Journal of the Korean Institute of Industrial Engineers
Vol. 51, No. 4, pp. 288-298, August 2025.
ISSN 1225-0988 | EISSN 2234-6457
https://doi.org/10.7232/JKIIE.2025.51.4.288

반도체 클러스터 혼류 공정 최적화를 위한 멀티에이전트 심층 강화학습 기반 로봇 협업 스케줄링

차민성[1] · 최종원[2] · 김성범[1†]

[1]고려대학교 산업경영공학과 / [2]삼성전자

Multi-Agent Deep Reinforcement Learning-Based Collaborative Robot Scheduling for Concurrent Processing Optimization in Semiconductor Cluster Tools

Minsung Cha[1] · Jongwon Choi[2] · Seoung Bum Kim[1]

[1]Department of Industrial and Management Engineering, Korea University
[2]Samsung Electronics

Scheduling semiconductor cluster tools is particularly challenging in concurrent processing environments, where wafers with diverse characteristics are processed simultaneously. Effective cooperative scheduling between the vacuum transfer module (VTM) robots and atmospheric transfer module (ATM) robots is essential because it directly impacts overall system throughput. However, traditional rule-based scheduling methods often degrade in performance under dynamic and complex conditions. To address this issue, this study proposes a multi-agent deep reinforcement learning (MADRL) framework in which the VTM and ATM robots are modeled as independent agents that jointly learn collaborative scheduling strategies without relying on expert-designed rules. Experimental results show that the learned scheduling strategies consistently outperform rule-based baselines-including heuristics commonly used in practice-in terms of throughput. Moreover, the proposed framework exhibits strong generalization capabilities under increased process complexity. These results highlight the potential of the MADRL-based approach as a rule-free, adaptive, and generalizable solution for optimizing scheduling in complex semiconductor manufacturing systems.

Keywords: Multi-Agent Deep Reinforcement Learning, Cluster Tool Scheduling, Concurrent Processing, Semiconductor Manufacturing, Throughput Optimization

1. 서 론

반도체 산업은 공정의 미세화가 가속화되고, 다양한 제품을 소량으로 생산하는 체제로 빠르게 전환되고 있다. 이로 인해 생산성 향상과 장비 활용률 극대화에 대한 요구가 지속적으로 증가하고 있으며, 다양한 공정 단계를 하나의 장비 내에서 통합 수행할 수 있는 클러스터 장비(cluster tools)는 첨단 반도체 제조 현장에서 핵심적인 역할을 수행하고 있다. 클러스터 장비는 높은 투자 비용이 수반되는 설비이기 때문에, 그 효율적인 운영은 상당히 중요하다(Pan *et al.*, 2017; Lee, 2008). 특히, 서로 다른 특성을 가진 다양한 종류의 웨이퍼가 혼합 투입되는 혼류 공정(concurrent processing) 환경에서는 웨이퍼마다 공정 시간, 이송 경로, 전환 시간이 상이하여 정교한 스케줄링 전략이 요구된다(Kim and Lee, 2024).

† 연락저자 : 김성범 교수, 02841 서울특별시 성북구 안암로 145 고려대학교 산업경영공학부, Tel: 02-3290-3397, Fax: 02-929-5888, E-mail : sbkim1@korea.ac.kr
2025년 4월 21일 접수; 2025년 5월 6일 수정본 접수; 2025년 5월 16일 게재 확정.

기존에는 전문가가 설계한 휴리스틱 규칙에 기반한 스케줄링 방식이 일반적으로 사용되어 왔다(Lee *et al.*, 2015; Lee and Lee, 2021). 이러한 규칙 기반 방식은 구조가 단순하고 구현이 용이하다는 장점이 있으나, 장비 구성이나 공정 조건이 변할 경우 유연하게 대응하기 어려우며, 새로운 조건에 적합한 규칙을 새롭게 설계하고 검증해야 한다는 점에서 유지보수 비용이 높고 전문가 의존도가 크다는 한계를 갖는다(Suerich and McIlroy, 2022). 이러한 문제를 해결하기 위한 대안으로, 최근에는 인공지능 기반 스케줄링 기법, 특히 심층 강화학습(deep reinforcement learning, DRL)을 활용한 접근이 활발히 연구되고 있다. DRL은 환경과 상호작용을 통해 보상을 최대화하는 행동 정책을 스스로 학습할 수 있어, 전문가 개입 없이도 동적 시스템에 적응할 수 있는 장점이 있다(Mnih *et al.*, 2015; Mnih *et al.*, 2016). DRL을 활용한 기존 연구들은 대부분 진공 영역의 진공 이송 모듈(vacuum transfer module, VTM)과 챔버 간 단순한 이송만을 고려하거나, 실제 복잡한 장비 구조나 다양한 공정 조건을 충분히 반영하지 못한 한계가 있다. 예를 들어, 멀티에이전트 심층 강화학습(multi-agent deep reinforcement learning, MADRL) 구조를 적용하여 condition-based chamber cleaning 환경에서 강건한 스케줄링을 수행할 수 있는 연구의 경우, 단일 웨이퍼 유형만을 대상으로 하여 복잡한 혼류 공정 환경에 적용하기에는 제한이 있다(Hong and Lee, 2018). 또한, 심층 Q-네트워크(deep Q-network, DQN)과 adaptive search를 결합하여 혼류 공정 환경에서 유연한 스케줄링을 구현한 연구에서도, 에이전트 구조와 학습 방식이 VTM 또는 챔버 단위로 고정되어 있어 대기압 이송 모듈(atmospheric transfer module, ATM) 등 다른 모듈과 협업 스케줄링까지 확장하는 데에는 한계가 존재한다(Kim and Lee, 2024).

본 연구는 이러한 기존 연구의 한계를 극복하고자, 진공 및 대기압 이송 모듈을 모두 포함한 현실적인 클러스터 장비 시뮬레이터를 구현하고, VTM 로봇과 ATM 로봇 각각에 대해 독립적인 에이전트를 구성하여 멀티에이전트 심층 Q-러닝(multi-agent deep Q-network, MADQN)을 기반으로 한 협업 스케줄링 학습 방식을 제안한다. 제안하는 방식은 전문가 설계 없이 상태와 보상 기반의 자율 학습을 통해 복잡한 장비 구조와 다양한 공정 조건을 유연하게 반영할 수 있으며, 상황에 따라 동적인 협업 전략을 도출하는 것이 가능하다. 본 연구에서는 두 가지 혼류 공정 환경-(1) 기본 환경, (2) 클러스터 장비 내 모듈 간 사양 차이를 반영한 환경-을 구성하였다. 각 환경은 혼합 투입되는 웨이퍼 종류의 수에 따라 난이도가 조절되며, 최소 두 종류에서 최대 네 종류의 웨이퍼가 혼합 투입되는 시나리오를 포함한다. 실험 결과, 제안한 방식은 학습한 환경과 동일한 환경에서 기존 규칙 기반 방식 대비 일관되게 높은 단위 시간당 처리량을 달성하였다. 또한, 두 종류의 웨이퍼가 등장하는 상대적으로 간단한 환경을 학습한 에이전트의 정책을 세 종류 또는 네 종류의 웨이퍼가 혼합된 보다 복잡한 환경에 적용한 경우에도 기존 규칙 기반 방식 대비 일관적으로 높은 성능을 보여, 제안 방법의 일반화 능력을 확인할 수 있었다. 본 연구의 주요 기여점은 다음과 같다.

- VTM 및 ATM 로봇 간 협업이 가능한 반도체 클러스터 장비 시뮬레이터를 구현하고, 이를 통해 실제 혼류 공정 환경에서 발생하는 로봇 간 협업 문제를 정밀하게 모델링하였다. 제안한 시뮬레이터는 OpenAI Gym 환경을 기반으로 하여 직접 제작하였으며, 다양한 심층 강화학습 알고리즘을 활용한 비교 실험 및 후속 연구로의 확장이 가능하며, 실제 장비 구성을 충실히 반영하였다는 점에서 의의가 있다.
- MADQN 구조를 활용하여 전문가 사전 지식 없이 로봇 간 협업 정책을 자율적으로 학습할 수 있는 스케줄링 기법을 제안하였다. 본 제안 방법론은 규칙 기반 접근 방식에서 요구되는 높은 비용과 낮은 유연성의 한계를 효과적으로 극복할 수 있으며, 환경 변화에도 능동적으로 적응 가능한 정책을 도출할 수 있다는 장점을 지닌다.
- 다양한 혼류 공정 환경에서의 실험을 통해 제안 방법론의 우수한 성능과 일반화 가능성을 입증하였다. 혼합 투입되는 웨이퍼 종류 수에 따라 난이도를 달리한 두 가지 환경에서 실험을 수행한 결과, 제안 방법론은 기존 규칙 기반 방식 대비 일관되게 높은 처리량을 기록하였다. 특히, 단순한 시나리오에서 학습한 정책이 복잡한 시나리오에서도 성능 저하 없이 적용되는 높은 강건성을 보여주어 그 실용성을 뒷받침하였다.

