http://www.jsebs.org
ISSN: 2288-3908

The Journal of Society for e-Business Studies
Vol.25, No.2, May 2020, pp.15-27
https://doi.org/10.7838/jsebs.2020.25.2.015

작업 준비비용 최소화를 고려한 강화학습 기반의 실시간 일정계획 수립기법

Real-Time Scheduling Scheme based on Reinforcement Learning Considering Minimizing Setup Cost

유우식(Woosik Yoo)*, 김성재(Sungjae Kim)**, 김관호(Kwanho Kim)***

초 록

본 연구는 일정계획을 위한 간트 차트(Gantt Chart) 생성과정을 세로로 세우면 일자형만 존재하는 테트리스(Tetris) 게임과 유사하다는 아이디어에서 출발하였다. 테트리스 게임에서 X축은 M개의 설비(Machine)들이 되고 Y축은 시간이 된다. 모든 설비에서 모든 종류(Type)의 주문은 분리 없이 작업 가능하나 작업물 종류가 다를 경우에는 시간지체 없이 작업 준비비용(SetupCost)이 발생한다는 가정이다. 본 연구에서는 앞에서 설명한 게임을 간트리스(Gantris)라 명명하고 게임환경을 구현 하였으며, 심층 강화학습을 통해서 학습한 인공지능이 실시간 스케줄링한 일정계획과 인간이 실시간으로 게임을 통해 수립한 일정계획을 비교하였다. 비교연구에서 학습환경은 단일 주문목록 학습환경과 임의 주문목록 학습환경에서 학습하였다. 본 연구에서 수행한 비교대상 시스템은 두 가지로 4개의 머신(Machine)-2개의 주문 종류(Type)가 있는 시스템(4M2T)과 10개의 머신-6개의 주문종류가 있는 시스템(10M6T)이다. 생성된 일정계획의 성능지표로는 100개의 주문을 처리하는데 발생하는 Setup Cost, 총 소요 생산시간(makespan)과 유휴가공시간(idle time)의 가중합이 활용되었다. 비교연구 결과 4M2T 시스템에서는 학습환경에 관계없이 학습된 시스템이 실험자보다 성능지표가 우수한 일정계획을 생성하였다. 10M6T 시스템의 경우 제안한 시스템이 단일 학습환경에서는 실험자보다 우수한 성능 지표의 일정계획을 생성하였으나 임의 학습환경에서는 실험자보다 부진한 성능지표를 보였다. 그러나 job Change 횟수 비교에서는 학습시스템이 4M2T, 10M6T 모두 사람보다 적은 결과를 나타내어 우수한 스케줄링 성능을 보였다.

ABSTRACT

This study starts with the idea that the process of creating a Gantt Chart for schedule planning is similar to Tetris game with only a straight line. In Tetris games, the X axis is M machines and the Y axis is time. It is assumed that all types of orders can be worked

이 논문은 2017년도 정부(미래창조과학부) 한국연구재단이 후원하는 개인기초연구사업(No.NRF-2017R1D1A1B03035639) 지원을 받아 수행된 연구임.

* Corresponding Author, Professor, Dept. of Industrial and Management Engineering, Incheon National University(wsyoo@inu.ac.kr)

** Co-Author, Master Student, Dept. of Industrial and Management Engineering, Incheon National University (bect3000@naver.com)

*** Co-Author, Associate Professor, Dept. of Industrial and Management Engineering, Incheon National University (khokim@inu.ac.kr)

Received: 2019-10-29, Review completed: 2020-03-12, Accepted: 2020-03-24

without separation in all machines, but if the types of orders are different, setup cost will be incurred without delay. In this study, the game described above was named Gantris and the game environment was implemented. The AI-scheduling table through in-depth reinforcement learning compares the real-time scheduling table with the human-made game schedule. In the comparative study, the learning environment was studied in single order list learning environment and random order list learning environment. The two systems to be compared in this study are four machines (Machine)-two types of system (4M2T) and ten machines-six types of system (10M6T). As a performance indicator of the generated schedule, a weighted sum of setup cost, makespan and idle time in processing 100 orders were scheduled. As a result of the comparative study, in 4M2T system, regardless of the learning environment, the learned system generated schedule plan with better performance index than the experimenter. In the case of 10M6T system, the AI system generated a schedule of better performance indicators than the experimenter in a single learning environment, but showed a bad performance index than the experimenter in random learning environment. However, in comparing the number of job changes, the learning system showed better results than those of the 4M2T and 10M6T, showing excellent scheduling performance.

키워드 : 일정계획, 테트리스, 강화학습, 실시간 스케줄링
Scheduling, Tetris, Reinforcement Learning, Real Time Scheduling

1. 서 론

1.1 연구 배경

제조 시스템은 장비를 효율적으로 운용하기 위해 수학적 최적화 알고리즘이나 휴리스틱 알고리즘 그리고 개미 집단 알고리즘(Ant Colony Algorithm), 타부탐색수법(Tabu Search)와 같은 메타 휴리스틱 알고리즘을 활용하고 있다. 그러나 이러한 전통적인 일정계획 방법은 제조 공정이 복잡하고 관련된 정보량이 많을 경우 최적해를 구하는데 시간이 오래 걸린다는 단점이 있다[12]. 제조업체 및 관련 연구자들은 이러한 문제를 해결하기 위해 다양한 방법을 연구하고 있다.

