

Development of Real-Time Scheduling System for OHT Mission Planning

Lee Bok-Ju[†] · Park Hee-Mun^{††} · Kwon Yong-Hwan^{†††} · Han Kyung-Ah^{††††} · Seo Kyung-Min^{†††††}

ABSTRACT

For smart manufacturing, most semiconductor sites utilize automated material handling systems(AMHS). As one of the AMHSs, the OHT control system(OCS) manages overhead hoist transports(OHT) that move along rails installed on the ceiling. This paper proposes a real-time scheduling system to efficiently allocate and control the OHTs in semiconductor logistics processes. The proposed system, as an independent subsystem within the OCS, is interconnected with the main subsystem of the OCS, so that it can be easily modified without the effect of other systems. To develop the system, we first identify the functional requirements of the semiconductor logistics process and classify several types of control scenarios of the OHTs. Next, based on SEMI(Semiconductor Equipment and Materials International) standard, we design sequence diagrams and interface messages between the subsystems. The developed system is interoperated with the OCS main subsystem and the database in real time and performs two major roles: 1) OHT dispatching and 2) pathfinding. Six integrated tests were carried out to verify the functions of the developed system. The system was normally operated on six basic scenarios and two exception scenarios and we proved that it is suitable for the mission planning of the OHTs.

Keywords : Semiconductor Manufacturing, Logistics Process, OHT(Overhead Hoist Transport), Vehicle Dispatching, Path Finding

OHT 작업 계획을 위한 실시간 스케줄링 시스템 개발

이 복 주[†] · 박 희 문^{††} · 권 용 환^{†††} · 한 경 아^{††††} · 서 경 민^{†††††}

요 약

반도체 물류 공정은 스마트 제조를 위해 물류 자동화 시스템을 운영한다. 물류 자동화 시스템 중 하나인 OCS(OHT Control System)는 천장에 설치된 레일을 따라 이동하는 OHT(Overhead Hoist Transport)를 자율적으로 제어하는 시스템이다. 본 논문은 반도체 물류 공정에서 효율적인 작업 계획을 위한 실시간 스케줄링 시스템을 제안한다. 제안 시스템의 주된 역할은 OHT 할당 및 최적경로 생성이며, 전체 OCS에서 별도의 독립된 시스템으로 개발하여 시스템의 수정 및 확장이 용이하도록 구성하였다. 시스템 개발을 위해 반도체 물류의 기능 요구사항을 식별하고, 명령 유형에 따른 OHT 제어 시나리오를 정의하였다. 그리고 시스템 간 연동 확장성을 위해 국제반도체장비재료협회(SEMI) 규격을 적용하여 시퀀스 다이어그램과 인터페이스 메시지를 설계하였다. 기능 요구사항과 설계 문서를 바탕으로 개발된 스케줄링 시스템은 Main 시스템 및 데이터베이스와 실시간 연동되어 OHT의 최적할당 및 경로제어를 수행한다. 개발 시스템의 기능 검증을 위해 실제 반도체 물류 현장에서 OCS Main 시스템과 통합시험을 수행하였다. 6가지의 기본 시나리오와 2가지의 예외 시나리오에서 개발 시스템이 OHT 최적할당과 경로제어를 성공적으로 수행함을 검증하였다.

키워드 : 반도체 제조, 물류 공정, 무인 이송차량, 차량 할당, 경로 생성

1. 서 론

스마트 공장은 효율적인 제품 생산을 위해 공정 설비 간 연동을 기반으로 한 자동화된 의사결정 시스템을 필요로 한다. 예를 들어, 스마트 공장은 현장 작업 지시와 공정 관리 등 생산

활동을 지원하는 MES(Manufacturing Execution System), 제품의 주문에서 출하까지 물류 흐름을 관리하는 SCM(Supply Chain Management), 그리고 고객 주문정보를 포함한 전사적 자원관리를 위한 ERP(Enterprise Resource Planning) 등이 유기적으로 연결되어 제품을 효율적으로 생산하고 공급할 수 있다[1].

Fig. 1은 반도체 제조를 위한 물류 시스템 구성을 나타낸다. 반도체 물류 공정은 무인 이송수단을 제어하기 위한 MCS(Material Control System)를 포함한다. VMS(Vehicle Management System)는 반송 차량 관리 시스템으로 MCS로부터 전달받은 반송 지시를 수행한다[2]. OHT(Overhead Hoist Transport), AGV(Automated Guided Vehicle), RGV

※ 이 논문은 2019년도 한국기술교육대학교 교수 교육연구진행과제 지원에 의하여 연구되었음.

† 준 회 원 : 한국기술교육대학교 컴퓨터공학과 박사과정

†† 비 회 원 : 한국기술교육대학교 컴퓨터공학과 박사과정

††† 비 회 원 : (주)휴민텍 소프트웨어팀 부장

†††† 비 회 원 : 한국기술교육대학교 산학협력교수

††††† 정 회 원 : 한국기술교육대학교 융합학과 조교수

Manuscript Received : March 3, 2021

Accepted : April 4, 2021

* Corresponding Author : Seo Kyung-Min(kmse0@koreatech.ac.kr)

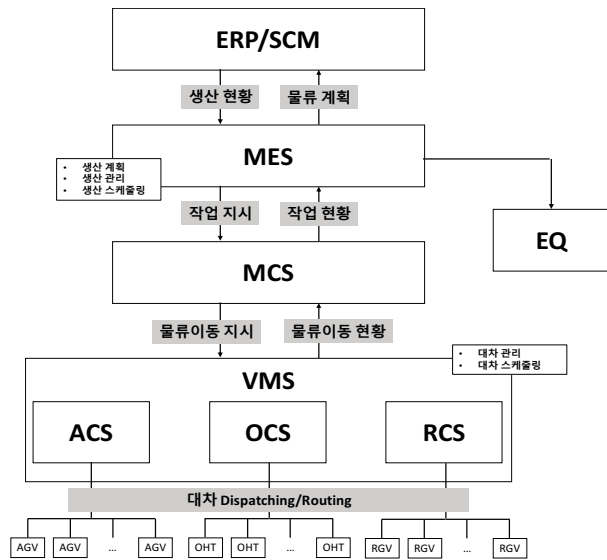


Fig. 1. System Configuration for Semiconductor Manufacturing

(Rail Guided Vehicle) 등 이송수단의 종류에 따라 VMS는 OCS(OHT Control System), ACS(AGV Control System), RCS(RGV Control System)로 구분된다[3].

