Journal of Logistics Science & Technology volume 5, No.4, pp. 62-86 (December 2024)
물류과학기술연구 제5권 제4호 pp. 62-86 (2024년 12월)
(e)ISSN : 2765-2351
https://doi.org/10.23178/jlst.5.4.202412.005

다중 에이전트 강화 학습을 이용한 트랜스포터 혼잡 회피 경로 계획 최적화

윤성재*

한양대학교 산업공학과

Multi-Agent Reinforcement Learning for Optimizing Path Planning of Transporters Considering the Congestion

Seong Jae Yoon*

Department of Industrial Engineering, Hanyang University

Advanced mega distribution centers(mega DCs) equipped with cutting-edge technologies are essential for enabling intelligent logistics operations. However, the efficient management of large-scale mega DCs with a dynamic array of autonomous transportation devices presents complexities that traditional path planning approaches struggle to address. To overcome these challenges, this study proposes an innovative path planning framework that integrates Multi-Agent Reinforcement Learning (MARL) with supplementary learning techniques to facilitate accelerated and stable training. This framework incorporates a specialized reward schema alongside auxiliary methodologies aimed at optimizing system-wide performance. Initially, we train MARL networks within a controlled, small-scale environment, focusing on path optimization for a limited number of transportation agents to validate efficiency. The trained model parameters are then leveraged as initial conditions for deployment in a larger-scale environment, thus expediting adaptation. Additionally, we employ a curriculum learning approach, segmenting the training process into levels of progressive difficulty to enhance convergence in complex, large-scale scenarios. Empirical results demonstrate that the proposed approach effectively mitigates operational challenges such as collisions, congestion, and deadlock, leading to significant improvements in overall system performance.

Keywords: Distribution center, Multi-agent reinforcement learning, Dynamic path planning, Transporter, Congestion

논문접수일 : 2023.10.20. 논문수정일 : 2024.09.23. 게재확정일 : 2024.12.20.
* 한양대학교, 산업공학과 석사과정, Corresponding Author: ysj1646@hanyang.ac.kr

1. 서론

1.1 4차 산업 혁명과 물류의 변화

4차 산업혁명의 급속한 발전은 다양한 산업 전반에 걸쳐 혁신과 기술 변화를 가속화하며, 물류 분야에서도 중요한 변혁을 이끌고 있다. 특히 온라인 판매가 전통적인 오프라인 판매를 앞지르면서 풀필먼트 서비스의 경쟁력이 주요한 요인으로 부각되었고, 이는 대규모 창고인 메가 유통 센터(mega DC)에 대한 수요 증가로 이어지고 있다. 메가 DC는 창고 운영을 최적화하고 고객에게 제품을 신속하게 전달하기 위해 첨단 기술을 통합하여 운영하고 있으며, 그 중 무인 운송 차량(AGV)과 자율 이동 로봇(AMR)을 포함한 운송 로봇은 시스템의 효율성에 중요한 기여를 하고 있다(Fazlolahtabar, 2015). 1955년 최초의 AGV가 발명된 이후(Muller, 1983), 2017년까지 전 세계적으로 13,000대 이상의 AGV 및 AMR이 도입되었으며(Bechtsis, 2017), 이들은 운영 체계의 발전과 함께 비전 기반 시스템으로 진화하였다(Fragapane, 2021).

과거 물류 현장에서는 작업자가 직접 제품을 찾고 운송하는 방식이 주를 이루었으나, 이는 물류 작업의 속도 및 정확성에서 한계를 보였다. 이러한 문제를 해결하기 위해 자동화 시스템이 도입되었으나, 예측 불가능한 장애물이나 교통 혼잡 상황에 실시간으로 대응하는 데에는 여전히 한계가 있었다. 하지만 최근 기술의 발전으로 인간 중심의 프로세스, 제한된 컴퓨팅 자원, 그리고 부족한 센서 기술 등의 한계가 상당 부분 극복되었다.

메가 DC의 경우, 작업자는 이제 수동으로 물품을 픽업할 필요 없이 지정된 선반을 트랜스포터가 자동으로 운반한다. 이에 따라 다수의 트랜스포터가 충돌 및 혼잡 없이 효율적으로 경로를 계획하고 제어하는 메커니즘이 필수적이다. 그러나 동적이고 복잡한 메가 DC 환경에서 전통적인 경로 계획 방법은 여러 복잡한 문제를 야기할 수 있다. 유전 알고리즘, Artificial Potential Field 방법, A* 알고리즘(Jia, 2017) 등 다양한 접근법이 개발되었으나, 이들 방법은 메가 DC와 같은 복잡한 환경에 적응하고 확장하는 데 한계를 가지고 있다. 특히 다수의 트랜스포터를 동적으로 제어하면서 교통 혼잡과 동적 장애물을 처리하는 능력에는 여전히 많은 도전 과제가 남아 있다. 이러한 문제를 해결하고 메가 DC의 디지털 트윈 시스템 목표를 달성하기 위해서는 보다 고도화된 제어 및 경로 계획 알고리즘이 필요하다.

본 연구에서는 다중 에이전트 강화 학습(MARL)과 다양한 학습 기법을 통합한 모델을 제안하여 메가 DC 환경에서 다수의 트랜스포터를 효과적으로 제어할 수 있는 솔루션을 탐구한다. 본 연구는 MARL의 확장성과 일반화 가능성을 기반으로, 복잡한 창고 환경에서 다수의 트랜스포터가 협력적이면서도 독립적으로 운영될 수 있는 강력한 경로 계획 솔루션을 제시하는 것을 목표로 한다. 이를 통해 메가 DC 운영의 효율성을 향상시키고, 대규모 물류 시스템에서의 최적화 가능성을 검증하고자 한다.

1.2 연구배경 및 목적

메가 DC 내에서 여러 트랜스포터를 운용하기 위한 효과적인 솔루션을 제공하기 위해 MARL과 성능을 보조해 줄 여러 학습 방법의 통합 모델을 제안한다. 본 연구는 동적 창고 운영을 위한 트랜스포터 경로 계획에서 MARL 기법의 확장성, 일반화 가능성을 탐구한다.

위 목표와 관련된 연구 질문은 다음과 같다.

RQ1: 시스템 성능은 무엇과 관련이 있으며, 어떻게 향상 시킬 수 있는가?

RQ2: 긴 학습 시간 문제를 어떻게 개선 하는가?

RQ3: 혼잡 또는 idle한 상태와 같은 트랜스포터의 이동과 관련된 문제를 어떻게 해결하는가?

1.3 문헌 검토

1) 물류 트랜스포터의 경로 계획

경로 계획은 차량, 장치 또는 움직이는 물체가 주어진 환경 내의 시작점에서 목적지까지 최적의 경로 또는 궤적을 결정하는 과정이다. 다양한 영역에서 트랜스포터를 위한 최적 경로를 설계하는 것은 여전히 중요한 과제로 남아 있다. 과거에는 환경이 상대적으로 정적이고 예측 가능한 경계 내에서 관리되었고, 이는 A* 알고리즘(Hart et al., 1968)을 사용하여 시작점과 끝점 쌍을 연결하는 최적해를 찾는 것을 가능하게 하였다. A* 알고리즘의 시간 복잡도는 O(bd)로 표현되며, 여기서 b는 각 단계에서 탐색한 검색 공간의 수를, d는 최적 솔루션의 깊이를 나타낸다. 그러나 오늘날의 시스템에서는 차량 수가 증가하고 변화하는 환경에 대한 적응력이 요구된다. 다중 에이전트 경로 계획 문제는 최단 경로를 찾는 것뿐만 아니라, 실시간 변화에 동적으로 적응하고 유휴 시간을 최소화하는 것을 포함한다. 이러한 이유로 탐욕 검색 기반 방법은 실질적인 응용 분야에서 적합하지 않을 수 있다. 많은 연구가 운송, 공급망 관리, 창고 자동화 등 다양한 물류 시나리오에서 솔루션을 찾기 위한 알고리즘 및 모델 개발에 기여하였으나, 항상 솔루션 품질과 계산 비용 사이의 trade-off 관계에 직면하였다. 많은 차량에 대한 경로를 개략적으로 찾는 것은 적은 계산 부담으로 가능하지만, 이는 모든 가능한 경우에 대한 솔루션을 보장하지 않는다. 반면, 최적의 충돌 없는 경로를 찾는 것은 상당한 계산 자원을 요구하며, 동적 조건을 가진 실제 물류 시스템에 적용하기 어렵다. Yalcin은 그리드 기반 창고에서 여러 항목의 동시 저장 및 검색을 위한 그래프 검색을 사용한 분리된 다중 에이전트 경로 계획을 제안하였다. Chen et al.(2021)은 MAPD(Multi-Agent Pickup and Delivery) 문제를 연구하였으며, TA(Task Assignment)와 MAPF(Multi-Agent Path Finding)의 두 가지 주요 구성 요소를 포함한다. 전통적인 방법은 일반적으로 TA와 MAPF를 순차적으로 해결하며, 해당 연구에서는 한계 비용 할당 휴리스틱과 대규모 이웃 검색을 기반으로 한 메타 휴리스틱 전략을 소개하였다. Wang et al.(2011)은 무방향 그래프를 사용하여 교착 상태와 사이클이 있는 경우에 대한 솔루션을 제안하였다. 다음 하위 절에서는 다중 에이전트 강화 학습 알고리즘의 기본 아이디어부터 알고리즘의 진화를 추적하고, 이 과정에서 알고리즘이 직면한 한계에 대해 논의할 것이다. 또한, 이러한 결점을 어떻게 보완하여 본 연구에 채택된 QMIX 알고리즘으로 발전했는지를 설명할 것이다.

2) 다중 에이전트 강화학습

단일 에이전트 상황에서의 경로 계획을 위한 강화 학습은 인상적인 성능을 보였다. Panov et al.(2018)과 Lei et al.(2018)은 에이전트의 센서 범위 내에서 패턴을 인식하기 위해 컨볼루션 계층을 사용하여 경로 계획을 위한 Q-네트워크의 심층 신경망을 설계하였다. 그러나 여러 에이전트가 있는 시나리오로의 전환은 도전 과제를 의미한다. 다중 에이전트 시스템과 관련된 고질적인 어려움 중 하나는, 특히 강화 학습을 포함한 대부분의 학습 방법론이 시스템의 방대하고 분산된 구조로 인해 국소 최적해에 빠지는 경향이 있다는 것이다. 더욱이 각 에이전트가 개인 효용을 최대화하는 과정이 전체 시스템 효용의 최대화로 이어지지 않을 가능성도 존재한다. 결과적으로, 개별 작업 중심의 접근 방식은 전체 시스템 성능 측면에서 최적이 아닌 시나리오를 초래할 수 있으며, 이는 대규모 다중 에이전트 환경에서 더욱 두드러진다(Bazzan, 2009). MARL(Multi-Agent Reinforcement Learning)은 지속적인 발전을 통해 알고리즘의 기능과 향상을 이루며 경로 계획 문제를 해결하는 데 상당한 진전을 보였다. 독립 Q-러닝(IQL)에서 시작한 MARL 알고리즘의 진화는 동적 환경에서 에이전트 간의 상호작용을 개선하는 것을 목표로 하고 있다.

