

강화학습을 활용한 작업 분할이 가능하고 설비 제약이 존재하는 동종 병렬 시스템의 총 납기 지연 최소화

이성태 · 유우식[†]

인천대학교 산업경영공학과

Minimizing Total Tardiness for Identical Parallel Machine Scheduling Problem with Splitting Jobs and Machine Eligibility Constraint

Sungtae Lee · Woosik Yoo

Industrial and Management Engineering, Incheon National University

Machine scheduling problems are one of the important issues in manufacturing systems. In particular, in complex conditions such as job split and machine eligibility constraint, it becomes challenging to obtain appropriate solutions and it has been proven as a NP-hard problem. Previously, heuristic and metaheuristic techniques were mainly used to solve machine scheduling problems. Recently, applying reinforcement learning to scheduling problems to overcome the limitations of the two techniques has been receiving attention. In this study, we propose a methodology to minimize total tardiness by using pointer network-based reinforcement learning. Job sequences are determined through reinforcement learning. Machine selection and job split is determined with heuristic methods to reduce the solution space while effectively minimizing total tardiness. The methodology of this study showed higher performance than COVERT, ATCS, GA, and it is expected to be more versatile than existing reinforcement learning methods because it does not require re-learning according to the changes in task lengths.

Keywords: Scheduling, Reinforcement Learning, Job Split, Machine Eligibility Constraint, Minimizing Total Tardiness, Machine Learning

1. 서론

최근 고객의 구매력이 증가하고 제품의 수명 주기가 짧아짐에 따라(Kim *et al.*, 2020) 제조시스템이 복잡해졌고, 납기일을 준수하는 것이 더욱 중요해지고 있는데 반해 효과적인 일정계획 수립의 어려움이 커지고 있다(Demirkv *et al.*, 2010). 비효율적인 일정계획은 생산성을 감소시켜 제조시스템 전체에 악영향을 끼친다. 반대로 일정계획의 개선을 통해 추가적인 장비의 도입 없이 막대한 비용을 절감시킬 수 있다. 또한 효과적인 일

정계획을 통해 총 납기 지연을 최소화하는 것은 기업의 경쟁력을 높일 수 있는 요소이다(Lee, 2000).

본 연구는 인쇄 회로기판의 천공 공정 등에서 찾아볼 수 있는 동종병렬설비에서 작업 분할이 가능하고 설비제약이 존재할 때 총 납기지연을 최소화하는 일정계획 생성을 목표로 한다. 각 작업은 정수 단위로 분할하여 복수의 설비에서 동시에 작업할 수 있으며 이때 이동시간은 고려하지 않는다. 작업물 전용 설비 제약이 있어 작업은 종류별로 할당할 수 있는 설비가 사전에 정의된다. 기존에 할당되어 있는 작업과 새로 할당

본 연구는 인천대학교 2022년도 자체연구비 지원에 의하여 연구되었음.

[†] 연락저자 : 유우식 교수, 22012 인천 연수구 아카데미로 119 인천대학교 산업경영공학과, Tel : 032-835-8488, Fax: 032-835-0777,

E-mail: wsyoo@inu.ac.kr

2023년 8월 2일 접수; 2023년 9월 10일 수정본 접수; 2023년 10월 4일 게재 확정.

하는 작업의 종류가 다를 경우 셋업이 발생하는 작업 순서의 존재적인 셋업이 존재한다. 작업이 납기일 이후에 처리되면 납기지연이 발생한다.

납기지연을 최소화하기 위해서는 설비 선택 및 작업 분할을 적절하게 하는 것이 중요하다. 예를 들어 각 작업을 처리할 때 납기지연을 최소화하기 위해 탐욕적으로 작업을 분할하면 셋업 횟수가 증가하여 이후 할당될 작업에서 납기지연이 증가하게 되고, 반대로 셋업 횟수를 줄이기 위해 작업 분할을 하지 않을 경우 작업이 처음 할당된 설비에서 장시간 동안 처리되어 작업물 전용 설비 제약에 의해 해당 설비에 할당할 수 있는 작업들의 납기지연이 증가한다.

셋업 시간이 작업 순서에 의존적이면 납기지연 최소화 문제의 복잡도가 매우 커진다(Kim and Bobrowski, 1994). 또한, 작업의 분할이 가능하면 총 의사결정수가 많아져 문제의 복잡도는 더욱 커진다.

이와 같이 의사결정의 길이가 길고 문제가 복잡한 경우 최적해를 구하는데 긴 시간이 소요되고, 강화학습을 통한 학습 또한 어려워진다. 이를 해결하기 위해 본 연구에서는 강화학습과 휴리스틱 기법을 결합하여 효율적으로 의사결정 수를 줄이면서 총 납기지연을 최소화하는 방법론을 제안한다.