2013년과 2015년 구글 Deepmind사에서 합성 곱 신경망(Convolution Neural Network)과 심층강화학습(Deep Reinforcement Learning)을 통해 Atari 2006의 게임들을 사람 수준 혹은 그 이상의 성과를 낸데 이어, 2016년 AlphaGo가 성공을 이루면서 실제 환경에 인공지능의 도입 가능성이 타진되었다[9, 13].

본 논문에서는 구글 Deepmind社의 인공지능 기반 머신 러닝을 활용하여 장비 4대와 10대에서 주문량과 주문 종류가 한 쌍으로 구성된 100개의 주문이 동적으로 할당되는 환경을 구축하였다.

본 연구에서 고려하는 동적 할당이란 인공지능을 수행하는 Agent에게 전체 주문 목록을 알려주지 않고 현재 주문과 다음 주문 만 제시하여 실시간으로 의사결정을 내리는 것을 말한다.

본 논문에서는 주문이 동적으로 할당될 때 단일 주문 목록 모델과 임의 주문 목록 모델에 대하여 인공지능 기법을 통해 수행한 결과와 동일한 환경에서 실험자가 수행한 결과를 비교하였다.

1.2 연구 목적

본 연구의 목적은 숫자가 정해진 장비에서 실시간으로 주문량과 주문 종류가 주어졌을 때 실시간으로 일정계획 테이블을 작성하는 인공지능을 학습시키는 것이다. 학습방법은 테트리스 게임 환경에서 합성곱신경망과 심층강화학습을 통하여 학습시키며 학습환경은 단일주문목록 학습환경과 임의 주문 목록 학습환경에서 학습하였다.

본 연구는 Tetris 게임 환경을 실시간 일정계획이라고 생각하고 게임에서 합성곱 신경망을 활용해 일정계획을 학습시켜 피실험자와의 비교연구를 수행하였다. 물론 단순한 인공신경망을 통하여 또는 휴리스틱 원칙을 적용하여 실시간 디스패쳐를 개발하면 더 좋은 성능을 얻을 수 있을 것으로 예상되나 본 논문은 Tetris 유사 게임을 통한 일정계획 연구에 의미를 부여하였다.

학습시 보상점수는 비용개념으로 음수로 표현되는데, 작업물 변경 보상과 총 소요시간 보상, 유휴 장비 보상 점수를 최대화 할 수 있는 장비에 할당 하는 것이다. 작업물 변경보상은 작업준비비용(Setup Cost)이고 총 소요 생산시간 보상은 makespan, 유휴장비보상은 idle time으로 모두 최소화하여야 좋은 지표들이므로 보상점수는 음수를 사용하였다.

본 연구에서 수행한 비교대상 시스템은 두 가지로 4개의 머신(Machine)-2개의 주문 종류(Type)가 있는 시스템(4M2T)과 10개의 머신-6개의 주문종류가 있는 시스템(10M6T) 이다. 생성된 일정계획의 성능지표로는 100개의 주문을 처리하는데 발생하는 Setup Cost, 총 소요시간(makespan)과 유휴가공시간(idle time)의 가중합이 활용되었다.

환경 구축을 위하여 Python과 Pygame을 사용하였으며 인공지능 모델 개발을 위하여 심층신경망 기법과 강화학습을 사용하였다.

단일 주문 목록 학습 모델은 학습시 동일한 주문목록이 계속 사용되므로 학습이 용이할 것으로 생각되며, 임의 주문 목록 학습모델은 학습은 어려울 것이나 실제로 활용될 수 있는 모델이다.

1.3 기존 연구

강화학습을 이용하여 스케줄링을 다양한 방법으로 접근하는 연구가 수행되고 있으나 대부분의 연구는 제안하는 방법론을 기존의 메타휴리스틱 알고리즘과 비교하는 연구를 수행하고 있다.

특히, 최근에 들어서는 DQN(Deep-Q-Network)를 활용하여 제조운영에 활용하는 방법론[7]이 연구가 활발하게 되고 있으며, 납기와 셋업을 모두 고려하는 강화학습 모델[15]과 반도체 공정에서의 응용모델[10, 14]이 제안된 바 있다. 특히, 김진영은 DQN(Deep-Q-Network) 알고리즘을 이용하여 무선 네트워크에서 DQN 패킷 스케줄링 알고리즘을 제안하였다[4]. 김애경은 강화학습을 이용하여 현재 스케줄링 상태를 고려하여 적합한 휴리스틱 원칙을 선택하는 연구[4]를 하였다. 이세훈은 자율주행 AGV를 운용할 수 있는 3D 가상환경을 설계하고 해당 환경에서 강화학습을 기반으로 한 스케줄링을 구현해 실시간으로 변화하는 공장에 적합한 동적인 스케줄링을 설계하였다[8]. 정현석은 잡샵 스케줄링에 대하여 강화학습적 측면에서 접근하는 방식을 제안하였다[2].

해외 연구로는 Jamal Shahrabi는 시뮬레이션과 강화학습을 사용하여 동적 작업장 스케줄링 방법의 성능을 향상 시켰으며[6], Jacob Schrim은 희귀 보상 함수인 테트리스를 간접적인 Low-Level Feature 인코딩을 통해 해결하였다[11].

