The Optimisation and Integration of AGVs with the Manufacturing Process

by

Fengjia Yao

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

WMG, University of Warwick

January 2022

THE UNIVERSITY OF
WARWICK

Declarations

This thesis is submitted to the University of Warwick in support of my application for the degree of Doctor of Philosophy. It has been composed by myself and has not been submitted in any previous application for any degree.

The work presented (including data generated and data analysis) was carried out by the author.

........[redacted]. (Signed)

Fengjia Yao

15/01/2022 .... (Date)

Abstract

In recent years, the manufacturing environment, driven by the growth of advanced technologies and the increasing demand for customised products, has becomes increasingly competitive. In this context, manufacturing systems are now required to be more automated, flexible and reconfigurable. Thus, Autonomous Guided Vehicle (AGV), as a key enabler of dynamic shop floor logistics, are being increasingly widely deployed into the manufacturing sector for the lineside materials supplying, work-in-progress transportation, and finished products collection.

A large number of companies and institutions are researching on different AGV systems to integrate AGVs-based shop floor logistics with manufacturing equipment and processes. However, these AGV systems are typically equipped with various communication protocols and utilise ad-hoc communication methods. They lack a generic framework to integrate the AGV systems into the manufacturing systems with minimal engineering effort and system reconfiguration. Current scheduling optimisation methods for multiple AGVs in shop floor logistics now support effective task allocation, shortest route planning, and conflict-free supervision, allocating the delivery tasks based on the location and availability of AGVs. However, these current methods do not give enough consideration to real-time operational information during the manufacturing process and have difficulties in analysing the real-time delivery requests from manufacturing work stations. This not only reduces the efficiency and flexibility of the shop floor logistics, but also significantly impacts on the overall performance of manufacturing processes.

This thesis presents a generic integration approach, called Smart AGV Management System (SAMS), to support the integration of AGVs with manufacturing processes. The proposed framework enables enhanced interoperability between AGVs-based shop floor logistics and the manufacturing process through a generic data-sharing platform. Moreover, a Digital Twin (DT)-based optimisation method is developed in SAMS that can simulate and analyse the real-time manufacturing process to schedule AGVs for optimising multiple objectives, including the utilisation of work stations, delivery Just-in-time (JIT) performance, charging of AGVs and overall energy consumption.

This approach is experimentally deployed and evaluated from various perspectives to identify its integration and optimisation capabilities during the reconfiguration and operational phases. The results show that the proposed integration framework can enable a more effective integration with manufacturing process compared to traditional integration methods. In addition, the results demonstrate that the proposed optimisation method can schedule and reschedule AGV-based shop floor logistics when facing a range of system disruptions.

Acknowledgments

There are many fellows and friends I would like to thank for supporting and contributing to my research. Without them, this research work would not have been completed.

I would like to thank my supervisors, Professor Robert Harrison, and Associate Professor Bilal Ahmad, who have continuous supported me in my PhD study and the related project research. And they helped me to find a clear direction for my PhD research. They shared their knowledge of manufacturing and automation systems to support me to become a good researcher. And they helped me for finding financial support at the beginning of my PhD study. Without it, I could not imagine continuing my PhD study. I am grateful to have worked with them for the last five years and looking forward to contributing to future work.

Many thanks to Associate Professor Daniel Vera for his insightful comments and encouragement. He enlightened me with impressive ideas when I was in the dark. His supervision and supports helped me have a better understanding of manufacturing and system integration practically.

Additionally, I would like to thank all the colleagues in the Automation Systems Group in WMG, the University of Warwick, in particular Alexander Keller, Dr. Jiayi Zhang, Georgina Haslop, Mohammed Waseem, Wajid Azam, Bethany Haynes, and Dr. Young Saeng Park and all those who supported in hardware, programming, electronics and administration to help me with my day-to-day research.

I wish to express my gratitude to my parents, who have given me selfless supports in my studies and lives. Their endless support in financial and well-being help me in all my studies. English vocabulary is not enough to express my gratitude for everything they did.

Finally, I would like to thank my wife, Yue Li, for constantly supporting and listening to me when I was in a tough time, for the sacrifices made for me to pursue the PhD degree.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AGV Autonomous Guided Vehicle

AI Artificial Intelligent

API Application Programming Interface

APS Advanced Planning and Scheduling

ASS Asset Administration Shell

CDF Cumulative Distribution Function

COVID-19 Coronavirus Disease 2019

CPS Cyber-Physical System

CPPS Cyber-Physical Production Systems

DES Discrete-Event Simulation

DRM Design Research Methodology

DT Digital Twin

ERP Enterprise Resource Planning

FMS Flexible Manufacturing System

GA Genetic Algorithm

HMI Human-Machine Interface

HTTP Hypertext Transfer Protocol

IIoT Industrial Internet of Things

IML Integrated Manufacturing & Logistics

IMU Inertial Measurement Unit

IoT Internet of Things

IT Information Technology

JIT Just-in-time

JSON JavaScript Object Notation

KPIs Key Performance Indicators

M2M Machine-to-machine

MES Manufacturing Execution System

MOPSO Multiple Objective Particle Swarm Optimisation

MS Microsoft

NSGA-II Non-dominated Sorting Genetic Algorithm II

ODBC Open Database Connectivity

OEE Overall Equipment Effectiveness

OLE Object Linking and Embedding

OPC Open Platform Communications

OPC UA Open Platform Communications (OPC) Unified Architecture

OPC DA OPC Data Access

OT Operational Technology

PC Personal Computer

PLC Programmable Logic Controller

RAMI 4.0 Reference Architectural Model Industrie 4.0

REST Representational state transfer

RFID Radio-frequency Identification

ROI Return on Investment

ROS Robot Operating System

SAMS Smart AGV Management System

SLAM Simultaneous Localisation and Mapping

SMEs Small and Medium-sized Enterprises

SQL Structured Query Language

TCP/IP Transmission Control Protocol/Internet Protocol

WCL WITNESS Commands Language

UML Unified Modelling Language

SDK Software Development Kit

List of Publications

1. Yao, F., Keller, A., Ahmad, M., Ahmad, B., Harrison, R. and Colombo, A.W., 2018, July. Optimizing the scheduling of autonomous guided vehicle in a manufacturing process. In 2018 IEEE 16th International Conference on Industrial Informatics (INDIN) (pp. 264-269). IEEE.

2. Yao, F., Alkan, B., Ahmad, B. and Harrison, R., 2020. Improving just-in-time delivery performance of IoT-enabled flexible manufacturing systems with AGV based material transportation. Sensors, 20(21), p.6333.

Chapter 1

Introduction

1.1 Research background

With the growth of advanced technologies and increasingly customised product demand, the manufacturing environment has become volatile and highly competitive. Thus, manufacturing systems are expected to be agile, flexible, and interoperable in order to carry out production processes effectively. Manufacturing systems are required to adapt to the production disturbance with minimal human intervention. With the emergence of Industry 4.0 and the associated enabling technologies, such as Industrial Internet of Things (IIoT), Cyber-Physical System (CPS), intelligent robotics, Artificial Intelligent (AI) and DT, a smart shop floor can be realised where manufacturing systems can be dynamically reconfigured in the event of a disturbance or change in product demands.

One of the important aspects of a smart shop floor is dynamically adaptable logistics. In a smart shop floor, manufacturing systems and logistics are closely coupled through end-to-end horizontal and vertical integration. The aim is to develop an intelligent, flexible and collaborative manufacturing environment where work stations are efficiently served by shop floor logistics, enabled by using real-time production process data.

The AGV is considered a key enabler of dynamically adaptable shop floor logistics, which can be deployed to carry out versatile tasks on a manufacturing shop floor. Because of the AGVs' advantages, including flexibility, cost reduction, reliability, safety, productivity and easier manipulation, they are now widely deployed in the manufacturing sector, and feature strongly in research papers from various academic fields. They are often named differently based on their functionality and working situation, such as Autonomous Indoor Vehicles (AIVs), Autonomous Intelligent Vehicles (AIVs), Autonomous Mobile Robots (AMRs) and Unmanned Ground Vehicles (UGVs). In this thesis, the name AGV is used, which in this context means a driverless indoor vehicle travelling around work stations autonomously to support transportation on a factory shop floor. At their early stage, AGVs were designed to follow pre-planned paths built with magnetic tapes or coloured stripes with limited flexibility[1]. With the advances in associated technologies, AGVs can now self-navigate within a pre-mapped environment[2], which offers greater flexibility and agility. Moreover, to build a highly flexible, responsive, and productive manufacturing system, the AGVs are also implemented to integrate with collaborative robots, pallet lifts, shelves, and conveyors to enable them to interact with automated stations autonomously.

