대한조선학회논문집 Research Paper
Journal of the Society of Naval Architects of Korea
eISSN:2287-7355, Vol. 63, No. 4, pp. 265-274, August 2026
https://doi.org/10.3744/SNAK.2026.63.4.265

조선소 판넬라인 블록 투입 순서 및 베이 할당 최적화를 위한 액션 마스킹 및 Self-Labeling 기반 학습 방법론

오현진[1] · 조영인[1] · 한영찬[1] · 최재호[2] · 김준현[2] · 우종훈[1,3]
서울대학교 조선해양공학과[1]
삼성중공업 스마트야드 연구센터[2]
서울대학교 해양시스템공학연구소[3]

An Action-masking and Self-labeling-based Learning Method for Block Sequencing and Bay Assignment Optimization in Shipbuilding Panel Lines

Hyunjin Oh[1] · Youngin Cho[1] · Yeongchan Han[1] · Jaeho Choi[2] · Junhyeon Kim[2] · Jonghun Woo[1,3]
Department of Naval Architecture and Ocean Engineering, Seoul National University[1]
Smart Yard Research Center, Samsung Heavy Industries[2]
Research Institute of Marine Systems Engineering, Seoul National University[3]

The block sequencing and bay assignment problem in a shipbuilding panel line is a complex combinatorial scheduling problem in which upstream serial processes and downstream parallel bay processes are coupled under multiple operational constraints. In practical panel-line planning, not all field rules have the same level of strictness. Some constraints must be maintained to ensure physical and procedural feasibility, whereas others are relaxable operational constraints that may be relaxed only when no candidate satisfies all rules simultaneously. This study proposes a constraint-priority-based learning method that combines action masking and self-labeling for two-stage panel-line scheduling. The proposed method first constructs a candidate set satisfying hard constraints and then applies relaxable operational constraints according to a predefined priority. If the candidate set becomes empty, only the relaxable operational constraints are gradually relaxed. A pointer-network-based policy selects the next block from the masked candidates, and a rule-based bay-assignment module assigns the selected block to a feasible downstream bay. Multiple rollouts are generated for each instance, and the best rollout is selected as a pseudo-label according to the lexicographic order of constraint noncompliance, makespan, and longi workload balance. Experiments on actual and generated panel-line data show that the proposed method improves makespan and reduces relaxable-constraint noncompliance compared with manual and heuristic schedules while maintaining hard constraints.

Keywords : Panel line scheduling(판넬라인 일정 계획), Block sequencing(블록 투입 순서), Bay assignment(베이 할당), Action masking(액션 마스킹), Self-Labeling Improvement Method(자가 라벨링 학습 방법론)

1. 서 론

조선 산업의 선박 건조 공정은 설계, 부재 가공, 소조립, 대조립, 탑재에 이르는 복합적 생산 체계로 구성된다. 이 가운데 판넬 블록 조립 공정은 선박 선체의 상당 부분을 차지하는 평 블록을 반복적으로 생산하는 핵심 상류 공정으로서, 후속 조립 및 탑재 공정의 착수 시점에 직접적인 영향을 미친다. 판넬 조립 라인은 공정 흐름이 비교적 일정하게 반복되는 흐름 공정(Flow Shop)의 성격을 가지므로, 블록 투입 순서에 따라 라인 유휴와 병목이 달라지고 결과적으로 전체 작업 완료시간(Makespan)이 크게 변한

Received : 21 April 2026 | Revised : 9 June 2026 | Accepted : 29 June 2026
† Corresponding author : Jonghun Woo, j.woo@snu.ac.kr

다. 또한 실제 현장에서는 블록 착수일, 작업 달력, 일일 용량, 특정 블록의 투입 순서, 연속 배치, 짝 블록 관계, 베이 할당 규칙 등 다양한 운영 제약을 동시에 고려해야 한다.

조선소 판넬라인 일정 계획에 관한 기존 연구는 최적화, 시뮬레이션, 메타 휴리스틱을 중심으로 발전해 왔다. Koh (1996)는 조선소 블록 조립 공정을 순열 흐름 공정으로 모델링하고 유전 알고리즘(GA)을 적용하여 Makespan 개선 가능성을 보였다. Lee et al. (2009)은 판넬 조립 공장 전용 시뮬레이션 기반 생산 실행 시스템을 개발하고, 모의 담금질(Simulated Annealing)과 이산 사건 시뮬레이션(DES)을 결합하여 시퀀싱 최적화를 수행하였다. Wang et al. (2016)은 자재 도착, 가공 시간, 납기의 불확실성을 고려한 비선형 정수 계획 및 롤링 호라이즌(Rolling Horizon) 기반 재계획 방법론을 제안하였다. Yang et al. (2019)은 판넬라인을 블로킹 흐름 공정(Blocking Flow Shop)으로 정의하고, 다목적 Memetic Algorithm과 퍼지 로직을 결합하여 Makespan과 부하 평준화를 함께 고려하였다. 최근에는 Zhou et al. (2023)이 그래프 기반 상태 표현과 강화학습을 결합한 판넬 블록 조립 라인 스케줄링 방법을 제안하였고, Li et al. (2024), Xu et al. (2024), Kwak et al. (2025)은 각각 복합 자원 제약, 블로킹 및 운송 제약, DES 검증을 포함한 조선 생산 일정 계획 문제를 다루었다.

보다 일반적인 관점에서 판넬라인의 블록 투입 순서 결정은 순열 흐름 공정 계획 문제(Permutation Flow Shop Scheduling Problem, PFSP)와 관련된다. PFSP는 2기계 환경에서는 Johnson (1954)의 규칙으로 최적해를 구할 수 있으나, 3기계 이상에서는 NP-hard로 알려져 있으며(Garey et al., 1976), 이후 다양한 정확해 기법, 휴리스틱, 메타 휴리스틱이 연구되어 왔다(Nawaz et al., 1983; Ruiz and Vázquez-Rodríguez, 2010; Naderi et al., 2023). 최근에는 스케줄링을 순차 의사결정 문제로 정의하고 강화학습을 적용하는 연구가 증가하고 있으며(Ren et al., 2021; Pan et al., 2023), 포인터 네트워크(Vinyals et al., 2015) 기반 정책 학습도 순열형 조합 최적화 문제에서 효과적인 구조로 활용되고 있다(Bello et al., 2016; Cho et al., 2022).

그러나 실제 판넬라인 일정 계획은 단순한 PFSP와 달리 블록 투입 순서와 베이 할당을 함께 결정해야 하며, 시점마다 선택 가능한 후보 블록 집합이 현장 상태와 제약조건에 따라 달라진다. 또한 일부 제약은 반드시 유지되어야 하지만, 일부 운영 제약은 후보가 없는 경우 현업 협의에 따라 제한적으로 완화될 수 있다. 따라서 실제 적용 가능한 스케줄링 방법은 하드 제약을 유지하면서, 완화 가능한 운영 제약의 미준수 횟수와 Makespan, 론지 부하 균형을 함께 고려할 수 있어야 한다.