2. 합성곱신경망과 심층강화학습 기반 일정계획

2.1 문제 정의

모의 실험 환경은 M개의 생산 장비가 있는 병렬기계에서 주문량 (N)개와, 제품 종류 (T)가 한 쌍으로 구성된 현재 주문을 생산 장비 번호 m에 할당하는 환경이다. 주문 O = {(O_N, O_T)}는 총 100개의 주문으로 구성되어 있으며 단일 주문 목록 학습에서는 고정된 100개 1세트의 주문으로 학습을 시키고 임의 주문 목록 학습에서는 임의로 생성된 다른 주문 셑트들로 학습을 시킨다. 본 연구에서 N은 5를 T는 2와 6으로 가정하여 실험하였다. 각 주문은 5개 이하의 임의 생산량과 2나 6 이하의 제품 종류를 갖는 두 정수의 조합이다.

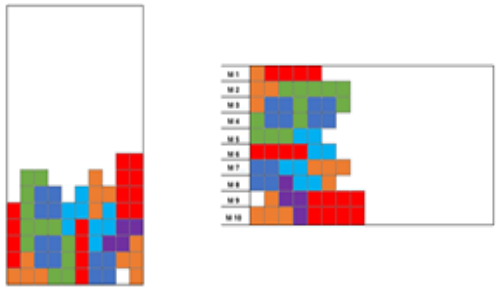
주문 분포는 임의로 난수를 생성하여 발생시켰으며, 각 주문은 작업 순서에 독립적으로 다른 작업과의 선후 관계에 관계없이 작업시간이 동일하다는 의미이다. 각각의 주문은 장비 제약이 없이 모든 기계에서 모든 작업이 가능하며, 한 주문은 분리되지 않고 한 장비에서만 작업되어야 한다는 가정이다. 또한 이때 장비 별 생산성도 동일한 것으로 가정 하였다.

2.2 모의 실험 환경

본 절에서는 앞에서 말한 문제 정의에 맞게 구축한 모의 실험 환경을 설명한다. 모의 실험 환경의 Framework는 Pygame의 Tetrimino를 변형하였으며 State는 장비 별 누적된 주문량, 장비 별 마지막으로 할당한 주문 종류, 현재 주문의 주문량, 현재 주문의 주문 종류, 다음 주문의 주문량, 다음 주문의 주문 종류로 정의된다. 의사결정 Action은 (O_N, O_T)를 할당할 장비 번호 (m)으로 정의되며 보상 점수 Reward 는 작업물 변경 보상 점수, 최대 생산시간 보상 점수, 유휴장비 보상 점수의 합으로 계산된다.

2.2.1 Framework

모의 실험 환경은 <Figure 1>에서 왼쪽 그림과 같은 고전 게임인 Tetris를 90º 회전하면 오른쪽 그림과 같은 일정계획 간트 차트와 매우 유사하다는 아이디어에서 출발하였다. Tetris를 Python으로 옮긴 Pygame의 Tetrimino를 변형하였다. 이하 모의 실험 환경은 간트차트(Gantt Chart)+테트리스(Tetris)를 합쳐 간트리스(Gantris)로 이름을 정하였다.



〈Figure 1〉 Concept Similarity between Tetris and Scheduling Table

2.2.2 환경(Environment) 구축

문제 정의에서 생산장비 M개, 주문량 N개, 제품 종류 T, 주문 (O_N, O_T)를 정의하였다. 간트리스에서 생산 장비 m 은 열(Column)로 표현하며 장비 4개 주문종류 2개 모델은 총 4개의 열로 표현되며 장비 10개 주문 종류 6개 모델은 총 10개의 열로 표현된다. 이하 장비 4개 주문종류 2개는 4M2T 모델, 장비 10개 주문종류 6개는 10M6T모델로 표현한다. 주문량 N은 생성되는 블록의 형태(길이)로 표현되며 본 문제는 각각의 주문은 동시의 여러 장비에서 작업할 수 없도록 정의했기 때문에 기존 테트리스의 블록 형태 'S, Z, O, J, L, T'는 삭제하고 1자 형태인 'I'만 생성된다. 본 연구에서는 N = 5로 가정하여 모든 주문은 길이가 1~5의 길이를 갖는 'I' 블록이 생성된다. 블록 길이는 또한 장비별 생산성이 동일하다고 정의하였기 때문에 제품 'I'를 완료하는데 걸리는 단위 시간을 의미한다.

제품 종류갯수 T는 생성되는 블록의 색상으로 표현되며 4M2T 모델은 2개, 10M6T 모델은 6개의 색상을 갖는다.

환경에서 주문은 현재 의사결정해야 할 현재 주문 (Onow_N,Onow_T)과 다음에 의사결정할 다음 주문 (Onext_N, Onext_T) 2개로 주어진다. 의사 결정이 내려지면 현재 주문 (Onow_N, Onow_T)는 (Onext_N, Onext_T)로 대체되며 다음 주문은 새로 생성된다.