Furthermore, to manage cooperation among multiple AGVs in the same working environment, fleet management software applications are developed to supervise and plan AGVs routing, aiming to minimise the travel time, allocate suitable AGVs and plan conflict-free paths. However, the current fleet managers cannot adequately analyse the real-time manufacturing process to provide the appropriate schedules for AGVs, such as optimising delivery time and allocating suitable tasks. The AGV-based logistics delivery time, including earliness and lateness, can significantly impact the overall takt time of the manufacturing process [3]. For example, an early delivery service causes waiting of AGVs and the poor utilisation of AGVs, while the lateness causes the related work station to have to wait, leading to a loss in production. Thus, in the smart factory, integration of multiple-AGV-based shop floor logistics with the Information Technology (IT)/ Operational Technology (OT) structure of the manufacturing process is essential. It provides an opportunity to optimise the AGV-based logistics delivery services by analysing the shop floor data and predicting the manufacturing behaviours.

Numerous works are reported in the literature which studied scheduling multiple AGVs in manufacturing systems based on mathematical or simulation models of AGVs without considering the overall real-time manufacturing process. However, due to the dramatically increasing complexity of current manufacturing systems, these approaches are not suitable for solving the scheduling problems and scenarios in which AGVs need to cooperate with various types of automation systems, such as standalone stations, robotics, and conveyors. The scheduling methodology not only needs to consider the job priorities but also needs to schedule the AGV dispatching time considering the takt time of each work station. DT technology can be used to develop a virtual model of the physical manufacturing system, which monitors and analyses live operational data and simulates the overall manufacturing process in real-time or near real-time. The DT method can provide the operational information of integrated AGV-based shop floor logistics and manufacturing process for scheduling/ re-scheduling the delivery services to respond to the interruptions of the manufacturing system and maintain high production performance.

1.2 Problem definition

There are still some challenges of the current technology to efficiently optimise and integrate the AGVs-based shop floor logistics with manufacturing systems.

Firstly, in the current AGV market, different AGV systems have different degrees and forms of connectivity. For instance, the MiR AGV [4] is controlled through Representational state transfer (REST) Application Programming Interface (API), and the Omron AGV [5] is controlled through its client. Their APIs-based software applications need to be developed to integrate the AGV systems with the shop floor manufacturing systems. Also, the current integration approaches are mainly focused on the physical interaction between the AGV and work stations to achieve the product transporting. However, this integration cannot properly support the control signals and data communication between the AGV and work stations. Thus, a generic integration framework needs to be developed to support the multiple AGVs cooperating with manufacturing systems, and considering IT and OT convergence.

Secondly, the current AGV manufacturers have developed various software applications to manage fleets of their AGVs. These applications focus on the traffic of AGV fleets, the task allocation, and the behaviour of AGV movement. Also, due to the limited data interaction between the AGVs and the manufacturing system, they lack the capability to schedule the AGV delivery tasks by analysing the manufacturing process in real-time.

1.3 Research motivation

The high-level automation, seamless connectivity, information exchange, and big data analytic in smart factory offer the potential to enable manufacturing systems to become more flexible, organised and intelligent. As one of the key components in the manufacturing system, AGV-based shop floor logistics is now being widely developed to support raw material delivery, work-in-process delivery and finished products collection on the shop floor. The AGV system provides an opportunity to create a flexibly and customisable intralogistics system with efficiency and reliable delivery services [6]. Especially, due to the Coronavirus Disease 2019 (COVID-19) pandemic, more AGV systems have been implemented into the manufacturing shop floor because of their autonomous capability and the associated reduction of the human operators to help to keep social distancing [7].

However, the current manufacturing system lacks the integration strategy to enable full cooperation between the AGV systems and the manufacturing process, which is needed to prevent the bottleneck in JIT delivery for production line-side supplying. The better performance of JIT delivery can keep the manufacturing processes running smoothly, reduce energy waste and improve the overall manufacturing system performance.

Thus, it is necessary to develop a generic framework for integrating AGV systems into shop floor logistics to cooperate with automation stations in the manufacturing system. Furthermore, the scheduling optimisation capability is significant in this framework to monitor the progress of the manufacturing process and optimise the tasks of AGV to improve the JIT delivery performance in shop floor logistics.

In this thesis, the SAMS is conceived to integrate AGVs with manufacturing systems and to schedule the AGV tasks by considering the real-time manufacturing processes. The SAMS can be deployed into the current manufacturing system to collect the operational information and integrate it with the Manufacturing Execution System (MES). In the SAMS, a DT-based process simulation is created to replicate the manufacturing process. A hybrid optimisation algorithm combined with the Non-dominated Sorting

Genetic Algorithm II (NSGA-II) and DT-based Discrete-Event Simulation (DES) model is designed to optimise the utilisation of AGVs and to schedule the AGV delivery tasks or execute re-scheduling tasks when a production abnormality is captured. Thus, the proposed methodology seeks to schedule the AGV-based shop floor delivery performance by considering the real-time manufacturing process to increase productivity, flexibility, utilisation of work stations and JIT performance.

1.4 Research questions and objectives

Based on the problem definition and the research motivation, three research questions have been proposed below:

1) How have the AGV systems been integrated and optimised in the shop floor manufacturing system in the current industrial factory and research, and what are the shortcomings of the integration and optimisation methods?

2) How to improve the integration of AGV systems with the shop floor manufacturing process to increase the flexibility of the manufacturing system?

3) How can the DT technology be used in the optimisation of the AGVs-based shop floor logistics during the real-time manufacturing process for improving the overall production Key Performance Indicators (KPIs)?

Driven by the research questions above, this research aims to develop a framework to support the integration of the AGV systems with the manufacturing processes on the shop floor. Also, this framework allows monitoring of the shop floor operational information and the scheduling optimisation of the AGVs' delivery tasks.

Moreover, this research aims to optimise the performance of JIT line-side supply and minimise the overall energy consumption of AGVs for saving the energy cost and improving the flexibility, reconfigurability, and agility of the shop floor logistics. The details of the research objectives are illustrated below:

1. Identify the current challenges faced by AGV systems working on manufacturing shop floors, and understand the research gaps.

2. Develop a framework that can support sharing the information between work stations, AGVs, and human operators to fully integrate the AGV systems with the manufacturing processes on the shop floor.

3. Design a scheduling optimisation methodology to optimise the scheduling of the delivery tasks for AGVs in shop floor logistics and manage the charging threshold of AGVs during product delivery.

4. Develop an engineering tool based on the proposed methodology and implement it into a suitable shop floor manufacturing process to evaluate and identify its capabilities and performance.

1.5 Research methodology

This thesis uses the Design Research Methodology (DRM) [8] to guide the research design. The methodology is divided into four stages as follows:

1. Research Clarification
The research clarification stage is to identify a realistic objective for the research. The literature in academics and industrial applications will be reviewed and critically anal-

ysed to understand research problems and hypotheses, and to identify research gaps.

2. Descriptive Study I

In the descriptive study I stage, the author aims to better comprehend the current manufacturing situation and clarify ideas to improve the existing manufacturing performance. These are reached through a thorough literature review and the author's related research experiences.

3. Prescriptive Study

In the prescriptive study stage, a comprehensive SAMS architecture-based methodology is proposed, and all function modules in SAMS are described in detail. This methodology improves the current manufacturing integration situation, optimising the integration of AGV-based logistics with a flexible manufacturing system. Also, the proposed SAMS architecture is further developed into a "plug-and-play" engineering application, which is implemented and tested in the Integrated & Manufacturing Logistics rig, as described in Chapters 4 - Case Study.

4. Descriptive Study II

In the descriptive study II stage, the application of the developed methodology in the prescriptive study is evaluated. The application evaluation is to identify the application accomplishment regarding the integration of AGV-based logistics with manufacturing processes, and the resultant performance optimisation. It is carried out in Chapter 5 - Evaluation of the Proposed Methodology.

The overview of DRM based research outline is shown in Figure 1.1.