본 논문은 다음과 같이 구성된다. 제2장에서는 반도체 클러스터 장비 스케줄링 문제에 대한 기존 연구들을 정리하고, 기존 전통적 접근과 DRL 기반 접근의 한계 및 본 연구와의 차별점을 고찰한다. 제3장에서는 본 연구에서 구현한 클러스터 장비 시뮬레이터와 실험 환경, 그리고 MADQN 기반 스케줄링 에이전트의 구조를 설명한다. 제4장에서는 두 가지 혼류 공정 환경을 기반으로 수행한 실험 조건 및 결과를 제시하고, 제안 기법의 성능과 일반화 능력을 분석한다. 마지막으로 제5장에서는 연구 결과를 요약하고, 결론과 향후 연구 방향을 제안한다.

2. 관련 연구

2.1 전통적 접근 방식

반도체 클러스터 장비 스케줄링 문제를 해결하기 위해, 다양한 이론적 모델과 실용적 알고리즘이 오랜 기간 동안 연구되었다. 초기에는 주로 Petri net을 활용한 연구가 주를 이루었다. Wu *et al.*(2010)은 원자층 증착 공정(atomic layer deposition, ALD) 공정과 같이 재방문이 필요한 공정에서 단일 로봇 클러스터 장비의 deadlock 문제를 해결하기 위해 Petri net을 기반으로 한 효율적인 모델과 분석적 스케줄링 기법을 제안하였으나, 불확실성 요소를 고려하지 않았다는 점과 단일 로봇 클러스터

장비에 국한된다는 점에서 한계를 가진다. Qiao *et al.*(2015)은 다회 재방문 공정을 갖는 듀얼 암 클러스터 장비의 사이클 타임을 분석하기 위한 Petri net 기반의 동적 모델과 수식에 기반한 계산 방법을 제시하였다. 하지만 해당 연구는 환경 변화에 따른 정책 적응성을 고려하지 않아 현실적인 혼류 공정 시나리오에는 적용에 한계가 있다.

혼합 정수 계획법(mixed integer programming, MIP)을 기반으로 한 접근은 다양한 공정 흐름과 제약 조건을 통합적으로 고려하여, 복잡한 클러스터 장비 스케줄링 문제를 수리적으로 최적화하려는 시도이다. Jung and Lee(2011)는 timed Petri net(TPN)을 활용해 다양한 조건을 동시에 반영한 클러스터 장비 스케줄링 문제를 모델링하고, 자동으로 MIP 모델을 생성하여 주기적 스케줄을 도출하는 방법을 제안하였다. 하지만 이 방식은 환경의 시간 요소가 고정적이라는 전제 하에 구성되었고, 스케줄링 조건이 변하면 MIP 모델을 재정의하거나 solver 기반으로 다시 계산해야 한다. 따라서 해당 방법론은 실시간 적용성이 낮고, 환경 변화에 유연한 대응이 어렵다는 한계를 가진다. Li and Yang(2025)은 residency time 제약이 존재하는 단일 로봇 클러스터 장비에서 최대 4종의 웨이퍼를 주기적으로 처리하는 순환 스케줄링 문제를 다루었으며, 이를 해결하기 위한 MIP 기반 모델을 제안하였다. 해당 연구는 이전 연구들과 달리 공유 챔버 수에 대한 제한 없이 비대칭 구조를 허용하여 보다 다양하고 일반화된 시나리오에 대한 해 도출 가능성을 보였다. 그러나 해당 연구는 여전히 사전 정의된 공정 흐름과 최소 생산 집합(minimal production set)에 의존하고 있어, 클러스터 장비 구조 등 환경 조건이 바뀔 경우 모델을 재정의해야 한다는 한계를 가진다. 또한, MIP는 NP-hard 문제로, 문제 크기가 커질수록 해를 찾는 데 필요한 계산 시간이 급격히 증가한다(Zhang *et al.*, 2023).

순환(cyclic) 스케줄링 기반 접근은 반복적인 장비 운영 환경에서 로봇 작업 순서를 주기적으로 반복하여 클러스터 장비의 처리량을 최적화하려는 시도로 발전해왔다. Lee *et al.*(2015)은 두 종류의 웨이퍼가 하나의 공정 챔버를 공유하는 혼류 공정 환경에서 alternating backward / swap 시퀀스를 제안하고, 특정 조건 하에서 해당 시퀀스가 사이클 타임을 최소화함을 이론적으로 증명하였다. 또한, 실험을 통해 제안된 시퀀스가 대부분의 실제 공정 조건에서 최적, 또는 근사 최적의 성능을 보임을 증명하였다. 다만 해당 연구에서는 여러 웨이퍼가 복수의 공정 챔버를 공유하거나, 복수 웨이퍼 유형이 동시에 처리되는 등의 조건을 고려하지 않았으며, 제안된 시퀀스 역시 고정된 작업 순서 기반이므로 환경 변화에 대한 적응성은 제한적이라는 한계를 가진다.

최근 소량 생산 체제로의 전환이 가속화되며, 일정 주기를 전제로 하는 순환 스케줄링이 적합하지 않은 사례가 증가하고 있다. 이에 따라 최근 연구에서는 보다 유연한 비순환(noncyclic) 스케줄링에 대한 관심이 확대되고 있다. Yan *et al.*(2020)은 다중 로봇 및 복수 클러스터 장비로 구성된 환경에서 비순환 스케줄링 문제를 다루며, Pareto 최적 기반 동적 계획법을 통한 효율적인 정책 공간 축소로 계산 효율성을 개선하였다. 해당 연구는 실험 결과를 통해 다양한 상황에서 성능 우수성을 입증하였으나, 여전히 규칙 기반의 고정된 정책에 의존하므로 동적 환경에 대한 실시간 적응성과 자율성에는 한계가 있다.

2.2 DRL 기반 연구 동향

최근에는 DRL을 이용하여 반도체 클러스터 장비 스케줄링 문제를 해결하려는 연구가 등장하고 있다. DRL 에이전트는 환경과 상호작용을 통해 보상을 최대화하는 방향으로 정책을 스스로 학습하며, 전문가 규칙 없이도 복잡한 조건에 유연하게 적응하여 보다 나은 의사결정을 할 수 있다(Mnih *et al.*, 2015; Mnih *et al.*, 2016).

Hong and Lee(2018)는 기존 k-periodic cleaning의 한계를 극복하기 위해, 챔버 오염 상태를 반영한 condition-based cleaning 개념을 제안하고, condition-based cleaning이 적용된 환경에서 장비들을 스케줄링하기 위한 멀티에이전트 강화학습(multi-agent reinforcement learning, MARL) 모델을 제안하였다. 실험 결과, 제안한 MARL 방식은 기존 규칙 기반 방식보다 더 높은 처리량과 cleaning 효율성을 달성하였다. 다만 해당 연구는 한 종류의 웨이퍼로만 구성된 단일 공정으로 실제 생산 환경에서 발생할 수 있는 복잡한 혼류 공정을 반영하지 못한다는 한계가 있다. Lee and Lee(2021)는 장비 내부의 웨이퍼 이송 로봇 동작을 최적화하기 위해 deep Q-network(DQN) 기반 스케줄러를 제안하였다. 해당 연구에서는 특히 챔버 간 처리 시간의 불균일성과 deadlock 상황을 고려하여 학습을 진행하였으며, action masking과 reward 설계 개선을 통해 학습의 안정성을 확보하였다. 또한 소규모 환경에서 사전 학습된 모델을 대규모 환경으로 전이학습 시켜 생산 환경 변화에 유연하게 적응할 수 있음을 실험적으로 입증하였다. 다만, 해당 연구는 단일 에이전트 구조로, 복수 로봇 또는 복잡한 공정 경로를 고려하지 못한다는 한계를 가진다. Kim and Lee(2024)는 비순환 스케줄링 환경에서 두 종류의 웨이퍼를 동시에 처리하는 싱글 암 클러스터 장비를 대상으로 로봇 작업 순서와 웨이퍼 투입 순서를 함께 최적화하는 새로운 DRL 기반 스케줄링 기법을 제안하였다. 제안된 방식은 기존 방식 대비 높은 처리량을 일관되게 달성하였으며, 다양한 처리 시간 조합에서도 뛰어난 일반화 성능을 보였다. 다만 해당 연구는 하나의 로봇 팔만을 사용하는 단일 클러스터 장비에 한정되어 있으며, 다중 로봇 및 확장된 환경으로의 적용 가능성은 향후 연구 과제로 남아 있는 한계가 있다.