〈Figure 2〉 Gantris Model Built in this Paper(Example of 10M6T)

2.2.3 의사결정(Action)

심층 신경망과 강화학습을 이용한 인공지능 모델들은 의사결정의 수가 많아지면 복잡도가 높아져 Q-Network가 수렴하지 않는 문제를 야기할 수 있다. 따라서 일반적인 연구에서는 의사결정에 특정한 룰을 넣어 의사결정을 최소화한다. 그러나 본 연구에서는 의사결정에 특정한 원칙을 적용하지 않고 현재 할당된 주문 (O_N, O_T)에 대해 장비 m을 선택하는 것으로 정의하였다.

2.2.4 보상점수(Reward)

간트리스에서 현재 주문 (O_N, O_T)에 대한 의사결정 m 을 결정하면 보상이 주어진다. 보상점수는 주문 100개에 대한 보상이 아닌 현재 State에서 내린 의사 결정 1개에 대한 보상을 의미한다.

Reward는 의사결정(Action)이 내려졌을 때 작업물 변경 보상 점수, 총 소요 생산시간 보상 점수, 유휴장비 보상 점수의 가중합으로 이루어진다.

작업물 변경 보상 점수는 의사 결정이 내려진 장비에 대하여 장비에서 작업하는 작업물 종류가 의사결정 전과 후가 다를 경우 음의 점수(-10점)를 가지며 같을 경우 양의 점수(+10점)를 가진다.

총 소요 생산시간 보상 점수는 의사 결정 전 최대 생산 시간과 의사 결정 후의 총 소요 생산시간의 차이로써 현재의 주문을 최대 생산시간인 기계에 배치하면 총 소요 생산시간이 주문량만큼 1에서 5사이에서 증가한다. 따라서 아래 식과 같이 계산하면 1에서 5사이의 음의 점수를 가진다. 그렇지 않을 경우 양의 점수를 가진다.

현재 총 소요 생산시간(makespan)이 증가되지 않은 경우는 makespan에서 배치된 주문의 생산완료시간을 뺀 값을 총 소요 생산시간 보상 점수로 사용하므로 양의 보상 점수를 갖는다.

유휴장비 보상 점수는 의사 결정 이후 최대 생산 시간을 모든 장비의 최대 생산시간으로 정의하고, 장비가 작업을 수행 할 수 있으나 작업이 할당되지 않아 쉬고 있는 시간×가중치로 이루어진다.

본 연구의 목적은 작업물 변경 보상과 총 소요 생산시간 보상 그리고 유휴시간 보상을 최대화 하는 일정 계획을 생성하는 모델을 개발하는 것으로 보상 점수(Reward)는 식은 다음과 같이 정의된다. 아래 idle Points 설명식에서 Board-Width는 설비기계 숫자를 의미하며 4M2T에서는 4, 10M6T에서는 10이다. 즉 makespan 시간 내에서 유휴장비를 의미하는 간트차트의 빈공간을 의미하는데 본 연구에서는 w = 0.25를 사용하였다. 시간의 단위는 Hour로 가정하였다.

Reward = Jobchange points+makespan Points +idle Points

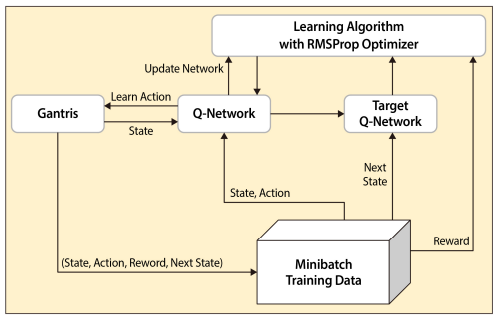
Jobchange points: -10 if Jobchang occur on i
10 else

makespan points : Last makespan-current makespan if makespan increased
else, current makespan-makespan of machine obtained from action

Ilde Points: -(current makespa×BoardWidth -Allocated job)×w

2.3 심층 신경망과 강화학습 기반의 일정 계획 생성 기법

본 절에서는 2.2절에서 구축한 간트리스에서 발생하는 정보를 합성곱 신경망(CNN)의 입력으로 사용, 심층강화 학습(DRL)을 진행하고 보상 점수를 최대화 하는 의사결정 m을 출력으로 도출하는 모델을 개발하였다. 본 연구에 사용된 모델은 구글 딥마인드의 코드를 활용하였다[8].



〈Figure 3〉 The Structure of the Real-Time Schedule Planning Model Used in this Paper

<Figure 3>은 본 논문에서 사용한 일정계획 모델의 구조이다. 간트리스에서는 장비의 수, 장비에 할당된 작업 수, 장비에 마지막으로 할당된 주문 종류, 현재 주문의 주문량 및 종류, 다음 주문의 주문량 및 종류를 생성하게 되면 본 절에서 구축한 모델이 해당 정보를 재배열 및 수정하여 상태(State)를 생성한다. t번째 재배열된 상태(State)는 다음과 같다.

$$State_t = [O_m(1 \le m \le M];\ T_m(1 \le m \le M);\ O_c;\ P_c;\ O_n;\ P_n]$$

M : *Number of Machines*
m : Machine number
O_m : *cumulated Order quantity of machine number*
T_m : *lastest product Type of machine number*
N_c : *the Number of orders in the current order*
T_c : *product Type of Current order*
N_n : *the Number of orders in the Next order*
T_n : *product Type of Next order*

M은 장비의 수량을 나타내며 m은 장비의 번호를 의미한다. O_m은 장비 번호 m에 누적된 주문량을 나타내며, T_m는 장비 번호 m에 마지막으로 할당된 주문 종류를 나타낸다. N_c는 의사결정을 내릴 현재 주문의 주문량을 나타내며, T_c는 의사결정을 내릴 현재 주문의 주문 종류를 나타낸다. N_n은 다음 주문의 주문량을 나타내며, T_n는 다음 주문의 주문 종류를 나타낸다.