최근 반도체 물류에서는 공장 천장에 설치된 레일을 따라 제품을 이송하는 OHT를 많이 활용한다[4]. OHT는 공장 규모에 따라 수십 혹은 수백 대가 천장 레일을 통해 이동하므로 공간 효율성이 좋다. 또한, 지상의 작업자나 다른 장비와 충돌 위험이 없어 안정성이 뛰어나다. OCS는 OHT를 제어하는 임무를 수행하는데, 구체적으로 MCS로부터 전달받은 반송 명령(Command)을 해석하여 명령 수행이 가능한 OHT를 할당하고, 해당 OHT가 목적지에 도달하기 위한 최적 경로를 실시간으로 생성한다.

이러한 OCS와 OHT는 반도체 물류 시스템의 핵심 장비 중 하나이다. 그러나 국내 반도체 산업에서는 그동안 이들을 해외 기술에 의존해왔고, 이로 인해 반도체 물류 공정의 기술 유출이 발생하고 있다. 반도체 소재·부품 산업의 국산화를 위해서는 소재·부품뿐 아니라 완제품을 생산하는 장비에 대한 국산화도 필요하다. 산학연의 관련 연구기관에서 OCS 국산화를 위한 연구개발을 진행하는 이유이다.

본 연구는 OCS의 하위 시스템으로 OHT 작업 계획이 가능한 실시간 Scheduling Agent(SA) 시스템을 제안한다. 전체 OCS는 Main 시스템과 SA 시스템으로 구성된다. 제안하는 SA 시스템은 Main 시스템으로부터 수신한 명령에 적합한 OHT를 할당한다. 그리고 할당된 OHT에 대하여 최적 경로를 실시간으로 생성하여 Main 시스템에 전달한다. OHT 할당 및 경로 생성을 위해 본 연구는 Dijkstra 알고리즘을 활용하였다.

SA 시스템 개발을 위해 반도체 물류 요구사항을 식별하고, 명령 유형에 따른 OHT 제어 시나리오를 정의하였다. 특히,

Main 시스템과 SA 시스템 간 연동 확장성을 위해 국제반도체장비재료협회(SEMI: Semiconductor Equipment and Materials International) 규격을 적용하여, SEMI 기반의 시퀀스 다이어그램과 인터페이스 메시지를 설계하였다. 구현된 Main 시스템과 SA 시스템은 TCP/IP 통신을 통해 인터페이스 메시지를 송수신하며 일련의 명령을 수행한다.

개발 시스템의 기능 검증을 위해 Main 시스템과 SA 시스템 간 통합시험을 수행하였다. 실험 목적은 반도체 물류 현장에 구축된 실제 레일과 OHT를 대상으로 특정 시나리오에서 OHT 작업 계획이 안정적으로 수행되는지 정량적으로 측정하는 것이다. 2020년 6월부터 10월까지 여섯 차례의 통합시험을 시행하였는데, 총 287개의 케이스(Case)에 대하여 OHT 작업 계획을 성공적으로 생성하였다. 본 연구에서 개발된 SA 시스템과 전체 반도체 물류 시스템은 사이버물리시스템(CPS: Cyber-Physical System) 구축의 기초 사례로 의미가 있으며, SA 시스템은 기능 확장을 통해 추후 반도체 물류 현장에 적용될 것으로 기대한다.

본 논문의 구성은 다음과 같다. 먼저, 2장에서 OHT 제어 알고리즘 관련한 기존 연구를 살펴본다. 3장에서는 OCS 분석 및 OHT 제어 시나리오를 식별하고, 4장에서 제안 시스템의 설계 및 구현을 설명한다. 5장에서는 Main 시스템과 SA 시스템 간 통합시험 결과를 논의하고, 6장에서 본 연구의 결론을 제시한다.

2. 관련 연구

OHT 스케줄링의 핵심은 OHT의 할당 및 이동 경로를 생성하는 알고리즘이다. 이러한 알고리즘의 특성은 생산 라인 수와 OHT의 운영 대수에 따라 구분될 수 있다. 하나의 물류 라인으로 운영되는 공정은 운영이 복잡하지 않아 OHT 300대 이하로 충분히 운영 가능하며 미리 경로를 지정하여 이동하는 pre-planning 방식을 사용한다[5-8]. 대표적인 pre-planning 알고리즘으로는 Dijkstra 알고리즘, Floyd-Warshall 알고리즘, Bellman-Ford 알고리즘, Genetic 알고리즘이 있다. 반면, 물류 라인이 두 개 이상에서 500대 이상의 OHT가 운용되는 물류 공정(Mega fab)에서는 OHT 운용의 복잡성과 불확실성으로 인해 최근 강화학습 기반의 Q-learning 알고리즘을 많이 활용한다 [9, 10]. 이러한 Q-learning 알고리즘은 물류 상황이 동적으로 변하는 경우에 적합하다.

본 연구의 대상이 되는 반도체 물류 공정은 최대 50대 이하의 OHT를 운용하므로 pre-planning 기반 알고리즘을 적용하였다. 개발 알고리즘은 OCS의 명령 수행을 위해 다수의 OHT 중에서 최적의 OHT를 할당하고 해당 OHT의 최적 경로를 생성한다. 특히, OHT의 현재 위치와 상태, 레일의 혼잡도 등을 고려하여 OHT를 제어하고, 이를 통해 전체 공정의 작업 효율을 향상한다.

반도체 공정은 지능화된 물류 시스템을 이용하여 공정 개선 및 생산 효율성 향상을 달성한다. 반도체 물류 공정은 구조적으로 계층적인 특징과 고가의 설비 운영, 연구개발 환경의 시간/공간적 제약사항 등 현실적 어려움을 포함한다. 이를 해결하기 위한 기술적 접근 방법으로 모델링 및 시뮬레이션 기술을 반도체 공정에 활용해 왔다[1,3]. 그러나 상기 연구들은 실제 공정의 특성을 반영하지 않고, 단순한 물류의 효율성 향상에 초점을 두고 있다. 따라서 실제 공정에 적용하는데 현실적인 어려움이 존재한다.

본 연구에서는 실제 반도체 물류 공정에 적용 가능한 실시간 스케줄링 시스템을 제안한다. 이를 위해 반도체 자동화 물류 시스템에 대한 전반적인 분석을 통해 SEMI 규격을 만족하는 SA 시스템을 개발하였다. 특히, 제안하는 SA 시스템은 기존의 전체 OCS에 종속된 형태가 아닌 별도의 독립된 시스템으로 개발하여 상황에 따라 SA 시스템을 유연하게 확장할 수 있다. 예를 들어, 물류 공정이 확장되어 OHT 운영 대수가 증가할 경우 기존의 pre-planning 알고리즘을 강화학습 알고리즘으로 유연하게 대체하여 활용할 수 있다.