본 연구는 이를 위해 조선소 판넬라인 블록 투입 순서와 론지 베이 할당을 연계한 2단계 학습 기반 스케줄링 방법론을 제안한다. 본 연구의 기여는 다음과 같다. 첫째, 판넬라인 문제를 블록 투입 순서 결정과 베이 할당이 결합된 순차 의사결정 문제로 정식화한다. 둘째, 현업 협의 기반 제약 우선순위를 반영하여 하드 제약을 유지하고, 완화 가능한 운영 제약만 단계적으로 조정하는 액션 마스킹 절차를 제안한다. 셋째, 포인터 네트워크 기반 정책과 자가 라벨링 학습을 결합하여 정답 스케줄 데이터 없이도 제약 미준수, Makespan, 론지 부하 균형을 함께 고려하는 스케줄링 정책을 학습한다. 넷째, 실적 데이터와 생성 데이터를 사용하여 제안 방법의 현장 적용 가능성과 일반화 성능을 검증한다.

2. 문제 정의

본 장에서는 연구 대상 문제를 일반적인 순열 흐름 공정 계획 문제에서 출발하여, 후공정 분기 구조가 추가된 분기형 판넬라인으로 확장하고, 마지막으로 제약 그룹과 목적함수를 정의한다.

2.1 순열 흐름 공정 계획 문제

순열 흐름 공정 일정 계획 문제는 n개의 작업으로 구성된 집합 $J=\{1,2,...,n\}$와 m개의 기계로 구성된 집합 $M=\{1,2,...,m\}$이 주어졌을 때, 모든 작업이 모든 기계를 동일한 순서로 통과하도록 하나의 공통 순열을 결정하는 문제이다. 각 작업 $j\in J$가 기계 $k\in M$에서 갖는 처리시간을 $p_{j,k}$라 하고, 공통 순열 $\pi=(\pi_1,\pi_2,...,\pi_n)$라 가정한다. 여기서 π_i는 순열 상 i번째로 처리될 작업의 인덱스를 의미한다.

일반적인 flow shop 또는 job shop과 PFSP의 핵심적인 차이점은 모든 기계에서 동일한 작업 순서가 유지된다는 점이다.

예를 들어 세 개의 작업 $J=\{1,2,3\}$과 세 개의 기계 $M=\{1,2,3\}$이 있다고 하자. 일반적인 flow shop 또는 job shop에서는 기계마다 작업 처리 순서가 다를 수 있다. 기계 1에서는 1-2-3, 기계 2에서는 2-3-1, 기계 3에서는 3-1-2와 같은 순서가 허용될 수 있다. 이 경우 각 기계마다 가능한 작업 순서는 $3!$개이며, 세 개의 기계에 대해 가능한 순서 조합은 $(3!)^3$개가 된다. 일반화하면 각 기계의 작업 순서를 독립적으로 결정하는 경우 탐색 공간은 $(n!)^m$에 해당한다.

반면 PFSP에서는 하나의 공통 순열만 결정한다. 예를 들어 $\pi=(2,1,3)$으로 결정되면 모든 기계에서 작업 2번, 1번, 3번 순서로 처리된다. 이 순열 제약은 탐색 공간을 $(n!)^m$에서 $n!$로 축소한다.

순열 π가 주어졌을 때, 기계 k에서 i번째 작업에 대한 완료시각 $C_{\pi_i,k}$는 다음과 같이 계산된다. 여기서 $i=1,\dots n$, $k=1,\dots m$이다.

$$C_{\pi_i,k}=\max(C_{\pi_i,k-1},C_{\pi_{i-1},k})+p_{\pi_i,k} \quad (1)$$

초기조건은 $C_{\pi_0,k}=0$, $C_{\pi_i,0}=0$이다. 목적함수는 Makespan $C_{\max}=C_{\pi_n,m}$을 최소화하는 최적 순열 π^*를 구하는 것이다.

그러나 본 연구의 판넬라인 문제는 공통 순열 결정만으로 완결되지 않는다. 병렬 론지 베이가 존재하고, 각 블록은 물리적 조건과 베이 상태를 고려하여 하나의 베이에 배정되어야 한다. 따라서 다음 절에서는 PFSP 구조를 판넬라인의 분기형 공정 구조로

확장한다.

2.2 분기형 흐름 공정 계획 문제

본 연구의 대상인 조선소 판넬라인은 전반부 공통 공정과 후반부 병렬 론지 베이 공정으로 구성된다. 전반부 공정에서는 모든 블록이 하나의 공통 투입 순서 π에 따라 처리된다. 후반부 공정에서는 각 블록이 베이 집합 중 하나의 베이에 할당되어 처리된다. 따라서 본 문제는 공통 순열 결정에 병렬 자원 배정이 결합된 분기형 하이브리드 흐름 공정으로 볼 수 있다.

블록 집합을 $J=\{1,2,...,n\}$, 전반부 공정 집합을 K^s, 후반부 병렬 베이 공정 집합을 K^b, 베이 집합을 B로 둔다. 블록 j의 공정 k의 처리시간은 $p_{j,k}$이며, 베이 할당 결과는 $b_j \in B$로 표현한다. 여기서 b_j는 블록 j가 할당된 론지 작업 구역을 의미한다.

전반부 공통 공정에서 순열 π의 i번째 블록 π_i가 공정 k를 완료하는 시각은 다음과 같다.

$$C_{\pi_i,k} = \max(C_{\pi_i,k-1}, C_{\pi_{i-1},k}) + p_{\pi_i,k}, \quad k \in K^s \tag{2}$$

후반부 베이 공정에서는 블록이 배정된 베이의 가용시각을 함께 고려한다. 블록 j가 할당된 베이 b_j에서 공정 k가 시작 가능한 가용시각을 $A_{b_j,k}$라 할 때, 후반부 공정 완료 시각은 다음과 같이 계산된다.

$$C_{j,k} = \max(C_{j,k-1}, A_{b_j,k}) + p_{j,k}, \qquad k \in K^b \tag{3}$$

블록 j가 공정 k를 완료하면, 해당 베이의 가용시각 $A_{b_j,k}$는 완료 시각 $C_{j,k}$로 갱신된다. 전체 스케줄의 Makespan은 모든 블록의 마지막 공정 완료 시각 중 최댓값으로 정의한다.

