J. Korea Saf. Manag. Sci. Vol. 25 No. 4 December 2023
http://dx.doi.org/10.12812/ksms.2023.25.4.131
ISSN 1229-6783(Print)
ISSN 2288-1484(Online)

이종 병렬설비에서 총납기지연 최소화를 위한 강화학습 기반 일정계획 알고리즘

이 태 희* · 김 재 곤** · 유 우 식**
*인천대학교 산업경영공학과 석사 과정 · **인천대학교 산업경영공학과

Scheduling Algorithm, Based on Reinforcement Learning for Minimizing Total Tardiness in Unrelated Parallel Machines

Tehie Lee* · Jae-Gon Kim** · Woo-Sik Yoo**
*Incheon National University Graduate School, Industrial & Management Engineering
**Incheon National University, Dept. of Industrial & Management Engineering

Abstract

This paper proposes an algorithm for the Unrelated Parallel Machine Scheduling Problem(UPMSP) without setup times, aiming to minimize total tardiness. As an NP-hard problem, the UPMSP is hard to get an optimal solution. Consequently, practical scenarios are solved by relying on operator's experiences or simple heuristic approaches. The proposed algorithm has adapted two methods: a policy network method, based on Transformer to compute the correlation between individual jobs and machines, and another method to train the network with a reinforcement learning algorithm based on the REINFORCE with Baseline algorithm. The proposed algorithm was evaluated on randomly generated problems and the results were compared with those obtained using CPLEX, as well as three scheduling algorithms. This paper confirms that the proposed algorithm outperforms the comparison algorithms, as evidenced by the test results.

Keywords : Machine scheduling, Unrelated parallel machine Scheduling, Total tardiness, Reinforcement learning

1. 서 론

본 논문은 작업 준비 시간이 없는 이종 병렬설비 환경에 대하여 작업마다 납기가 존재할 경우, 이를 어기는 정도를 합한 총납기지연을 최소화하는 일정계획을 수립하는 강화학습 기반 알고리즘 개발을 목적으로 한다. 현대 제조 시스템 산업에 있어 일정계획 문제는 고객이 원하는 기간 내에 제품을 출품하여 수요를 만족하기 위해 핵심적으로 해결해야 하는 조합 최적화 문제다. 그러나 이 문제는 작업량이 많아지거나 가동할 수 있는 설비 수가 많아질수록 문제 해결 난이도가 크게 상승하는 NP-hard 문제이다. 그 때문에 실제 현장에서는 주로 간단한 우선순위 규칙을 사용하거나 작업자의 경험을 토대로 직접 일정계획을 수립한다.

본 연구에서는 일정계획 문제를 해결하기 위하여 기존에 연구된 강화학습 기반 일정계획 알고리즘을 응용하여 이종 병렬설비 환경에서 총납기지연을 최소화하기 위한 일정계획을 수립하기 위한 알고리즘을 제안하고, 기존에 연구된 일정계획 알고리즘과의 성능을 비교한다.

일정계획 문제를 해결하기 위한 기존 연구는 다음과 같다. Vepsalainen, Morton(1987)[18]에서는 단일 설비 환경에 대하여 Morton, Rachamadugu (1981)[14]에서 제안한 우선순위 규칙을 응용하여 설비 수 $m=10$ 인 Job Shop에 작업이 포아송 분포를 따라 도착하는 일정계획 문제에서 가중납기지연을 최소화하기 위하여 우선순위 규칙인 Apparent Tardiness Cost 알고리즘을 제안한다. Randhawa, Smith(1995)[16]에서는 작업 준비 시간이

†본 연구는 과학기술정보통신부 및 정보통신기획평가원의 학석사연계ICT핵심인재양성사업의 연구결과로 수행되었음 (IITP-2023-RS-2023-00260175)
†Corresponding Author : Woo-Sik Yoo, Industrial and management Engineering, INCHEON NATIONAL UNIVERSITY, E-mail: wsyoo@inu.ac.kr
Received November 17, 2023; Revision December 4, 2023; Accepted December 18, 2023

있는 환경에서 납기보다 지연된 작업의 수 최소화와 흐름 시간, 설비 부하 최대화로 세 가지 지표를 만족하는 일정계획을 수립하는 것을 목적으로 하여 두 가지 우선순위 규칙을 제안한다. Lin, Hsieh(2014)[13]에서는 작업 준비 시간과 작업 할당 가능 시각이 존재하는 이종 병렬설비 환경에서 가중납기지연을 최소화하기 위한 우선순위 규칙을 제안한다.

Kim et al.(2006)[5]에서는 작업 준비 시간이 있는 이종 병렬설비 환경에서 총납기지연을 최소화하기 위한 Tabu Search 기반 메타휴리스틱 알고리즘을 제안한다. Balin(2011)[1]에서는 작업 준비 시간이 없는 이종 병렬설비 환경에서 총소요시간(Makespan)을 최소화하기 위한 유전 알고리즘 기반 메타휴리스틱 알고리즘을 제안하며, 이후 Balin(2011)[2]에서 작업 시간이 퍼지 이론적 분포를 따른다는 문제 환경을 추가하여 동종 병렬설비 환경에서 총소요시간을 최소화하기 위한 유전 알고리즘 기반 메타휴리스틱 알고리즘을 제안하였다. Lee, Yoo(2023)[11]에서는 수행하는 작업 종류가 바뀌어도 작업 준비 시간이 없는 이종 병렬설비 환경에 작업이 처음부터 모두 도착하며 주문 수량이 정규분포를 따르는 일정계획 문제에서 총소요시간을 최소화하기 위하여 메타휴리스틱 알고리즘인 미미틱(Memetic) 알고리즘을 제안한다.