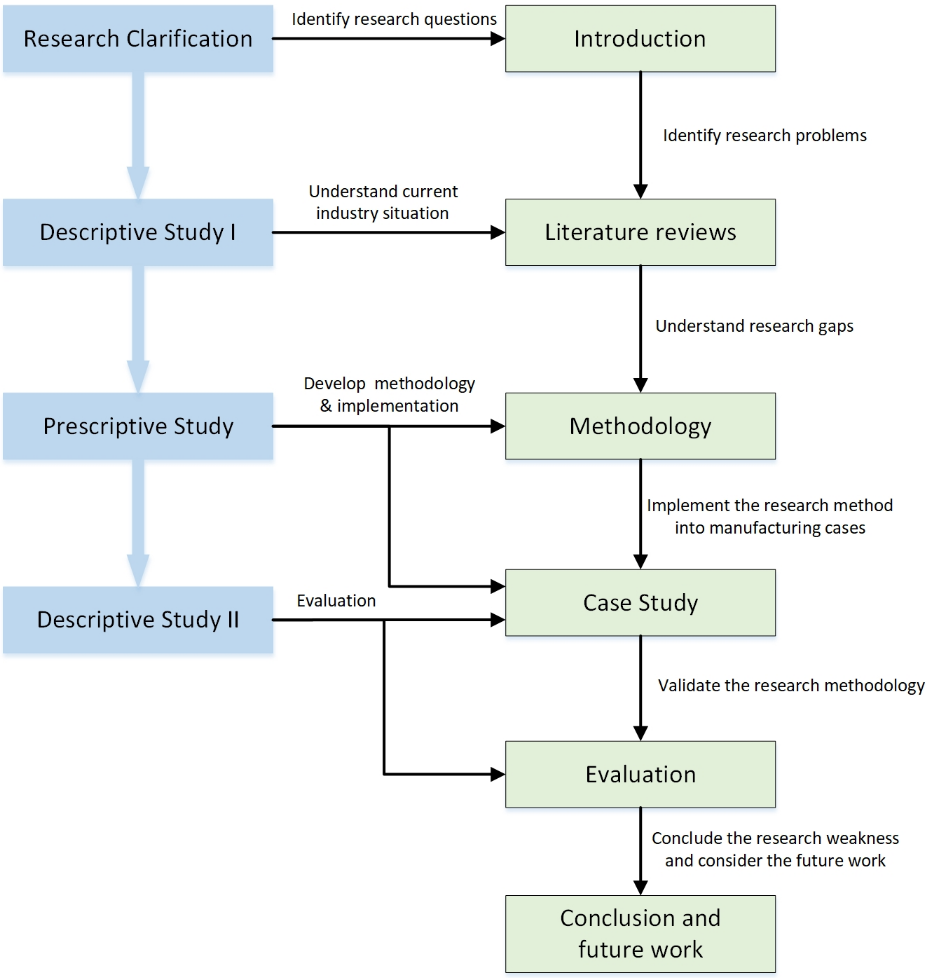

Figure 1.1: The overview of research methodology

1.6 Research outline

For this PhD research, the thesis structure is depicted in Table 1.1:

Table 1.1: The thesis structure

Chapter	Title	Content	Outcomes
1	Introduction	Summarise the research background, motivation, focus and thesis structure.	Objectives and structure of PhD research
2	Literature Review	Review the literature of manufacturing system, AGV systems integration and scheduling.	Research gaps
3	Methodology	Introduce the methodology of scheduling the AGV-based shop floor logistics in the manufacturing process	Research methodology and framework
4	Case Study	Develop the software application and carry out an experiment	Case study to validate the proposed methodology
5	Evaluation	Evaluate the proposed methodology in different scenarios	Evaluate the performance of the proposed methodology
6	Conclusion and Future work	Summarise the objectives and contribution of research, and discuss the future works	Contribution of presented methodology and future research plan

Chapter 1 introduces the overview of PhD research background, motivation, questions, and objectives. Chapter 2 comprehensively reviews the literature on AGV development and the integration of AGVs with manufacturing processes on the shop floor. Chapter 3 presents a methodology and framework to support the integration of AGV systems and the optimisation of AGV task scheduling. In Chapter 4, the case studies are developed to demonstrate the capability of the presented methodology. In Chapter 5, the performance of the proposed methodology is evaluated in different scenarios. Finally, Chapter 6 discusses the contribution of this PhD research and future work.

Chapter 2

Literature Review

2.1 Introduction

Nowadays, the manufacturing environment is more competitive and customised than ever before. Flexibility and agility are becoming significant requirements for manufacturing systems. With a high level of flexibility and agility, the manufacturing system can be operated more optimally. AGVs have an important role to play to enable greater flexibility in manufacturing systems. This is evident from the fact that its market is globally growing at a fast pace in various sectors including manufacturing to provide autonomous and flexible logistics systems. According to Research and Markets Report [9] the global automated guided vehicle market was worth USD 2.5 billion in 2020 and is projected to reach USD 13.2 billion by 2026. In parallel, the research on the integration of AGVs in the manufacturing industry is also growing rapidly. The Clarivate Analytics sources show the increase of the citations and publications with keywords, "shop floor logistics" and "AGV" (Figure 2.1). Especially in the recent five years, there has been a significant increase in interest in both the academic and industry fields.

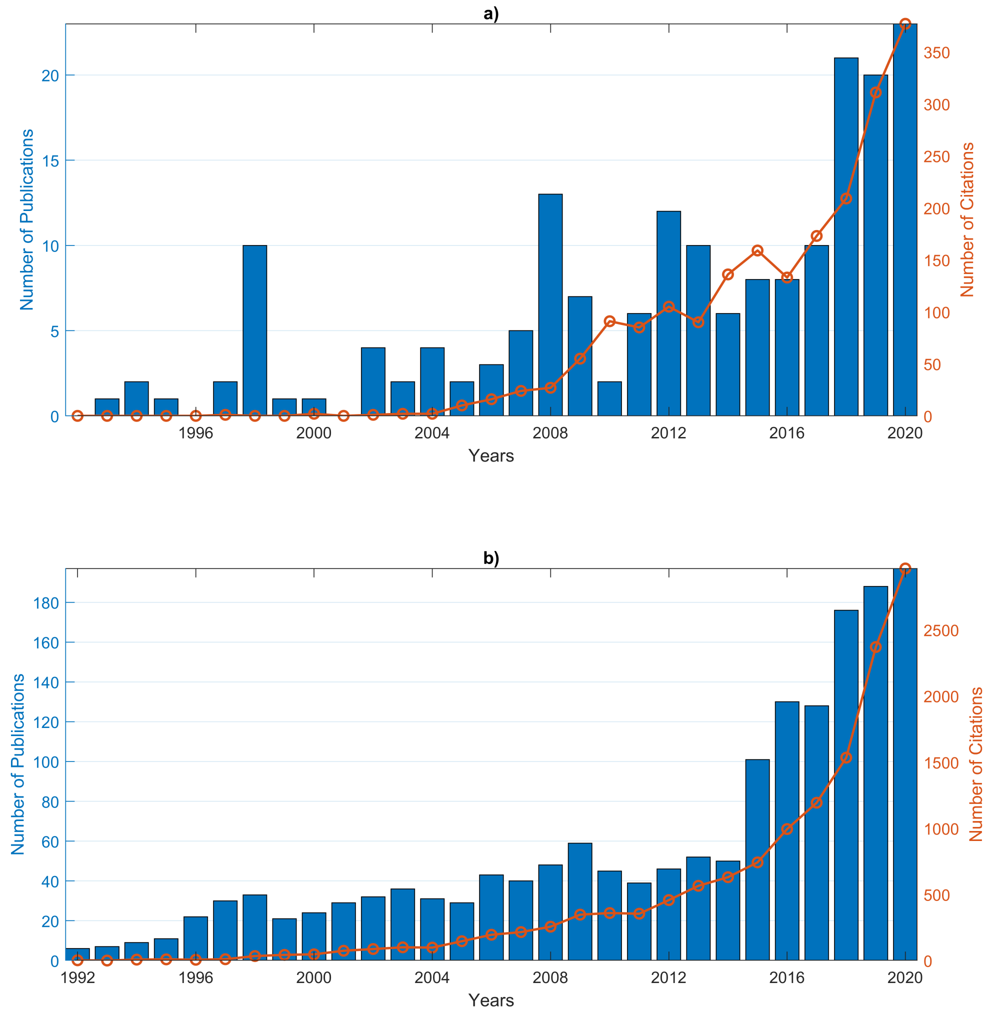

Figure 2.1: The literature review summary: a) Numbers of citations and publications with keywords “shop floor logistics” over time. b) Numbers of citations and publications with the keyword “AGV” over time.

The literature reviews cover the studies from the recent three decades (1995-2021). Several keywords are considered to search the literature, including “AGV systems”, “development of AGVs”, “applications of AGV in the manufacturing sector”, “integration of AGV systems", "integration of manufacturing systems", "fleet manager of AGVs", "scheduling of multiple AGVs", "optimisation of AGV system on the shop floor" and "digital twin in scheduling AGVs", via Google Scholar and ResearchGate. These literature reviews are evaluated and published through these electronic databases: IEEE Xplore, ScienceDirect, Directory of Open Access Journals, JSTOR, British Standards Institution, Scopus, SciFinder, Web of Science, and ACM Digital Library. Additionally, the library search of the University of Warwick is used to find related books, standards and dissertations. Moreover, the Connect Paper [10] search engine is used to discover the most relevant research and derivative works.

Thus, this chapter discusses: 1) a comprehensive review of the AGV-based shop floor logistics; 2) a review of the scheduling methods for multiple AGVs; 3) a review of DT technology in the scheduling of AGVs. The chapter identifies the gaps in optimally scheduling AGV-based shop-floor logistics and its integration in manufacturing industries, and aims to provide a clear understanding of the state-of-the-art in DT-based scheduling methodologies.

2.2 The state-of-the-art of AGV applications in the manufacturing sector

The concept of smart factories providing high flexibility, agility, and efficiency has become very prominent in recent years. Smart factories are expected to be able to self optimise, adapt to interruptions and minimise human interventions [11]. The AGV is considered a key enabler for the automation of versatile tasks in the intralogistics of a smart factory. With the enhancement of navigation and control technologies, AGVs are becoming widely deployed on shop floors for materials delivery and handling [12].

Exploiting the AGVs' potential for collaboration, robustness, and flexibility they have started to replace the manual-forklift to reduce labour costs and improve manufacturing efficiency.