$$C_{\max}(\pi,b) = \max_{j \in J} C_{j,m} \tag{4}$$

2.3 문제 제약조건

본 연구에서는 판넬라인 스케줄링의 제약조건을 Capacity, Sequencing, Adjacency, Bay Assignment의 네 그룹으로 구분한다. Capacity는 일일 작업량, 작업 달력, 생산 가능 용량과 관련된 제약으로, 예를 들어 특정 일자의 허용 작업량이나 작업 가능 시간을 초과하는 블록은 후보에서 제외된다. Sequencing은 착수일 순서, 선후 관계, 연계 블록 관계를 반영하는 제약으로, 후공정 착수 순서를 역전시키거나 필수 선후 관계를 위반하는 블록을 제외한다. 두 제약은 생산 흐름과 실행 가능성을 유지하기 위한 하드 제약으로 적용한다. Adjacency는 특정 유형 블록 또는 고부하 블록의 연속 투입을 제한하는 제약이다. 예를 들어 동일 특성 블록이 연속으로 투입되거나, 작업량이 큰 블록이 충분한 간격 없이 배치되는 경우를 제한한다. 이는 작업 안정성과 부하 분산을 위한 운영 제약이므로, 모든 후보가 차단되는 경우 사전 정의된 우선순위에 따라 단계적으로 완화될 수 있다. Bay Assignment는 선택된 블록을 후반부 베이에 배정할 때 적용되는 제약이다. 블록의 폭, 길이, 론지 작업량 등으로 특정 베이에서 작업할 수 없는 경우 해당 베이는 후보에서 제외된다. 또한 대응 관계가 있는 블록은 현업 규칙에 따라 동일 베이 또는 지정 베이에 배정되어야 할 수 있다. 물리적 작업 가능성 및 필수 배정 규칙은 하드 제약으로 유지하고, 베이 선호, 연속 배정 패턴, 부하 균형은 운영 제약 또는 평가 기준으로 활용한다. 본 연구에서 $V(\pi,b)$는 스케줄(π,b)의 제약 미준수 횟수를 의미한다. 단, 이는 하드 제약 위반 수가 아니라, 현업 협의에 따라 불가피한 경우 제한적으로 완화 가능한 운영 제약의 미준수 횟수를 의미한다.

2.4 목적함수

본 연구의 목적은 현업상 반드시 유지해야 하는 제약을 만족하면서, 완화 가능한 운영 제약의 미준수 횟수, 전체 작업 완료 시간, 론지 부하 편차를 순차적으로 줄이는 것이다. 이를 위해 스케줄을 블록 투입 순서 π와 베이 할당 b의 쌍 (π,b)로 표현한다.

론지 베이의 작업 부하 균형을 함께 고려하기 위해 베이 부하 편차 $\Delta_{bay}(\pi,b)$를 정의한다. 블록 j의 론지 작업량을 ℓ_j라 한다. 베이 q에 할당된 총 론지 작업량 $L_q(\pi,b)$는 다음과 같이 계산된다.

$$L_q(\pi,b) = \sum_{j \in J:\ b_j = q} \ell_j, \quad q \in B \tag{5}$$

론지 부하 편차 $\Delta_{bay}(\pi,b)$는 베이별 총 론지 작업량의 최댓값과 최솟값의 차이로 정의한다.

$$\Delta_{bay}(\pi,b) = \max_{q \in B} L_q(\pi,b) - \min_{q \in B} L_q(\pi,b) \tag{6}$$

따라서 본 연구의 판넬라인 스케줄링 문제는 다음과 같은 사전식 목적을 갖는 최적화 문제로 정의한다. $C_{\max}(\pi,b)$는 식 (4)에서 정의한 Makespan이며, $V(\pi,b)$는 제약 미준수 횟수이다.

$$(\pi^*,b^*) = \min_{\pi,b}(V(\pi,b), C_{\max}(\pi,b), \Delta_{bay}(\pi,b)) \tag{7}$$

여기서 $\min$은 일반적인 가중합 최소화가 아니라, 괄호 안의 항을 왼쪽부터 순서대로 비교하는 사전식 최소화를 의미한다. 즉, $V(\pi,b)$가 더 작은 스케줄을 우선한다. $V(\pi,b)$가 동일한 경우에는 $C_{\max}(\pi,b)$이 더 짧은 스케줄을 선택하고, 두 값이 모두 동일한 경우에는 $\Delta_{bay}(\pi,b)$가 더 작은 스케줄을 선택한다.

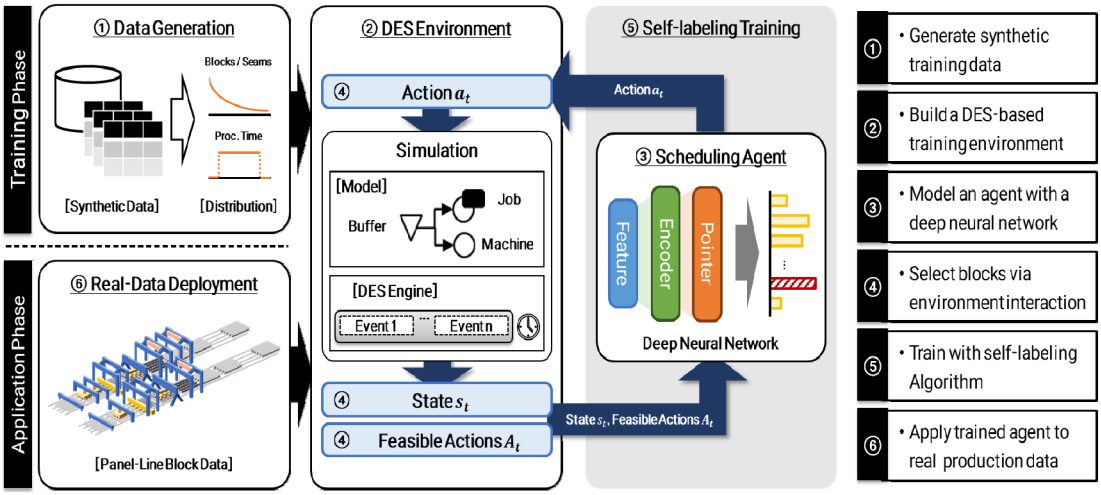

Fig. 1 Self-labeling-based framework for block sequencing and bay assignment in shipyard panel lines

3. 제안 방법론

본 장에서는 제안 방법의 전체 구조를 설명한다. 제안 방법은 DES 기반 판넬라인 환경과 스케줄링 에이전트의 상호작용으로 구성된다. DES 환경은 현재 생산 상태를 갱신하고, 각 시점에서 선택 가능한 후보 블록 집합을 생성한다. 스케줄링 에이전트는 액션 마스킹을 통과한 후보 집합 위에서 다음 투입 블록을 선택하며, 선택된 블록은 규칙 기반 베이 할당 절차에 따라 후반부 공정에 할당된다. 이 과정을 모든 블록이 투입될 때까지 반복하여 하나의 스케줄을 생성한다.

