Journal of the Korean Institute of Industrial Engineers
Vol. 49, No. 4, pp. 344-353, August 2023.
ISSN 1225-0988 | EISSN 2234-6457
https://doi.org/10.7232/JKIIE.2023.49.4.344


# 인간-AGV 충돌 위험을 고려한 강화학습 기반의 출하대기장 내 AGV 운영 최적화

**황인근 · 이현록†**

인하대학교 산업경영공학과

# Optimization of AGV Operation in the Dispatch Area Based on Reinforcement Learning Considering the Risk of Human-AGV Collision

Ingeun Hwang · Hyun-Rok Lee

Department of Industrial Engineering, INHA University

Automated guided vehicles (AGV) are a crucial component to achieve automation in logistics. AGVs can autonomously transport heavy pallets with the pre-designed rules, hence reducing the need for human resources in the logistics area. However, since AGVs cannot completely replace human workers doing maintenance tasks, inspections, etc., AGVs sometimes share operating areas with human workers. As such an example, we optimize the operation of an AGV in the dispatch area which has a human operator for final inspections. While most previous studies focus on efficient operation of AGVs, this study considers the possibility of human-AGV collisions as well as efficient operation. We also propose a PPO-R algorithm to prevent conservative behaviors of AGV when introducing a collision penalty. Numerical experiments show that PPO-R can maintain throughput while reducing the number of potential collisions.

***Keywords:*** Reinforcement learning, Automated Guided Vehicle, Dispatch area, Human-AGV Collision

## 1. 서 론

최근 대규모 물류 창고에는 많은 물동량을 신속하게 옮기는 작업에 무인운반차(Automated Guided Vehicle, 이하 'AGV')가 핵심적으로 이용되는 사례가 증가하고 있다(Oyekanlu *et al.*, 2020). 물류 산업 현장의 AGV는 물류 자동화를 위해 필수적인 로봇으로 미리 학습된 방식으로 자율적으로 주행하여 정해진 경로나 주어진 창고 공간 내에서 자유롭게 물류를 운송하는 임무를 수행할 수 있다. 때문에 AGV를 도입하면 물류 창고 운용 시 사람의 역할을 줄일 수 있으므로 인건비를 절감하고 높은 작업 효율을 얻을 수 있다는 장점이 존재하며 이러한 효과를 극대화하기 위해서는 AGV 운영을 최적화하는 것이 중요하다. AGV에게 어떤 작업을 할당하고, 어떤 경로를 따라 이송 업무를 수행하게 할지에 관한 연구들은 활발하게 이뤄지고 있으며, 수리계획법 기반의 전통적인 최적화 알고리즘(Chen, 1996)뿐만 아니라 시스템의 상태에 따라 동적인 의사결정을 내리는 에이전트를 학습시키는 강화학습 알고리즘을 활용하여(Liu *et al.*, 2022) AGV 운영을 최적화한다.

하지만 AGV 운영과 관련된 현재 연구들은 AGV의 운반 업무에만 초점이 맞춰져 있어 실제 현장에서 발생할 수 있는 "사람"과 관련된 변수를 충분히 고려하지 못한다는 한계점을 지니고 있다. AGV를 이용하는 자동화 창고에서는 사람의 개입을 최소화하는 것이 궁극적인 목표이나 장비의 수리나 제품의 검품 등 사람의 개입이 비교적 필수적인 영역이 여전히 존재한다(Agnisarman *et al.*, 2019). 따라서 AGV의 작업 공간 내에 사람이 존재하는 경우가 충분히 발생할 수 있으므로 AGV 운

이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. RS-2022-00165642). 이 논문은 2023년도 인하대학교의 지원에 의하여 연구되었음.
† 연락저자 : 이현록 교수, 22212, 인천광역시 미추홀구 인하로 100 인하대학교 2북관 481호, Tel : 032-860-7373, E-mail : hyunrok.lee@inha.ac.kr
2023년 4월 28일 접수; 2023년 6월 13일 수정본 접수; 2023년 7월 4일 게재 확정.

영 최적화 시 운반 업무의 효율성만을 최대화하는 것이 아닌 AGV와 사람 간의 충돌 가능성 또한 고려해야 한다.

본 논문에서 다루는 출하대기장은 AGV가 운용될 수 있는 물류 창고 내 다양한 환경 중 특히 사람과 AGV의 작업 공간이 중첩되는 곳이다. 출하대기장은 고객에게 제품이 전달되기 직전에 파렛트(pallet) 단위의 재고가 출하되는 장소로 출하 직전에는 제품의 수량과 품질에 대한 검품이 이뤄진다. 이러한 검품 작업은 사람에 의해 행해지기 때문에 출하대기장에 AGV가 도입되면 제품의 적재공간 사이를 오가며 검품을 수행하는 사람과 파렛트를 입고공간, 적재공간, 출하공간 사이에 운송하는 AGV가 공통의 공간에서 작업을 수행하게 된다. AGV만이 주로 운영되어 물류 운반이 이뤄지는 다른 공간과 달리 출하대기장은 검품하는 사람의 위치에 따라 AGV의 작업 순서와 경로를 달리해야 하므로 본 연구에서는 강화학습을 통해 검품원의 현재 위치를 고려한 AGV 운영 방식을 찾고자 한다.

최적 AGV 운영 방식을 찾기 위해 본 논문에서 사용되는 강화학습은 시스템으로부터 받는 보상 값을 토대로 시스템 상태마다 최적 의사결정이 무엇인지를 탐색하는 방식이다. 특히 AGV가 운영되는 물류 창고에 존재하는 다양한 상황들과 제약을 수리적으로 표현하기 복잡한 경우에는 강화학습을 적용하는 것이 수리계획법 기반 방법들보다 더욱 적합하며 딥러닝 기반의 심층강화학습 알고리즘이 발달함에 따라 복잡한 환경에서도 높은 성능을 갖는 AGV 운영 방침을 비교적 쉽게 도출할 수 있는 장점이 있다. 본 연구에서는 AGV와 사람 작업자가 동시에 존재하는 출하대기장 환경을 강화학습 모델로 표현하고, 심층강화학습 알고리즘 중의 하나인 PPO(Proximal policy Optimization) 알고리즘을 사용하여 출하대기장 내 최적 AGV 운영 방식을 찾는다. 특히, 출하 속도를 최대화하기 위해 입고된 파렛트를 최종 출하 전에 거쳐야 하는 지점들로 AGV가 효율적으로 이송하기 위한 보상 함수를 설계하고 그 구조를 해석한다. 또한 PPO 알고리즘을 변형한 PPO-R 알고리즘을 제안하여 사람 작업자와의 작업 공간 중첩을 최소화하기 위한 보상을 도입했을 때의 학습 성능 저하를 방지한다.

본 논문의 남은 내용은 다음과 같이 구성된다. 제2장에서는 출하대기장 내 AGV 최적 운영과 관련된 선행 연구를 살펴본다. 제3장에서는 본 연구 내용을 이해하기에 필요한 배경 이론을 설명하고, 출하대기장 내 AGV 최적 운영 모델과 학습 방법을 제4장에서 제시한다. 제안하는 알고리즘을 적용하고 평가한 실험 내용을 제5장에서 분석한 후 제6장의 결론으로 내용을 마무리한다.

