JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 26 No. 3, pp. 9-17, March 2021
https://doi.org/10.9708/jksci.2021.26.03.009

A Study of Collaborative and Distributed Multi-agent Path-planning using Reinforcement Learning

Min-Suk Kim*

*Professor, Dept. of Human Intelligence and Robot Engineering, Sangmyung University, Cheonan, Korea

[Abstract]

In this paper, an autonomous multi-agent path planning using reinforcement learning for monitoring of infrastructures and resources in a computationally distributed system was proposed. Reinforcement-learning-based multi-agent exploratory system in a distributed node enable to evaluate a cumulative reward every action and to provide the optimized knowledge for next available action repeatedly by learning process according to a learning policy. Here, the proposed methods were presented by (a) approach of dynamics-based motion constraints multi-agent path-planning to reduce smaller agent steps toward the given destination(goal), where these agents are able to geographically explore on the environment with initial random-trials versus optimal-trials, (b) approach using agent sub-goal selection to provide more efficient agent exploration(path-planning) to reach the final destination(goal), and (c) approach of reinforcement learning schemes by using the proposed autonomous and asynchronous triggering of agent exploratory phases.

▸ **Key words**: Reinforcement Learning, Multi-agent, Sub-goal, Sharing Information, Collaborative

[요 약]

동적 시스템 환경에서 지능형 협업 자율 시스템을 위한 기계학습 기반의 다양한 방법들이 연구 및 개발되고 있다. 본 연구에서는 분산 노드 기반 컴퓨팅 방식의 자율형 다중 에이전트 경로 탐색 방법을 제안하고 있으며, 지능형 학습을 통한 시스템 최적화를 위해 강화학습 방법을 적용하여 다양한 실험을 진행하였다. 강화학습 기반의 다중 에이전트 시스템은 에이전트의 연속된 행동에 따른 누적 보상을 평가하고 이를 학습하여 정책을 개선하는 지능형 최적화 기계학습 방법이다. 본 연구에서 제안한 방법은 강화학습 기반 다중 에이전트 최적화 경로 탐색 성능을 높이기 위해 학습 초기 경로 탐색 방법을 개선한 최적화 방법을 제안하고 있다. 또한, 분산된 다중 목표를 구성하여 에이전트간 정보 공유를 이용한 학습 최적화를 시도하였으며, 비동기식 에이전트 경로 탐색 기능을 추가하여 실제 분산 환경 시스템에서 일어날 수 있는 다양한 문제점 및 한계점에 대한 솔루션을 제안하고자 한다.

▸ **주제어**: 다중 멀티 에이전트, 강화학습, 분산 컴퓨팅, 학습 정책, 정보 공유, 인공지능

• First Author: Min-Suk Kim, Corresponding Author: Min-Suk Kim
*Min-Suk Kim (minsuk.kim@smu.ac.kr), Dept. of Human Intelligence and Robot Engineering, Sangmyung University
• Received: 2021. 02. 10, Revised: 2021. 03. 09, Accepted: 2021. 03. 11.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

교통, 항공, 위성 시스템에서 발생하는 대규모 Critical Infrastructure and Key Resource(CIKR)를 단일 시스템에서 모두 처리하기에는 충분하지 않기 때문에 이러한 문제를 해결하기 위한 많은 응용 어플리케이션들이 연구개발되고 있다. 다중 에이전트(Multi-agent)를 이용한 분산형 학습 시스템(Distributed Learning System) 또한 이러한 문제점을 해결하기 위해 사용되는 대표적인 응용 방법 중 하나이다. 에이전트는 제한적이고 불완전한 대규모 분산 환경에서 지식을 습득하고 학습하여 최적화를 만들기 때문에 이 방법을 이용하여 지능형 다중 에이전트 시스템을 구성하고 문제를 해결하는 솔루션 개발들이 활발히 이뤄지고 있다. 다중 에이전트 시스템은 컴퓨팅 노드의 분산 메모리에 부분적으로 상주하는 환경 로컬 정보(Local Information)만 접근할 수 있으며, 필요한 경우에만 통신 네트워크를 통해 나머지 환경 정보를 공유할 수 있다. 따라서 에이전트는 제한적이고 불완전한 분산 메모리 설정을 사용하기 위해서는 매우 계산적인 분산 지능형 협업(Collaborative) 기반 다중 에이전트 시스템을 구성해야 한다.

분산 다중 에이전트 시스템은 에이전트가 주어진 로컬 시스템에서 사용할 수 없는 데이터를 시스템에 요구하면서 데이터 전송 오버헤드(overhead)와 같은 문제들이 발생할 수 있다. 이때 요구되는 데이터는 다른 협업 에이전트로부터 학습된 데이터이거나 별도의 분산 컴퓨팅 및 메모리 노드에서 학습된 데이터일 수 있다. 또한, 위와 같은 오버헤드는 데이터의 인식이나 스케줄링 감소에도 영향을 미칠 수 있으므로 이를 해결하기 위해 전송된 데이터 양을 최소화하여 각각의 분산 노드에 효율적으로 할당하는 것이 매우 중요하다[1].

시간 변화에 따른 동적 환경(Dynamic Time-varying Environment)에서 다중 에이전트 최적화 문제를 해결하기 위해 강화학습(Reinforcement Learning) 방법을 제안하고 있다[2]. 이는 다중 에이전트 시스템에서 에이전트가 시간 변화에 따른 환경 정보를 학습하여 주어진 목표에 최단 경로로 도달하게 하는 방법이며, 각각의 에이전트는 정보를 공유하는 방식(Sharing Information scheme)으로 분산 시스템 및 네트워크의 다중 에이전트 연결성과 수렴 안정성을 확보하여 에이전트의 학습 속도를 개선할 수 있다[3]. 이때 공유되는 정보는 환경 요소에 대한 인식, 표현 및 커뮤니케이션에 따라 경로 탐색(Path Planning)의 사용 가능한 자원과 기술 등을 수반하고 있으며, 이러한 정보로 인해 에이전트의 경로 탐색 성능을 증가시킬 수 있다. 이와 더불어 탐색 속도 및 학습 정확성 향상을 위해 에이전트 간의 정보를 공유하는 방법 또한 단일 노드/다중 에이전트 시스템뿐만 아니라 다중 노드/다중 에이전트 시스템 성능 개선에 매우 중요한 요소들이다.