Kwon et al.(2021)[9]에서는 작업 준비 시간이 없는 이종 병렬설비가 세 구역이 존재하며, 모든 작업이 모두 같은 설비 구역을 방문해야 하는 유연한 흐름 생산 시스템 환경에서의 일정계획 문제에서 총소요시간을 최소화하기 위하여 Transformer 구조[17]를 응용함으로써 조합 최적화 문제의 행렬 데이터를 강화학습 환경에 적용하는 Matrix Encoding Network 구조와 REINFORCE with Baseline 알고리즘[19]을 응용한 Policy Optimization with Multiple Optima (POMO) 알고리즘[8]의 병행을 제안한다. Lee et al.(2023)[10]에서는 작업 준비 시간과 설비별 가능한 작업 제약이 존재하는 동종 병렬설비 환경에서 총납기지연을 최소화하기 위하여 Double DQN 기반의 강화학습 알고리즘을 제안한다.

본 연구는 다음과 같은 기존 연구와의 차별점을 두고 진행된다. [18, 16, 13, 5, 1, 2, 11]에서 제안된 휴리스틱 기반 방법론과는 다르게 본 연구에서는 강화학습에 기반한 방법론으로 접근한다. 또한 [9]에서는 일정계획이 수립된 후 전체에 대하여 발생하는 총소요시간의 최소화가 목적이지만, 본 연구에서는 작업을 할당하는 순서에 따라 각 작업에서 발생하는 총납기지연의 최소화가 목적이기 때문에 그 순서가 추가적인 중요 고려 사항으로 작용한다.

2. 문제 및 수리모형

2.1 문제 설명

본 연구는 이종 병렬설비 환경 중에서 $R_m||Total\ Tardiness$ 라 정의되는 환경에서의 일정계획 문제를 다룬다. 모든 작업은 선후 관계가 없는 독립적인 작업이며, 수행 중인 작업을 도중에 중단하는 것이 불가하다. 또한 작업 준비 시간이 없으며 모든 작업은 모든 설비에 할당할 수 있으나, 동일한 작업임에도 할당되는 설비에 따라 작업 시간이 달라진다. 마지막으로 각 작업은 납기가 존재한다. 본 연구는 이러한 환경에서 일정계획을 수립하였을 때 각 각의 작업이 완료된 시점이 그 작업의 납기를 어긴 정도의 합, 즉 총납기지연을 최소화하는 것을 목적으로 진행한다.

2.2 수리모형

본 연구에서는 2.1장에서 서술한 문제 상황과 동일한 문제에 대하여 De-Alba et al.(2022)[3]에서 수립한 수리모형을 사용한다. 해당 수리모형은 표기법과 함께 식 1~9와 같이 기술된다. 이 수리모형의 의사결정 변수는 Y_{ijl}이며, 이는 설비 i에 할당된 모든 작업 중 작업 j가 l번째 할당되는지 여부를 의미한다. 해당 순서에 할당 되었을 경우에는 1을 부여하며, 그렇지 않으면 0을 부여한다.

$$Minimize \quad \sum_{i=1}^{m}\sum_{l=1}^{n} T_{il}, \tag{1}$$

$$subject\ to$$

$$\sum_{i=1}^{m}\sum_{l=1}^{n} Y_{ijl} = 1 \,\forall\, j = \{1,\ldots,n\}, \tag{2}$$

$$\sum_{l=1}^{n} Y_{ijl} \le 1 \,\forall \begin{cases} i = \{1,\ldots,m\} \\ l = \{2,\ldots,n\} \end{cases}, \tag{3}$$

$$C_{il} = \sum_{j=1}^{n} p_{ij} Y_{ijl} \,\forall\, i = \{1,\ldots,m\}, \tag{4}$$

$$C_{il} \ge C_{i(l-1)} + \sum_{j=1}^{n} p_{ij} Y_{ijl} \,\forall \begin{cases} i = \{1,\ldots,m\} \\ l = \{2,\ldots,n\} \end{cases}, \tag{5}$$

$$T_{il} \ge C_{il} - \sum_{j=1}^{n} d_j Y_{ijl}, \,\forall \begin{cases} i = \{1,\ldots,m\} \\ l = \{1,\ldots,n\} \end{cases}, \tag{6}$$

$$C_{il} \ge 0, \,\forall \begin{cases} i = \{1,\ldots,m\} \\ l = \{1,\ldots,n\} \end{cases}, \tag{7}$$

$$T_{il} \ge 0, \,\forall \begin{cases} i = \{1,\ldots,m\} \\ l = \{1,\ldots,n\} \end{cases}, \tag{8}$$

$$Y_{ijl} \in \{0,1\}, \,\forall \begin{cases} i = \{1,\ldots,m\} \\ j = \{1,\ldots,n\} \\ l = \{1,\ldots,n\} \end{cases}, \tag{9}$$

ISSN 1229-6783(Print)
ISSN 2288-1484(Online)

<Table 1> Notation for Equation 1~9

Variable	Definition
d_j	Due date of job j
p_{ij}	Processing time of job j in machine i
C_{il}	Completion time of l-th dispatched job on machine i
T_{il}	Tardiness of l-th dispatched job on machine i
Y_{ijl}	Decision Variable. Is 1 if Job j was processed in l-th sequence on machine i, 0 otherwise

3. 기존 일정계획 알고리즘

3.1 우선순위 규칙

실제 제조 시스템 환경에서 일정계획을 수립할 때는 우선순위 규칙을 기반으로 두는 휴리스틱 알고리즘이 주로 사용된다. 우선순위 규칙은 최적해를 보장할 수 없다는 단점이 존재하지만, 직관적이며 간단하게 구현할 수 있고 실행하는 데에 고사양 장비가 필요하지 않기 때문이다.

본 연구에서 비교군으로써 인용한 우선순위 규칙은 가장 간단한 우선순위로 작업을 정렬하고 설비에 할당하는 Earliest Due Date - Minimum Completion Time 알고리즘(이하 “EDD-Min”)과 Vepsalainen, Morton(1987)[18]에서 제안한 Apparent Tardiness Cost 알고리즘(이하 “ATC”)으로 총 두 가지이다.