3. 시스템 분석

3.1 OCS 구조 분석

OHT는 천장에 설치된 레일을 통해 단방향으로 이동하며, 작업물(FOUP: Front Opening Unified Pod)을 적재 및 하역할 수 있는 로더가 포함된 무인 이송수단이다. OCS는 천장 레일의 배치를 고려하여 OHT 할당(Dispatching), 경로 생성(Path Finding), OHT 충돌 및 교차 방지를 수행한다.

본 연구의 대상 시스템인 OCS는 크게 Main 시스템과 SA 시스템으로 구성된다. OCS는 상위 시스템인 MCS로부터 물류 이송 작업을 전달받는다. OCS의 Main 시스템은 MCS로부터 전달받은 물류 이송 작업을 개별 명령으로 나누어 SA

시스템에 전달한다. SA 시스템은 명령에 대한 OHT 작업 계획을 수행하고, 수행 결과는 Main 시스템을 통해 OHT에 전달된다.

OHT는 SA 시스템의 작업 계획에 따라 레일을 주행하며 FOUP을 이송한다. OHT에 부착된 IoT 센서는 OHT의 현재 위치를 OCS에 전달한다. 따라서 OCS의 Main 시스템과 SA 시스템은 데이터베이스를 통해 실시간으로 설비, 레일, OHT 등 전체 물류 공정의 상태를 확인한다.

3.2 OHT 제어 시나리오 분석

OCS에서 수행하는 물류 이송 명령은 Table 1과 Fig. 2에 나타내었다. 이송 명령은 기본적으로 From, To, Cancel/Abort, Pause/Resume, Go/MustGo, Replace와 같이 여섯 가지로 구분된다. 반도체 물류는 기본 명령의 조합을 통해 부품이 다양한 설비를 거치게 되면서 하나의 완제품이 생산된다.

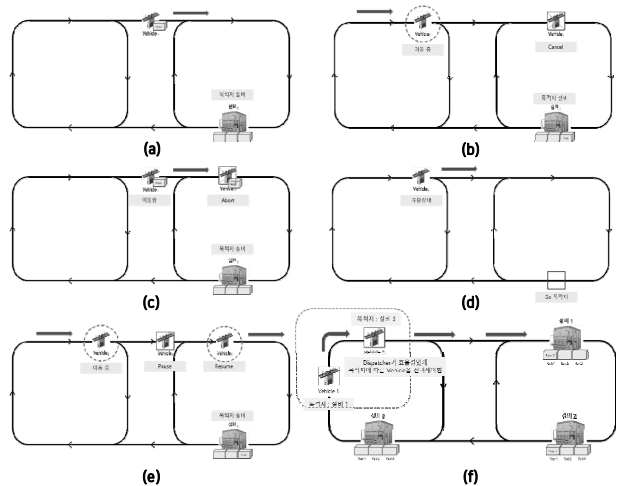


Fig. 2. Command Scenarios : (a) From, (b) To, (c) Cancel /Abort, (d) Go/MustGO, (e) Puase/Resume, (f) Replace

Table 1. OCS Command Type

Command	Explanation
From	OHT A moves to process A to transfer FOUP A and loads (Get) FOUP
To	OHT A transports to the next Facility (Facility B) of FOUP A loaded (Get) through the From command
Replace	In case of two From/To commands occur in three Facilities
Cancel	In case of receiving cancellation command before transferring (Cancel command for From operation)
Abort	In case of receiving a cancellation command after transferring (Cancel command for To operation)
Pause	In case of the OHT moving for a From/To destination pauses its movement
Resume	In case of the OHT in the Pause state moves back to the original From/To destination
Go	OHT A moves to specific area R (R is created by SA)
MustGo	OHT A moves to a specific area R (R is set in MCS)

Table 2. OCS Command Characteristic Classification

Command	Command Generating Entity	SA->Main Request Available	Start	Destination	Destination Generating Entity	Vehicle Assignment	Related Command
From	MCS		Variable (Current position of optimal Vehicle)	From Port	MCS	O	To, Cancel
To	MCS		From Port	To Port	MCS	O	From, Abort
Go	OCS Main	O	Variable (Current position of related Vehicle)	Variable	SA		Cancel, Abort
MustGo	MCS, OCS Main		Variable (Current position of related Vehicle)	Variable	MCS, OCS Main	O	
Pause	MCS, OCS Main	O	Variable (Current position of related Vehicle)	Variable	SA	O	Resume, Cancel, Abort
Resume	MCS, OCS Main		Variable (Current position of related Vehicle)	Variable	MCS, OCS Main	O	Pause, Cancel, Abort
Cancel	MCS, OCS Main		Variable (Current position of related Vehicle)	Current Node	MCS, OCS Main	O (Unassign upon completion)	From
Abort	MCS, OCS Main		Variable (Current position of related Vehicle)	Current Node	MCS, Main	O (Unassign upon completion)	To

Table 2는 OHT 제어 시나리오의 유형을 명령 별로 설명한 표이다. Command Generating Entity는 명령을 전달하는 주체로 From과 To 명령은 MCS에서, Go 명령은 OCS Main 시스템에서만 전달할 수 있다. 이외의 MustGo, Pause/Resum, Cancel/Abort 명령은 MCS와 OCS Main 시스템 모두 명령을 전달할 수 있다. 기본적으로 SA 시스템은 OCS Main 시스템으로부터 명령을 수신하여 스케줄링 기능을 수행한다. 예외적인 상황으로 OHT가 From/To 명령을 수행하기 위해 특정 OHT의 위치를 이동시킬 필요가 있다면 SA 시스템은 OCS Main 시스템에 Go와 Pause 명령을 요청할 수 있다. Start와 Destination은 OHT의 시작 위치와 목표 위치를 나타낸다. 예를 들어, To 명령은 From 명령 이후 수행되므로 From 명령이 끝난 From port가 Start가 된다. 대부분의 명령은 최적 OHT를 할당하는 과정(Vehicle Assignment)을 포함하지만 Go 명령의 경우에는 특정 OHT를 지정하여 위치를 이동시키므로 OHT 할당을 포함하지 않는다.

명령 별로 구체적으로 설명하면 다음과 같다. 먼저, From 명령은 특정 설비에서 작업이 완료된 FOUP을 적재하기 위

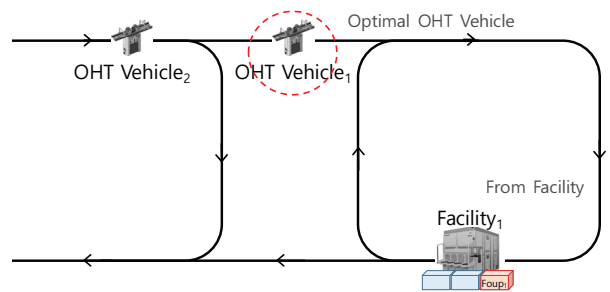


Fig. 3. From Command Scenario

해 이동하는 명령이다. Fig. 3과 같이 From 명령 수행을 위해 레일 위에 있는 유휴 OHT 중 최적 OHT가 선택되어 From 설비 위치로 이동한다. To 명령은 From 명령 수행을 통해 OHT에 적재한 FOUP을 다음 공정에 해당하는 설비로 이송하는 명령이다.