본 연구는 기존 연구들의 한계를 보완하고자, 두 대의 로봇(VTM 로봇 및 ATM 로봇)을 대상으로 한 환경을 구성하여 복수 로봇의 스케줄링 문제를 해결하고자 했다. 이를 위해 두 종류의 로봇을 동시에 제어할 수 있도록 MADQN 방식을 채택

하였다. 각 로봇은 두 개의 팔을 가진 듀얼 암 구조로, 독립적인 DQN 에이전트를 통해 행동을 개별적으로 결정하되, 전체 시스템 상태를 공유함으로써 협업 가능한 정책을 효과적으로 학습한다. 또한, 서로 다른 종류의 웨이퍼가 한 작업 순서 내에서 처리되는 혼류 공정 환경을 구성하였으며, 다양한 조건을 조합하여 실험을 수행하였다. 그 결과, 학습된 정책은 기존의 규칙 기반 정책보다 우수한 성능을 보였으며, 환경 변화에도 일관된 성능을 유지하는 일반화 능력을 갖추고 있음을 확인하였다.

3. 방법론

3.1 클러스터 장비 시뮬레이터 개요

본 연구에서는 실제 반도체 클러스터 장비의 동작 특성을 반영한 시뮬레이터를 개발하여 실험 환경으로 사용하였다. 개발 언어로는 Python을 사용하였다. 시뮬레이터 개발은 강화학습 환경 개발 프레임워크인 OpenAI Gym(Gymnasium)을 기반으로 하였으며, Gym의 사용자 정의 환경(custom environment)으로 등록하여 다양한 알고리즘 실험이 가능하도록 하였다. 시뮬레이터의 구현 코드는 GitHub 공개 저장소(https://github.com/djpanda1217/clustertool_simulator)에서 확인할 수 있다.

장비 구성 측면에서, 시뮬레이터에 구현된 반도체 클러스터 장비는 진공 영역(vacuum area)과 대기압 영역(atmospheric area)으로 구성되어 있다. 반도체 클러스터 장비는 VTM 로봇, ATM 로봇, 정렬기(aligner), 공정 챔버(chamber), 입력 로드락(load lock entry, LL entry), 출력 로드락(LL exit), 입력 로드 포트(load port IN, LP IN), 출력 로드 포트(LP OUT) 등 다양한 구성 요소로 이루어진다. LP IN과 LP OUT은 반도체 클러스터 장비 외부에서 내부, 혹은 내부에서 외부로 웨이퍼가 이동할 때 웨이퍼가 통과하는 모듈이다. LL Entry, LL Exit은 웨이퍼가 반도체 클러스터 장비 내에서 대기압 영역에서 진공 영역, 또는 진공 영역에서 대기압 영역으로 이동할 때 통과해야 하는 모듈로, 장비 내에서 진공과 대기압 상태를 구분하여 관리할 수 있도록 하는 역할을 한다. Aligner는 웨이퍼의 방향 정렬을 수행하며, Chamber는 실질적인 공정이 이루어지는 모듈로 총 6개가 구성된다. VTM 및 ATM 로봇은 각각 진공과 대기압 환경에서 독립적으로 동작하며, 각 로봇은 두 개의 arm(left, right)을 가진다. <Figure 1>은 시뮬레이터에 구현된 클러스터 장비의 전체 구조를 도식화한 것이다. LP IN에서 시작된 웨이퍼는 대기압 영역의 ATM 로봇에 의해 Aligner로 이동되며, 이후 LL Entry을 통해 진공 영역의 VTM 로봇에 의해 Chamber로 운반된다. 공정 완료 후 Aligner 방문을 제외한 반대 경로를 통해 장비 외부로 반출되며, 각 모듈을 거치는 웨이퍼의 흐름은 <Figure 1>의 알파벳 경로를 기준으로 정의된다.

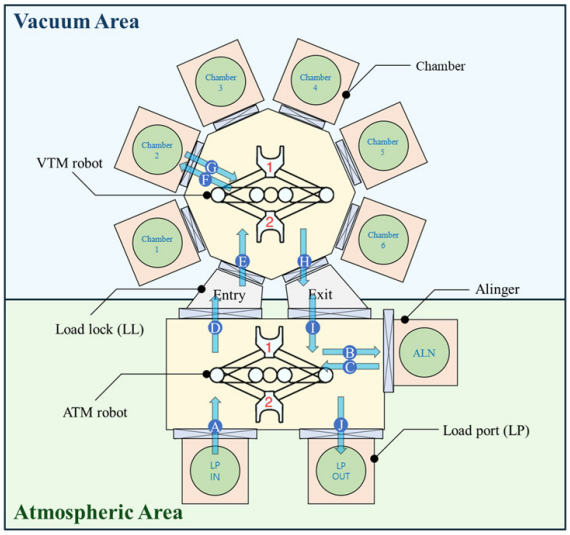

Figure 1. Semiconductor Cluster Tool Environment and Wafer Flow

시뮬레이터는 multi-agent Markov decision process(MMDP)로 정의되며, 상태 공간(state space, S), 행동 공간(action space, A), 상태 전이 함수(state transition function, P), 보상 함수(reward function, R), 시간 할인 요소(discount factor, γ)로 구성된다. 본 문제의 MMDP는 다음과 같이 형식적으로 표현할 수 있다.

$$MMDP = \langle S, A = A_{VTR} \times A_{ATR}, P, R, \gamma \rangle$$

여기서 상태 공간 S는 총 45차원의 이산(discrete) 상태 벡터이며, 네 가지 하위 구성 요소로 나뉜다. 첫째, 웨이퍼 존재 여부는 $\{0, 1\}^{15}$이진 벡터로 표현되며, 각 모듈의 웨이퍼 보유 여부를 나타낸다. 이는 총 15개의 모듈에 대해 정의되며, 해당 모듈들은 LL Entry, LL Exit, VTR 로봇의 양쪽 arm, 6개의 챔버, ATR 로봇의 양쪽 arm, Aligner, LP IN, LP OUT이다. 둘째, 웨이퍼의 공정 상태는 $\{0, 1, 2\}^{15}$ 정수 벡터로 나타내며, 웨이퍼 존재 여부 상태와 같은 모듈을 대상으로 정의된다. 0은 웨이퍼가 존재하지 않거나, 존재하더라도 아직 공정을 시작하지 않았거나, 공정이 진행 중인 상태를 모두 포함한다. 1은 aligner에서 정렬이 완료된 상태를, 2는 챔버에서 공정이 완료된 상태를 의미한다. 셋째, 각 모듈의 사용 가능 여부는 $\{0, 1\}^{13}$ 이진 벡터로, VTM 로봇의 양쪽 arm, chamber 6개, ATM 로봇의 양쪽 arm, aligner, LP IN, LP OUT 모듈을 대상으로 정의된다. 해당 모듈의 잔여 처리 시간이 0이면 1, 아니면 0으로 설정된다. 넷째, LL Entry 및 LL Exit 모듈의 진공/대기압 상태는 $\{0, 1\}^2$ 이진 벡터로, 0은 진공 상태, 1은 대기압 상태를 의미한다. 이를 종합하면, 상태 공간은 다음과 같이 표현된다.

$$S \subseteq \{0, 1\}^{15} \times \{0, 1, 2\}^{15} \times \{0, 1\}^{13} \times \{0, 1\}^2$$

행동 공간 A는 두 에이전트의 이산 행동의 곱집합으로 정의된다.

$$A_{VTR} = \{0, 1, ..., 28\},\ A_{ATR} = \{0, 1, ..., 12\}$$

VTR 로봇은 총 29개의 행동을 가지며, 웨이퍼를 LL Entry에서 arm으로 이동, arm에서 chamber로 이동, chamber에서 arm으로 이동, arm에서 LL Exit으로 이동시키는 동작들과, 아무 행동도 취하지 않는 NO_ACTION이 포함된다. ATR 로봇은 13개의 행동을 가지며, 웨이퍼를 LP IN에서 arm으로 이동, arm과 aligner 사이에서 이동, arm과 LL Entry/Exit, LP IN/OUT 사이에서 이동시키는 동작과, 마찬가지로 NO_ACTION을 포함한다. 각 에이전트는 매 스텝에서 하나의 행동을 선택하며, 유효하지 않은 행동은 마스킹을 통해 제한된다.