단일 노드/다중 에이전트 시스템 아키텍처는 에이전트가 실시간으로 정보를 공유하기 위한 중요한 타이밍 제약조건들이 포함되어 있다. 예를 들어, 에이전트가 실시간으로 시간제한 서비스를 노드에 제공하여 정보 손실 없이 데이터를 공유할 수 있고, 사용 가능한 리소스(Resource)와 시간 제약(Time Limitation)을 기반으로 신속히 목표점에 도달할 수 있게 도와줄 수 있다. 이러한 특징들은 시간에 수반되는 중요한 정보를 이용하여 시스템에 응답하는 기능으로 실시간 분산 다중 에이전트 시스템에서 발생할 수 있는 여러 가지 문제점들을 해결하는데 매우 유용하다. 또한, 시간을 이용한 다중 분산 노드를 사용하여 에이전트가 대규모 시스템 환경과 지리적으로 분산된 하위 로컬 환경을 모니터링하여 효율적으로 렌더링할 수 있다. 더욱이 충분한 시간을 갖고 다중 노드에서 동기화를 계산할 수 있으므로 분산 환경에서의 다중 에이전트 시스템에서 매주 중요한 기능으로 사용되고 있다.

본 연구에서는 강화학습을 이용한 다중 에이전트 기반 경로 탐색 성능 향상을 위해 다양한 방법으로 접근하였다. 단일/다중 에이전트가 주어진 목표에 도달하기 위해서는 탐색할 때 발생하는 환경 정보, 학습 속도(Learning Speed), 노드수(Number of Node) 및 탐색 환경의 크기(Node Size) 등의 여러 가지 측면을 고려해야 한다. 따라서 단일 목표가 아닌 다중 목표(Multi-goal)를 설정하여 학습 속도의 성능 변화를 살펴봐야 하며, 이를 위해 다양한 시나리오를 구성하여 여러 가지 실험을 진행하였다. 이와 더불어 다중 에이전트의 동기식/비동기식 탐색 방법에 따른 탐색 속도의 차이, 정보 공유 방법 등을 실험에 추가하여 검증하였다. 본 연구는 사례의 검증을 위해 강화학습 기반 다중 에이전트 시뮬레이션 환경을 구성하여 실험하였고, 이에 따른 다양한 결과를 비교하며 성능 향상에 노력하였다.

II. Preliminaries

1. Related works

1.1 Distributed Multi-agent System

에이전트는 시스템 환경에서 동작하여 작업을 수행할 수 있는 가상(Software) 또는 물리적(Physical Entity)

작업 방식이며 다른 에이전트와 직접 통신할 수 있다. 에이전트는 개별의 목표를 설정하고 이를 달성하기 위한 기능적 형태로 구동되며, 각 에이전트는 최적화를 위해 자체 리소스를 보유하고 이를 공유 할 수 있다. 에이전트의 가장 큰 특징 중 하나는 자신의 환경을 인식할 수 있고 환경에 대한 특징을 획득하여 다른 에이전트들에게 정보를 제공할 수도 있다는 것이다[4].

본 연구는 다양한 시나리오 환경을 구성하고 실시간으로 스케줄링을 실행할 수 있는 다중 에이전트 시스템 환경을 구성하였다. 실시간 다중 에이전트 시스템 환경에서 문제를 해결하기 위해서는 솔루션 제공과 더불어 이를 적절한 타이밍에 맞춰 시스템에 적용하는 과정이 매우 중요하다. 특히 시스템의 시간적 개념을 적용하여 제공되는 솔루션은 시간 타이밍에 따라 사용자에게 더이상 유용하지 않을 수 있으므로, 시스템 응답에 따른 최대 시간 간격으로 시스템을 정의해야 한다.

다중 에이전트 개념에서 데이터의 정확성과 전달 신속성은 단일 노드/다중 에이전트 환경, 다중 노드/다중 에이전트 환경에서 매우 중요한 문제이다. 특히 단일 노드 시스템에서 실행되는 실시간 다중 에이전트는 시스템 아키텍처 측면에서 타이밍 제약 요소를 고려해야 한다. 예를 들어, SIMBA[5]와 같은 아키텍처는 실시간으로 시간을 제한하는 서비스를 제공하는 실시간 다중 에이전트 시스템 기반 플랫폼이며, 시스템에 적용하는 시간적 개념을 매우 중요하게 정의하고 있다. SRTA[6] 아키텍처는 사용 가능한 리소스 및 시간 제약을 조절하여 에이전트가 목표에 도달하기 위한 스케줄링을 생성하는 방법을 제공하기 때문에 이 역시 다중 에이전트 시스템 환경에서 매우 중요한 개념이다.

분산 다중 실시간 에이전트는 사용자가 요구하는 문제를 빠르게 처리하고 해결하는데 더 많은 시간을 할애할 수 있고, 효율적인 시간 분배 방식을 여러 노드에 걸쳐 사용할 수 있어서 다중 에이전트가 처리해야 할 공통의 문제를 해결하기에 매우 유리하다. 또한, 다중 노드 방식에서 충분한 계산 시간을 확보할 수 있으므로 이전 시간 문제로 해결할 수 없었던 많은 응용 어플리케이션 문제들을 하나씩 해결할 수 있다. 결과적으로 이러한 시스템 아키텍처는 시간에 따라 적절한 타이밍에 데이터를 전달하고, 이에 따른 응답 시간을 결정하는 것이 매우 중요한 요인이다.

