https://doi.org/10.7840/kics.2023.48.9.1123

신뢰성 있는 지구 저궤도 위성 통신망 최적화를 위한 강화학습 기반 동적 라우팅 알고리즘

김 규 선•, 박 수 현°, 김 중 헌*, 김 영 구**, 하 재 경**, 전 병 현**

Reinforcement Learning-Based Dynamic Routing for Robust Optimization of Low Earth Orbit (LEO) Satellite Communication Networks

Gyu Seon Kim•, Soohyun Park°, Joongheon Kim*,
Yeonggoo Kim**, Jaekyoung Ha**, Byung Hyun Jun**

요 약

최근 차세대 통신시스템인 6G와 함께, 전 지구적 인터넷 공급을 가능케 하는 위성통신에 대한 많은 연구가 진행되고 있다. 특히, 상대적으로 낮은 고도에서 궤도운동을 하는 지구 저궤도 위성은 다른 위성 시스템에 비해 지구와의 거리가 가까워 통신 시스템 구성에 이점이 있다. 지구 저궤도 위성의 특성상 동일 궤도에 다수의 위성이 배치되며, 위성이 빠른 속도로 공전하기 때문에, 향상된 통신 성능을 위한 위성 간 라우팅 기술 연구가 필수적이다. 이에, 본 논문에서는 기계학습 방법 중 하나인 강화학습을 활용하여, 지구 저궤도 위성 통신망 최적화를 위한 라우팅 알고리즘을 제안한다. 우주 환경에서 발생 가능한 위성 토폴로지에 다양한 강화학습 알고리즘을 적용하고, 이를 통해 해당 알고리즘의 우수성을 평가함과 동시에 실제 우주 공간에서 위성 간 라우팅의 적용 가능성을 제시한다.

키워드 : 심층 강화학습, 지구 제궤도 위성, 라우팅
Key Words : Deep Reinforcement Learning, Low Earth Orbit (LEO), Routing

ABSTRACT

Recently, satellite communication has garnered significant attention as a novel industry capable of providing global internet access in conjunction with the next-generation communication system, 6G. Notably, low-Earth orbit satellites, operating at comparatively lower altitudes, offer an advantage in communication system configuration due to their closer proximity to Earth. The inherent characteristics of LEO satellites, such as their high orbital speed and deployment of numerous satellites in the same orbit, necessitate research into inter-satellite routing technology for enhanced communication performance. Consequently, this study presents a

※ 본 연구는 ㈜솔빛시스템의 재원으로 한국연구재단 기초연구실지원사업 (2021R1A4A1030775)의 연구비 지원으로 수행됨. 이는 ㈜솔빛시스템의 지원을 받아 수행된 연구임. (강화학습 기반 저궤도 위성 라우팅 알고리즘 연구 용역) 본 논문의 교신저자는 박수현임.
• First Author : Korea University School of Electrical Engineering, kingdom0545@korea.ac.kr, 학생회원
° Corresponding Author : Korea University School of Electrical Engineering, soohyun828@korea.ac.kr, 정회원
* Korea University School of Electrical Engineering, joongheon@korea.ac.kr, 종신회원
** Solvit System, solvit@solvitsystem.co.kr; jkha@solvitsystem.co.kr; bhjun@solvitsystem.co.kr
논문번호 : KICS202305-104-B-RN, Received May 23, 2023; Revised June 18, 2023; Accepted June 18, 2023

routing algorithm aimed at optimizing the LEO satellite communication network by employing reinforcement learning, a machine learning technique. By applying various reinforcement learning algorithms to satellite topologies that may arise in space environments, the superiority of the algorithm is assessed, and simultaneously, the feasibility of implementing inter-satellite routing in space is demonstrated.

Ⅰ. 서 론

전지구적 6G 통신 시스템 기술 개발의 핵심 요소로 여겨지는 위성 통신망은 전세계 인터넷 접근성을 향상시키기 위한 혁신적인 기술로 주목받고 있다. 특히, 지구 저궤도 위성(Low Earth Orbit, LEO)은 지구 중궤도 위성(Medium Earth Orbit, MEO), 지구 정지궤도 위성(Geostationary Orbit, GEO)과 같은 다른 위성시스템에 비해 상대적으로 가까운 고도에서 궤도운동을 하기 때문에 지연 시간이 줄어들며 이를 통해 높은 통신 효율성을 제공할 수 있다. 이러한 기술적 장점을 바탕으로 최근 Elon Reeve Musk의 SpaceX사는 전 지구 인터넷 공급을 목표로 하는 Starlink 사업을 추진하고 있다[1]. 이는 정해진 궤도에서 지구 위를 도는 위성을 통해 위성 인터넷망을 구축하는 것을 목표로 한다. 재사용 로켓인 Falcon 9 등 안정화된 발사체 기술을 갖춘 SpaceX는 현재까지 2,000개의 저궤도 위성을 운용하며, 전쟁으로 인해 통신 인프라가 파괴된 우크라이나에서도 이미 상용화되었다[2],[3]. 위와 같은 시스템은 40ms의 지연 시간 및 50Mbps의 다운링크 등 우수한 통신 성능을 자랑한다. 이처럼 위성 통신 시스템에서 뛰어난 속도를 유지하고 위성 통신망을 확장하기 위해서는 효율적인 라우팅 기술이 필수적이다. 위성 라우팅은 저궤도 위성-위성 간의 통신(Inter Satellite)이나 위성-지구와의 통신에서 어떠한 경로를 거쳐 통신할 것인지를 의미한다. 본 논문에서는 이러한 위성 라우팅 기술에 강화학습 알고리즘인 Monte Carlo, SARSA, Q-Learning, Deep Q Network (DQN)를 접목시켜 효율적인 위성 라우팅 방법을 제안한다. [그림 1]은 저궤도 위성 라우팅의 개념을 설명하는 그림으로, 여러 개의 지구 저궤도 위성들이 서로 통신하며 송신지(Source)에서 목적지(Destination)까지 라우팅을 수행하는 모습을 나타낸다. 해당 토폴로지에서 주변 위성들의 링크 연결도를 고려하여, 최단 경로상에 있음에도 불구하고 링크가 끊긴 위성의 경우 라우팅에 활용되지 않는다.

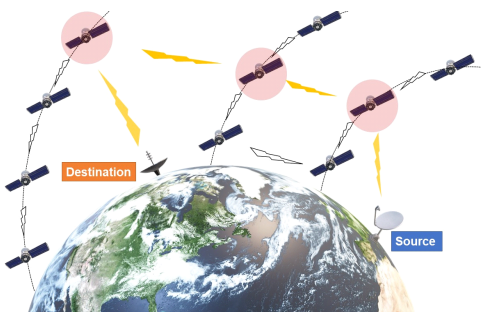

그림 1. 저궤도 위성 라우팅 개념도
Fig. 1. Overview of LEO satellite routing

본 논문에서는 2장에서 위성통신 시스템과 라우팅에 대해 서술한다. 이 장에서는 저궤도 위성 특징과 그로 인한 위성 라우팅의 필요성에 대해 서술한다. 3장에서는 강화학습에 대한 기본 개념을 소개하고 지구 저궤도 위성 라우팅 문제를 해결하기 위한 최적의 라우팅 알고리즘을 제안한다. 이어, 4장에서는 제안한 알고리즘에 대한 성능 평가를 제시하고 결론을 맺는다.