Cancel과 Abort 명령은 모두 취소와 관련한 명령이다. 현재 수행 중인 명령의 종류에 따라 구별되는데, OHT가 From 명령을 수행 중인 경우는 Cancel로 명령을 취소하고, To 명

령을 수행 중인 경우는 Abort로 명령을 취소한다. Go와 MustGo 명령은 FOUP의 이송이 아닌, 단순 이동을 위해서 임의의 OHT를 임의의 위치로 이동시키는 데 필요하다. 단, MustGo 명령은 OHT 배터리 충전 또는 사용자 명령을 통해 목적지가 정해져서 명령이 송신된다. Pause와 Resume 명령은 From, To 명령을 수행 중인 OHT가 사용자 명령 혹은 이동 경로에 고장 OHT 발생 등 예외 상황이 발생한 경우 사용된다.

마지막으로 Replace 명령은 2개의 From/To 명령으로 구성된다. Fig. 2를 예로 들면, Facility B의 공정이 긴급하여, 대기시간을 최소화하여 공정을 진행해야 하는 상황이다. 이를 위해 Facility B의 공정 완료된 FOUP A를 Facility C로 보낸 즉시, Facility A의 FOUP B를 처리하여 대기시간을 최소화해야 한다. 이때, OHT A는 FOUP B에 대한 From/to를, OHT B는 FOUP A에 대한 From/to를 수행해야 한다.

Fig. 4는 From 명령에 대하여 SEMI 규격을 바탕으로 작성한 개발 시스템 간 시퀀스 다이어그램을 나타낸다. OCS의 SA 시스템은 3.3장에서 설명할 Dijkstra 알고리즘을 활용하여 OHT의 작업 계획 결과를 Main 시스템에 전달한다. 그리고 Main 시스템으로부터 해당 명령이 완수되면 완료 메시지를 (Fig. 4의 14번 메시지)를 수신한다.

3.3 경로 탐색 알고리즘 : Dijkstra 알고리즘

경로 탐색 문제는 시작 지점부터 도착 지점까지 최단 거리 또는 최단 비용으로 가는 경로를 찾는 문제이다. 물류 시스템, 교통 시스템, 통신 시스템 등에서는 종종 이러한 경로 탐색 문제를 푸는 일을 요구한다. 경로 탐색 문제를 푸는 방법으로는 Dijkstra 알고리즘, Floyd-Warshall 알고리즘, Bellman-Ford 알고리즘 등이 알려져 있다.

물류 시스템이 속한 공장 환경은 노드(Node)와 엣지(Edge)

를 가지는 방향성 있는 그래프로 표현할 수 있다. 노드는 작업이 일어나는 위치, 레일의 교차점과 같이 의사결정이 일어나는 위치를 표현하는 데 사용된다. 엣지는 각각의 노드들 연결하는 레일을 표현한다. 그리고 엣지의 가중치(Weight)는 엣지를 통과하는 데 걸린 시간을 나타낸다. 또한, 시스템이 작동하는 과정에서 일어나는 일을 노드와 엣지에 대하여 아래와 같이 가정하였다.

먼저, 공정 작업은 노드에서만 일어난다. 즉, 설비의 포트는 노드로 표현된다. 엣지의 중간에 명령이 송신되더라도 OHT는 연결된 노드까지 이동 후 경로를 따라가는 행태를 보인다. 따라서 연결된 노드에서 명령을 받은 것과 동일하다고 간주한다.

다음으로, OHT의 작업은 엣지의 끝에 위치한 노드에서만 일어난다. 작업이 일어나는 위치를 노드로 표현하였으므로 작업은 엣지의 시작 노드 또는 종료 노드에서만 일어난다. 작업이 엣지의 시작 노드에서 일어나면, 임의의 명령을 받아 경로를 지정하는데 각각의 OHT가 작업이 끝나고 이동할 불필요한 엣지 역시 경로에 포함되므로 문제가 복잡해진다.

마지막으로, OHT가 경로를 따라 움직이는 중 다음 엣지가 가득 차 움직일 수 없으면 현재 엣지의 끝에서 대기한다. 물리적으로는 엣지의 종료 노드는 다음 엣지에도 포함되므로 현재 엣지의 중간에서 대차가 기다려야 한다. 또한 엣지의 중간에서 기다렸다가 이동하여 작업을 수행하는 것과 미리 종료 노드로 이동하여 작업을 수행한 후 기다리는 것은 후자의 경우 훨씬 적은 시간을 기다린다. 이 문제를 해결하기 위해 본 과제에서는 엣지는 선로의 시작점을 포함하지 않고 끝에 위치한 노드만 포함하는 것으로 정의하여 사용한다.

해당 문제를 해결하기 위해 본 논문에서는 Dijkstra 알고리즘을 사용하였다. Dijkstra 알고리즘은 다이나믹 프로그래밍을 활용한 대표적인 최단 경로(Shortest Path) 탐색 알고리즘이다. 흔히 인공위성 GPS 소프트웨어 등에서 가장 많이 사용된다. Dijkstra 알고리즘은 특정한 하나의 정점에서 다른 모든 정점으로 가는 최단 경로를 계산할 수 있다. 다만, 이때 음의 간선을 포함할 수 없다. 물론 현실 세계에서는 음의 간선이 존재하지 않기 때문에 Dijkstra 알고리즘은 현실 세계에 사용하기 매우 적합한 알고리즘 중 하나라고 할 수 있다.

