방식은 $N^3 \times q^3$개의 양자화된 영역에 대한 행동 가능성에 대해 검토를 수행해야 한다. 여기서 단일 영역 별 가능한 모든 행동의 수인 $|A|$가 고려되어, 최종적으로 계산복잡도는 $N^3 \times q^3 \times |A|$가 된다. 즉, 양자화 단계가 많아질 수록 계산 복잡도가 제곱꼴로 커지는 것을 확인할 수 있다. 반면, 제안한 알고리즘의 계산 복잡도는 스텝 수 t_{end}와 인공 신경망의 계산 복잡도가 고려된 $t_{end} \times 32x^2$

표 2. 인공 신경망 레이어 별 계산 복잡도
Table 2. Computational complexity of neural network layers

	Matrix size	Complexity
The 1st network layer	$LSTM/MLP$	$24x^2 + 24x/2x^2$
The 2nd network layer	$W_2 \in \mathbb{R}^{4x \times 2x}$	$8x^2$
The output layer	$W_{out} \in \mathbb{R}^{y \times 4x}$	$4xy$

표 3. 커리큘럼 학습에서의 문제 영역 확장
Table 3. Computational complexity comparison

	Centralized solution	Proposed solution
Complexity	$L_1 \times L_2 \times q^2 \times \|A\|$	$t_{end} \times 4xy$

$+ 4x(6 + 4y)$로, 환경의 크기와 무관한 값을 가진다. 해당 계산 복잡도는 표 3을 통해 확인할 수 있다.

V. 결 론

본 논문은 애드혹 네트워크의 연결이 일부 노드의 동작 불능으로 끊어졌을 때 무인 항공기를 이용하여 복구하는 방법에 대해 고려한다. 제안된 커리큘럼 방식 기반의 심층강화학습으로 학습된 무인항공기는 완벽한 상태정보가 아닌 부분 관측 정보만을 이용하여 3차원 공간에서의 이동을 통해 네트워크를 복구 및 유지할 수 있다. 모의 실험을 통해 주어진 시나리오에서 제안된 방식이 성공적으로 네트워크를 재구성 및 유지할 수 있음을 확인하였다.

References

[1] K. Sohrabi, et al., “Protocols for self-organization of a wireless sensor network,” *IEEE Pers. Commun.*, vol. 7, no. 5, pp. 16-27, 2000. (https://doi.org/10.1109/98.878532)

[2] İ. Bekmezci, et al., “Flying ad-hoc networks (FANETs): A survey,” *Ad Hoc Networks*, vol. 11, no. 3, pp. 1254-1270, 2013. (https://doi.org/10.1016/j.adhoc.2012.12.004)

[3] D. Yu, et al., “On the definition of ad hoc network connectivity,” *ICCT*, vol. 2, pp. 990-994, 2003. (https://doi.org/10.1109/ICCT.2003.1209696)

[4] M. Deruyck, et al., “Emergency ad-hoc networks by using drone mounted base stations for a disaster scenario,” *IEEE WiMob*, pp. 1-7, 2016. (https://doi.org/10.1109/WiMOB.2016.7763173)

[5] T. Plesse, et al., “OLSR performance measurement in a military mobile ad hoc network,” *Ad Hoc Networks*, vol. 3, no. 5, pp. 575-588, 2005. (https://doi.org/10.1016/j.adhoc.2004.08.005)

[6] R. S. Sutton, et al., “*Reinforcement Learning: An Introduction*,” 2nd Ed., MIT Press, pp. 1-22, 2018. (https://doi.org/10.1109/TNN.1998.712192)

[7] A. Ecoffet, et al., “First return, then explore,” *Nature*, vol. 590, no. 7847, pp. 580-586, 2021. (https://doi.org/10.1038/s41586-020-03157-9)

[8] S. Narvekar, et al., “Curriculum learning for reinforcement learning domains: A framework and survey,” *JMLR*, 2020.

[9] S. Fujimoto, et al., “Addressing function approximation error in actor-critic methods,” *ICML*, 2018.