IQL: IQL은 Tan et al.(1993)이 제시한 것으로, 에이전트 간의 독립적인 학습을 특징으로 하며 다른 에이전트는 환경의 일부로 간주된다. 즉, IQL의 에이전트는 의사 결정을 할 때 동료 에이전트의 행동이나 전략을 고려하지 않고 자신의 경험에만 기반하여 학습한다. 이 과정에서 각 에이전트는 Boltzmann 분포를 사용하여 행동을 선택하고, 선택된 행동으로부터 얻은 보상을 통해 Q 값을 업데이트한다. 그러나 이 접근법

은 협력이 부족하고 상호 의존적인 목표 달성에 적합하지 않아 부정적인 결과를 초래할 수 있다. 이러한 한계로 인해 IQL을 다중 에이전트 경로 계획에 적용하여 에이전트 간의 협력과 상호작용이 중요한 시나리오에서 성공적인 결과를 도출하기 어려운 경우가 많다.

MADDPG: MADDPG(Multi-Agent Deterministic Policy Gradients)는 IQL의 문제를 해결하기 위해 중앙 집중 훈련(Centralized Training) 및 분산 실행(Decentralized Execution, CTDE) 개념을 도입한 정책 기반 알고리즘이다. Lowe et al.(2017)은 다중 에이전트 환경에서 기존 강화 학습 방법이 비정상적으로 변동하기 때문에 성능 저하를 초래한다는 점을 강조한다. 이를 극복하기 위해 기존의 actor-critic 방법을 확장한 MADDPG를 제안하였다. 이 알고리즘은 훈련 중 크리틱 네트워크가 에이전트 간의 협력을 장려하여 효율적인 학습과 실시간 의사 결정을 균형 있게 수행하도록 돕는다. 크리틱 네트워크는 각 에이전트의 개별 작업뿐만 아니라 모든 에이전트의 상태 및 작업에 대한 중앙 정보를 입력으로 받아, 환경 내에서 수행된 공동 작업에 대한 포괄적인 결과를 제공한다. 훈련 후, 각 에이전트는 중앙 집중 크리틱 네트워크에 접근하지 않고 독립적으로 작동하며, 환경에 대한 로컬 관찰을 바탕으로 행동을 결정한다. Lowe et al.(2017)의 연구에서는 협력적 및 경쟁적 환경에서의 실험 결과, MADDPG가 기존의 IQL 및 AC 기반 다중 에이전트 강화 학습 방법들보다 높은 성능이 확인되었다. Hu et al.(2023)은 정수 프로그래밍(Integer Programming)과 MADDPG 혼합 모델을 사용하여 AGV의 최단 경로를 도출하고 여러 AGV에 대한 충돌 없는 경로를 생성하는 방법을 제안하였다. Qie et al.(2019)은 MADDPG 알고리즘을 이용하여 무인 항공기(UAV) 협업을 위한 제어 시스템을 개발하였다. 그러나 MADDPG는 제한된 탐색성과 비정상성을 내포하고 있는 단점이 있다. MADDPG는 결정론적 정책을 활용하기 때문에 행동 공간을 효과적으로 탐색하는 데 어려움을 겪을 수 있으며, 특정 행동으로 수렴하는 경향이 있어 더 나은 정책의 발견을 저해할 수 있다.

VDN: VDN(Value-Decomposition Networks)은 통합 에이전트 값을 개별 에이전트 값으로 분해하여 위에서 언급한 제한 사항을 해결한 값 기반 알고리즘이다. 각 에이전트는 현재 상태에서 달성 가능한 예상 누적 보상을 추정하여 환경에 미치는 영향을 평가하는 자체 로컬 가치 함수를 유지한다. 통합 Q 값인 Q_{tot}은 개별 에이전트 Q 값을 합산하여 계산된다.

$$Q_{tot}(s,a) = \sum Q(a_i|s_i),\ \forall a \tag{1}$$

수식 (1)은 각 에이전트의 동작이 전체 가치에 미치는 영향을 나타낸다. 여기서 Q(ai|si)는 에이전트i의 관측에 기반하여 학습된 Q 값이다. 각 에이전트가 독립적으로 최적화한 Q값들이 합산되어 시스템의 전체 Q값이 되는 구조이다. 해당 구조는 분산 환경에서 학습을 위한 효과적인 접근 방식이지만, 에이전트 간의 상호작용을 고려하지 않고 값을 단순히 결합하는 문제점이 있다. 따라서 VDN은 비선형적이고 복잡한 다중 에이전트 시나리오를 처리하는 데 어려움을 겪는다.

QMIX: QMIX는 개별 에이전트 값을 유연하게 결합할 수 있는 비선형 혼합 네트워크를 도입하였다. 이러한 비선형 혼합을 통해 QMIX는 에이전트와 작업 간의 복잡한 관계를 포착할 수 있으며, 이를 통해 복잡한 다중 에이전트 시나리오를 해결하는 데 더 효과적이다. Yun et al.(2021)은 실시간으로 다중 드론 택시의 궤적을 최적화하기 위한 MARL의 활용에 대해 논의한다. 많은 수의 전기 수직 이착륙 항공기(eVTOL)를 실시간으로 처리하기 위해서는 중앙 집중식 계산이 불가능하므로, QMIX의 분산 특성이 이러한 요구에 적합하다고 판단하였다. 또한, Wei et al.은 해상 충돌을 피하기 위한 QMIX 기반의 자동 선박 충돌 회피 알고리즘을 제안하였다. 이 방법은 다중 선박 교차 상황에서 항해 경로, 선박의 방향각 및 속도 등을 고려하여 충돌 없는 안전한 항해를 보장한다. QMIX에 대한 보다 자세한 설명은 3장에서 이어질 것이다.

2. 모델 배경지식

모델에 대한 설명을 시작하기에 앞서 이 장은 본 연구의 배경지식을 설명한다.

2.1 QMIX

QMIX는 분산된 정책을 훈련시킬 수 있는 가치 기반 접근 방식이다. 개별 에이전트는 RNN(Recurrent Neural Network)을 사용하며, 특히 게이트형 GRU(Gated Recurrent Unit) 또는 LSTM(Long-Short-Term Memory)과 같은 아키텍처를 사용하여 자체 행동 값 Q_a를 추정한다. QMIX는

$$L(\theta) = \sum_{i=1}^{b}[(y_i \tag{2}$$

을 최소화하도록 학습된다. 여기서 τ∈T는 협동 행동이고, Q_{tot}는 협동 행동 값이며, y_{tot}는

$$y_i^{tot} = r + \gamma \max Q \tag{3}$$

인 TD(Temporal Difference) target이다. 수식 (2)의 목적은 예측한 Q_{tot} 값과 TD target사이의 차이를 최소화하는 값 θ를 찾는 것이다. 수식 (3)은 다음 상태의 가치 혹은 가치 함수의 추정치를 나타낸다. 여기에서 TD는 시간차를 의미하며, 현재 상태에서 시작하여 다음 상태로 진행한 후의 보상과 그 다음 상태에서의 가치 함수 추정치 사이의 차이를 계산한다. 이는 에이전트가 미래 보상을 예측하고 학습 과정에서 최적의 정책을 찾는 데 사용된다. γ은 미래 보상의 중요성을 나타내는 할인 계수이다. QMIX 알고리즘을 설명하기 위해 CTDE의 추가 개념을 덧붙힌다. CTDE에서 CT는 중앙 집중화된 훈련 단계를 의미하며, 이는 공유 정책 함수 또는 가치 함수를 배우는 과정에서 에이전트가 공동으로 학습하는 과정을 말한다. 이 학습 방법은 에이전트가 복잡한 협동 작업을 처리할 때 개별 동작이 전체에 미치는 영향에 대한 이해에 특히 도움이 된다. 에이전트 간 목적이 상반되지 않는 경우, 전체 행동-관측 공간에 대한 확정적인 최적 정책이 보장된다. 즉, 에이전트가 항상 최상의 행동을 선택할 수 있는 경우가 항상 존재한다. 따라서, 기본적으로 CT에서의 확정적인 정책과 DE에서의 확정적인 정책을 일치시키는 것을 목표로 한다(Rashid et al. 2020). DE는 학습이 끝난 후 평가 단계에서 실행한다. 에이전트가 학습을 한 후, 그들은 학습 정책 또는 가치 함수를 기반으로 독립적으로 작동한다. 이 분산 접근 방식은 실제 시나리오에서 확장성 및 적응성을 보장한다.

QMIX 알고리즘에서 CTDE는 학습 과정을 중앙 집중화하는 데 사용된다. 에이전트들은 모든 에이전트 간의 상호작용을 고려한 공동 행동 가치 함수를 학습하며, 이 과정을 수행하는 네트워크를 혼합 네트워크로 지칭한다. 이를 이용하여 QMIX는 개별 에이전트의 행동 가치가 비선형적으로 통합된 혼합 가치를 형성한다. 수식 (4)는 각 에이전트의 전체 혼합 가치에 대한 변화가 그들 자신의 가치 함수가 향상됨에 따라 증가하거나 적어도 비음수여야 한 것을 나타낸다. 이는 QMIX 알고리즘의 핵심 아이디어인 중앙 정책과 분산 정책 간의 일관성을 보장하기 위해 혼합 네트워크 내 가중치를 항상 비음으로 유지 하는 것과 관련 있다.

$$\frac{\partial Q_{tot}}{\partial Q_a} \geq 0, \forall a \tag{4}$$

이 가정이 유지되지 않으면, 에이전트가 다른 에이전트와 협력하지 않는 정책을 학습할 수 있어 전반적인 시스템의 성능을 저해할 수 있다. QMIX는 양의 가중치를 사용하는 비선형 혼합 네트워크를 통해 에이전트가 전반적인 시스템 성능에 기여하도록 장려함으로써, 다중 에이전트 환경에서 더 효과적이고 협력적인 의사 결정을 끌어낸다.

2.2 전이 학습

전이 학습은 다른 데이터셋에서 훈련된 소스 모델이 관련된 타겟 문제를 해결하는 데 유용한 지식과 기능을 습득할 수 있는 아이디어에 기반한다. 전이 학습은 소스 모델이 좋은 초기값을 제공하기 때문에 훈련 중 빠른 수렴을 촉진한다. 소스 모델은 에이전트의 행동 선택을 위한 Q 값을 추정하는 방법을 학습하며, 타겟 모델은 소스 작업과 유사한 상태 공간, 행동 공간, 또는 목표를 가진다. 전이 학습은 다음의 세 가지 주요 특징을 만족할 때 효과적이다:

1. 관련성: 소스 환경과 타겟 환경이 유사한 특징을 공유함
2. 일반화: 모델이 소스 환경에 과적합되지 않음
3. 파인튜닝: 가중치 매개변수가 새로운 환경에 적응하기 위해 일부 조정됨

전이 학습은 훈련 데이터를 처음부터 수집하는 것이 시간이나 비용이 많이 들 때 특히 유용하며, 이러한 특성 덕분에 강화 학습의 시작점으로 기능할 수 있다.