Ⅱ. 위성 통신망과 라우팅 기술

2.1 지구 저궤도 위성 통신망

네트워크 기술의 지속적인 발전에도 불구하고 전 세계적으로 원활한 글로벌 인터넷 서비스를 제공하는 것은 여전히 어려운 과제이다. 아직도 많은 지역에서 자연재해나 전쟁 등으로 인해 인터넷을 사용할 수 없다. 그래서 오늘날 원활한 글로벌 인터넷 서비스를 제공하기 위한 위성 기반 솔루션이 활발히 연구되고 있다. 그러나 기존의 위성 통신 시스템은 고도 36,000km에서 공전하는 지구 정지궤도 위성에 의존했기 때문에, 장거리 전파 지연 발생으로 인해 통신 속도가 상당히 낮았다[4]. 하지만 저궤도 위성 통신망은 이러한 문제점에 해결책을 제시할 수 있다. 고속 및 광역 위성 통신을 위해 (i) 위성의 고도를 낮추어 지연 시간을 줄이고 (ii) 위성의 수를 늘려 통신 가능 영역을 확장한다. 즉, 지구와 비교적 가까운 거리인 500km에 수많은 위성을 배치하여 통신시스템을 구축하는 것이 저궤도 위성 통신망의 기본적인 원리이다[5].

2.2 지구 저궤도 위성 라우팅

지구 저궤도 통신 시스템에서 라우팅은 데이터 패킷을 송신지에서 목적지까지 효율적으로 전달하는 과정을 뜻한다. 그러나 위성은 끊임없이 움직이기 때문에,

위성 통신 시스템에서의 라우팅은 지상에서의 통신 시스템과 상당히 다르게 작동한다. 대부분의 통신용 지구 저궤도 위성은 상공 500km 부근에 가까이 위치하기 때문에 행성 공전주기의 제곱이 궤도 장반경의 세제곱에 비례한다는 케플러 제3법칙에 의해 공전 주기 또한 짧다. 공전 주기가 짧기 때문에 공전 속도는 다른 위성 시스템에 비해 굉장히 빠르다. 저궤도 위성의 7.5km/s의 빠른 공전 속도는 지속적인 핸드오버를 야기하여 사용자 편의성을 저하시킨다. 그 결과 끊임없이 변화하는 시변성 동적 네트워크 토폴로지가 생겨 업데이트가 자주 발생한다. 이러한 빈번한 업데이트는 지연 시간 증가, 처리량 감소 등 통신 성능에 부정적인 영향을 끼친다. 그래서 저궤도 위성 시스템을 통해 원활한 인터넷을 제공하려면 네트워크 라우팅 정보를 실시간으로 업데이트하고 변화에 신속하게 대응 가능한 라우팅 알고리즘이 필요하다. 저궤도 위성 시스템을 통해 원활한 인터넷을 제공하려면 이러한 라우팅 문제를 해결해야 하고 이 문제를 해결함에 있어 강화학습은 좋은 해결책 중에 하나이다[6].

Ⅲ. 강화학습 기반 라우팅 매커니즘 개발

3.1 강화학습과 MDP

강화학습은 경험을 바탕으로 행동을 개선하는 인공지능 기법으로, 머신러닝의 주요 범주인 지도학습(Supervised Learning), 비지도학습(Unsupervised Learning), 강화학습(Reinforcement Learning) 중 하나에 속한다[7]. 강화학습의 핵심 원리는 “좋은 결과를 얻은 행동은 더 자주 수행하고, 좋지 않은 결과를 얻은 행동은 줄이는 것”이다. 환경(Environment) 내의 에이전트(Agent)는 경험을 통해 얻은 결과를 보상(Reward)의 형태로 받게 된다. 에이전트는 이러한 보상을 바탕으로 환경에서 스스로 학습하며, 지속적으로 최적의 정책을 수정해 나가면서 학습이 이루어진다.

강화학습은 순차적 의사결정(Sequential Decision Making) 문제를 해결하는 데 사용되며, 이러한 문제는 마르코프 결정 프로세스(Markov Decision Process, MDP)를 통해 정확하게 표현될 수 있다. MDP는 “현재 상황에서 해야 하는 일과 과거 경기 양상은 어떠한 관련도 없다.”는 마르코프 성질(Markov Property)을 따르는 프로세스를 의미한다. 즉, 미래는 오로지 현재에 의해 결정된다는 것이다. MDP의 구성요소는 식 (1)과 같다.

$$MDP \equiv (S, A, P, R, \gamma) \tag{1}$$

S는 상태(State), A는 행동(Action), P는 전이 확률 행렬(Transition Probability Matrix), R은 보상(Reward), γ는 감쇄 인자(Discount Factor)를 의미한다. 상태(S)는 에이전트가 환경에서 가질 수 있는 모든 상태가 포함된 집합을 의미한다. 행동(A)는 에이전트가 해당 환경에서 취할 수 있는 모든 행동을 포함하는 집합이다. 전이 확률 행렬(P)은 시간 t에서 특정 상태에 있을 때 에이전트가 특정 행동을 취했을 경우, 시간 t+1에서 다른 상태로 전이될 확률을 나타낸다. 보상(R)은 시간 t의 특정 상태에서 에이전트가 행동을 수행하였을 때 받는 결과값, 즉 보상을 출력하는 함수를 의미한다. 감쇄 인자(γ)는 미래에 얻을 보상을 조절하는 요소로, 현재 받은 보상에 비해 미래에 얻을 보상의 중요도를 나타낸다. γ값이 0에 가까울수록 에이전트는 당장 눈앞에 보이는 이익만을 추구하는 탐욕적인 행동을 취하게 되고, 반면에 γ값이 1에 가까울수록 미래에 얻을 보상에 더 큰 가중치를 두어 행동을 취하는 에이전트가 된다.

위를 통해, 보상에서 확장된 개념인 리턴(Return, G_t)을 이해할 수 있다. 리턴은 특정 시점 t로부터 미래에 받을 감쇄된 보상의 합으로, 식 (2)와 같이 표현된다. 강화학습에서 모든 에이전트는 누적된 보상의 합인 리턴을 최대화하는 방향으로 학습이 진행된다.

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{2}$$

위에서 설명한 바와 같이 감쇄 인자(γ)는 0과 1사이의 값을 가진다. 그렇기 때문에 현재로부터 멀어질수록 γ를 제곱해서 계속 곱해주기 때문에 미래에 받을 보상이 0에 수렴하게 된다. 따라서 γ를 통해 미래에 얻을 보상, 그리고 현재 얻을 보상의 가중치를 조절할 수 있다. 한편, 감쇄 인자를 사용하는 이유는 수학적 편의성을 고려하기 위함이다. γ를 1보다 작게 설정해줌으로써, 리턴이 무한대로 발산하는 것을 방지할 수 있다. 리턴이 무한대로 발산하면 리턴끼리의 비교가 힘들어지고, 그 값을 예측하기도 어려워진다. 만약 감쇄 인자가 $0 < \gamma < 1$의 값을 가지지 않는다면, 리턴은 수렴하지 않고 무한대로 발산하게 될 수 있다. 이러한 보상의 발산을 방지하기 위해 수학적 안전장치로서 감쇄 인자가 사용된다.