작업의 처리 순서는 포인터 네트워크 기반의 강화학습 네트워크를 통해 결정한다. 트랜스포머 모델과 같이 멀티 헤드 어텐션 구조를 이용하여 각 주문에 대한 query, key, value를 생성하고, 가장 최근에 할당된 작업의 query와 남은 작업의 key 간의 유사도를 비교해 유사도가 가장 높은 작업을 다음에 할당하여 모든 작업의 처리 순서를 결정한다. 작업을 처리할 설비와 작업 분할은 Heuristic rule을 통해 결정한다. 본 연구에서 개발한 모델을 휴리스틱 모델인 COVERT(cost over time), ATCS(apparent tardiness cost with setup)과, GA(Genetic Algorithm)의 결과와 비교하였다.

본 연구는 다음과 같이 구성된다. 제2장에서는 기존 동종병렬설비 문제를 풀기 위해 사용된 방법론들을 소개한다. 제3장에서는 제안 기법의 구조를 설명한다. 제4장에서는 여러 환경에서 제안 방법론의 성능을 분석한다. 마지막으로 제5장에서는 본 연구에 따른 결론과 향후 연구 방향을 제시한다.

2. 기존 연구

기존 현업에서는 동종병렬설비에서 납기지연을 최소화하기 위해 휴리스틱 방법과 유전 알고리즘(Genetic Algorithm)과 같은 메타 휴리스틱 방법을 주로 사용했다. 하지만 이러한 방법들은 제조시스템에 적용하여 효율적인 일정계획을 하기에 애로사항이 존재한다. 휴리스틱 기법은 결과를 빠른 시간 내에 낼 수 있지만 좋은 해를 보장하지 못한다. 메타 휴리스틱 기법은 휴리스틱 기법에 비해 좋은 해를 얻을 수 있지만 탐색 시간이 오래 걸려 빠른 의사결정이 중요시되는 실제 제조시스템에 적용하기 어렵다(Ryu et al., 2000).

이러한 애로사항을 개선하기 위해 딥러닝을 일정계획 수립에 적용하는 연구들이 진행되었다. 그중에서도 강화학습을 활용한 연구가 활발하게 이뤄지고 있다. Paeng et al.(2021)은 병렬설비에서 작업 및 설비에 대한 상태를 입력으로 받아 설비와 작업을 선택해 총 납기지연을 최소화하는 DQN 기반의 강화학습 모델을 제안하였다. Liu et al.(2023)은 작업이 동적으로 도착하는 병렬설비에서 특정 시점에 적합한 휴리스틱 기법을 골라 그에 따라 작업을 할당하는 강화학습 기법을 제안하였다. 두 연구 모두 실험 결과 휴리스틱 모델보다 더 나은 결과를 보였다. 하지만 출력의 크기가 정해진 DQN은 모든 상황에 대해 행동의 갯수만큼의 후보만 할당을 고려할 수 있어 깊은 탐색에 어려움이 존재한다. 또한, 입력 차원의 크기가 고정되어야 하므로 학습의 용이성을 위해 추상화를 하여 상태를 정의하는데, 이 과정에서 정보가 손실된다.

이러한 단점을 극복하기 위해 최근 조합 최적화 문제에서 다양한 시도가 이뤄지고 있다. Kool et al.(2018)은 외판원 문제에서 점들의 좌표를 입력으로 받아 방문 순서를 출력하는 방법론을 제안하였다. 포인터 네트워크를 사용하여 방문할 점들의 순서를 출력함으로써 특정 시점에 방문하지 않은 모든 점을 고려할 수 있고 모든 점의 좌표를 입력으로 받기 때문에 정보가 손실되지 않았다. 실험 결과 방문할 지점이 100개 이내인 문제에서 최적해에 매우 근접한 출력을 얻었다. Ireland et al.(2022)은 대규모의 벡터 커버 문제에서 인공지능망을 사용해 하나의 그래프를 여러 개의 하위 그래프로 분할하고 각각의 하위 그래프는 휴리스틱 기법으로 처리하는 방법론을 제안하였다. Vienna 데이터와 같이 노드의 수가 매우 많은 환경에서 효과적으로 해공간을 줄이기 위해 성능이 좋은 휴리스틱 기법을 사용하면서 하위 그래프는 강화학습 네트워크를 통해 결정하여 깊은 탐색을 진행하였고, 실험 결과 최적해에 가까운 솔루션을 도출했다.

본 연구에서는 일정계획 성능에 큰 영향을 미치는 작업의 순서는 포인터 네트워크 기반의 강화학습 모델을 통해 결정하고, 경우의 수가 많아 학습은 어렵지만 좋은 방법론이 존재하는 작업 분할은 휴리스틱으로 결정하는 일정계획 방법론을 제안한다.