2.3 그래프 이론

그래프는 노드와 이를 연결하는 엣지로 구성된 집합체이다. 그래프는 교통 등 여러 분야에서 관계, 연결 및 구조를 나타내는 데 사용된다. 본 연구에서 사용한 방향 그래프는 엣지에 방향이 지정된 그래프의 한 유형으로, 각 엣지는 시작 노드와 끝 노드를 가지며 방향은 흐름을 나타낸다. 본 연구에서는 에이전트가 한 상태에서 다음 상태로 전환할 때 그래프 내의 엣지를 활용한다. 이 그래프는 약하게 연결된 구성 요소로 분할되며, 각 구성 요소 내에서 사이클이 존재하는지 확인한다. 사이클 내의 에이전트는 서로 충돌이나 교착 상태를 피할 수 있도록 이동을 허용한다. 사이클의 길이가 2인 경우에는 트랜스포터 간의 충돌이 발생하므로 해당 상황을 제외한다. 사이클에 속하지 않는 에이전트는 약하게 연결된 구성 요소 내의 경로를 따라 이동하며, 연결된 요소 내의 모든 트랜스포터가 충돌이나 혼잡 없이 이동할 수 있도록 보장된다. 이러한 접근 방식은 전체 시스템의 효율성을 유지하는 데 기여한다.

2.4 커리큘럼 학습

커리큘럼 학습은 점진적으로 작업이나 복잡성을 증가시키는 학습 전략이다. MARL에서 커리큘럼 학습을 적용하는 목표는 에이전트가 효과적으로 학습하도록 돕는 것이며, 이는 인간이 시간이 지남에 따라 점차 어려운 개념을 학습하는 방식과 유사하다. 커리큘럼 학습은 쉬운 작업부터 시작하여 점차 어려운 작업으로 나아가는 순서를 설계하는 것을 의미한다. 이 방식은 에이전트가 지식을 점진적으로 쌓을 수 있게 하여 기존에 해결하기 어려운 문제를 풀 수 있도록 한다. 협력적 시나리오에서는 에이전트가 기본 협력 작업에서 시작하여 고급 협력이 필요한 복잡한 작업으로 진행할 수 있다. Figure 1은 커리큘럼 학습의 핵심 개념을 설명하며, 전체 데이터셋을 학습 난이도별로 분리하는 데 중점을 둔다. Bengio et al.(2009)은 커리큘럼 학습이 학습 과정에서 수렴 속도뿐만 아니라 비선형 시나리오에서의 국소 최적값의 크기에도 영향을 미친다고 제안하였다. 본 연구에서는 충분히 수렴 가능한 상대적으로 작은 환경에서 커리큘럼 학습을 사용하지 않는다. 그러나 환경의 크기가 증가함에 따라 학습 단계 전반에 걸쳐 수렴 문제가 발생하였고, 초기값 수정을 위해 전이 학습을 시도했음에도 큰 개선이 이루어지지 않았다. 따라서 본 연구는 이러한 문제를 해결하기 위해 대규모 환경에서 커리큘럼 학습의 적용 가능성을 실험하고자 한다.

본 연구는 QMIX 알고리즘을 기반으로 전이 학습과 커리큘럼 학습을 통해 학습의 효율성을 향상시키고, 그래프 이론을 적용하여 트랜스포터 간의 위치 관계를 표현하고 충돌을 방지하는 방식을 도입하였다. 이를 통해 복잡한 환경에서 트랜스포터가 협력하면서도 독립적으로 동작할 수 있는 제어 시스템을 구현하고자 하였다.

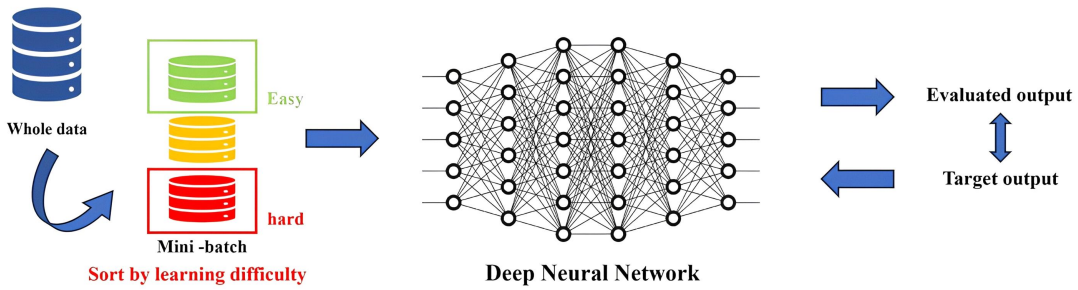

Figure 1. The general idea of curriculum learning

3. 모델 구현

아래는 본 실험 모델의 몇 가지 가정들이다.

1. 중앙 시스템: 에이전트와 통신할 수 있는 단일 중앙 시스템을 가정한다. 이 중앙 시스템에서 현재 환경 내에 존재하는 가장 가까운 작업 좌표가 트랜스포터에게 전달된다.
2. 충전소: 실험 모델에서 트랜스포터의 충전을 고려하지 않는다.
3. 동적 작업 할당: 트랜스포터는 물건을 depot으로 옮긴 후에 돌아오는 도중에 해당 선반에 있는 다른 물건을 운반하는 작업을 할당될 수 있다.
4. 선반 아래로 이동: 트랜스포터가 선반을 이동시키는 작업을 수행하지 않을 때 선반 아래로 통과할 수 있는 로봇임을 가정한다.

상기 가정들은 중앙 시스템의 경우 트랜스포터 센서 탐지 거리의 한계를 보완하기 위해 설정되었으며 환경 전체 정보를 전달하는 것이 아니므로 타당성을 지닌다. 또한 제안된 회피 경로 전략의 효과를 보다 명확하기 확인하기 위해 트랜스포터의 충전을 고려하지 않았다.

3.1 모델 설명

- **실험 환경:** 본 실험 환경은 Christianos et al.(2020)연구 결과를 참고하여 구성하였다. 이 다중 에이전트 환경은 실제 세계의 물류 센터에서 상품을 이동시키는 트랜스포터의 움직임을 본뜬 것이다. 환경은 크기에 따라 세 가지 종류로 구성되어 있으며 특징을 공유한다. 작은, 중간 및 큰 세 가지 유형으로 나뉘며 작은 크기의 환경은 10x6셀로 이루어진 그리드로, 총 12개의 선반을 6개씩 두 그룹으로 나누어 포함하고 있다. 맨 아래에 depot이 있으며, 4개의 에이전트가 협력하여 요청된 선반을 창고로 이동시키고 선반을 원래 위치로 되돌려놓는 일을 반복한다. 선반은 꼭 원래 위치로 돌려놓을 필요는 없으므로 선반을 위치 시킬 수 있는 빈 곳 어디에든 가져다 놓을 수 있도록 설계하였다. 중간 크기의 환경은 Figure 2에 나타나 있는 것처럼 16x10 그리드로 구성되며, 9개 그룹으로 나뉜 총 54개의 선반이 포함되어 있다. 작은 크기의 환경과 유사하게 맨 아래에 창고가 있지만, 이 확장된 지도에서 더 많은 에이전트가 추가되어 모두 8개의 에이전트가 활동한다. 마지막으로 큰 크기의 환경은 16x20

그리드로 이루어진 총 140개의 선반을 14개 그룹으로 나누어 포함하고 있다. 각 환경은 거의 모든 기능을 공유하지만, 지도의 크기만 다를 뿐, 모든 특성을 공유하도록 설계하여 전이 학습을 용이하게 하기 위한 중요한 특징 중 하나인 관련성을 고려하였다.

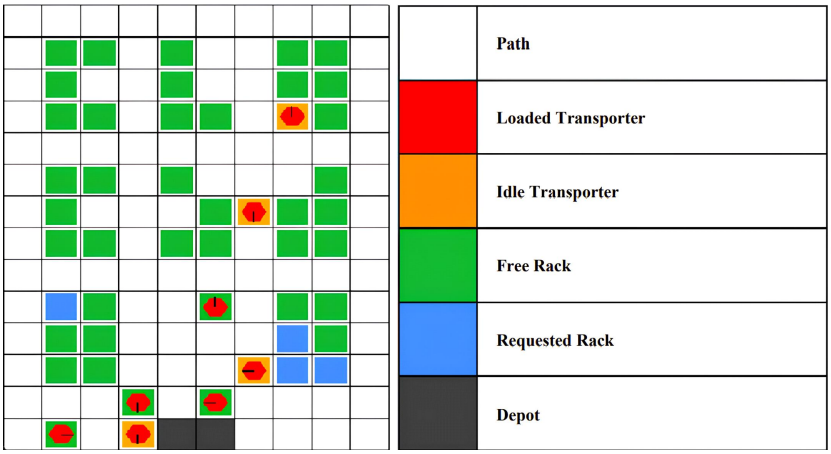

Figure 2. Layout & Description

- 보상 함수: 본 연구에서의 보상 설계는 협력적 행동을 장려하기 위해 모든 트랜스포터가 동일한 보상을 받도록 구성되었으며, 이는 개별 성과보다 전체 시스템의 효율성을 향상시키는 데 중점을 둔다. 구체적으로, 특정 시간 단계에서 특정 트랜스포터가 보상 행동을 수행한 경우, 해당 행동을 수행하지 않은 다른 트랜스포터들 역시 동일한 보상을 받는다. 이러한 보상 메커니즘을 전역 보상(global reward) 전략이라고 정의한다. 전역 보상 전략은 에이전트 간의 협력과 조화를 유도하며, 트랜스포터들이 상호 성과를 통해 학습할 수 있도록 함으로써 보다 원활한 학습 과정과 빠른 수렴을 촉진한다. 그러나 전역 보상 전략은 일부 트랜스포터가 수동적이거나 아무런 행동을 하지 않음으로써 free-riding 문제를 발생시킬 가능성이 있다. 이에 반해, 개별 보상(individual reward) 전략은 트랜스포터들의 적극적인 참여를 유도하고 free-riding 성향을 억제하는 데 초점을 맞추고 있다. 또한, free-riding을 방지하고 교착 상태를 완화하기 위해 일정 기간 동안 아무런 행동을 하지 않은 트랜스포터에게는 음의 보상(negative reward)을 부여하는 전략을 도입하였다. 마지막으로, 보상 부여 간격이 지나치게 길어질 경우 학습 과정이 지연될 수 있으므로, 보상 구조는 트랜스포터들이 단계적으로 보상을 받을 수 있도록 설계되었다. 이를 통해 최종적으로 효과적인 작업 수행을 가능하게 하며, 커리큘럼 학습에서 데이터셋의 난이도에 따라 보상 체계를 조정하는 기준이 된다. 본 연구에서는 보상 구조를 세 가지 범주로 나누어 설계하였다.

  - 로딩 보상: 해당 보상은 트랜스포터가 요청된 선반을 성공적으로 로드할 때 부여된다. 각 트랜스포터는 이 작업에 대해 1의 보상을 받으며, 이는 비교적 간단한 작업에 대한 기준 보상을 제공하여 트랜스포터들이 신속하게 선반을 로드하도록 동기를 부여하기 위함이다.
  - 운반 보상: 해당 보상은 트랜스포터가 로드된 선반을 성공적으로 Depot으로 운반할 때 부여된다. 이 경우 3의 보상이 주어지며, 이는 로드된 선반을 창고로 운반하는 작업의 복잡성이 로딩에 비해 더 높음을 반영하여 설계되었다.
  - 언로딩 보상: 해당 보상은 트랜스포터가 운송을 완료하고 선반을 원래 위치에 돌려놓았을 때 부여된다. 언로딩 작업은 일 주기에서 마지막 단계에 위치하며, 가장 난이도가 높은 보상 행동으로 간주되므로 5의 보상이 주어진다.

