순서 의존적 작업 준비시간을 갖는 단일기계 작업장을 위한 강화학습 기반 작업 배정 모형

박진성[O], 김준우[*]
[O]동아대학교 산업경영공학과,
[*]동아대학교 산업경영공학과
e-mail: pjs0958@donga.ac.kr[O], kjunwoo@dau.ac.kr[*]

Reinforcement Learning based Job Dispatching Model for Single Machine with Sequence Dependent Setup Time

Jin-Sung Park[O], Jun-Woo Kim[*]
[O]Dept. of Industiral and Management Systems Engineering, Dong-A University,
[*]Dept. of Industiral and Management Systems Engineering, Dong-A University

● 요 약 ●

순서 의존적 준비시간을 갖는 단일기계 생산라인에서 주어진 작업들을 효율적으로 수행하기 위해서는 최대한 동일하거나 유사한 유형의 작업물들을 연속적으로 처리하여 다음 번 작업물의 처리를 시작하기 전에 발생하는 준비시간을 최소화하여야 한다. 따라서, 대기 중인 것들 중 기계에 투입할 작업물을 적절히 선택하는 것이 중요하며, 이를 위해 작업 배정 규칙과 같은 휴리스틱을 사용할 수도 있지만, 이러한 해법들은 일반적으로 다양한 상황을 동적으로 고려하지 못하는 한계점을 갖는다. 따라서, 본 논문에서는 상용 3D 시뮬레이션 소프트웨어인 FlexSim을 사용하여 모형을 구성한 다음, 강화학습을 적용하여 대기 중인 작업물 중 최적의 후보를 선택하기 위한 작업 배정 모형을 개발하고자 한다. 세부적으로는 강화학습의 상태 및 보상을 달리 설정하면서 학습된 모형의 성능을 비교하고자 한다. 실험 결과를 통해 적절한 시뮬레이션 모형 구성과 강화학습의 파라미터 변수들을 적절히 조합하여 적절한 작업 배정 모형의 개발이 가능하다는 점을 알 수 있었다.

키워드: 인공지능(artificial intelligence), 순서 의존적 준비시간(sequence dependent setup time), 강화학습(reinforcement learning), 시뮬레이션(simulation), FlexSim 소프트웨어(FlexSim software)

I. Introduction

단일기계 작업장에서는 모든 작업물이 동일한 기계 1대를 거쳐 완성된다. 따라서, n개의 작업 J_1, J_2, ..., J_n이 주어질 때, 이들의 투입 순서를 적절히 정하여 지연 벌금이나 지연 작업수 등과 같은 성능 평가지표를 최소화하는 것이 중요한 목표가 된다[1].

한편, 기계의 예열이나 세팅 변경 등과 같이, 실제로 공정을 진행하는 것은 아니나, 공정을 진행하기 전에 수행해야 하는 절차를 준비(setup)라 하는데, 어떤 작업을 J_j를 처리하기 전에 발생하는 준비시간이 직전에 처리한 작업 J_i와 J_j의 유형에 따라 달라지는 경우를 순서 의존적 준비시간(sequence dependent setup time)이라 한다. 일반적으로 순서 의존적 작업 준비시간이 존재하는 공정인 경우, 작업물들의 투입 순서가 작업장 내 작업물들의 대기시간, 흐름률, 납기준수율 등에 상당한 영향을 미치기 때문에 작업물 1개에 대한 처리가 완료되었을 때, 다음에 처리할 작업물을 적절히 선택하여야 한다.

본 논문에서는 의존적 작업 준비시간이 있는 단일 기계 생산라인에서 인공지능(artificial intelligence, AI) 기법 중의 하나인 강화학습(reinforcement learning)을 적용하여 대기 중인 것들 중 최적의 작업물을 선택하는데 사용할 수 있는 작업 배정 모형을 개발하고자 한다. 또한, 상태(state)나 보상(reward) 함수를 달리 하면서 학습한 모형들의 성능을 비교해 보고, 학습 조건과 모형의 성능 간의 관계에 대해서도 토의해볼 것이다.