2.2.1 The AGV applications in the manufacturing sector

In smart factories, manufacturing systems are required to be effective, responsive, and flexible. Intelligent robots, like AGVs and collaborative robots, have an important role to play in helping to achieve this, and they must be widely applied on the manufacturing shop floor. AGVs can enable material handling systems and manufacturing systems to integrate seamlessly to increase productivity and efficiency [13]. Recently, numerous use cases have described the implementation of the AGV system into the manufacturing environment. For example, in an automotive company in the Czech Republic [14], AGVs collect finished car frames from the production line and deliver them to the warehouse. Also, Theunissen et al. [15] developed an Radio-frequency Identification (RFID) technology-based AGV system serving on a manufacturing shop floor to show the flexibility of materials handling.

Mehami et al. [6] demonstrated multiple AGVs working in a factory internal logistics system by using RFID technology for tracking and controlling the movement. In these applications, the AGV system helps in reducing production time by delivering the shop floor material for production lines in a short time. It has shown that the AGV based logistic system makes the manufacturing industry more flexible, reconfigurable, and customisable.

In the automotive manufacturing sector, AGVs can similarly enable more modular, automated, production processes in a smart shop floor environment [16]. In Slovakia [17], CEIT AGVs transport car engines, gearboxes, doors, and bodies during production and connect all the main work stations in the shop floor factory. The AGV also replaces traditional conveyor systems in the factory production line, improving system adaptability and flexibility [18]. Moreover, in a highly automated factory AGVs can replace manual trucks for material delivery [19]. Smart AGVs can track and transport items between work stations to ensure the process is traceable, undisrupted and robust. In the automotive industry assembly lines, Cech et al. [20] introduced the AGV system in a car assembly line for irregular consumption materials delivery. It not only demonstrated the efficiency improvement but also showed the challenge of implementing a large fleet of AGVs. Also, Zhang et al. [21] designed a CPS-based control model to support to the implementation of AGVs in the shop floor for materials handling. The overall production efficiency was improved after involving the AGV system in shop floor logistics.

A large number of manufacturing companies are also adapting AGVs for various applications on their shop floors. Various use cases in manufacturing companies are enumerated in Table 2.1

Table 2.1: The lists of AGV use cases in manufacturing companies

Company	AGV type	Use cases	benefits
VW AG (CEIT) [22]	Tractor	Work-in-progress movement: Deliver engines and car bodies.	Production time reduced 25% in SEAT factory.
Toyota (BT Autopilot) [23]	Truck, carts, load carrier	Raw material delivery: Repetitive work in the warehouse and shop-floor for horizontal transporting.	The ROI was approximately in 2 years.
Mercedes-Benz (KUKA) [24]	Omnidirectional mobile platform	Work-in-progress movement: Transporting truck components for the assembly line.	Short the truck assembled process time to 8.9 minutes per truck.
Ford, Spain (MiR) [25]	Mobile platform	Raw material delivery: Deliver parts and welding materials for human workers and robotic stations.	Continuously work up to 40 work hours.
Boeing (Fori automation) [26]	Mobile platform	Work-in-progress movement: Transport tools between stations in aircraft wing assembly line.	Eliminated the crane usages and increased shop floor safety.
Airbus (KUKA) [27]	Omnidirectional mobile platform	Work-in-progress movement: Carry aircraft components up to 90 tonnes in production hanger.	Continuously operation for 48 hours
Audi (Grenzebach) [28]	Load carrier	Work-in-progress movement: Delivery vehicle components in shop floor logistics.	30 AGVs are running in the factory at the same time.
Dell (Geek+) [29]	Mobile platform	Work-in-progress movement: Load, deliver and unload products in the warehouse	Increase the ten times of packing efficiency.
AER Manufacturing (Dematic) [30]	Tugger	Finished product delivery: Transport finished engines from shop floor to ship	Increase the production efficiency to 220 engines per day, and the expected ROI is around 18 months.

Furthermore, the current AGVs are mainly navigated based on the Simultaneous Localisation and Mapping (SLAM) technology, by which the AGVs can map the shop floor layout and localise themselves simultaneously. Thus, the AGVs are not limited by the markers or lines, and they can travel around the shop floor environment with more flexibility. AGVs are not only used to transport material, handle pallets or collect the finished goods. As smart mobile robots, they are equipped with end-effectors to autonomously interact with work stations or used as mobile work stations. An AGV system was developed by Cronin et al. [31] to load and unload materials for conveyors. This design helps the company keep competitive by increasing manufacturing productivity and building up a flexible shop floor. Also, Cronin et al. [32] reviewed the AGV applications in manufacturing sectors and pointed out that the AGV connects the manufacturing cells more flexible than the conveyor production line. It indicates the necessity of AGV-based shop floor logistics because it supports the connectivity and flexibility of the manufacturing systems. Various application examples of AGV on a manufacturing shop floor is shown in Figure 2.2

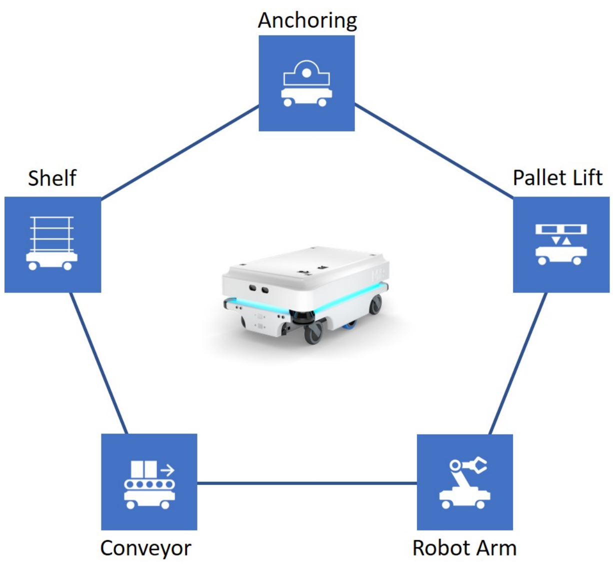

Figure 2.2: Examples of AGVs with different functionalities

2.2.2 The advantages of implementing the AGV on the shop floor

Key advantages of deploying AGVs on a manufacturing shop floor are summarised below:

1) Cost reduction. AGVs can replace labour used for materials handling, which reduces the yearly and indirect costs. An example from Kollmorgen [33] shows the cost comparison between the manual forklift and AGVs in Figure 2.3. It shows that in this particular case the AGV solution payback is one year and six months.

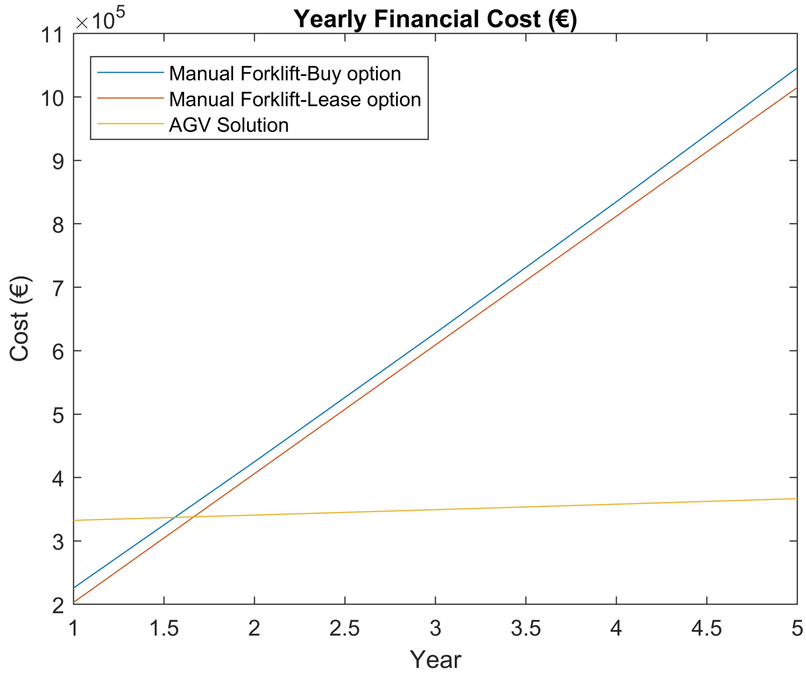

Figure 2.3: The comparison of yearly financial cost of manual forklift and AGVs

2) Reliability. In shop floor logistics, operational mistakes are often caused by a wrong pick or destination delivery location, which could raise the unexpected breakdown time in manufacturing [34]. GE Digital [35] reports that human errors cause 23% of unplanned downtime. In contrast, AGV could keep aware and operate precisely after being appropriately programmed. So AGV-based shop floor logistics can increase the reliability of production processes.

3) Flexibility. The shop floor layout does not limit AGVs, and AGVs can travel dynamic paths during the delivery and handling of materials or products. AGVs can be customised to perform various tasks such as quality checks, safety inspection and materials handling. Moreover, the AGV system can replace the traditional conveyor systems and adapts to the MES or manufacturing system, which helps AGVs deal with dynamic delivery tasks. Thus, the AGV system can provide more flexible and dynamic transportation workflows.