강화 학습의 특성상 미래에 대한 보상은 할인 인자 γ 로 할인되어 미래 조치에 대한 기대 보상이 감소한

다. 따라서 본 모델에서는 Depot으로부터 먼 선반에 대한 기대 보상이 그 거리에 비례하여 감소하게 된다. 학습이 진행됨에 따라 트랜스포터는 선반을 Depot에 가까운 위치로 이동시키는 것을 우선시하며, 창고에서 멀리 떨어진 선반에 효율적으로 대응하지 못하는 경향을 시뮬레이션을 통해 확인하였다. 이 문제를 완화하기 위해 본 실험에서는 맨해튼 거리를 기반으로 한 보상 조정을 도입하였다. 이 조정은 Depot으로부터 먼 곳에 있는 선반을 운송하는 데 대한 보상을 강화하여 트랜스포터의 효율적인 운용을 장려한다. 조정된 보상 r_i는 다음과 같이 결정된다:

$$r_i = r(d_i \cdot \omega) \tag{5}$$

수식(5)에서 r은 원래 주어진 보상을 나타내며, d_i는 트랜스포터 i가 운반하는 선반에서 depot까지의 맨해튼 거리를 나타내며, ω는 가중치 요소이다. 이 맨해튼 거리 조정은 로딩 보상과 언로딩 보상 둘 모두에 적용하였다.

- 관찰 공간과 상태 공간: 관찰 공간은 각 트랜스포터의 개별 관찰 값들의 집합으로 구성되며, 이는 자체 데이터와 센서 데이터로 구분된다. 자체 데이터는 트랜스포터의 현재 위치, 이동 방향, 선반 운반 여부, 그리고 주행 가능한 경로 상에 있는지 여부를 포함한다. 센서 데이터는 트랜스포터의 센서 범위 내에서 주변 환경을 감지하는 정보를 제공한다. 이를 통해 트랜스포터는 다른 트랜스포터, 선반, 그리고 작업이 요청된 선반의 위치를 인식할 수 있으며, 인접 트랜스포터의 이동 방향에 대한 정보를 얻을 수 있다. 관찰 공간의 주요 과제 중 하나는 트랜스포터가 환경 정보를 수집하기 위해 반드시 센서 범위 내에 있어야 한다는 점이다. 따라서 특정 위치에 트랜스포터가 존재하지 않을 경우, 해당 위치에 있는 작업 요청 선반의 존재를 감지하지 못해 해당 작업이 수행되지 않을 수 있으며, 이로 인해 전체 시스템의 효율성이 저하될 수 있다. 센서 범위를 확장하는 방법을 실험적으로 고려할 수 있으나, 이 경우 요구되는 데이터셋의 크기가 기하급수적으로 증가하여 시스템 부하 측면에서 비효율적이라는 결론을 내렸다. 이에 따라, 본 연구에서는 중앙 시스템이 모든 환경 정보를 접근할 수 있도록 하고, 트랜스포터의 관찰 공간에 실시간으로 가장 근접한 선반의 위치를 제공하는 해결책을 제시하였다. QMIX의 혼합 네트워크에서 사용되는 상태 정보는 모든 트랜스포터의 관찰 공간을 단일 벡터로 연결한 형태로 정의된다. 이러한 상태 정보는 여전히 전체 환경에 대한 부분 관측에 의존하지만, 트랜스포터들이 상호 간의 정보 공유 및 통신이 가능한 상황에서 학습이 진행되므로 효과적인 해결책이 될 수 있다.

- 트랜스포터: 본 모델에서 트랜스포터는 네 가지 주요 행동을 수행할 수 있다: 정지, 앞으로 이동, 왼쪽으로 회전, 그리고 오른쪽으로 회전. 트랜스포터는 1단위의 센서 범위를 가지고 있으며, 이를 통해 자신의 위치에서 1단위 거리 내의 8개 인접 그리드에서 환경 정보를 수집한다. 본 연구에서 사용된 트랜스포터는 아마존의 키바(Kiva) 로봇과 유사한 방식으로 작동한다. 트랜스포터는 밀집된 그리드 기반의 물류 센터에서 제품이 적재된 선반을 들어 올려 운반하는 역할을 담당하며, 이러한 선반은 재고가 배치된 밀집된 배열로 저장된다. 트랜스포터는 무거운 선반을 처리하고 운송하는 작업을 수행하며, 인간 근로자는 depot에서 피킹(picking) 및 패킹(packing)과 같은 작업을 관리한다.

3.2 알고리즘 적용

트랜스포터는 과거 데이터를 참조하여 다음 행동을 선택하는 데 도움을 주기 위해 GRU(Gated Recurrent Unit) 셀로 구성된 자체 네트워크를 갖추고 있다. 실제 산업 현장에서는 계산 자원을 절약하기 위해 비용 효율적인 GRU 셀을 사용하는 것이 실용적일 수 있다. 그러나 계산 자원이 충분한 경우, LSTM(Long Short-Term Memory) 셀로 전환하면 더 많은 과거 정보를 유지하고, 더 나은 성능을 기대할 수 있다.

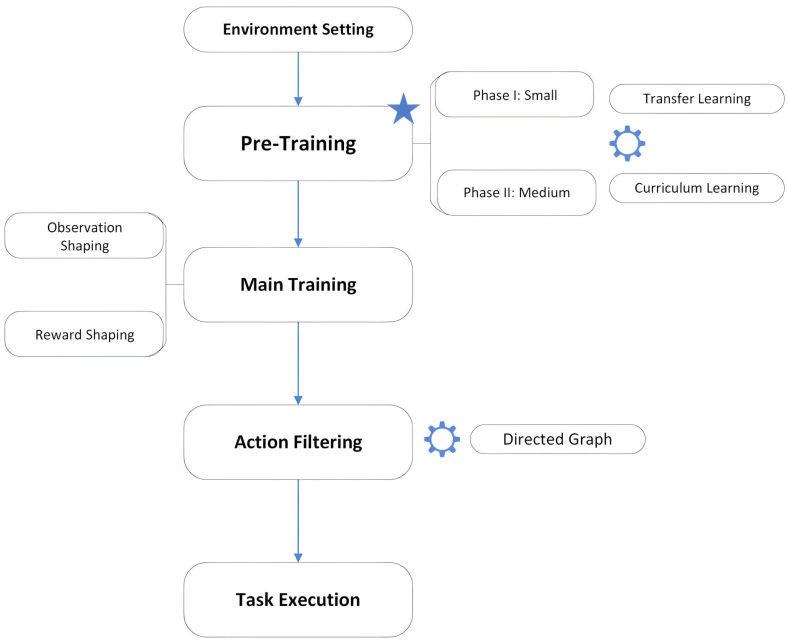

Figure 3. The flow chart representation of the key stages

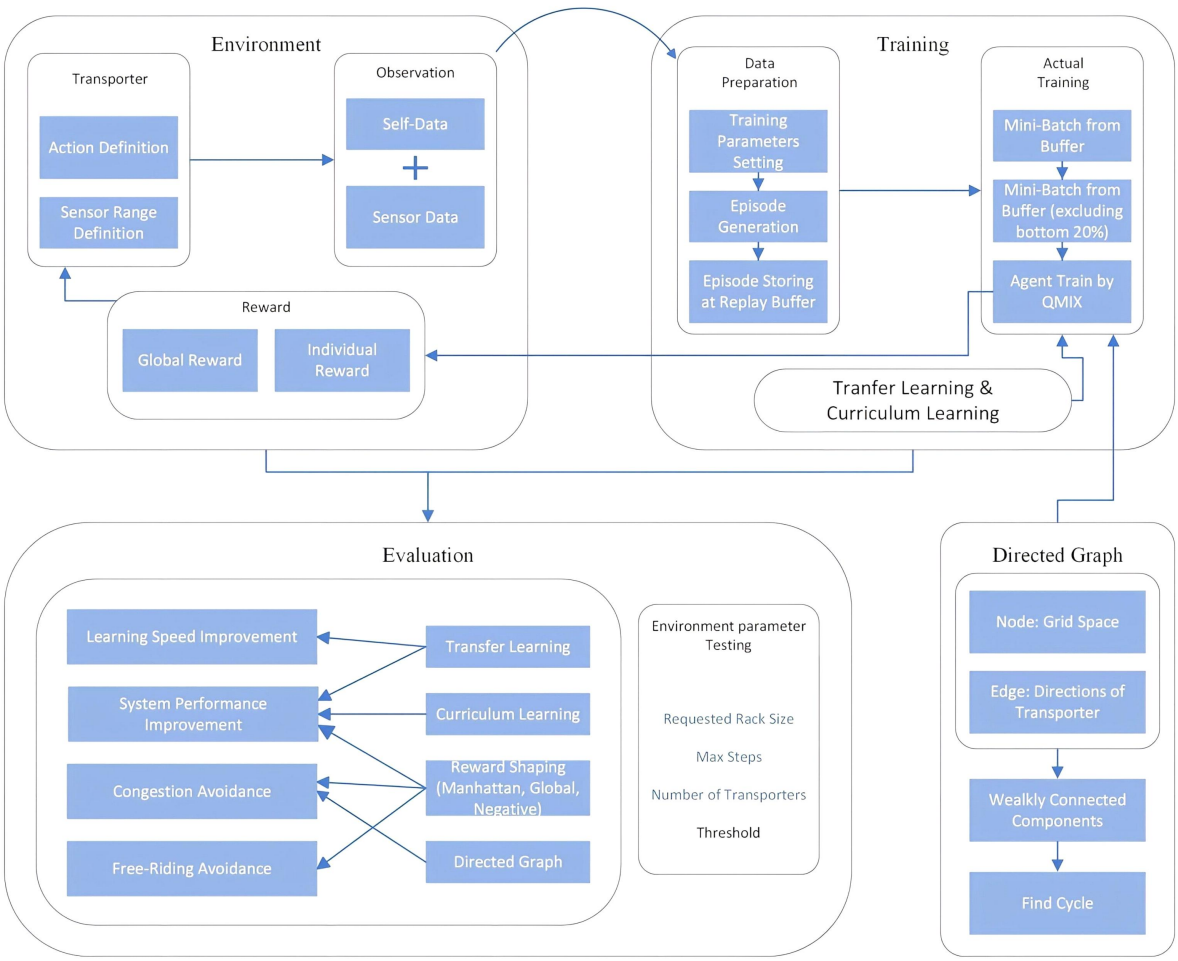

Figure 4. Block diagram illustrating the whole process

모델은 총 세 개의 레이어로 구성된 단순한 구조를 가지고 있다. 첫 번째 레이어는 선형 네트워크로, 두 번째 레이어는 GRU 셀이며, 세 번째 레이어는 또 다른 선형 네트워크로 이루어진다. 트랜스포터의 관찰 데이터가 입력으로 들어오며, 마지막 선형 네트워크의 출력은 각 행동에 대한 Q 값을 제공한다. 첫 번째 선형 네트워크의 입력은 ReLU(Rectified Linear Unit) 활성화 함수를 통해 처리된다. ReLU로 처리된 관측 값과 hidden state는 GRU 셀로 전달된다. Hidden state는 순환 신경망에서 중요한 개념으로, 네트워크의 메모리 역할을 하여 이전 정보를 현재 계산에 활용함으로써 시간적 의존성을 파악하는 데 기여한다. 초기 숨겨진 상태 값은 학습 에피소드 간의 독립성을 보장하기 위해 0으로 초기화된다. 혼합 네트워크는 하이퍼 네트워크를 사용하는 비선형 접근 방식을 채택하고 있다. 본 연구의 모든 실험에서 사용된 QMIX 및 채택한 학습 전략에 대한 의사코드는 부록의 알고리즘 1에서 제공된다.