II. 연구 배경

제조 현장의 주요 문제 중 하나인 작업 스케줄링 최적화 문제는 모든 문제에 대해 최적의 솔루션을 제공하기에는 까다로운 NP-Hard 문제에 속한다[2]. 이로 인해, 실제 현장에서 작업 순서를 결정할 때는 휴리스틱 작업 배정 규칙을 이용하여 비교적 짧은 시간 안에 근사 최적해(nearly optimal solution)을 찾는 것을 목표로 하는 연구가 많다. 하지만 휴리스틱 규칙들은 특정 유형의 문제에만 적용할 수 있고 새로운 규칙을 개발하는 것이 까다롭다는 문제점이 존재한다. 최근에는 새로운 인공지능 알고리즘이 많아지고 성능이 강력해짐에 따라 강화학습과 Q-러닝 기반 알고리즘을 사용하여 최적에 가까운 스케줄링 솔루션을 찾는데 상당한 연구가 이루어지고 있다[3][4]

강화학습의 목표는 기본적으로 에이전트(agent)가 환경(observation)과의 상호 작용해서 얻은 데이터를 기반으로 행동(action)을 하여 얻은 보상을 최대화 하는 것을 목표로 한다. 강화학습의 모델링은 MDP(markov decision process)을 이용해 (S, A, P, R, γ)로 모델링 할 수 있다고 가정한다. 여기서 S는 연속 상태 공간, A는 연속 작업 공간, P는 다음 상태의 확률 분포, R은 에이전트가 받는 보상이다. 에이전트가 어떤 상태에 대해 행동(action)을 할 경우 올바른 행동인지 잘못된 행동인지 판단하기 위한 지표이다. γ은 할인율을 의미하며, 현재 즉시 받는 보상과 미래에 받게되는 보상은 다른 가치를 가지므로 할인율을 적용해 미래 가치를 현재 가치로 환산하는 것을 의미한다[5].

본 연구에서는 3D 시뮬레이션 소프트웨어인 FlexSim을 사용하여 강화학습에 필요한 환경을 구성하고 모델 훈련 및 평가를 위해 stable-baseline3 라이브러리를 사용하였다[6].

III. 강화학습을 위한 FlexSim 모형

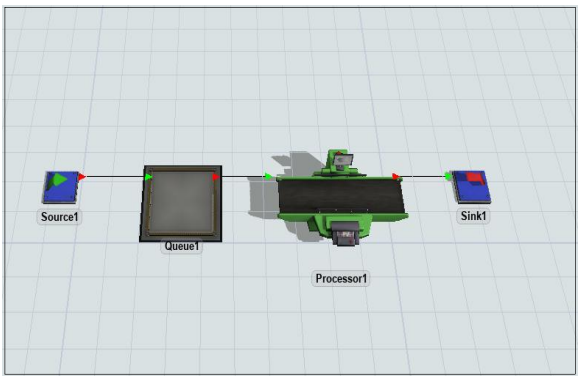

Fig. 1. Single Machine Model Layout for Reinforcement Learning

<Fig. 1>은 FlexSim 소프트웨어를 사용하여 작성한 단일기계 작업장 모형의 전반적인 레이아웃을 보여준다. 대상 모형은 1개의 설비가 배치되어 있으며 총 5가지 유형의 아이템을 생산한다. 각 유형별 아이템 도착 간격은 평균이 10초인 지수 분포를 따르며, 도착한 작업물들은 처음에 대기 장소에 해당하는 Queue1 개체를 거쳐 기계를 의미하는 Processor1개체에서 처리된 다음 Sink1에 도달 시 모형에서 삭제되는 단순한 구조이다. 또한, 각 아이템을 처리하는 시간은 10초로 동일하지만 <Table 1>과 같이 아이템들이 처리되기 전에 순서 의존적 준비시간을 갖는다.

