논문 2023-18-22

동적 저궤도 위성 네트워크를 위한 Dueling DQN 기반 라우팅 기법

(Dueling DQN-based Routing for Dynamic LEO Satellite Networks)

김 도 형, 이 상 현, 이 헌 철*, 원 동 식
(Dohyung Kim, Sanghyeon Lee, Heoncheol Lee, Dongshik Won)

Abstract : This paper deals with a routing algorithm which can find the best communication route to a desired point considering disconnected links in the LEO (low earth orbit) satellite networks. If the LEO satellite networks are dynamic, the number and distribution of the disconnected links are varying, which makes the routing problem challenging. To solve the problem, in this paper, we propose a routing method based on Dueling DQN which is one of the reinforcement learning algorithms. The proposed method was successfully conducted and verified by showing improved performance by reducing convergence times and converging more stably compared to other existing reinforcement learning-based routing algorithms.

Keywords : Dueling DQN, Low-earth-orbit satellite network (LEO-SN), Routing algorithm

Ⅰ. 서 론

최근 지구 전역의 초고속 네트워크 구성을 위한 여러 방법이 대두되고 있다. 그중 특히 많은 관심을 받는 분야는 저궤도 위성 네트워크 (Low Earth Orbit Satellite Network)를 사용한 방법이다. 저궤도 위성 네트워크는 수백 개 이상의 작은 위성을 지구 저궤도에 배치하여 구성된 시스템이다. 이러한 위성들은 대부분의 통신, 인터넷, 센서 및 관측 서비스를 제공하는 데에 사용된다. 저궤도 위성 네트워크를 사용하면 전송 지연이 적으면서 전역 커버리지를 달성할 수 있고, 지상의 설치된 지점 간 통신을 향상하거나 원격 지역이나 통신 기반이 없는 지역에서도 인터넷 접근성을 개선할 수 있다는 장점 때문에 활발히 연구가 진행 중이다. 하지만 저궤도 위성의 빠른 실행속도와 빈번하게 변화하는 위성 토폴로지 때문에 지상에 적용되는 라우팅 알고리즘을 저궤도 위성 네트워크에 바로 적용할 수는 없다. 따라서 저궤도 위성 네트워크에서는 실시간으로 변화하는 환경에 맞출 수 있는 새로운 라우팅 알고리즘을 제시해야 한다.

표 1은 저궤도 위성군 라우팅 알고리즘에 대한 기존 논문들이다. LEO-SN을 위한 라우팅 연구들은 지구 전역의 전역 커버리지 달성을 위해 활발하게 연구됐다. 다음의 제시되는 논문들은 LEO-SN에서 어떤 부분들을 고려하여 작성

*Corresponding Author (hclee@kumoh.ac.kr)
Received: Apr. 20, 2023, Revised: May. 23, 2023, Accepted: Jun. 14, 2023.
D. Kim: Department of IT Convergence Engineering, Kumoh National Institute of Technology (B.S. Student)
S. Lee: Research Institute of Manufacturing Technology, Kumoh National Institute of Technology (Researcher)
H. Lee: Department of IT Convergence Engineering, Kumoh National Institute of Technology (Assist. Prof.)
D. Won: TelePIX Co., Ltd (Director)
※ 이 연구는 2022년 정부 (방위사업청)의 재원으로 국방과학연구소의 지원을 받아 수행된 연구임 (UI220033VD).

ⓒ IEMEK J. Embed. Sys. Appl. 2023 Aug. 18(4) 173-183
ISSN : 1975-5066
http://dx.doi.org/10.14372/IEMEK.2023.18.4.173

표 1. LEO-SN 환경에 적용된 라우팅 알고리즘 관련 논문
Table 1. Papers on routing algorithm applied to LEO-SN environment

Related Works	Based on Reinforcement Learning	Consider dynamic environment	Consider link disconnection situation
[1]	O	X	X
[2]	O (actor-critic)	O	O
[3]	O	X	X
[4]	O	O	X
Ours	O	O	O

됐는지를 보여준다. 첫 번째로 강화 학습 기반의 라우팅 알고리즘을 사용하였는가에 대한 부분이다 [1-4]. 다음으로 실제 네트워크 환경은 위성이 지구 주위를 정해진 궤도로 공전하기 때문에 동적 라우팅 환경을 고려하였는지를 구분하였는지를 보여준다 [2, 4]. 마지막으로는 위성 링크 단절 상황 고려 여부에 관한 표이다. 위성은 실시간으로 지구를 공전하고 있고, 또한 이들의 위치 및 링크 단절 상황도 실시간으로 변화하기 때문에 이를 고려하는 것은 매우 중요하다. 우리가 제시하는 방법은 강화 학습 기반의 라우팅 알고리즘이며 동적 라우팅 환경을 표현하였다. 또한 위성 간 링크 단절 상황을 고려하였다. 표 1에서 고려하는 점들을 모두 고려한 actor-critic을 사용한 논문의 경우, Ⅳ 실험 결과와 함께 서술하였다.

강화 학습 기반 라우팅 알고리즘이 필요한 이유는 실시간으로 변화하는 환경에 대해 강한 내성을 가지기 때문이다. 실제 LEO-SN 환경에서는 위치가 실시간으로 변화하고 링크 단절 구간도 존재한다. 이를 유연하게 대처하기 위해서

는 기존의 전통적인 라우팅 알고리즘이 아닌 강화 학습이 필요하다 [5]. 저궤도 위성군에서는 다양한 환경적 요인들이 동적으로 변화하고, 위성군 내부에서 수많은 위성이 서로 상호작용을 해야 한다 [6]. 이에 따라 위성 간의 데이터 통신 과정에서 발생하는 라우팅 문제는 매우 복잡하며, 많은 요인이 영향을 미친다. 이러한 복잡성과 동적성으로 인해 이러한 문제를 해결하는데 강화 학습이 필요하다. 강화 학습은 복잡하고 동적인 환경에서 적응적인 행동을 결정하는데 효과적인 방법이다 [7]. 또한 강화 학습은 장애물 등 실시간으로 등장하는 변화에 대응할 수 있는 능력을 갖추고 있다. 이러한 이유로, 저궤도 위성군에서의 라우팅 문제를 강화 학습으로 해결하는 것은 매우 유용하다. DQN은 강화 학습 알고리즘 중 하나로, 실제로 복잡한 문제에서 높은 성능을 보인다. 따라서 DQN을 활용하여 위성군에서의 라우팅 문제를 해결하는 것은 매우 효과적이다.

이 논문은 크게 2가지 문제점을 해결하였다. 첫 번째로 동적 라우팅 환경을 고려하여 강화 학습 기반 라우팅 알고리즘을 제시한다는 것, 두 번째는 위성의 링크 단절 상황을 고려하여 강화 학습을 진행하였다는 점이다.

본 논문의 나머지 부분은 다음과 같이 구성된다. 우선 저궤도 위성 네트워크에서의 라우팅 문제를 정의한다. 다음으로 기존 강화 학습 기반 라우팅에 관한 기존 연구의 한계점에 관해 기술하고 우리의 기법을 제안한다. 기법은 크게 3가지고 1. 랜덤 Reward 모델 2. Grid MDP 기반 동적 네트워크 모델 3. Dueling DQN 기반 학습이다. 마지막으로는 여러 가지 상황 및 타 알고리즘과의 비교를 통해 결과를 도출한다. 여기서 여러 가지 상황이란 장애물의 개수나 목표 지점의 변화 등을 의미하며, 해당 결과들에 대한 정량적 분석을 포함한다.

Ⅱ. 문제점 기술

1. 저궤도 위성 네트워크에서의 라우팅 문제

저궤도 위성 네트워크에서의 라우팅 문제는, 여러 개의 위성으로 이루어진 네트워크에서 각 위성 간의 데이터 전송 경로를 결정하는 문제이다. 이러한 문제를 수학적으로 정의하면, 그리드 마르코프 결정 과정 (Grid Markov Decision Process, Grid MDP)으로 표현할 수 있다. 이는 주로 “시간 t에서의 상태는 t-1에서의 상태에만 영향을 받는다”라는 first-order Markov assumption을 기반으로 고안됐다 [8].

Grid MDP는 상태 (State), 행동 (Action), 보상 (Reward), 상태 전이 확률 (State Transition Probability)로 구성된다. 상태는 각 위성의 위치와 해당 위성에 저장된 패킷의 정보를 나타내며, 행동은 각 위성에서 패킷을 전송하는 방향을 선택하는 것이다. 보상은 패킷 전송에 성공했을 때 얻게 되는 가치를 나타내며, 상태 전이 확률은 패킷 전송에 따른 위치의 변화를 나타낸다.