3.3 학습 전략

모델이 다양한 상황을 학습하고 과적합을 줄이며 일반화 능력을 향상시키기 위해, 모든 실험은 초기 탐색을 허용하는 epsilon 값을 1로 설정하였다. 강화 학습에서 탐색은 환경을 학습하기 위해 새로운 행동을 시도하는 것을 의미하며, 활용은 즉시 보상을 극대화하기 위해 학습을 통해 얻은 최적 행동을 선택하는 것을 의미한다. 이러한 두 가지 요소 간의 적절한 균형을 맞추는 것은 학습의 성공에 중요한 역할을 한다.

초기 탐색 단계에서 생성된 에피소드는 에포크가 진행됨에 따라 반복적으로 사용하기 위해 리플레이 버퍼에 저장된다. 이후 각 에포크에서는 epsilon 값을 점진적으로 감소시키는 epsilon 감소 과정을 적용하였다. 학습 초기 단계에서 적은 수의 에피소드로 인한 과적합 문제를 방지하기 위해, epsilon 값이 1인 상태에서 960개의 에피소드를 생성하여 리플레이 버퍼를 적절히 채웠다. 리플레이 버퍼는 최대 버퍼 크기에 도달하면 가장 오래된 에피소드를 순차적으로 제거하는 방식으로 관리된다. 양질의 에피소드를 높은 빈도로 학습하기 위해, 리플레이 버퍼에 저장된 에피소드 중 하위 20%를 제외한 나머지 80%에서 무작위로 샘플링하여 배치를 생성하였다. 이러한 접근 방식은 과적합 문제의 위험을 수반할 수 있으므로, 이를 완화하기 위해 버퍼의 모든 에피소드에서 무작위 샘플링하는 방법과 결합하여 강건한 모델 구축을 목표로 하였다. Figure 3은 학습 흐름을 개략적으로 나타내는 플로 차트이며, Figure 4는 세부적인 학습 전략을 설명하는 블록 다이어그램으로, 학습 과정에서 환경과의 상호작용을 나타낸다.

4. 수치 실험

실험의 주요 목표는 본 시스템에서 학습의 성능 및 효율성을 향상시키기 위한 다양한 전략의 영향을 평가하는 것이며, 이는 본 연구에서 설정한 연구 질문(RQ)에 대한 답변을 제공한다. 구체적으로, 본 연구는 전이 학습, 맨해튼 거리를 활용한 보상 조정, 대규모 환경에서의 커리큘럼 학습 효과, 그리고 혼잡 완화를 위한 방향 그래프의 활용에 중점을 두었다. 또한, 본 연구에서는 교육 진행 상황을 모니터링하고 교육 완료 후 시스템의 작동 상태를 확인하기 위해 시각적 시뮬레이션을 생성하였다. 이 시뮬레이션은 2D 및 3D 맵 제작을 위한 파이썬 오픈소스 라이브러리인 pyglet을 사용하여 구현되었다. Figure 5에서는 시뮬레이션을 통해 각 시간 단계에서 트랜스포터가 어떻게 움직이고 협력하는지, 그리고 완료된 작업 수와 같은 주요 성능 지표(KPI)를 추적하는 과정을 보여준다. Table 1은 실험에서 사용된 하이퍼파라미터를 정리한 표이다. 본 연구에서 주목할 만한 파라미터 설정은 감마(γ) 값이 일반적으로 사용되는 0.99가 아닌 0.9로 설정되었다는 점이다. 감마 값을 낮추면 에이전트가 보상을 더 빨리 얻기 위해 노력하게 되어, 결과적으로 더 빠른 학습이 가능할 수 있다. 또한, 감마 값을 낮추는 것은 즉각적인 보상에 대한 중요성을 증가시키고, 에이전트가 장기적인 불확실성을 고려하는 정도를 줄이는 효과를 가져온다. 이는 불확실한 환경에서 안정성을 향

상시키고, 보다 견고한 모델을 만드는 데 기여할 수 있다. 그러나 감마 값을 낮추는 것은 특정 상황이나 환경에서 성능 저하를 초래할 수 있는 trade-off 관계가 존재하므로, 본 연구에서는 감마 값을 보수적으로 설정하기로 결정하였다. Table 2는 각 연구 질문과 이와 관련된 실험에 대한 설명을 제시한다.

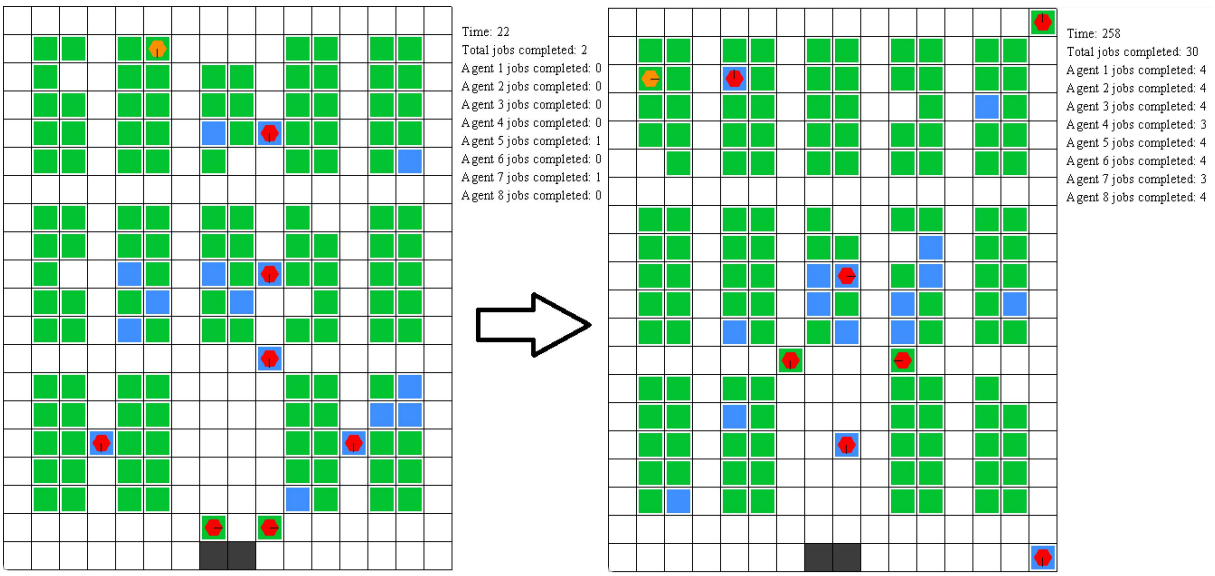

Figure 5. Visual simulation of the environment setting

Table 1. Experimental parameter

Parameter	Value
RNN hidden dimension	64
QMIX hidden dimension	32
Discount factor γ	0.9
Optimizer	Adam
Learning rate	0.001
Initial epsilon	1
Minimum epsilon	0.001
Number of steps per episode	300
Number of episodes per epoch	50
Number of training steps per epoch	300
Evaluate cycle	100
Batch size	64
Replay buffer size	5000
Save cycle (per training steps)	150
Target network update cycle	300

Table 2. Brief overview of the numerical study

RQ	Experiment Objective	Experiment Details
RQ1	Effect of Manhattan Distance Reward Design on System Performance	Simulate the impact of Manhattan distance-based rewards on learning and evaluate overall system performance using average episode reward values.
RQ1	Impact of Negative Rewards on System Performance Improvement	Analyze how the prevention of free-riding contributes to system performance improvement and perform a t-test to assess intergroup utility.
RQ1	Effect of Curriculum Learning on Mega DC System Performance	Approach previously unsolvable problems by incrementally increasing the difficulty level of training data.
RQ2	Impact of Transfer Learning on Training Speed	Compare training speeds before and after applying transfer learning, measure the average percentage increase in training speed, and validate effectiveness through a t-test.
RQ2	Effect of Global Rewards on Convergence Speed Improvement	Analyze the differences in system convergence among global rewards, individual rewards, and mixed rewards, and conduct an ANOVA test to evaluate the utility of the three groups.
RQ3	Effect of Directional Graphs on Congestion Prevention	Verify the effectiveness of directional graphs by comparing heatmaps before and after application.

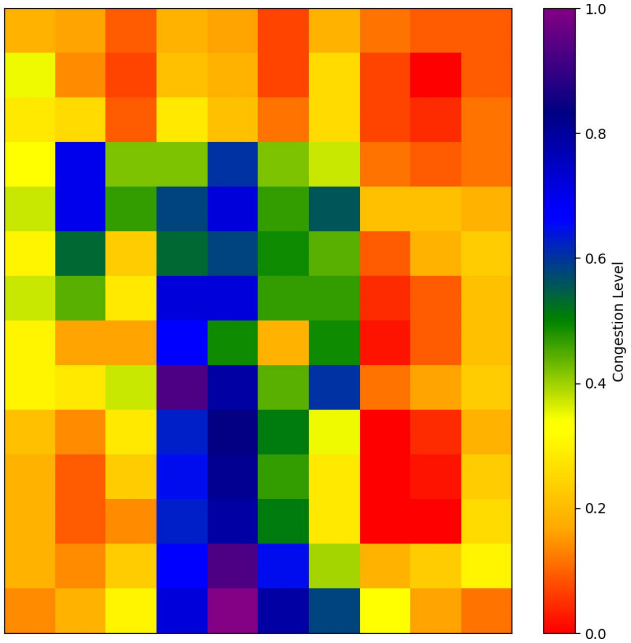

Figure 6. Transporter heatmap

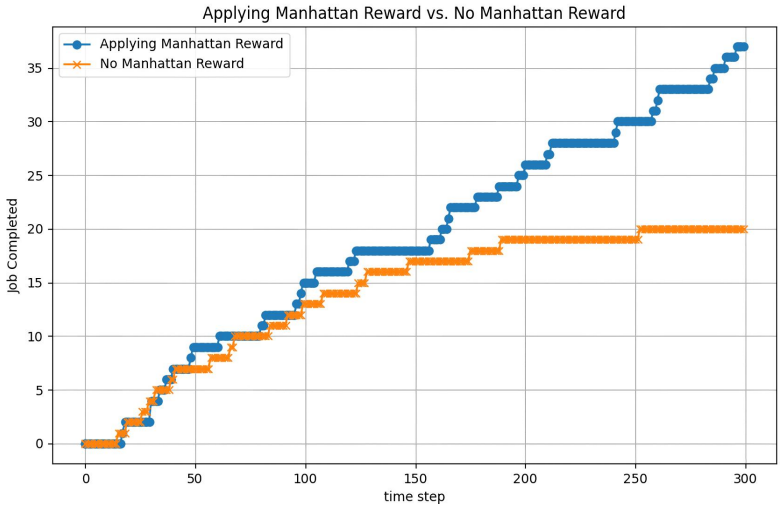

Figure 7. The effect of Manhattan distance reward

4.1 시스템 성능은 무엇과 관련이 있으며, 어떻게 향상 시킬 수 있는가?