3. 제안 기법

3.1 배경 이론

(1) 포인터 네트워크

포인터 네트워크는 Vinyals et al.(2015)에 의해 제안된 구조로 어텐션 메커니즘을 응용하여 만들었다. 이는 입력과 출력의 대상이 같은 외판원 문제, 스케줄링과 같이 조합 최적화에 최적화된 기법이다. 기존에 어텐션 메커니즘이 사용된 모델의 경우 입력할 단어를 번역하여 출력하기 위해 정해진 단어 사전을 기반으로 학습을 진행하기 때문에 새로운 단어가 추가될 시 다시 학습을 진행

해야 한다. 즉, 출력의 차원이 사전에 정의되어야 한다. 이는 문제에 따라 출력이 크게 달라지는 조합 최적화 문제에 사용하기에 부적합하다. 포인터 네트워크는 입력 데이터의 인덱스로 이뤄진 순서를 출력하는 모델로, 입력값의 길이에 따라 가변적인 길이의 시퀀스를 출력할 수 있다는 장점이 있다.

인코더와 디코더의 은닉 상태를 각각 $(e_1, e_2, \dots, e_n), (d_1, d_2, \dots, d_m)$ 라고 정의하고, V, W_1, W_2 는 학습 가능한 가중치일 때, 포인터 네트워크는 입력값들의 i 시점의 인코딩 벡터 u_j^i 는 식 (1)과 같이 계산되며 u_j^i 의 다음 출력에 대한 조건부확률을 식 (2)와 같이 계산하여 가변적인 출력을 얻을 수 있다(Bahdanau *et al.*, 2015).

$$u_j^i = V^T \tanh(W_1 e_j + W_2 d_i) \quad (1)$$

$j \in \{1, 2, \dots, n\}$

$$p(C_i | C_1, C_2, \dots, C_{i-1}, P) = \text{softmax}(u^i) \quad (2)$$

(2) POMO 알고리즘

일반적으로 REINFORCE 알고리즘(Sutton *et al.*, 1999)은 학습 과정에서 분산이 매우 커 학습이 불안하다. 분산의 크기는 성능에 영향을 미치므로 이를 줄이는 것은 매우 중요하다(Van *et al.*, 2016; Fujimoto *et al.*, 2018).

REINFORCE 알고리즘에서는 각 보상에서 상태에 대한 가치인 baseline을 빼서 분산을 줄인다. Baseline은 주로 같은 문제를 여러 번 풀어 얻은 결과의 평균을 사용한다. 이 경우, 지역 최적해에 빠지기 쉽다는 문제가 발생한다. POMO 알고리즘은 하나의 솔루션을 다양하게 표현할 수 있는 조합 최적화 문제의 특징을 이용하여 해당 문제를 해결한다. 예를 들어 모든 점을 방문하고 시작 지점으로 돌아와야 하는 TSP 문제에서는 시작 지점에 따라 동일한 솔루션을 다양하게 표현할 수 있다. 방문 지점이 4개인 경우, $(a_1, a_2, a_3, a_4, a_1), (a_2, a_3, a_4, a_1, a_2), (a_3, a_4, a_1, a_2, a_3)$ 은 모두 동일한 솔루션이다. POMO 알고리즘은 위와 같이 다양한 출력 시퀀스를 얻기 위해 하나의 문제에 대해 처음 입력을 <START> 토큰을 넣지 않고 시작점을 임의의 여러 점으로 지정하여 넣는다. 이때, 각 시작점에 따라 도출된 거리를 보상으로 사용하고, 거리들의 평균을 baseline으로 사용하여 가치함수를 업데이트한다. 이를 통해 같은 문제를 다각화하여 여러 번 풀어 지역 최적해 문제를 해결하고 분산을 줄인다.

본 연구에서는 하나의 문제를 작업 수 n 번만큼 반복하여 풀고 각 작업을 처음으로 처리했을 때 발생한 총 납기지연의 평균을 baseline으로 사용하여 학습을 진행했다.

3.2 문제 정의

본 연구는 작업의 종류별로 설비 할당 제약이 있으며 분할이 가능한 n 개의 작업을 m 개의 동종 병렬설비에 할당하는 문

제를 다루며 문제정의에 필요한 변수는 <Table 1>과 같다. 목적함수는 모든 작업의 총 납기지연인 $\sum_{j=1}^n \tau_j$ 의 최소화다. 각 작업은 J_j 로 정의된다. 각 작업의 크기 Q_j 는 1~10 사이의 정수로 정의된다. 작업의 처리 시간은 크기와 동일하고, Q_j 는 균등 분포로 생성된다.