4. 시스템 설계 및 구현

4.1 시스템 구조 설계

본 연구에서 제안하는 시스템 구조 설계는 Fig. 5와 같다. 먼저, MCS는 제품 생산을 위한 물류 프로세스를 생성하여 OCS의 Main 시스템에 전달한다. OCS Main 시스템은 물류 프로세스를 해석 및 단위 명령으로 분할하여 SA 시스템에 명령 수행에 필요한 정보를 요청한다. SA 시스템은 From scheduler, To scheduler 등과 같은 명령 객체를 생성하여

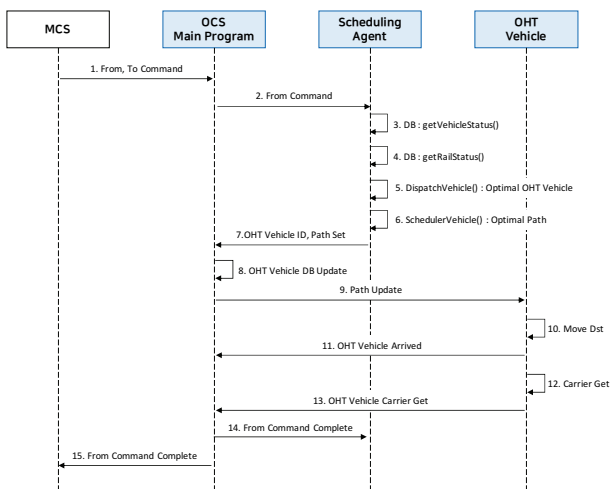


Fig. 4. Sequence Diagram for From Command

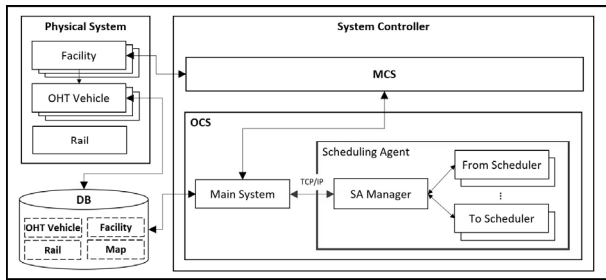


Fig. 5. Proposed System Configuration

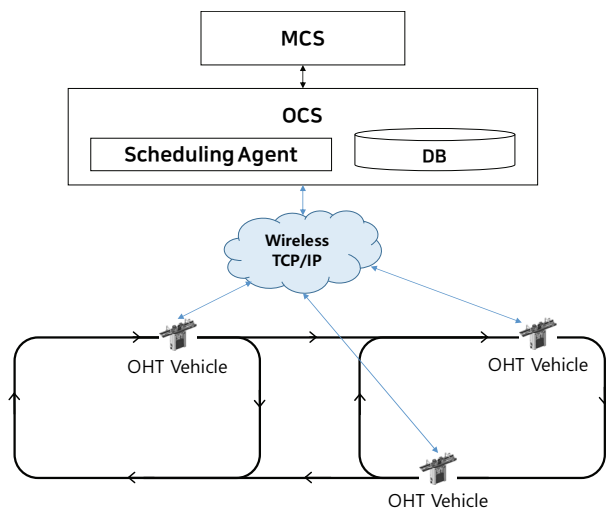


Fig. 6. Network Interface Structure for OHT System

OHT 할당 및 경로 생성을 수행하고 그 결과를 Main 시스템 전달한다. OCS Main 시스템은 전달받은 명령 수행 결과를 데이터베이스에 업데이트 하여 OHT를 제어한다. OHT가 명령 수행을 완료하면 SA 시스템의 해당 명령 객체는 삭제된다.

4.2 시스템 인터페이스 설계

반도체 제조 공정에 있어서 모든 관련 설비들이 갖추어야 할 조건 및 특성은 SEMI에서 규정하고 있다. 따라서 장치 간의 인터페이스 및 통신 방식은 SEMI 규격을 반드시 준수해야 한다. MCS 및 OCS, OHT 상호 간의 통신 인터페이스 구조는 Fig. 6과 같다. OHT는 천장에 설치된 레일을 주행하며 원거리 이송 작업을 하는 특성이 있으므로 유선 통신 방식을 적용할 수 없다. 따라서 이송 작업 명령을 송신하는 OCS와 무선으로 통신한다.

MCS는 목표한 생산량을 달성하기 위하여 각 제조 장치 및 물류시스템을 통합적으로 관리한다. 실시간으로 전체 공정의 작업 상황을 모니터링 하면서, MES의 작업 지시를 수행하기 위해 작업이 완료된 작업물을 이동시키기 위해 OCS에 명령을 송신한다.

OCS는 OHT를 통합 제어/관리하는 역할을 한다. 또한, 반도체 제조 공정 간의 작업물 이송을 위해 OHT를 할당하

Table 3 Type A Message

	Message Type	Send/Receive Direction
A01	Command	Main → Scheduling Agent
A02	Command Reply	Scheduling Agent → Main
A03	Complete Command	Main → Scheduling Agent
A04	Complete Command Reply	Scheduling Agent → Main
A05	Block	Main → Scheduling Agent
A06	Block Reply	Scheduling Agent → Main
A07	Error	Main → Scheduling Agent
A08	Command Request	Scheduling Agent → Main

Table 4 Type B Message

	Message Type	Send/Receive Direction
B01	Command	Main → OHT
B02	Command Reply	OHT → Main
B03	Event	OHT → Main
B04	Event Reply	Main → OHT
B05	Alarm Event	OHT → Main
B06	Alarm Event Reply	Main → OHT

고, 경로를 스케줄링 하여 해당 OHT에 이송 명령을 송신한다. OCS로부터 이송 명령을 받은 OHT는 송신 받은 이송 경로를 통하여 각 설비 간에 이송 작업을 수행한다. 이때, 이송 중인 OHT 상호 간의 충돌을 방지하기 위해서 이송 경로에 존재하는 타 OHT와 충돌을 방지할 수 있는 위치로 이동 명령을 송신한다.

본 연구에서는 SEMI 규격을 준수하는 시스템을 구현하기 위해 OCS Main 시스템과 SA 시스템 간, OCS Main 시스템과 OHT 간의 인터페이스 데이터 구조를 설계하였다. Table 3은 Type A에 대한 분류이고, Table 4는 Type B에 대한 분류이다. 먼저, OCS Main 시스템과 SA 시스템간의 인터페이스 데이터 구조는 Type A로 분류하였고, 총 8개의 세부 유형으로 분류된다. OCS Main 시스템과 OHT 간의 인터페이스 데이터 구조는 Type B로 분류하였고, 총 6개의 세부 유형으로 분류된다.

Type A는 OCS Main 시스템과 SA 시스템 간의 인터페이스이다. Type A01은 OCS Main 시스템이 SA 시스템으로 명령 계산을 요청하는 메시지이며, Type A02는 OCS Main 시스템으로부터 받은 Type A01에 대한 응답 메시지이다. Type A01로 받은 명령에 대한 계산 결과를 포함하고 있다.

Type A03 메시지는 Type A02의 계산 결과에 따라 OHT의 명령 수행 결과를 SA 시스템으로 알려주는 메시지이다. SA 시스템은 해당 메시지의 수신을 통해, 현재 수행 중인 명령을 파악하고, 완료된 명령의 객체를 삭제할 수 있다. Type A04 메시지는 Type A03에 대한 응답 메시지다.