MARL에서 시스템 성능은 여러 요소와 관련이 있으며 대체적으로 보상 설계, 에이전트 간 통신을 통한 상태 공유, 알고리즘의 성능 등이 중요하다고 알려져 있다. 시각적 시뮬레이션을 통해 시스템 성능 저하의 원인들로 보이는 것들에 접근하는 방식으로 실험을 설계하였다.

1) 실험 I: 맨해튼 거리 보상 설계가 시스템 성능에 미치는 영향

본 연구는 8개의 에이전트를 사용하여 맨해튼 거리 보상의 효과를 관찰하기에 적합한 대규모 환경에서 실험을 수행하였다. Figure 6은 맨해튼 거리를 적용하기 전 에이전트의 방문 횟수에 따른 히트맵을 나타내며, 이로 인해 Depot 근처에서만 활동이 집중되는 경향을 확인할 수 있다. 본 연구에서는 로딩 및 언로딩 보상에 맨해튼 거리 기반의 보정을 적용하였으며, 이러한 조정은 더 먼 위치에 있는 요청된 선반에 대한 보상을 적절하게 증가시켰다. 이를 통해 작업이 주로 더 가까운 Depot 주변에 집중되는 것을 방지하고, 전체 시스템의 효율성을 향상시키는 결과를 시뮬레이션에서 확인하였다. 충분한 학습이 이루어지고 시스템 성능이 일정 수준으로 수렴한 후, 맨해튼 거리를 사용한 경우와 그렇지 않은 경우의 훈련 결과 간 시스템 성능 차이는 Figure 7에 제시되어 있다. 맨해튼 거리를 사용한 경우, 무작위 시나리오에 대해 더 강건한 성능을 보였으며, 수렴 후 평균 보상 값 또한 더 높은 것을 확인하였다. 반면, 맨해튼 거리를 적용하지 않은 경우, 시뮬레이션 결과 Depot 근처에서 교통 혼잡이 빈번하게 발생하는 경향을 보였다. 두 방법을 사용하여 훈련된 네트워크로부터 생성된 37개 에피소드의 평균 보상 값에 대한 t-검정 결과는 Table 3과 같다:

Table 3. T-test result & average percetage increase for Manhattan distance

t-statistic	**26.9982**
p-value	2.1219×10^{-39}
average increase (%)	39.8 %

2) 실험 II: 음의 보상이 시스템 성능에 미치는 영향

실험은 중간 크기의 환경에서 8대의 트랜스포터를 사용하여 진행하였다. Figure 8의 실험은 전역 보상의

단점으로 지적된 free-riding 문제를 방지하기 위해 설계되었다. 일정 시간 동안 정지해 있는 트랜스포터에게 패널티를 부여하여 해당 행위를 하지 않도록 학습시키는 방식이다. 음의 보상을 부여하는 것은 의도하지 않은 방향으로 학습이 진행될 가능성이 있으므로, 다른 기법들과 병행하여 사용하였다. 이러한 기법에는 맨해튼 거리 보상, 이후 설명할 전역 보상, 그리고 방향 그래프가 포함된다. 결과적으로, 본 연구는 free-riding 문제를 해결하여 시스템 성능의 개선을 가져온 결과를 보여주었다. 주황색 데이터는 작업 완료 수가 약 30에서 정체되는 모습을 나타내는데, 이는 Depot 근처에서 free-riding을 하는 트랜스포터의 영향을 반영한다. 이러한 경로를 차단하는 행위를 방지함으로써, 정상적인 시스템 운용이 가능해졌음을 파란색 데이터에서 확인할 수 있다.

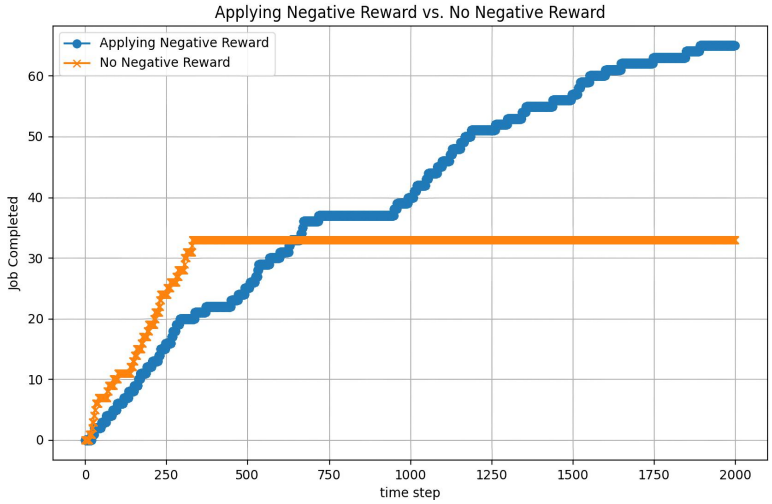

Figure 8. Penalty reward for preventing free-riding

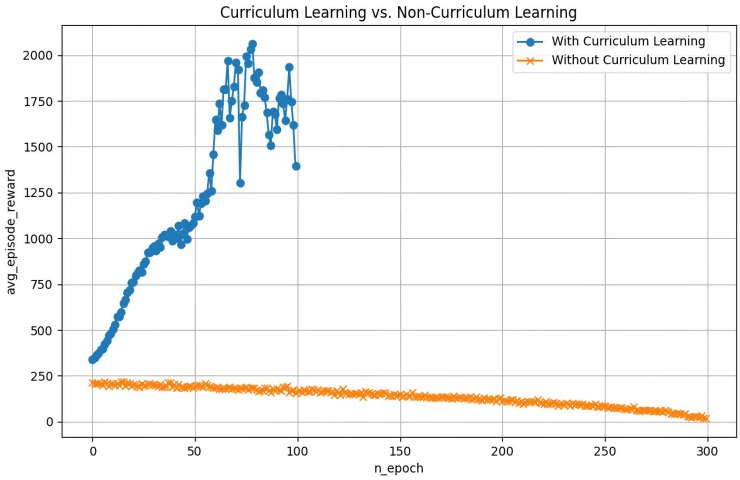

Figure 9. The effect of curriculum learning in the mega DC size environment

3) 실험 III: 커리큘럼 학습이 메가DC 시스템의 성능에 미치는 영향

본 연구의 궁극적인 목적은 다양한 다중 에이전트 강화 학습(MARL) 기법과 여러 학습 방법을 활용하여 현실의 메가 DC 환경에서 다수의 트랜스포터를 효과적으로 운용하는 것이다. 이를 위해 메가 DC라고 정의할 수 있는 대규모 환경으로의 확장을 실험적으로 검증하였다. 학습 기법을 적용하지 않고 학습한 결과

는 Figure 9의 오렌지색 데이터로 나타나 있으며, 에피소드를 난이도별로 분류하여 세 차례에 걸쳐 학습한 결과는 파란색 데이터로 표시되었다. 형평성을 고려하여 동일한 에포크 동안 학습을 진행하였고, 그 결과로 눈에 띄는 성능 향상이 관찰되었다. 비록 100 에포크 내에서 수렴하는 모습을 보이지는 않지만, 기존에 전혀 학습이 이루어지지 않은 상황과의 비교를 통해 커리큘럼 학습이 효과적임을 입증하였다.

4.2 긴 학습 시간 문제를 어떻게 개선 하는가?

강화 학습의 가장 큰 단점 중 하나는 초기 모델의 학습 시간이 지나치게 오래 소요된다는 점이다. 강화 학습에서는 강건한 모델을 구축하기 위해 에이전트가 초기 단계에서 환경을 광범위하게 탐색하고 경험을 축적해야 하며, 이는 네트워크 수렴 속도와 상충되는 개념이다. 특히, 메가 DC와 같은 대규모 데이터를 활용하는 문제에서는 학습을 위해 더 많은 데이터를 생성해야 하므로 학습 시간이 현저하게 증가하는 문제를 초래한다. 이러한 문제를 해결하기 위해, 학습한 경험을 재사용하는 리플레이 버퍼와 사전 훈련(pre-training) 방법이 제안되었다. 리플레이 버퍼의 효과는 이미 여러 연구에서 검증되었으므로, 본 연구에서는 사전 훈련의 효과에 대해 실험을 통해 검증하고자 한다.

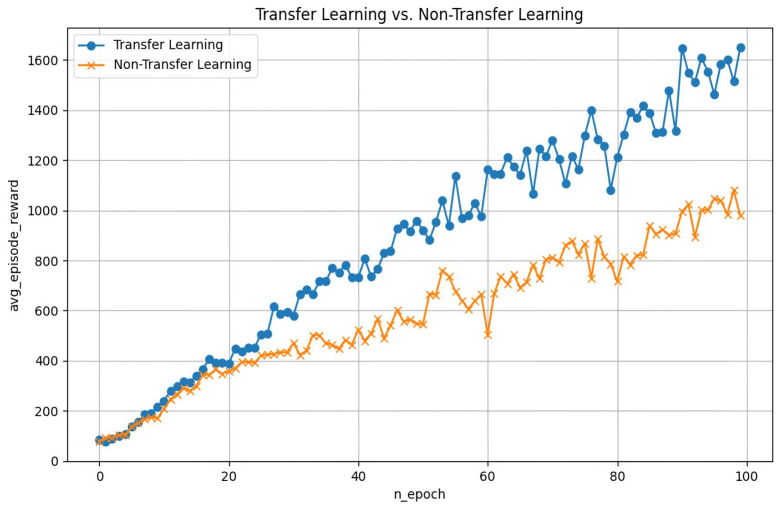

Figure 10. The effect of pre-training

1) 실험 IV: 전이 학습이 학습 속도에 미치는 영향

전이 학습을 통한 사전 훈련은 다중 강화 학습에서 학습 속도에 중요한 영향을 미치는 것으로 나타났다. 이는 모든 최적화 기법에서 좋은 초기 해를 가질 경우 빠르게 수렴하는 원리와 유사하다. 본 연구에서는 전이 학습의 영향을 탐색하기 위해 대규모 환경에서 8개의 에이전트를 활용한 실험을 수행하였다. 중간 규모 환경에서 모델을 훈련한 후, 해당 모델의 미리 훈련된 네트워크 매개변수를 대규모 환경에서의 훈련을 위한 초기값으로 사용하였다. 이후, 전이 학습을 적용하지 않고 동일 규모의 환경에서 훈련된 모델과 성능을 비교하였다. Figure 10은 학습이 진행됨에 따라 평균 에피소드 보상이 증가하는 모습을 보여준다. 전이 학습 적용 전후의 학습 속도를 비교하고 관찰된 개선의 통계적 유의성을 검증하기 위해 t-검정을 사용하였다. 구체적으로, 전이 학습 적용 후 평균 보상의 백분율 증가를 측정하였다. 유의수준이 0.05인 경우, t-검정 결과는 다음과 같다:

Table 4. T-test result & average percetage increase for transfer learning

t-statistic	5.6499
p-value	5.5277×10^{-8}
percentage increase (%)	49.9%

Table 4에서 관찰된 성능 향상은 학습 속도를 49.9% 높이는 결과를 나타냈다.