EDD-Min 알고리즘으로 일정계획을 수립할 경우, 작업의 할당 순서를 납기가 급한 순으로 정렬한다. 정렬된 순서에 따라 할당하려는 작업과 그 순서가 되었을 때 설비의 운용 현황을 관측하고, 현 작업을 가장 이르게 완료하는 설비에 할당한다.

ATC 알고리즘으로 일정계획을 수립할 경우, 작업의 할당 순서는 사전에 정렬되지 않는다. 설비의 운용 현황을 관측하여 가장 첫 작업을 할당받아야 하는 설비나 직전 작업을 다른 설비보다 이르게 완료하여 다음 작업을 할당받을 수 있는 설비 즉, 작업 할당이 필요한 설비가 발생하는 시점에 아직 할당되지 않은 작업마다 점수를 계산하여 가장 높을 점수를 가지는 작업을 할당한다. 부여되는 점수를 계산하는 식은 [18]에서 제안되나, 현 문제 상황은 작업 및 설비의 조합에 따라 작업 시간이 이미 정해져 있어서 설비의 속도를 나타내는 v_{ij}는 1로 고정되어 있다. 이에 맞추어 수정된 점수 계산식은 식10과 같다.

$$ATC(t_i)_j = \frac{1}{p_{ij}} \exp\left(-\frac{\max(d_j - p_{ij} - t_i, 0)}{k\bar{p}_i}\right) \quad (10)$$

<Table 2> Notation for Equation 10

Variable	Definition
t_i	Ready time of machine i
$\bar{p}_i$	Average processing time of un-allocated jobs in machine i
k	Look-ahead parameter that scales the slack according to the expected number of competing jobs

3.2 메타휴리스틱 알고리즘

메타휴리스틱 알고리즘은 규칙 기반 알고리즘에 비하여 연산 시간이 느리고 문제의 크기에 따라 중간 사양 이상의 장비가 필요하다는 단점이 존재하지만, 최적해에 가까운 해를 얻을 수 있다는 큰 장점이 있어서 일정계획 문제를 포함한 조합 최적화 문제를 해결하기 위하여 사용된다. 메타휴리스틱 방법론 중 생물학적 진화에서 착안하여, 해 집단을 생성하고 해 공간을 탐색하는 유전 알고리즘[4, 6]이 주로 사용된다. 이 알고리즘에 관한 수많은 연구가 진행되면서 다른 방법론을 함께 적용함으로 기존 알고리즘의 성능을 보강하는 혼합형 알고리즘이 제안되었는데, 유전 알고리즘의 진화 연산을 완료한 해 집단에 대하여 해마다 지역 탐색 연산을 하는 Moscato(1989)[15]에서 제안한 미미틱(Memetic) 알고리즘(이하 “MA”)이 대표적이다.

본 연구에서 비교군으로써 인용한 메타휴리스틱 기반 일정계획 알고리즘은 Lee, Yoo (2023)[11]에서 제안하는 MA 기반 일정계획 알고리즘이다. 진화 연산 과정과 지역 탐색 연산 과정, 알고리즘 최적화를 위한 하이퍼파라미터는 [11]에서 제안한 수치와 동일하게 적용한다. 그러나 총소요시간을 최소화함이 목적이었던 [11]에서와 달리 본 연구는 총납기지연을 최소화함이 목적이므로 총소요시간에 기반을 두었던 유전자 적합도 평가지표를 총납기지연에 기반을 두도록 변경한다.

4. 강화학습 기반 일정계획 알고리즘

일정계획, 외판원 문제와 같은 조합 최적화 문제를 해결하는 방법론으로 인공신경망을 정책 네트워크로 접목한 강화학습 알고리즘이 최근 활발히 연구되고 있다. 충분히 훈

련된 강화학습 알고리즘을 사용한다면 메타휴리스틱 알고리즘보다 빠른 일정계획 수립 속도로 우선순위 규칙보다 더 좋은 해를 기대할 수 있다는 강점이 있다. 그러나 강화학습 알고리즘을 조합 최적화 문제에 적용하기 위해서는 문제 설정을 기반으로 Markov Decision Process (이하 "MDP")를 구축이 필수인데 많은 문제가 MDP 구축이 어렵다는 점과 정책 네트워크가 수렴하기까지 알고리즘을 훈련하는 데에 오랜 시간이 소요될 수 있다는 점, 고사양의 장비가 필요하다는 점 등의 단점이 존재한다.

강화학습 알고리즘을 본 연구에서 다루는 문제 환경에 적용하기 위해 MDP를 구축하는 방법은 다음과 같다. 상태(State)는 n개의 작업과 m개의 설비에 대하여 작업 및 설비의 조합에 따른 작업 시간, 그리고 각 작업의 납기를 행렬 형식으로 나열한 $n\times(m+1)$ 크기의 행렬 데이터로 정의한다. 이 상태를 관측한 정책 네트워크가 내리는 의사결정인 행동(Action)은 두 가지이다. 먼저, 작업 및 설비의 조합에 따른 인코딩된 정보를 담은 n×m 크기의 행렬 데이터를 산출하는 것이 첫 번째 행동이다. 두 번째 행동은 작업 할당이 필요한 설비가 발생하는 시점마다 수행한다. 해당 설비와 아직 할당되지 않은 작업에 대한 인코딩된 정보를 기반으로 각 작업의 할당 적합 점수를 산출하여 가장 높은 점수를 받은 작업을 설비에 할당하는 것이다. 마지막으로 일정계획이 수립될 때까지 정책 네트워크가 수행한 모든 행동을 통해 얻게 되는 보상(Reward)은 일정계획이 수립된 이후 산출되는 총납기지연을 부호를 반전시켜 정의한다. 총납기지연은 비용이므로 최소화를 목적으로 두어야 하며, MDP는 정책 네트워크가 보상을 최대화하도록 구축되어야 하기 때문이다.

본 연구에서는 서술한 내용과 같이 구축된 MDP에 적용할 수 있는 강화학습 기반 일정계획 알고리즘을 제안한다.