Grid MDP는 일반적인 MDP와 다르게 상태 공간이 그리드 형태로 구성되어 있어서, 상태 전이와 보상 계산이 비교적 간단하다. 하지만 그리드 형태로 구성되어 있어서 상태 공간이 매우 크고, 각 상태에서 가능한 행동도 다양하므로 이 문제를 효율적으로 해결하기 위해서는 강화 학습 알고리즘을 적용해야 한다.

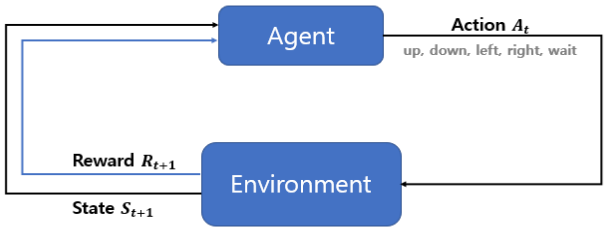


그림 1. MDP 과정 도식화
Fig. 1. Schematic illustration of the MDP process

Grid 환경에서 MDP를 활용한 방법은 상태, 행동, 보상, 상태 전이 확률 등을 미리 모델링하여 최적의 정책을 찾을 수 있다. 이 방법은 전체 상태 공간을 미리 알고 있어야 하며, 각 상태에서 가능한 모든 행동과 보상에 대한 정보가 미리 정의되어야 한다. 또한 상태 전이 확률 또한 정의되어야 하므로, 모델링에 많은 노력과 시간이 필요하다. 강화 학습을 활용한 방법은 모델링을 하지 않고 데이터를 통해 학습하는 방법이다. 이 방법은 환경에 대한 사전 정보를 필요로 하지 않으며, 상태 전이 확률도 따로 정의할 필요가 없다. 대신, 상태와 보상 정보를 실시간으로 수집하고, 이를 기반으로 강화 학습 알고리즘이 최적의 정책을 찾아간다. 이 방법은 초기에는 학습 시간이 길어지고, 학습 결과가 불안정할 수 있으나, 학습이 진행됨에 따라 점차 높은 성능을 보이게 된다. 따라서 MDP를 활용한 방법은 모델링의 노력과 시간이 많이 소요되지만, 확실한 정책을 찾을 수 있는 반면, 강화 학습을 활용한 방법은 모델링의 노력이 적게 들어가지만, 초기 학습 시간이 길고, 학습 결과가 불안정할 수 있다. 하지만 본 논문에서 해결하고자 하는 동적 환경에서의 위성 링크 단절 상황을 고려한 라우팅 알고리즘에는 강화 학습의 이러한 대처 능력이 적합하다고 판단했다. 그림 1은 본 논문의 MDP 과정을 도식화 한 그림이다. Action으로 총 5개의 행동을 할 수 있는 것을 볼 수 있다.

2. 기존 강화 학습 기반 라우팅에 관한 기존 연구의 한계

강화 학습 기반 라우팅 알고리즘에서 중요한 것은 환경을 어떻게 구성하는지에 대한 것이다. 그중에서도 환경을 동적으로 설계할지, 정적으로 설계할지에 관한 것과 장애물의 여부에 대한 것은 매우 중요한 문제이다.

2.1 위성의 링크 단절 문제

본 논문에서는 위성의 링크 단절에 대한 것을 고려하여 설계하였지만, 대다수의 LEO-SN 라우팅 알고리즘에서는 위성의 링크 단절을 고려하지 않고 설계되었다 [1, 3, 4]. 위성 간 링크 단절 상황이 일시적으로 발생한다면 패킷 손실을 감수하고 재전송은 의 경우 재전송 또는 라우팅 변경 등의 방법을 통해 해결할 수 있다. 하지만 위성 간 링크 단절

상황이 일시적이지 않고 특정 주기를 가지고 반복되거나 무작위로 나타날 경우에는 과도한 패킷 손실을 방지하기 위한 향상된 라우팅 기법이 필요하다. 본 논문에서는 특히 위성 간 링크 단절 상황이 일시적이지 않고 특정 주기를 가지고 반복되는 동적 네트워크 상황에서의 라우팅 문제를 다룬다. 위성에서 라우팅 알고리즘을 설계할 때, 위성 간의 링크 단절 현상을 고려해야 하는 이유는 다음과 같다.

첫째, 외부 요인에 의한 단절을 고려하여야 한다. 위성간 링크 단절은 빈번하게 발생할 수 있다. 위성들 사이의 링크는 대기 오염, 자기장 등의 영향으로 인해 빈번하게 단절될 수 있다. 이러한 상황에서도 효율적인 통신이 이루어질 수 있도록 라우팅 알고리즘은 링크 단절 상황에 대한 대응 능력을 가지고 있어야 한다. 둘째, 데이터 정보는 반드시 유지되어야 한다. 링크 단절 상황에서도 데이터 전송을 유지해야 한다. 링크 단절이 발생하면, 데이터 전송이 중단되고 전송 중이던 패킷은 손실될 가능성이 있다. 따라서 라우팅 알고리즘은 이러한 상황에서도 데이터 전송을 유지하고, 손실을 최소화할 수 있는 경로를 찾아내야 한다. 마지막으로, 전체 위성군의 신뢰성을 확보하여야 한다. 링크 단절 상황에서도 효율적인 통신을 유지할 수 있는 라우팅 알고리즘은 위성군의 신뢰성을 높여주고, 효율적인 데이터 전송을 가능하게 한다. 따라서 이러한 상황을 고려하지 않은 라우팅 알고리즘은 위성군의 전체적인 성능에 부정적인 영향을 미칠 수 있다. 또한 신뢰성이 떨어진다는 것은 전체 프로세스에 큰 악영향을 미칠 수 있다. 위의 3가지 이유에 따라서, 링크 단절 상황을 고려하지 않은 라우팅 알고리즘은 위성군에서 효과적인 데이터 전송을 보장할 수 없다. 이를 위해 링크 단절 상황에 대한 대응 능력을 가지고 있는 라우팅 알고리즘을 설계해야 한다.

2.2 위성의 동적 환경 문제

저궤도 위성군은 우주에서 통신, 탐사, 관측 등의 특정 목적으로 운용되면서 매우 높은 속도로 움직이면서 궤도 및 높이에 변화가 생길 수 있다. 이러한 예측하지 못한 변화로 인해 위선 간 링크가 단절될 수 있다. 즉, 예측하지 못한 위성 간 링크 단절 상황으로 인해 네트워크 토폴로지가 정적이지 못하고 시간이 지남에 따라 변경되는 동적 네트워크 환경이 된다. 따라서 저궤도 위성 네트워크에서는 보다 안정적이로 효율적인 통신 상태 유지를 위해, 동적 네트워크 상황을 고려한 라우팅 알고리즘이 설계될 필요가 있다.

Ⅲ. 제안하는 기법

1. Random Reward Model

본 논문에서는 강화학습의 보상함수를 표현하기 위해 Random Reward Model을 제시한다. 강화학습에서는 보상함수 (Reward function)을 통해 에이전트가 특정 행동을 하였을 때, 얼마나 좋은 결과를 얻었는지를 측정한다. 하지만, 보상 함수가 정확하게 정의되어있지 않거나, 행동의 결과가 불확실한 경우에는 학습이 어렵다. 이러한 문제를 해결하기 위해 본 논문에서는 Random Reward Model을 사용하였다.

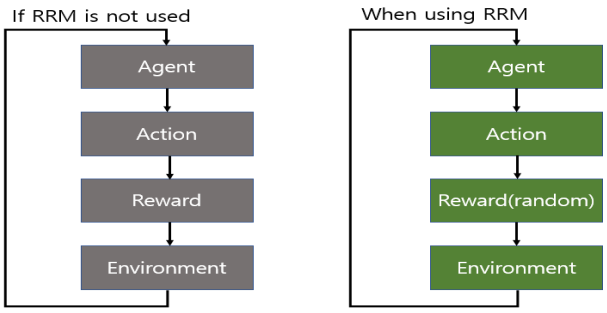


그림 2. RRM 사용 여부에 따른 순서도
Fig. 2. Flowchart with or without RRM

Random Reward Model (RRM)은 보상 함수를 랜덤하게 만들어 에이전트에게 예측할 수 없는 보상을 제공한다. 이를 통해 에이전트는 다양한 행동을 시도하고, 예측하지 못한 보상을 얻을 때마다 학습을 진행한다. 이러한 방식으로 에이전트는 다양한 상황에서 최적의 행동을 찾아갈 수 있다 [9].