2) 실험 V: 전역 보상이 학습 수렴 속도에 미치는 영향

실험은 중간 크기의 환경에서 8대의 트랜스포터를 사용하여 진행되었다. Figure 11의 실험에서는 시간 제약으로 인해 에피소드의 길이를 100으로 설정하여 진행하였다. 해당 실험에서 전역 보상 체계가 가장 안정적인 학습 추이를 보이는 것으로 확인되었다. 그러나 학습이 완료된 후 ANOVA 분석 결과, 그룹 간의 평균 차이가 유의 수준 0.05에서 통계적으로 유의하지 않다는 결과가 나타났다. 이는 짧은 에피소드 길이로 인해 학습 데이터셋 내에서 언로딩 보상이 수행된 사례가 매우 적었기 때문이라고 판단된다. 따라서 에피소드 길이를 300으로 늘린 상태에서 동일한 조건으로 실험을 재진행하였다. 해당 실험의 결과는 Figure 12에 나타나 있다. 예상한 대로, 전역 보상 체계가 가장 빠른 속도로 수렴하는 모습을 보여주었다. 주목할 점은 두 가지 방법을 혼합한 mixed 보상 체계가 학습 안정성 측면에서도 활용 가능성을 나타냈다는 것이다. 학습 초반에는 빠른 수렴과 시스템 전체 성능 향상에 초점을 두어 전역 보상 체계를 채택하고, 학습 후반에는 mixed 또는 개별 보상 체계를 채택하여 전역 보상 체계의 단점인 free-riding 문제와 에이전트 행동의 정교함을 향상시키는 방법을 실험해볼 수 있을 것이다.

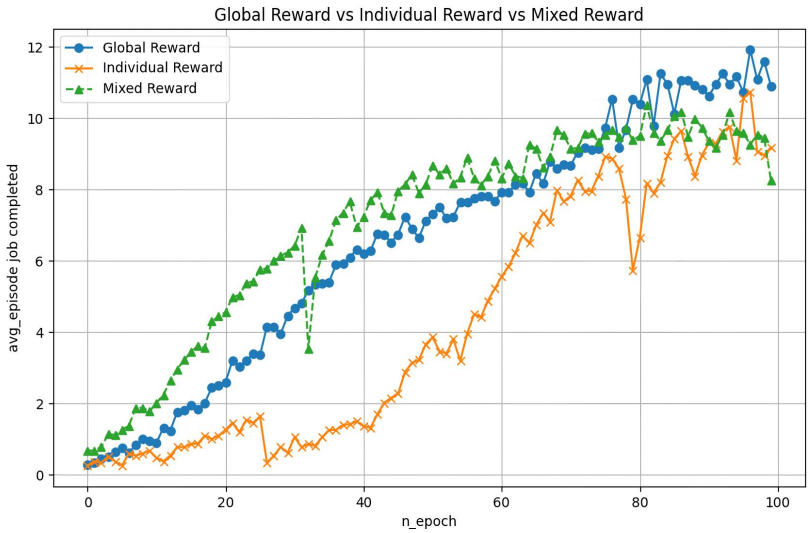

Figure 11. 100 step in an episode

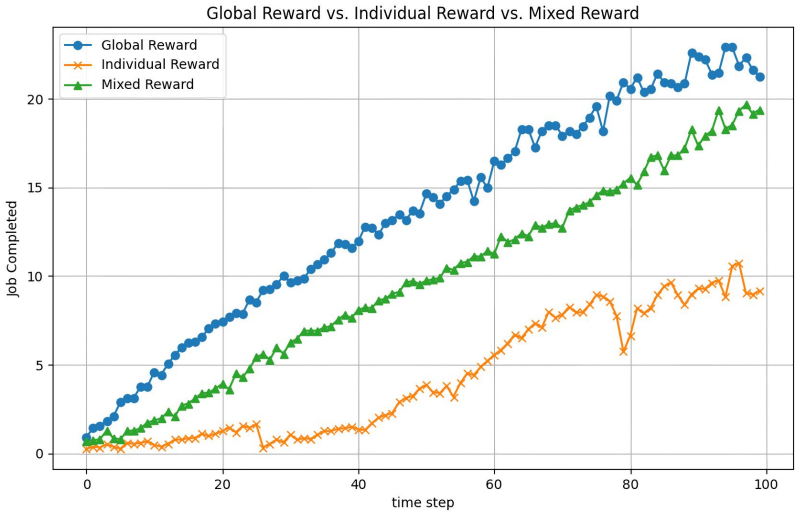

Figure 12. 300 steps in an episode

4.3 혼잡 또는 교착 상태와 같은 트랜스포터의 이동과 관련된 문제를 어떻게 해결할 수 있는가?

복잡한 작업 환경에서 다수의 트랜스포터를 운용할 때 가장 경계해야 할 문제는 교착 상태이다. 본 연구에서는 충돌 가능성 또는 혼잡 유발 가능성이 있는 경로를 사전에 식별하기 위한 다양한 방법 중, 그래프 이론을 활용하여 경로상에 존재하는 트랜스포터의 다음 행동을 조정함으로써 충돌, 혼잡, 및 교착 상태에 효과적으로 대응하고자 한다.

1) 실험 VI: 방향 그래프가 혼잡과 교착 상태 방지에 미치는 영향

복잡한 작업 환경을 구현하기 위해, 중간 크기의 환경에서 트랜스포터의 수를 24대로 증가시킨 상황을 가정하였다. 교통흐름에 따른 정차 상태와 교착 상태를 구분하기 위해, 특정 지역에 지속적으로 머무르는 경우 머무는 시간이 일정 기준을 초과하면 해당 위치를 heatmap에 표시하는 방식을 적용하였다. 이를 통해 그래프 적용 전후의 heatmap을 비교하여 방향 그래프의 실제 효과를 평가하였다. 극한의 작업 환경을 가정하였기 때문에 두 가지 케이스 모두 비효율적인 작업 수행을 보였으나, Figure 13에서 확인할 수 있듯이 방향 그래프를 적용하지 않은 경우 교착 상태가 발생하였다. Heatmap에서 보라색에 가까운 색상은 혼잡이 발생한 지역을 나타내며, 교착 상태는 트랜스포터가 한 대만 지나갈 수 있는 좁은 통로에서 자주 발생하였다. 이는 매우 복잡한 환경에서 트랜스포터가 막다른 위치에서 정차하여 서로 이동할 수 없는 deadlock 현상이 발생했음을 보여준다. 방향 그래프의 연결된 요소 집합을 사용함으로써 각 사이클에 속한 트랜스포터를 식별할 수 있으며, 사이클 내의 에이전트는 서로 충돌하거나 교착 상태의 위험을 일으키지 않으므로 이동을 허용하여 후속 혼잡 및 교착 상태를 방지하는 효과를 얻을 수 있다. Figure 14에서는 대부분의 경우 넓은 중앙 복도를 이용하는 모습을 확인할 수 있으며, 이는 사이클에 따라 한 칸씩 이동하는 전략이 혼잡한 상황에서 효과적임을 나타낸다.

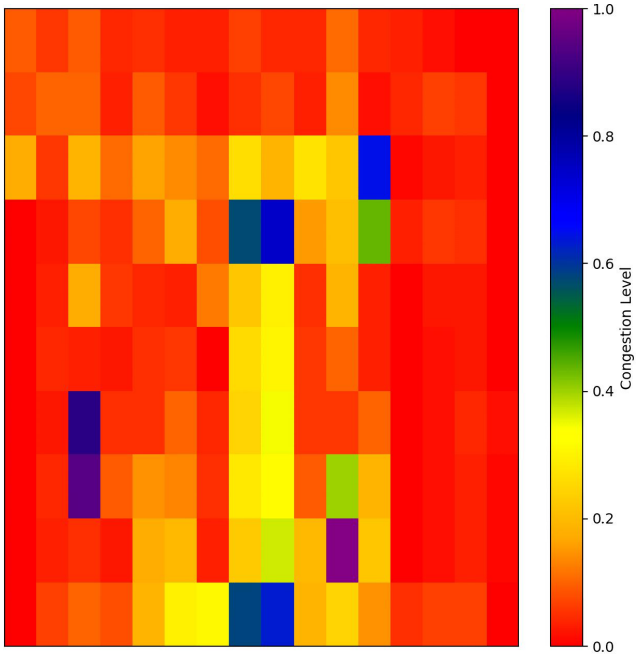

Figure 13. Heatmap without graph exhibit signs of deadlock

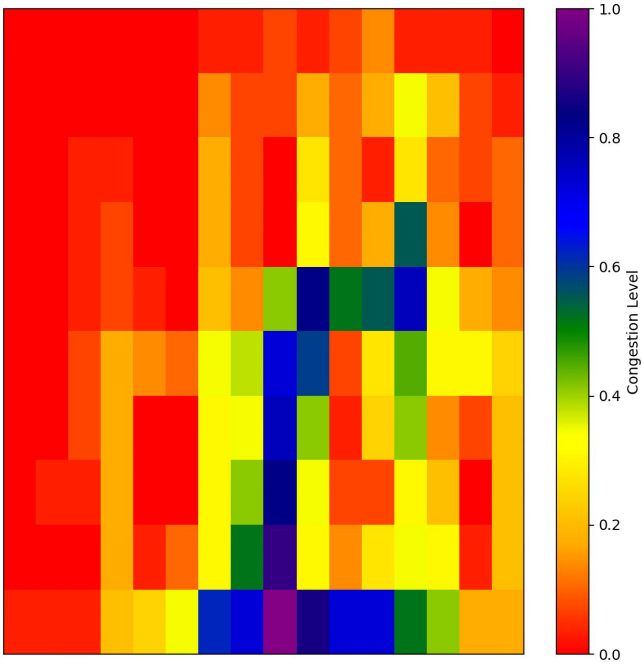

Figure 14. Directed graph heatmap

5. 결론

본 연구는 다양한 학습 기법을 적용하여, 메가 DC와 같은 대규모 창고 환경에서의 MARL 기반 운영 가능성을 다수의 실험을 통해 검증하였다. 이를 통해 전체 시스템 성능 향상과 학습시간 단축을 목표로 설정하였으며, 구체적인 실험 결과를 통해 해당 목표의 실현 가능성을 입증하였다. 그러나 현재 수준에서는 실

제 응용에 한계가 존재하며, 이러한 한계는 하이퍼파라미터 튜닝 및 보다 정교한 환경 상태 정보를 활용함으로써 극복할 수 있을 것으로 판단된다. 향후 연구에서는 멀티프로세싱을 통한 에피소드 생성 및 학습 가속화 전략을 고려할 예정이며, 효과적인 관측 및 상태 공간 정보 구성 역시 주요 연구 주제가 될 것이다. 이러한 연구는 대규모 환경에서 MARL의 적용 효율성을 향상시키는 데 중요한 기여를 할 것으로 기대된다. 본 연구는 기존 연구들이 다루지 못한 대규모 환경에서의 운용 가능성을 제시하였으며, 시스템 성능을 개선하기 위한 다양한 기법들과 그 성능에 영향을 미치는 주요 요인들의 효과를 실험적으로 규명하였다는 점에서 중요한 시사점을 제공한다.