상태 전이 함수 $P : S \times A \rightarrow S$는 현재 상태와 두 에이전트의 행동에 따라 결정되며, 본 시뮬레이터에서는 이를 완전한 결정론적(deterministic) 방식으로 구현하였다. 즉, 동일한 상태와 동일한 행동이 주어질 경우 항상 동일한 다음 상태가 반환된다. 로봇의 이동, 모듈의 처리 시간 업데이트, 웨이퍼 이송 및 공정 완료 여부는 일관된 규칙에 따라 계산된다.

보상함수 $R : S \times A \rightarrow \mathbb{R}$ 은 매 스텝마다 에이전트의 행동 및 시스템 처리 성과에 기반하여 보상 값을 반환한다. 두 에이전트가 모두 유효한 행동을 수행한 경우에는 +0.2, 한 에이전트만 유효한 행동을 수행한 경우에는 +0.1의 보상이 주어진다. 반대로, 두 에이전트 모두 NO_ACTION을 선택할 경우에는 -10의 페널티가 부여된다. 웨이퍼가 성공적으로 처리 완료될 경우 +1의 보상이 주어지며, 최대 처리 개수에 도달하면 +100의 보상이 추가로 주어지고 시뮬레이션이 종료된다. 이러한 보상 설계는 에이전트가 협력적이면서도 생산성을 극대화하는 방향으로 학습하도록 유도한다.

시간 할인 요소 γ는 0.99로 설정했으며, 미래 보상의 가치를 현재와 거의 동등하게 반영함으로써 에이전트가 단기적인 보상보다는 장기적인 생산성 향상에 초점을 맞출 수 있도록 설계되었다. 이는 웨이퍼가 여러 단계에 걸쳐 처리되는 본 시뮬레이션 환경의 특성과도 잘 부합한다.

본 연구에서는 다양한 웨이퍼 종류가 동시에 처리되는 혼류 공정 문제를 다룬다. 이 때, 각 챔버가 웨이퍼의 종류와 관계없이 모든 웨이퍼를 처리할 수 있도록 구성되어 있으며, 이로 인해, 한 챔버에서 서로 다른 종류의 웨이퍼가 연속적으로 처리될 경우, 서로 다른 공정 조건으로 인해 교차 오염이 발생할 수 있다. 이를 방지하기 위해, 챔버에서 웨이퍼 한 장을 처리하고 내보내면, 해당 챔버를 강제 청소하는 과정(force purge)이 반드시 수행되도록 하였다. 이 과정은 공정 지연의 원인이 되므로, VTM 로봇 에이전트와 ATM 로봇 에이전트는 force purge로 인한 시간 손실까지 반영하여 스케줄링 전략을 학습해야 한다.

3.2 실험 환경 구성

본 연구에서는 MADQN 기반 스케줄링 기법의 성능과 일반화 능력을 평가하기 위해 두 가지 유형의 혼류 공정 환경을 구성하였다. 또한, 한 처리 시퀀스에 등장하는 웨이퍼 종류의 수를 변화시켜 총 여섯 개의 시나리오를 설계하였다. 각 환경과 시나리오는 장비 사양 및 웨이퍼 특성의 차이를 반영하고 있으며, 장비 사양이 달라지거나 등장 웨이퍼의 종류 수가 많아짐에 따라 공정의 복잡성과 난이도가 상승하도록 구성되었다.

- 환경 A

  환경 A는 기본 설정으로, 클러스터 장비 내 모든 챔버가 동일한 처리 사양을 갖는다. 웨이퍼는 최대 네 종류까지 혼합 투입될 수 있으며, 각 웨이퍼가 챔버에서 처리되는 시간은 웨이퍼 A가 30 step, B가 90 step, C가 100 step, D가 50 step이다. 시나리오에 따라 투입되는 웨이퍼 종류 수는 2종, 3종, 4종으로 조정되며, 각 환경에서 웨이퍼 종류에 따른 등장 확률은 <Table 1>에 A_k제시되어 있다. <Table 1>에서 는 환경 A에서 k 종의 웨이퍼가 혼합 투입되는 시나리오를 의미한다. 이러한 구성은 실제 혼류 공정 상황에서 다양한 종류의 웨이퍼가 하나의 생산 배치에 함께 투입되는 현실적인 특성을 반영한 것이다. 실제로 Li and Yang (2025)은 다양한 수요에 대응하고 리드타임을 단축하기 위해 유사한 공정을 갖는 복수의 웨이퍼 종류를 동일 클러스터 장비 내에서 동시에 처리하는 방식이 점차 보편화되고 있다고 언급하였으며, 최대 4종의 웨이퍼가 등장하는 혼류 공정 환경을 대상으로 클러스터 장비의 최적화 문제를 다룬 사례를 제시하였다.

Table 1. Configuration of wafer type appearance probabilities in each scenario. As the number of wafer type(k) increases, the scheduling becomes more complex. A_k refers to the scenario in which k wafer types appear in environment A.

Scenario	Wafer A	Wafer B	Wafer C	Wafer D
A_2	0.67	0.33	-	-
A_3	0.5	0.3	0.2	-
A_4	0.4	0.3	0.2	0.1

- 환경 B

  환경 B는 클러스터 장비 내 챔버들이 서로 다른 처리 사양을 갖는 환경이다. 이는 하나의 장비 내에서도 챔버가 모델, 내부 구성 부품, 사용 이력 등에 따라 서로 다른 처리 성능을 보일 수 있다는 실무적 경험에 기반한 설정이다. 이러한 차이는 챔버 별로 서로 다른 생산 특성을 유도하며, 효과적인 스케줄링 전략 수립 시 중요한 고려 요소가 된다. 환경 B는 환경 A와 동일한 기본 구조를 유지하면서도, 각 웨이퍼가 특정 챔버에서 처리될 때 처리 시간에 일정 계수를 곱해주어 처리 성능 차이를 반영하였다. 챔버별 처리 시간 계수는 <Table 2>에 제시되어 있으며, B_k는 환경 B에서 k 종의 웨이퍼가 혼합 투입되는 시나리오를 의미한다.

Table 2. Configuration of processing time coefficients applied to each chamber for each wafer type in environment B. Wafer A shows no variation across chambers, while wafer B, C, and D show slower processing in early-numbered chambers. This setting reproduces heterogeneous chamber behavior that may occur in real-world equipment.

	Chamber 1	Chamber 2	Chamber 3	Chamber 4	Chamber 5	Chamber 6
Wafer A	1	1	1	1	1	1
Wafer B	1.3	1.3	1.15	1.15	1	1
Wafer C	1.5	1.5	1.25	1.25	1	1
Wafer D	1.5	1.5	1.25	1.25	1	1

이와 같이 총 두 가지 환경(A, B)에 대해 각각 2, 3, 4종의 웨이퍼가 혼합 투입되는 총 6개의 실험 시나리오를 정의하였다. 각 실험 환경은 MADQN 에이전트가 다양한 종류의 공정 복잡성에 얼마나 효과적으로 적응하여 학습하는지, 나아가 학습한 정책이 공정 조건이 변화하는 새로운 시나리오에서도 얼마나 강건하게 작동하는지를 검증하기 위한 기반이 된다.

3.3 VTM 로봇과 ATM 로봇 간 협업을 위한 MADQN 기반 에이전트 설계

본 연구는 반도체 클러스터 장비 내 VTM 로봇과 ATM 로봇 간의 협업 스케줄링 문제를 해결하기 위해 MADQN 구조를 설계하였다. 본 구조는 Tan(1993)에서 제안된 independent Q-learning(IQL) 방식에 기반하며, 두 로봇을 각각 독립적인 DQN 에이전트로 정의하였다. 다만, 본 연구에서는 두 에이전트가 동일한 상태(state) 및 보상(reward) 정보를 공유하되, 각기 다른 Q-Network을 통해 개별 행동을 학습하도록 구성하였다.

실제 반도체 장비에서는 VTM과 ATM 로봇이 서로 다른 영역에서 동작하며, 이동 경로와 모듈 구성 또한 물리적으로 완전히 분리되어 있다. 각 로봇은 자신이 담당하는 모듈 내에서 독립적으로 작동하고, 동시에 병렬적인 작업 수행이 가능하다. 이러한 물리적 구조를 반영하여 로봇을 개별 에이전트로 모델링한 것은, 시스템 운용 방식과의 정합성 측면에서 실용적인 접근으로 평가할 수 있다.