RRM은 DQN과 같은 값 기반 강화학습 알고리즘에서 주로 사용된다. DQN에서는 큐 함수 (Q-function)를 학습하는데, 이 함수는 state-action 쌍에 대한 기댓값을 나타내며, 이를 통해 최적의 행동을 선택한다. RRM을 적용한 DQN은 보상 함수가 랜덤하게 변하므로, 큐 함수도 예측이 어려워진다. 이를 통해 DQN이 더욱 강건하게 학습될 수 있다.

우리가 목표로 하는 저궤도 위성군에서의 동적 환경에서 RRM을 사용하게 되면 다양한 환경적 변화에서 보상을 최대화하기 위해 더 많은 정보를 수집하고, 불확실성을 감안하여 더욱 안정적인 행동을 취할 수 있게 한다. 이를 통해 보상의 불확실성을 고려하면서 좋은 결과를 얻을 수 있다.

또한, RRM은 특정 상황에서 에이전트가 지나치게 적응하는 것을 방지하는 역할도 한다. 만약 보상이 고정되어 있다면 에이전트가 그 보상을 최대화 하기 위해 학습을 하다가, 그쪽으로만 결과가 편향되어 좋은 결과를 얻지 못 할 수도 있다. 하지만, RRM을 사용하면 에이전트는 학습을 통해 여러 상황에서 최적의 행동을 결정할 수 있게 된다.

따라서, RRM은 강화학습 문제에서 중요한 역할을 한다. 보상이 불확실하거나 랜덤하게 주어지는 경우에 이를 고려하여 안정적이고 최적의 행동을 결정할 수 있도록 한다. 그림 2는 RRM을 사용하지 않았을 때와 사용하였을 때의 차이를 나타낸 그림이다. RRM을 사용하면 보상이 고정된 값을 가지는 것이 아니라 확률 분포를 따르기 때문에, 각 상태에서 선택된 행동에 대한 보상값이 매번 다를 수 있다. 이에 따라, 각 상태에서 선택된 행동에 대한 보상값을 저장하고, 정책을 업데이트 한다.

2. Grid MDP 기반 동적 네트워크 모델

본 논문에서는 저궤도 위성군 라우팅 알고리즘의 적용을 위해 Grid MDP 기반 동적 네트워크 모델을 적용하였다.

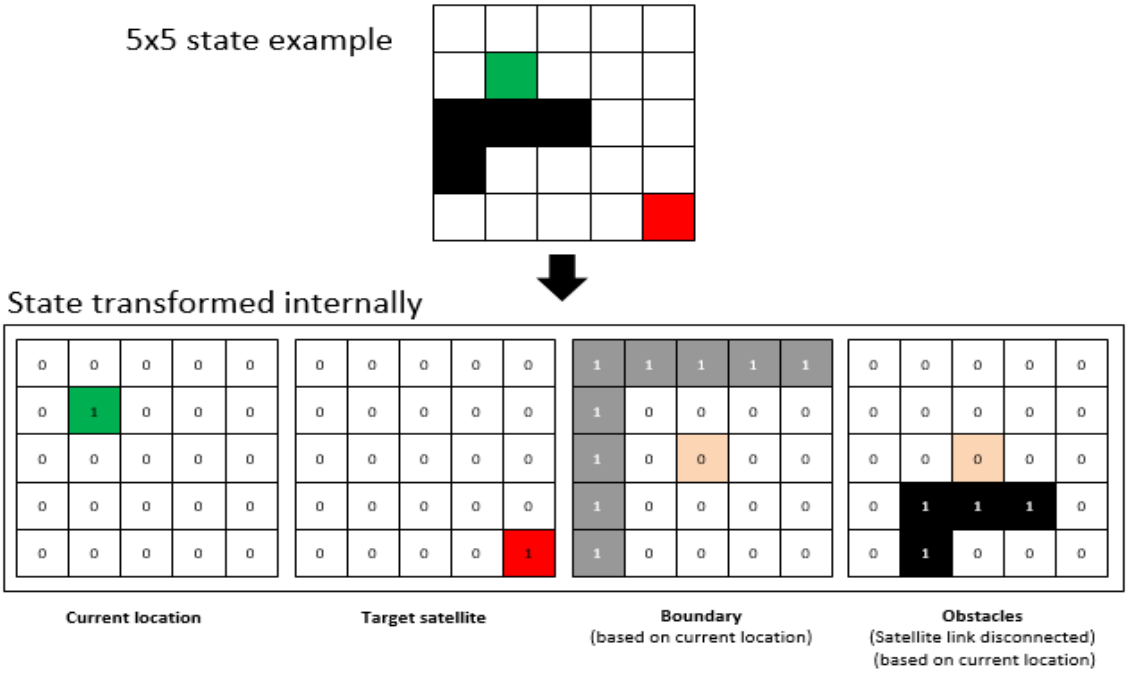


그림 3. 5x5 state를 예로 든 각 채널 변환 방식
Fig. 3. Each channel conversion method using 5x5 state as an example

Grid MDP 기반 동적 네트워크 모델은, 마르코프 결정 과정 및 그리드와 같은 이산형 공간을 사용하여 동적 네트워크 시스템을 모델링하는 방법이다. 이 모델은 동적 시스템에서의 상호작용과 변화를 모델링하며, 최적 제어와 같은 응용 분야에서 활용될 수 있다 [10]. Grid MDP는 이산적인 상태 공간과 액션 공간을 갖는 마르코프 결정 과정이다. 이 모델에서는 2D 그리드를 사용하여 에이전트의 현재 위치, 목표하고자 하는 위치, 경계면, 장애물 등의 정보를 표현한다. 동적 네트워크 모델링에서는 네트워크 토폴로지가 변할 수 있으며, 이러한 변화를 표현하기 위해 시간, 즉 time step마다 변화하는 환경을 표현하였다. 이 모델에서는 네트워크 토폴로지의 변화가 발생할 때마다, 새로운 상태 및 액션 공간을 계산하고, 최적 의사결정 전략을 다시 계산하여 최적 제어 문제를 해결한다.

그림 3은 본 논문에서 제시하는 Grid MDP 기반 동적 네트워크 모델이다. 해당 모델은 15x15 크기의 격자 모양 맵을 기반으로 하며, 각 채널은 다양한 정보를 포함하고 있다. 먼저, 앞의 두 채널은 현재 위치, 목표 위치를 나타내며 이는 에이전트가 이동하는 데 필요한 정보이다. 이 두 채널은 15x15 환경에서 1로 표현되고 나머지 부분은 0으로 표현된다. 다음은 경계면 정보와 장애물 정보이다. 이 두 정보는 현재 위치를 중심으로 구성되게 된다. 장애물은 시뮬레이션 환경에서 위성 링크 단절을 의미한다.

시뮬레이션을 함에 있어서, 동적 환경을 표현하기 위해 위성을 1 time step 마다 우측에서 좌측으로 한 칸씩 이동시켰다. 이렇게 움직이는 환경들 속에서 에이전트는 목표하는 위성까지 장애물들을 최대한 회피하며 이동하여야 한다.

이 환경에서의 Action은 상, 하, 좌, 우, 대기의 총 5가지 동작으로 구분된다. Reward는 목표 위성에 도달 하였을 경우 +1점의 Reward를 부여하고 그렇지 못했을 때는 -1점을 Reward로 부여한다.

3. Dueling DQN 기반 학습

본 논문에서는 저궤도 위성군 라우팅 알고리즘 설계를 위해 Dueling DQN을 적용하였다. Dueling DQN은 DQN과 같이 Q-learning 알고리즘을 기반으로 하며, 신경망 기반의 알고리즘이다. DQN과 Dueling DQN은 몇 가지 차이점이 존재한다.

Dueling DQN은 Q-value를 두 개의 값 함수로 분리한다. 하나는 상태의 가치 (value)를 나타내는 state-value 함수이고, 다른 하나는 각 행동의 상대적 가치 (advantage)를 나타내는 action-value 함수이다. 이를 통해, 에이전트가 특정 상태에서 어떤 행동을 선택할지 결정할 때, 가치와 상대적 가치를 결합하여 최종 Q-value 값을 계산할 수 있다. 그림 4는 Q-network와 Dueling Q-network의 차이를 나타내는 그림이다 [11]. 신경망에서 State-value와 Advantage로 나뉘는 모습을 볼 수 있다.