참고문헌

Bazzan, A. L. (2009). Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. Autonomous Agents and Multi-Agent Systems, 18, 342-375.

Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum learning. In Proceedings of the 26th annual international conference on machine learning (pp. 41-48).

Boutsioukis, G., Partalas, I., & Vlahavas, I. (2012). Transfer learning in multi-agent reinforcement learning domains. In Recent Advances in Reinforcement Learning: 9th European Workshop, EWRL 2011, Athens, Greece, September 9-11, 2011, Revised Selected Papers 9 (pp. 249-260). Springer Berlin Heidelberg.

Chen, X., Kong, Y., Fang, X., & Wu, Q. (2013). A fast two-stage ACO algorithm for robotic path planning. Neural Computing and Applications, 22, 313-319.

Chen, Z., Alonso-Mora, J., Bai, X., Harabor, D. D., & Stuckey, P. J. (2021). Integrated task assignment and path planning for capacitated multi-agent pickup and delivery. IEEE Robotics and Automation Letters, 6(3), 5816-5823.

Christianos, F., Schäfer, L., & Albrecht, S. (2020). Shared experience actor-critic for multi-agent reinforcement learning. Advances in neural information processing systems, 33, 10707-10717.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. IEEE transactions on systems, man, and cybernetics, part b (cybernetics), 26(1), 29-41.

Ebben, M., van der Heijden, M., Hurink, J., & Schutten, M. (2005). Modeling of capacitated transportation systems for integral scheduling. Container Terminals and Automated Transport Systems: Logistics Control Issues and Quantitative Decision Support, 287-306.

Fazlollahtabar, H., & Saidi-Mehrabad, M. (2015). Methodologies to optimize automated guided vehicle scheduling and routing problems: a review study. Journal of Intelligent & Robotic Systems, 77, 525-545.

Fragapane, G., De Koster, R., Sgarbossa, F., & Strandhagen, J. O. (2021). Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. European Journal of Operational Research, 294(2), 405-426.

Gendreau, M., Guertin, F., Potvin, J. Y., & Taillard, É. (1999). Parallel tabu search for real-time vehicle routing and dispatching. Transportation science, 33(4), 381-390.

Hu, H., Yang, X., Xiao, S., & Wang, F. (2023). Anti-conflict AGV path planning in automated container terminals based on multi-agent reinforcement learning. International Journal of Production Research, 61(1), 65-80.

Hussein, A., Mostafa, H., Badrel-din, M., Sultan, O., & Khamis, A. (2012). Metaheuristic optimization approach to mobile robot path planning. In 2012 international conference on engineering and technology (ICET) (pp. 1-6). IEEE.

Lowe, R., Wu, Y. I., Tamar, A., Harb, J., Pieter Abbeel, O., & Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. Advances in neural information processing systems, 30.

Lei, X., Zhang, Z., & Dong, P. (2018). Dynamic path planning of unknown environment based on deep reinforcement learning. Journal of Robotics, 2018.

Oliehoek, F. A., Spaan, M. T., & Vlassis, N. (2008). Optimal and approximate Q-value functions for decentralized POMDPs. Journal of Artificial Intelligence Research, 32, 289-353.

Oliehoek, F. A., & Amato, C. (2016). A concise introduction to decentralized POMDPs (Vol. 1). Cham, Switzerland: Springer International Publishing.

Panov, A. I., Yakovlev, K. S., & Suvorov, R. (2018). Grid path planning with deep reinforcement learning: Preliminary results. Procedia computer science, 123, 347-353.

Papoudakis, G., Christianos, F., Schäfer, L., & Albrecht, S. V. (2020). Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. arXiv preprint arXiv:2006.07869.

Qie, H., Shi, D., Shen, T., Xu, X., Li, Y., & Wang, L. (2019). Joint optimization of multi-UAV target assignment and path planning based on multi-agent reinforcement learning. IEEE access, 7, 146264-146272.

Rashid, T., Samvelyan, M., De Witt, C. S., Farquhar, G., Foerster, J., & Whiteson, S. (2020). Monotonic value function factorisation for deep multi-agent reinforcement learning. The Journal of Machine Learning Research, 21(1), 7234-7284.

Sedighi, K. H., Ashenayi, K., Manikas, T. W., Wainwright, R. L., & Tai, H. M. (2004). Autonomous local path planning for a mobile robot using a genetic algorithm. In Proceedings of the 2004 congress on evolutionary computation (IEEE Cat. No. 04TH8753) (Vol. 2, pp. 1338-1345). IEEE.

Tampuu, A., Matiisen, T., Kodelja, D., Kuzovkin, I., Korjus, K., Aru, J., ... & Vicente, R. (2017). Multiagent cooperation and competition with deep reinforcement learning. PloS one, 12(4), e0172395.

Tan, M. (1993). Multi-agent reinforcement learning: Independent vs. cooperative agents. In Proceedings of the tenth international conference on machine learning (pp. 330-337).

Ulusoy, Gündüz, Funda Sivrikaya-Şerifoğlu, and Ümit Bilge. (1997) "A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles." Computers & Operations Research 24.4: 335-351.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., ... & Silver, D. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature, 575(7782), 350-354.

Wang, K. C., & Botea, A. (2011). MAPP: a scalable multi-agent path planning algorithm with tractability and completeness guarantees. Journal of Artificial Intelligence Research, 42, 55-90.

Xue, T., Zeng, P., & Yu, H. (2018). A reinforcement learning method for multi-AGV scheduling in manufacturing. In 2018 IEEE international conference on industrial technology (ICIT) (pp. 1557-1561). IEEE.

Yang, E., & Gu, D. (2004). Multiagent reinforcement learning for multi-robot systems: A survey (pp. 1-23). tech. rep.

Yang, Y., Juntao, L., & Lingling, P. (2020). Multi‐robot path planning based on a deep reinforcement learning DQN algorithm. CAAI Transactions on Intelligence Technology, 5(3), 177-183.

Yun, W. J., Jung, S., Kim, J., & Kim, J. H. (2021). Distributed deep reinforcement learning for autonomous aerial eVTOL mobility in drone taxi applications. ICT Express, 7(1), 1-4.

Wei, G., & Kuo, W. (2022). COLREGs-compliant multi-ship collision avoidance based on multi-agent reinforcement learning technique. Journal of Marine Science and Engineering, 10(10), 1431.

Appendix

Algorithm 1 Training Pseudocode

Initialize the parameters θ of the mixing network, agent networks, and the target networks of both.
Set the learning rate α
Set the episilon ϵ
Set the replay buffer D.
$step = 0$
while $terminated \neq 1$ **do**
  $t = 0$, $o_0 =$ Initial observation
  while not $terminated$ and $t <$ max time steps **do**
    for every agent a **do**
      $\tau_t^a = \tau_{t-1}^a \cup \{(o_t, u_{t-1})\}$ {τ_t^a: The trajectory of agent a up to time t}
      $u_t^a = \begin{cases} \text{argmax}_{u_t^a} Q(\tau_t^a, u_t^a) & \text{with probability } 1-\epsilon \\ randint(1, |U|) & \text{with probability } \epsilon \end{cases}$
    end for
    Get reward r_t, next obs o_{t+1} and next state s_{t+1}
    $D = D \cup \{(\mathbf{o_t}, \mathbf{s_t}, \mathbf{u}_t, \mathbf{r_t}, \mathbf{o_{t+1}}, \mathbf{s_{t+1}}, padded, terminated)\}$
    $t = t + 1$, step = step + 1
    if step = $\text{step}_{\max}$ or job_completed $\geq$ threshold **then**
      Set $terminated = 1$
      for each remaining step in the episode **do**
        Set $padded$ to 1
      end for
    end if
  end while
  if $|D| >$ batch-size **then**
    $b \leftarrow$ random batch of episodes from D
    for each time step t in each episode in batch b **do**
      $Q_{\text{tot}} = \text{Mixing-network}(Q_1(\tau_t, u_t^1), \ldots, Q_n(\tau_t, u_t^n), \text{Hypernetwork}(s_t; \theta))$
      Calculate target Q_{tot} using Mixing-network with $\text{Hypernetwork}(s_t; \theta^-)$
    end for
    $target = r + \gamma \cdot \text{target } Q_{\text{tot}} \cdot (1 - terminated)$
    $TD\_error = Q_{\text{tot}} - \text{target}$
    $mask = 1 - padded$
    $masked\_TD\_error = mask \cdot TD\_error$
    $loss = \frac{1}{\text{batch\_size}} \sum_{i=1}^{\text{batch\_size}} \frac{masked\_TD\_error[i]^2}{mask[i]}$
    $\theta = \theta - \alpha \cdot \Delta loss$
  end if
  if target update cycle **then**
    $\theta^- = \theta$
  end if
end while

Training Pseudocode는 Rashid et al.의 QMIX 알고리즘 pseudocode를 참고하여 작성하였고 학습 알고리즘의 각 기호에 대한 설명은 아래와 같다.

θ: 혼합 네트워크(Mixing network)와 에이전트 네트워크, 타겟 네트워크의 파라미터
α: 학습률(learning rate), Q 값 업데이트 시, 새로운 정보에 대해 얼마나 반영할지 결정하는 인자
ϵ: 탐험률(exploration rate), 에이전트가 무작위 행동을 선택할 확률
D: 재현 메모리(replay buffer), 학습 과정에서 수집한 경험을 저장, 무작위 배치를 선택하여 학습에 활용
τ_t^a: 에이전트 a의 시간 t까지의 행동 궤적(trajectory), 이전의 모든 행동을 포함한 기록
u_t^a: 에이전트 a가 시간 t에 선택한 행동
mask: Padding된 시간 단계의 영향을 제거하는 데 사용

요약문

첨단 기술이 적용된 대규모 유통 센터(mega DC)는 지능형 물류 운영을 실현하는 데 필수적인 역할을 한다. 그러나 다수의 자율 운송 장치가 동적으로 운용되는 대규모 mega DC의 효율적 관리는 기존 경로 계획 방식으로는 효과적으로 대응하기 어렵다. 이러한 문제를 해결하기 위해 본 연구는 다중 에이전트 강화 학습(MARL)과 보조 학습 기법을 결합한 혁신적인 경로 계획 프레임워크를 제안한다. 제안된 프레임워크는 학습의 가속화와 안정성을 위해 특화된 보상 구조와 추가적인 기법들을 도입하여 시스템 전반의 성능을 최적화하는 데 중점을 둔다. 소규모 환경에서 제한된 수의 운송 장치에 대한 경로 최적화를 목표로 MARL 네트워크를 훈련하여 효율성을 검증하였으며, 훈련된 모델 파라미터는 대규모 환경으로의 적용 시 초기 조건으로 활용하여 적응 과정을 가속화하였다. 또한, 복잡한 대규모 환경에서 수렴을 촉진하기 위해 학습을 난이도에 따라 단계적으로 진행하는 커리큘럼 학습 접근법을 적용하였다. 실험 결과, 제안된 접근법은 충돌, 혼잡, 교착 상태와 같은 문제를 효과적으로 해결함으로써 시스템 전체 성능이 유의미하게 향상되는 것을 확인하였다.

주제어: 메가 유통 센터(mega DC), 다중 에이전트 강화 학습, 동적 경로 계획, 운송 로봇, 교통 혼잡