4) Safety. AGVs are designed and programmed to work in a safe manner. They are equipped with safety sensors, such as laser scanners, cameras, ultrasonic sensors, and bumpers so that they can be operated around humans and work stations safely and collaboratively.

5) Productivity. Better reliability, flexibility and safety inevitably improve manufacturing productivity. From the manufacturing company use cases (mentioned in Table 2.1) it is evident that AGVs can dramatically reduce product delivery times and keep the longer working hours on the shop floor.

2.2.3 The key enabling technologies for integrating AGV on the shop floor

On the shop floor, a number of advanced technologies, such as IIoT, CPS, intelligent robotics, AI and DT, are deployed and integrated to establish an intelligent manufacturing environment with the aim of improving the performance, flexibility and agility of manufacturing systems [36]. Also, in the smart factory, work stations are considered autonomous entities that can be operated independently, communicate with each other, and update the status and processes automatically [37]. The use of AGV as a smart mobile robot with conflict-free routing and scheduling abilities to integrate with the manufacturing process on the shop floor becomes of significant importance in such a manufacturing environment. It can interact with autonomous stations, human operators, and production lines to achieve a high level of automation by transporting materials and products on the shop floor.

Key aspects of the AGVs-based shop floor logistics include agility, flexibility, interoperability and intelligent decision-making, which enables efficient performance and lower manufacturing costs compared to traditional shop floor logistics [38, 39]. According to a recent study [40], these key technologies, including the Cyber-Physical Production Systems (CPPS), IIoT, DT, AGVs and Intelligent robots, are required to build this new form of shop floor logistics.

1) CPPS

As one of the vital technologies on the shop floor, CPPS contain autonomous and cooperative components and subsystems. They are connected and communicate through all levels of production, from the automated stations to the production and logistics [41]. Potential benefits of CPPS are in production process optimisation, product customisation, resource-efficient production, and human-centred production processes [42]. Thus, the CPPS has been widely involved in manufacturing automation, and it stimulates to the development of reconfigurability of autonomous machines and software control systems [43]. In recent years, a significant number of papers have been published on CPPS. For example, Engelmann et al. [44] mentioned that CPPS contributes to increasing the transparency of the manufacturing process by intelligent analysis of sensor data to improve the Overall Equipment Effectiveness (OEE). Blume et al. [45] applied CPPS to cooling towers of technical building services to improve system understanding and identify the critical factors for cooling tower performance.

However, there are still some challenges for implementing the CPPS on the shop floor. Towers of Hanoi is presented by [46] to show the challenges in data sharing, functionality sharing, and collaborative functionality testing. Similarly, Leitao et al. [47] stated CPPS challenges in capacities, management, engineering, ecosystems, infrastructures, and information systems.

2) DT

With computer simulation and 3D modelling development, simulation is becoming a powerful technology for understanding and analysing dynamic manufacturing environments [48]. DT is one of the new simulation paradigms. It is a virtual copy of the physical system and can potentially communicate with the physical world in real-time [49]. Because of the implementation of IIoT technologies in the factory, a DT can access physical system information, which can be used for monitoring assets and carrying out process optimisation and predictive maintenance [50]. DT was first applied by NASA to mirror the life of air vehicles for improving its physical model [51]. Tao and Zhang [52] proposed a DT shop floor framework to support a high fidelity and continuous interconnection between the shop floor and virtual world to achieve smart interaction and intelligent control during manufacturing processes.

3)AGVs and Intelligent Robots

AGVs are becoming more flexible, intelligent, and cooperative. They can interact with the human operator and even learn technical skills [52]. Today, AGVs and Intelligent Robots are not only following pre-defined tasks, but they can also learn to carry out tasks using artificial intelligence, e.g., machine learning, deep learning, and reinforcement learning [53]. Many companies are researching AGVs and robots, such as Kuka, Omron, Universal Robots and Swisslog, to develop autonomous vehicle-based shop floor logistics systems. Also, in the smart factory, the AGVs and robot systems are needed which are capable of understanding the manufacturing processes via self-learning, self-optimisation and self-maintenance to perform the appropriate actions [54].

In summary, these technologies are essential to AGV-based shop floor logistics. Currently, AGVs predominantly perform repetitive deliveries, reliably and productively, and they can also offload the ergonomically-challenging work from human operators, and carry out tedious tasks autonomously. However, these AGV solutions mostly work on pre-defined delivery tasks, which cannot adapt to dynamic manufacturing processes. Thus, the integration of the AGV systems with, potentially dynamically changing, the manufacturing process is of great significance.

2.2.4 The integration of the AGV system on the shop floor

Considering the dynamics and flexibility of the shop floor, the integration of AGVs with the manufacturing process is necessary. Recently, the customisation of AGV has been considered by many manufacturers. For example, with the cooperation of Mechatronic Production Systems Ltd [55] with Omron AIV [56] and the MiRGo [57] solutions, they built customised top modules for AGV to interact with work stations in different applications by attaching the different modules to a standard AGV platform. However, these solutions only considered mechanical interaction without data communication. To achieve the communication between the top modules and the AGV platform, Sell et al. [58] proposed an architecture to support the interconnection of control signals between AGV and top modules. The hierarchical structure consisted of two-level of software and three-level of hardware. Controller Area Network (CAN) and Universal Datagram Protocol (UDP) were adapted for inter-module communication.

Furthermore, the communication structure for connecting AGV systems with IT and OT should also be considered. It can provide the shop floor data and work stations status and help the AGV system to optimise its performance, such as task allocation, mission schedule and AGV maintenance planning. The integration of flexible shop floor logistics and manufacturing processes provides the potential to build a highly automated and dynamically configurable shop floor. In the traditional manufacturing system, the ISA-95 standard [59] has been implemented to aid the development of interfaces between manufacturing systems. This standard describes a functional hierarchy-based automation system architecture including the field level, control level, supervisory level, planning level, and enterprise level. This architecture is shown in Figure 2.4.

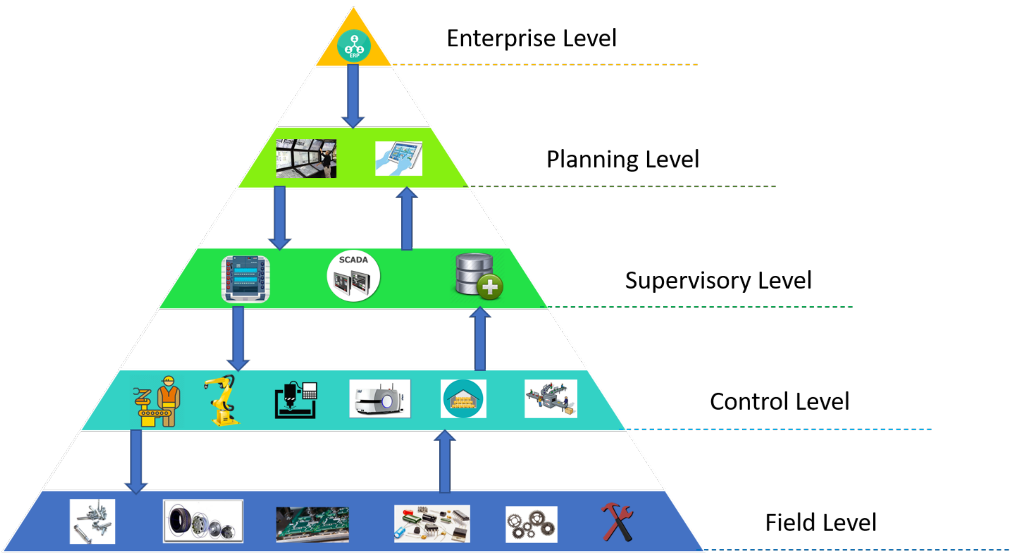

Figure 2.4: The traditional automation systems architecture

However, under this traditional ISA-95 automation systems architecture, insufficient data transaction has impacted the real-time interaction ability between the physical and virtual worlds. That is because the shop floor operation data has to be collected and analysed at the lower level and then transmitted to a higher level for process planning and managing. It is necessary to migrate the traditional hierarchy-based architecture to a more interoperable and flexible architecture, thereby enabling end-to-end interaction from physical devices to MES, Enterprise Resource Planning (ERP), process optimisation and AGV fleet management across different levels. Thus, a CPPS-based automation system (shown in Figure 2.5) is introduced to build an interconnection platform, which supports the data exchange both within the same level of automation systems and between different levels of control and planning.