Fig. 1은 제안 방법의 전체 흐름을 나타낸다. DES 환경은 3.1절의 시뮬레이션 환경과 상태 갱신 절차에 해당하며, 유효 행동 집합은 3.3절의 액션 마스킹을 통해 생성된다. 스케줄링 에이전트는 3.4절의 포인터 네트워크 기반 정책으로 다음 투입 블록을 선택하고, 3.5절의 규칙 기반 베이 할당으로 할당한다. 이후 생성된 rollout은 3.6절의 Self-Labeling 학습에서 사용된다.

3.1 DES 환경

본 연구에서는 판넬라인의 블록 투입, 공정 진행, 베이 할당, 상태 갱신 과정을 이산 사건 시뮬레이션(DES) 환경으로 모델링한다. DES 환경은 제안 방법의 학습과 평가에 공통으로 사용되는 스케줄 생성 환경이며, 각 의사결정 시점에서 현재 생산 상태와 선택 가능한 후보 블록 집합을 제공한다.

DES 환경의 기본 작업 단위는 판넬 블록이다. 에이전트가 하나의 블록을 선택하면, DES 환경은 해당 블록의 공정 시작 시각과 완료 시각을 계산하고, 베이 할당 규칙에 따라 후반부 작업 구역을 결정한다. 이후 설비 가용시각, 작업 구역 가용시각, 일일 용량 사용량, 직전 선택 블록, 잔여 블록 집합, 작업 구역별 누적 작업량이 갱신된다. 갱신된 상태를 기준으로 다음 시점의 후보 블록 집합이 다시 계산되므로, 후보 집합은 고정되어 있지 않고 이전 선택 결과와 현재 생산 상태에 따라 매 step 달라진다.

DES 환경의 구현 타당성은 실적 데이터의 현업 수기 계획을 동일 환경에서 재현하는 방식으로 확인하였다. 구체적으로 현업 수기 계획의 블록 투입 순서와 베이 할당 결과를 DES 환경에 입력하고, 동일한 공정시간 계산 및 상태 갱신 절차로 Makespan과 제약 미준수 결과를 산출하였다. 이후 휴리스틱 방법과 제안 방법도 동일한 DES 환경에서 평가하여, 비교 결과가 평가 환경의 차이가 아니라 블록 선택 방식의 차이에서 발생하도록 하였다.

3.2 마르코프 결정 과정(MDP)

본 연구는 판넬라인 스케줄링 문제를 마르코프 결정 과정으로 정식화한다. 시점 t에서의 상태 s_t는 마스킹을 통과한 후보 블록 행렬 X_t와 전역 환경 벡터 e_t의 결합으로 정의된다.

$$s_t = (X_t, e_t) \tag{8}$$

후보 행렬 $X_t \in \mathbb{R}^{N_t \times d_b}$는 현재 선택 가능한 N_t개의 후보 블록 특성 행렬이다. 후보 블록 특성에는 블록의 물리적 크기, 심수, 론지 수 등이 포함된다. 환경 벡터 $e_t \in \mathbb{R}^{d_e}$는 현재 날짜와 시각, 당일 용량 사용률 등 전역 상태를 포함한다.

행동 a_t는 유효 후보 집합 A_t 내에서 하나의 블록을 선택하는 것이다. 액션 마스킹을 통과한 후보 집합 위에서 행동이 정의되므로, 행동 공간의 크기와 구성은 매 시점 달라진다.

전이는 DES 환경에서 수행된다. 정책이 블록을 선택하면, 환경은 해당 블록의 전반부 공정 시작 시각과 완료 시각을 계산하고, 베이 가용시각과 작업 부하를 갱신한다. 동시에 선택된 블록은 잔여 블록 집합에서 제거되며, 일일 용량 사용량과 직전 선택 블록 정보도 갱신된다. 종료 조건은 모든 블록이 투입 및 베이 할당을 완료한 경우이다.

본 연구에서는 별도의 step-wise 보상으로 정책을 업데이트하지 않고, rollout이 완료된 뒤 산출되는 평가 벡터를 후보 스케줄의 비교 기준으로 사용한다. 하나의 rollout을 τ라 하면, τ는 모

든 블록의 투입 순서와 베이 할당이 완료된 하나의 스케줄을 의미한다. 본 연구에서는 2.4절의 목적함수와 동일하게 다음 평가 벡터를 사용한다.

$$F(\tau) = (V(\tau), C_{\max}(\tau), \Delta_{bay}(\tau)) \tag{9}$$

여기서 $F(\tau)$는 rollout τ의 평가 벡터이다. 즉, 중간 step의 보상을 별도로 설계하지 않고, rollout이 종료된 뒤 $F(\tau)$를 기준으로 후보 rollout을 비교한다. 이 평가 결과는 3.6절의 Self-Labeling 단계에서 최상 rollout을 선택하는 기준으로 사용된다.

3.3 계층적 액션 마스킹

액션 마스킹은 현재 상태에서 선택 불가능한 블록을 후보 집합에서 제외하는 절차이다. 액션 마스킹은 무효 행동을 사전에 제거하여 탐색공간을 줄이고 정책 학습을 안정화하는 데 효과적인 방법으로 알려져 있다(Huang and Ontañón, 2022; Zahavy et al., 2018). 본 연구에서는 단순히 이미 선택된 블록만 제외하는 정적 마스킹이 아니라, 현재 DES 상태와 운영 제약을 반영하여 매 step 유효 후보 집합을 다시 구성하는 동적 마스킹을 사용한다.

마스킹은 계층적으로 수행된다. 먼저 아직 선택되지 않은 잔여 블록을 초기 후보로 두고, Capacity와 Sequencing 등 하드 제약을 적용하여 생산 용량, 착수 순서, 선후 관계를 만족하지 못하는 후보를 제거한다. 이후 Adjacency 계열 운영 제약을 적용하여 특정 블록군이나 고부하 블록이 연속으로 배치되는 후보를 추가로 제한한다. 베이 할당과 관련된 물리적 적합성은 후속 베이 할당 단계에서 별도로 확인한다.

모든 운영 제약을 적용한 뒤 후보 집합이 공집합이 되면, 일정 생성을 중단하지 않기 위해 완화 가능한 운영 제약을 사전 정의된 순서에 따라 단계적으로 완화한다. 본 연구의 완화 순서는 라인 그룹 연속 배치 제한, 조립 타입 혼합 제한, Cross seam 혼합 배치 제한, C/Seam 블록 간격 제한, 곡판 블록 간격 제한, 고심수 블록 간격 제한 순서로 적용된다. 이들은 실행 가능성보다는 작업 안정성과 부하 분산을 위한 운영 제약이므로, 후보가 모두 차단되는 경우에 완화할 수 있다. 반면 Capacity, Sequencing, 물리적 베이 적합성은 완화하지 않는다. 완화로 인해 발생한 미준수는 $V(\pi, b)$에 반영된다.