## 2. 관련 연구

자동화된 시스템을 구축하기 위해 강화학습은 다양한 산업에서 자주 사용되고 있다. 예를 들어, Lee *et al.*(2022)은 반도체 제조 공정 내 스케줄링 문제를 DQN 알고리즘을 이용하여 해결하고, Tsai *et al.*(2020)은 DDPG 알고리즘을 이용해 신발 제조 공정에 사용되는 로봇 팔을 학습시킴으로써, 생산성과 효율성을 향상시킨다. 물류 산업도 강화학습을 적용하여 해결할 수 있는 의사결정 문제가 다수 존재하는 분야로 물품 운송, 스케줄링, 경로 계획 등의 문제를 강화학습으로 해결하는 연구가 다수 존재한다. He *et al.*(2023)은 온라인 상거래의 발달로 인해 증가한 수요에 대응하기 위해 사용되는 높은 저장 효율을 가지는 퍼즐 기반 저장 시스템(puzzle-based storage system, 이하 'PBS')에서의 다중 물품 획득 문제를 다룬다. PBS 시스템의 경우는 공간 활용도가 높지만, 보관 중인 물품을 꺼내기 위한 검색 프로세스가 상당히 복잡한데, 이를 dueling DQN, double DQN을 이용해 해결하였다. Ahn *et al.*(2021)은 공장에서 이용되는 오버헤드 호이스트(Overhead Hoist Transport, 이하 'OHT')가 정체로 인해 일시적으로 이동할 수 없는 공백이 발생하는 문제를 해결한다. 이 논문은 다수의 OHT 간의 협력적인 이동을 제어하기 위해 멀티에이전트 강화학습 알고리즘인 QMIX와 그래프 기반 학습을 통해 OHT 이동 경로를 최적화함으로써 물류 시스템에서 발생하는 비효율적인 문제를 해결한다.

물류 시스템 내에서 AGV의 사용 빈도가 점점 늘어나고 있는 만큼 AGV의 최적 운용을 강화학습을 통해 학습시키는 것에 대한 관심도 높아지고 있다. Lee *et al.*(2021)은 Q-Learning 알고리즘과 Dyna-Q 알고리즘을 이용하여 여러 개의 작업대가 존재하는 격자모양 환경에서 최적 경로 찾기 문제를 해결한다. 이를 통해 AGV가 격자모양 환경에서 경로 탐색을 할 때 Q-Learning 및 Dyna-Q 알고리즘의 활용 가능성을 제시하였다. Li *et al.*(2019)은 DQN 알고리즘을 이용해 로봇이 해결해야 할 작업을 예측하고 최적의 경로를 계획하도록 하였다. 이 논문에서는 다양한 작업 환경에서 효과적으로 작동하며, 자율적인 결정과 경로 계획이 가능한 AGV 운영 방침을 찾았다. 실제 AGV의 운영 환경에서는 AGV가 직면할 수 있는 충돌 상황이 다수 존재하지만, 앞의 두 논문들은 AGV 운영 시 충돌 가능성에 대한 고려가 없다는 한계점을 갖고 있다. 이러한 한계점을 해결한 Hu *et al.*(2023)은 Q-Learning을 이용하여 항구에서 컨테이너를 운반하는 여러 대의 AGV간의 충돌을 방지하면서 효율적인 경로를 선택하도록 한다. 또한 Choi *et al.*(2022)에서는 주문처리 센터를 모델링하고 QMIX 알고리즘을 활용하여 AGV 간의 성능 저하 방지를 위해 엄격히 충돌을 고려하였다. 하지만 AGV의 충돌 가능성을 고려한 연구들에서도 AGV간의 충돌을 중요하게 고려하나 사람과의 충돌은 고려되지 않았다. 따라서 본 연구는 사람 작업자가 필연적으로 존재하는 환경에서 AGV와 작업자 간의 충돌에 주목하였다.

## 3. 배경 이론

### 3.1 MDP(Markov Decision Process) 모델

강화학습의 기반 모델은 순차적 의사결정 모델인 MDP

(Markov Decision Process, 이하 MDP) 모델이다 (Puterman, 2014). MDP 모델은 크게 5가지 요소로 구성된다. 시스템(환경)의 상태(state) $s \in S$, 에이전트의 결정을 나타내는 행동(action) $a \in A(s)$, 현재 상태와 에이전트의 행동에 따른 다음 상태로의 전이 확률(transition probability) $p(s' \mid s,a) \in [0,1]$, 상태 전이에 따른 보상(reward) $r(s,a,s') \in \mathbb{R}$ 그리고 미래의 보상을 현재 가치로 계산하는 할인율(discount factor) $\gamma \in [0,1]$이다. $S$는 상태 공간, $A(s)$는 상태 $s$에서 선택 가능한 행동 공간을 의미한다. MDP 모델의 해는 에이전트의 행동 정책(policy)을 정의하는 것으로 각 상태에서 액션 선택확률을 정의하는 함수 $\pi(a \mid s) \in [0,1]$이다. 이 중 최적해는 매 상태에서 현재 상태와 그 이후로 얻게 되는 보상의 누적 합을 최대화하는 액션을 선택하는 것으로 Bellman의 최적 방정식(Bellman's optimality equations)을 풀어서 찾을 수 있다.

### 3.2 PPO(Proximal Policy Optimization) Algorithm

상태/액션 공간이 방대해지거나 정확한 수리 모델의 표현이 어려운 상황에서는 강화학습을 통해 근사적으로 MDP 모델의 최적해를 찾는다. 강화학습에서는 가치 함수 $V(s)$와 정책 함수 $\pi(a \mid s)$를 샘플 기반으로 추정하는 것이 핵심이고, 최근에는 복잡도가 높은 문제들을 해결하기 위해 심층신경망(deep neural network;DNN)을 사용하는 심층 강화학습 알고리즘들이 주로 사용된다. 이 중 PPO 알고리즘은 가치 함수 $V(s)$와 정책 함수 $\pi(a \mid s)$를 모두 심층신경망으로 설계하여 최적해를 찾는 방법으로 심층신경망에 급격한 변화가 생기면서 학습의 안정성이 저하되는 것을 그래디언트 클리핑(gradient clipping)으로 간단하게 방지하는 알고리즘이다(Schulman *et al.*, 2017). PPO 알고리즘은 비교적 구현이 간단하고, 실제 문제에서도 성능이 우수한 편이라 로봇 제어 등의 분야에서 효과적으로 적용되고 있다(Melo *et al.*, 2019; Zhang *et al.*, 2021). PPO 알고리즘은 다음과 같이 작동한다(<Table 1> 참조).

**Table 1.** PPO Algorithm

- Initialization: Policy network parameter $\theta$; Value network parameter $\phi$
- Repeat for a fixed number of iterations
  - Collect a fixed length of state-action-reward sequence by $\pi_\theta(a|s)$; $s_t, a_t, r_t, s_{t+1}, a_{t+1}, r_{t+1}, \cdots$
    - For each state, action $s, a$
      - Compute $A(s,a)$ using $V_\phi(s)$; $A(s,a) \approx r + \gamma V_\phi(s') - V_\phi(s)$
      - Compute $L(s,a,\theta_k,\theta)$ for policy optimization
  - Update $\theta$ using the summation of $L(s,a,\theta_k,\theta)$ of state, actions in the sequence
  - Update $\phi$ to minimize $MSE$ using state, actions in the sequence; $MSE \approx r + \gamma V_\phi(s') - V_\phi(s)$

PPO는 가치함수를 평균제곱오차(Mean Squared Error; MSE)를 최소화하는 것을 목적으로 추정하며 가치함수 추정 시 시퀀스 단위의 샘플을 활용할 수 있는 GAE(Generalized Advantage Estimation)를 사용한다. 본 논문에서는 AGV가 작업자와의 충돌을 고려하여 지나치게 보수적으로 행동하는 것을 방지하기 위해 PPO 알고리즘을 변형한 PPO-R 알고리즘을 제안한다.