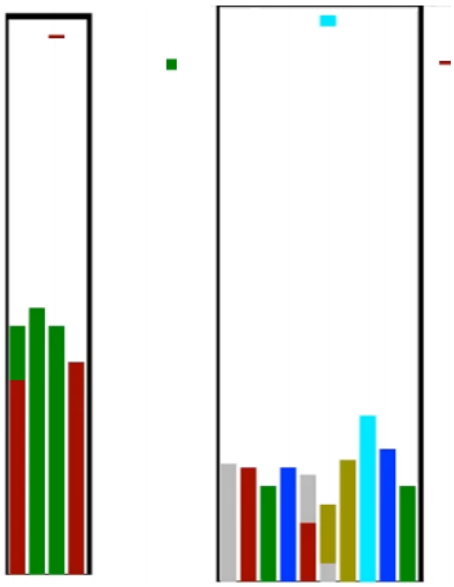
이때 주문 종류 T를 벡터 하나의 크기로 가늠할 경우 인공지능이 주문 종류가 바뀌는 것을 더 좋다고 판단 할 수 있기 때문에 하나의 벡터가 아닌 One-Hot Encoding[1]을 적용했다.

따라서 State size는 다음과 같이 정의된다.

$$\text{State Size} = (1+T)\times M+2\times(1+T),$$

T: product Type

M: Number of Machines



〈Figure 4〉 Gantris Training Snapshot of 4M2T and 10M6T Model

<Figure 4>는 본 논문에서 구축한 간트리스 환경에 인공지능을 연결한 모델이 학습하고 있는 모습으로 왼쪽은 4M2T 모델이며 오른쪽은 10M6T 모델이다.

3. 성능평가

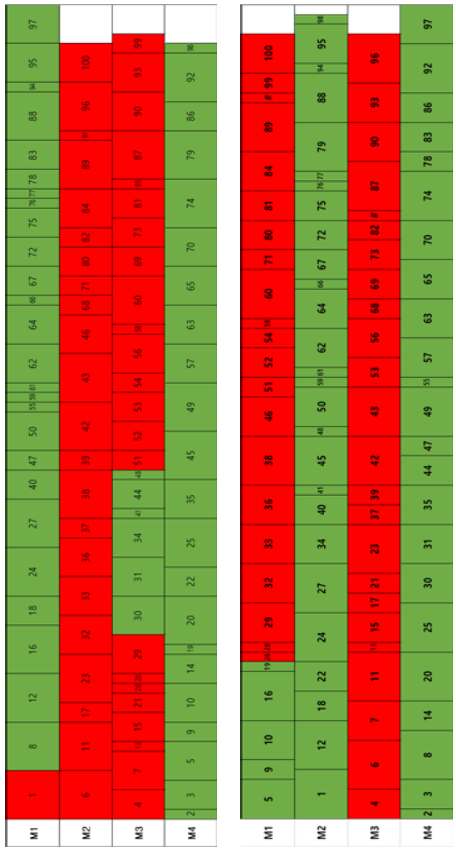
3.1 4M2T 모델의 성능평가

제안 기법을 검증하기 위한 실험데이터의 생성 방법은 다음과 같다. Gantris에서 블록을 생성 할 때 주문량 1~5개, 제품 종류가 2개에 대한 임의의 주문 (O_5, O_2)를 순차적으로 생성한다. 단일주문 학습 모델에서는 생성된 주문 100개를 매 Episode 마다 동일한 순서대로 해당 주문을 이용하여 학습을 진행한다. 학습이 완료된 이후 Q-Value를 저장, 테스트 간트리스 환경에서 수행한 결과와 사람이 테스트 간트리스 환경에서 수행한 결과를 비교하였다.

임의 주문목록 학습환경에서는 매 Episode 마다 완전히 다른 주문 리스트가 생성된다. 학습이 완료된 이후 단일 주문 목록 모델과 동일하게 Q-Value를 저장, 테스트 간트리스 환경에서 수행한 결과와 사람이 테스트 간트리스 환경에서 수행한 결과를 비교하였다. 테스트 간트리스 환경은 단일 주문 목록 모델과 동일한 주문 목록으로 이루어져 있다.

장비가 4대 주문의 종류 T가 2개인 4M2T에 대해서 Training은 1,200개 Episode에 대해 총 12,000 Step을 진행하였으며 Training이 완료된 이후 동일한 주문 세트로 Test를 진행하였다.

모델의 CNN 레이어의 Filter size 48×48, 강화학습의 Discount factor는 0.9, Epsilon greedy policy의 Epsilon은 0.1로 설정하였으며, Replay memory size는 10,000으로 하여 학습을 진행하였다.