최종 마스크 벡터 M_t는 유효 행동에 0, 무효 행동에 $-\infty$를 부여하며, 정책이 무효 행동을 선택하지 않도록 한다. 정책 로짓을 u_t라 하면, 다음과 같은 행동 확률 분포를 생성한다.

$$p_t = softmax(u_t + M_t) \tag{10}$$

이처럼 제안 방법은 전체 잔여 블록 중에서 임의로 다음 블록을 선택하는 것이 아니라, 하드 제약을 먼저 만족시키고, 운영 제약을 단계적으로 적용한 후보 집합 안에서 정책이 다음 투입 블록을 선택하도록 한다.

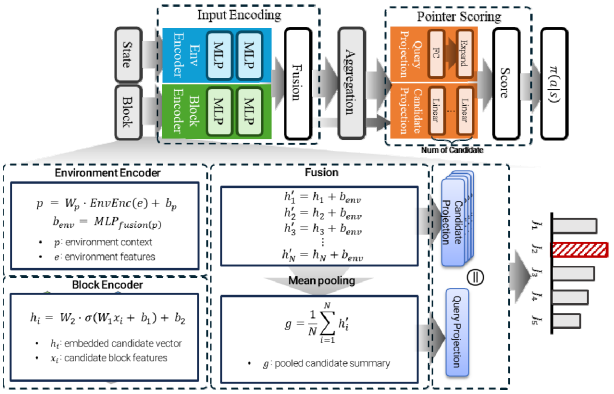

Fig. 2 Architecture of the proposed scheduling agent

3.4 포인터 네트워크(Pointer Network)

정책 신경망은 포인터 네트워크 기반의 단일 스텝 후보 선택 모델로 구현하였다. 후보 블록 행렬 X_t와 전역 환경 벡터 e_t의 차원은 다음과 같다.

$$X_t \in \mathbb{R}^{N_t \times 34},\ e_t \in \mathbb{R}^{31} \tag{11}$$

입력 인코딩 단계에서 각 후보 블록 벡터 $x_i \in \mathbb{R}^{34}$는 블록 인코더 Φ_x를 통해 256차원 임베딩 h_i로 변환된다.

$$h_i = \Phi_x(x_i),\quad \Phi_x : \mathbb{R}^{34} \to \mathbb{R}^{256} \tag{12}$$

전역 환경 벡터 $e_t \in \mathbb{R}^{31}$는 별도의 환경 인코더 Φ_e를 통과하여 환경 표현 r_t를 생성한 뒤, 선형 투영을 통해 256차원 환경 컨텍스트 c_t로 변환된다.

$$r_t = \Phi_e(e_t),\quad \Phi_e : \mathbb{R}^{31} \to \mathbb{R}^{31} \tag{13}$$

$$c_t = W_e r_t,\quad W_e : \mathbb{R}^{31} \to \mathbb{R}^{256} \tag{14}$$

환경 융합 단계에서는 환경 컨텍스트 c_t를 2층 MLP Ψ에 통과시켜 환경 편향 벡터 δ_t를 생성하고, 이를 모든 후보 블록 임베딩에 잔차 형태로 더한다.

$$\delta_t = \Psi(c_t),\quad \Psi : \mathbb{R}^{256} \to \mathbb{R}^{256} \tag{15}$$

$$h_i' = h_i + \delta_t \tag{16}$$

후보 집합의 전역 요약은 융합된 후보 임베딩들의 평균 풀링으로 계산한다.

$$g_t = \frac{1}{N_t}\sum_{i=1}^{N_t} h_i',\quad g_t \in \mathbb{R}^{256} \tag{17}$$

이후 전역 요약 벡터 g_t와 환경 컨텍스트 c_t를 결합하여 query 벡터 q_t를 생성한다.

$$q_t = W_c(g_t + c_t), \qquad W_c : \mathbb{R}^{256} \rightarrow \mathbb{R}^{256} \tag{18}$$

다음으로 각 후보 블록에 대한 로짓은 포인터 메커니즘으로 계산된다.

$$z_{t,i} = C \cdot \tanh(v^{\top} \tanh(W_q q_t + W_r h_i')) \tag{19}$$

여기서 C는 로짓 스케일링 상수이다. 정책은 현재 후보 집합 위에서 temperature-scaled softmax를 적용하여 확률 분포를 형성하며, 그로부터 다음 블록을 선택한다. 즉, 본 모델은 고정 클래스 분류기가 아니라 시점마다 달라지는 후보 집합 위에서 동작하는 동적 선택 모델이다.

3.5 규칙 기반 베이 할당

정책이 다음 투입 블록을 선택하면, 베이 할당은 DES 환경 내부의 규칙 기반 절차로 수행된다. 본 연구의 학습 에이전트는 블록 투입 순서 결정에 집중하고, 베이 할당은 블록의 물리적 조건, 연계 관계, 베이 가용 상태, 누적 작업량을 고려한 결정론적 규칙에 따라 수행된다.

결정론적 베이 할당 규칙은 다음 순서로 적용된다. 첫째, 폭이나 부재 수 등 설비 조건을 기준으로 물리적으로 작업 가능한 베이 집합을 구성한다. 둘째, 짝 블록의 기배정 베이가 작업 가능한 후보에 포함된 경우 동일 베이를 우선한다. 셋째, 특정 유형 블록의 동일 베이 연속 배치를 제한한다. 마지막으로 남은 후보 중 누적 작업량이 적은 베이를 선택한다.

3.6 Self-Labeling 학습

본 연구에서는 Corsini et al. (2024)이 제안한 Self-Labeling Improvement Method(SLIM)를 판넬라인 스케줄링 문제에 맞게 적용한다. 매 에피소드마다 현재 정책으로부터 M개의 rollout을 생성하고, 목적함수의 사전식 순서에 따라 최상 rollout τ^*을 선택한다. 이후 τ^*에서 추출된 step-wise 행동 시퀀스 $\{a^*\}$을 pseudo-label로 사용하여, 정책을 다음의 cross-entropy 손실로 학습한다.

$$L_{sl}(\theta) = -\mathbb{E}_t[\log \pi_\theta(a_t^* \mid s_t)] \tag{20}$$

이 방식은 매 step 보상으로 정책을 직접 업데이트하는 방식이 아니라, 현재 정책이 생성한 후보 스케줄 중 가장 좋은 스케줄의 선택 패턴을 학습하는 방식이다. 또한 보상 설계에 대한 의존도를 낮추고, 희소 보상이나 잡음이 있는 보상 환경에서 발생할 수 있는 학습 불안정을 완화하는 데 유리하다.