4.1 강화학습 알고리즘

Kwon et al.(2020)[8]에서 제안하는 Policy Optimization with Multiple Optima (이하 "POMO") 알고리즘은 첫 번째 의사결정을 내릴 수 있는 지점이 여럿 존재하는 조합 최적화 문제를 해결하기 위하여 REINFORCE with Baseline 알고리즘[19]을 응용한 알고리즘이다. 일정계획 문제에서 작업을 할당받을 첫 번째 설비, 외판원 문제에서 외판원이 출발할 첫 번째 도시와 같이 조합 최적화 문제는 의사결정이 이루어지는 가장 첫 번째 지점이 존재한다. 기존의 강화학습 알고리즘은 하나의 첫 지점을 고정한 뒤 도출되는 해를 기반으로 정책 네트워크를 훈련한 이후, 다른 첫 지점을 지정하는 것을 반복한다. 훈련이 잘 이루어진다면 최적해를 기대할 수 있지만, 정책 네트워크가 특정 지점을 첫 지점으로써 선호하도록 편향되어 훈련될 가능성이 존재한다. 이 때문에 POMO 알고리즘은 모든 첫 지점에서 같은 최적해를 구할 수 있다는 일부 조합 최적화 문제의 특성을 이용하여 정책 네트워크가 같은 문제를 다양한 시각에서 여러 번 해결하도록 가능한 모든 첫 지점에서의 해를 도출시킨 뒤, 그 결과를 취합하여 훈련함으로써 첫 지점에 대한 편향을 해소한다.

POMO 알고리즘은 REINFORCE with Baseline 알고리즘을 기반으로 하여 경사상승법을 따르도록 설계되었다. 기반이 된 알고리즘에서는 경사상승법 계산을 위해 추출되는 경사 사이의 분산을 줄이는 목적으로 기준($bi(s)$, Baseline)가 설정되어야 하는데, 이전 연구에서는 정책 네트워크가 도출하는 해마다 개별의 기준을 설정하였다. 그러나 POMO 알고리즘은 한 번의 훈련 Batch에서 도출되는 다양한 문제에 대한 여러 개의 해의 평균을 공유된 기준으로 설정함으로써 세 가지 차별점을 제시한다. 제시되는 첫 번째 차별점으로는 경사상승법에서 가치 기반 기준을 통해 산출되는 이점이 주로 음의 값을 가지는 데에 비하여 공유된 기준을 통해 산출되는 이점의 평균이 0에 가깝다는 점이다. 두 번째 차별점으로는 이점을 계산하기 위한 개별의 가치 추정 네트워크가 필요한 이전 연구와 달리 공유된 기준을 사용하면 가치 추정 네트워크가 필요 없어 효과적인 계산이 가능해진다는 점이다. 마지막으로 가치 기반 기준을 사용하면 유사한 상태를 경험한 해 집단마다 독립적으로 작용하여 각 해 집단 안에서만 비교가 이루어지기 때문에 지역 최적해에 빠질 가능성이 우려된다. 이에 비해 공유된 기준을 사용하면 다른 상태를 경험한 해끼리도 비교가 이루어지기 때문에 모든 해가 서로 관계성을 띠게 되며, 적절한 수의 병렬적인 해를 통해 지역 최적화를 방지한다는 차별점이 존재한다.

기존 강화학습 알고리즘은 정책 네트워크를 훈련할 때 다양한 상태와 해를 경험하기 위하여 확률 기반 정책을 사용하여 탐험 단계를 거치며, 이 네트워크를 시험할 때 가치 기반 정책을 사용하여 하나의 해를 도출함으로써 활용 단계를 거친다. 탐험 단계에서의 정책 네트워크는 평균적으로 활용 단계보다 낮은 성능을 보이나, 계산 능력 범위 내에서 반복할 수 있으며 활용 단계보다 높은 성능을 보이는 순간도 존재하는 등 불안정성이 존재한다. POMO 알고리즘은 탐험 단계에서도 가치 기반 정책을 사용하는데, 한 번의 훈련 Batch 내에 가능한 모든 첫 지점의 개수만큼 해가 존재하기 때문에 기존 알고리즘보다 적은 횟수로도 충분한 탐험을 기대할 수 있다.

본 연구에서는 서술한 특징을 근거로 정책 네트워크를 훈련하기 위한 강화학습 알고리즘으로 POMO 알고리즘을 채택한다.

ISSN 1229-6783(Print)
ISSN 2288-1484(Online)