1.2 Reinforcement Learning (RL)

강화학습(Reinforcement Learning)은 학습을 통해 지식을 습득하고 이를 통합하여 적용할 수 있는 특징이 있다[7]. 강화학습의 구성 요소인 에이전트는 경험을 통한 학습 과정을 기반으로 시간적 변화에 따른 환경 상태 정보를 반복적으로 학습하고, 보상값을 통합하여 개선하는 기계학습 알고리즘 중 하나이다[8]. 특히 에이전트의 모든 행동(Action)과 연관된 정책(Policy)을 빠르게 개선하여, 신속히 최적 경로에 대한 누적 보상(Cumulative Reward)값을 극대화하는데 초점이 있다[9]. 일반적으로 강화학습은 다중 에이전트 기반 모니터링 시스템에서 많이 사용되고 있으며, 에이전트의 자율적인 접근 방식을 사용하여 학습 전략 문제를 해결하고 개선하는 문제에도 많이 사용되고 있다[10].

강화학습은 다중 에이전트 시스템[11]에서 학습 알고리즘이 적용된 이후에 가장 많이 사용되는 메모리 영역 중 하나이다. 비록 동적 환경에서의 강화학습 방법은 느린 학습 속도 및 하드웨어에 따른 보상 제한과 같은 문제가 존재하지만, CPU, GPU, 메모리 등의 하드웨어 발전과 Deep Neural Network(DNN)와 같은 심층 신경망 접근 방식들이 적용되어 빠르게 발전하고 있다.

강화학습은 모바일 로봇과 같은 시간적 행동 변화에 따른 실시간 응용 어플리케이션[12]에서도 많이 사용되고 있다. 다중 에이전트를 이용한 응용 어플리케이션 시스템에서도 강화학습은 에이전트의 협업(Collaborative)을 통해 문제를 지능적이고 효과적으로 해결하는 학습 방법이다[13].

본 연구에서는 강화학습을 이용한 협업 에이전트 기반의 분산형 다중 에이전트 최적화 시나리오를 적용하여 다양한 지능형 학습 방법을 제안하고자 한다. 또한, 에이전트 학습 및 경로 탐색 성능을 최적화하기 위해 다양한 실험 및 검증 환경을 구성하여 연구를 진행하였다.

III. The Proposed Scheme

1. Proposed Methods

1.1 Dynamics-Based-Motion Constraints Multi-Agents Path-planning

본 연구는 강화학습 기반 분산 다중 에이전트 시스템을 구현하여 효과적인 에이전트 경로 탐색과 이를 위한 최적화 성능 향상에 목표가 있다. 위 Fig.1은 본 연구의 시스템 구조이며, Centerized 기반의 Hybrid[9][14][15] 방식인 Master/Slave 구조를 사용하였고, 메인 Master가 각각의 Slave에게 먼저 메모리(작업)를 할당한다.

메모리를 할당받은 Slave는 특정 작업을 수행한 후 결과를 다시 Master에게 보고하며, 이때 Peer to Peer[16] 방식을 사용하여 각 노드의 정보를 독립적으로 다른 노드

로 전송하는 노드간 통신(Knowledge Transfer)을 진행한다.

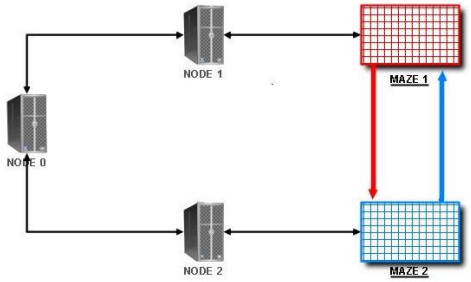

Fig. 1. Hybrid P2P and Master/Slave Architecture Overview

Fig. 2과 같이 에이전트는 하나의 중앙 집중식 노드에서 실행되어 탐색 및 학습을 수행하지만, 분산 노드에서도 탐색 및 학습을 진행할 수 있다. 이 경우 에이전트는 하나의 노드에서 다른 노드로 이동하여 탐색을 시도하는 경우가 발생한다. 이때 에이전트는 지식 기반 경로 탐색을 사용하며 전체 에이전트의 학습 및 시스템 성능을 위해 탐색 경로에 대한 정보를 공유한다.

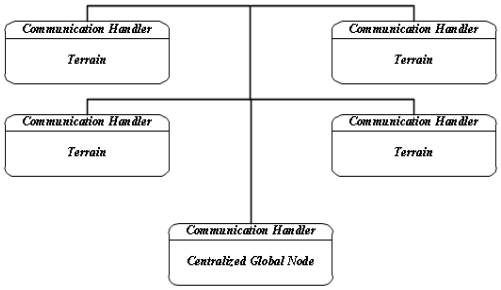

Fig. 2. Structure of Communication Handler

강화학습(Fig. 3) 기반 시스템 환경에서 에이전트는 대상 간의 행동(Action), 상태(State) 및 보상(Reward) 관계를 나타낸다. 다중 에이전트는 최종 목표에 도달하기 위해 단계(Step)마다 연속되는 보상값을 누적하여 최적의 경로 생성을 시도한다(Fig. 4).