Table 1. Indices and Parameters of Problem

Element	Description
$J_j \in \{1, 2, \dots, n\}$	Indices for jobs
$Q_j \in \{1, 2, \dots, 10\}$	Size of job j
$e_j \in \{A, B, C\}$	Type of job j
$M_k \in \{1, 2, \dots, m\}$	Indices for machine
$s_{a,b} \in \{1, 2, \dots, 6\}$	Setup time parts of type b just after those of type a
τ_j	Tardiness of job j
d_j	Due date of job j
x_{jk}	Split job j processed by machine k
c_{jk}	Processing time of job jk
τ_{jk}	Tardiness of job jk

작업의 종류는 e_j 로 정의된다. e_j 는 A, B, C 세 개 중 하나로 임의로 결정되며 <Table 2>와 같이 작업의 종류에 따라 할당할 수 있는 설비가 결정되어있다. 설비 내에서 작업의 종류가 a 에서 b 로 발생하는 셋업 $s_{a,b}$ 는 1~6 사이의 정수로 정의되며 균등 분포로 생성되며 문제에 따라 다르게 생성된다.

Table 2. Allocatable Machines by Job Type

m	Job Type	Allocatable Machines
5	A	1, 2, 3
	B	2, 3, 4
	C	3, 4, 5
10	A	1, 2, 3, 4, 5
	B	4, 5, 6, 7, 8
	C	6, 7, 8, 9, 10

작업은 정수단위로 분할하여 복수의 설비에서 처리할 수 있으며, 이때 분할된 작업을 다른 설비에 옮기는데 걸리는 시간은 고려하지 않는다. 분할된 작업은 x_{jk} 로 정의되며 분할된 작업이 각 설비에 할당될 때 완료시간은 c_{jk} 로 정의된다. x_{jk} 의 납기지연 τ_{jk} 는 $\max(c_{jk} - d_j, 0)$ 로 계산되며 작업 J_j 의 납기지연 τ_j 는 τ_{jk} 의 합으로 정의된다.

3.3 제안 프레임워크

본 연구에서 다룬 문제는 작업 순서 결정, 할당할 설비 결정,

작업 분할 결정 세 과정에 대한 의사결정이 요구되며, 작업의 수가 상대적으로 적은 20개인 경우에도 CPLEX를 통해 이산적인 시간 내에 최적해를 구할 수 없는 매우 복잡한 문제다.

이에 본 연구에서는 해공간을 줄이기 위해 설비 및 작업 분할 결정과 작업 순서 결정을 구분하였다. 설비 및 작업 분할 결정은 heuristic rule로도 좋은 솔루션을 도출할 수 있기 때문에 rule을 적용하여 처리한다. 반대로 휴리스틱 규칙의 성능이 제한적이지만 중요한 의사결정인 작업 순서 결정은 REINFORCE 알고리즘을 통해 학습하는 PNRL(Pointer Network Reinforcement Learning)을 제안한다.

PNRL의 전체 프레임워크는 <Figure 1>과 같다. 포인터 네트워크를 기반으로 전체작업의 처리 순서를 결정하는 policy network와 결정된 작업 순서에 따라 작업을 할당할 설비와 작업 분할을 결정하는 휴리스틱 디코더로 구성된다. 네트워크 구조는 kwon *et al.*(2020)이 제안한 POMO(Policy optimization with multiple optima) 알고리즘을 사용하여 학습하며 사용한 하이퍼파라미터는 <Table 3>과 같다.

Table 3. Model Hyperparameters

Hyperparameters	Value
Input embedding dim	128
encoder layer num(N)	6
head num	8
Feed forward layer dim	512
Leaning rate for Adam	0.0001

네트워크는 모든 작업에 대한 정보를 입력으로 받아 인코더에서 어텐션 네트워크를 통과시켜 각 작업의 정보 및 작업간

의 유사도를 고려한 고차원의 벡터로 인코딩한다. 이때 작업에 대한 정보는 <Table 4>와 같다. 디코더에서는 가장 최근에 할당된 작업에 대한 query와 다른 작업들의 key, value를 어텐션하여 나온 어텐션 스코어를 기반으로 다음 순서에 가장 적합한 작업을 선택한다. 이때, 이미 선택된 작업은 masking하여 선택되지 않도록 한다. 이 과정을 모든 작업이 선택될 때까지 반복하여 작업을 처리할 순서를 결정한다.