4. 실 험

본 장에서는 제안 방법의 성능을 생성 데이터와 실적 데이터에서 평가한다. 모든 비교 방법은 3.1절에서 정의한 동일 DES 기반 판넬라인 환경에서 평가하였다. 생성 데이터 실험에서는 블록 수와 문제 분포 변화에 따른 일반화 성능을 확인하고, 실적 데이터 실험에서는 현업 수기 계획 및 휴리스틱 대비 제안 방법의 현장 적용 가능성을 검증한다.

실험 평가는 2.4절에서 정의한 목적함수 $V(\pi, b)$, $C_{\max}(\pi, b)$, $\Delta_{bay}(\pi, b)$를 기준으로 수행한다. Makespan 비교에는 인스턴스별 절대 완료 시간 차이를 정규화하기 위해 상대 RPD(Relative Percentage Deviation)를 추가로 사용한다.

$$RPD_m(\%) = \frac{C_{\max}^m - C_{\max}^{best}}{C_{\max}^{best}} \times 100 \tag{21}$$

여기서 m은 비교 방법, $C_{\max}^m$는 방법 m의 Makespan, $C_{\max}^{best}$는 동일 인스턴스에서 비교 방법 중 가장 짧은 Makespan을 의미한다.

4.1 실험 환경

실험 데이터는 생성 데이터와 실적 데이터로 구분된다. 생성 데이터는 실제 판넬라인 블록의 속성 분포와 작업 특성을 반영하여 구성하였으며, 하나의 블록을 하나의 스케줄링 단위로 사용한다. 블록 수는 Small, Medium, Large 구간으로 나누고, 문제 분포는 여섯 가지로 설정하였다. General은 기준 조건, Paired Block은 짝 블록 관계 비중이 높은 조건, Subassembly는 연계 구조 블록 비중이 높은 조건, Mixed Relation은 짝 블록과 연계 구조가 함께 존재하는 조건이다. Heavy Workload는 심 수, 처리시간, 블록 크기 등 작업 부하가 증가한 조건이며, High Utilization은 생산 부하율과 용량 압박이 높은 조건이다.

생성 데이터 실험에서는 SPT(Shortest processing time first), MSF(Minimum seam first), LPT(Longest processing time first), Proposed를 비교한다. 실적 데이터 실험에서는 현업 수기 계획을 추가하여 Manual, MSDH(masking-based start-date heuristic), SPT, MSF, LPT, Proposed를 비교한다. 여기서 Manual은 현업 수기 계획을 그대로 재현한 결과이고, MSDH는 수기 계획의 착수일 구간을 유지한 상태에서, 마스킹 기반으로 블록 순서를 재구성한 휴리스틱이다. SPT, MSF, LPT는 각각 총 처리시간이 짧은 블록, 심수가 적은 블록, 총 처리시간이 긴 블록을 우선 선택하는 규칙이다.

4.2 생성 데이터 실험

생성 데이터 실험은 문제 규모와 분포 변화에 따른 성능을 확인

Table 1 Generated-data evaluation: Makespan RPD (%) with mean $C_{\max}$

Scale	Type	SPT	MSF	LPT	Proposed
Small	T1	10.72 (71.81)	10.83 (71.89)	5.45 (68.39)	0.00 (64.86)
	T2	10.99 (91.69)	7.95 (89.17)	5.59 (87.23)	0.00 (82.61)
	T3	8.90 (70.78)	8.65 (70.62)	5.73 (68.71)	0.00 (64.99)
	T4	10.28 (84.44)	8.82 (83.32)	7.38 (82.22)	0.00 (76.57)
	T5	8.85 (75.49)	6.94 (74.17)	4.14 (72.23)	0.00 (69.35)
	T6	13.37 (66.94)	6.35 (62.80)	6.22 (62.72)	0.00 (59.05)
Medium	T1	8.29 (175.35)	4.09 (168.55)	1.45 (164.27)	0.00 (161.93)
	T2	8.72 (204.56)	5.53 (198.56)	4.20 (196.07)	0.00 (188.16)
	T3	7.38 (170.16)	5.41 (167.05)	1.11 (160.24)	0.00 (158.47)
	T4	8.74 (181.12)	2.72 (171.08)	1.29 (168.71)	0.00 (166.56)
	T5	7.15 (174.98)	3.62 (169.22)	2.07 (166.69)	0.00 (163.30)
	T6	6.94 (174.47)	4.04 (169.76)	1.80 (166.09)	0.00 (163.16)
Large	T1	6.31 (280.94)	3.43 (273.33)	1.64 (268.60)	0.00 (264.26)
	T2	7.34 (305.67)	3.66 (295.19)	0.98 (287.56)	0.00 (284.77)
	T3	6.23 (252.19)	3.31 (245.26)	1.33 (240.56)	0.00 (237.39)
	T4	8.71 (276.20)	3.70 (263.48)	1.21 (257.15)	0.00 (254.08)
	T5	7.37 (276.65)	3.67 (267.14)	1.68 (261.99)	0.00 (257.68)
	T6	6.87 (260.03)	3.91 (252.83)	1.65 (247.31)	0.00 (243.31)

하기 위해 수행하였다. Table 1은 생성 데이터의 각 Scale-Type 조건별 평균 Makespan RPD(%)를 제시하며, 괄호 안에는 평균 Makespan $C_{\max}$을 나타낸다. Table 2는 동일 조건에서의 제약 미준수 횟수 $V(\pi,b)$를 나타낸다.

Table 1과 Table 2에서 Scale은 Small, Medium, Large 인스턴스를 의미하며, 각각 블록 수 20~80개, 90~140개, 150~200개로 구성된다. T1-T6은 문제 분포를 의미하며, T1은 General, T2는 Paired Block, T3은 Subassembly, T4는 Mixed Relation, T5는 Heavy Workload, T6은 High Utilization이다.

Table 1에서 제안 방법은 모든 생성 데이터 조건에서 가장 낮은 Makespan RPD를 달성하였다. Proposed의 RPD가 모든 생성 데이터 조건에서 0.00으로 표시된 것은 제안 방법이 각 인스턴스에서 비교 방법 중 최저 Makespan을 달성하여 식 (21)에 따른 상대 편차가 0으로 계산되었기 때문이다.