두 번째로, Dueling DQN은 Q-value를 계산하는 과정에서 advantage 함수를 통해 각 행동에 대한 가치를 독립적으로 추정할 수 있다. 이를 통해 에이전트는 특정 행동이 다른 행동에 비해 더 좋은지, 나쁜지를 더 정확하게 판단할 수 있다.

마지막으로 Dueling DQN은 Q-value를 추정할 때 행동의 수에 따라 연산량이 증가하는 문제를 해결할 수 있다. 일반적인 DQN은 모든 행동에 대해 동일한 신경망을 사용하여

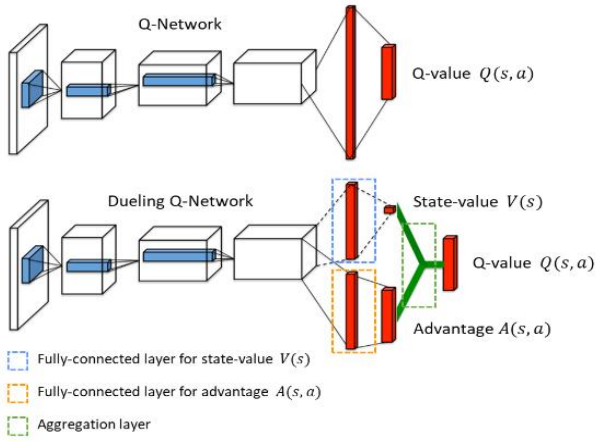


그림 4. Q-network와 Dueling Q-network 도식화
Fig. 4. Illustration of Q-network and Dueling Q-network

Q-value를 추정한다. 따라서 행동의 수가 많아질수록 계산량이 증가하게 된다. 그러나 Dueling DQN은 value와 advantage를 각각 추정하기 때문에, 행동의 수와 관계없이 계산량이 일정하게 유지된다.

우리가 제시하는 방법에서 DQN이 아닌 Dueling DQN을 사용한 이유는 앞서 설명한 장점과 더불어 FPGA에서의 사용을 위해 설계되었다. 강화학습에서 state를 advantage layer와 value layer의 두 가지 방향으로 나누어 FPGA의 PL (Programmable Logic) 영역에서 독립적으로 연산하기 위해서 Dueling DQN을 사용하였다.

본 논문에서 쓰인 모델에서는 state-value 함수를 직접 학습시키는 것이 아니라 간접적으로 학습시킨다. 이 때, Q값을 구성하는 두 가지 요소인 즉각적인 보상과 다음 상태의 가치 중에서, 즉각적인 보상을 학습시키는 것은 어렵지 않지만, 다음 상태의 가치를 학습시키는 것은 어렵기 때문에 이를 근사화하기 위한 두 가지 신경망 모델을 사용한다. Advantage Layer는 현재 상태 s에서 가능한 모든 행동 a에 대한 가치 차이를 계산한다. 이 가치 차이는 현재 상태에서 다른 행동보다 해당 행동이 얼마나 좋은지를 나타낸다. 예를 들어, 현재 상태에서 행동 1의 가치가 3이고, 행동 2의 가치가 2이면, 행동 1의 advantage는 1이 된다.

Value Layer는 현재 상태의 가치를 계산한다. 즉, 현재 상태에서 가능한 모든 행동의 가치 차이를 평균 내어, 현재 상태의 가치를 추정한다. 이는 현재 상태에서 가능한 모든 행동에 대해 평균을 내기 때문에 모든 행동에 대한 가치를 골고루 학습할 수 있다. 따라서 이 두 값의 합이 DQN에서의 Q값에 해당하게 된다.

이러한 방법들의 채택으로 인해 Dueling DQN은 DQN에 비해 보다 안정적인 학습을 보일 수 있고 상대적으로 높은 성능을 보여주기 때문에 논문에서 제시하는 저궤도 위성군 라우팅 알고리즘에 채택되었다.

본 논문에서 제시하는 Dueling DQN의 진행 과정은 다음과 같다. 우선 input으로 15x15x4의 데이터를 입력한다. 여기서 15x15는 배열의 크기, 즉 환경의 크기를 의미하며 이는 각각 환경 내에서 현재 위치, 목표로 하는 위성의 위치, 경계면, 장애물 (위성 링크 단절)을 의미한다. 이렇게 가공된 데이터는 Dueling DQN 클래스 안에서 advantage와 value값으로 나뉘게 된다. 먼저 convolution을 통해 4x15x15의 데이터가 4x4의 크기를 가지는 커널을 통과하여 16x12x12의 데이터가 된다. 이 데이터를 다시 4x4의 크기를 가지는 커널을 통과시켜 32x9x9의 데이터로 만들고, 그 다음부터는 2x2 크기의 커널을 총 두 번 통과시킨다. 그렇게 되면 최종적으로는 128*7*7의 사이즈를 가지는 데이터가 나오게 된다. 이렇게 가공한 데이터를 Affine 과정을 두 번 진행시킨 뒤 각각 advantage layer와 value layer에 입력하여 데이터를 가공한다. 두 과정 모두 Affine을 두 번씩 더 진행하고 이 과정에서 나온 값을 조합하여 Q값을 도출한다. 학습을 진행 함에 있어서 max timestep은 300으로, max episode는 100으로 정하였다. 우선 max timestep의 경우는 한 회차 동안에 에이전트가 이동할 수 있는 총 횟수를 말하는데, 여기에는 상, 하, 좌, 우는 물론 정지하고 있는 경우도 포함된다. max timestep을 300으로 정한 이유는 15x15의 맵에서 이동할 때 300번을 넘게 움직이는 것은 학습이 실패하였다고 판단하였기 때문이다. 또한 max episode는 100으로 제한하였는데, 이는 실험 결과에서 알 수 있듯이 다른 알고리즘이 아닌 Dueling DQN을 사용한 알고리즘에서는 최적 경로로의 수렴 속도가 매우 빠른 편이라 100회가량 진행하게 되면 온전히 최적 경로에 수렴한 것으로 판단되기 때문이다. 또한 batch size는 1024로 설정하였다. batch가 너무 작으면 모델이 학습 데이터의 일부에만 의존하는 과적합(overfitting)이 발생할 가능성이 높기 때문과 데이터가 많이 바뀌어서 학습 안정도가 떨어지는 것을 고려하여 설정하였다. 반대로 batch의 크기가 너무 크면 메모리 사용량이 늘어나고 이는 학습시간과 추론에 악영향을 미친다.

우리는 학습 불안정성을 유발하는 요인들을 해결하기 위해 Experience replay를 사용하였다. Experience replay의 기본적인 컨셉은 Replay memory라는 버퍼를 하나 만들어 주어서 현재 생성된 샘플을 저장하지 않고 현재 선택된 action을 수행하여 결과 값과 샘플을 얻지만 이를 바로 평가에 사용하지 않고 의도적으로 지연시킨다. 이를 통해 학습 불안정성을 유발하는 두 가지 요인들을 해결할 수 있다. 첫 번째로 Sample correlation을 해결할 수 있는데, 딥러닝에서는 학습 샘플들이 independent하게 추출되었다고 가정하고 모델을 학습한다. 하지만 강화학습에서는 연속된 샘플 사이에 dependency가 존재하는데, 즉 이전 샘플이 다음 샘플이 생성되는 것에 영향을 미치는데 이는 현재 샘플에서의 정책과 state transition probability에 의해 다음 샘플이 생성되기 때문이라고 볼 수 있다 [12]. correlation이 강한 연속된 샘플들로 모델을 학습하게 되면 제대로 된 Q-function의 모양을 찾을 수 없다 [13, 14]. 두 번째로는 Data distribution의 변화인데, 모델을 on-policy로 학습할 경우 Q-update로 인해 생성되는 training data의 분포도 갑자기 변화할 수 있