모델은 모든 작업에 대한 정보를 입력으로 받아 각 작업 $J_j(j = 1, 2, \dots, n)$ 을 첫 번째로 처리했을 때 나머지 작업을 처리할 순서를 출력한다. 작업의 처리 순서가 결정되면 휴리스틱 디코더의 규칙에 따라 설비 및 작업의 분할을 결정해 작업을 할당하고 총 납기 지연을 계산한다. 이때 발생한 n 개의 total tardiness를 reward로 받아 네트워크를 업데이트한다. 작업 j 를 처음으로 처리할 때 발생한 총 납기 지연을 r_j 라 할 때 baseline $b(s)$ 의 계산은 식 (3)과 같으며, 네트워크 업데이트 방법은 식 (4)와 같다.

$$b(s) = \frac{1}{n} \sum_{j=1}^n r_j \quad (3)$$

$$\nabla_{\theta} J(\theta) \approx \frac{1}{n} \sum_{i=1}^n (r_j - b(s)) \nabla_{\theta} \log p_{\theta}(r_j | s) \quad (4)$$

휴리스틱 디코더는 kim *et al.*(2020)이 제안한 방법을 사용하였으며 강화학습 네트워크를 통해 결정된 작업이 할당될 설비와 분할 지점을 결정한다. 작업이 할당될 설비는 해당 작업을 할당할 수 있는 모든 설비 중에서 작업 할당 시 완료 시간이 가장 빠른 설비로 결정된다. 설비에 작업을 할당할 때, 모든 작업이 마감일 이전에 완료될 시 작업 분할을 진행하지 않는다. 작

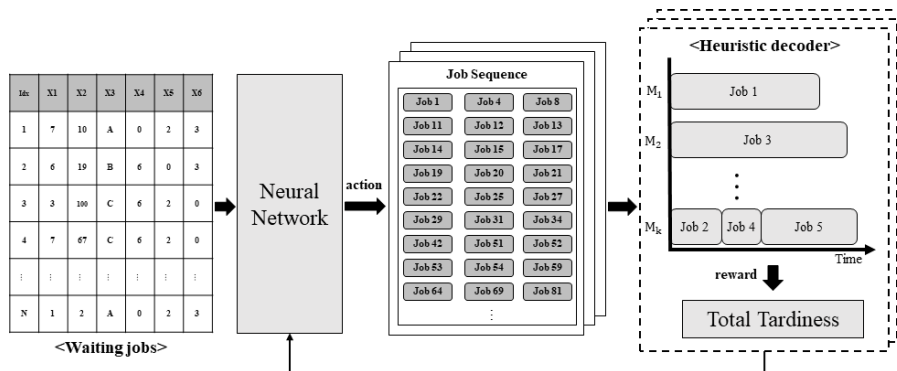


Figure 1. Framework of Pointer Network Reinforcement Learning

Table 4. State Definition

Element	Description	Type	Num of features
$Q_j \in \{1, 2, \dots, n\}$	Size of job j	Integer	1
$e_j \in \{A, B, C\}$	Type of job j	One-hot	3
$s_{k,e_j}, k \in \{A, B, C\}$	Setup time parts of type e_j just after those of all types	Matrix	3
d_j	Due date of job j	Integer	1

업 처리 도중 마감일을 넘기게 되면 마감일까지 해당 작업을 처리하고 남은 작업을 분할하여 다른 설비에 할당한다. 이때, 다음에 할당될 설비는 기존과 같이 할당 시 완료 시간이 가장 빠른 설비다. 만약 작업이 마감일 이후에 시작될 경우 작업의 최소 단위인 1만큼만 해당 설비에 할당하고 남은 작업을 분할하여 다른 설비에 할당한다. 이 방법은 후속 작업의 납기 지연을 고려하지 않고 탐욕적으로 지연을 줄이기 위해 작업을 분할하는 방법이다. 이에 따라 한번 납기일을 넘기기 시작하면 전체 납기 지연이 폭발적으로 증가하는 단점이 있다. 하지만, 납기일이 여유가 있게 분포되어 있다면 최적에 가까운 의사결정을 할 수 있으며, 구조가 간단하여 네트워크가 할당-분할의 규칙을 이해하기 쉬워 학습이 빠르다는 장점이 있다.

4. 실험 및 결과 분석

4.1 실험 환경

본 연구에서는 설비의 수가 5개, 10개인 제조시스템 환경에서 총 작업의 수와 납기일 분포의 긴급 정도를 조절하여 12개의 조건에 대해 모델의 학습 및 성능 평가를 진행했다. 조건별 파라미터는 <Table 5>와 같다.