## 4. 수리 모델 및 알고리즘

### 4.1 문제 정의: 출하대기장의 구조 및 출하업무 흐름

출하는 화물이 창고 바깥으로 나가는 것을 의미하며 출하대기장은 상차 이전에 제품을 보관하며 출하검품이 수행되는 곳이다. 본 연구에서는 출하대기장을 25 × 25m의 정사각형 공간으로 가정하고, 총 25개의 칸을 가진 격자로 구역을 구분하였다. 즉, 1칸은 5 × 5m의 공간이고 출하대기장은 각각 1칸을 차지하는 입고공간, 출하도크장, 그리고 4곳의 파렛트 임시 보관장소를 포함한다. 또한 해당 출하대기장 내에는 파렛트 운반을 위한 AGV 1대, 출하 검품 작업을 위한 작업자 1명이 존재하는 것으로 가정한다. AGV의 이동은 격자 단위로 이뤄지는데 실제 사용되는 AGV의 평균적인 이동 속도가 2.0m/s임을 고려하여 1칸 이동 시 2.5초의 시간이 소요되는 것으로 계산하였다. 또한, AGV가 운송하는 파렛트 단위의 재고 크기는 1.1 × 1.1m이기 때문에 1칸에는 여유 공간을 포함하여 최대 10개의 파렛트를 보관할 수 있는 상황으로 가정하였다.

AGV가 수행해야 할 작업은 출하대기장 내 기본적인 업무 흐름을 따른다. 먼저 출고작업을 마친 파렛트 단위의 재고가 입고공간을 통해 출하대기장으로 들어온다. 입고공간에 놓여 있는 파렛트는 AGV를 통해 검품이 이뤄지는 파렛트 임시 보

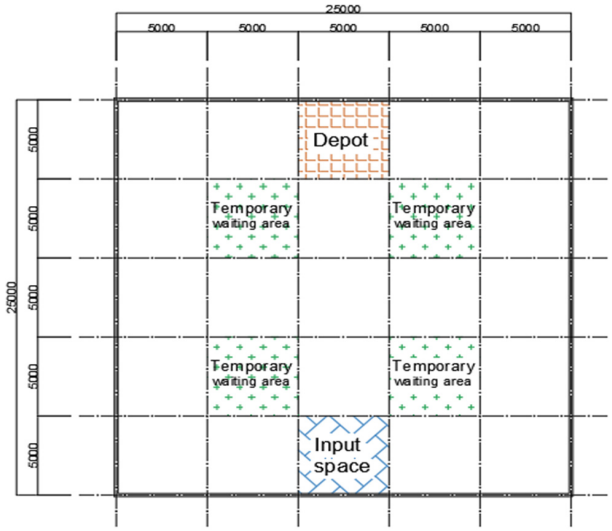



**Figure 1.** Dispatch Area Layout

관 장소로 운반되고, 작업자는 파렛트 임시 보관 장소를 순환하며 검품이 안 된 파렛트를 찾아 검품 작업을 수행한다. 검품이 완료되면 파렛트는 최종 출하가 가능하고, 출하지시가 발생하면 AGV가 검품이 끝난 파렛트를 임시 보관장소에서 집어 출하도크장까지 운반한다. 최종적으로 파렛트가 출하도크장으로 운반되면 해당 파렛트에 대한 출하처리 업무가 마무리된다. 이 과정에서 단위 시간당 파렛트 입고 수와 출하 지시 수는 포아송 분포를 따르고, 출하지시는 최대 20건까지 쌓여 있을 수 있음을 가정하였다. 또한 검품 작업은 파렛트에 붙여진 바코드 라벨을 작업자가 직접 스캔하는 방식을 감안하여 2.5초가 소요되고, 일반인의 평균 보행속도인 1.29m/s를 고려하여 작업자는 1칸을 이동하는데 AGV의 두 배인 5초가 소요되는 것으로 가정하였다. 본 연구는 이러한 출하업무 흐름 내에서 강화학습으로 학습시킨 AGV가 상황에 맞게 이동하며 파렛트를 효율적으로 운송하여 1시간 동안 출하량을 최대화하는 것을 목적으로 한다.

또한, 본 연구는 출하대기장에서 AGV로 인해 발생할 수 있는 충돌을 고려하였으며, 특히 필수적으로 존재하는 사람 작업자와 AGV 사이의 충돌 위험을 고려한다. 작업자와 AGV 사이의 물리적인 충돌은 충돌방지 센서 등을 통해 하드웨어적으로 방지하는 편이다(Sun *et al.*, 2019). 하지만 실제 충돌이 발생하지 않더라도 같은 공간 내에 작업자와 AGV가 동시에 위치하는 상황이 빈번하게 발생할 경우, AGV와 작업자의 서로를 의식한 움직임은 전체적인 출하처리 업무에 능률 저하를 발생시킬 수 있다. 따라서 5 × 5m 크기의 격자 한 칸에 AGV와 작업자가 동시에 존재하는 것은 가능하나 해당 상황을 잠재적 충돌 상황이라 산정하였다. 또한 출하대기장 공간을 벗어나는 이동을 택하거나, 파렛트를 집은 상태로 입고 공간에 들어가려 하는 경우와 같이 유효하지 않은 움직임으로 인해 작업자와 상관없이 AGV 혼자 충돌이 발생할 수 있는 상황들은 학습 환경에서 발생하지 않는 대신 AGV에게 음의 보상을 전달하도록 문제를 설정하였다. 정리하자면, 정의된 충돌 상황을 최대한 피하면서 시간당 출하량을 최대화할 수 있는 AGV 운영 방식을 찾는 것이 본 연구에서 다루는 강화학습 문제의 최종 목적이다.

### 4.2 MDP 모델

이 장에서는 출하대기장 내 AGV 운영에 관한 MDP 모델의 6가지 구성 요소를 설명한다.

• 상태(state) $s$

시스템의 상태는 AGV의 현재 위치를 나타내는 행 번호 $r$와 열 번호 $c$, AGV의 파렛트 적재 여부 $l$, AGV에 적재된 파렛트의 목적지 $d$, 입고공간에 적재된 파렛트 수 $n^I$, 출하지시 수 $n^D$, 임시 보관장소 $i$에 놓여 있는 검품 전, 후 파렛트 수 $n^W_{0,i}$, $n^W_{1,i}$, 작업자의 현재 위치를 나타내는 행 번호 $r^h$와 열 번호 $c^h$, 작업자의 이동 단계 $m^h$로 정의된다; $s=(r, c, l, d, n^I, n^D, n^W_{0,1}, ..., n^W_{0,N_W}, n^W_{1,1}, ..., n^W_{1,N_W}, r^h, c^h, m^h)$. 격자 공간의 행 개수 $H$, 열 개수 $W$, 격자 한 칸에 보관 가능한 최대 파렛트 수 $n^{\max}$, 최대 출하지시 건수 $o^{\max}$, 임시 보관장소 수 $N^W$가 주어졌을 때, 각 상태 요소가 가질 수 있는 값의 범위와 상태 요소의 정의는 <Table 2>에 정리되어 있다.