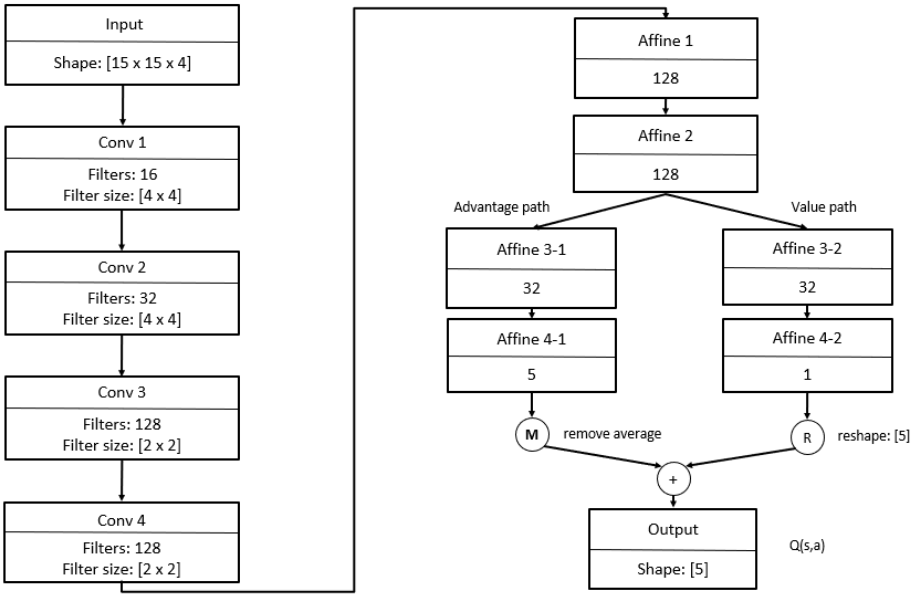


그림 5. 해당 논문의 Dueling DQN 알고리즘 순서도
Fig. 5. Flowchart of the Dueling DQN algorithm in this paper

다. 이러한 data distribution의 갑작스러운 변화는 학습 파라미터의 oscillation을 유발하고 이러한 불안정성으로 인해 알고리즘이 발산하게 만들수도 있다. 따라서 replay memory를 이용해 랜덤하게 추출된 샘플들은 각각 다른 시간에서 수행된 샘플들이므로 sample correlation이 작다 [15]. 또한 랜덤 추출에 의해 Q-function이 다양한 action을 동시에 고려하여 업데이트 되므로 분포가 편향하지 않는 효과를 얻을 수도 있다는 장점이 있다.

그림 5는 Dueling DQN에 대한 Flowchart이다. 앞서 설명했다시피 해당 논문에서의 Dueling DQN은 4번의 Convolution 과정을 거친 뒤 2번의 Affine 과정, 그리고 Advantage와 Value로 나뉘어 각각 두 번의 Affine 과정을 더 거치게 된다.

value function $V(s)$은 현재 state의 가치가 얼마나 되는지를 나타낸다. 이는 앞으로 선택될 action과 관계없이 state 자체의 좋고 나쁨을 나타낸다. advantage function $A(s,a)$은 현재 state에서 해당 행동이 다른 행동에 비해 가지는 상대적인 가치를 뜻한다. 이 advantage function은 Q-value에서 value function을 뺀 값으로 정의된다.

$$A(s,a) = Q(s,a) - V(s). \tag{1}$$

Q-value를 얻기 위해서는 식 (1)에서 얻은 결과를 사용할 수 있다.

$$Q(s,a) = V(s) + A(s,a). \tag{2}$$

하지만 식 (2)처럼 Q-value를 얻기 위해 단순히 value function과 advantage function을 더해주게 되면 문제가 발생한다. 신경망이 학습하는 과정에서는 value와 advantage가 구분되지 않기 때문에 식별 가능성 부족 문제(unidentifiable problem)가 발생할 수 있다. 이 문제를 해결하기 위해 advantage function estimator가 선택한 행동에서 0 advantage를 가지도록 유도할 수 있다.

$$Q(s,a) = V(s) + (A(s,a) - \max_{a' \in |A|} A(s,a')). \tag{3}$$

advantage function의 정의에 따라 $\max_{a' \in |A|} A(s,a')$는 항상 0을 만족한다. 하지만 실제로 학습 시 식 (3)은 타당한 식은 아니기 때문에 대체 모듈을 사용한다.

$$Q(s,a) = V(s) + (A(s,a) - \frac{1}{|A|}\sum_{a} A(s,a')). \tag{4}$$

식 (4)는 max operator를 avarage로 대체한 식이다. 최대값 대신 평균을 사용하면 advantage의 변화량이 적어지기 때문에 학습 안정성이 올라갈 수 있다.

Ⅳ. 실험 결과

1. 저궤도 위성군 라우팅 알고리즘 구현을 위한 환경 설정

1.1 구현 환경

본 논문에서 제시하는 라우팅 알고리즘은 CPU와 GPU가 함께 사용된 PC를 사용하였다. 세부적으로는 Intel core i7-10700 및 Nvidia Geforce RTX 3070을 사용하였다. 시뮬

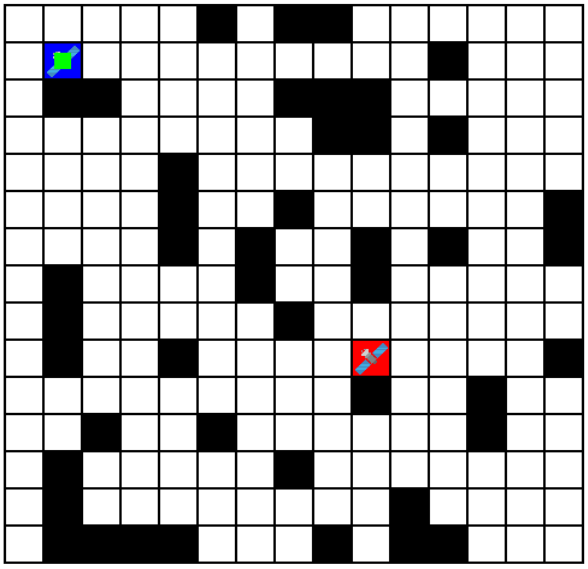


그림 6. 저궤도 위성군 환경 구성
Fig. 6. Configuring the Low-Earth-Orbit Satellite Network Environment

레이션은 MATLAB (ver 2022)을 사용하여 진행되었다.

알고리즘 간의 차이를 확인하기 위해 DQN으로 구현한 환경과 Dueling DQN으로 구현한 환경 간의 학습 차이를 포함하며 다른 비교군으로는 Actor-Critic 알고리즘과 PPO 알고리즘을 비교하였다. 또한 장애물의 개수와 출발지, 목적지 변화에 따른 실험 결과도 함께 기록하였다. 학습은 MATLAB 기반으로 개발하였지만, 임베디드 시스템 적용을 위해 python을 활용해 설계해서 사용하였다.

1.2 동적 장애물을 포함한 환경

본 논문에서 제시하는 동적 장애물을 포함한 학습 환경을 만들기 위해 15x15 크기의 격자 맵을 만들고, 장애물을 설정하였다. 장애물은 랜덤하게 생성 후 배열에 저장 후 사용하였다. 그림 6은 실제 환경 구성에 사용된 Grid map의 모습이다. 청색 위성은 agent, 적색 위성은 목표 위성을 의미하며 백색은 통신 가능한 위성, 흑색은 위성 링크 단절을 의미한다. 각 장애물은 1 time step 마다 우측에서 좌측으로 한 칸씩 이동하며, 장애물이 경계면에 부딪히면 최우측으로 이동하여 무한하게 반복하는 동적 장애물을 만들었다. 장애물의 개수는 45개이며 위성 링크가 정상적인 위성은 225-45개로 총 180개다. 그 중 시작 위치인 2,2는 에이전트고, 목표로 하는 위성은 10,10에 위치해 있다. 에이전트가 장애물을 완벽하게 회피하고 목표로 하는 지점까지 이동하게 되면 16번의 움직임만으로 목표하는 위성까지 도달할 수 있다.

학습은 총 100번 진행된다. 각 회차 당 time step은 300번으로 제한되며, 300번이 넘어가게 되면 현재 진행 중이던 학습이 종료되고 다음 회차로 넘어간다. batch size는 1024로 정해두었는데, 처음 학습 시 최소 batch size만큼 샘플이 모이지 않으면 학습을 진행하지 않는다.

agent의 움직임은 상, 하, 좌, 우, 정지의 5가지로 움직일 수 있으며, 한 time step 후 목표하는 위성에 도착하였을 경우에만 +1의 Reward를 받고 도착하지 못했을 경우에는 -1의 Reward를 부여받게 된다. 또한 agent가 움직이는 동안에 장애물 (위성 링크 단절)을 만나게 되면 agent는 1 time step만큼 움직이지 않게 된다. 결론적으로 agent는 목표하는 위성까지 최대한 빠르게 움직이되, 장애물과 만나는 횟수를 최소한으로 줄여야 좋은 성적을 얻을 수 있다.