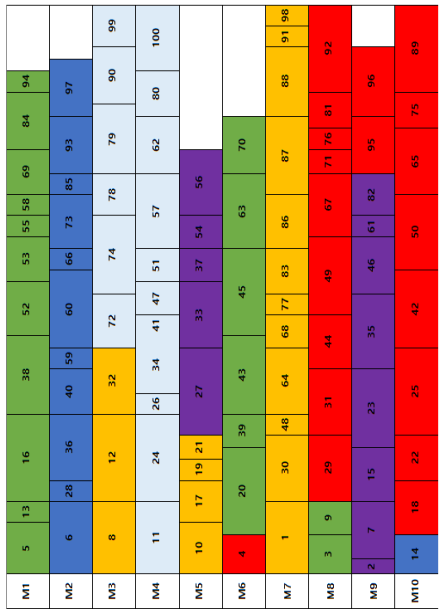


(a) single order list (b) multi order list

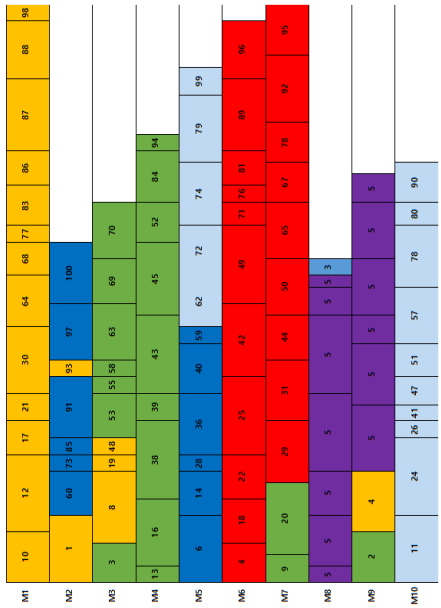
〈Figure 5〉 4M2T Model Test Results

4M2T에 대해 단일 주문 목록 학습모델은 최종 Reward 점수 1194점을 획득하였으며 작업물 변경은 3회 발생하였다, 임의 주문 목록(multi order list) 학습모델은 최종 Reward 점수 1180을 획득하였으며 작업물 변경은 1회 발생하였다.

3.2 10M6T 모델의 성능평가



(a) single order list



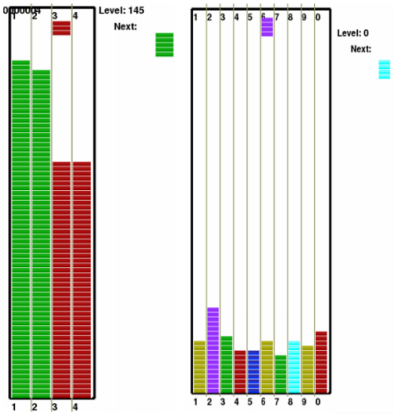
(b) multi order list

〈Figure 6〉 10M6T Model Test Results

장비가 10대 주문의 종류 T가 6개인 10M6T에 대해서도 Training은 1,200개 Episode에 대해 총 12,000Step을 진행하였으며 Training이 완료된 이후 동일한 테스트 주문에 대하여 실험을 진행하였다. 10M6T에 대해 단일 주문 목록 학습모델은 최종 Reward 점수 391점을 획득하였으며 작업물 변경은 6회 발생하였다, 임의 주문 목록 학습모델은 최종 Reward 점수 46점을, 작업물 변경은 10회 발생하였다.

3.3 사람과의 점수 비교

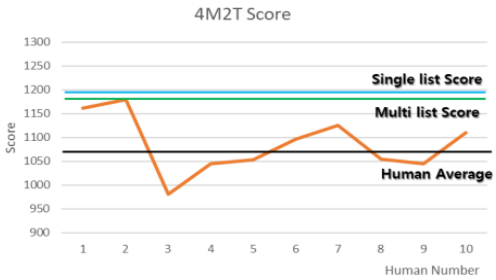
4M2T, 10M6T에 대한 단일 주문 목록 학습모델과 임의 주문 목록 학습모델의 Training 및 Testing 완료 이후 성능 평가를 진행하기 위해 간트리스 모델을 사람이 직접 할 수 있도록 모델을 구축하였다. 보상 체계는 인공지능에서 사용했던 보상체계와 동일하다. 또한 동일한 데이터 셋에 대해 비교하기 위해 각 모델들을 테스트하기 위해 사용한 고정 주문 리스트를 동일하게 이용하였지만 보다 알아보기 쉽게 하기 위해 블록의 가로 크기와 세로크기를 넓히고 가이드라인을 추가하였다.



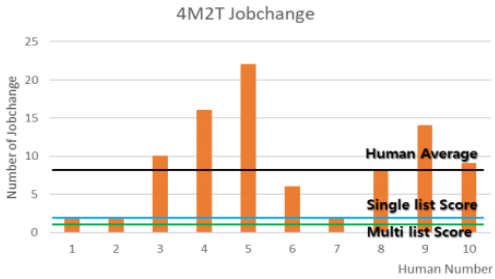
〈Figure 7〉 Human-Performed Test Model

<Figure 8>과 <Figure 9>는 4M2T의 단일 주문 목록 학습모델과, 임의 주문 목록(multi orde list) 학습모델, 사람이 직접 같은 주문 목록으로 수행한 Test 결과 그래프이다. 4M2T에 대해 단일 주문 목록 학습모델은 최종 Reward 점수 1194점을 획득하였으며 작업물 변경은 3회 발생하였다, 임의 주문 목록 모델 은 최종 Reward 점수 1180을 획득하였으며 작업물 변경은 1회 발생하였다.