Table 2 Generated-data evaluation: Violation Count

Scale	Type	SPT	MSF	LPT	Proposed
Small	T1	9.00	6.14	7.00	0.57
	T2	10.71	8.71	4.57	1.00
	T3	9.29	7.71	6.29	0.86
	T4	8.57	9.14	5.57	0.71
	T5	9.43	6.71	5.43	1.14
	T6	7.00	4.71	3.86	1.00
Medium	T1	29.33	13.17	10.33	1.50
	T2	27.00	23.17	9.00	1.50
	T3	21.50	12.33	8.67	1.67
	T4	28.50	21.33	6.00	0.50
	T5	24.67	9.17	5.83	0.67
	T6	27.33	13.83	10.67	1.33
Large	T1	48.67	20.00	14.83	2.17
	T2	46.00	31.50	12.00	2.17
	T3	40.50	20.83	13.17	2.00
	T4	40.00	33.67	10.17	2.67
	T5	43.17	14.50	9.50	1.67
	T6	44.67	15.83	6.67	1.83

Table 2에서도 제안 방법은 블록 수가 증가해도 제약 미준수 횟수를 낮게 유지하였다. 특히 Large 구간에서 SPT와 MSF의 제약 미준수 횟수는 크게 증가하는 반면, 제안 방법은 모든 분포에서 3건 이하를 유지하였다. 이는 제안 방법이 단순 우선순위 규칙보다 문제 규모와 분포 변화에 안정적으로 대응함을 보여준다.

4.3 실적 데이터 실험 결과

실적 데이터 실험은 실제 판넬라인 생산계획 사례를 대상으로 수행하였다. 평가 대상은 총 72개 블록으로 구성되며, 현업 수기 계획(Manual), MSDH, SPT, MSF, LPT, Proposed를 비교하였다.

Table 3은 실적 데이터의 종합 성능을 나타낸다. Fig. 3, Fig. 4, Fig. 5는 각각 Makespan RPD, 제약 미준수 횟수, 론지 부하 균형 결과를 시각화한 것이다.

Table 3의 Makespan RPD 열과 Fig. 3에서 Proposed는 RPD 0.00으로 가장 짧은 Makespan을 보였다. Manual은 RPD 18.30, MSDH는 RPD 18.17로 나타났다. MSDH는 현업 수기 계획의 착수일 구간을 유지하면서 마스킹 로직과 베이 할당 규칙을 적용한 방법이므로, Manual 대비 Makespan 측면에서는 소폭 개선되었다. 그러나 SPT, MSF, LPT와 같은 단순 휴리스틱은 Makespan을 더 크게 줄였음에도, 제약 미준수와 론지 부하 균형을 동시에 고려하는 데에는 한계가 있었다. Proposed는 학습된 정책을 통

Table 3 Actual-data evaluation summary

Method	Makespan RPD	Violation Count	Longi Balance
Manual	18.30	71	145
MSDH	18.17	36	27
SPT	3.27	10	17
MSF	2.69	6	27
LPT	1.15	3	43
Proposed	0.00	0	15

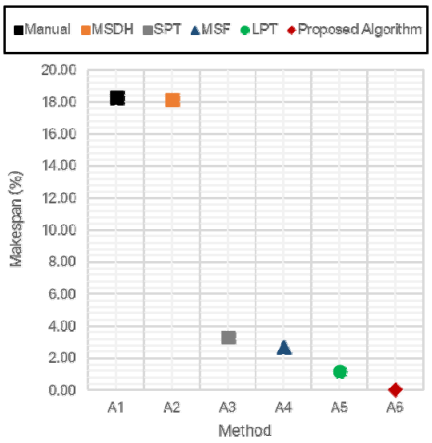

Fig. 3 Makespan RPD comparison on actual data

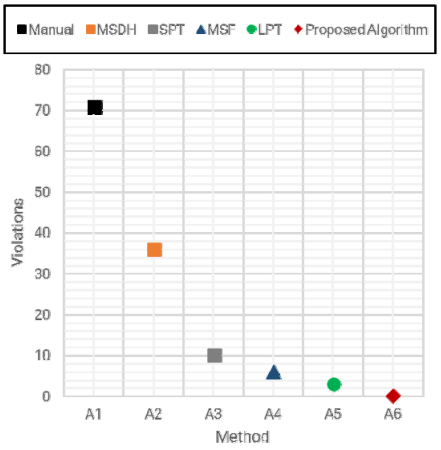

Fig. 4 Constraint violation comparison on actual data

해 후보 블록을 선택함으로써 Makespan 측면에서도 가장 우수한 결과를 달성하였다.

Table 3의 Violation Count 열과 Fig. 4에서 제약 미준수 횟수를 비교하였다. Manual은 71건의 제약 미준수를 보였으나, MSDH는 36건으로 감소하였다. 이는 마스킹 로직만 적용해도 현업 수기 계획 대비 제약 미준수 횟수를 크게 줄일 수 있음을 의미한다. 그러나 MSDH는 여전히 36건의 미준수가 남아 있으므로, 마스킹 이후 후보 중 어떤 블록을 선택하는지가 중요하다. Proposed는 학습 기반 정책을 적용하여 제약 미준수 0건을 달성하였으며, 이는 단순 마스킹을 넘어 학습된 후보 선택 정책이 추가적인 개선 효과를 제공함을 보여준다.

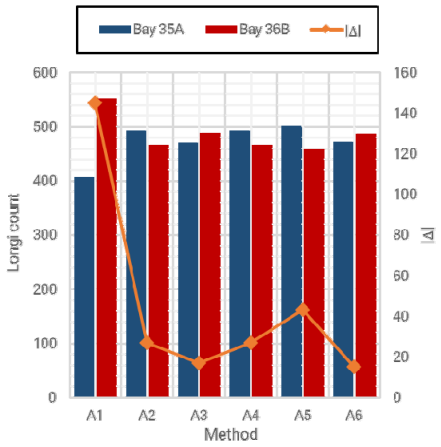

Fig. 5 Longi workload balance comparison on actual data

Table 3의 Longi Balance 열과 Fig. 5는 론지 부하 균형 결과를 나타낸다. Manual은 $\Delta = 145$로 가장 큰 부하 불균형을 보였으나, MSDH는 $\Delta = 27$로 크게 개선되었다. 이는 마스킹 로직과 베이 할당 규칙이 론지 작업량 분산에도 효과가 있음을 보여준다. Proposed는 $\Delta = 15$로 가장 낮은 부하 편차를 보였으며, LPT는 Makespan과 제약 미준수 측면에서는 비교적 우수했지만 $\Delta = 43$으로 Proposed보다 부하 균형이 좋지 않았다.

Manual과 MSDH의 비교는 마스킹 로직과 베이 할당 규칙만으로도 RPD, 제약 미준수, 론지 부하 편차가 개선됨을 보여준다. Proposed는 여기에 학습 기반 후보 선택 정책을 결합하여 세 지표에서 가장 우수한 결과를 달성하였다.