Table 1. Sequence dependent setup time

Type	1	2	3	4	5
1	0	10	20	30	40
2	10	0	10	20	40
3	20	10	0	10	30
4	30	20	10	0	10
5	40	30	20	10	0

Table 2. Configurations of reinforcement learning environment for each cases

case	상태(s)	보상(r)
1	$Type_{t-1}$	R1
2	$Type_{t-1}$	R2
3	$Type_{t-1}$, W_1 ~ W_5	R1
4	$Type_{t-1}$, W_1 ~ W_5	R2

예를 들어, 1번 타입의 아이템을 처리한 후에 후속으로 1번 아이템을 처리하게 되면 셋업시간이 0초이지만, 5번 아이템을 처리할 경우 셋업시간이 40초이다. 즉, 효율적인 작업장 운영을 위해서는 이번에 처리할 작업물과 다음번에 처리할 작업물의 셋업시간을 최소한으로 하는 것이 바람직할 것이다.

본 논문에서는 <Table 2>와 같이 관찰영역의 상태 값과, 보상함수를 4가지 시나리오로 구성하였으며 case1, 2는 상태변수는 동일하나 보상함수가 다르며 case3, 4는 case 1, 2에서 상태변수를 추가하여 시나리오를 구성하였다. 각 시나리오에서 언급한 변수에 대한 설명은 다음과 같다.

t번째로 처리할 작업물을 Quque1 개체에서 선택하고자 할 때, 에이전트가 관찰하는 상태 S_t는 다음과 같다.

$$S_t = [Type_{t-1}, W_1, W_2, W_3, W_4, W_5] \quad (1)$$

단, $Type_{t-1}$은 현재 직전에 $t-1$번째로 처리한 작업물의 유형 번호를 의미하고, W_k $(k=1,2,...,5)$는 Queue1에서 대기 중인 작업물 중 k번 유형인 것들의 개수이다.

에이전트가 취할 행동은 t번째로 선택할 아이템의 유형 번호이며, 행동이 정해지는 경우, Queue1 개체에 대기 중인 해당 유형의 아이템 중 가장 먼저 도착했던 것이 Processor1로 진행하게 된다. 나아가, 에이전트가 얻는 보상은 아래와 같이 2가지를 사용하였다.

$$R_1 = \frac{10}{T_t(\text{처리완료}) - T_{t-1}(\text{처리완료})} \quad (2)$$

$$R_2 = \frac{10}{T_t(Sink1\ \text{진입}) - T_t(Queue1\ \text{이탈})} \quad (3)$$

단, $T_t(E)$는 t번째 아이템에 사건 E가 발생한 시점을 뜻한다.

IV. 실험결과

강화학습을 통해 모델을 훈련하기 위해 FlexSim에서 제공하는 Python 강화학습 라이브러리와 Stable-Baseline3의 PPO 알고리즘을 사용하여 앞 절에서 언급한 4가지 시나리오에 대해 timestep = 50,000회에 대한 학습을 실시하였다. 나아가, 4가지 시나리오에서 학습된 모형의 성능을 평가하기 위해 Queue1의 평균 대기시간 지표를 사용하여 분석을 하였고 실험결과는 <Table 3>에 요약하였다.

4가지 시나리오에 대해 10000초 동안에 걸쳐 시뮬레이션 실험을 한 결과, Processor1에서 아이템 타입을 당 겨올 때, "임의 선택"을 하는 것보다, RL 모형을 이용하여 선택하는 것이 평균대기시간이 훨씬 짧다는 것을 확인할 수 있었다. 4가지 시나리오에서 중에서도 case3과 같이 상태를 (1)과 같이 설정하고, 보상함수로는 R_1을 사용했을 때, 학습을 통해 얻어진 작업 배정 모형이 가장 좋은 성능을 나타내는 것으로 확인되었다.