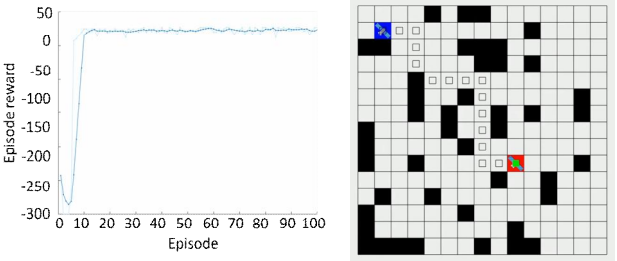
2. 결과 및 분석

2.1 학습 및 추론 결과

다음은 MATLAB을 사용하여 총 100번의 학습을 진행한 후, Dueling DQN 알고리즘을 사용하여 학습한 결과이다. 이 학습에는 총 199초가 소요되었으며 학습 결과, 에이전트가 출발점인 (2, 2)에서 목적지인 (10, 10)까지 이동하는 가장 빠른 경로는 우측으로 8 step, 하단으로 8 step을 이동하여 목적 위성에 도달하는 방법이었다. 이 방법은 총 16 step을 거치며, 이때 에이전트가 얻은 Score는 -14점이다. 이 경로에서는 총 15번의 step 동안 -1점의 Reward를 받아 -15점을 기록하였고, 마지막 16번째 step에서는 +1점의 Reward를 받았다. 처음 3회차는 batch size를 수집하는 동안이었기 때문에, -300점의 점수를 기록하였다. 5회차부터는 점수를 기록하였고, 최종적으로 최적 경로로 판단되는 -14점의 점수를 얻기까지는 39회차의 데이터 수집이 필요하였다. 그림 7은 MATLAB으로 학습시킨 Dueling DQN의 대한 학습정보와 그렇게 추론된 경로이다. (a)는 학습 결과, (b)는 그 때의 에이전트의 경로이다. 파란색 실선은 평균 보상을 의미하는데, 10회차 정도 학습을 하고 난 후 평균 보상이 수렴하는 모습을 보인다. 처음에는 최소 batch size 까지 데이터를 수집하기 때문에 점수가 비교적 낮게 측정되는 경향이 있으며, 학습을 성공한 뒤로는 빠르게 점수가 상승했다.

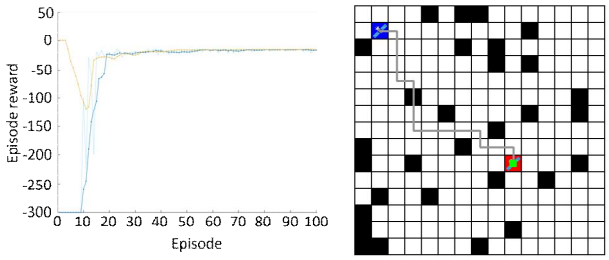
2.2 장애물 개수에 따른 결과 비교 및 분석

이번에는 장애물에 개수에 따른 Dueling DQN의 성능이 어떻게 변화하는지 알아보기 위해 장애물의 개수를 기존 45개에서 30개, 75개로 변화시켰다. 그림 8은 장애물 개수 별 학습 결과 및 환경에 대한 그림이다. (a)와 (c)는 학습 결과이며, (b)와 (d)는 그 때의 경로이다. 장애물의 개수를 30개

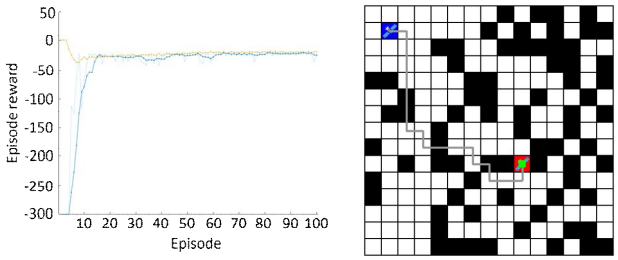


(a) Dueling DQN learning result (b) Dueling DQN path

그림 7. MATLAB을 이용한 Dueling DQN 학습 결과 및 추론 결과
Fig. 7. Dueling DQN learning result and inference result using MATLAB

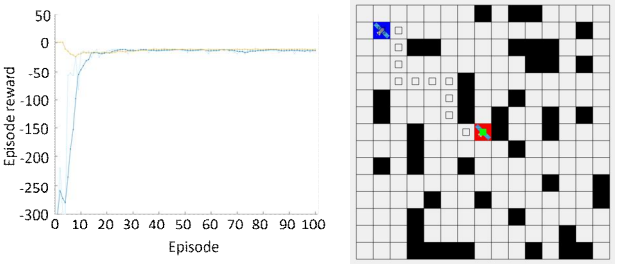


(a) Obstacle 30 learning (b) Obstacle 75 path

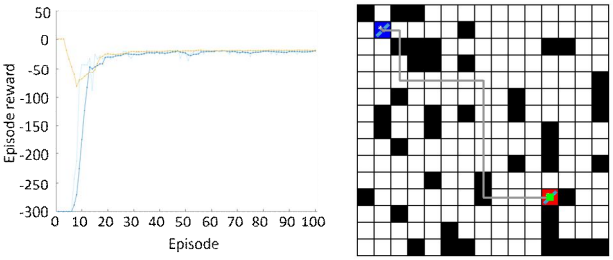


(c) Obstacle 75 learning (d) Obstacle 75 path

그림 8. 장애물 개수 변경 결과 (30 / 75)
Fig. 8. Change the number of obstacles (30 / 75)



(a) 2,2 to 8,8 learning (b) 2,2 to 8,8 path



(c) 2,2 to 12,12 learning (d) 2,2 to 12,12 path

그림 9. 목표 위성과의 거리 변화
Fig. 9. Change in distance to target satellite

로 낮췄을 경우에는 평균 수렴에 더 많은 학습 횟수를 필요로 했다. 그 이유로는 적은 수의 장애물이 오히려 학습을 어렵게 만든다는 것이다. 맵이 간단해졌다고 해서 에이전트가 목적지를 반드시 빨리 찾을 수 있는 것은 아니다. 에이전트가 목적지를 찾기 위해서는 다양한 상황과 환경에서 정확한 행동을 선택해야 하는데, 간단한 환경이라면 학습이 더 오래 걸리고 결과 수렴에 오래 걸릴 수 있다. 또한 목적지를 찾기 위해서는 최적의 경로를 찾는 것이 중요한데, 경로를 찾는 과정에서 불필요한 움직임을 하거나 지나치게 멀리 돌아다닐수도 있다. 이러한 경우 에이전트가 불필요한 경험을 많이 하게 되어 학습이 느려질 수 있다.

75개의 장애물을 사용하였을 때에는 에이전트가 최단 경로로 목적 위성에 도달하는데에 필요한 학습 횟수가 줄어들었다. 이는 몇 가지 이유로 설명할 수 있다. 먼저 다른 장애물이 추가되면서 상태 공간이 더 다양해지면 그로 인해 여러 상태를 학습하기에 좋은 조건이 만들어진다. 이는 Dueling DQN 알고리즘의 학습을 도와준다. 또한 Replay memory로 인해 과거의 저장된 다양한 경험을 무작위로 추출해 학습에 사용하게 되어 장애물이 많아진 상태에서도 다양한 경험이 가능할 수 있도록 도와준다.

2.3 목표물 변경에 따른 결과 비교 및 분석

2.3.1 목표 위성으로의 거리 변화

목표물의 변화에 따라 어떤 변화가 있는지 알아보기 위해 크게 2가지 변화로 나누어 실험을 진행하였다. 그림 9의 (a), (c)는 학습 결과이며, (b), (d)는 그때의 경로이다. 첫 번째는 시작점은 고정한 채 목표 위성의 위치가 기존보다 가깝거나 먼 환경이다. 기존의 2, 2에서 출발하여 목표 위성 위치인 10, 10으로 가는 상황에서 목표 위성 위치를 8, 8과 12, 12로 +-2씩 수정하였다. 그림 9는 목표 위성으로의 거리 변화에 따른 학습 결과 및 환경에 대한 그림이다. 두 상황 모두 최종적으로 추론 된 score만 변화하게 되었고 학습의 수렴 속도에는 변화가 없었다. 이는 즉 목표 위성의 위치와 수렴 속도의 차이에는 거의 영향이 없음을 의미한다.