모든 로봇의 행동을 하나의 중앙집중형 에이전트를 통해 통합적으로 제어하는 방식도 고려할 수 있으나, 이러한 구조는 복잡성과 학습 효율성 측면에서 여러 한계를 내포한다. 중앙집중형 구조에서는 두 로봇의 행동 조합으로 인해 전체 행동 공간의 크기가 지수적으로 증가하며, DQN 기반의 정책 구조에서는 출력 노드 수 증가로 학습 및 탐색의 계산 비용이 급격히 상승한다(Rashid *et al.*, 2020; Lin *et al.*, 2022). 반면, 각 로봇이 공통 상태 정보를 기반으로 독립적인 정책을 학습하도록 구성한 분산형 구조에서는 이러한 문제를 완화할 수 있을 뿐만 아니라, 이후 로봇 구성의 변경이나 기능 확장 시 기존 정책

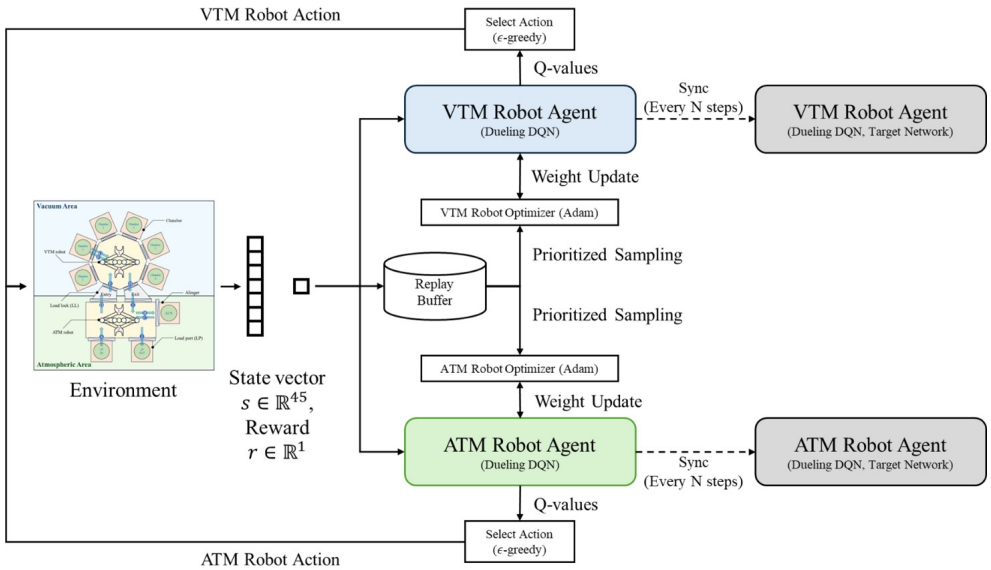

Figure 2. Overview of the proposed multi-agent learning architecture. The VTM and ATM robots are each controlled by a separate Duling DQN agent with distinct action spaces. The environment offers the same state vector and reward to both agents. Each agent independently computes Q-values, updates its parameters via backpropagation using a shared replay buffer. Actions for the VTM and ATM robots are selected independently using Q-value computed. Target networks are synchronized periodically every N step for stable learning.

을 모듈화하여 재활용할 수 있다는 실용적인 이점이 있다. 이에 따라, 본 연구에서는 하나의 중앙집중형 에이전트를 활용한 구조 대신 각 로봇이 공통의 상태 정보를 바탕으로 독립적으로 행동을 선택하고, 이를 통해 시스템 차원의 협업을 유도하는 MADQN 구조를 채택하였다. <Figure 2>는 제안된 MADQN 프레임워크의 전체 구조를 보여준다.

3.4 핵심 학습 구조 및 전략

본 연구에서 적용된 MADQN 프레임워크의 학습 구조 및 주요 구현 전략이 반영된 Python 코드는 GitHub 공개저장소(https://github.com/djpanda1217/clustertool_simulator)를 통해 확인할 수 있다. 구체적으로는 dueling DQN 구조를 기반으로 한 독립적인 에이전트를 구성하고, 다양한 학습 전략을 결합하여 강화학습의 안정성과 수렴 효율을 향상시켰다.

Dueling DQN은 가치 함수(value function)과 행동의 이점 함수(advantage function)를 별도로 추정하는 이중 스트림 구조를 통해 기존 DQN의 비효율적인 Q값 추정을 개선하고자 제안된 방식이다(Wang *et al.*, 2016). Dueling DQN은 각 상태 s의 가치 함수 $V(s)$를 독립적으로 학습하고, 각 상태에서의 취할 수 있는 행동의 상대적 중요도를 나타내는 행동 이점 함수 $A(s, a)$를 별도로 추정한다. 기존 DQN과 dueling DQN의 구조적 차이는 <Figure 3>에 도식화되어 있다.

$$Q(s, a,;\theta, \alpha, \beta) = V(s;\theta, \beta) + \left(A(s, a;\theta, \alpha) - \frac{1}{|A|}\sum_{a}^{'} A(s, a';\theta, \alpha)\right)$$

Dueling DQN에서 Q값 계산 방식은 위 수식과 같다. 여기서, $V(s;\theta, \beta)$는 상태 s가 갖는 가치를 나타내는 가치 함수이며, $A(s, a;\theta, \alpha)$는 상태 s에서 행동 α를 취하는 것이 다른 행동을 취하는 것과 비교하여 얼마나 유리한지를 나타내는 상대적 지표인 행동 이점 함수이다. θ는 feature extraction 네트워크의 파라미터를, α와 β는 각각 advantage stream과 value stream의 파라미터를 의미한다. 행동 이점 함수의 평균을 감산하는 방식은 Q값의 식별 불가능성(unidentifiability)을 해결하고, 정책 학습의 안정성을 높인다.

Prioritized experience replay(PER) (Schaul *et al.*, 2015)를 도입하여 replay buffer로부터 temporal difference(TD) 오차가 큰 transition을 우선적으로 샘플링하는 prioritized sampling을 수행하였다. TD 오차가 크다는 것은 에이전트의 Q-value 예측이 실제 환경 결과와 크게 차이가 난다는 것을 의미하며, 이는 해당 transition이 충분히 학습되지 않았고, 따라서 추가 학습을 통해 에이전트의 정책에서 큰 개선을 기대할 수 있는 중요한 경험임을 의미한다. 샘플링 확률은 다음과 같이 정의된다.

$$P(i) = \frac{p_i^{\alpha}}{\sum_k p_k^{\alpha}}, \text{ where } p_i = |\delta_i| + \epsilon$$

여기서 $\delta i = r_i + \gamma \max_{a'} Q(s_i', a') - Q(s_i, a_i)$는 TD 오차이며, α는 우선순위의 민감도를 조절하는 하이퍼파라미터, ϵ은 모든 transition이 0 이상의 확률로 선택되도록 보장하는 작은 상수이다. 이와 함께 PER에서는 중요도 보정(importance sampling, IS)을 위해 다음과 같은 weight를 사용한다.

$$w_i = \left(\frac{1}{N} \cdot \frac{1}{P(i)}\right)^{\beta}$$

여기서 N은 전체 transition 수, β는 보정 계수를 나타내며, 학습 초기에 발생할 수 있는 편향을 줄이고 학습 후반으로 갈수록 정확한 보정을 수행하는 역할을 한다. 실제 학습 시에는 이 weight w_i를 TD 오차에 곱해 bias를 최소화한다. TD 오차가

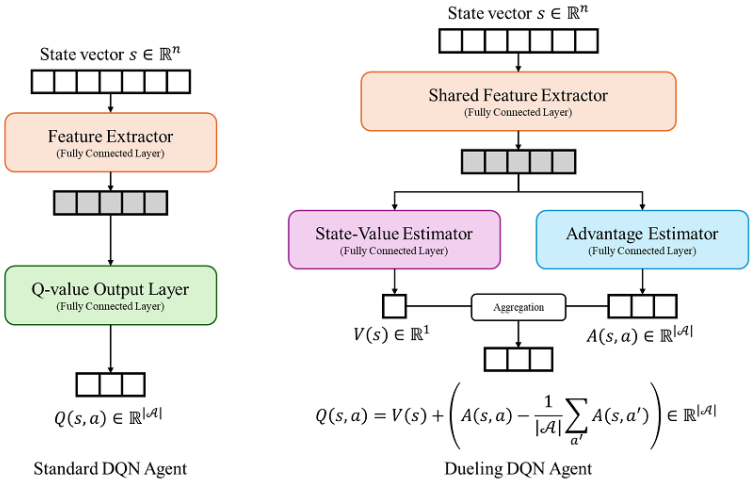

Figure 3. Comparison of standard DQN (left) and Dueling DQN (right) architectures using state vector inputs. Dueling DQN separates the estimation of state-value and advantage, which are then combined through an aggregation module to compute the final action-value function , as proposed by Wang *et al.* (2016).