한편, 리턴은 에이전트의 정책 함수(Policy, π)에 의존적이다. 정책 함수란 특정 시점 t일 때, 각 상태(s_t)에서 에이전트가 어떤 행동(a_t)을 선택할지 정해주는 함

수이며, 이는 식 (3)과 같다.

$$\pi(a|s) = P[A_t = a|S_t = s] \quad (3)$$

정책 함수는 에이전트 안에 존재하며, 모든 에이전트는 보상의 합을 최대화하는 정책을 택한다. 에이전트는 환경 속에서 보상을 통해 학습하며 더 큰 보상을 얻기 위해 계속해서 정책을 교정해 나가며, 리턴의 기댓값을 가장 크게 할 수 있는 정책을 '최적정책(π^*)'이라고 한다. 결론적으로 강화학습은 최적정책을 학습하는 일련의 과정이라고 할 수 있다. 에이전트가 환경 속에서 정책π에 따라 학습하며 보상이 가장 높은 행동을 하기 위해서는, 상태별 가치(value)를 계산해야만 한다. 그리고 '가치'는 벨만 최적 방정식을 통해 구해질 수 있고 이는 식 (4)로 표현된다.

$$q_*(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \max[q_*(s', a')] \quad (4)$$

식 (4)에서, $q_*(s,a)$는 상태 액션 가치 함수(State-Action Value Function)로, 각 상태에서 선택할 수 있는 행동을 평가해주는 함수이다. 이는 통상적으로 간략히 Q 가치(Q-Value)로 표현되며, 본 논문에서도 동일한 표기법을 사용한다. Q 가치 함수에서는 상태 s와 행동 a가 동시에 입력값으로 들어간다. 동일한 행동 a_0를 취했어도, 행동을 취한 상태가 s_0냐, s_2냐에 따라 전혀 다른 상황이 되기 때문에 상태와 행동을 모두 고려해줘야 한다. R_s^a, γ, $P_{ss'}^a$는 각각 보상, 감쇄 인자, 전이 확률 행렬을 의미한다. 벨만 최적 방정식에서는 max(최대) 연산자가 가장 큰 특징이다. 벨만 최적 방정식에서는 π에 의한 확률적 요소가 사라지고, 최대값 연산자를 통해 Q 가치가 가장 높은 행동을 취한다. 그렇기 때문에 벨만 최적 방정식은 최적의 가치를 찾고자 할 때 사용되며, 이를 통해 최적 정책 π^*를 학습한다.

일반적으로 강화학습에서 순차적 의사결정 문제를 푸는 대표적 방법은 동적 계획법(Dynamic Programming)이다. 이는 벨만 방정식을 반복적으로 사용하여 임의로 초기화되어 있던 가치값들을 보다 실제적인 값으로 근사하는 방법론이다[8]. 그러나 상태/행동 집합의 크기에 따른 연산 복잡도가 증가하기 때문에, 비교적 간단한 문제만 해결할 수 있다[9]. 이에 본 논문의 Ⅲ.3장 부터, 보다 복잡한 환경에서 학습을 하고 위성의 라우팅 경로를 도출해 낼 수 있는 다양한 알고리즘(Monte Carlo, SARSA, Q-Learning, DQN)에 대해 소개한다.

3.2 라우팅 문제와 강화학습의 적합성

광범위한 커버리지 영역에 고품질 인터넷을 끊김 없이 공급하기 위해서, 위성 라우팅 기술은 시변(Time-Varing) 동적 토폴로지를 고려해야 한다. 위성들은 빠른 속도로 공전하기 때문에 시간에 따라 토폴로지가 지속적으로 변화하며 그에 맞춰 통신 요구사항도 끊임없이 바뀐다. 이러한 시변성 동적 토폴로지 환경에서 강화학습은 큰 힘을 발휘한다. 저궤도 위성 라우팅의 경우, 위성들이 지속적으로 이동하고 변화하는 환경에서 최적의 통신 경로를 찾아야 한다. 이러한 시변성 동적 환경에서 강화학습은 실시간으로 정보를 수집하고 경험을 통해 효율적인 라우팅 경로를 도출해 낼 수 있다.

또한 앞서 말했듯, 강화학습이 가장 잘 푸는 문제는 순차적 의사결정 문제이다. 위성 라우팅 문제 또한 여러 단계의 라우팅 의사결정을 통해 최종 목적 위성까지 통신 경로를 결정하는 것이기 때문에, 이러한 일련의 과정을 순차적 의사결정 과정이라고 볼 수 있다. 이러한 다양한 이유로 인해 복잡하고 동적인 위성 토폴로지 환경에서도 강화학습을 통해 효율적인 라우팅 전략을 도출해낼 수 있다.

3.3 Monte Carlo

Monte Carlo (MC)는 몬테카를로 방법론을 통해 정해진 MDP에서 가치를 평가하는 강화학습 알고리즘이다. 몬테로카를로 방법론은 "직접 측정하기 곤란한 통계량에 대해, 여러 차례 샘플 추출을 통해 해당 값을 추정하는 기법"이다[10]. 즉, 리턴을 여러 번 계산하여 그 평균을 내면 그 값은 실제 가치값에 수렴할 것이라는 원리를 이용한 학습방법이다. 즉, 대수의 법칙(Law of Large Numbers)에 의해, 샘플링을 많이 할수록 에이전트는 더 많은 경험을 하여, 가치 예측치가 점점 정확해 지는 것이다. MC는 수식 (5)를 사용하여 학습을 진행한다.

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha(G_t - Q(s_t,a_t)) \quad (5)$$

MC에서 에이전트는 리턴 G_t를 따라가도록 학습한다. 그래서 G_t를 MC의 목표(Target)라고 표현한다. 리턴 G_t와 에이전트가 관측한 가치값 $Q(s_t,a_t)$의 차이인 오류(Error)에 학습률(Learning Rate, α)을 곱해주고 이를 더해준다. 학습률은 한 번에 얼마만큼 학습을 업데이트할 것인지에 대한 크기를 결정해 주는 상수이다. MC에서 에이전트는 각 에피소드마다 리턴을 저장하고 에피소드가 끝나면 그 값을 평균 내어 업데이트를 한다.