2.3.2 목표 위성으로의 이동 방향 변화

두 번째는 목표물의 거리는 유지한 채 이동하는 방향이 변화하게 되면 어떤 차이를 보여주는지 확인하기 위한 실험을 진행하였다. 실험한 환경은 기존 2, 2에서 10, 10으로 가는 경로와 반대되는 경로, 그리고 2, 10에서 10, 2로 가는 경로와 그에 반대되는 경로로 실험하였다. 그림 10은 위성 위치에 따른 학습 결과 및 환경에 대한 그림이다. 각각 (a), (c), (e)는 학습 결과이며, (b), (d), (f)는 그 때의 경로이다. 우선 장애물이 우측에서 좌측으로 1칸씩 이동하고 있기 때문에 먼저 기존과 같이 에이전트가 좌측, 목표 위성이 우측인 경우를 먼저 실험하였다. 이 경우에는 평균 보상이 빠르게 수렴하였다. 그러나 뒤에 나오는 두 실험의 경우는 장애물의 이동 방향과 똑같은 방향으로 에이전트가 이동하게 되는데, 두 상황 모두 평균 보상이 비교적 늦게 수렴하였다. 우선 기존의 좌측에서 우측으로 이동 방향보다 반대로 이동하는 경로가 평균 보상이 빠르게 수렴하는 이유는 바로 앞서 얘기하였던 상태 공간의 다양화 때문이다. 좌측에서 우측으로 이동하게 되면 장애물의 이동 방향과는 반대되기 때문에 학습할 수 있는 환경이 더 다양하게 되고 이는 여러 상태를 학습하기에 알맞은 상황이 만들어진다. 반대로 장애

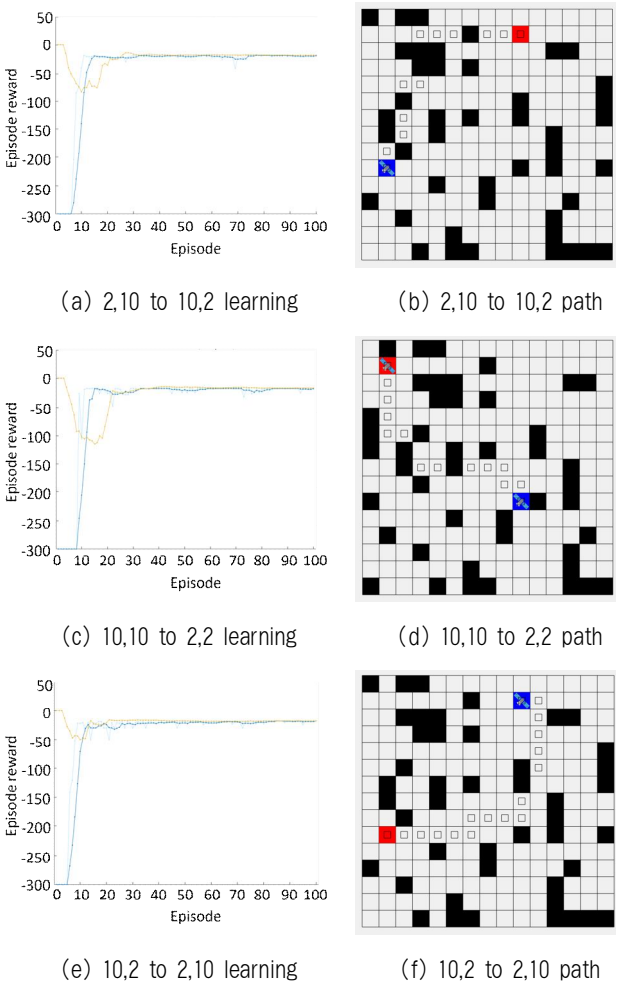


그림 10. 위성 위치에 따른 학습 결과
Fig. 10. Learning outcomes based on satellite location

물과 에이전트의 이동 방향이 같은 경우는 상대적으로 장애물을 만날 확률이 줄어들게 되고 이는 오히려 환경이 단순하게 바뀌었기 때문에 학습이 더 오래 걸리고 불필요한 경험을 더 많이 경험하게 되는 원인이 된다. MATLAB으로 출력한 그래프를 자세히 보면 첫 번째 그래프는 평균 보상이 일정 수치에 도달한 뒤 비교적 완만하게 유지되는 것을 확인할 수 있는데, 그렇지 않은 두 그래프의 경우에는 평균 보상이 초반부에 많이 변화하는 것을 확인할 수 있다.

3. 타 알고리즘과의 비교 결과

본 논문에서는 Dueling DQN의 성능을 평가하기 위해 DQN과 Actor-Critic, PPO를 비교 대상으로 하였다. 우선 Dueling DQN의 바탕이 되는 DQN 알고리즘은 그림 11의 (a)에서 확인할 수 있듯 학습의 수렴 속도에서 차이가 있다. 순정 DQN의 경우 학습이 20회 정도 반복되고 나서 결과가 수렴하는 모습을 보여주는 반면에 (b) Dueling DQN은 학습 10회 정도부터 수렴하는 모습을 보여준다. 하지만 학습 완료 후 최종 시뮬레이션 시 스코어는 유사했으며 이는 즉

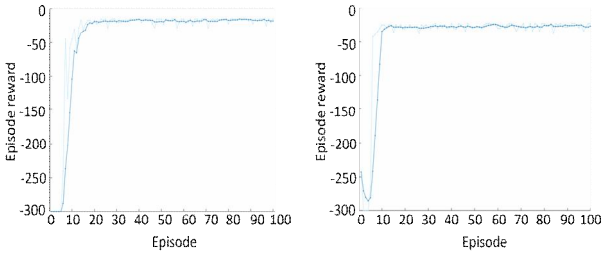


그림 11. DQN과 Dueling DQN의 학습 차이
Fig. 11. DQN and Dueling DQN learning differences

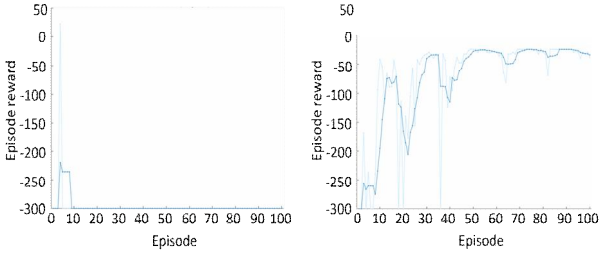


그림 12. Actor-Critic과 PPO 실험 결과
Fig. 12. Actor-Critic and PPO experiment results

DQN과 Dueling DQN의 학습 수렴 속도의 차이만이 존재한다는 것을 의미한다. 다음으로는 Actor-Critic 및 PPO 알고리즘을 사용하여 실험하였다. DQN이나 Dueling DQN과는 다르게 Actor-Critic은 2개의 인공신경망을 사용하는데에 차이가 있다. Actor-Critic에서 Actor Network가 policy를 추정하며, Critic Network가 value를 추정한다. 이 과정을 통해 최종적으로 2개의 값을 통해 agent의 다음 행동을 계산한다 [16]. PPO (Proximal Policy Optimization) 방식은 policy 기반 최적화 방식 중 하나로 TRPO에서 파생된 알고리즘이다. 연산량이 많은 TRPO를 개선하기 위해 목적함수인 surrogate objective function 업데이트 시 Clipping 하여 연산량을 감소시켰다. 또한 first-order optimization 만을 사용하여 구현이 매우 쉬우며 TRPO만큼의 성능을 지니면서 data efficiency 문제등을 해결한 알고리즘이다 [17].

학습 결과는 그림 12과 같다. (a) Actor-Critic의 경우 제대로 된 학습이 이루어지지 않았다. Actor-Critic algorithm은 Actor Network가 policy를 추정하며, Critic Network가 value를 추정하여 최종적으로 2개의 값을 통해 agent의 다음 행동을 추정하는데, 이를 통해 좀 더 정확한 정책 추정을 성공할 수 있지만, 학습이 불안정해질 수 있다. 반면에 Dueling DQN은 Q-value function을 분리하여 학습하므로 상태와 행동에 대한 가치함수의 추정이 더 정확해질 수 있으며, 이를 통해 학습이 더 안정적으로 이루어질 수 있다. (b) PPO의 경우는 DQN 대비 연산속도에서는 우위를 가져

갈 수 있었으나, 결과값에 수렴하는 데에 더 많은 학습을 필요로 하는 단점이 있고 최종 score가 DQN 대비 낮게 측정되었다.