예를 들어, $state \in \{3, 2, 0, 1, 2, 3, 0, 0, 1, 0, 1, 0, 0, 0, 2, 1, 1\}$에서 첫 두 개 요소는 AGV가 세 번째 행, 두 번째 열에 있음을 의미한다. 세 번째 요소인 0은 AGV가 현재 파렛트를 싣지 않았음을 의미하며, 네 번째 요소인 1은 AGV의 다음 목적지가 파렛트를 실을 수 있는 위치로 이동해야 함을 의미한다. 다섯 번째, 여섯 번째 상태 요소는 각각 입고공간에 2개의 파렛트가 존재하고, 출하지시 건 수가 3건임을 뜻한다. 이후 연달아 나오는 (0, 0, 1, 0)은 검품이 되지 않은 파렛트가 세 번째 파렛트 임시 보관 장소에만 1개 존재함을 의미하고, 다음의 (1, 0, 0, 0)은 검품이 끝난 파렛트가 첫 번째 파렛트 임시 보관 장소에만 1개 존재함을 의미한다. 열다섯, 열여서 번째 상태는 작업자의

**Table 2.** State Definition

| state element | description |
|---|---|
| $r \in \{1,2,...,H\}$ | current row number of AGV |
| $c \in \{1,2,..., W\}$ | current column number of AGV |
| $l \in \{0,1\}$ | 1 if AGV loaded a pallet, otherwise 0 |
| $d \in \{1,2,3\}$ | destination of a loaded pallet on AGV; 1 indicates pallet loadable area, 2 indicates temporary waiting area, 3 indicates depot |
| $n^I \in \{0,1,...,n^{\max}\}$ | the number of pallets waiting in input space |
| $n^D \in \{0,1,...,o^{\max}\}$ | the number of pallets that should be dispatched |
| $n^W_{0,i} \in \{0,1,...,n^{\max}\}$ | the number of uninspected pallets in temporary waiting area $i \in \{1,2,...,N_W\}$ |
| $n^W_{1,i} \in \{0,1,...,n^{\max}\}$ | the number of inspected pallets in temporary waiting area $i \in \{1,2,...,N_W\}$ |
| $r^h \in \{1,2,...,H\}$ | current row number of human operator |
| $c^h \in \{1,2,..., W\}$ | current column number of human operator |
| $m^h \in \{0,1\}$ | human operator's movement step |

위치가 두 번째 행, 첫 번째 열이라는 것을 나타내며, 마지막 상태 요소 값 1은 작업자가 격자 하나의 절반 정도를 이동했음을 의미한다.

• 행동(action) $a$

의사결정시점마다 AGV 에이전트가 선택할 수 있는 행동은 AGV가 움직일 방향을 나타내며 상, 하, 좌, 우, 대기로 총 5가지 종류이다; $a \in \{상,하,좌,우,대기\}$. 대기 액션의 경우에는 불필요한 움직임을 최소화하며, AGV가 작업자와의 충돌 상황을 효과적으로 피할 때 사용될 수 있다.

• 전이 확률(transition probability) $p(s' \mid s,a)$

AGV가 의사결정시점마다 움직일 방향 $a$를 결정하면 다음 상태 $s'$는 확정적으로 결정된다. AGV의 위치는 공간 밖을 나가거나 파렛트가 적재된 상태로 입고 공간에 진입하려는 경우 현재 위치에 그대로 머물고, 나머지의 경우에는 선택된 방향에 맞춰 행 번호 $r$와 열 번호 $c$ 값이 달라진다. AGV가 빈 상태($l=0$)로 입고공간이나 임시 보관장소에 진입하면 파렛트를 적재($l=1$)할 수 있다. 적재하기 위해서는 입고공간에서는 대기 중인 파렛트가 존재해야 하며($n^I>0$), 임시 보관장소에는 검품이 끝난 파렛트가 존재해야 한다($n^W_{1,i}>0$). AGV가 파렛트를 운송 중일 때($l=1$)는 목적지 번호 $d$에 부합하는 지점에 진입하면 파렛트를 내려놓고 적재 상태 $l$이 0으로 변경된다. 입고공간에 적재된 파렛트 수 $n^I$, 출하지시 수 $n^D$는 도착률 $\lambda$을 갖는 포아송 분포에 따라 증가한다.

작업자는 임시 보관장소를 오가면서 파렛트를 검품한다. 작업자가 임시 보관장소에 위치하면 검품 전의 파렛트 하나가 검품 상태로 변경된다; $n^W_{0,i} \leftarrow n^W_{0,i}-1$, $n^W_{1,i} \leftarrow n^W_{1,i}+1$. 작업자는 임시 보관장소를 시계 방향으로 순회하면서 검품해야 할 파렛트를 찾다가, 임시 보관장소에 검품이 필요한 파렛트가 발생한 경우, 순회 방향을 결정하여 가까운 방향으로 이동한다. 또한 작업자의 이동속도는 AGV보다 느리기 때문에 작업자의 위치는 두 단계에 걸쳐 변경된다. 즉, 작업자의 위치를 나타내는 행 번호 $r^h$, 열 번호 $c^h$는 작업자의 이동 단계 $m^h=1$일 때만 새로운 값으로 변경된다.

**Table 3.** Reward

| reward type | description |
|---|---|
| $r_W(s,a,s') \in \mathbb{R}^+$ | reward when pick or drop a pallet at temporary waiting area |
| $r_I(s,a,s') \in \mathbb{R}^+$ | reward when pick a pallet from input space |
| $r_D(s,a,s') \in \mathbb{R}^+$ | reward when drop a pallet at depot |
| $r_C(s,a,s') \in \mathbb{R}^+$ | penalty when AGV and human operator are on the same cell |
| $r_L(s,a,s') \in \mathbb{R}^+$ | penalty when AGV is not moving correctly near destination |
| $r_B(s,a,s') \in \mathbb{R}^+$ | penalty when AGV is blocked by unavailable move |
| $r_H(s,a,s') \in \mathbb{R}^+$ | holding cost of pallets at input space |
| $r_R(s,a,s') \in \mathbb{R}^+$ | cost caused by waiting dispatch orders |

• 보상(reward) $r(s,a,s')$

보상함수는 AGV가 수행한 결과물에 따라 여섯 가지 세부 요소로 계산된다.

$$\begin{aligned} r(s,a,s') = {} & r_W(s,a,s') + r_I(s,a,s') + r_D(s,a,s') \\ & - r_C(s,a,s') - r_L(s,a,s') - r_B(s,a,s') \\ & - r_G(s,a,s') - r_C(s,a,s') \end{aligned}$$

AGV가 양의 보상을 얻을 수 있는 경우는 세 경우이다. $r_W(s,a,s')$은 AGV가 파렛트를 파렛트 임시 보관장소에서 내려놓거나 적재한 경우 얻는 보상, $r_I(s,a,s')$는 입고공간의 파렛트를 AGV가 집었을 때 얻는 보상, $r_D(s,a,s')$는 출하지시가 존재하는 상황에서 검품된 파렛트를 출하도크장에 운반한 경우 발생하는 보상이다. 반대로 AGV와 작업자가 같은 위치에 있을 경우 잠재적 충돌에 의해 $r_C(s,a,s')$만큼 벌금이 발생하며, AGV가 목적지에 진입할 수 있는 상황에서 진입하지 않는 경우는 $r_L(s,a,s')$, AGV가 불가능한 움직임을 시도할 경우는 $r_B(s,a,s')$만큼의 벌금이 부과된다. 또한 입고공간과 쌓인 파렛트 수와 대기 중인 출하지수 수에 비례하여 각각 $r_H(s,a,s')$와 $r_R(s,a,s')$의 비용이 발생한다.