3.4 SARSA & Q-Learning

3.4.1 SARSA

MC는 학습 업데이트를 하기 위해서 리턴이 필요한데 이는 에피소드가 종료돼야지 얻을 수 있기 때문에 MC는 종료하는 MDP에서만 사용할 수 있다. 그래서 종료되지 않는 MDP에서는 MC를 사용할 수 없게 되고, 이때는 Temporal Difference (TD) 학습을 사용해야 한다. TD는 에피소드가 종료되기 전에 가치값을 업데이트 한다. 이름에서 볼 수 있듯, 시간적 차이(Temporal Difference)를 이용하여 과거의 추측을 업데이트하기 위해 미래의 추측을 사용한다. 즉, 환경에서 에이전트가 한 스텝이라도 더 진행하면 과거보다 더 정확한 추측을 할 수 있게 되고, 이를 통해 학습 업데이트를 진행한다. TD를 통해 Q 가치를 근사하는 알고리즘을 State - Action - Reward - State-Action (SARSA)라고 하고, 이는 식 (6)을 통해 업데이트된다. MC의 업데이트 식과 유사하게 가치값이 목표 "$R+\gamma Q(s',a')$"에 근사될 수 있도록 오류에 α를 곱하여 업데이트가 진행된다.

$$Q(s,a) \leftarrow Q(s,a) + \alpha(R+\gamma Q(s',a') - Q(s,a)) \quad (6)$$

3.4.2 Q-Learning

Q-Learning은 현재 상태에서 특정 행동을 취할 때 받는 보상의 기댓값인 Q 가치를 학습시켜 최적의 정책을 찾아내는 강화학습 방법이다[11,12]. TD방식으로 업데이트 한 다는 점에서 SARSA와 동일하지만, 업데이트에 사용되는 수식이 다르다. 벨만 기대 방정식을 사용하는 SARSA와 다르게 Q-Learning은 벨만 최적 방정식을 통해 학습을 업데이트 한다. Q-Learning은 식 (7)을 통해 업데이트가 진행된다.

$$Q(s,a) \leftarrow Q(s,a) + \alpha(R+\gamma \max_{a'} Q(s',a') - Q(s,a)) \quad (7)$$

식 (6)과 식(7)의 가장 큰 차인점은 최대값 연산자(max)의 유무이다. Q-Learning은 on-policy인 SARSA와 다르게, 타깃 정책과 행동정책이 다른 off-policy 학습법이다. 여기서 타깃 정책은 학습의 대상이 되는 정책, 행동 정책은 실제 환경과 상호 작용하며 경험을 쌓고 있는 정책을 의미한다. SARSA는 행동 정책과 타깃 정책이 모두 Q에 대한 ε-greedy를 취하는 반면, Q-Learning의 타깃 정책은 Q에 대해 Greedy 정책을 취한다. Q-Learning에서 에이전트의 행동 정책에는 Exploration을 위한 ε이 들어있는 반면, 타깃 정책은 단순히 Q 가치가 높은 행동만 선택하는 Greedy 정책을 따른다. 이는 Q-Learning이 벨만 최적 방정식을 대상으로 학습하고, 벨만 최적 방정식에는 최댓값 연산자가 있기 때문이다. 그래서 Q-Learning은 On-Policy학습이 할 수 없는 과거의 경험 재사용, 일대다/다대일 학습이 가능하다.

3.4.3 Monte Carlo와 Temporal Difference 비교

이번 장에서는 MC와 TD에 대해 비교하고, 각각의 특징들에 대해 서술한다. MC와 TD이 대한 비교는 [표 1]과 같다.

MC와 TD 모두 목표에서 원래 가치값을 빼주고 거기에 학습률(α)를 곱한 후 다시 원래 가치값을 더해주어 업데이트 해나간다. 다만 업데이트를 하는 시점과 업데이트 하는데 사용되는 수식이 "가치함수의 정의"이냐 "벨만 방정식" 이냐의 차이이다. Ⅲ. 3.장에서도 언급했듯, MC는 가치함수의 정의인 리턴의 기댓값이, TD는 벨만 방정식이 학습에 근간이 되는 식이다. MC는 리턴을 기반으로 학습하기에 에피소드가 종료된 후에 업데이트 되지만, TD는 에피소드가 끝나지 않아도 n-step 후에 업데이트가 가능하다.

MC는 가치 함수의 정의인 리턴의 기댓값을 통해 학습하고 여러 개의 샘플 리턴을 모아서 평균을 내기 때문에, 편향되어 있지 않는(Unbiased) 방법이다. 그러나 TD의 경우, 타깃은 업데이트 하고 있는 테이블에 담겨있던 값일 뿐 실제 가치값과는 차이가 있을 수 있다. 즉, TD는 학습 데이터들이 편향(Biased)되어 있다고 할 수 있다.

MC의 경우, 리턴을 통해 업데이트가 이루어지는데, 리턴을 얻기 위해 에이전트는 수많은 확률적 과정을 거쳐야 한다. 그렇기 때문에 평균으로부터 각각의 학습 데이터들이 멀리 펴져있을 수 있고, 이는 분산(Variance) 및 변동성이 크다는 것을 의미한다. 그에 반해, TD는 하나의 샘플을 통해서 바로 업데이트를 할 수 있기 때문에 MC에 비해 분산 및 변동성이 작다.

표 1. Monte Carlo와 Temporal Difference 비교
Table 1. Comparing Monte Carlo and Temporal Difference

	Monte Carlo	Temporal Difference
목표	G_t	$R_{t+1}+\gamma Q(s_{t+1})$
오류	$G_t - Q(s_t)$	$R_{t+1}+\gamma Q(s_{t+1}) - Q(s_t)$
학습시점	episode 종료시	한 스텝이 종료시
편향성	편향되지 않음	편향됨
분산	변동성이 큼	변동성이 작음

TD는 고작 한/두 스텝후에 바로 업데이트를 하기 때문에 그 사이에 확률적 요소가 작용하기가 어렵기 때문이다.

3.5 Deep Q Network (DQN)

지금까지 본 MC, SARSA, Q-Learning은 모두 테이블 기반 학습법이다. 각 상태/행동에 대한 가치값을 테이블에 저장하고 그 값을 계속해서 업데이트하는 방식으로 학습이 이루어진다. 그러나 상태집합 s가 굉장히 커지거나, 이산적이지 않고 연속적인 경우에는 테이블을 만들기가 어렵다. 이 경우에는, 일일이 모든 데이터를 테이블에 저장할 수 없기 때문에, 인공신경망(Neural Network)을 이용하여 값을 일반화하고 학습을 진행해야 한다. 인공신경망은 많은 계층을 쌓음으로써 비선형성을 손쉽게 다룰 수 있는 유연한 함수이다. DQN은 위에서 언급한 Q-Learning에 인공신경망 개념을 도입한 학습법으로, 테이블 기반이 아닌 인공신경망을 통해 Q 가치값을 학습한다[13]. 특정한 정책을 학습했던 기존의 강화학습과 달리 DQN에서는 인공신경망의 파라미터 θ를 학습한다. 인공신경망의 구조는 [그림 2]와 같다. 위의 예시는 길이 4인 벡터를 인풋으로 받아 하나의 아웃풋을 도출하는 인공신경망이다. 5개의 노드로 구성된 3개의 히든레이어가 쌓여져 있는 형태이다. 노드는 이전 노드에서 들어오는 값을 선형 결합한 후 'relu', 'sigmoid' 같은 비선형 함수에 입력값(Input)으로 대입한다. n번째 레이어의 노드들은 n-1레이어의 노드들의 결합에 의해 생성된 피처(Feature)이기 때문에, 레이어가 윗단으로 갈수록 더욱 추상화된 피처가 형성된다. DQN에서는 Q-Learning과 유사하게 벨만 최적 방정식인 $R+\gamma \max Q(s',a')$을 타겟으로 보며, 손실함수를 식 (8)과 같이 정의한다.