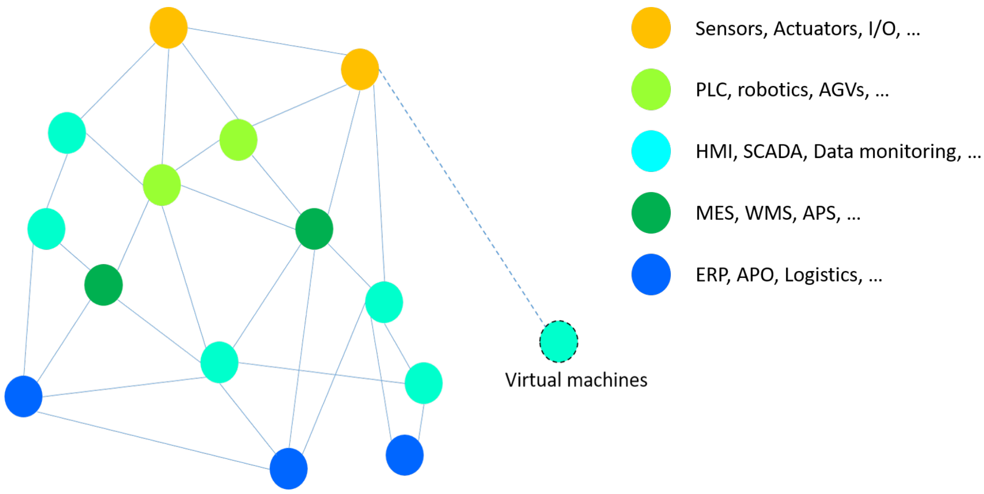

Figure 2.5: CPPS-based automation system [60]

In the context of Industry 4.0, lots of institutional researchers and industrial consortia have recommended this CPPS-based automation system for building an interoperable, modular and flexible smart factory. For example, a generic architecture was proposed by Trunzer et al. [61]. This architecture provides a flexible middleware that can be widely implemented into different use cases for the interconnectivity in distributed systems. Similarly, Saqlain et al. [62] presented an Internet of Things (IoT)-based Industrial Data Management System (IDMS) framework to manage the manufacturing processes data from several devices using state-of-the-art protocols. Also, these data can be analysed to improve productivity and prognosis in manufacturing production. In recent years, a Smart Information Platform and Ecosystem (SIMPLE) framework has been demonstrated by Harrison et al. [63] to envision the connection in the smart factory to connect the digital twins and physical assets throughout the whole lifecycle. These generic communication platforms provide consistent and tight integrations in the manufacturing process for data transaction and control system management.

Therefore, in this research work, a generic communication platform is developed and implemented to enable the data transaction and interaction among sensors, AGVs, fleet managers, MES/ ERP and the shop floor work stations beyond the limitation of hierarchical layers.

To build up the CPPS-based automation system and enable the integration of AGVs-based shop floor logistics with a real-time manufacturing process monitoring capability, the Asset Administration Shell (ASS) concept [64], which is defined by RAMI 4.0 (shown in Figure 2.6), has been studied to deploy into the shop floor environment. It acts as a digital representation of a wide range of shop floor information from various assets, including the AGVs, work stations, IT systems, and plants, thereby enabling vertical and horizontal integration in shop floors.

One of the implementations of ASS metamodel is based on the OPC Unified Architecture (OPC UA), an industry communication protocol [65]. The OPC UA protocol can support the Service-Oriented Architecture (SOA) communication and combine semantic information for physical assets [66]. The contributions of OPC UA are mainly related to the Information, Communication and Integration layers in RAMI 4.0 [67].

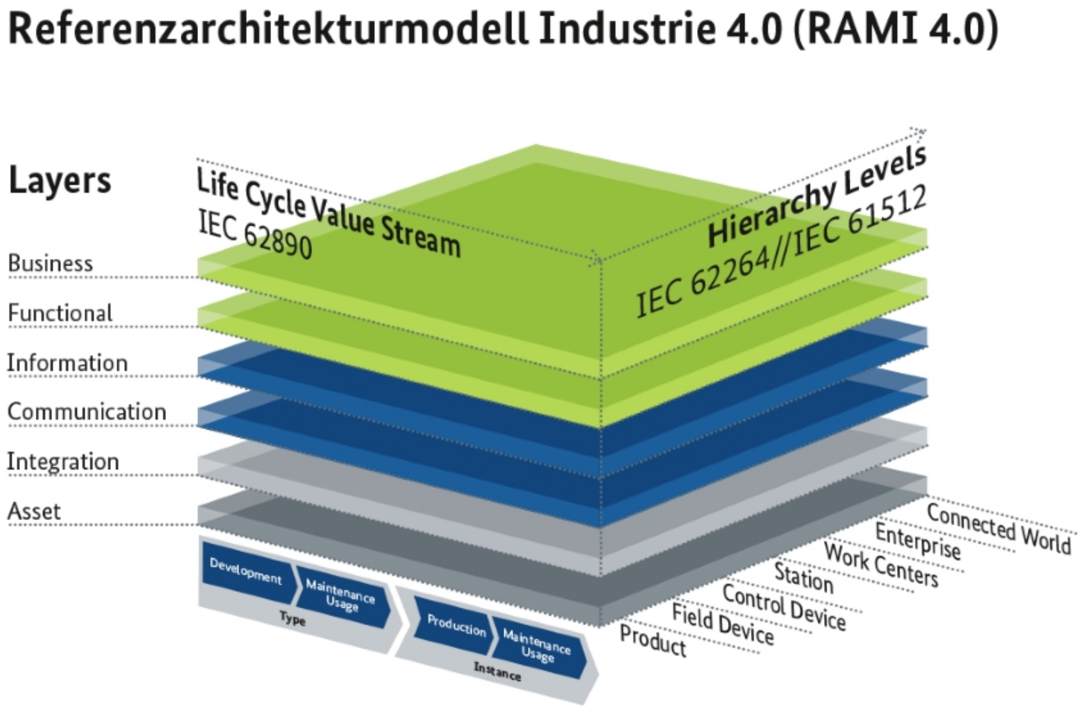

Figure 2.6: RAMI 4.0 [68]

Based on the architecture of RAMI 4.0, the manufacturing industries have an opportunity to merge the advanced technologies in aspects of IT, OT, digitalisation, automation systems and digital twin to build a smart factory. For example, Stefan-Helmut Leitner and Wolfgang Mahnke [69], from ABB Corporate, described OPC UA as a significant factor to integrate Industry 4.0-based technologies into the shop floor, because OPC UA simplifies the complex communication between the IT/ OT software, including MES, ERP and Human-Machine Interface (HMI)s. OPC UA has been implemented into various scenarios to build a smart factory. To be specific, W. Dai et al. [70] built an industrial CPS model by combining OPC UA and IEC 61499. OPC UA works as a general data transaction platform to support the communication between different automation equipment and systems. Also, Schleipen et al. [67] applied OPC UA in different scenarios, which were designed by using different approaches, architectures and software. OPC

UA was used in these cases for defect tracking, data monitoring, information gathering and cross-platform communication. These cases have shown the OPC UA's flexibility, transparency and adaptability.

Moreover, because of its distinguishing characterises, including reliability, scalability, security, cross-platform ability and historical data accessibility, OPC UA is considered to be used in automation systems for data monitoring and control. Seilonen et al. [71] designed an aggregation server using OPC UA protocol and evaluated the sever in an Flexible Manufacturing System (FMS) and mobile work station environments. Similarly, an OPC UA and RAMI 4.0 based control interface was developed by Melo et al. [72]. The OPC UA interface combines RFID identification, functional programming, data communication and real-time process monitoring through a standard and interoperable method. Thus, it has shown the potential generic capability to share information between different communication protocols. In this thesis, the OPC UA server is used to create a generic data sharing framework that supports the integration of AGVs-based shop floor logistics with manufacturing systems. In summary, significant research has been focused on developing a communication platform to support the integration of AGVs with the work stations and the IT/ OT infrastructures on the shop floor.

In order to create interoperability, virtualisation, and real-time capability in shop floor logistics, not only the communication platform needs to be considered. Other key aspects include implementing IIoT into the factory, developing intelligent manufacturing, and integrating manufacturing systems [73].

1) Implementation of IIoT

The IIoT supports data sharing between the automated stations throughout the manufacturing process. It provides a capability for system management to achieve real-time production information analysis and dynamic decision-making on the shop floor. It also supports the development of DT on the shop floor. The DT technology can be used to pre-plan the shop floor layout and optimise the shop floor logistics and manufacturing process by simulating and analysing the physical shop floor scenario. For instance, Park et al. [74] applied an IIoT-based application into a micro smart factory to build a digital twin of the factory model for operation monitoring, product tracking and decision-making. Similarly, Kuts [75] utilised IIoT middleware to support the implementation of a digital twin of an industrial robot. The middleware supports a dual-way synchronisation of the digital and physical worlds.

2) Development of intelligent manufacturing

Intelligent manufacturing merges the advanced technologies of IIoT, AGVs, intelligent robots and HMIs to produce products in a customised way [76]. In intelligent manufacturing, the shop floor logistics can perceive uncertain information and cooperate with different work stations [77]. On the other hand, they can work independently and autonomously to handle production processes and interruptions.