4.2 정책 네트워크

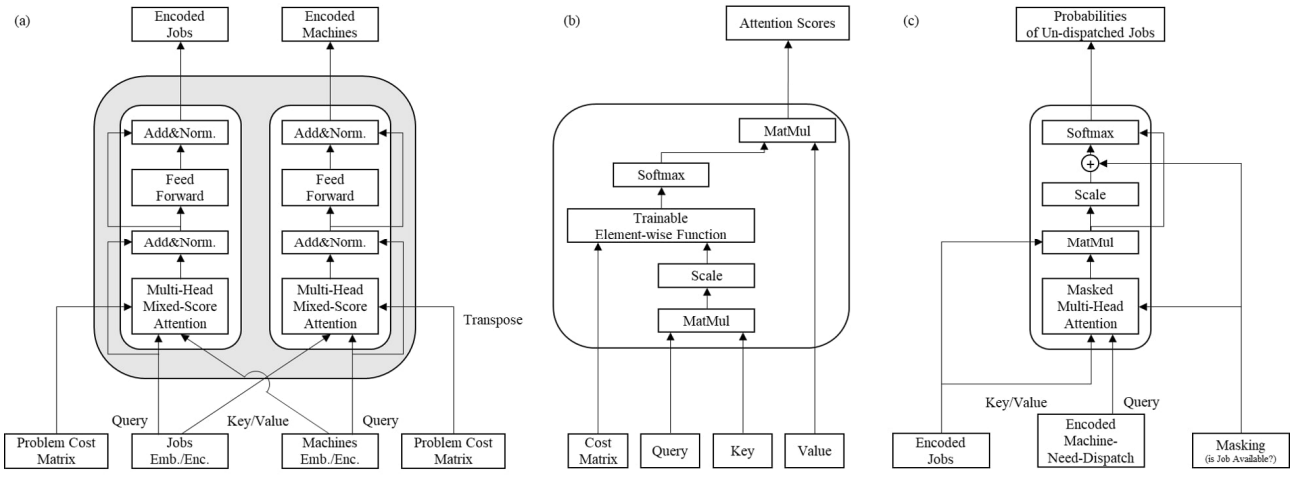
Kwon et al.(2021)[9]에서는 자연어 처리 분야에서 제안된 Transformer 구조[17]를 응용하여 행렬 형식의 조합 최적화 문제 데이터를 처리하기 위한 Matrix Encoding Networks(이하 "MatNet") 구조를 제안한다. 기존 Transformer 구조를 사용하여 여러 줄의 문장 대신 행렬 형태의 데이터를 처리할 때 각 문장을 독립적으로 연산하듯 각 행에 대하여 독립적인 Self-Attention 연산을 통해 인코딩된 정보를 도출하는 기존 연구인 Kool[7]에서와는 달리 MatNet 구조는 [Figure 1-(a)]에서 설명하는 인코더(Encoder)의 구조와 같이 Mixed-Score 연산법을 Multi-Head Attention 연산에 도입하여 각 행과 각 열에 대한 교차적인 연산 이후 원본 문제 데이터에 대한 정보와 교차적인 연산을 통해 Attention 점수를 도출한다. [Figure 1-(b)]는 Multi-Head Mixed-Score Attention 연산 과정을 설명하며, 이 과정에서 Mixed-Score 연산법을 수행하는 과정은 Trainable Element-wise Function이다. [9]에서는 해당 과정을 Query와 Key를 연산한 교차적인 정보 행렬과 원본 문제 데이터 행렬을 쌓은 뒤, 두 개의 선형 은닉층을 통해 두 행렬을 입력받아 하나의 행렬을 산출하는 인공신경망으로 구현한다. 이를 통해 각 행과 각 열, 원본 문제 데이터 사이의 교차적인 정보를 담은 인코딩된 정보를 산출할 수 있게 된다. 이 정보는 [Figure 1-(c)]에서 설명하는 구조를 가지는 디코더(Decoder)를 통해 설비마다 각 작업에 대한 할당 적합 점수를 산출하기 위한 입력 행렬로 쓰인다. 디코더가 점수를 산출하는 과정에서 과거에 이미 할당된 작업은 Masking 처리하므로 그 작업의 점수는 모두 0점으로 결정된다. 본 연구에서는 서술한 특징을 근거로 MatNet 구조를 기반으로 정책 네트워크를 구축하여 의사결정을 하는 데에 사용한다.

구축된 MDP와 제안 알고리즘으로 일정계획을 수립하는 과정은 다음과 같다. 정책 네트워크 중 인코더가 일정계획이 필요한 문제의 데이터를 관측하여 작업 및 설비의 조합에 따른 인코딩된 정보가 담긴 행렬 데이터를 산출한다. 작업 할당이 필요한 설비에 대하여 아직 할당되지 않은 작업과의 조합에 따른 인코딩된 정보를 디코더에 입력하여 할당 적합 점수를 산출하고 그 점수가 가장 높은 작업이 할당된다. 일정계획은 모든 작업이 설비에 할당되어 모두 완료되는 때에 수립되며, 이 일정계획에서 본 연구의 최소화 목적 대상인 총납기지연이 산출된다. 이 과정을 그림으로 설명한 것이 [Figure 2]이며, 정책 네트워크의 첫 번째 행동 결과인 설비 i와 작업 j에 대한 인코딩된 정보는 $info_{ij}$로 표기하며, 두 번째 행동 결과인 설비 i와 작업 j에 대한 할당 적합 점수는 $score_{ij}$로 표기한다.

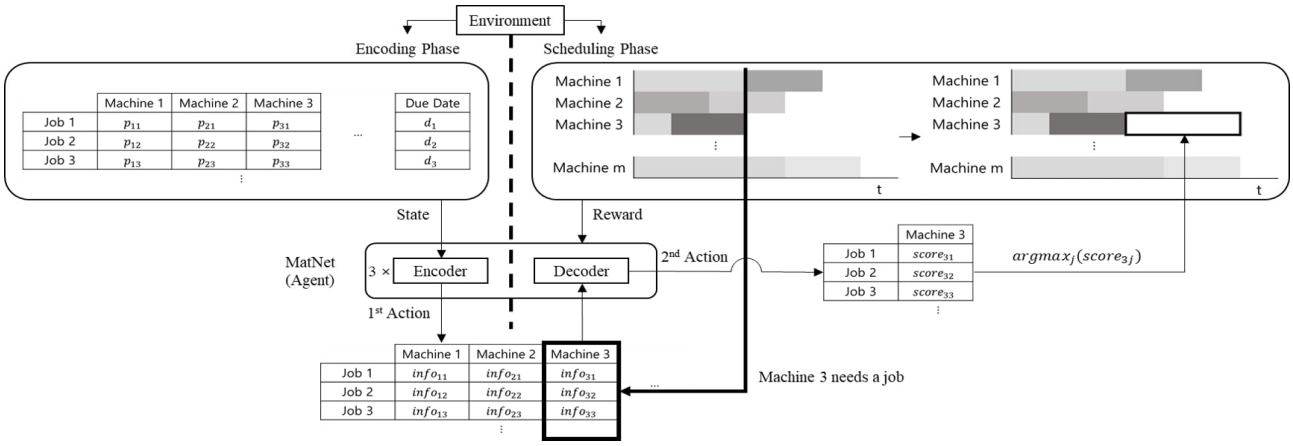
5. 성능 평가 실험

5.1 실험 방법

본 연구에서 다룰 문제 상황 조건에 맞추어 문제 크기마다 10개씩 생성하였다. 먼저, 문제 상황의 크기는 작업 수 $n = \{50, 100\}$ 수준과 설비 수 $m = \{5, 10\}$ 수준으로 지정하여 문제 크기를 4가지로 정의한다. 각 작업의 수량은 작업 수마다 평균 30, 표준편차 12의 정규분포를 따르도록 임의 생성한다. 같은 작업 종류에 대하여 각 설비는 색인 값이 클수록 느리거나 같은 시간을 소요하도록 세 가지 경우의 수 {1, 2, 3}에서 작업 종류별 개당 작업 시간에



[Figure 1] (a) Overviewing figure of the MatNet encoder architecture. (b) Flow chart of Multi-Head Mixed-Score Attention. (c) Overviewing figure of decoder architecture.