$$L(\theta)' = E[(R+\gamma max Q_\theta(s',a') - Q_\theta(s,a))^2] \quad (8)$$

손실함수에 대한 파라미터 θ의 영향력을 파악하기 위해 편미분을 하는데, DQN의 학습은 편미분이 사용된 식 (9)를 통해 업데이트된다.

$$\theta' \leftarrow \theta + \alpha(R+\gamma \max_{a'} Q_\theta(s_{t+1},a_{t+1}) - Q_\theta(s_t,a_t)) \times \nabla_\theta Q_\theta(s_t,a_t) \quad (9)$$

기존에도 강화학습에 인공신경망을 접목하려는 시도들은 많이 존재해왔으나, 인공신경망을 활용해 Q 함수를 근사하려고 했던 과거의 시도들은 알고리즘이 수렴하지 않거나 학습이 불안정한 단점이 존재했다. 그 이유는 크게 '샘플 간의 상관관계', '움직이는 타겟 네트워크'였다. DQN은 경험 재사용(Experience Replay), 과 타겟 네트워크(Target Network)를 활용하여 위의 문제를 해결하였고 안정적인 학습 수렴을 보장하였다[14].

3.5.1 리플레이 버퍼(Replay Buffer)

리플레이 버퍼는 데이터 사이의 상관성을 줄여주기 위해 나온 방법이다. 가장 최근 n개의 데이터를 버퍼에 넣어줘서 선입선출로 데이터를 관리한다. 그리고 학습할 때, 버퍼에서 특정 미니배치 크기만큼 데이터를 랜덤하게 뽑게 되면, 모든 데이터를 여러 번 재사용할 수 있게 된다. 이와 같이 다양한 데이터로 학습을 하게 되면, 데이터 사이의 상관성을 크게 떨어뜨려, 보다 빠르게 수렴하는 학습 모델을 도출해낼 수 있다.

3.5.2 타겟 네트워크 (Target Network)

손실함수를 줄이는 방향으로 θ가 업데이트 되는데, 여기서 $R+\gamma \max Q_\theta(s',a')$가 타겟으로 활용된다. 수식에서 알 수 있듯, 타겟이 θ가 바뀜에 따라 같이 바뀐다. 즉, θ가 업데이트될 때 마다 타겟인 $R+\gamma \max Q_\theta(s',a')$도 같이 움직여서 안정적인 학습에 해가 된다. 이러한 문제점을 해결하기 위해, DQN에서는 별도의 타깃 네트워크 개념을 도입했다. 이는 손실함수를 계산할 때, 네트워크 파라미터를 잠시 얼려서 타겟을 고정시키는 것이다. 그리고 특정 주기마다 얼려 놓았던 θ를 새로운 파라미터로 교체해줘서 학습의 안정성을 도모했다.

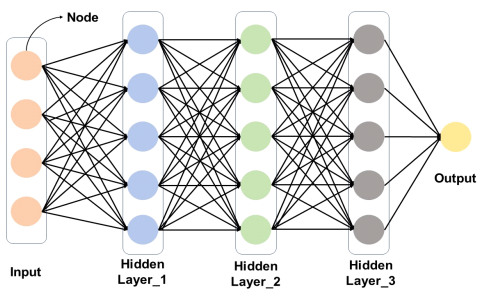

그림 2. 인공신경망의 구조
Fig. 2. Structure of artificial neural network

Ⅳ. 실험 및 평가

4.1 실험 설계

본 장에서는 강화학습 수식화에 필요한 상태, 행동, 보상함수, ε-greedy 및 기타 변수를 정의함으로써 실험

설계에 관해 논한다. 위성 라우팅 문제에서 전송지연을 고려하는 Queueing 기반 모델링은 흔히 수행된다. 본 논문에서의 위성은 비디오 스트리밍이나 대용량 이미지 전송과 같은 고대역폭 어플리케이션용으로 설계된 High Throughput Satellite (HTS)이다. 또한 본 논문에서는 위성끼리 위성사진 데이터를 보낸다고 가정하였다. 일반적인 4K Ultra-High-Definition (UHD) 프레임 용량은 압축시 최대 10MB 정도이다. 초당 하나의 이미지를 전송하려면 이 경우, 80Mbps가 필요하다. 따라서 처리량이 높게 설계된 HTS를 고려하는 본 논문에서는 100Gbps의 처리량으로 대역폭을 관리할 수 있기 때문에 Queuing 지연이 발생할 여지는 존재하지 않는다[15].

4.1.1 상태

실험은 15×15 그리드, 총 225개의 위성이 있는 환경에서 진행된다. 지구 위를 공전하는 위성 환경을 모방하기 위해 본 실험 환경에서의 그리드는 일반적인 그리드와 다르게, 모든 꼭지점이 연결돼 있다. 즉, 가장 오른쪽 그리드에서 에이전트가 Right 행동을 취하면, 에피소드가 종료되는 것이 아니라, 가장 왼쪽 그리드로 상태 변화를 하게 된다. 이를 통해, 본 실험에서의 15×15 그리드는 2차원 평면이 아닌, 3차원 구의 형태를 띄고 있다.

상태로는 출발 위성(Source)의 절대 위치, 목표 위성(Destination)의 절대 위치, 링크가 끊긴 위성의 개수, 링크가 끊긴 위성의 절대 위치와 같이 라우팅에 영향을 줄 수 있는 시변성 동적 토폴로지가 들어가게 된다. 에이전트는 시간에 따라 빠르게 변하는 다양한 위성의 토폴로지에 대응하여 해당 상태에서 어떠한 위성으로 라우팅을 해줘야 하는지 학습하게 된다. 이러한 환경에서 에이전트는 Source에서 Destination까지 최단 경로로 라우팅을 함에 있어, 다음 홉(Hop)을 선택하게 되고, 이는 에이전트의 행동에 해당한다.

4.1.2 행동

식 (10)은 정해진 토폴로지에서 위성이 선택할 수 있는 행동 집합을 나타낸다. 행동 A는 구처럼 연결된 그리드 환경에서 위성이 위, 아래, 오른쪽, 왼쪽 방향으로 라우팅하는 것을 의미한다. 주어진 환경에서 에이전트는 보상함수를 기반으로 하여, 가장 효율적이고 빠른 라우팅 경로를 채택하기 위해 행동 집합 A에서 최적의 행동을 선택한다.

$$A = [Up, Down, Right, Left] \tag{10}$$

4.1.3 보상함수

보상함수는 식 (11)과 같이 정의하였다. 위성이 라우팅 함에 있어, Hop-by-Hop으로 한 스텝씩 진전할 때마다 -1의 리워드를 받게 된다. 한 스텝당 경미한 음의 리워드를 받기 때문에, 에이전트는 최대의 보상을 얻기 위해서 최단 경로로 라우팅을 한다. 또한, 에이전트가 링크가 끊긴 위성으로 라우팅 경로를 설정할 경우, -100의 음의 보상을 줘서 링크가 끊겨서 데이터를 전송하지 못 하는 일을 방지한다. 즉, 위성은 링크 연결도가 좋은 위성들에게만 라우팅 경로를 설정할 수 있도록 하여 통신 과부하를 예방한다. 환경은 위성 라우팅이 Source에서 Destination까지 도달하면 에피소드가 끝났음을 에이전트에 알려주고 초기상태로 초기화해준다.