제안된 방법의 결과와 관련하여 다음과 같은 논의점을 생각해 볼 수 있다. 동적 네트워크 라우팅 문제는 만약 기존 최단 경로 기법과 함께 패킷 손실을 감수한 재전송 등과 같은 추가적인 라우팅 전략을 사용함으로써 접근할 수도 있다. 다만, 네트워크가 커질수록 이러한 전략은 복잡해지게 된다. 본 논문에서 제안한 강화학습 기반 라우팅 기법은 라우팅을 수행하는 에이전트가 이미 동적 상황에 대해 학습이 되어 있기 때문에 이러한 동적 상황에서 보다 효율적인 라우팅을 수행할 수 있다. 다만, 동적 상황이 과도하게 불규칙해지게 되면 기존의 최단 경로 알고리즘과 같이 추가적인 라우팅 전략이 필요할 수 있다. 따라서 실제 저궤도 위성 네트워크에서 라우팅 문제를 효율적으로 수행하기 위해서는, 네트워크 상태에 따라 기존의 최단 경로 알고리즘과 강화학습 기반 라우팅 방법을 적응적으로 선택해서 사용하는 것과 같은 실용적인 방법이 필요하다.

V. 결 론

본 논문에서는 동적 저궤도 위성 네트워크에서 라우팅 문제를 해결하기 위해 Dueling DQN 기반 라우팅 기법을 제안하였다. Dueling DQN을 사용하는 알고리즘의 경우 목표 점수로의 수렴이 타 알고리즘 대비 빠른 것을 확인할 수 있었으며 실험 결과 또한 가장 좋게 나온 것을 확인할 수 있다. Dueling DQN을 사용한 알고리즘은 평균 보상을 높게 가져갈 수 있다고 기대할 수 있으며 위성 링크 단절의 위치나 개수가 변화하는 환경, 출발점과 도착점의 위치를 유동적으로 조절하는 환경 속에서도 우수한 성적을 보였다. 학습 및 비교를 위해 본 논문에서는 학습을 100회 진행하였지만 실제로는 그의 절반인 50회의 학습만 진행해도 우리가 원하는 최단 경로를 얻을 수 있었다. Dueling DQN을 사용했을 때 가장 좋은 점은 환경 변화에 강인하다는 점에 착안하였고 실험 결과 역시 우수한 결과를 나타내었다. 향후에는 위성 간 링크 단절 상황이 무작위로 발생하는 상황에서의 라우팅 문제를 해결하고자 한다. 또한 제안된 방법이 실제 시스템에 적용될 때 발생할 수 있는 데이터 전송 관련 이슈들을 해결하고자 한다.

References

[1] B. S. Roh, M. H. Han, D. W. Kum, K. S. Jeon, “A Study on the Reinforcement Learning Routing for LEO Satellite Network”, Proceedings of the Korean Institute of Communication Sciences Conference, pp.537-538, 2022.

[2] J. H. Lee, Y. C. Ko, “A Study on the Low-earth Orbit Satellite Based Non-terrestrial Network Systems Via Deep-reinforcement Learning”, Proceedings of the Korean Institute of Communication Sciences Conference, pp.1306-1307, 2021.

[3] X. Wang, Z. Dai, Z. Xu. "LEO Satellite Network Routing Algorithm Based on Reinforcement Learning." In 2021 IEEE 4th International Conference on Electronics Technology (ICET), pp. 1105-1109. IEEE, 2021.

[4] P. Zuo, C. Wang, Z. Yao, S. Hou, H. Jiang. "An Intelligent Routing Algorithm for Leo Satellites Based on Deep Reinforcement Learning." In 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall), pp. 1-5. IEEE, 2021.

[5] A. Pattanaik, Z. Tang, S. Liu, G. Bommannan, G. Chowdhary. "Robust Deep Reinforcement Learning with Adversarial Attacks." arXiv preprint arXiv:1712.03632 (2017).

[6] A. Cigliano, F. Zampognaro. "A Machine Learning Approach for Routing in Satellite Mega-Constellations." In 2020 International Symposium on Advanced Electrical and Communication Technologies (ISAECT), pp. 1-6. IEEE, 2020.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller. "Playing Atari with Deep Reinforcement Learning." arXiv preprint arXiv:1312.5602 (2013).

[8] R. S. Sutton, A. G. Barto, "Reinforcement Learning: An Introduction." MIT press, 2018.

[9] Y. Burda, H. Edwards, A. Storkey, O. Klimov "Exploration by Random Network Distillation." arXiv preprint arXiv:1810.12894 (2018).

[10] W. Zhao, J. P. Queralta, T. Westerlund. "Sim-to-real Transfer in Deep Reinforcement Learning for Robotics: a Survey." In 2020 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 737-744. IEEE, 2020.

[11] Z. Wang, T. Schaul, M. Hessel, H. V. Hasselt, M. Lanctot, Na. de Freitas, "Dueling Network Architectures for Deep Reinforcement Learning." In International Conference on Machine Learning, pp. 1995-2003. PMLR, 2016.

[12] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, "Human-level Control Through Deep Reinforcement Learning." Nature 518, No. 7540, pp. 529-533, 2015.

[13] J. H. Baek, H. T. Oh, S. J. Lee, S. H. Kim. "A Study about Application of Indoor Autonomous Driving for Obstacle Avoidance Using Atari Deep Q Network Model." In Proceedings of the Korea Information Processing Society Conference, pp. 715-718. Korea Information Processing Society, 2018.

[14] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra. "Continuous Control with Deep Reinforcement Learning." arXiv preprint arXiv:1509.02971 (2015).

[15] T. Schaul, J. Quan, I. Antonoglou, D. Silver. "Prioritized Experience Replay." arXiv preprint arXiv:1511.05952 (2015).

[16] V. Konda, J. Tsitsiklis. "Actor-critic Algorithms." Advances in Neural Information Processing Systems 12 (1999).

[17] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov. "Proximal Policy Optimization Algorithms." arXiv preprint arXiv:1707.06347 (2017).

Dohyung Kim (김 도 형)

2021~Department of IT Convergence Engineering, School of Electronic Engineering, Kumoh National Institute of Technology (B.S.)

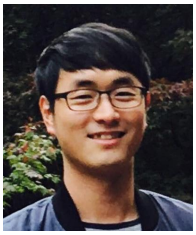
Field of Interests: Algorithm acceleration with GPU and FPGA, FPGA-based acceleration of reinforcement learning algorithms.
Email: askalif@kumoh.ac.kr

Sanghyeon Lee (이 상 현)

2021 Department of IT Convergence Engineering, School of Electronic Engineering, Kumoh National Institute of Technology (B.S.)

Career:
2013 Researcher, Research Institute of Manufacturing Technology, Kumoh National Institute of Technology
Field of Interests: Reinforcement learning
Email: freeg159@gmail.com

Heoncheol Lee (이 헌 철)



2006 Electronic-Electrical Engineering and Computer Sciences from Kyungpook National University (B.S.)
2008 Electrical Engineering and Computer Sciences from Seoul National University (M.S.)
2013 Electrical Engineering and Computer Sciences from Seoul National University (Ph.D.)
2019~Department of IT Convergence Engineering, School of Electronic Engineering, Kumoh National Institute of Technology (Assist. Prof.)

Career:
2011 Researcher, ASRI, Seoul National University
2013 Senior Researcher, Agency for Defense Development
2019 Technical Adviser, LG Electronics
Field of Interests: Vision-based Anomaly Detection, SLAM, Path Planning, Algorithm Acceleration, Deep Learning
Email: hclee@kumoh.ac.kr

Dongshik Won (원 동 식)

2011 Electrical Engineering from University of Minnesota, Twin-Cities (B.E.E.)
2014 Electrical and Electronic Engineering from Korea Advanced Institute of Science and Technology (M.S.)
2019~Aerospace Engineering from Korea Advanced Institute of Science and Technology (Ph.D. Candidate)

Career:
2013 Research Fellow, Institut und Lehrstuhl für Integrierte Photonik RWTH Aachen
2014 Senior Researcher, Agency for Defense Development
2021 Director of Engineering (Data), Glassdome, Inc
2023 Director, TelePIX Co., Ltd
Field of Interests: Satellite Mission Analysis and Design, Electro-Optical Infrared Systems, Tracking and Navigation, Machine Learning for Space Mission Systems, Reinforcement Learning
Email: dw@telepix.net