Table 5. Data Parameters

Equipment	Job Size	α
5	20	0.3(Loose)
5	20	0.6(Tight)
5	50	0.3(Loose)
5	50	0.6(Tight)
5	100	0.3(Loose)
5	100	0.6(Tight)
10	20	0.3(Loose)
10	20	0.6(Tight)
10	50	0.3(Loose)
10	50	0.6(Tight)
10	100	0.3(Loose)
10	100	0.6(Tight)

모델의 학습은 각 조건에 대해 500,000개의 데이터를 생성하여 설비의 수와 전체작업의 길이(Job size)가 같은 데이터를 묶어 총 1,000,000개의 데이터로 여섯 개의 모델을 학습하였다. 테스트 세트는 각 문제 조건별로 10개의 데이터를 만들어 다른 방법론과 결과를 비교하였다. 비교군으로는 모델의 성능을 평가하기 위해 납기 지연 시간을 최소화하는데 효과적이라고 알려진 휴리스틱 기법인 ATCS(Lee et al., 1997), COVERT(Carroll, 1965)와, 메타 휴리스틱 기법 GA를 사용하였다. COVERT는 작업을 할당할 때 발생하는 비용대비 잠재적인 이득을 고려하여 우선순위에 따라 작업을 할당하는 방법으로 단일 및 병렬기계 환경에서 주로 쓰인다. ATCS는 추정 완료시간

및 셋업 시간의 상관계수를 기반으로 작업의 처리 순서를 결정하는 방법이다. 일반적으로 COVERT와 ATCS 모두 병렬기계 환경에서 총 납기지연을 최소화하는데 효과적이라고 알려진 휴리스틱 기법이다. GA는 Kim et al.(2020)이 제안한 기법을 사용했다. 해당 연구에서는 50세대 동안 개선이 되지 않으면 탐색을 종료했지만, 본 연구에서는 충분한 탐색을 고려하기 위해 5,000세대 동안 개선이 되지 않거나 탐색 시간이 1시간을 넘기면 탐색을 종료하여 결과를 비교하였다. 문제의 특성상, 작업 수가 20개인 경우에 대해서도 최적해를 구할 수 없어 다른 탐색 방법과의 결과만 비교했다. 모든 실험은 32GB RAM, 11th Gen Intel(R) Core(TM) i9-11900K CPU, RTX 3080 GPU가 장착된 PC에서 진행되었으며 문제의 조건별로 10개의 데이터를 생성하여 결과를 비교했다.

4.2 납기 생성 방법

작업의 납기일은 d_j 로 Lee and Pinedo(1997)가 제안한 방법에 따라 생성하였으며, 납기일의 분포는 <Figure 2>와 같으며 사용된 변수에 대한 설명은 <Table 6>과 같다.

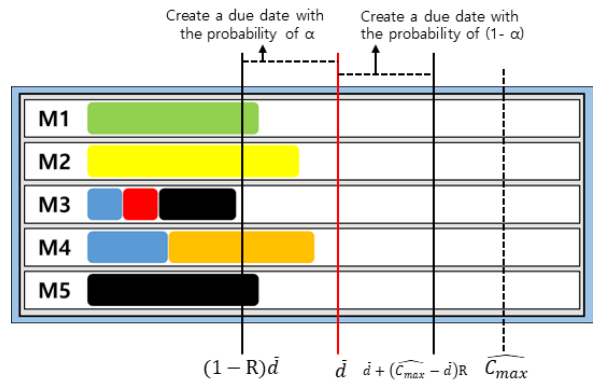


Figure 2. Due Date Generation Rule

Table 6. Parameters of Due Date Generation Rule

Element	Description
\bar{Q}	Average of all jobs size
\widehat{C}_{max}	Estimated completion time
$\alpha \in \{0.3, 0.6\}$	due date distribution of problem set
\bar{d}	Average of due date
$R = 0.5$	Due date range factor

모든 작업의 크기 평균인 \bar{Q} 에 총 작업수 n 과 총 설비수 m 을 통해 전체 작업 완료 예정시간 $\widehat{C}_{max} = \bar{Q} * n/m$ 를 정의한다. 납기일 분포 변수 α 와 \widehat{C}_{max} 를 사용하여 납기일 생성 기준 $\bar{d} = (1-\alpha) * \widehat{C}_{max}$ 로 정의한다. 본 연구에서는 $\alpha \in \{0.3, 0.6\}$ 두 환경에 대해 실험을 진행하였는데, $\alpha = 0.3$ 인 경우 납기일에 여유가 있고, $\alpha = 0.6$ 인 경우 납기일이 촉박하다.

납기일 범위 변수 R 은 0.5로 고정하여 실험하였고, 납기일은 α 의 확률로 $[(1-R)\bar{d}, \bar{d}]$ 구간의 정수로 임의 생성되고, $(1-\alpha)$ 의 확률로 $[\bar{d}, \bar{d}+(\widehat{C}_{\max}-\bar{d})R]$ 구간의 임의로 생성된다.