$$R = \begin{cases} -1 & per\ one\ step \\ -100 & \text{if}\ routing\ fails \end{cases} \tag{11}$$

4.1.4 ε-greedy 및 기타 변수

해당 학습과정에서는 Greedy정책 대신에 ε-greedy 정책을 사용한다. 강화학습에서 원활한 학습을 위해서는 에이전트의 Exploration과 Exploitation을 각각 보장해줘야 한다[16]. 에이전트가 모든 상태에서 모든 행동의 가치값을 정확히 알고 있다면, 현재 상황에서 가치값이 가장 높은 행동만 취하는 것이 가장 좋은 방법이다. 하지만 학습 초기에는 예측한 가치값에 불확실성(Uncertainty)이 존재하기 때문에 에이전트가 충분히 학습할 수 있도록 Exploration을 보장해줘야 한다. ε-greedy는 에이전트의 탐색의 정도를 보장해주기 위한 가장 단순하고 강력한 방법론이다. 에이전트는 항상 가치가 최대가 되는 행동을 취하지 않고, ε이라는 작은 확률 만큼은 랜덤하게 행동을 취한다. 그리고 1-ε이라는 나머지 확률만큼은 가치값이 큰 방향으로 탐욕적으로(Greedy) 행동을 선택한다. 학습 초기에는 학습 데이터가 많이 없기 때문에 랜덤으로 행동을 취할 확률인 ε값을 높혀줬다가 학습이 어느 정도 진행되면 ε값을 낮춰준다. 이러한 방법론을 'Decaying ε-greedy 방식'이라고 하고 본 학습에서는 Decaying ε-greedy방식을 사용하였다. 본 학습 단계에서 초기 ε값은 0.3으로 설정되어 있고, 에피소드가 진행됨에 따라 ε이 에피소드 당 0.005씩 떨어지면서 0.03까지 감소한다.

DQN 알고리즘으로 에이전트를 학습시킬 때, 뉴럴넷 학습 파라미터는 [표 2]에 표현돼 있다. 에이전트는 학습을 할 때, 미니배치 크기인 64만큼 데이터를 뽑아와 학습 데이터로 사용한다. 에피소드는 1000단위로 학습이 진행되며, 감쇄 인자는 0.99, 학습률은 0.001로 설

표 2. 실험 초기 환경 설정
Table 2. Experimental environment setup

Notation	Value
미니배치 크기, $n(M)$	64
감쇄 인자, γ	0.99
학습을 진행한 에피소드 수, E	1,000
신경망 층 높이, L	3개
노드 수, n	64
학습률, α	0.001
초기 입실론 값, $\epsilon_{\max}$	0.3
에피소드 진행에 따른 입실론의 감쇄 정도, $\epsilon_{decay\ speed}$	0.005
입실론의 최솟값, $\epsilon_{\min}$	0.03
활성함수	ReLU
옵티마이저	Adam

정돼 있다. 뉴럴넷은 64개의 노드로 이루어진 히든레이어가 3개 존재한다. 다음과 같은 실험환경에서 학습을 진행하였고, 다음 장에서는 본 실험 환경에서 진행한 성능평가에 대해 서술한다.

4.2 실험 결과

4.2.1 최종 보상

강화학습에서 에이전트는 누적된 보상의 합을 최대화하는 방향으로 학습이 이루어진다. 그렇기 때문에 에피소드가 진행됨에 따라 보상이 어떻게 변화하는지를 보고 해당 에이전트가 잘 학습이 됐는지 판단할 수 있다. 이를 통해 학습 과정에서 얻은 보상을 관찰하여, 강화학습 알고리즘이 해당 문제에서 잘 수렴하는지를 확인할 수 있다. [그림 3]은 에피소드가 진행됨에 따라 각 알고리즘 별 보상 추이를 나타낸다.

MC, SARSA, Q-Learning 모두 랜덤하게 행동을 선택하는 것에 비해 에피소드가 진행됨에 따라, 보상값의 추이가 우상향하는 것을 확인할 수 있다. 또한 보상값의 변동 추이 또한 제안한 알고리즘들이 대조군(랜덤)보다 적은 것을 확인할 수 있다.

[그림 4]는 DQN을 위성 라우팅에 접목했을 때, 에피소드 진행에 따른 보상값의 변화를 보여준다. DQN의 경우, 이전의 MC, SARSA, Q-Learning과 다르게 보상함수를 한 스텝 당 -0.01, 링크가 끊긴 위성으로 라우팅 했을 때 -0.5, 라우팅에 성공했을 때 +1로 설정하였다. 다른 하이퍼 파라미터들은 이전의 실험과 동일하다. DQN의 경우도, MC, SARSA, Q-Learning과 비슷하게, 에피소드가 진행됨에 따라 보상값이 우상향하는 것

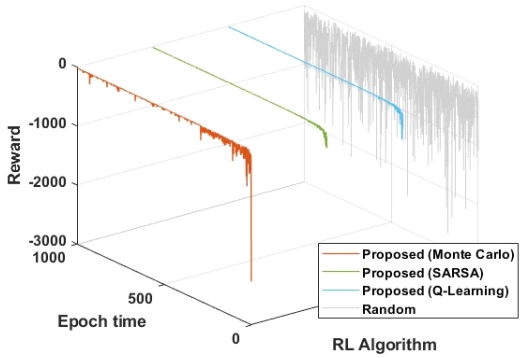

그림 3. 에피소드 진행에 따른 보상 값 (MC, SARSA, Q-Learning)
Fig. 3. Reward value according to episode (MC, SARSA, Q-Learning)

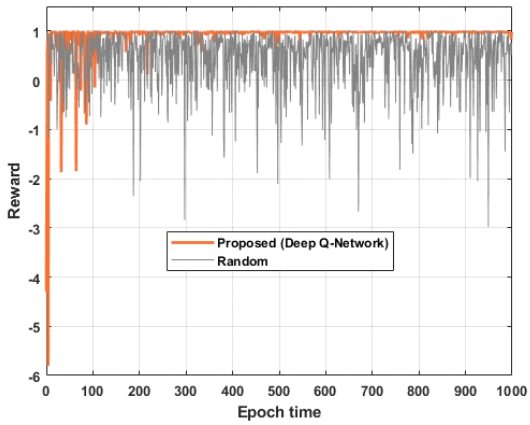

그림 4. 에피소드 진행에 따른 보상 값 (DQN)
Fig. 4. Reward value according to episode (DQN)

을 확인할 수 있다. 보상값의 변동 또한 대조군에 비해 안정적인 모습을 보여준다. 위성이 링크가 끊긴 위성을 피해 최적 경로로 빠르게 라우팅할 수 있도록 보상 함수를 설계했기 때문에 보다 높은 보상값을 가지는 MC, SARSA, Q-Learning, DQN 알고리즘이 대조군보다 목표를 더욱 잘 달성했다고 볼 수 있다.