• 할인율(discount factor) $\gamma$

미래의 보상을 현재 가치로 환산할 때 사용하는 $\gamma$는 0.99로 설정하였다.

• 의사결정 기간(time horizon) $T$

AGV가 1시간 동안 2.5초 간격으로 1440번 이동하여 출하량을 최대화하도록 설정하였다.

설계한 MDP 모델은 출하대기장에서의 AGV와 작업자의 상세한 상태를 포함하여 Markov property를 갖는 상태 전이를 표현하기 때문에 수리적으로 최적해를 구하기는 어려운 상태/액션 공간을 갖는다. 따라서 함수 근사를 통해 관심 있는 상태, 액션에 대해 주요하게 탐색하는 강화학습 방법을 통해 해를 도출하고자 한다.

### 4.3 PPO-R Algorithm

본 연구에서는 AGV와 작업자 간의 충돌을 방지하기 위해 충돌 시 $r_C(s,a,s')$만큼의 벌금을 부과한다. 하지만, $r_C(s,a,s')$가 과도하게 책정될 경우, PPO 알고리즘으로 학습하는 AGV가 작업자와의 충돌을 방지하기 위해 지나치게 보수적으로 반응하여 파렛트 운송을 포기하는 경우가 종종 발생한다. 따라서 본 연구

**Table 4.** PPO-R Algorithm

- Initialization: Policy network parameter $\theta$; Value network parameter $\phi$
- Repeat for a fixed number of iterations
  - if the average number of dispatched orders < $\delta$ and for each k iterations:
    Initialize $\theta$, $\phi$
  - Collect a fixed length of state-action-reward sequence by $\pi_\theta(a|s)$; $s_t, a_t, r_t, s_{t+1}, a_{t+1}, r_{t+1}, \cdots$
    - For each state, action $s, a$
      - Compute $A(s,a)$ using $V_\phi(s)$;
        $A(s,a) \approx r + \gamma V_\phi(s') - V_\phi(s)$
      - Compute $L(s,a,\theta_k,\theta)$ for policy optimization
  - Update $\theta$ using the summation of $L(s,a,\theta_k,\theta)$ of state, actions in the sequence
  - Update $\phi$ to minimize $MSE$ using state, actions in the sequence;
    $MSE \approx r + \gamma V_\phi(s') - V_\phi(s)$

에서는 AGV가 학습 초기에 작업자와의 충돌을 방지하기 위해 구석으로 이동하여 머물도록 학습이 된다면, 처음부터 다시 학습할 기회를 제공하는 PPO-R (PPO-Reset) 알고리즘을 제안한다. PPO-R은 PPO 알고리즘과 같은 원리로 학습은 진행되지만, 일정 주기마다 학습 상태를 확인하여, 성공적인 파렛트 운반 횟수가 일정 기준을 넘지 못한다면 처음부터 다시 학습을 시작하는 방식이다. 이를 통해 비교적 큰 값의 충돌 벌금 $r_C(s,a,s')$이 책정되더라도 작업자와의 충돌을 방지하면서 파렛트를 운송할 수 있는 방법을 탐색해나갈 수 있다. PPO-R의 학습 방식은 <Table 4>와 같다.

본래 PPO는 정책 신경망과 가치 신경망을 각각 파라미터 $\theta$와 $\phi$ 로 초기화한 후, 환경과 상호 작용하며 $\theta$와 $\phi$ 를 지속적으로 업데이트한다. PPO-R은 환경과의 상호작용 횟수(step 수)가 $k$번이 될 때마다 현재 학습된 정책이 얼마만큼 성공적으로 파렛트를 출하하였는지를 평가한다. 만약, 평균 출하 처리 수가 0에 가까운 $\delta$ 값을 넘지 못한다면, AGV가 학습되지 못하고 있는 것으로 간주하고 정책 신경망과 가치 신경망의 파라미터 $\theta$와 $\phi$ 를 다시 초기화한다. AGV가 적절하게 학습하지 못하고 있을 때 다시 한 번 학습의 기회를 줌으로써 기존 PPO 알고리즘에 비해 작업자와의 충돌을 방지하면서 성공적으로 파렛트를 출하하는 정책을 얻을 수 있다. 가중치를 초기화하더라도 전체 step 수가 추가되는 것은 아니기 때문에 추가적인 학습 시간 및 자원이 필요하지는 않다.

## 5. 실험 결과

### 5.1 실험 환경

출하대기장에서 AGV와 사람 작업자가 업무를 수행하는 환경은 강화학습 알고리즘을 적용하기에 적합한 openAI의 gym 라이브러리의 형식에 맞춰 작성하였다 (Brockman, 2016). 이는 다양한 강화학습 알고리즘을 구현해둔 라이브러리 사용이 가능케 하여 stable-baselines2 (Hill *et al.*, 2018) 에서 구현한 PPO 알고리즘과 이를 수정한 PPO-R 알고리즘을 실험에 사용하였다. PPO와 PPO-R 알고리즘의 hyper parameter 중 학습율(learning rate)은 0.0001, 0.00025, 0.001, 0.0025, 0.01, 0.025 중 가장 최적의 결과값을 낸 0.001을 사용했고, 이 외의 hyper parameter 값들은 stable-baselines2의 기본값을 사용하였다. 학습은 에이전트가 환경과 상호작용하는 횟수(step 수)가 총 500,000번이 되는 동안 진행하였으며, 매 10,000 step마다 50개의 에피소드를 사용하여 모델의 성능을 평가하였다. PPO-R 알고리즘은 50,000 step마다 평균 출하 건수가 0.0001을 넘기지 못하면 네트워크를 초기화하였다; $k = 50{,}000, \delta = 0.0001$. 최종 성능을 비교할 때는 알고리즘마다 학습 동안 평가된 정책 중 가장 높은 출하량을 가진 정책을 사용하였다. 실험은 입고 파렛트 수와 출하지시 수의 도착률 $\lambda$가 작을 때($\lambda = 0.04$)와 클 때($\lambda = 0.08$) 두 가지 경우를 가정하여 수행하였고, 각각 초당 0.96, 1.92개가 도착함을 의미한다. 실험적으로 확인했을 때 $\lambda = 0.04$은 AGV 1대가 여유롭게 처리 가능한 수치이며, $\lambda = 0.08$는 출하대기장 모형에서 처리 가능한 한계치에 가까운 값이다.