Type A05 메시지는 전체 공정 지도에서 발생한 Block 정보를 SA 시스템으로 알려주는 메시지다. SA 시스템은 Type A05 메시지를 통해 예외 상황을 파악하고, 필요한 명령을 OCS Main 시스템으로 요청한다. 이때, SA 시스템에서 요청을 위해 사용하는 메시지 유형은 Type A08이며, Type A06 메시지는 Type A05의 응답 메시지다. Type A07 메시지는 OCS Main 시스템이 Type A02 메시지를 수신하여, 메시지를 식별하는 과정에서 메시지 구조 및 값 등에 문제가 발생하는 경우, SA 시스템으로 파악된 문제가 무엇인지 반환하는 메시지다. SA 시스템은 해당 메시지를 수신하는 경우 별도의 메시지를 수신하지 않는다. 주로 OCS Main 시스템에서 로그를 저장하기 위해 사용한다.

Type A08 메시지는 SA 시스템이 예외 상황을 해결하기 위해, OCS Main 시스템으로 명령 송신을 요청하는 메시지다. 예를 들어, 명령의 수행을 위해서 유휴 OHT를 Go 명령을 통해 이동해야 하는 경우, SA 시스템은 해당 OHT에 명령을 송신할 수 있는 권한이 없으므로, OCS Main 시스템에서 Go 명령을 송신해 달라고 요청해야 한다. 이때 명령 송신의 요청을 위해 사용하는 메시지가 Type A08 메시지다. Type A08 메시지 안에는 SA 시스템이 요청하는 메시지가 몇 개이며, 현재 몇 번째 요청인지 파악할 수 있는 자료가 포함되고, 관련된 명령이 무엇인지 파악할 수 있는 자료도 포함된다.

Type B는 OCS Main 시스템과 OHT 간의 인터페이스이다. Type B 01은 OCS Main 시스템이 SA 시스템으로부터 받은 계산 결과를 OHT에 할당하는 메시지이며, Type B02는 Type B 01에 대한 응답 메시지다. Type B03은 OHT가 각각의 이벤트가 완료되었을 때, OCS Main 시스템으로 완료된 이벤트를 보고하는 메시지다. OCS Main 시스템은 Type B03 메시지 수신을 통해 해당 OHT의 명령 수행 진행 상황을 파악할 수 있다. Type B04는 Type B 03에 대한 응답 메시지다. Type B05는 OHT 내부에서 오류가 발생하였거나, H/W적으로 외부의 예외를 식별하여 알람이 발생하는 경우, 해당 내용을 OCS Main 시스템으로 보고하는 메시지이며, Type B06은 Type B 05에 대한 응답 메시지다.

5. 실험

본 장에서는 4장에서 제안한 구조를 바탕으로 OCS Main 시스템과 SA 시스템을 가상의 OHT와 연동하여 실험을 진행하였다. 실험 목적은 개발한 SA 시스템의 알고리즘 및 인터페이스 검증이다.

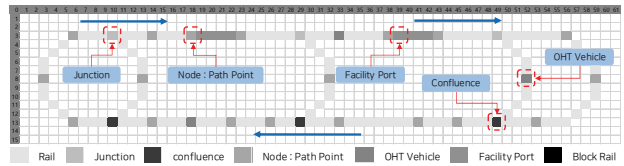


Fig. 7. Network Interface Structure

Fig. 7은 합류와 분기 지점을 포함하여 실험에서 사용한 물류 라인을 나타내는데, 실제 적용된 라인의 일부를 간소화하였다. 실험에서 총 3대의 OHT가 FOUF을 이송하며, 공정은 총 2대의 설비를 통해 이루어진다고 가정하였다. 또한, 동시에 수행 가능한 명령은 2개로 제한하였으며, 모든 명령은 MCS에서 송신된다고 가정하였다.

Table 5는 본 연구에서 수행한 통합시험 결과를 정리한 표이다. 통합시험은 6가지의 기본 시나리오와 2가지의 예외 시나리오로 실험을 진행하였다. 추가적으로 예외 시나리오는 From과 To 명령을 수행 중 진행 경로에 진입 불가능(Block) 구역이 발생한 경우이다. 예외 시나리오에서 SA 시스템은 OCS Main 시스템으로부터 Block 발생 통보와 Block의 Go/MustGo 요청을 송신하여 Block이 처리된 후 기존 명령을 재수행한다. 모든 시나리오마다 OCS Main 시스템에서 SA 시스템으로 명령을 송신하고 SA 시스템이 해당 명령을 정상적으로 수행하는지 확인하였다. 통합시험은 총 6회 진행되었으며, 총 77개의 SEMI 규약 기반의 명령 시나리오에 대하여 통합시험을 수행하였다. Table 5의 Success는 SA 시스템이 SEMI 규약의 명령을 정상적으로 수행 및 응답한 경우를 말하며, 이를 통해 해당 시스템이 OHT 제어에 적합함을 검증하였다.

실험환경은 Fig. 8과 같이 두 개의 프로그램이 TCP/IP 통신을 통해 메시지를 주고받으며, 공용 데이터베이스를 사용하였다. 통합시험은 Fig. 9와 같은 순서로 진행된다. 먼저, OCS Main 시스템에서 SA 시스템으로 명령 계산을 요청하면, SA 시스템은 계산 결과를 수신하고, OCS Main 시스템은 SA 시스템으로부터 받은 계산 결과를 DB를 통해 Vehicle로 전달한다. OHT는 전달받은 명령을 수행함에 따라 발생하는 이벤트를 OCS Main 시스템과 주고받는다. OHT가 명령 수행을 완료하면, OCS Main 시스템은 해당 결과를 SA 시스템에 전달하고 하나의 Command를 완료한다.

기본 시나리오 실험은 OCS Main 시스템에서 SA 시스템으로 From, To, Cancel, Abort, Pause/Resume, Go/MustGo의 명령이 송신되었을 때, SA 시스템이 응답하는 메시지의 분석을 통해, SA 시스템이 해당 명령을 제대로 분석/계산하였는지 검증한다. 예외 시나리오 실험도 같은 방식으로 진행하였다. OCS Main 시스템에서 From To 명령을 송신했을 때, SA 시스템이 이를 정확히 식별하고, 해결 방안을 계산할 수 있는지를 검증한다. 이를 위해서 OCS Main 시스템에서 송신되는 From,

Table 5. Integrated Test Result

Command	1st(20. 06)		2nd (20. 06)		3rd (20. 07)		4th (20. 08)		5th (20. 09)		6th (20. 10)	
	Case	Success	Case	Success	Case	Success	Case	Success	Case	Success	Case	Success
From	11	11	11	11	16	16	16	16	16	16	16	16
To	5	5	5	5	10	10	15	15	15	15	15	15
Cancel	3	3	3	3	3	3	3	3	3	3	3	3
Abort	3	3	3	3	3	3	3	3	3	3	3	3
Pause/Resume	6	6	6	6	6	6	6	6	6	6	6	6
Go/MustGo	4	4	4	4	14	14	19	19	19	19	19	19
Exception : From	-	-	-	-	5	5	5	5	5	5	5	5
Exception : To	-	-	-	-	5	5	10	10	10	10	10	10
Total	32	32	32	32	62	62	77	77	77	77	77	77
Success Rate	100%		100%		100%		100%		100%		100%	