4.2.2 라우팅 스텝

[그림 5]는 위성이 Source부터 Destination까지 라우팅을 함에 있어, 평균적으로 간 스텝(홉)을 나타낸다. 그래프에서 각 Box Plot의 중앙에 있는 빨간색 선은 라우팅 스텝의 중앙값을 의미하고 가장 밑에 있는 선과 가장 위에 있는 선은 각각 각각 25%, 75% 백분위에 위치한 라우팅 스텝값이다. 제안하는 알고리즘 (Monte Carlo, SARSA, Q-Learning, DQN)이 대조군에 비해

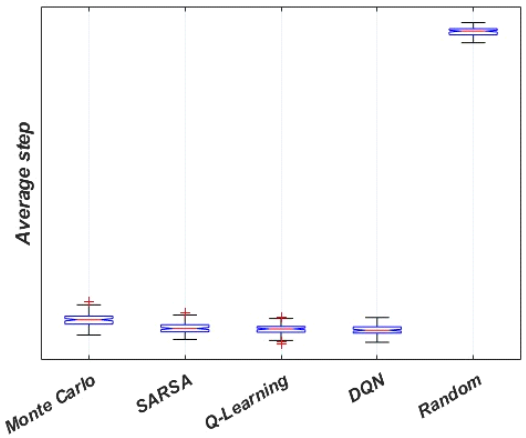

그림 5. 평균 라우팅 거리
Fig. 5. Average routing distance

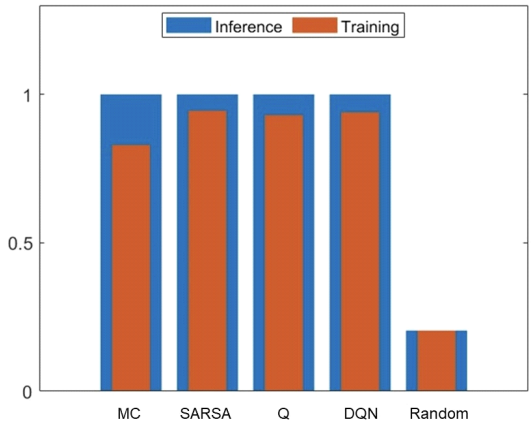

그림 6. 라우팅 성공률
Fig. 6. Routing success rate

표 3 라우팅 성공률 비교
Table 3 Routing success rate

	MC	SARSA	Q	DQN	Random
추론	100%	100%	100%	100%	20.2%
학습	83%	93.5%	93.0%	94.1%	20.2%

평균 라우팅 스텝이 굉장히 짧은 것을 확인할 수 있다. 동일한 네트워크 토폴로지에서, 랜덤하게 행동을 취하는 대조군은 본 논문에서 제시하는 알고리즘보다 평균적으로 23.4배 더 많은 스텝을 거쳐갔다. 이는 같은 토폴로지여도 제안하는 알고리즘이 더욱 빠르게 라우팅에 성공했음을 의미한다. 제안하는 알고리즘의 에이전트는 학습을 진행하며, 링크가 끊긴 위성을 피해 최대한 빠르게 라우팅할 수 있는 최단 경로를 학습하게 된다. Source부터 Destination까지 라우팅을 함에 있어, 제안하는 알고리즘은 대조군에 비해 월등한 성능을 보이며, 각 알고리즘끼리는 비슷한 평균 라우팅 스텝을 보인다.

4.2.3 라우팅 성공률

[그림 6]은 위성의 라우팅 성공률을 나타낸다. 파란색 막대 그래프는 추론과정에서, 주황색 막대 그래프는 학습 과정에서의 라우팅 성공률을 의미한다. 기본적으로 학습 과정에서는 랜덤하게 행동을 취하는 확률적 요소인 ε값이 살아 있다. 반면, 추론 과정에서는 항상 가치값이 높은 행동을 취하기 때문에, 모든 막대 그래프에서 추론 라우팅 성공률이 학습 라우팅 성공률보다 높은 모습을 보인다. [그림 6]을 통해, 제안하는 알고리즘(Monte Carlo, SARSA, Q-Learning, DQN)이 대조군에 비해 라우팅 성공률이 높은 것을 알 수 있다. 각 알고리즘 별 정확한 라우팅 성공률 수치는 [표 3]과 같다.

제안한 알고리즘 (MC, SARSA, Q-Learning, DQN)이 대조군보다 약 4.5배 더 높은 라우팅 성공률을 보였다. 이는 본 논문에서 제안하는 알고리즘을 저궤도 위성 라우팅에 접목하였을 때, 우수한 성능을 보였음을 의미한다. [그림 3, 4, 5, 6]은 제안하는 알고리즘이 라우팅 성공률 및 위성 통신 안정성 향상에 크게 기여했다는 것을 보여준다.

V. 결 론

본 논문은 지구 저궤도 위성의 자율적인 네트워크 구축을 위해, 강화학습 기반 라우팅 기술을 제안하고 그 성능을 검증하였다. 지구 저궤도 위성 통신망에서는 동적인 궤도 환경, 빠른 공전 속도, 위성 밀집도 등의 특성으로 인해 효율적인 라우팅 알고리즘이 필수적으로 요구된다. 강화학습은 이러한 도전 과제를 해결하기 위한 유망한 방법으로 간주되며, 본 논문에서는 해당 문제에 다양한 강화학습 알고리즘을 적용하여 지구 저궤도 위성 네트워크의 라우팅 성능을 평가하였다. 실험 결과를 통해, 제안한 강화학습 기반 라우팅 기술이 높은 성능과 안정성을 보임을 증명하였고, 이를 통해 지구 저궤도 위성 네트워크의 자율화 및 효율성을 높이는데 기여할 것으로 기대된다.

References

[1] M. Neinavaie, J. Khalife, and Z. M. Kassas, “Acquisition, doppler tracking, and positioning with starlink LEO satellites: First results,” *IEEE Trans. Aerospace and Electr. Syst.*, vol.