사람이 직접 수행한 결과는 최소 980점부터 최대 1179점을 획득하였으며 평균 점수는 1085점을 획득하였다. 작업물 변경의 경우 최대 22회, 최소 2회를 기록하였으며 평균적으로 9.1회를 기록하였다.



〈Figure 8〉 Score Graph for 4M2T Model



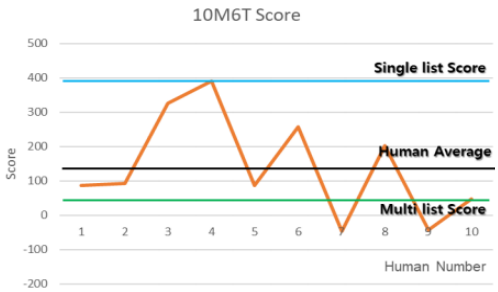
〈Figure 9〉 Job Change Graph for 4M2T Model

사람이 직접 수행한 결과값은 총 10명에 대한 결과로 일정계획을 전혀 모르는 인원부터 일정계획 분야 연구를 수행하고 있는 연구자까지

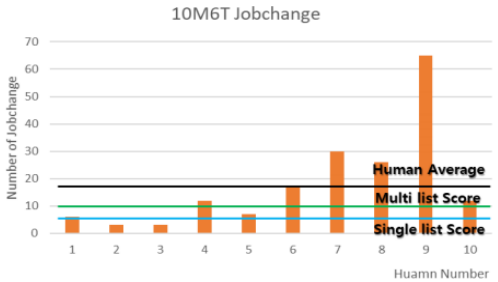
다양한 피실험자가 수행한 결과이다. 실험 방법은 피 실험자에게 본 실험 모델을 10회 연습하도록 하고 3회 측정하여 가장 좋은 성적을 측정하였으며 인공지능 모델과 동일하게 아무런 정보를 부여하지 않았다.

본 실험 결과 4M2T의 경우 단일 주문 목록 학습모델과 복수 주문 목록 학습모델 모두 일정계획 분야에 어느 정도 지식이 있는 사람만큼 혹은 그 이상 수행 할 수 있다는 것을 확인할 수 있었다.

<Figure 10>과 <Figure 11>은 10M6T의 단일 주문 목록 학습모델과, 복수 주문 목록 학습모델, 사람이 직접 수행한 Test 결과 그래프이다. 10M6T에 대해 단일 주문 목록 학습모델은 최종 Reward 점수 391점을 획득하였으며 작업물 변경은 6회 발생하였다, 복수 주문 목록 모델은 최종 Reward 점수 46점을, 작업물 변경은 10회 발생하였다.



〈Figure 10〉 Score Graph for 10M6T Model



〈Figure 11〉 Job Change Graph for 10M6T Model

사람이 직접 수행한 결과는 최소 -42점부터 최대 390점을 획득하였으며 평균 점수는 140점을 획득하였다. 작업물 변경의 경우 최대 65회, 최소 3회를 기록하였으며 평균적으로 18회를 기록하였다. 사람이 직접 수행한 결과값는 총 10명의 피실험자 평균값이다.

4. 결 론

본 연구는 셋업 코스트가 있는 병렬기계에서 머신 러닝을 활용하여 실시간 일정계획 모델을 개발하였다. 모두 가공할 수 있는 장비의 숫자가 정해져 있고 주문의 량, 주문의 종류가 있는 환경에서 실시간으로 주문이 발생할 때 Tetris와 같은 게임 환경에서 사람이 일정계획을 배치하는 것과 학습된 시스템이 하는 상황을 비교 연구하였다. 학습은 고정 주문목록 학습 방법과 임의 주문목록 학습방법의 두 가지를 비교하였다.

개발된 모델을 검증하기 위해서 동일한 주문 세트에 대해 사람이 직접 수행한 결과와 학습시스템의 결과를 Job change 숫자를 중심으로 비교하였다. <Table 1>에서 보는 바와 같이 장비가 4대 작업물의 종류가 2개 있는 방식에선 고정목록 학습모델과 임의목록 학습모델 모두 사람보다 좋은 성능을 발휘하였다. Job change 수가 사람의 평균 9.1번에 비하여 각각 3번 1번으로 우수하게 배치되었다. 장비가 10대 작업물의 종류가 6개 있는 모델에서도 사람의 Job change 수는 18번인데 비하여 고정목록 학습모델과 임의목록 학습모델 각 6번 10번으로 피실험자보다 우수한 성능을 나타내었다.