반면, case2, 3에서는 Queue1에서 대기중인 타입별 아이템 개수 W_1 ~ W_5와 현재 t시점에서 지난번에 처리한 아이템 타입번호 $Type_{t-1}$의 변수와의 상관관계가 크지 않은 것으로 확인되었다. 즉, 에이전트가 상태(S_t)를 측정하고 현재 상태에서 보상을 최대화하기 위한 적절한 행동을 선택하여, 다음 상태(S_{t+1})로 전환될 때, 환경으로 주어지는 즉각적인 보상을 사용하여 장기적인 성과를 계산해서 지속적으로 학습을 한다고 하였을 때, 보상함수와 관찰영역에서 획득할 수 있는 데이터들에 대한 상관관계를 적절히 구성하는 것이 좋은 강화학습 모델을 만들 수 있을 것으로 판단된다.

Table 3. Comparisons of Performance Measures

	Queue1 평균대기시간(초)	
Case	임의 선택	RL 모형 이용
1	3058.56	1271.9
2	3019.4	1706.66
3	2896.28	1098.51
4	3023.43	2058.59

V. 결론

본 연구에서는 의존적 작업 준비시간이 있는 단일기 계 작업장에 강화학습을 적용하였고, 시뮬려이션 모형 작성 및 이를 이용한 실험을 통해 아래와 같은 결론들을 도출하였다.

첫째, 강화학습을 통해 NP-hard 문제에 속하는 스케줄링 최적화 문제 풀이에 다양하게 응용할 수 있을 것으로 보인다. 즉, 단순히 어떤 조합을 시도하는 것을 넘어 주변의 환경을 감지해 데이터를 얻고 이에 기반하여 최대한의 보상을 얻기 위해 행동하는 등의 과정을 통해 비선형적 패턴을 학습할 수 있으며 새로운 패턴을 찾아갈 수 있다는 점에서 기존 휴리스틱 알고리즘보다 굉장히 좋은 성능을 낼 수 있을 것으로 보인다.

둘째, FlexSim 소프트웨어에서 제공하는 강화학습 프레임워크와 이미 개발된 강화학습 외부 라이브러리를 적절히 이용하면 제조현장뿐만 아니라 현실세계에서 요구하는 다양한 제약조건 고려한 의사결정 모형을 비교적 쉽게 만들 수 있을 것으로 판단된다.

ACKNOWLEDGEMENT

This research was supported by the Ministry of Education of the Republic of Korea and National Research Foundation(NRF) (NRF-2022S1A5C2A03093301)

REFERENCES

[1] B. Yang, and J. Geunes, "A single resource scheduling problem with job-selection flexibility, tardiness costs and controllable processing times," Computer & Industrial Vol. 53, No. 3, pp.420-432, 2007.

[2] J. M. Framinan, J. N. Gupta, and R. Leisten, "A review and classification of heuristics for permutation flow-shop scheduling with makespan objective," Journal of the Operational Research Society, Vol. 55, No. 12, pp. 1243-1255, 2004.

[3] W. Jiahao, P. Zhiping, C. Delong, L. Qirui, and H. Jieguang, "A multi-object optimization cloud workflow scheduling algorithm based on reinforcement learning," In International Conference on Intelligent Computing, Springer, Cham, pp. 550-559, 2018.

[4] Y. Wei, D. Kudenko, S. Liu, L. Pan, L. Wu and X. Meng, "A reinforcement learning based workflow application scheduling approach in dynamic cloud environment," in Collaborative Computing: Networking Applications and Worksharing, Springer, Cham, pp. 120-131, 2018.

[5] R. Martín-Martín, M. A. Lee, R. Gardner, S. Savarese, J. Bohg, and A. Garg, "Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks," In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 1010-1017

[6] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baseline3: Reliable reinforcement learning implementations," Journal of Machine Learning Research, Vol.22, pp.1-8, 2021.