### 5.2 보상값 설정

우선 기본 PPO 알고리즘에서 AGV 에이전트의 출하 속도를 최대화하기 위해 적절한 보상값을 결정하기 위한 실험을 진행하였다. 작업자와의 충돌과 관련된 $r_C(s,a,s')$을 제외한 값들을 결정하였으며, 특히 AGV가 적절하게 파렛트를 이동시킬 때 발생하는 세 개의 보상 $r_W(s,a,s')$, $r_I(s,a,s')$, $r_D(s,a,s')$의 우선순위를 변경해가며 최적의 값을 찾았다. 9개의 서로 다른 우선순위에 대하여 각각 5번씩 실험을 진행한 후 출하 성공 수의 평균을 비교하여 보상값을 설정하였다. 결과적으로 파렛트 임시 보관 장소에 파렛트를 두거나 적재할 때의 보상, 출하 도크장에 파렛트를 내려둘 때의 보상, 입고공간에서 파렛트를 실었을 때의 보상 순으로 보상값의 순서가 정해질 때 가장 좋은 성능을 보였으며, 이에 따라 $r_W(s,a,s')$, $r_I(s,a,s')$, $r_D(s,a,s')$는 각각 13, 7, 10의 값으로 설정하였다. 나머지 변수들은 AGV가 적절하게 이동하지 못해 발생하는 벌금 $r_L(s,a,s')$과 $r_B(s,a,s')$는 3, 3, $r_H(s,a,s')$는 입고 공간에 있는 파렛트 수의 제곱하여 0.01을 곱한 값, 그리고 $r_R(s,a,s')$는 출고 대기 중인 주문 수의 제곱하여 0.005을 곱한 값으로 설정하였다.

추가로 작업자와의 충돌 횟수를 줄이며 출하 성공 수가 높은 정책을 찾기 위하여 충돌에 대한 페널티 값 $r_C(s,a,s')$을 변경하며 실험을 진행하였다. PPO 알고리즘은 $r_C(s,a,s')$ 값이 증가함에 따라 학습이 잘 안되는 경우가 빈번하게 발생하

**Table 5.** Collision Penalty

| Collision Penalty Value | Number of dispatched orders | Number of collisions | Number of resets |
|---|---|---|---|
| 3 | 49.62 (±2.34) | 85.788 (±36.24) | 3 |
| 5 | 49.844 (±3.75) | 51.504 (±30.06) | 2 |
| 7 | 51.384 (±1.11) | 53.188 (±27.58) | 7 |
| 10 | 51.668 (±2.99) | 43.56 (±25.41) | 7 |
| 12 | 29.868 (±24.60) | 37.293 (±8.25) | 29 |

기 때문에 PPO-R 알고리즘을 사용하였다. $r_C(s, a, s')$값을 3, 5, 7, 10, 12, 15로 점점 늘려가며, 각각 5번씩 실험을 진행한 결과는 <Table 5>와 같다. PPO-R 알고리즘에서 $r_C(s, a, s')$ 값에 따라 얼마나 자주 심층신경망을 초기화하는지에 대한 횟수(number of resets) 또한 기록하였다. 괄호에 있는 값은 5번 실험 결과의 표준 편차 값이다.

PPO-R 알고리즘 사용 시, $r_C(s,a,s')$ 값을 증가시키면 비슷한 출하 성공 횟수를 유지하면서 충돌 횟수를 점차 줄일 수 있지만 과도하게 증가된 값에서는 심층신경망을 초기화하는 횟수가 증가하여 PPO-R 알고리즘의 사용에도 제대로 된 학습이 진행되지 못하였다. 결과적으로, 충돌을 최소화하며 가장 좋은 출하 처리 성능을 보인 것은 $r_C(s,a,s')$ 값이 10인 경우였고, 이 값을 $r_C(s,a,s')$ 값으로 선정하였다.

### 5.3 알고리즘 비교 : 휴리스틱, PPO, PPO-R

선정된 보상 값들을 사용하여 PPO 알고리즘과 PPO-R 알고리즘, 룰 기반의 휴리스틱(heuristic)의 성능을 비교하였다. PPO 알고리즘은 $r_C(s, a, s')$ 값을 10으로 적용했을 시 적절한 학습이 이뤄지지 않기 때문에 충돌을 고려하지 않고 $(r_C(s, a, s') = 0)$ 학습한 결과를 사용하였다. 사용한 휴리스틱은 AGV의 목표 지점을 룰 기반으로 지정하여 AGV가 최소한의 움직임으로만 출하처리 업무를 수행할 수 있게 불필요한 움직임을 제거한 알고리즘이다. AGV가 파렛트를 적재한 상태이면 검품 여부에 따라 파렛트 임시 보관 장소나 출하도크장으로 이동하여 파렛트를 내려둔다. 반면에 AGV가 파렛트를 적재하지 않은 상태이면, 입고공간으로 우선적으로 이동하여 파렛트를 운송한 뒤 임시 보관장소의 검품이 끝난 파렛트를 이송한다. 만약 AGV가 수행할 어떠한 업무도 존재하지 않는다면 그 자리에 머물도록(STAY) 설계함으로써, 움직임을 최소화하였다. <Table 6>과 <Table 7>은 각각 입고와 출하지시 발생률이 0.04, 0.08일 때의 실험 결과이다.

**Table 6.** Comparision of Heuristic, PPO, and PPO-R ($\lambda$ = 0.04)

| Run | Heuristic | | PPO | | PPO-R | |
|---|---|---|---|---|---|---|
| | Number of dispatched orders | Number of collisions | Number of dispatched orders | Number of collisions | Number of dispatched orders | Number of collisions |
| 1 | 48.4 | 107.1 | 52.54 | 156.6 | 52.84 | 38.62 |
| 2 | | | 49.6 | 161.78 | 54.68 | 31.9 |
| 3 | | | 50.8 | 200.36 | 52.62 | 93.46 |
| 4 | | | 53.76 | 146.8 | 45.92 | 30.32 |
| 5 | | | 51.58 | 143.82 | 52.28 | 23.5 |
| Average | 48.4 | 107.1 | 51.656 (±1.43) | 161.872 (±20.31) | 51.668 (±2.99) | 43.56 (±25.41) |

**Table 7.** Comparision of Heuristic, PPO, and PPO-R ($\lambda$ = 0.08)

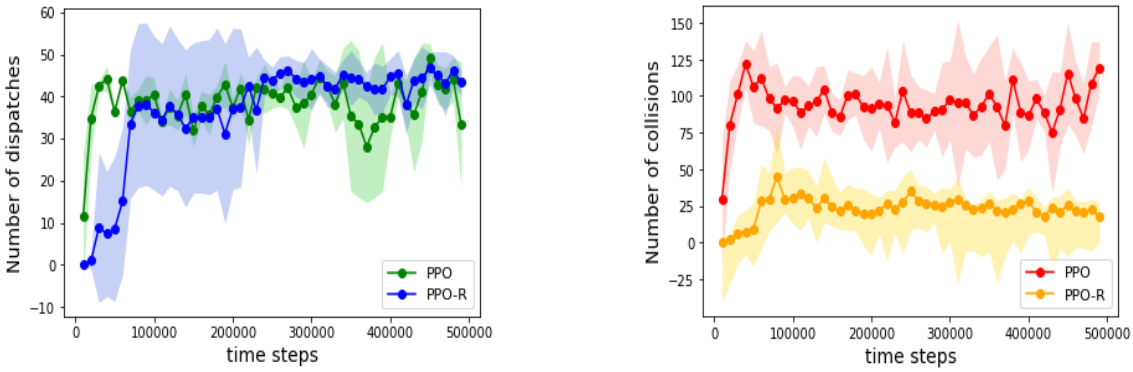
| Run | Heuristic | | PPO | | PPO-R | |
|---|---|---|---|---|---|---|
| | Number of dispatched orders | Number of collisions | Number of dispatched orders | Number of collisions | Number of dispatched orders | Number of collisions |
| 1 | 89.88 | 128.08 | 105.1 | 155.86 | 100.46 | 43.98 |
| 2 | | | 104.16 | 105.78 | 101.94 | 20.46 |
| 3 | | | 105.88 | 120.28 | 100.94 | 12.32 |
| 4 | | | 101.04 | 91.92 | 99.76 | 32.96 |
| 5 | | | 95.36 | 138.68 | 102.32 | 20.6 |
| Average | 89.88 | 128.08 | 102.308 (±3.84) | 122.504 (±22.78) | 101.084 (±0.94) | 26.064 (±11.12) |