[Figure 2] Explaining how the proposed algorithm solves a scheduling problem

서 임의 지정한다. 각 작업 및 각 설비에 따른 작업 시간은 작업의 수량과 해당 종류 작업 개당 작업 시간의 곱으로 결정되어 ($n \times m$) 크기의 행렬로 표현된다.

각 작업 및 각 설비에 따른 작업 시간이 결정된 이후 Lee, Pinedo(1997)[12]에서 제안하는 작업별 납기 계산식을 기반으로 각 작업에 납기를 부여하며, 이 식은 식 11~13과 같이 정의된다. 이 중 납기가 급한 작업의 비율을 나타내는 변수 τ는 [0, 1]의 범위에서 결정되며 1에 가까울수록 납기가 급한 작업의 수가 많아진다. 또한 납기의 분산을 결정하는 변수 R는 [0, 1]의 범위에서 결정되며 1에 가까울수록 납기의 분산이 커진다.

$$\hat{C}_{max} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}}{n \times m} \times \frac{n}{m}, \tag{11}$$

$$\bar{d} = (1-\tau)\hat{C}_{max}, \tag{12}$$

$$\begin{cases} d_j \sim U((1-R)\bar{d}, \bar{d}) & with\ probability\ 1-\tau \\ d_j \sim U(\bar{d},\ \bar{d} + (\hat{C}_{max} - \bar{d})R) & with\ probability\ \tau \end{cases} \tag{13}$$

<Table 3> Notation for Equation 11~13

Variable	Definition
p_{ij}	Processing time of job j in machine i
$\hat{C}_{max}$	Predicted total completion time
τ	Tightness of Tardiness
R	Dispersion of tardiness

각 일정계획 알고리즘이 수립한 결과에 대한 성능지표로 사용하기 위하여, 2.2장에서 서술된 수리모형을 최적화 Solver 소프트웨어 'CPLEX'로 구현하였다. 실제 현장에의 적용을 고려하여 CPLEX의 실행시간은 최대 1시간으로 하였으며, 1시간 이내에 최적의 일정계획을 수립하지 못하면 1시간 동안 탐색하며 수립한 지역 최적 일정계획을 성능지표로 사용한다.

모든 일정계획 알고리즘은 Python을 이용하여 구현되었다. 우선순위 규칙의 성능은 각 실험 데이터세트에 알고리즘을 적용하여 수립되는 일정계획의 총납기지연을 기록하여 확인한다. 미미틱 알고리즘은 각 실험 데이터세트에 적용할 때마다 연속으로 개선되지 않는 횟수 1,000회 혹은 Python 상 구동 시간 1시간 제한을 제약 조건으로 구축하였으며, 제약 조건 이내에 탐색 및 수립한 일정계획 중 총납기지연을 기준으로 최적인 일정계획을 통해 알고리즘의 성능을 확인한다. 제안 알고리즘은 훈련 단계에서는 실험 데이터세트에 적용하지 않는다. 대신 한 번의 훈련 에피소드마다 10,000개의 훈련 데이터세트를 새로 임의 생성하여 알고리즘을 훈련한다. 매 훈련은 48시간으로 제한하며, 훈련을 마친 알고리즘을 실험 데이터세트에 적용하여 수립되는 일정계획의 총납기지연으로써 알고리즘의 성능을 확인한다.

본 연구에서 제안 알고리즘에 적용된 MatNet 구조는 Transformer 구조에 기반을 두었기 때문에 작업의 개수에 상관없이 연산이 가능한 유연성을 가지고 있다. 실무적인 관점에서 이 유연성은 훈련에 투자하는 시간 등의 비용을 줄일 수 있다는 장점을 불러올 수 있다. 따라서 본 연구에서는 제안 알고리즘의 성능을 검증하기 위하여 모든 문제 크기마다 알고리즘을 훈련한 결과뿐만 아니라, n = 50, m = 5 크기의 문제에서 훈련한 알고리즘을 n = 100, m = 5 크기의 문제에서 실험한 결과와 n = 50, m = 10 크기의 문제에서 훈련한 알고리즘을 n = 100, m = 10 크기의 문제에서 실험한 결과를 구한다.

ISSN 1229-6783(Print)
ISSN 2288-1484(Online)

<Table 4> Result table of scheduling algorithms:
Ours trained and tested for each problem size

Problem Size		Algorithm	Total Tardiness	RPD (%)	CPU Time (sec.)	Problem Size		Algorithm	Total Tardiness	RPD (%)	CPU Time (sec.)
m = 5	n = 50	EDD-Min	2353	142.85	< 1	m = 10	n = 50	EDD-Min	1498	49.55	< 1
		ATC	1328.7	37.13	< 1			ATC	1279.3	27.71	< 1
		MA	1160.72	19.80	1031			MA	1122	12.01	1049
		Ours	1019	5.17	2.1			Ours	1094.8	9.29	2.7
		CPLEX	968.9	0	1722.4			CPLEX	1001.7	0	3600
	n = 100	EDD-Min	8468.8	255.06	< 1		n = 100	EDD-Min	4376.3	59.02	< 1
		ATC	3206.6	34.44	< 1			ATC	3238.8	17.69	< 1
		MA	3069.22	28.68	1848			MA	3126.65	13.61	2044
		Ours	2489.9	4.39	4.8			Ours	2805.2	1.93	7.2
		CPLEX	2385.2	0	3600			CPLEX	2752	0	3600