큰 transition은 여전히 loss에 더 크게 반영되므로, PER은 에이전트가 중요한 경험을 더 자주, 더 강하게 학습하도록 유도한다. 이 과정은 에이전트가 Q-value를 잘 예측하지 못한, 즉 에이전트가 학습해야 할 정보량이 많은 transition을 집중적으로 할 수 있도록 하며, 결과적으로 샘플 효율성과 수렴 속도를 향상시킨다.

이외에도, PyTorch의 StepLR 스케줄러를 사용하여 일정 step마다 learning rate를 0.5배씩 줄이는 방식으로 학습률을 조정하였고, 각 에이전트의 신경망에는 layer normalization을 적용하여 gradient 안정성을 확보하고 학습 수렴 속도를 향상시켰다.

위와 같은 구조적 및 학습 전략적 개선은 혼류 공정 기반의 복잡한 스케줄링 문제에서도 안정적인 정책 학습을 가능하게 하며, 학습 효율성과 성능 향상 모두에 긍정적인 기여를 하였다.

4. 실험 및 결과

4.1 평가 지표 및 실험 설정

성능 평가는 단위시간 당 웨이퍼 처리량을 기준으로 수행하였다. 이를 위해 각 에피소드 길이를 10,000 step으로 고정하였으며, 한 에피소드에서 처리된 웨이퍼 수를 주요 평가 지표로 활용하였다. 모든 실험은 동일한 환경 조건에서 10회 반복 수행되었으며, 결과는 평균 및 표준편차 형태로 제시하였다.

MADQN의 성능을 평가하고자, 산업 현장에서 실제 적용되는 로직과 유사한 세 가지 규칙 기반 정책을 비교 대상으로 설정하였다. 각 정책은 클러스터 장비 개발 및 운영 과정에서의 실무적 고려사항을 반영하고 있으며, 기능적 타당성 확보부터 생산성 최적화에 이르는 일련의 개발 우선순위를 기반으로 구분된다. 첫째, Random 정책은 가능한 모든 행동 중 하나를 무작위로 선택하는 방식으로, 강화학습 기반 정책의 의미 있는 성능을 판단하기 위한 최소한의 기준선 역할을 한다. 별도의 도메인 지식이나 전략이 개입되지 않으므로, agent가 일정 수준 이상의 정책을 학습했는지를 검증하는 기초적인 비교 기준으로 활용하였다. 둘째, Rule 1은 가능한 행동이 하나 이상 있다면 '아무것도 하지 않음'에 해당하는 NO_ACTION을 제외하고, 실행 가능한 행동 중 하나를 임의로 선택하는 방식이다. 이 정책은 클러스터 장비 스케줄러를 개발할 때 가장 먼저 고려되는 조건인 "정상 동작 보장"을 목적으로 구성된 정책으로, deadlock이나 물리적으로 불가능한 작업 시퀀스를 방지하면서도 시스템의 유휴 상태가 최소화되도록 설계된다. 이러한 정책은 실제 장비 제어 로직의 초기 버전에 주로 반영되는 방식이다. 셋째, Rule 2는 실행 가능한 행동 중 소요 시간이 가장 짧은 행동을 선택하는 휴리스틱 기반 정책으로, 로봇의 물리적 위치에 따른 이동 시간 절감 및 로봇 작업의 우선순위를 고려하여 설비의 생산성의 향상을 목표로 한다. 이는 Rule 1을 기반으로 한 시스템 안정성 확보 이후, 실제 생산 환경에서 throughput 극대화를 위해 적용되는 개선 전략으로, 현업에서는 직관성과 일정 수준 이상의 효과를 바탕으로 개발 초기부터 고도화 단계에 이르기까지 널리 활용되고 있다.

이처럼 비교 대상이 되는 세 가지 규칙 기반 정책은 산업 현장의 로직 설계 흐름을 체계적으로 반영하고 있으며, 본 연구에서는 이러한 규칙 기반 정책을 비교군으로 활용하여 MADQN 에이전트가 얼마나 효과적으로 성능을 상회할 수 있는지를 평가하였다. MADQN을 학습하는 과정에서 사용한 주요 하이퍼파라미터는 <Table 3>에 정리하였다. 모든 에이전트는 Dueling DQN 네트워크 구조를 기반으로 하였으며, 탐색을 위해 ϵ-greedy 기법을 적용하였다.

Table 3. Major hyperparameters used in training the proposed MADQN agent

Hyperparameter	Value
Initial learning rate	0.0005
Learning rate scheduler	StepLR (step = 5000, gamma = 0.5)
Discount factor	0.99
Network sync rate	10,000 steps
Replay memory size	10,000
Batch size	32
Initial exploration rate (ϵ)	1.0
Epsilon decay	0.9995
Number of training episodes	10,000
Number of hidden units	1,024
PER priority exponent (α)	0.6
PER IS weight exponent (β)	0.4

본 연구의 실험은 Ubuntu 22.04.5 LTS 운영체제가 설치된 워크스테이션에서 수행되었다. 하드웨어 구성은 다음과 같다. 중앙처리장치(CPU)로는 Intel Core i9-9900KF(기본 동작 속도 3.60GHz, 8코어 16스레드)를 사용하였으며, 메인 메모리는 64GB DDR4를 장착하였다. 그래픽처리장치는 NVIDIA GeForce RTX 2080 Ti (11GB VRAM)를 사용하였으며, 딥러닝 연산을 위한 CUDA Toolkit은 버전 12.1을 기반으로 설정하였다. 모든 실험은 단일 GPU 환경에서 수행되었으며, 시뮬레이터 구동 및 강화학습 학습 과정 전반에 해당 사양이 사용되었다.

4.2 환경별 성능 비교

<Table 4>은 각 시나리오에서 MADQN 정책과 세 가지 규칙 기반 정책 간 처리량(throughput)을 비교한 결과이다. 총 여섯 개의 시나리오에서 실험을 수행하였으며, 각 시나리오는 장비 사양(동일 또는 비동일)과 혼합 투입되는 웨이퍼 종류 수(2종, 3종, 4종)의 조합으로 구성되었다. 모든 실험은 에이전트가 학습한 환경과 동일한 조건 하에서 수행되었으며, 결과는 각 에피소드에서 처리된 웨이퍼 수의 평균으로 표현하였다.

Table 4. Performance evaluation of MADQN and rule-based policies across six scheduling scenarios. Each value represents the average number of wafers processed per 10,000-step episode (mean with standard deviation in parentheses). MADQN shows consistently superior throughput in all cases.

Environment	MADQN	Random	Rule 1	Rule 2
A_2	**128.80(0.40)**	108.10(1.70)	110.50(1.75)	119.00(0.00)
A_3	**126.10(0.30)**	107.00(1.61)	110.30(1.95)	118.60(0.49)
A_4	**127.10(0.30)**	106.90(2.26)	110.30(2.05)	119.00(0.00)
B_2	**128.00(0.00)**	106.00(1.84)	108.20(1.60)	100.00(0.00)
B_3	**122.10(1.76)**	104.00(2.10)	109.20(1.47)	100.40(0.92)
B_4	**117.30(1.42)**	104.80(1.08)	107.70(1.79)	100.80(1.17)

환경 A와 B를 포함한 모든 시나리오에서 MADQN은 규칙 기반 정책을 크게 상회하는 처리량을 기록하였다. 이는 MADQN 에이전트가 단순한 조건부터 복잡한 조건에 이르기까지, 상태 정보를 바탕으로 전체 장비 구조, 로봇 간 협업 관계, 그리고 고정 흐름을 종합적으로 고려한 스케줄링 전략을 효과적으로 학습할 수 있음을 보여준다. 이러한 학습 능력은 다양한 환경에서도 일관된 성능 우위를 가능하게 한 주요 요인으로 작용하였다.