[10] A. Srivastava, et al., “Future FANET with application and enabling techniques: Anatomization and sustainability issues,” *Comput. Sci. Rev.*, vol. 39, pp. 100359, 2021. (https://doi.org/10.1016/j.cosrev.2020.100359)

[11] O. K. Sahingoz, “Networking models in flying ad-hoc networks (FANETs): Concepts and challenges,” *J. Intell. & Robotic Syst.,* vol. 74, no. 1, pp. 513-527, 2014. (https://doi.org/10.1007/s10846-013-9959-7)

[12] A. Chriki, et al., “FANET: Communication, mobility models and security issues,” *Computer networks*, vol. 163, 2019. (https://doi.org/10.1016/j.comnet.2019.106877)

[13] C. G. Atkeson, et al., “A comparison of direct and model-based reinforcement learning,” *ICRA*, 1997. (https://doi.org/10.1109/ROBOT.1997.606886)

[14] T. Degris, et al., “Model-Free reinforcement learning with continuous action in practice,” *ACC*, 2012. (https://doi.org/10.1109/ACC.2012.6315022)

[15] D. Yarats, et al., “Improving sample efficiency in model-free reinforcement learning from images,” *AAAI*, 2021. (https://doi.org/10.1609/aaai.v35i12.17276)

[16] V. Mnih, et al., “Playing atari with deep reinforcement learning,” *NIPS*, 2013.

[17] H. van Hasselt, et al., “Deep reinforcement learning with double Q-learning,” *AAAI*, 2016. (https://doi.org/10.1609/aaai.v30i1.10295)

[18] T. P. Lillicrap, et al., “Continuous control with deep reinforcement learning,” *ICLR*, 2017.

[19] Y. Song, et al., “Autonomous overtaking in gran turismo sport using curriculum reinforcement learning,” *ICRA*, pp. 9403-9409,

2021.
(https://doi.org/10.1109/ICRA48506.2021.9561049)

[20] Q. Li, et al., "Understanding the complexity gains of single-task RL with a curriculum," *ICML*, pp. 20412-20451, 2023.

[21] A. S. Azad, et al., "CLUTR: Curriculum learning via unsupervised task representation learning," *ICML*, pp. 1361-1395, 2023.

[22] J. Park., et al., "Indoor path planning for an unmanned aerial vehicle via curriculum learning," *ICCAS*, pp. 529-533, 2021.
(https://doi.org/10.23919/ICCAS52745.2021.9649794)

[23] N. Kim, M. Kwon, and H. Park, "Q-learning based ad-hoc network formation strategy for wireless nodes with random mobility models," *J. KICS*, vol. 46, no. 11, pp. 1833-1844, 2021.
(https://doi.org/10.7840/kics.2021.46.11.1834)

[24] M. Maleki, et al., "A model-based reinforcement learning algorithm for routing in energy harvesting mobile ad-hoc networks," *Wireless Pers. Commun.*, vol. 95, pp. 3119-3139, 2017.
(https://doi.org/10.1007/s11277-017-3987-8)

[25] Q. Wang, et al., "MPRdeep: Multi-objective joint optimal node positioning and resource allocation for FANETs with deep reinforcement learning," *IEEE LCN*, pp. 315-318, 2021.
(https://doi.org/10.1109/LCN52139.2021.9524972)

[26] Z. Ye et al., "Multi-UAV navigation for partially observable communication coverage by graph reinforcement learning," *IEEE Trans. Mobile Comput.*, 2022.
(https://doi.org/10.1109/TMC.2022.3146881)

[27] DJI, *DKI - FPV Drone Specification*, Retrieved Jun., 01, 2023, from https://www.dji.com/kr/dji-fpv/specs.

[28] D. Wang, et al., "Deep deterministic policy gradient with compatible critic network," *IEEE Trans. Neural Networks and Learning Systems*, vol. 34, no. 8, pp. 4332-4344, 2023.
(https://doi.org/10.1109/TNNLS.2021.3117790)

이 동 수 (Dongsu Lee)

2022년 2월 : 숭실대학교 의생명시스템학부 빅데이터컴퓨팅 융합전공 학사
2022년 3월~현재 : 숭실대학교 지능형반도체학과 석박사통합과정
<관심분야> 강화학습, 계산신경과학, 자율주행, 모바일네트워크
[ORCID:0000-0002-9238-4106]

어 제 연 (Jeyeon Eo)

2023년 2월 : 숭실대학교 컴퓨터학부 학사
<관심분야> 인공지능

권 민 혜 (Minhae Kwon)

2011년 8월 : 이화여자대학교 전자정보통신공학과 학사
2013년 8월 : 이화여자대학교 전자공학과 석사
2017년 8월 : 이화여자대학교 전자전기공학과 박사
2017년 9월~2018년 8월 : 이화여자대학교 전자전기공학과 박사 후 연구원
2018년 9월~2020년 2월 : 미국 Rice University, Electrical and Computer Engineering, Postdoctoral Researcher
2020년 3월~현재 : 숭실대학교 전자정보공학부 및 지능형반도체학과 조교수
<관심분야> 강화학습, 자율주행, 모바일네트워크, 연합학습, 계산신경과학
[ORCID:0000-0002-8807-3719]