4.3 실험 결과

<Table 7>과 <Table 8>은 각 방법론의 총 납기 지연과 처리 시간의 평균을 나타낸 표다. PNRL의 일반화 성능을 확인하기 위해 각 작업 수 $S=\{20, 50, 100\}$ 에 대해 훈련한 모델과 작업 수가 50개인 데이터로만 학습한 모델의 성능을 비교하였다. 휴리스틱 모델은 처리 시간이 가장 빠르지만 다른 방법론 대비 총 납기지연이 매우 컸다. GA 모델은 총 납기지연이 휴리

스틱 모델 대비 낮은 모습을 보였지만 처리시간이 매우 길며, 특히 작업 수가 100개인 경우 시간 내에 탐색을 끝내지 못하는 모습을 보였다. PNRL 모델로 각 작업 수에 대해 학습을 진행했을 때 총 납기지연이 가장 낮은 모습을 보였다. 또한, 각 작업 수에 대해 학습한 모델이 가장 좋은 성능을 보였지만, 작업 수가 50개인 데이터로만 학습한 모델도 하나의 조건을 제외하고 다른 방법론보다 더 좋은 성능을 보였고, 이를 통해 모델의 일반화 성능도 좋음을 확인하였다. 처리시간은 휴리스틱 모델 대비 길지만, 최대 2.412초로 합리적인 시간 내에 의사결정이 가능하다. 다만, 다른 방법론과 다르게 훈련 시간이 존재한다.

본 연구에서 사용한 test set과 같이 문제의 수가 적은 경우에는 heuristic이나 meta-heuristic이 더욱 적합하지만 적절한 시간

Table 7. Total Tardiness for COVERT, ATCS, GA, PNRL

Equipment Size	Job Size	α	Heuristic		Meta Heuristic	PNRL	
			COVERT	ATCS	GA	Trained by 50	Trained by S
5	20	0.3(Loose)	45.9	36.9	2.3	6.4	0.8
		0.6(Tight)	149	141.5	90.9	85.2	81.5
	50	0.3(Loose)	209.5	97.3	19.6	10.1	10.1
		0.6(Tight)	868.2	751.3	539.5	496.2	496.2
	100	0.3(Loose)	844.1	236.3	39.4	10.1	6.7
		0.6(Tight)	2,857.5	2312.1	1627.2	1555	1482
10	20	0.3(Loose)	1.1	0.3	0.3	0.3	0.3
		0.6(Tight)	62.8	47.6	39.8	30.9	28.7
	50	0.3(Loose)	95.7	8.6	5.7	0.9	0.9
		0.6(Tight)	410.6	328	295.5	207.8	207.8
	100	0.3(Loose)	370.6	34.9	20.3	11.8	6.5
		0.6(Tight)	1,442.8	1,129.1	1,032.5	901.7	900.5

Table 8. Processing Time for COVERT, ATCS, GA, PNRL(sec)

Equipment Size	Job Size	α	Heuristic		Meta Heuristic	PNRL		
			COVERT	ATCS	GA	Trained by 50	Trained by S	Training time
5	20	0.3(Loose)	0.002	0.001	109.442	0.71	0.075	7.18h
		0.6(Tight)	0.004	0.001	503.063	0.069	0.070	
	50	0.3(Loose)	0.008	0.005	1295.613	0.481	0.481	42.36h
		0.6(Tight)	0.010	0.006	3219.559	0.512	0.499	
	100	0.3(Loose)	0.029	0.019	3600	1.921	1.977	166.17h
		0.6(Tight)	0.029	0.021	3600	2.412	2.356	
10	20	0.3(Loose)	0.001	0.001	9.763	0.165	0.160	7.72h
		0.6(Tight)	0.001	0.001	200.898	0.167	0.168	
	50	0.3(Loose)	0.002	0.003	395.291	0.496	0.488	46.56h
		0.6(Tight)	0.003	0.003	466.169	0.517	0.522	
	100	0.3(Loose)	0.007	0.008	3600	1.899	1.956	179.22h
		0.6(Tight)	0.007	0.008	3600	2.172	2.056	

내에 결과를 내야 하며, 지속적으로 양질의 솔루션이 요구되는 실제 공정과 같은 상황에서는 학습 시간을 고려하더라도 PNRL을 적용하는 것이 장기적으로 적합할 것으로 기대된다.

5. 결론

본 연구에서는 작업의 분할이 가능한 동종병렬설비에서 총 납기지연을 최소화하는 일정계획 문제를 다뤘다. 문제 해결을 위해 포인터 네트워크 기반의 REINFORCE 알고리즘을 이용하였고, 작업을 처리할 설비와 작업 분할은 휴리스틱 기법을 통해 처리하는 일정계획 기법을 제안하였다.