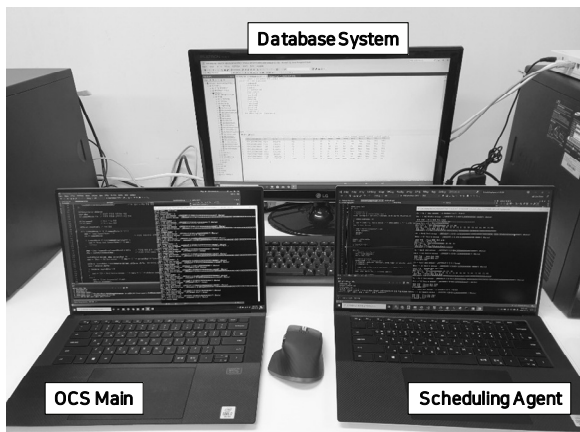


Fig. 8. Integrated Test Environment

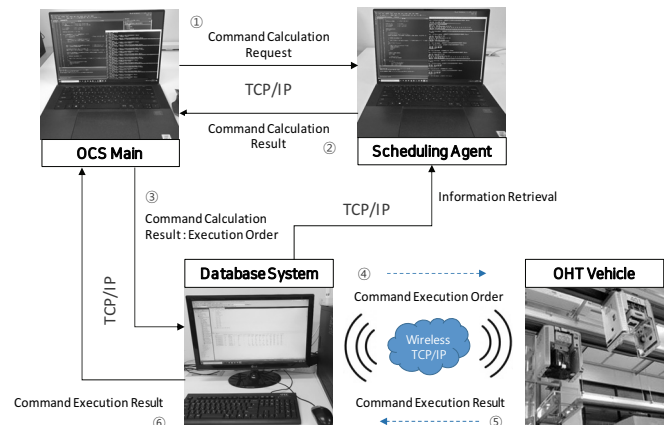


Fig. 9. Integrated Test Interface Sequence

```

SchedulingAgent.exe
Server Connected
[Main -> SA] Receive message : 70400S01FF11111VEHICLE00000000000110000054 ( 45bytes) Command Receive : From
명령 유형 : [From 명령 계산 요청]
TargetVehicleID = VEHICLE00000000000001
목적지 노트 : 5 (3, 19)
발달 Vehicle : VEHICLE00000000000001 29 (13, 18)
(13, 18) (13, 14) (13, 10) (13, 6) (8, 3) (3, 6) (3, 10) (3, 14) (3, 18) (3, 19)
종말 유형 : [From 명령 계산 결과]
7088S002AF11111VEHICLE000000000000109000028000027000026000021000001000002000003000000400000054 Calculation Result Sending
[SA -> Main] Send message : 7088S002AF11111VEHICLE000000000000109000028000027000026000021000001000002000003000000400000054 ( 93bytes)
[Main -> SA] Receive message : 70910S03F111111VEHICLE00000000000014 ( 36bytes) Command Receive : From Complete
명령 유형 : [From 명령 전달]
종말 유형 : [From 종료 ACK]
7012S004AF1111111 Calculation Result Sending : ACK
[SA -> Main] Send message : 7012S004AF1111111 ( 17bytes)
    
```

Fig. 10. Sending/receiving Messages : From Command

To 명령에는 예외 상황이 포함되도록 구성하였다.

Fig. 10은 기본 시나리오 실험 중 From 명령에 대하여 SA 시스템이 OCS Main 시스템과 명령 및 계산 결과를 주고

받은 결과이다. OCS와의 연결 상태, 수신한 메시지, 메시지 분석의 결과, 최적 Vehicle과 최적 경로할당 결과를 보여준다. OCS Main 시스템이 From 명령을 SA 시스템으로 송신


```

OHT_Sim_event_162516.log - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
1 | 1 [0] Generator, model is created
2 | 2 file name: vehicleinfo.txt
3 | 3 file name: facilityinfo.txt
4 | 4 file name: InputEventList.txt
5 | 5 [0] Analyzer, model is created
117 | 117 [0] MCS, (-> MainController), send (from, to) msg
120 | 120 [0] MainController, (-< MCS), receive (from, to) msg
124 | -----
125 | 125 [0] MainController, (-> SAManager), send (from) msg
128 | 128 [0] SAManager, (-< MainController), receive (from) msg, Facility Name - 000005
134 | -----
135 | 135 TargetVehicleID = VEHICLE00000000000001
136 | -----
137 | 137 목적지 노드 : 5 (3, 19)
138 | 138 할당 Vehicle : VEHICLE00000000000001 29 (13, 18)
149 | -----
150 | 150 [0] FromScheduler_111111, (-< SAManager), receive (from) msg
154 | 154 [0] FromScheduler_111111, (-> SAManager), send (from) response msg (vehicleID, path)
157 | 157 [0] SAManager, (-< FromScheduler), receive (from response) msg
161 | 161 [0] SAManager, (-> MainController), send (from response) msg
164 | 164 [0] MainController, (-< SAManager), receive (from response) msg
168 | 168 [0] MainController, (-> Vehicle), send (start) msg
174 | 174 [0] VEHICLE00000000000001, (-> MainController), send (ack. - start) msg
177 | 177 [0] MainController, (-< Vehicle), receive (ack) msg, msgType: F, vehicleID: VEHICLE00000000000001
181 | 181 [1] VEHICLE00000000000001, current position (DST_MOVE_FROM): (000028)
184 | 184 [2] VEHICLE00000000000001, current position (DST_MOVE_FROM): (000027)
187 | 187 [3] VEHICLE00000000000001, current position (DST_MOVE_FROM): (000026)
190 | 190 [4] VEHICLE00000000000001, current position (DST_MOVE_FROM): (000021)
193 | 193 [5] VEHICLE00000000000001, current position (DST_MOVE_FROM): (000001)
198 | 198 [6] MainController, (-< MCS), receive (Pause) msg
102 | -----

```

Fig. 11. OCS, SA, OHT Event log : From Command

하면, SA 시스템은 명령을 계산하여 최적의 OHT와 최적 경로를 응답하게 된다. OCS Main 시스템은 SA로부터 응답받은 결과를 OHT에 전달하여 From 명령을 수행하고, 수행 결과를 다시 SA 시스템과 송수신한다.