5. 결 론

본 연구에서는 조선소 판넬라인의 블록 투입 순서와 베이 할당을 함께 고려하는 학습형 스케줄링 방법론을 제안하였다. 대상 문제는 전반부 직렬 공정과 후반부 베이 공정이 결합한 Hybrid Flow Shop 구조를 가지며, 생산 용량, 작업 순서, 연속 배치, 베이 할당 등 복수의 운영 제약을 동시에 고려해야 한다.

이를 위해 본 연구는 판넬라인 스케줄링을 DES 기반 순차 의사결정 문제로 정식화하고, 현재 상태와 제약조건을 반영하여 유효 후보 집합을 생성하는 계층적 액션 마스킹을 적용하였다. 또한 포인터 네트워크 기반 정책과 Self-Labeling 학습을 결합하여, 정답 스케줄 데이터 없이도 제약 미준수, Makespan, 론지 부하 균형을 함께 고려하는 정책을 학습하였다.

실험 결과, 제안 방법은 실적 데이터에서 현업 수기 계획 및 휴리스틱 대비 Makespan, 제약 미준수 횟수, 론지 부하 편차를 모두 개선하였다. 또한 생성 데이터 실험에서도 문제 규모와 분포가 변화해도 안정적인 성능을 보였으며, 이를 통해 제안 방법의 일반화 가능성을 확인하였다.

향후 연구에서는 제약 우선순위와 Self-Labeling 학습 전략을 고도화하고, 긴급 블록 투입, 설비 지연, 재스케줄링과 같은 동적 생산 상황으로 확장하고자 한다.

후 기

본 연구는 다음의 지원을 받아 수행되었습니다.
(1) 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원 (RS-2025-00555741)
(2) 산업통상자원부의 재원으로 기술혁신사업의 지원(RS-2025-02372996)
(3) 산업통상자원부와 한국산업기술진흥원(KIAT)의 지원 (국제공동기술개발사업 0022929, 산업혁신인재성장지원사업 RS-2025-02263945, RS-2023-KI002688)
(4) 해양수산부 재원으로 해양수산과학기술진흥원의 지원 (첨단선박 블루테크 인재양성 RS-2025-02221147)

References

Bello, I., Pham, H., Le, Q.V., Norouzi, M. and Bengio, S., 2016. *Neural combinatorial optimization with reinforcement learning.* arXiv preprint arXiv:1611.09940.

Cho, Y.I., Nam, S.H., Cho, K.Y., Yoon, H.C. and Woo, J.H., 2022. Minimize makespan of permutation flowshop using pointer network. *Journal of Computational Design and Engineering,* 9(1), pp.51-67.

Corsini, A., Porrello, A., Calderara, S. and Dell'Amico, M., 2024. Self-labeling the job shop scheduling problem. *Advances in Neural Information Processing Systems*, 37.

Garey, M.R., Johnson, D.S. and Sethi, R., 1976. The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research,* 1(2), pp.117-129.

Huang, S. and Ontañón, S., 2022. A closer look at invalid action masking in policy gradient algorithms. *Proceedings of the International Florida Artificial Intelligence Research Society Conference,* 35.

Johnson, S.M., 1954. Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly,* 1(1), pp.61-68.

Koh, S.G., 1996. A production schedule with genetic algorithm in block assembly shop. *Korean Management Science Review*, 13(1), pp.1-12.

Kwak, D.H., Cho, K.Y., Ryu, C. and Woo, J.H., 2025. Scheduling optimization of hull block assembly line using constraint programming and discrete-event simulation. *International Journal of Naval Architecture and Ocean Engineering,* 17, 100675.

Lee, K., Shin, J.G. and Ryu, C., 2009. Development of simulation-based production execution system in a shipyard: A case study for a panel block assembly shop. *Production Planning & Control,* 20(8), pp.750-768.

Li, J., Lin, P., Wu, X., Song, D., Yang, B. and Zhou, L., 2024. Scheduling optimization of ship plane block flow line considering dual resource constraints. *Scientific Reports*, 14, 30765.

Naderi, B., Ruiz, R. and Roshanaei, V., 2023. Mixed-integer programming vs. constraint programming for shop scheduling problems: New results and outlook. *INFORMS Journal on Computing,* 35(4), pp.817-843.

Nawaz, M., Enscore Jr., E.E. and Ham, I., 1983. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. Omega, 11(1), pp.91-95.

Pan, Z., Wang, L., Wang, J. and Lu, J., 2023. Deep reinforcement learning based optimization algorithm for permutation flow-shop scheduling. *IEEE Transactions on Emerging Topics in Computational Intelligence,* 7(4), pp.983-994.

Ren, J., Ye, C. and Yang, F., 2021. Solving flow-shop scheduling problem with a reinforcement learning algorithm that generalizes the value function with neural network. *Alexandria Engineering Journal*, 60(3), pp.2787-2800.

Ruiz, R. and Vázquez-Rodríguez, J.A., 2010. The hybrid flow shop scheduling problem. *European Journal of Operational Research,* 205(1), pp.1-18.

Vinyals, O., Fortunato, M. and Jaitly, N., 2015. Pointer networks. *In Advances in Neural Information Processing Systems,* 28.

Wang, C., Mao, P.X., Mao, Y.S. and Shin, J.G., 2016. Research on scheduling and optimization under uncertain conditions in panel block production line in shipbuilding. *International Journal of Naval Architecture and Ocean Engineering,* 8(4), pp.398-408.

Xu, K., Ye, C., Gong, H. and Sun, W., 2024. Reinforcement learning-based multi-objective of two-stage blocking hybrid flow shop scheduling problem. *Processes,* 12(1), 51.

Yang, Z., Liu, C., Zhang, S. and Shi, J., 2019. A multi-objective memetic algorithm for a fuzzy parallel blocking flow shop scheduling problem of panel block assembly in shipbuilding. *Journal of Ship Production and Design,* 35(2), pp.170-181.

Zahavy, T., Haroush, M., Merlis, N., Mankowitz, D. J. and Mannor, S., 2018. Learn what not to learn: Action elimination with deep reinforcement learning. *In Advances in Neural Information Processing Systems,* 31, pp.3562-3573.

Zhou, T., Luo, L., He, Y., Fan, Z. and Ji, S., 2023. Solving panel block assembly line scheduling problem via a novel deep reinforcement learning approach. *Applied Sciences*, 13(14), 8483.

Authorship Contribution Statement

Hyunjin Oh: Conceptualization, Data curation, Methodology, Validation, Writing – original draft; **Youngin Cho:** Conceptualization, Formal analysis, Methodology; **Yeongchan Han:** Investigation, Methodology; **Jaeho Choi:** Supervision, Validation; **Junhyeon Kim:** Supervision, Validation; **Jonghun Woo:** Funding acquisition, Supervision, Writing – review & editing.


오 현 진


조 영 인


한 영 찬


최 재 호


김 준 현


우 종 훈