PPO와 PPO-R 알고리즘은 휴리스틱에 비해 높은 시간당 출하량을 보인다. 휴리스틱은 λ가 0.04일 때는 시간당 48.4개의 출하량을 기록하는 준수한 출하 처리 성능을 보였지만, λ가 0.08로 증가하여 출하해야 할 파렛트량이 증가할 때는 원활한 출하 처리 업무를 수행하는 데 있어 상대적으로 한계가 존재함을 확인하였다. PPO 알고리즘은 출하 성능이 휴리스틱보다는 좋지만 작업자와의 충돌을 고려하지 않고 학습을 했기 때문에 높은 수의 충돌 횟수를 기록하였다. 반면, PPO-R 알고리즘은 출하 성능은 PPO 알고리즘과 표준편차 내에서 차이가 없지만 충돌 횟수를 1/4 정도로 감소시킬 수 있었다. 이는 충돌에 대한 고려 없이 출하 속도를 최대화할 때는 사람 작업자와 작업 공간이 겹치는 횟수가 높은 정책을 학습하여 잠재적으로 물리적인 충돌이나 운송 업무의 효율성이 떨어질 수 있는 것에 비해 학습 방법의 변화를 통해 충분한 크기의 충돌 비용을 고려하게 함으로써 출하 속도뿐만 아니라 잠재적인 위험성을 고려할 수 있음을 의미한다. 이에 추가적으로 PPO와 PPO-R의 학습곡선(learning curve)과 네트워크 초기화를 도입하지 않았을 때의 알고리즘 성능을 분석함으로써 PPO-R이 다른 방법들에 비해 우수한 시간당 출하량을 가지면서 충돌 횟수를 줄인 결과를 가질 수 있던 점을 분석한다.

<Figure 2>와 <Figure 3>는 PPO와 PPO-R의 학습 곡선으로 두 알고리즘이 출하량을 증가시키는 AGV 운영 방침을 찾아감에 있어 작업자와의 충돌 횟수는 어떻게 변화하였는지를 보여준다. 각 점은 50개의 에피소드를 통해 얻은 출하 횟수와 충돌 횟수의 평균이고, 음영은 95% 신뢰구간을 나타낸다. PPO 알고리즘은 충돌에 대한 비용이 존재하지 않기 때문에 높은 출하량을 얻을 수 있다면 충돌 횟수와 무관하게 정책을 학습시키지만, PPO-R 알고리즘은 출하량을 높이기 위한 정책을 학습하면서도 충돌 횟수가 증가하지 않도록 억제하고 있는 것을 확인할 수 있다.

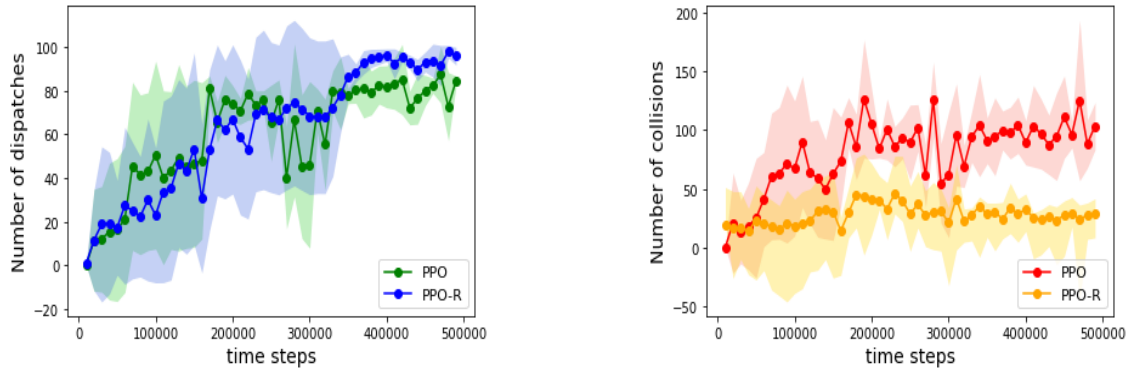
PPO-R이 PPO에 비해 특히 충돌 횟수적인 측면에서 개선된 정책을 찾을 수 있던 것은 네트워크를 초기화하는 학습 방식 변화의 효과가 주요했다. PPO를 학습할 때 PPO-R을 학습시킬 때처럼 충돌 시의 비용 $r_C(s,a,s')$ 값을 동일하게 10으로 설정할 경우 학습이 제대로 이뤄지지 않는다. <Table 8>은 $r_C(s,a,s')$를 고려한 PPO와 PPO-R의 학습 결과를 보여준다.

충돌을 고려한 PPO 알고리즘은 다섯 번의 학습 시도 중 한 번을 제외하고 모두 성공적인 AGV 운영 방법을 학습하지 못하였다. 반대로 PPO-R 알고리즘은 5번의 실험 모두 학습에 성공하였으며, 전체적으로 비슷한 출하량을 기록하였다. 이는 학습 초기에 AGV가 충돌을 지나치게 회피하여 구석에서 머물도록 학습이 되었을 때 PPO는 구석에서 벗어날 수 있는 정




(a) Number of dispatched orders (b) Number of collisions

**Figure 2.** Learning Curves of PPO and PPO-R ($\lambda$ = 0.04)




(a) Number of dispatched orders (b) Number of collisions

**Figure 3.** Learning Curves of PPO and PPO-R ($\lambda$ = 0.08)

**Table 8.** Comparision of PPO with Collision Penalty and PPO-R

| Run | PPO-R Algorithm | | PPO with collision penalty | |
|---|---|---|---|---|
| | Number of dispatched orders | Number of collisions | Number of dispatched orders | Number of collisions |
| 1 | 52.84 | 38.62 | 52.02 | 32.22 |
| 2 | 54.68 | 31.9 | 0 | 0 |
| 3 | 52.62 | 93.46 | 0 | 0 |
| 4 | 45.92 | 30.32 | 0 | 0 |
| 5 | 52.28 | 23.5 | 0 | 0 |
| Average | 51.668 (±2.99) | 43.56 (±25.41) | 10.404 (±20.81) | 32.22 (±0) |

책을 결국에는 학습하지 못하지만 PPO-R은 학습한 것을 잊어버리고 처음부터 다시 도전할 기회를 주었기 때문에 얻을 수 있는 차이이다. 강화학습을 통해 AGV와 사람 작업자와의 충돌을 방지하기 위해서는 충돌 비용을 통해서 AGV가 작업자를 피할 수 있도록 유도해야 하는데, 기존의 PPO 알고리즘으로 학습할 때는 충돌 비용이 적다면 충돌 횟수를 줄이려고 노력하지 않고 충돌 비용이 커지면 금방 출하 업무를 포기하는 문제가 있었다. 하지만, 학습 방법의 변화를 통해 충분한 크기의 충돌 비용을 고려할 수 있게 됨으로써, PPO-R은 작업자와의 충돌 가능성도 고려하면서 출하 업무도 수행하는 AGV 운영 방식을 학습할 수 있었다.