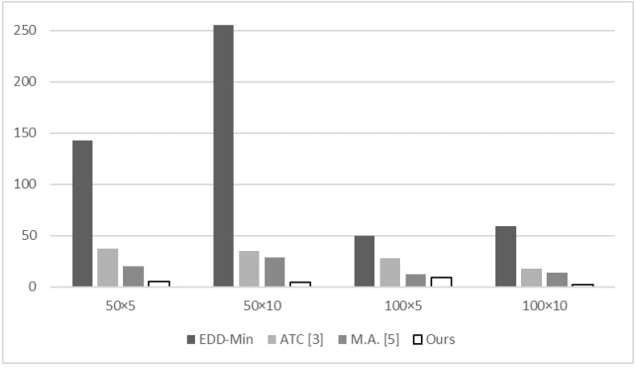
<Table 5> Result table of scheduling algorithms:
Ours trained in problem sizes of 50 jobs for each number of machine

Problem Size		Algorithm	Total Tardiness	RPD (%)	CPU Time (sec.)	Problem Size		Algorithm	Total Tardiness	RPD (%)	CPU Time (sec.)
m = 5	n = 50*	EDD-Min	2353	142.85	< 1	m = 10	n = 50*	EDD-Min	1498	49.55	< 1
		ATC	1328.7	37.13	< 1			ATC	1279.3	27.71	< 1
		MA	1160.72	19.80	1031			MA	1122	12.01	1049
		Ours	1019	5.17	2.1			Ours	1094.8	9.29	2.7
		CPLEX	968.9	0	1722.4			CPLEX	1001.7	0	3600
	n = 100	EDD-Min	8468.8	255.06	< 1		n = 100	EDD-Min	4376.3	59.02	< 1
		ATC	3206.6	34.44	< 1			ATC	3238.8	17.69	< 1
		MA	3069.22	28.68	1848			MA	3126.65	13.61	2044
		Ours	2764	15.88	4.8			Ours	3225	17.19	7.2
		CPLEX	2385.2	0	3600			CPLEX	2752	0	3600

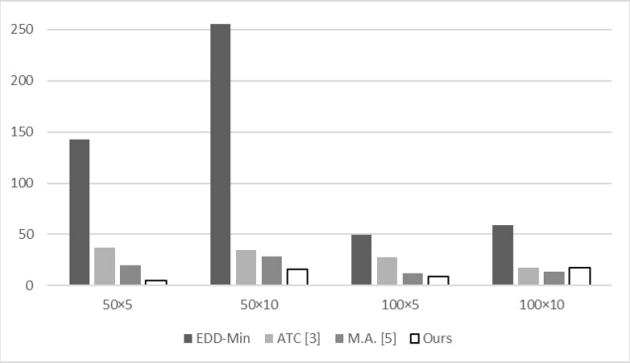
* Problem Size for training Ours

5.2 실험 결과

<Table 4>은 제안 알고리즘을 모든 문제 크기마다 훈련하여 실험 데이터세트에 적용한 첫 번째 실험의 결과를 정리한 표이며 <Table 5>은 제안 알고리즘의 유연성을 확인하기 위한 두 번째 실험의 결과를 정리한 표이다. 각 일정계획 알고리즘 사이의 성능 차이를 명확히 알아보기 위하여 식14에 따라 상대 비교지표인 Relative Percentage Deviation (이하 "RPD")을 구한다. [Figure 3, 4]는 각 실험 결과 중 상대 비교지표인 RPD를 시각적으로 비교하기 위한 막대 그래프다.



[Figure 3] Comparing scheduling algorithms by RPD values:
Ours trained and tested for each problem size



[Figure 4] Comparing scheduling algorithms by RPD values: Ours trained in problem sizes of 50 jobs for each number of machine

$$RPD(\%) = \frac{Result_{Algorithm} - Result_{CPLEX}}{Result_{CPLEX}} * 100\% \qquad (14)$$

첫 번째 실험의 결과를 통해 제안 알고리즘을 48시간 동안 훈련한 후 도출시키는 해가 CPLEX의 해와 최대 2% 이내로 근접할 정도로 비교군 알고리즘에 비해 좋은 해를 보임으로써 제안 알고리즘의 높은 성능을 확인할 수 있다. 두 번째 실험에서는 n = 100, m = 10 문제 크기에서 MA보다 좋지 않은 해를 보였으나, 우선순위 규칙보다 2~4배 좋은 일정계획을 MA보다 빠르게 수립하였다는 점과 다른 문제 크기에서는 MA보다 좋은 해를 도출한 점에서 제안하는 정책 네트워크가 가지는 Transformer 구조의 유연성이 훈련에 대한 부담을 줄일 수 있을 정도로 유의함을 확인할 수 있다.

6. 결 론

본 연구에서는 작업을 분리할 수 없고 작업 준비 시간과 설비별 가능한 작업 제약, 설비의 고장 또는 예방정비가 없는 이종 병렬설비 환경에서 총납기지연을 최소화하기 위한 일정계획 문제를 해결하기 위하여 강화학습 기반 일정계획 알고리즘을 제안하였다.

제안 알고리즘의 성능을 실험하기 위하여 작업 수에 대한 두 가지 경우의 수와 설비 수에 대한 두 가지 경우의 수를 두어 문제의 크기에 대한 총 네 가지 경우의 수를 설정하였으며, 각 문제의 크기마다 열 개의 실험 데이터세트를 임의 생성하였다. 이 실험 데이터세트에서 일정계획을 수립하여 발생하는 총납기지연으로써 성능을 비교하기 위해 수리모형을 수립[3]하고 비교군 알고리즘으로 유선순위 규칙 기반 휴리스틱 일정계획 알고리즘[18]과 미미틱 알고리즘 기반 메타휴리스틱 일정계획 알고리즘[11]을 선정하였다.