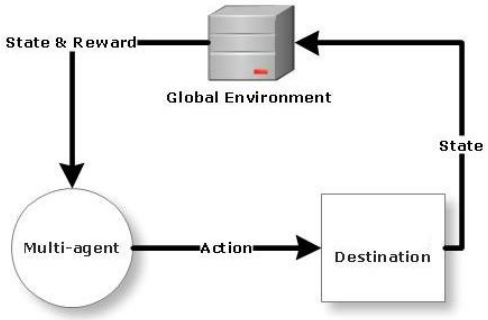

Fig. 3. Workflow of Multi-agent System based on RL

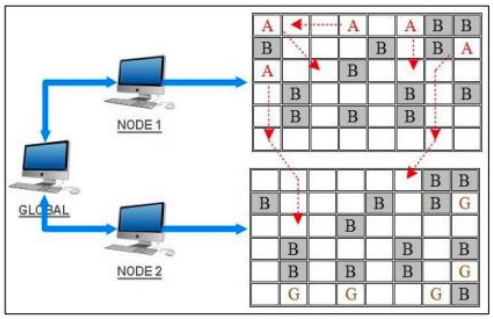

Fig. 4. Exploratory Trials by Multi-agent

에이전트는 Fig. 5와 같이 경로 탐색 및 이동을 위해, 4개 (Up, Down, Right, Left) 혹은 8개 (Four Directions & Up-right, Up-left, Down-right, Down-left) 이동 방향을 선택할 수 있고, 경로 탐색 중 직면하는 장애물(Obstacle)을 피해 신속하게 목표점에 도달할 수 있도록 최소 곡률 반경(Minimum Radius of Curvature), 회전 각도(Turning Angle), 반경 제약 조건(Radius Constraints)등을 사용하여 경로를 따라 이동한다. 또한, 분산 환경에서 에이전트는 다음 m개의 움직임을 결정하기 위해 슬라이딩 윈도우 메커니즘을 사용하여 최소 회전 반경과 목적지까지의 경로가 가장 짧은 최적/최단 경로를 탐색한다[9][17].

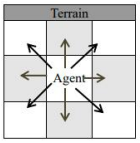

Fig. 5. All Directions of Agent Move

본 연구에서 제안된 분산 다중 에이전트 주요 알고리즘은 다음과 같다.

Table 1. Main Proposed Algorithm

Main Proposed Algorithm
For every Node (N_i), where i = 1, 2, ..., n
1. Let Ni denotes the number of nodes
2. Let Ni denotes the number of agents
3. Let Gk denotes the number of agents
4. Place an initial number of agents A_j in random positions (x_{ij}, y_{ij}), where j denotes number of agents
5. Every A_j (agents) in N_i
a. Do an initial exploration(Random-trial) to the corresponding G_k
b. Do exploration (using RL: Optimal-trial) for Tn denotes the number of trials

Table 1에서 제시하고 있는 것과 같이 에이전트는 초기 경로 탐색으로 무작위시도(Random-trial) 방법을 채

택한다. 이 방법은 강화학습에서 초기학습과 같은 맥락이며 최적 경로 탐색을 위해, 무작위시도를 생성하여 다양한 환경 정보를 초기에 탐색하는 방법이다. 에이전트는 탐색 경로 이동을 위한 2차원 환경의 차순위 동작을 위해 다음 식을 이용하여 경로 탐색을 위한 학습을 진행한다.

$$f(x,y) = O(x,y) \tag{1}$$

$$- \frac{\epsilon^2}{2}(x_g - x_c)[(x_g - x_c)^2 + (y_g - y_c)^2]^{-\frac{1}{2}} \tag{2}$$

$$+ \frac{\epsilon^2}{visited} \tag{3}$$

$$- \frac{\epsilon^2}{reward} \tag{4}$$

$$O(x,y) = \frac{1}{2}\epsilon^2\left(\frac{1}{\rho(x,y)} - \frac{1}{\rho_0}\right)$$

에이전트는 제공되는 환경에 반응하는 위치 경로값에 따라 4개의 항목으로 구분하여 평가된다. 첫 번째, 에이전트가 경로 탐색 중 탐지할 수 있는 장애물의 위치 정보(1)와 이를 파악하기 위한 장애물 위치 정보 값을 계산해야 한다, 두 번째, 목적지(Goal/Destination)를 향해 에이전트가 경로 탐색을 진행하면서 얻는 해당 위치 정보 값(2), 그리고 에이전트가 현재 혹은 이전 경로 탐색 중에 방문한 위치 정보값(3)을 반복적으로 평가한다. 마지막으로 에이전트의 반복되는 경로 탐색의 누적 보상값(4)을 평가하여 경로 탐색을 위한 최적화 학습을 진행한다[14]. 결국 에이전트는 목적지에 도달하기 위한 경로 탐색의 학습 및 평가 개선을 반복적으로 누적 적용하여 효과적인 최적 경로 탐색을 할 수 있다. (Effect of Obstacle(1), Effect of Goal(2), Effect of Visited Positions(3), Effect of Reward(4))

$$D(s_k, a_k) = \frac{1}{d_{k+1,G}+1}\ ,\ where\ f(s_k, a_k) = \max_D(\max_v(s_k, a_k)) \tag{5}$$

또한, 에이전트가 목표점에 빠르게 도달하기 위해서는 환경으로부터 제공받는 상태정보(State) 값과 이에 따른 행동(Action)의 최대값(5)을 각각 학습 및 개선하여 효과적인 최적 경로 탐색을 달성하는 것이 최종 목표이다[14].

주요 학습 알고리즘에 따른 에이전트의 경로 탐색 방법은 다음과 같이 구분된다. 위에서 언급했듯이 강화학습은 최적 경로를 위해 초기에는 무작위시도가 선택되어 경로 탐색을 진행하고, 1차 탐색이 끝난 이후 2차 경로 탐색부터는 1차 경로 탐색에서 저장된 데이터를 토대로 경로 학습을 반복적으로 시도하는 최적화 학습(Optimal-trial)을 한다.