3) Integration of manufacturing systems

The system integration in a shop floor includes horizontal integration, vertical integration, and end-to-end integration. Horizontal integration of the shop floor means integrating the devices, software, and services at the same level. Vertical integration is a network-based manufacturing organisation from sensors/ actuators to production planning and automation systems through the different hierarchical levels [78]. End-to-end integration describes the connection throughout the product lifecycle, from raw materials to products [79], which can enable products to be processed with high quality and high efficiency [14]. AGV can play an important role in connecting the work stations and operators by transporting the materials or products at the field level. For integrating the AGVs into the shop floor, the vertical integration provides a potential opportunity for AGV systems to access the manufacturing data and perform a better delivery service.

2.2.5 The limitation of current AGV system integration on the shop floor

Currently, there are some limitations in deploying AGVs on a shop floor to cooperate with manufacturing processes, as identified below :

1. To implement the AGV in shop floor logistics, extra IT infrastructure needs to be developed based on the relevant equipment vendors' connectivities and the protocols of the industrial devices, such as Programmable Logic Controller (PLC)s, robotics, and sensors. AGV systems cannot easily be integrated with the existing manufacturing systems. Thus, the current AGV systems are typically not capable of easy implementation and plug-and-play.

2. Current AGV systems have limited data access from the manufacturing systems and thus have difficulty analysing and responding to the real-time manufacturing processes. This reduces the flexibility of shop floor logistics, and even interrupts manufacturing processes if AGVs cannot be appropriately scheduled.

2.3 The state-of-the-arts of AGV scheduling

2.3.1 The fleet manager – the software to manage a fleet of AGVs

With the increasing need for flexibility and agility in manufacturing systems, the AGV, as an intelligent mobile robot, is widely deployed in the warehouse and shopfloor logistics for delivery, handling, and distributing products. On the industrial shop floor, managing and organising multiple AGVs working cooperatively and safely is essential. Thus, a software application called fleet manager is developed to control and integrate a fleet of AGVs by monitoring the AGVs' position, tasks and battery level in order to achieve collision avoidance, shortest routes, and task allocation. For example, the MiR Fleet [80] was developed by Mobile Industrial Robots. It provides a centralised control system to manage up to 100 AGVs. This fleet manager can prioritise and choose the most suitable AGV based on the programmed missions by considering availability and location. Moreover, many manufacturers have developed different fleet managers to enhance their multi-AGV capabilities of collaborative, flexible, and interactable operation in various transport scenarios. The comparison of the fleet manager capabilities from different AGV manufacturers is provided in Table 2.2

Table 2.2: The comparison of different Fleet Managers capabilities

Vendor	Fleet allocation	Routes optimisation	Task scheduling	Data analysis	Task prediction	Connectivity	Review
Casun [81]	✓	✓	×	×	×	Low	Limited connectivity with automation systems.
MAXAGV [82]	✓	✓	×	×	×	Moderate	Limited function to analyse the utilisation of AGVs.
AutoGuide [83]	✓	✓	×	×	×	Moderate	Limited capability to schedule the tasks.
BA System [84]	✓	✓	✓	✓	×	High	Limited ability to analyse executed tasks.
DS Automotion [85]	✓	✓	✓	✓	×	Moderate	Able to analyse the order information, but cannot predict the task progress.
JBT [86]	✓	✓	×	✓	×	Low	Limited integration ability with automation systems.
MiR [87]	✓	✓	×	×	×	Moderate	Only support the REST API-based communication.
OTTO [88]	✓	✓	✓	✓	×	High	Flexible APIs and facility integration.
Seegrid [89]	✓	✓	×	×	×	Moderate	Poor compatibility with different operating systems.
Omron [90]	✓	✓	✓	✓	×	Moderate	Only offer manually prioritise the tasks

In conclusion, these fleet managers still have limitations considering the integration, scheduling, and optimisation of the multiple AGVs with the manufacturing process in the shop floor environment. The drawbacks of the current fleet managers are summarised in Table 2.3:

Table 2.3: The limitations of the current fleet managers

Current Functionalities	Limitations
• Manage the AGVs: control the AGV for confliction-free travel. • Allocate the AGVs: select the most suitable AGV to move from point to point by considering the shortest time or travel distance. • Plan the routes: plan the best route for multiple AGVs simultaneously. • Auto Charging: manage the AGV charging by considering the pre-defined charging threshold. • Visualisation: show the real-time positions of multiple AGVs, and the status of tasks. • User Interface: allow the operator to control the movement/tasks of AGVs manually. • Interoperability: enable the development and connectivity of other APIs.	• Unable to optimise the number of AGVs by considering the utilisation of AGV on the shop floor. • Only able to schedule tasks based on the first-in-first-out method without prioritising tasks. • Unable to optimise the charging plans of AGVs to maximise the lifetime of battery and utilisation of AGVs. • Lack of ability to analyse the travel time of AGVs. • Lack of a generic gateway to integrate AGVs with the manufacturing system.

2.3.2 The AGV scheduling approach in literature

In order to reduce the limitations of the fleet manager and organise multiple AGVs working efficiently on the shop floor, the scheduling and path planning for fleets of AGVs have been widely researched in the last decade [91–104].

In the early stage, the AGV scheduling algorithms focused on managing multiple AGVs running in the same environment without conflicts by task assignment and path planning [105]. For example, Gaskins and Tanchoco [106] developed a zero-integer based mathematical model for optimising the AGV travelling path. In 1988, Daniels [107] combined branch-and-bound theory and shortest path algorithm to design conflict-free routes for complex AGV systems. Similarly, Qiu and Hsu [108, 109] presented a bi-directional route algorithm for AGV to complete pick up and drop off jobs without conflicts.

Recently, enabled by improvements in digital technology, computer power, microchip techniques, and software, scheduling and routing approaches now aim to achieve the integration and cooperation of AGV systems with manufacturing processes by minimising machine waiting time, minimising energy consumption, optimising the manufacturing processes and improving JIT performance.

In recent literature, the approaches to the scheduling of multiple AGVs with manufacturing systems can be classified into three types: offline scheduling, online scheduling and simulation-based scheduling. The offline scheduling methods are developed for scheduling AGVs tasks based on the pre-defined manufacturing process. Machine and AGV parameters are assumed to depend on historical data or manufacturer manuals. The offline scheduling approach is usually implemented in the manufacturing system before it starts. The online scheduling method can schedule the AGVs during the manufacturing execution and aims to provide dynamic scheduling strategies as production status changes based on the abstracted mathematics models. As part of online scheduling, the simulation-based scheduling approach has gained the attention of researchers [110]. It converts the complex manufacturing scenario into a digital simulation model to analyse and predict the production status, searching optimal or near-optimal scheduling decisions.

2.3.3 Offline scheduling approaches

The offline scheduling methods can be divided into three main parts: 1) mixed-integer non-linear programming; 2) divide-and-conquer algorithm; 3) heuristics search algorithm. In the early stage, the scheduling algorithms were mainly applied to design conflict-free routes for AGVs working in the manufacturing system. For example, Oboth et al. [111] presented a route-generation approach providing conflict-free routes for AGVs with different speeds. Also, the task assignment and idle AGVs positioning are considered in this approach. Demesure et al. [112] proposed a navigation method for AGVs travelling in an FMS. This approach combined the priority negotiation and the motion planner to minimise the operation completion time by supporting optimal resources and conflict-free routes for AGVs. They [113] further designed decentralised motion planning and scheduling method for AGVs. In this method, the presumed trajectories were firstly planned by a central system to avoid collisions. Secondly, an overall conflict-free trajectory was generated by combining the neighbours' presumed plan and priority policies. Similarly, Fontes et al. [114] used a new mixed-integer linear programming model to address scheduling problems of machines and AGVs in an FMS. This model made interconnected decisions for work stations and AGVs. It was constrained by the completion time of operation and delivery tasks. Fazlollahtabar [115] presented a minimum-cost network flow (MCF) method to assign multi-AGV for inter-assembly lines and intra-assembly lines. This method reduced the task completion time and organised resource assignments for every short-term window. However, for large-scale shop floor environments, the model becomes more complex, and these algorithms are difficult to schedule the AGV systems.

With the increasing number of AGVs in the manufacturing environment, the decomposition-based approach was applied to separate the problems into smaller parts. For instance, Correa [116] proposed a divide-and-conquer method to address multiple AGVs dispatching and conflict-free routing problems. In this method, the scheduling problem was converted into conflict-free routing based sub-problems and was solved by mixed-integer programming. In similar, Nishi et al. [117] presented a mixed-integer model-based decomposition approach to address the task scheduling and routing problems separately. Moreover, Gelareh et al. [118] exploited a Lagrangian relaxation based decomposition algorithm to minimise the overall production operation time by scheduling AGV delivery. However, this approach would rapidly lead to increased computing time when faced with a complex manufacturing system, for example, a manufacturing process consists of multi-stage work stations and a large number of AGVs.