〈Table 1〉 # of Job Change of 4M2T and 10M6T

system	4M2T	10M6T
Single list Training	3	6
Multi list Training	1	10
Human Average	9.1	18

본 연구에서는 장비 제약이 없는 가정으로 즉, 모든 장비에서 모든 주문을 처리할 수 있는 경우의 연구를 수행하였다. 셋업 코스트가 있는 병렬기계에서 총 생산시간과 작업물 변경을 최소화 하는 문제의 실시간 일정계획모델 환경을 구현하고 간트리스 게임 환경에서 학습 모델을 구현하여 피실험자 군과 비교 연구하였다. 본 비교실험에서 아쉬운 점은 피 실험자군의 실험결과 분포가 매우 크다는 점인데, 이는 피실험자의 일정계획 경험 유무에서 비롯된 것으로 파악된다. 향후 일정계획 전문가 집단에서 피실험자를 선정하여 제안하는 학습시스템과의 비교실험이 필요할 것으로 생각된다. 또한 간트리스 게임에서 작업물 종류 변경시 셋업시간이 추가되도록 보완하여 makespan을 비교하는 실험이 추가 되면 일정계획의 성능평가 지표로 활용 가능할 것이다.

추후 장비 제약이 있는 문제와 주문을 분할하여 할당하는 문제 등과 같이 복잡한 문제로 확장하여 추가 연구가 진행된다면 제조시스템 현장에서도 적용될 수 있을 것으로 판단된다.

References

[1] Beck, J. E. and Woolf, B. P., High-Level Student Modeling with Machine Learning, In: Gauthier, G., Frasson, C., VanLehn, K.(eds) Intelligent Tutoring Systems, ITS 2000, Lecture Notes in Computer Science, Vol, 1839, 2000, Springer, Berlin, Heidelberg.

[2] Jeong et al., “An Neural Network Approach to Job-shop Scheduling based on Reinforcement Learning,” Proceedings of the Korean Society of Computer Information Conference, pp. 47-48, 2018.

[3] Kaplan, R., Sauer, C., and Sosa, A., “Beating atari with natural language guided reinforcement learning,” arXiv preprint arXiv:1704.05539, 2017.

[4] Kim, A., “Ensemble-based Quality Classification and Deep Reinforcement Learning-based Production Scheduling: Ensemble-based Quality Classification and Deep Reinforcement Learning-based Production Scheduling,” KyungHee University, Ph.D. Thesis, 2018.

[5] Kim, J., “Packet scheduling algorithm using deep Q-Network in wireless network”, Yonsei University, M.S Thesis, 2018.

[6] Lee, S.-H., “Reinforcement Learning based AGV Scheduling,” The Korean Society of Computer And Information, Proceedings of the Korean Society of Computer Information Conference, pp. 23-24, 2018.

[7] Lin, C.-C., Deng, D.-J., Chih, Y.-L., and Chiu, H.-T., “Smart Manufacturing Scheduling with Edge Computing Using Multi-Class Deep q Network”, IEEE Transac-

tions on Industrial Informatics, 2019.

[8] Mnih et al., “Human-level control through deep reinforcement learning”, Nature, Vol. 518, No. 7540, pp. 529-533, 2015.

[9] Mnih et al., “Playing Atari with Deep Reinforcement Learning”, Proceedings of the NIPS Deep Learning Workshop, 2013.

[10] Park, I.-B., Huh, J., Kim, J., and Park, J., “A reinforcement learning approach to robust scheduling of semiconductor manufacturing facilities,” IEEE Transactions on Automation Science and Engineering, 2019.

[11] Schrum, J., Evolving Indirectly Encoded Convolutional Neural Networks to Play Tetris With Low-Level Features, Proceedings of the Genetic and Evolutionary Computation Conference, (GECCO), 2018.

[12] Shahrabi, J., Adibi, M. A., and Mahootchi, M., “A reinforcement learning approach to parameter estimation in dynamicjob shop scheduling,” Computers & Industrial Engineering, Vol. 110, pp. 75-82, 2017.

[13] Silver et al., “Mastering the game of Go with deep neural networks and tree search,” Nature, Vol. 529, pp. 484-489, 2016.

[14] Waschneck et al., “Deep reinforcement learning for semiconductor production scheduling”, Proc. 29th. Annu. SEMI Adv. Semicond. Manuf. Conf, 2018.

[15] Yoo, W., Seo, J., Kim, D., and Kim, K., “Machine scheduling models based on reinforcement learning for minimizing due date violation and setup change,” The Journal of Society for e-Business Studies, Vol. 24, No. 3, pp. 19-33, 2019.

저 자 소 개

유우식	(E-mail: wsyoo@inu.ac.kr)
1985년	서울대학교 공과대학 산업공학과 (학사)
1987년	한국과학기술원 산업공학과 (석사)
1991년	한국과학기술원 산업공학과 (박사)
1992년~현재	현재 인천대학교 산업경영공학과 교수
관심분야	물류정보시스템, CAD/CAM, 생산정보시스템

김성재	(E-mail: bect3000@naver.com)
2016년	인천대학교 산업공학과 (학사)
2019년	인천대학교 산업경영공학과 (석사)
2019년~현재	㈜브이엠에스솔루션스 연구원
관심분야	CAD/CAM, 생산정보시스템, 딥러닝

김관호	(E-mail: khokim@inu.ac.kr)
2012년	서울대학교 산업공학과 (박사)
2014년~현재	인천대학교 산업경영공학과 교수
관심분야	산업지능, 통계적기계학습, 빅데이터