Fig. 11은 From 명령의 수행을 기록한 로그이다. 이벤트 흐름에 따라 OCS Main 시스템, SA 시스템, OHT 간의 메시지 송수신 흐름 및 OHT의 명령 수행 진행 상황을 확인 가능하다.

6. 결 론

본 연구는 반도체 물류 공정을 대상으로 OHT 할당 및 경로 생성을 위한 SA 시스템을 제안하였다. 실제 반도체 물류 공정에 적용하기 위해 SEMI 규격을 기반으로 OHT 제어 시나리오를 식별하고, 시퀀스 다이어그램과 인터페이스 메시지를 설계하였다. 특히, 동적으로 OHT 할당 및 경로 생성이 가능하도록 SA 시스템을 설계하였는데, SA 시스템을 전체 OCS에 종속된 형태가 아닌 별도의 독립된 시스템으로 개발하여 반도체 물류 공정의 확장 등 상황에 따라 알고리즘을 유연하게 적용할 수 있도록 하였다.

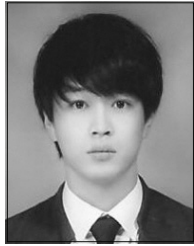
SA 시스템의 기능 검증을 위해 OCS Main 시스템과 통합 시험을 수행하였다. Dijkstra 알고리즘 적용을 위해 반도체 물류 라인과 OHT 이동을 노드와 엣지를 가지는 방향성 있는 그래프로 표현하였다. 여섯 가지의 기본 시나리오와 두 가지의 예외 시나리오에 대하여 SA 시스템이 정상적인 작업 계획 여부에 대한 통합시험을 수행하였다. 모든 시나리오에 대하여 SA 시스템이 정상적인 출력을 확인하였고, 제안하는 시스템이 OHT 제어 및 관리에 적합한 시스템임을 검증하였다.

개발된 SA 시스템과 전체 반도체 물류 시스템은 CPS 구축의 기초 사례로 의미가 있지만 본 연구에서는 실제 반도체 물

류 라인을 간소화하고 몇 가지 사항을 가정하여 개발하였다. 추후에는 확장된 반도체 물류 라인에 적용이 가능하도록 제안 시스템의 기능을 보완하여 다양한 반도체 물류 현장에 적용할 수 있을 것으로 기대한다.

References

- [1] H. S. Song, J. Y. Lee, and T. G. Kim, "DEVS-based Modeling Simulation for Semiconductor Manufacturing Using an Simulation-based Adaptive Real-time Job Control Framework," *Journal of the Korea Society for Simulation*, Vol.19, No.3, pp.45-54, 2013.
- [2] J.T. Lin, F.K. Wang, and C.K. Wu, "Connecting transport AMHS in a wafer fab," *International Journal of Production Research*, Vol.41, No.3, pp.529-544, 2003.
- [3] F. K. Wang and J. T. Lin, "Performance evaluation of an automated material handling system for a wafer fab," *Robotics and Computer-Integrated Manufacturing*, Vol.20, No.2, pp.91-100, 2004.
- [4] K. W. Lee, W. G. Lee, and S. C. Park, "Virtual Environment Hardware-In-the-Loop Simulation for Verification of OHT Controller," *Journal of the Korea Society for Simulation*, Vol.28, No.4, pp.11-20, 2019.
- [5] K. Magzhan and H. M. Jani, "A review and evaluations of shortest path algorithms," *International Journal of Scientific & Technology Research*, Vol.2, No.6, pp.99-104., 2013.
- [6] U. Meyer, "Single-source shortest-paths on arbitrary directed graphs in linear average-case time," In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp.797-806, 2001.
- [7] K. Bartlett, J. H. Lee, S. Ahmed, G. Nemhauser, J. Sokol, and B. S. Na, "Congestion-aware dynamic routing in automated material handling systems," *Computers & Industrial Engineering*, Vol.70, pp.176-182, 2014.
- [8] B. Xiao, J. Cao, Z. Shao, and E. H.-M. Sha, "An efficient algorithm for dynamic shortest path tree update in network routing," *Journal of Communications and Networks*, Vol.9, No.4, pp.499-510, 2007.
- [9] I. Hwang and Y. J. Jang, "Q(λ) learning-based dynamic route guidance algorithm for overhead hoist transport systems in semiconductor fabs," *International Journal of Production Research*, Vol.58, No.4, pp.1199-1221, 2020.
- [10] H. J. Kim, D. E. Lim, and S. M. Lee, "Deep learning-based dynamic scheduling for semiconductor manufacturing with high uncertainty of automated material handling system capability," *IEEE Transactions on Semiconductor Manufacturing*, Vol.33, No.1, pp.13-22, 2020.



이 복 주

<https://orcid.org/0000-0002-5306-5292>
e-mail : bokju618@koreatech.ac.kr
2014년 한국기술교육대학교 컴퓨터공학부
(학사)
2018년 한국기술교육대학교 컴퓨터공학과
(석사)

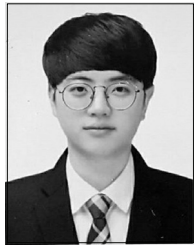
2018년 ~ 현 재 한국기술교육대학교 컴퓨터공학과 박사과정
관심분야 : Image Processing, Artificial Intelligence



한 경 아

<https://orcid.org/0000-0002-1606-4602>
e-mail : kyungahh@koreatech.ac.kr
2008년 충남대학교 전산학(박사)
1995년 KT 기술원 책임연구원
2014년 KAIST 전산학부 연구원
2017년 ~ 현 재 한국기술교육대학교
산학협력교수

관심분야 : Machine Learning, Data Science



박 희 문

<https://orcid.org/0000-0002-2622-100X>
e-mail : phm1321@koreatech.ac.kr
2017년 한국기술교육대학교
전기전자통신공학부(학사)
2020년 한국기술교육대학교
전기전자통신공학과(석사)

2021년 ~ 현 재 한국기술교육대학교 컴퓨터공학과 박사과정
관심분야 : Simulation, Artificial Intelligence



서 경 민

<https://orcid.org/0000-0003-1017-1674>
e-mail : kmseo@koreatech.ac.kr
2014년 KAIST 전기및전자공학(박사)
2019년 (주)대우조선해양 특수성능연구소
책임연구원
2019년 한국생산기술연구원

창의엔지니어링센터 선임연구원
2019년 ~ 현 재 한국기술교육대학교 융합학과 조교수
관심분야 : Simulation, Digital Twin, Cyber-Physical System



권 용 환

<https://orcid.org/0000-0003-4990-2913>
e-mail : yhkwon@ihumin.co.kr
2000년 울산대학교 생물학과(학사)
울산대학교 전자계산학
(학사 부전공)
2019년 ~ 현 재 (주)휴민텍 소프트웨어팀
부장

관심분야 : OHT System, Digital Twin, Machine Learning