Table 2. Initial Random Exploration

Random-trial
1. Let S_k denotes the current state
2. Relinquish S_k so that the other agent can occupy the position
3. Assign the agent new position
4. Update the current state $S_k \leftarrow S_{k+1}$

Table 2와 같이 에이전트는 처음 최종 목표 도달까지(RL-based 1st Episode) 무작위시도 방식을 사용하며 에이전트가 처음으로 목표에 도달한 후 계속해서 연속적인 탐색 및 학습을 이어간다. 두 번째 탐색부터는 에이전트가 처음 경로 탐색 중에 획득한 정보를 기반으로 학습 정책을 개선하며 경로 탐색을 최적화하는 학습 알고리즘(Table 3)을 이용한다. 각각의 경로 탐색마다 에이전트는 시스템 환경의 상호 작용 및 학습 정보를 토대로 탐색을 진행하며 학습을 최적화한다.

Table 3. Optimal Knowledge-based Exploration

Optimal-trial
1. Let S_k denotes the current state
2. Let P_k denotes an action
3. Let Q_k denotes the discounted reward value
4. Choose an action $P_k \leftarrow$ Policy(S_k, P_k), where the policy used in the work (using RL)
5. Choose an action $P_{k+j} \leftarrow$ Policy(S_{k+j}, P_{k+j}), where j is for 1 to N
6. Move in one of the directions by agent (four or eight available directions)
7. Update the learning model in the brain (global environment) with new value Q(S_k, P_k)
8. Update the current state $S_k \leftarrow S_{k+1}$

1.2 Path-Planning via Selected Sub-goals

앞에서 언급했듯이 에이전트는 초기 경로 탐색을 위한 무작위시도(Random-trial)를 진행하면서 경로 탐색 정보를 수집한다. 이때 수집된 환경 정보 요소(시뮬레이션 환경 크기, 장애물 수 등)에 따라 경로 탐색의 시간 및 단계수(Number of Step)가 급격히 증가하여 탐색 속도에 영향을 미칠 수가 있다. 이런 문제를 해결하기 위해 다중 혹은 하위 목표(Sub-goals)를 추가로 구성하여 다중 목표를 이용한 동적 에이전트 경로 탐색 방법을 새롭게 시도하였다(Fig. 6). 다시 말해서 에이전트가 다중 혹은 하위 목표 탐색 중에 수집한 누적 경로 정보를 다른 목표 탐색에 이전 사용 및 응용하여 초기 무작위시도에서 발생하는 경로 탐색 성능 감소를 방지하는데 목표를 두고 있다.

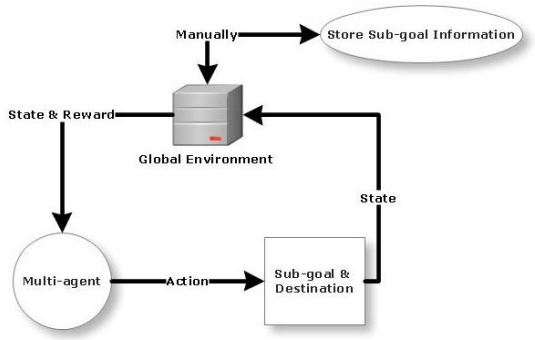

Fig. 6. Workflow of Dynamic Agent Path-planning via Sub-goal based on RL

Fig. 6은 초기 시스템 구성도에 하위 목표 경로 탐색 정보를 저장하여 학습을 위한 보상 체계를 사용하는 분산 다중 에이전트 방법을 새롭게 제시하고 있다. 각각의 에이전트는 첫 번째 하위 목표에 도달할 때까지 무작위시도 방법으로 경로 탐색을 시작하고, 이때 수집되는 에이전트의 하위 목표 경로 환경 정보를 임의 학습/저장하여 이를 다음 혹은 최종 목표 경로 탐색에 적용하는 방법으로 경로 탐색 성능을 향상시킨다. 그리고 이미 하위 목표에 도달한 임의에 에이전트는 다음 혹은 최종 목표를 위한 경로 탐색을 잠시 중단하고 다른 에이전트와 경로 탐색 정보를 공유하여 모든 에이전트 성능 향상을 위한 동기식 에이전트 경로 탐색 방법을 진행한다. 이 방법은 전체 에이전트의 정보를 공유하여 경로 탐색 성능 향상에 매우 효과적인 영향을 미칠 수 있다.

Table 4. Sub-goal Selection Algorithm

Sub-goal Selection
1. Let B denotes Terrain broadly for optimal-trial
2. Let NA_k denotes number of agents (k=1,2,3,...)
3. Let NG_k denotes number of goals (k=1,2)
4. Let RW denotes random-trial (without knowledge)
5. Let OW denotes optimal-trial (with knowledge)
Upon Receiving the event *If(NA_k < NG_k)* *Do take RW to find a goal* *Save B in memory for next exploration* *Then, do take OW to find a goal* *Else:* *Do take RW to find a goal (without sub-goal)*

1.3 Autonomous and Asynchronous Triggered Exploratory Multi-agent Path-Planning

본 연구는 위에서 언급한 분산형 다중 에이전트 경로 탐색 방법을 보다 효율적으로 개선하기 위해 에이전트의 학습 단계에서 시스템 스케줄링 방법을 추가하였다. 이는 환경 정보 공유를 위한 에이전트의 동기식 경로 탐색에서 발생하는 탐색 지연(Latency) 시간을 감소시키는 비동기식 다중 에이전트 경로 탐색 방법이다. 이 방법은 기존의 동기식 에이전트 경로 탐색 방법의 단점을 보완하여 에이전트가 다른 에이전트의 무작위시도를 통한 하위 목표 탐색 종료까지 대기하면서 정보를 공유하지 않고, 연속적으로 다음/최종 목표까지 탐색을 진행하는 자율 및 비동기 트리거 탐색 학습 알고리즘이다(Fig.7). 모든 에이전트는 학습을 위한 지연 시간 없이 하위 및 최종 목표까지 연속적으로 경로 탐색을 진행할 수 있으므로 매우 효율적인 분산형 다중 에이전트 탐색을 실현할 수 있다.