환경 A는 모든 챔버의 사양이 동일하여, 동일한 종류의 웨이퍼는 어느 챔버에서든 동일한 처리 시간을 갖는 비교적 단순한 구조의 환경이다. 이러한 단순한 환경에서는 기존 규칙 기반 정책들도 일정 수준의 성능을 유지하였으며, 특히 Rule 2는 기존 규칙 기반 정책 중 가장 높은 성능을 달성하였다. 이렇게 단순한 규칙만으로도 일정 수준의 성능이 나오는 비교적 간단한 환경에서도, MADQN는 이를 넘어서는 수준의 협업 전략을 학습함으로써 명확한 성능 우위를 보였다. 반면 환경 B는 각 챔버의 사양이 서로 달라 같은 종류의 웨이퍼라도 처리 시간이 서로 다른 상대적으로 복잡한 환경이다. 이러한 환경에서는 기존의 규칙 기반 정책들이 복잡성 증가에 효과적으로 대응하지 못하는 한계를 드러냈다. 특히 Rule 2는 챔버 간 처리 시간 차이를 고려하지 않고 단순히 소요 시간이 가장 짧은 행동을 선택하는 방식이기 때문에, 환경 B와 같이 처리 시간의 불균형이 존재하는 상황에서 적절한 대응이 어렵다. 예를 들어, 웨이퍼 B의 경우 entry 로드락에서 가장 가까운 챔버 1과 챔버 2에서 처리할 경우, 다른 챔버에 비해 약 30%의 시간이 더 소요된다. 그럼에도 불구하고 Rule 2는 가장 짧은 이동 거리를 기준으로 챔버 1 또는 챔버 2를 선택하게 되며, 이로 인해 상대적으로 비효율적인 자원 배분이 발생한다. 결과적으로, 처리 시간이 긴 챔버들에 작업이 집중되고, 처리 시간이 짧은 챔버들이 유휴 상태로 남게 되는 자원 불균형 현상이 초래된다. 이러한 비효율은 실제로 환경 B의 시나리오들에서 Rule 2가 Random 정책보다도 낮은 성능을 기록하는 결과로 이어졌으며, 이는 규칙 기반 방식이 가지는 구조적 한계를 명확히 보여준다. 반면, MADQN은 클러스터 장비의 상태 정보를 바탕으로 장비 구조, 챔버 간 성능 차이, 로봇 간 협업 관계 등 다양한 요인을 통합적으로 고려하는 전략을 학습함으로써, Rule 2가 직면하는 자원 불균형 문제를 효과적으로 회피하였다. 그 결과, 환경 B와 같이 복잡성이 높은 환경에서도 MADQN은 규칙 기반 정책 대비 일관적으로 높은 성능을 달성할 수 있었다.

4.3 일반화 성능 분석

본 절에서는 제안한 MADQN 에이전트의 일반화 능력을 평가하기 위해, 웨이퍼 2종 환경(A_2, B_2)에서만 학습한 에이전트를 보다 복잡한 시나리오(A_3, A_4, B_3, B_4)에 적용하는 실험을 수행하였다. 실험 결과는 <Table 5-1>과 <Table 5-2>에 각각 정리되어 있으며, 각 표에서는 2종 웨이퍼 환경에서 학습된 모델(MADQN$_2$)과 학습 환경과 테스트 환경이 동일한 경우(MADQN)의 성능을 비교하였다. 또한 상대 성능 변화율도 함께 제시하여 일반화 성능을 정량적으로 평가하였다.

Table 5-1. Generalization performance in Environment A. Each value represents the average number of wafers processed per 10,000-step episode (mean with standard deviation in parentheses). "MADQN" refers to the agent trained and tested in the same scenario. "MADQN$_2$" refers to the agent trained only in A_2 and tested in A_2, A_3 and A_4.

	A_2	A_3	A_4
MADQN	128.80(0.40)	126.10(0.30)	127.10(0.30)
MADQN$_2$	128.80(0.40) (0.00%)	125.30(4.78) (-0.63%)	125.20(4.75) (-1.49%)
Random	108.10(1.70)	107.00(1.61)	106.90(2.26)
Rule 1	110.50(1.75)	110.30(1.95)	110.30(2.05)
Rule 2	119.00(0.00)	118.60(0.49)	119.00(0.00)

<Table 5-1>은 환경 A에서 수행한 일반화 성능 평가 결과를 보여준다. 환경 A는 모든 챔버의 웨이퍼 처리 사양이 동일한 구조적으로 단순한 환경이지만, 혼합 투입되는 웨이퍼 종류 수가 증가함에 따라 공정의 복잡도는 점진적으로 높아진다. 그럼에도 불구하고, 2종 웨이퍼 시나리오(A_2)에서만 학습된 MADQN$_2$는 보다 복잡한 시나리오인 A_3 및 A_4에서도 각각 -0.63%, -1.49%의 경미한 성능 저하만을 보이며, 처리량 125 이상을 안정적으로 유지하였다. 이는 MADQN$_2$가 단순한 조건

에 과적합되지 않고, 공정 조건 변화에도 유연하게 적응할 수 있는 일반화된 정책을 학습했음을 시사한다.

Table 5-2. Generalization performance in Environment B. Each value represents the average number of wafers processed per 10,000-step episode (mean with standard deviation in parentheses). "MADQN" refers to the agent trained and tested in the same scenario. "$MADQN_2$" refers to the agent trained only in B_2 and tested in B_2, B_3 and B_4.

	B_2	B_3	B_4
MADQN	128.00(0.00)	122.10(1.76)	117.30(1.42)
$MADQN_2$	128.00(0.00) (0.00%)	111.20(1.33) (-8.93%)	111.60(1.62) (-4.86%)
Random	106.00(1.84)	104.00(2.10)	104.80(1.08)
Rule 1	108.20(1.60)	109.20(1.47)	107.70(1.79)
Rule 2	100.00(0.00)	100.40(0.92)	100.80(1.17)

<Table 5-2>는 환경 B에서의 일반화 성능에 대한 실험 결과이다. 환경 B는 동일한 종류의 웨이퍼라도 챔버의 사양에 따라 처리 시간이 상이한 구조적 특성을 가지며, 이에 따라 선택 편향과 자원 불균형이 동시에 발생할 수 있는 복잡한 환경이다. B_2에서 학습한 $MADQN_2$는 보다 복잡한 B_3 및 B_4 시나리오에 적용되었을 때, 기준 대비 각각 -8.93%, -4.86%의 성능 저하를 보였다. 성능 감소폭만 보면 환경 A의 경우보다 크지만, 그럼에도 불구하고 $MADQN_2$는 여전히 기존 규칙 기반 정책들을 상회하는 처리량을 유지하였다. 이는 MADQN이 단순한 환경에서 학습되더라도, 단편적인 상황에 과적합되지 않고 기본적인 챔버 선택 전략, 큐 관리 전략 등을 구조화된 형태로 학습하고 있음을 보여준다.

결론적으로, 본 절의 실험 결과는 MADQN이 단순한 조건에서 학습되더라도 환경 상태와 그에 따른 전략을 일반화 가능한 구조로 학습함으로써, 장비 사양이나 공정 조건이 변화하더라도 강건하고 범용적인 성능을 유지할 수 있음을 입증한다. 이는 실제 제조 환경에 적용 가능성이 높다는 점에서 제안된 방법론의 실용적 가치를 뒷받침한다.

5. 결 론

본 연구는 서로 다른 공정 특성을 지닌 웨이퍼가 혼합된 혼류 공정 환경에서, 반도체 클러스터 장비 내 VTM 로봇과 ATM 로봇의 협업 스케줄링 문제를 다루었다. 기존 규칙 기반 정책은 공정 조건이 복잡하거나 변화하는 상황에서 성능이 저하되는 한계를 지닌다. 이를 극복하기 위해, 본 연구에서는 멀티에이전트 심층 강화학습을 기반으로 한 자율 스케줄링 정책을 제안하였다. 제안 방식은 각 로봇이 상태 정보를 기반으로 협업 전략을 스스로 학습하도록 구성되었으며, 다양한 공정 제약 조건과 환경 설정을 반영한 시뮬레이터를 통해 그 효과를 실험적으로 입증하였다.

실험은 챔버 간 사양이 동일한 환경 A와, 챔버별 처리 속도가 상이한 보다 현실적인 환경 B를 포함하여 총 여섯 개의 시나리오에서 수행되었다. 실험 결과, MADQN은 모든 시나리오에서 기존 규칙 기반 정책을 상회하는 처리량을 기록하였으며, 특히 챔버 간 처리 시간의 차이가 존재하는 복잡한 환경 B에서도 안정적인 성능을 유지하였다. 규칙 기반 정책 중 실무에서 자주 활용되는 Rule 2는 이동 시간을 기준으로 작업을 선택하는 단순한 전략에 기반하고 있어, 챔버 간 사양이 상이한 환경에서는 자원 불균형과 병목 현상을 유발하며, 오히려 Random 정책보다 낮은 성능을 나타내는 한계를 보였다. 반면, MADQN은 각 시점에서의 시스템 상태를 종합적으로 고려하여 전체 공정 흐름을 최적화하는 협업 전략을 효과적으로 학습하였다. 또한, 웨이퍼 2종만 등장하는 단순한 환경에서 학습된 MADQN이, 더 많은 종류의 웨이퍼가 혼합 투입되는 복잡한 환경에서도 높은 성능을 유지한 결과는, 제안된 방식이 특정 조건에 과도하게 적응하지 않고 일반화 가능한 정책을 학습하고 있음을 실증적으로 보여준다. 구조가 단순한 환경 A에서는 성능 저하 없이 원활한 전이가 이루어졌으며, 상대적으로 난이도가 높은 환경 B에서도 기존 규칙 기반 정책 대비 안정적으로 높은 처리량을 기록하였다. 이러한 결과는 제안된 스케줄링 프레임워크가 공정 조건과 장비 구조가 변화하는 다양한 제조 환경에서도 강건하게 작동할 수 있음을 의미하며, 실제 산업 현장에서의 적용 가능성을 뒷받침한다.