제안 모델은 강화학습을 적용하기 위하여 행렬 형식의 조합 최적화 문제 데이터를 각 행과 각 열, 원본 문제 데이터 사이의 교차적인 정보를 얻기 위하여 MatNet 구조[9]를 정책 네트워크에 적용하였으며, 그 네트워크를 훈련하는 강화학습 기반 알고리즘으로 REINFORCE with Baseline 알고리즘을 응용하여 다량의 문제 풀이 결과의 평균을 기준으로 사용함으로써 상태 평가 함수의 분산을 줄이는 POMO 알고리즘[8]을 채택하였다. 제안하는 알고리즘은 비교군으로 선정된 알고리즘보다 총납기지연이 적게 발생하였으며, CPLEX보다 빠른 속도로 그 성능보다 RPD 기준 10% 이내의 성능 차를 보여줌으로써 현업에서 사용하기 적합한 성능을 보이는 알고리즘임을 알 수 있었다. 또한, 정책 네트워크가 Transformer 구조에 기반하였기 때문에 같은 설비 수를 가지는 문제에 대하여 적은 수의 작업이 주어진 문제 크기에서 훈련된 정책 네트워크가 많은 수의 작업이 주어진 문제 크기에서도 좋은 성능을 보이는 실험을 통해 제안하는 알고리즘에서 구축한 정책 네트워크도 유연성을 갖추고 있음을 확인하였다.

그러나 본 연구는 작업 준비 시간이 없는 이종 병렬설비 환경에서 총납기지연을 최소화함을 목적으로 설정하였기 때문에 작업 준비 시간이 존재하거나 설비마다 가능한 작업 제약이 존재하는 문제 환경에는 적용할 수 없다. 또한 제안하는 알고리즘이 수행하는 연산 구조가 행렬을 다루는 연산 구조이기 때문에 실험 도구로 사용된 워크스테이션의 사양 한계로 본 연구보다 큰 문제에서의 성능을 확인할 수 없었다는 한계가 존재한다. 따라서 다양한 문제 상황에 적용할 수 있는 데이터 전처리 방법과 정책 네트워크 구조를 경량화하는 방법에 관한 추가 연구가 필요하다.

7. References

[1] S. Balin(2011a), “Non-identical parallel machine scheduling using genetic algorithm.” Expert Systems with Applications, 38(6):6814-6821.

[2] S. Balin(2011b), “Parallel machine scheduling with fuzzy processing times using a robust genetic algorithm and simulation.” Information Sciences, 181(17):3551-3569.

[3] H. G. De-Alba, et al.(2022), “A mixed integer formulation and an efficient metaheuristic for the unrelated parallel machine scheduling problem: Total tardiness minimization.” EURO Journal on

Computational Optimization, 10:100034.

[4] J. H. Holland(1975), Adaptation in natural and artificial system. Cambridge: MIT Press.

[5] S. I. Kim, et al.(2007), "Scheduling algorithms for parallel machines with sequence-dependent set-up and distinct ready times: Minimizing total tardiness." Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 221(6):1087-1096.

[6] Y. G. Kim(2017), Metaheuristics. Chonnam National University Press.

[7] W. Kool, et al.(2019), "Attention, learn to solve routing problems!." International Conference on Learning Representations, 2019:1-25.

[8] Y. D. Kwon, et al.(2020), "POMO: Policy optimization with multiple optima for reinforcement learning." Advances in Neural Information Processing Systems, 33:21188-21198.

[9] Y. D. Kwon, et al.(2021), "Matrix encoding networks for neural combinatorial optimization." Advances in Neural Information Processing Systems, 34:5138-5149.

[10] D. H. Lee, et al.(2023), "Deep reinforcement learning-based scheduler on parallel dedicated machine scheduling problem towards minimizing total tardiness." Sustainability, 15(4):2920.

[11] T. H. Lee, W. S. Yoo(2023), "A study on memetic algorithm-based scheduling for minimizing makespan in unrelated parallel machines without setup time." Journal of the Korea Safety Management & Science, 25(2):1-8.

[12] Y. H. Lee, M. Pinedo(1997), "Scheduling jobs on parallel machines with sequence-dependent setup times." European Journal of Operational Research, 100(3):464-474.

[13] Y. K. Lin, F. Y. Hsieh(2014), "Unrelated parallel machine scheduling with setup times and ready times." International Journal of Production Research, 52(4):1200-1214.

[14] T. E. Morton, R. M. V. Rachamadugu(1981), "Myopic heuristics for the single machine weighted tardiness problem." Robotics Institute, Carnegie Mellon University, Technical Report, CMU-RI-TR-83-09.

[15] P. Moscato(1989), "On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms." Caltech Concurrent Computation Program 158-79, C3P Report, 826:37.

[16] S. U. Randhawa, T. A. Smith(1995), "An experimental investigation of scheduling non-identical, parallel processors with sequence-dependent set-up times and due dates." The International Journal of Production Research, 33(1):59-69.

[17] A. Vaswani, et al.(2017), "Attention is all you need." Advances in Neural Information Processing Systems, 30.

[18] A. P. J. Vepsalainen, T. E. Morton(1987), "Priority rules for job shops with weighted tardiness costs." Management Science, 33(8):1035-1047.

[19] R. J. Williams(1992), "Simple statistical gradient-following algorithms for connectionist reinforcement learning." Machine Learning, 8(3):229-256.

저자 소개

이 태 희

인천대학교 산업경영공학과에서 학사 취득. 현재 인천대학교 일반대학원 산업경영공학과에서 석사과정 이수 중

관심분야 : 스마트팩토리, 산업인공지능 등

김 재 곤

KAIST 산업공학과에서 학사, 석사, 박사학위를 취득, 현재 인천대학교 산업경영공학과 교수로 재직 중

관심분야: 산업인공지능, 데이터 기반 운영 최적화

유 우 식

서울대학교 산업공학과와 과학기술원 산업공학과에서 석사, 박사를 취득, 현재 인천대학교 산업경영공학과에서 교수로 재직 중

관심분야 : 스마트 팩토리, CAD/CAM, 제조시스템공학