58, no. 3, pp. 2606-2610, Jun. 2022. (https://doi.org/10.1109/taes.2021.3127488)

[2] K. E. Eichensehr, "Ukraine, cyberattacks, and the lessons for international law," *Am. J. Int. Law*, vol. 116, pp. 145-149, May 2022. (https://doi.org/10.1017/aju.2022.20)

[3] H. Lee and J. Kim, "Survey on deep reinforcement learning applied for LEO satellites," *J. KICS*, vol. 48, no. 2, pp. 196-205, Feb. 2023. (https://doi.org/10.7840/kics.2023.48.2.196)

[4] Y. Su, Y. Liu, Y. Zhou, J. Yuan, H. Cao, and J. Shi, "Broadband LEO satellite communications: Architectures and key technologies," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 55-61, Apr. 2019. (https://doi.org/10.1109/mwc.2019.1800299)

[5] B. Di, L. Song, Y. Li, and H. V. Poor, "Ultra-Dense LEO: Integration of satellite access networks into 5G and beyond," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 62-69, Apr. 2019. (https://doi.org/10.1109/mwc.2019.1800301)

[6] P. Zuo, C. Wang, Z. Yao, S. Hou, and H. Jiang, "An intelligent routing algorithm for leo satellites based on deep reinforcement learning" in *Proc. IEEE VTC*, pp. 1-5, Norman, OK, USA, Sep. 2021. (https://doi.org/10.1109/vtc2021-fall52928.2021.9625325)

[7] G.-P. Antonio and C. Maria-Dolores, "Multi-agent deep reinforcement learning to manage connected autonomous vehicles at tomorrow's intersections," *IEEE Trans. Veh. Technol.*, vol. 71, no. 7, pp. 7033-7043, Jul. 2022. (https://doi.org/10.1109/tvt.2022.3169907)

[8] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34-37, Jul. 1966. (https://doi.org/10.1126/science.153.3731.34)

[9] C. Park, G. S. Kim, K. Lee, and I. Yun, "Multi-agent deep reinforcement learning for efficient unattended information gathering and monitoring of autonomous UAM systems," *J. KICS*, vol. 48, no. 2, pp. 176-184, Feb. 2023. (https://doi.org/10.7840/kics.2023.48.2.176)

[10] K. Q. Nguyen and R. Thawonmas, "Monte carlo tree search for collaboration control of ghosts in Ms. Pac-Man," *IEEE Trans. Computational Intell. and AI in Games*, vol. 5, no. 1, pp. 57-68, Mar. 2013. (https://doi.org/10.1109/tciaig.2012.2214776)

[11] V. B. Ajabshir, M. S. Guzel, and E. Bostanci, "A low-cost q-learning-based approach to handle continuous space problems for decentralized multi-agent robot navigation in cluttered environments," *IEEE Access*, vol. 10, pp. 35287-35301, Mar. 2022. (https://doi.org/10.1109/access.2022.3163393)

[12] T. Jung, S. Kim, and K. Kim, "N-DQN: Study on the implementation and research of hierarchical parallel reinforcement learning model," *J. KICS*, vol. 44, no. 10, pp. 1961-1974, Oct. 2019. (https://doi.org/10.7840/kics.2019.44.10.1961)

[13] N. Gholizadeh, N. Kazemi, and P. Musilek, "A comparative study of reinforcement learning algorithms for distribution network reconfiguration with deep q-learning-based action sampling," *IEEE Access*, vol. 11, pp. 13714-13723, Feb. 2023. (https://doi.org/10.1109/access.2023.3243549)

[14] V. Mnih, et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529-533, Feb. 2015. (https://doi.org/10.1038/nature14236)

[15] L.D.C.H.R. Gaytan, et al., "Dynamic scheduling for high throughput satellites employing priority code scheme," *IEEE Access,* vol. 3, pp. 2044-2054, Oct. 2015. (https://doi.org/10.1109/access.2015.2495226)

[16] M. Kaloev and G. Krastev, "Experiments focused on exploration in deep reinforcement learning," in *Proc. IEEE ISMSIT*, pp. 351-355, Ankara, Turkey, Nov. 2021. (https://doi.org/10.1109/ismsit52890.2021.9604690)

김 규 선 (Gyu Seon Kim)

2023년 2월 : 인하대학교 공과대학 기계항공공학부 항공우주공학과 졸업(공학사)
2023년 3월~현재 : 고려대학교 공과대학 전기전자공학과 석박사통합과정

<관심분야> Reinforcement Learning Based Autonomous Aircraft Control, Aerial/Satellite Communication, Aircraft Embedded System
[ORCID:0000-0002-5559-9749]

박 수 현 (Soohyun Park)

2019년 2월 : 중앙대학교 소프트웨어대학 컴퓨터공학과 졸업 (공학사)
2023년 8월 : 고려대학교 공과대학 전기전자공학과 졸업 (공학박사)
2023년 9월~현재 : 고려대학교 공과대학 전기전자공학과 박사후 연구원
<관심분야> Deep Learning Theory, Network/Mobility Applications, Quantum Machine Learning, AI-based Autonomous Control
[ORCID:0000-0002-6556-9746]

김 중 헌 (Joongheon Kim)

2004년 2월 : 고려대학교 컴퓨터학과 졸업
2006년 2월 : 고려대학교 컴퓨터학과 석사
2014년 8월 : University of Southern California Computer Science 박사
2016년 3월~2019년 8월 : 중앙대학교 소프트웨어대학 조교수
2019년 9월~현재 : 고려대학교 전기전자공학부 부교수
<관심분야> Stochastic Optimization, Mobility, Reinforcement Learning, Quantum
[ORCID:0000-0003-2126-768X]

김 영 구 (Yeonggoo Kim)

1986년 2월 : 한양대학교 전자통신공학과 졸업(공학사)
1988년 2월 : 한양대학교 전자통신공학과 졸업(공학석사)
1988년 1월~1994 7월 : ㈜LG정보통신 선임연구원
1997년 3월~2001년 1월 : ㈜LG텔레콤 책임연구원
2001년 7월~2002년 12월 : ㈜시머스 기술이사
2004년 5월~2006년 12월 : ㈜바른정보커뮤니티 기술이사
2007년 6월~현재:㈜솔빛시스템 대표이사
<관심분야> Network Modeling & Simulation, Edge/On Device AI
[ORCID:0009-0000-3706-8859]

하 재 경 (Jaekyoung Ha)

1994년 2월 : 한양대학교 전자통신학과 졸업(공학사)
1998년 8월 : 한양대학교 전파공학과 졸업(공학석사)
1994년 2월~2003년 2월 : ㈜LG전자 선임연구원
2003년 2월~2006년 4월 : ㈜바른정보커뮤니티 책임연구원
2006년 9월~2007년 10월 : ㈜기산텔레콤 책임연구원
2007년 11월~현재 : ㈜솔빛시스템 전무
<관심분야> Edge AI SoC & Lightening Neural Network, Reinforcement Learning, Network Modeling & Simulation
[ORCID:0009-0000-3797-7809]

전 병 현 (Byung Hyun Jun)

1996년 2월 : 경희대학교 전자계산공학과 졸업(공학사)
1998년 2월 : 경희대학교 전자계산공학과 졸업(공학석사)
2014년 8월 : 경희대학교 전자계산공학과 졸업(공학박사)
2000년 5월~2001년 6월 : ㈜모인밸리 선임연구원
2001년 6월~2004년 3월 : ㈜에프아이디 책임연구원
2004년 7월~2005년 1월 : ㈜업필 선임연구원
2006년 3월~2006년 5월 : ㈜아이컨택트 수석연구원
2009년 11월~2015년 1월 : ㈜아이컨택트 수석연구원
2015년 3월~2016년 9월 : ㈜웨이브일렉트로닉스 수석연구원
2016년 10월~2017년 7월 : 플러스컴퓨팅 수석연구원
2017년 8월~현재 : ㈜솔빛시스템 수석연구원
<관심분야> Combinatorial Optimization, Graph Theory, Reinforcement Learning, Generative AI
[ORCID:0009-0005-8482-7981]