모델의 성능을 검증하기 위해 납기 지연 문제에서 주로 쓰는 휴리스틱 기법인 COVERT, ACTS와 메타휴리스틱 모델과의 성능을 비교했다. 6가지 문제 조건에서 각 조건별로 10개의 데이터 세트를 생성하여 성능을 비교한 결과, 본 연구에서 제안한 모델이 기존 모델에 비해 더 좋은 성능을 보임을 확인할 수 있었다. 또한 대다수의 경우 훈련에 사용한 작업 길이와 검증에 사용한 작업의 길이가 달라져도 범용적으로 좋은 성능을 보임을 확인했다. 이를 통해, 해공간을 적절히 줄이면 트랜스포머 기반의 강화학습이 복잡한 문제에서도 작업 간의 관계를 분석해 기존 방법론 대비 좋은 성능을 낼 수 있음을 확인했다. 하지만 인코딩 과정에서 모든 입력 데이터를 병렬적으로 계산하는 트랜스포머 모델의 특징상, 처리해야 하는 작업의 길이가 길어지면 복잡도가 매우 커진다. 이 경우 메모리 용량이 더 큰 GPU를 사용하거나, 하나의 문제를 여러 개로 나눠서 풀어야 한다.

추후, 본 연구에서 제시한 방법론을 발전시켜 OLED 디스플레이의 EVEN 공정에서 실제 데이터를 확보하여 동적 스케줄링 연구를 진행할 예정이다.

참고문헌

- Bahdanau, D., Cho, K., and Bengio, Y. (2014), Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.
- Carroll, D. C. (1965), Heuristic sequencing of single and multiple component jobs, Massachusetts Institute of Technology.
- Demirkov, E. and R. Uzsoy(2010), Decomposition methods for reentrant flow shops with sequence-dependent setup times, *Journal of Scheduling*, 3(3), 155-177
- Fujimoto, S., Hoof, H., and Meger, D. (2018), Addressing function approximation error in actor-critic methods, In *International conference on machine learning*, PMLR, 1587-1596.
- Ireland, D. and Montana, G. (2022), Lense: Learning to navigate subgraph embeddings for large-scale combinatorial optimisation, *Conference on Machine Learning*, PMLR, 9622 -9638.
- Kim, J. G., Song, S., and Jeong, B. (2020), Minimising total tardiness for the identical parallel machine scheduling problem with splitting jobs and sequence dependent setup times, *International Journal of Production Research*, 58(6), 1628-1643.
- Kim, S. C. and Bobrowski, P. M. (1994), Impact of sequence-dependent setup time on job shop scheduling performance, *The International Journal of Production Research*, 32(7), 1503-1520.
- Kool, W., Van Hoof, H., and Welling, M. (2018), Attention, learn to solve routing problems!, arXiv preprint arXiv:1803.08475.
- Kwon, Y. D., Choo, J., Kim, B., Yoon, I., Gwon, Y., and Min, S. (2020), Pomo: Policy optimization with multiple optima for reinforcement learning, *Advances in Neural Information Processing Systems*, 33, 21188-21198.
- Lee, Y. H., Bhaskaran, K., and Pinedo, M. (1997), A heuristic to minimize the total weighted tardiness with sequence-dependent setups, *IIE Transactions*, 29(1), 45-52.
- Lee, Y. S. (2000), A study on Developing algorithms for Manufacturing networks using meta-Heuristics (Doctoral dissertation, Ph. D. Thesis, Sangmyung University).
- Liu, C. L., Tseng, C. J., and Huang, T. H. (2023), Dynamic Parallel Machine Scheduling With Deep Q-Network, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.
- Paeng, B., Park, I. B., and Park, J. (2021), Deep reinforcement learning for minimizing tardiness in parallel machine scheduling with sequence dependent family setups, *IEEE Access*, 9, 101390 -101401.
- Ryu, J. W., Kang, M. K., and Kim, M. W. (2000), A Clustering Algorithm based on Heuristic Evolution Algorithm, *Korean Institute of Information Scientists and Engineers*, 27(2), 78-80.
- Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (1999), Policy gradient methods for reinforcement learning with function approximation, *Advances in Neural Information Processing Systems*, 12.
- Van Hasselt, H., Guez, A., and Silver, D. (2016), Deep reinforcement learning with double q-learning, *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 30, No. 1).
- Vinyals, O., Fortunato, M., and Jaitly, N. (2015), Pointer networks, *Advances in Neural Information Processing Systems*, 28.

저자소개

이성태: 인천대학교 산업경영공학과에서 2022년 학사학위를 취득하고 인천대학교에서 산업경영공학과 석사과정에 재학 중이다. 연구분야는 최적화, 스마트제조이다.

유우식: 1986년 서울대학교에서 학사학위를 취득하고 한국과학기술원에서 1988년 석사, 1992년 박사학위를 취득하고 1996년부터 인천대학교 산업경영공학과에서 교수로 재직 중이다. 연구분야는 스마트제조, CAD/CAM이다.