본 연구에서 제안한 MADQN은 로봇 간의 물리적 독립성, 관측 공간 및 행동 공간의 구분성, 처리 흐름의 명확한 경계 등을 고려하여 독립형 Q-러닝 구조를 선택하였고, 상대적으로 간단한 해당 구조를 반도체 클러스터 장비의 혼류 공정 문제 환경에 알맞게 적용하여 충분히 높은 성능을 달성할 수 있음을 보였다. 하지만 이는 최근 MARL 분야에서 활발히 논의되고 있는 중앙집중 학습-분산 실행(centralized training with decentralized execution, CTDE) 프레임워크, 공유된 critic 구조, 또는 상호작용 기반의 정책 공유 방식에 비해 정교한 협업 전략 도출 측면에는 한계를 가질 수 있다. 향후에는 최근 논의되고 있는 협업 전략 학습을 위한 MARL 프레임워크와 함께, attention 기반의 상호작용 모듈 또는 메시지 교환 구조 등을 도입함으로써 로봇 간 협업 전략을 보다 세밀하게 설계할 필요가 있으며, 이를 통해 일반화 성능과 협업 효율성을 동시에 향상시킬 수 있을 것으로 기대한다. 또한, 본 연구에서 활용한 실험 환경은 반도체 클러스터 장비의 구조 및 공정 조건 일부를 단순화한 모델에 기반하고 있어, 실제 반도체 생산 현장 시스템 전체를 정밀하게 반영하기에는 어려움이 있다. 예를 들어, 행동의 가능 여부 등을 판단하기 위해 실제 현장에서는 다양한 정보를 활용하지만, 본 연구에서는 장비 점유 여부와 행동 소요 시간만을 고려하였다. 또한, 사용자 요구에 따라 사전에 정의된 웨이퍼 처리 우선순위, 납기 제한, 챔버 간 품질 차이

등과 같은 현실적인 생산 조건은 아직 반영되지 않았다. 향후 연구에서는 이러한 요소들을 반영할 수 있는 시뮬레이션 환경으로 확장함으로써, 제안된 프레임워크의 현실 적용 가능성을 보다 정밀하고 다각도로 평가할 수 있을 것이다. 이와 함께, 하이퍼파라미터 튜닝의 자동화 및 학습 전략의 효율화 등을 통해 제안 방식의 실제 운영 환경 적용 가능성을 한층 향상시킬 수 있을 것으로 기대된다. 나아가, 이러한 조건들을 더욱 정교하게 반영하고, 다양한 DRL 구조 및 학습 전략과의 비교 연구를 수행함으로써, 보다 강건하고 일반화된 스케줄링 프레임워크로 확장해 나갈 수 있을 것으로 기대한다.

참고문헌

Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. (2018, April), Counterfactual multi-agent policy gradients, In *Proceedings of the AAAI conference on artificial intelligence*, **32**(1).

Hong, C. and Lee, T. E. (2018, December), Multi-agent reinforcement learning approach for scheduling cluster tools with condition based chamber cleaning operations, In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA),* IEEE, 885-890.

Jung, C. and Lee, T. E. (2011), An efficient mixed integer programming model based on timed Petri nets for diverse complex cluster tool scheduling problems, *IEEE Transactions on Semiconductor Manufacturing*, **25**(2), 186-199

Kim, H. J. and Lee, J. H. (2024), Scheduling Cluster Tools for Concurrent Processing: Deep Reinforcement Learning With Adaptive Search, *IEEE Transactions on Automation Science and Engineering*.

Lee, C. and Lee, S. (2021, March), A practical deep reinforcement learning approach to semiconductor equipment scheduling, *In 2021 22nd IEEE International Conference on Industrial Technology (ICIT)*, IEEE, **1**, 979-985.

Lee, J. H., Kim, H. J., and Lee, T. E. (2015), Scheduling cluster tools for concurrent processing of two wafer types with chamber sharing, *International Journal of Production Research*, **53**(19), 6007-6022.

Lee, T. E. (2008, December), A review of scheduling theory and methods for semiconductor manufacturing cluster tools, In *2008 Winter Simulation Conference*, IEEE, 2127-2135.

Li, X. and Yang, W. (2025), Cyclic scheduling of multi-type wafers concurrent processing in single-arm cluster tools with residency time constraints, *Expert Systems with Applications*, **269**, 126443.

Lin, A. T., Debord, M., Estabridis, K., Hewer, G., Montufar, G., and Osher, S. (2022, April), Decentralized multi-agents by imitation of a centralized controller, In *Mathematical and Scientific Machine Learning*, PMLR, 619-651.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., ... and Kavukcuoglu, K. (2016, June), Asynchronous methods for deep reinforcement learning, In *International Conference on Machine Learning*, PMLR, 1928-1937.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... and Hassabis, D. (2015), Human-level control through deep reinforcement learning, *Nature*, **518**(7540), 529-533.

Pan, C., Zhou, M., Qiao, Y., and Wu, N. (2017), Scheduling cluster tools in semiconductor manufacturing: Recent advances and challenges, *IEEE Transactions on Automation Science and Engineering*, **15**(2), 586-601.

Qiao, Y., Wu, N., Zhu, Q., and Bai, L. (2015), Cycle time analysis of dual-arm cluster tools for wafer fabrication processes with multiple wafer revisiting times, *Computers & Operations Research*, **53**, 252-260.

Rashid, T., Samvelyan, M., De Witt, C. S., Farquhar, G., Foerster, J., and Whiteson, S. (2020), Monotonic value function factorisation for deep multi-agent reinforcement learning, *Journal of Machine Learning Research*, **21**(178), 1-51.

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. (2015), Prioritized experience replay, arXiv preprint arXiv:1511.05952.

Suerich, D. and McIlroy, T. (2022, May), Artificial intelligence for real time cluster tool scheduling: EO: Equipment optimization, In *2022 33rd Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, IEEE, 1-3.

Tan, M. (1993), Multi-agent reinforcement learning: Independent vs. cooperative agents, In *Proceedings of the Tenth International Conference on Machine Learning*, 330-337.

Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., and Freitas, N. (2016, June), Dueling network architectures for deep reinforcement learning, In *International Conference on Machine Learning*, PMLR, 1995-2003.

Wu, N., Chu, F., Chu, C., and Zhou, M. (2010), Petri net-based scheduling of single-arm cluster tools with reentrant atomic layer deposition processes, *IEEE Transactions on Automation Science and Engineering*, **8**(1), 42-55.

Yan, Y., Wang, H., Tao, Q., Fan, W., Lin, T., and Xiao, Y. (2020), Noncyclic scheduling of multi-cluster tools with residency constraints based on pareto optimization, *IEEE Transactions on Semiconductor Manufacturing*, **33**(3), 476-486.

Zhang, J., Liu, C., Li, X., Zhen, H. L., Yuan, M., Li, Y., and Yan, J. (2023), A survey for solving mixed integer programming via machine learning, *Neurocomputing*, **519**, 205-217.

저자소개

차민성: 고려대학교 산업경영공학부에서 2023년 학사 학위를 취득하고, 고려대학교 산업경영공학과에서 석사과정에 재학 중이다. 연구분야는 Reinforcement Learning, Optimization이다.

최종원: 고려대학교 컴퓨터학과에서 2015년 학사, 고려대학교 산업경영공학과에서 2025년 석사학위를 취득하였다. 현재 삼성전자에서 Staff Engineer로 재직 중이다.

김성범: 고려대학교 산업경영공학부 교수로 2009년부터 재직하고 있으며, 인공지능공학연구소 소장, 기업산학연협력센터 센터장을 역임했다. 미국 University of Texas at Arlington 산업공학과에서 교수를 역임하였으며, 한양대학교 산업공학과에서 학사 학위를 미국 Georgia Institute of Technology에서 산업시스템공학 석사 및 박사학위를 취득하였다. 인공지능, 머신러닝, 최적화 방법론을 개발하고 이를 다양한 공학, 자연과학, 사회과학 문제에 응용하는 연구를 수행하고 있다.