## 6. 결론

본 논문은 자동화된 물류 운송을 위해 AGV가 도입될 때도 사람 작업자가 같은 공간에 위치할 가능성이 큰 출하대기장이라는 공간에서 AGV를 이용한 출하업무 최적화에 대한 문제를 다뤘다. 특히 작업자와 AGV 간의 충돌을 최대한 회피하기 위해 기존의 PPO 알고리즘으로 학습할 때의 문제점을 개선한 PPO-R 알고리즘을 제안하였다. PPO-R 알고리즘은 학습 도중 학습 성능이 충분하게 증가하지 않으면 심층신경망을 새롭게 초기화함으로써 새로운 학습의 기회를 제공한다. 이를 통해 AGV가 작업자와의 충돌 시에 발생하는 높은 충돌 비용을 줄이기 위해 파렛트 운송 업무를 지나치게 회피하고 있는 경우, 처음부터 다시 학습하여 작업자와 충돌을 줄이면서 파렛트 운송 업무도 수행할 수 있게 한다.

가상환경에서 수행된 실험에서 PPO-R을 통해 학습된 AGV는 작업자와의 충돌을 최소화하며 스스로 효율적인 업무 프로세스를 파악하여 최적의 경로로 시간당 출하처리 수를 상승시켰다. 강화학습으로 학습한 PPO와 PPO-R은 목표지점 선정 룰 기반의 휴리스틱에 비해 시간당 출하량을 높은 값을 기록하였으며, PPO-R은 PPO와 휴리스틱에 비해 충돌 횟수를 1/4 정도로 줄일 수 있었다. 적절한 보상 값들을 실험을 통해 확인하고 학습 방식의 변경을 통해 사람 작업자와의 충돌을 고려하면서 우수한 성능을 가진 출하대기장 내 AGV 운영 방침을 얻을 수 있었지만, 출하대기장이라는 한정된 공간을 고려하여 한 대의 AGV를 운영하고 격자 공간을 비교적 단순하게 표현했다는 점 등은 현재의 모델을 물류창고 내 사람 작업자가 존재하는 다른 공간에 적용하기에 한계점을 발생시킬 수 있다. 따라서, 추후 연구에서는 더 복잡한 작업 환경에서 AGV와 사람 작업자와의 상호작용을 고려한 AGV 운영 방침을 찾고자 한다.

## 참고문헌


Agnisarman, S., Lopes, S., Madathil, K. C., Piratla, K., and Gramopadhye, A. (2019), A Survey of Automation-enabled Human-in-the-loop Systems for Infrastructure Visual Inspection, *Automation in Construction*, **97**, 52-76.

Ahn, K. and Park, J. (2021), Cooperative Zone-based Rebalancing of Idle Overhead Hoist Transportations Using Multi-agent Reinforcement Learning with Graph Representation Learning, *IISE Transactions*, **53**(10), 1140-1156.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016), Openai gym, arXiv preprint arXiv:1606.01540.

Chen, M. (1996), A Mathematical Programming Model for AGVs Planning and Control in Manufacturing Systems, *Computers & Industrial Engineering*, **30**(4), 647-658.

Choi, H.-B., Kim, J.-B., Han, Y.-H., Oh, S.-W., and Kim, K.-H. (2022), Collision Avoidance Path Control of Multi-AGV Using Multi-Agent Reinforcement Learning, *KIPS Transactions on Computer and Communication Systems*, **11**(9), 281-288.

He, J., Liu, X., Duan, Q., Chan, W. K. V., and Qi, M. (2023), Reinforcement Learning for Multi-item Retrieval in the Puzzle-based Storage System, *European Journal of Operational Research*, **305**(2), 820-837.

Hill, A., Raffin, A., Ernestus, M., Gleave, A., Kanervisto, A., Traore, R., Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., and Wu, J. (2018), Stable baselines, https://github.com/hill-a/stable-baselines, 2018, Accessed: 2023-04-19.

Hu, H., Yang, X., Xiao, S., and Wang, F. (2023), Anti-conflict AGV Path Planning in Automated Container Terminals Based on Multi-agent Reinforcement Learning, *International Journal of Prodution Research*, **61**(1), 65-80.

Lee, Y. H. and Lee, S. (2022), Deep Reinforcement Learning Based Scheduling Within Production Plan in Semiconductor Fabrication, *Expert Systems with Applications*, **191**, 116222.

Lee, H. and Jeong, J. (2021), Mobile Robot Path Optimization Technique Based on Reinforcement Learning Algorithm in Warehouse Environment, *Applied Sciences*, **11**(3), 1209.

Liu, P., Liu, Z., Wang, J., Wu, Z., Li, P., and Lu, H. (2022), Reinforcement Learning Empowered Multi-AGV Offloading Scheduling in Edge-cloud IIoT, *Journal of Cloud Computing*, **11**(1), 1-14.

Li, M. P., Sankaran, P., Kuhl, M. E., Ptucha, R., Ganguly, A., and Kwasinski, A. (2019), Task Selection by Autonomous Mobile Robots in a Warehouse Using Deep Reinforcement Learning, *In 2019 Winter Simulation Conference (WSC)*, IEEE, 680-689.

Melo, L. C., and Máximo, M. R. O. A. (2019), Learning humanoid robot running skills through proximal policy optimization, In 2019 Latin american robotics symposium (LARS), 2019 Brazilian symposium on robotics (SBR) and 2019 workshop on robotics in education (WRE), IEEE, 37-42

Oyekanlu, E. A., Smith, A. C., Thomas, W. P., Mulroy, G., Hitesh, D., Ramsey, M., Kuhn, D. J., Mcghinnis, J. D., Buonavita, S. C., Looper, N. A., Ng, M., Ng'oma, A., Liu, W., Mcbride, P. G., Shultz, M. G., Cerasi, C., and Sun, D. (2020), A Review of Recent Advances in Automated Guided Vehicle Technologies: Integration Challenges and Research Areas for 5G-based Smart Manufacturing Applications, *IEEE Access*, **8**, 202312-202353.

Puterman, M. L. (2014), *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley & Sons.

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

Sun, S., Hu, J., Li, J., Liu, R., Shu, M., and Yang, Y. (2019), An INS-UWB Based Collision Avoidance System for AGV, *Algorithms*, **12**(2), 40.

Tsai, Y. T., Lee, C. H., Liu, T. Y., Chang, T. J., Wang, C. S., Pawar, S. J., Huang, P. H., and Huang, J. H. (2020), Utilization of a Reinforcement Learning Algorithm for the Accurate Alignment of a Robotic Arm in a Complete Soft Fabric Shoe Tongues Automation Process, *Journal of Manufacturing Systems*, **56**, 501-513.

Zhang, Z. and Zheng, C. (2021), Simulation of Robotic Arm Grasping Control Based on Proximal Policy Optimization Algorithm, Journal of Physics: Conference Series, 2203, 012004.

## 저자소개

**황인근 :** 인하대학교 산업경영공학과 학사과정에 재학 중이다. 연구분야는 최적화, 강화학습이다.

**이현록 :** KAIST 산업및시스템공학과에서 2013년 학사, 2015년 석사, 2020년 박사학위를 취득하였다. 토론토대학교에서 박사 후연구원으로 근무 후 2022년부터 인하대학교 산업경영공학과 교수로 재직하고 있다. 주요 연구분야는 최적화, 강화학습, 시스템 디자인 분야이다.