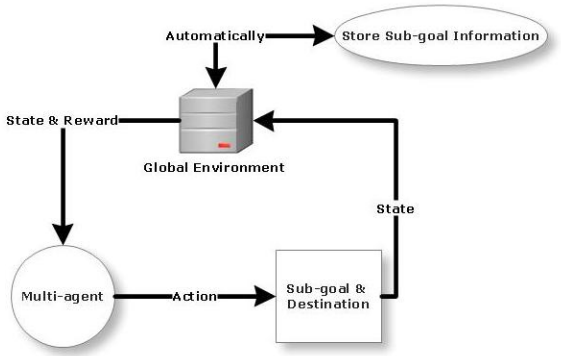

Fig. 7. Workflow of Asynchronously Triggered Path-planning via Sub-goal based on RL

아래 Table 5에서 제안된 알고리즘은 에이전트가 하위 목표에 도달한 이후의 초기 단계부터 정의하고 있으며, 이는 각각의 에이전트가 최종 목표에 도달하기 위해 비동기식으로 경로 탐색 및 학습을 진행하고 있음을 의미하고 있다.

Table 5. Asynchronously Triggered Path-planning

Asynchronously Triggered Exploration
1. Let A_k denotes number of agents (k=1,2,3,...)
2. Let S denotes sub-goal
3. Let D denotes destination (Goal)
4. Let O denotes obstacle barriers in the terrain
5. Based on the current state, choose RW or OW by the received event
6. Based on next state,
a. If A_k reaches at the S, set to update policy with knowledge. After then, agent asynchronously trigger/switch the agent exploratory trials with no latency to reach the goal. *b. If A_k still cannot reach S, go continue of exploration depending on the event*

2. Experimental Results

2.1 Multi-Agent Distributed Agent Path-planning Based on Sharing-information

분산 다중 에이전트 시스템에서 에이전트는 개별적으로 다른 컴퓨팅 노드에 위치할 수 있고, 다른 분산 노드에서 실행될 수 있으며, 최종 목표를 찾기 위해 동일 노

드에서 다른 노드로 이동하는 경우가 발생할 수 있다. 또한, 에이전트는 자신이 탐색 중인 노드에 있는 정보에만 액세스할 수 있으며 다른 에이전트와 정보를 공유하기 위해서 시스템을 연결하여 통신을 진행할 수 있다.

Fig. 8과 9의 결과는 5개의 다중 에이전트가 두 개의 분산 메모리 노드에서 동시에 실행되고 있으며, 다른 분산 컴퓨팅 노드에서 같은 목표를 향해 다수의 경로 탐색을 진행하면서 도출한 결과이다. 이 결과는 각각의 에이전트가 정보 공유의 유/무에 따라 경로 탐색 성능의 차이가 큰 것을 보여주고 있다.

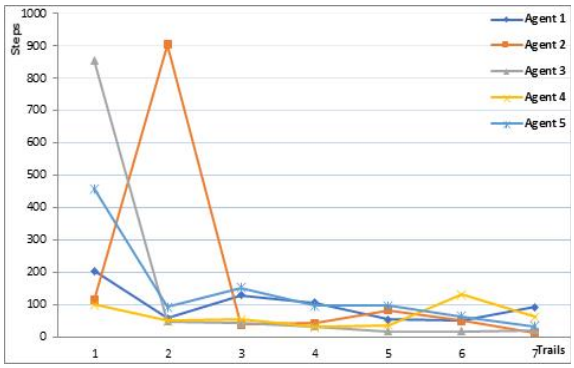

Fig. 8. Result of Multi-agent Path-planning without Sharing Information toward Destination

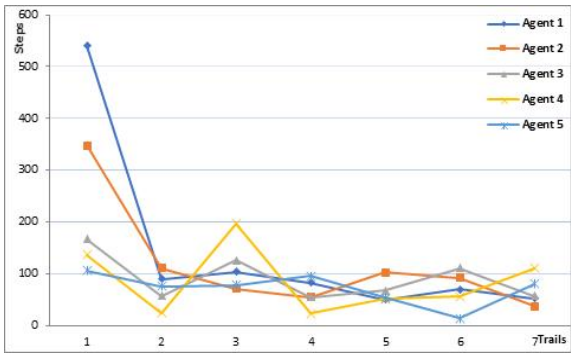

Fig. 9. Result of Multi-agent Path-planning with Sharing Information toward Destination

2.2 Multi-Agent Distributed Agent Path-planning depending on Sub-goal Selection Scheme

다음 실험은 에이전트가 두 개의 분산 컴퓨팅 노드에서 하위 목표(Sub-goal)에 도달하면서 획득하는 환경 정보를 학습하여 이를 다른 에이전트와 공유하는 문제를 실험한 결과이며, 에이전트의 정보 공유 유/무에 따른 성능 결과를 비교한 것이다. 각각의 에이전트는 동일 환경 정보(하위 목표, 장애물 위치)를 사용하고 있으며, 최종 목표에 도달하기 위해 서로 정보를 공유하고 학습하는 협업 경로 탐색법을 사용하고 있다.

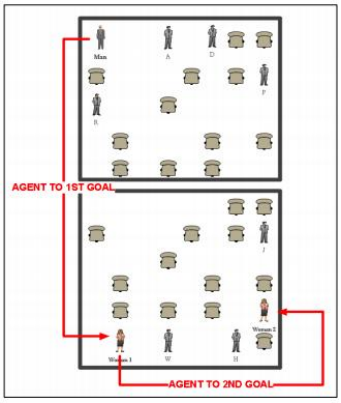

Fig. 10. Scenario of Sub-goal Selection for Path-planning

Fig. 10 실험 결과는 위에서 언급한 시나리오를 기반으로 도출한 모의 시뮬레이션 환경이다. 이 실험에서는 에이전트가 하위 목표에 도달하여 획득한 탐색 정보를 다른 에이전트와 공유하여 학습하였을 경우와 그렇지 못한 경우, 그리고 하위 목표와 관련 없이 직접 최종 목표에 도달하였을 때의 성능을 각각 비교하였으며(Fig.11), 결과적으로 분산 다중 에이전트 시스템에서 에이전트간의 탐색 정보 공유 및 경로 학습은 전체 시스템 성능 개선에 매우 효과적인 것을 알 수 있다.

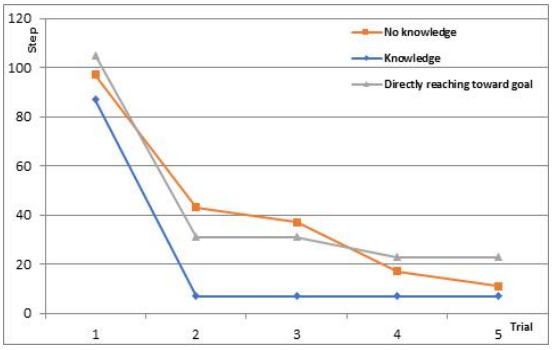

Fig. 11. Result of Agent Path-planning Whether to Share Data by Sub-goal Selection

2.3 Automatic and Asynchronous Triggered Exploratory Path-Planning

이번 연구의 실험은 각각의 에이전트가 비동기식 경로 탐색 방법을 채택하였을 때 나타나는 결과이다. 이전 실험들에서는 에이전트의 정보 공유를 위해 동기식 다중 에이전트 경로 탐색을 선택하였지만, 에이전트가 하위 목표에 도달한 후 다음/최종 목표 탐색전까지 시간적 타이밍을 위한 동기화를 진행하기 위해 다른 에이전트의 경로 탐색 종료 시까지 대기하면서 정보를 공유하는 상황이 발생하였으며, 이로 인해 경로 탐색 도중 지연 시간이 발생

하는 경우가 나타났다. 따라서 이번 실험을 통해 에이전트는 하위 목표 도달 후 지연 시간 없이 개별적으로 다음/최종 목표까지 연속적으로 이동하면서 경로 탐색을 진행하는 방법을 구현하였으며, 이전 동기식 경로 탐색과의 차이에 대한 비교 실험 및 결과를 도출하였다.

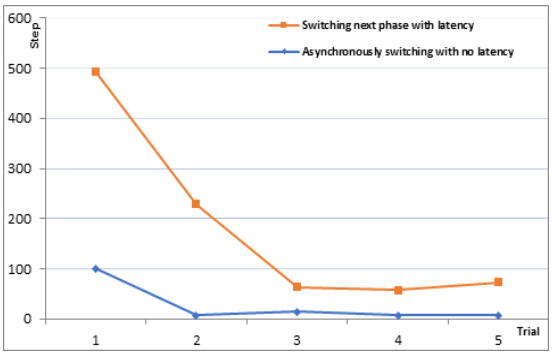

Fig. 12. Result of Path-planning (Synchronous vs Asynchronous)

Fig.12의 실험 결과를 통해 에이전트의 자율적 비동기식 에어전트 경로 탐색 방법이 매우 효과적인 결과를 만들어 낼 수 있다는 것을 알 수 있다. 이와 같은 컴퓨팅 스케줄링 문제는 실제 시스템 환경에서도 매우 유사하게 나타날 수 있으므로 위와 같은 연구 방법 및 검증을 통해 향후 보다 효과적인 지능형 다중 에이전트 연구 방법을 제안할 수 있을 것으로 예상된다.

IV. Conclusions

본 연구에서는 에이전트 학습 속도 향상을 위한 강화학습 기반 분산형 다중 에이전트 시스템 방법을 제안하였다. 특히 다양한 시나리오 및 실험을 통해 새로운 지능형 접근 방법을 연구 및 분석하였고, 이에 따른 전반적인 지능형 시스템 성능 향상을 위해 다음과 같은 연구 방법들을 진행하였다.

첫째, 본 연구에서 강화학습 기반 다중 에이전트 분산 협업 에이전트 시스템을 제안하였다. 분산 환경에서 발생할 수 있는 환경적 변화에 따른 시나리오를 구현하였고, 다양한 실험을 통해 효율적인 지능형 분산 시스템의 모델링을 구현하였다. 또한, 분산형 다중 에이전트 시스템에서 강화학습을 적용하는 방법을 제안하였다. 이를 토대로 추후 분산형 시스템 관련 분야에 적용 및 확장 가능할 것으로 기대한다.

둘째, 에이전트 경로 탐색 및 학습 성능 향상을 위해 에이전트 탐색 정보 공유에 대한 지능형 시스템의 협업 과정을 강조하였다. 이는 서로 다른 목표를 가진 다중 에이전트 사이의 정보 공유 여부를 통해 효과적인 학습 방법을 구현하는 것에 초점을 두고 있다.

마지막으로 분산 다중 에이전트 시스템의 효율성을 위해 비동기식 에이전트 경로 탐색 방법으로 전환하였다. 이는 에이전트가 자율적으로 개별 경로 탐색을 할 수 있게 만드는 스케줄링 기반 시스템 성능 향상 방법이다. 이 방법은 현실적인 시스템 개선 방법으로써 시스템 스케줄링 및 최적화 지능형 시스템 구현에 궁극적인 목표가 있으며 협업 다중 에이전트 학습 프로세스 성능 향상에 그 목표가 있다.

본 연구에서 제안한 방법들은 다른 분산형 시스템 환경에서 확장 및 적용 가능할 것이며[18][19], 스마트팩토리, 스마트시티 도메인 및 지능형 로봇 시스템과 같은 응용 환경에서도 모델링하여 효과적으로 사용할 수 있을 것으로 기대한다.

ACKNOWLEDGEMENT

This research was funded by a 2021 research Grant from Sangmyung University.

REFERENCES

[1] D. B. Megherbi, D. C. Xu, “Multi-Agent Distributed Dynamic Scheduling for Large Distributed Critical Key Infrastructures and Resources (CKIR) Surveillance and Monitoring”, in Proceeding of IEEE International Conference on Technology for Homeland Security(HST), 2011. DOI: 10.1109/THS.2011.6107907

[2] K. Zhang, Z. Yang, and T. Basar, “Networked Multi-Agent Reinforcement Learning in Continuous Spaces”, in Proceeding of 2018 IEEE Conference on Decision and Control (CDC), 2018.DOI: 10.1109/CDC.2018.8619581

[3] D. B. Megherbi, P. Levesque, “A Distributed Multi-Agent Tracking, awareness, and communication System Architecture for Synchronized Real-Time Situational Understanding, Surveillance, Decision-Making, and Control”, in Proceeding of IEEE International Conference on Technology for Homeland Security(HST), 2009. DOI: 10.1109/THS.2010.5654983

[4] D. B. Megherbi, Radumilo-Franklin, Jelena, “An Intelligent Multi-agent Distributed Battlefield via Multi-Token Message Passing”, in Proceeding of IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, 2009. DOI: 10.1109/CIMSA.2009.5069929

[5] J. Soler, V. Julian, M. Rebollo, C. Carrascosa, V. Botti., “Towards a Real-Time Multi-Agent System Architecture”, Universidad Politécnica de Valencia, Valencia, Spain, 2002.

[6] B. Horling, V. Lesser, R. Vincent, T. Wagner, “The Soft Real Time Agent Control Architecture”, UMASS Department of Computer Science Technical Report WS-02-15, USA, 2002.

[7] Stuart Russell, Peter Norvig, “Artificial Intelligence”, A Modern Approach 2nd edition, Prentice Hall, 2003.

[8] Xue Jinlin, Gao Qiang, Ju Weiping, “Reinforcement Learning for Engine Idle Speed Control”, in Proceeding of 2010 International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), 2010. DOI: 10.1109/ICMTMA.2010.249

[9] D. B. Megherbi, M. Madera, “A hybrid P2P and master-slave architecture for intelligent multi-agent reinforcement learning in a distributed computing environment: A case study”, in Proceeding of IEEE International Conference, Computational Intelligence for Measurement Systems and Applications (CIMSA), 2010. DOI: 10.1109/CIMSA.2010.5611770

[10] M. Madera, D. B. Megherbi, “An Interconnected Dynamical System Composed of Dynamics-based Reinforcement Learning Agents in a Distributed Environment: A Case Study”, in Proceeding of IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, 2012. DOI: 10.1109/CIMSA.2012.6269597

[11] W. M. Zuberek, “Performance Limitations of Block-Multithreaded Distributed-Memory System”, in Proceeding of the Winter Simulation Conference(WSC), 2009. DOI: 10.1109/WSC.2009.5429718

[12] M.R Shaker, S. Yue, T. Duckett, “Vision-based reinforcement learning using approximate policy iteration”, in Proceeding of 2009 International Conference, 2009.

[13] J. JIANG, S. Zhao-Pin, Q. Mei-Bin, G. ZHANG, “Multi-task Coalition Parallel Formation Strategy Based on Reinforcement Learning”, Acta Automatica Sinica, Vol.34, No.3, pp.349-352, 2008.

[14] D. B. Megherbi, M. Kim, “A Collaborative Distributed Multi-Agent Reinforcement Learning Technique for Dynamic Agent Shortest Path Planning via Selected Sub-goals in Complex Cluttered Environments”, in Proceeding of IEEE Conference, CogSIMA, 2015.

[15] D. B. Megherbi, V. Malaya, “A Hybrid Cognitive/Reactive Intelligent Agent Autonomous Path Planning Technique in a Networked-Distributed Unstructured Environment for Reinforcement Learning”, The Journal of Supercomputing, Vol. 59, Issue3, pp.1188-1217, 2012.

[16] C. Picus, L. Cambrini, W. Herzner, “Boltzmann Machine Topology Learning for Distributed Sensor Networks Using Loopy Belief Propagation Inference. Machine Learning and Applications”, in Proceeding of 2008th Seventh International Conference, ICMLA, 2008. DOI: 10.1109/ICMLA.2008.60

[17] D. B. Megherbi, M. Kim, M. Madera, “A Study of Collaborative Distributed Multi-Goal and Multi-agent based Systems for Large Critical Key Infrastructures and Resources (CKIR) Dynamic Monitoring and Surveillance”, in Proceeding of IEEE International Conference on Technologies for Homeland Security, 2013. DOI: 10.1109/THS.2013.6699087

[18] J. Kim, H. Lim, C. Kim, M. Kim, Y. Hong, Y. Han, “Imitation Reinforcement Learning-Based Remote Rotary Inverted Pendulum Control in OpenFlow Network” Published in IEEE Access, Vol. 7, 2019.

[19] A. Sharma, S. Gu, S. Levine, V. Kumar, K. Hausman, “DADS: Unsupervised Reinforcement Learning for Skill Discovery”, posted by AI Resident, Google Research at the Google Brain team and the Robotics at Google team, May. 2020.

Authors

Min-Suk Kim received his M.S. in Telecommunication and Networks from University of Pittsburgh, USA, in 2010. He also received a Ph.D. from Electrical and Computer Engineering in the University of Massachusetts Lowell, USA, in 2016 respectively.

He was a senior engineer at the Electronics and Telecommunications Research Institute (ETRI) from 2016 to 2020. Since 2020, he has been an assistant professor with the Department of Human Intelligence and Robot Engineering, Sangmyung University, Cheonan, Korea. His research involves Reinforcement Learning, Deep Learning, Edge Computing and Centralized Cloud Computing.