Furthermore, heuristic and AI-based search algorithms are widely used to schedule AGVs in manufacturing systems for finding a near-optimal solution in an acceptable timeframe. For instance, Dang and Nielsen [119] proposed a heuristic-based Genetic Algorithm (GA) to schedule the machines and AGVs in an FMS simultaneously. Meherabian et al. [120] combined the NSGA-II and multi-objective particle swarm optimisation (MOPSO) to solve a two-objective problem: 1) Total jobs processing time; and 2) FMS delay time. Also, the environment parameters were considered as fuzzy in a mix-integer programming model for imitating the physical shop floor environment. Similarly, Mousavi et al. [121] developed a hybrid genetic algorithm and particle swarm optimisation (GA-PSO) integrated mathematics model to schedule AGV tasks for optimising: 1) Total jobs operation time; 2) Number of AGV. To provide better performance for the same objectives, they [122] proposed a fuzzy hybrid GA-PSO algorithm later. Liu et al. [123] integrated a multi-adaptive genetic algorithm (MAGA) with two adaptive genetic algorithms (AGA) based on a multi-objective mathematical model. This model considered AGV charging time and variable speed for scheduling multiple AGVs to optimise the makespan, the number of AGVs and energy consumption. However, the offline scheduling methods only can handle deterministic systems. For the dynamic manufacturing environment, they, therefore, exhibit limitations in the scheduling of AGV tasks.

2.3.4 Online scheduling approaches

Online scheduling methods provide dynamic schedules for the AGVs and FMS during the manufacturing processing to match demands from the customer or address the production interruptions in time. In general, online schedulings are time-constraint approaches which mean schedule solutions should be generated in a limited time. For example, Jin et al. [124] designed a GA based method to solve dynamic multi-AGV scheduling problems for a container terminal. It aimed to minimise the materials handling time of AGVs. Weynes et al. [125] proposed a DynCNET protocol which was extended from standard contract net (CNET) [126] to dynamically manage tasks for AGV transportation. The DynCNET enables AGVs to switch assignments dynamically.

More recently, Wang et al. [127] developed an architecture based on a multi-agent system (MAS) for real-time scheduling manufacturing in an IoT-enabled FMS. In this architecture, the Bargaining-Game-based module was designed to optimise multiple objectives, including production completion time, machine workload, and energy consumptions. However, the Bargaining-Game-based method focuses on the short-term interest, which means the results may be limited to a local optimal solution.

Zhao et al. [128] established a multi-AGV-based unmanned smart factory. AGVs were guided by magnetic tape, and the movement deviations of each AGV from this tape were monitored and analysed based on the data from magnetic sensors, which helps the AGVs drive over the tape steadily. Also, in this system, the A* algorithm was developed to handle conflict-free routing and scheduling problems. Xue et al. [129] presented a reinforcement learning algorithm to minimise the production makespan and jobs delay by scheduling multiple AGVs tasks on the shop floor. The AGV system obtains and shares the information with machines, so the control system can make the dynamic decision by understanding the shop floor production status. However, this method requires a huge amount of training time for its optimisation module, especially in a large-scale shop floor environment. It also cannot reschedule multiple AGV in a short time when facing unexpected events.

2.3.5 Simulation-based scheduling approaches

With increasing numbers of automated stations, on larger-scale shop floors, the simulation-based scheduling method has become a more effective approach, especially for complex and dynamic shop floor logistics. Thus, Lin et al. [130] proposed an L-GAOCBA algorithm to schedule AGVs and machines on the shop floor using a Siemens Plant Simulation-based simulation model. This model is used to evaluate the system performance under uncertain factors. Similarly, Viharos et al. [131] developed a simulation model by using a Siemens Plant Simulation to schedule the AGVs and assembly operations for minimising the manufacturing time. However, there is little literature focused on the simulation-based scheduling methodology.

2.3.6 The classification of AGV scheduling algorithms

Table 2.4: The literature of offline scheduling approaches

Authors/Ref	Scheduling Algorithm	Objective	Mathematical model
Oboth et al. [111]	Conflict-free routing approach	Provide conflict-free routes for multi-AGVs	Bi-directional single-lane network model
Correa et al. [116]	Divide and conquer based hybrid approach	Solve dispatching and conflict-free routing problems for AGVs	Mix-integer programming model
Demesure et al. [112]	Combination of a scheduling motion planner and a priority-based negotiation method	Minimise the tasks processing time	Mobile agent model
Demesure [113]	Decentralized motion planning and scheduling approach	Provide overall conflict-free routes multiple AGVs	Mathematical model
Fontes and Homayouni [114]	A novel MILP model	inter-connected decisions to minimise machines operation time and AGVs delivery time	MILP model
Fazlollahtabar [115]	a minimum-cost network flow (MCF) method	optimised the task completion time and resource assignment.	MCF model

Continued on next page

Table 2.4 – *Continued from previous page*

Authors/Ref	Scheduling Algorithm	Objective	Mathematical model
Rashidi et al. [132]	Extension of standard Network Simple Algorithm (NSA+)	Scheduling AGVs in container terminals to meet delivery requirements.	MCF model
Nishi et al. [117]	A two-level decomposition algorithm	Minimise the total weighted tardiness of tasks	Mix-integer programming model
Gelareh et al. [118]	Lagrangian relaxation based decomposition method	Minimise the operations makespan	Mix-integer programming model
Fazlollahtabar et al. [133]	A heuristic search algorithm based mathematical programming	Minimise the penalty of tardiness in a conflict-free and JIT production	heuristic search
Udhayakumar et al. [134]	GA and Ant Colony Optimization (ACO) algorithm	Multi-objective for AGVs: 1) workload balance, 2) travelling time minimise based on utilization maximum condition	heuristic search
Yang et al. [135]	Congestion Prevention Rule-based Bi-level GA	Minimise makespan	heuristic search

Continued on next page

Table 2.4 – *Continued from previous page*

Authors/Ref	Scheduling Algorithm	Objective	Mathematical model
Huang et al. [136]	Hybrid strategies of K1 and OA	Minimise the product waiting time	Mix-integer programming model
Sankar et al. [137]	ParallelGA based multi-objective evolutionary algorithm (MOEA)	Multiple objectives were considered:1) maximise the utilisation; 2) minimise the jobs tardiness	heuristic search
Dang and Nielsen [119]	Heuristic-based genetic algorithm	Minimise the time cost of production tasks	heuristic search
Nageswararao et al. [138]	Binary particle swarm Vehicle Heuristic Algorithm	Minimise the mean tardiness	heuristic search
Baruwa and Piera [12]	A timed coloured Petri net (TCPN) method	Minimise the makespan and job processing time	Heuristic search
Sanches et al. [139]	Adaptive genetic algorithm	Minimise the makespan with a short computing time	Heuristic search
Nouri Driss [140]	Hybrid heuristic algorithm	Minimise the makespan	Heuristic search

Continued on next page

Table 2.4 – *Continued from previous page*

Authors/Ref	Scheduling Algorithm	Objective	Mathematical model
He et al. [141]	State-dependent part input sequencing approach, shortest remaining processing (SRPT), and SPT dispatching rules.	Schedule the parts input sequences, AGVs and machines	Simulation Model
Tavakoli et al. [142]	NSGA-II	Minimise the production processing time and AGVs' idle time.	Heuristic search
Mehrabian et al. [120]	NSGAII and MOPSO	Solve Two-objective problems: 1) Minimise total jobs processing time; 2) Minimise delay time.	Heuristic search
Mousavi et al. [121]	Hybrid GA-PSO algorithm	Solve a two-objective problem: 1) Minimise makespan; 2) minimise AGV number.	Heuristic search
Mousavi et al. [122]	Fuzzy hybrid GA-PSO algorithm	Solve a two-objective problem: 1) Minimise makespan; 2) minimise AGV number.	Heuristic search
Zhong et al. [143]	Hybrid GA-PSO algorithm	Minimise the AGV delay time during delivery.	Heuristic search

Continued on next page

Table 2.4 – *Continued from previous page*

Authors/Ref	Scheduling Algorithm	Objective	Mathematical model
Liu et al. [123]	Integration of a MAGA and two AGA	Multiple objectives: 1) makespan; 2) AGVs number; 3) AGV electricity consumption.	Heuristic search
Wang et al. [144]	Bilevel heuristic algorithm	Minimise AGVs energy consumption	Heuristic search
Li et al. [145]	An improved harmony search algorithm	Minimise 1) the standard deviation of machine waiting time. 2) AGV travel distance	Heuristic search
Zou et al. [146]	Nearest neighbour-based heuristic search and discrete artificial bee colony algorithm	Minimise: 1) AGV travel distance; 2) standard deviation of machine waiting time	Heuristic search
Zhang and Li [147]	Improved particle swarm optimisation algorithm	Minimise jobs makespan	Heuristic search
Chawla et al. [148]	Modified Memetic Particle Swarm Optimisation (MMPSO) Algorithm	Minimise the AGV running and waiting time	Heuristic search
Chen et al. [149]	GA	Minimise production processing time	Heuristic search

Continued on next page

Table 2.4 – *Continued from previous page*

Authors/Ref	Scheduling Algorithm	Objective	Mathematical model
Riazi et al. [150]	Heuristic-based Benders decomposition	Solve AGVs conflict-free routing and scheduling problem	Heuristic search

Table 2.5: The literature of online scheduling approaches

Authors/Ref	Scheduling Algorithm	Objective	Mathematical model
Weyns et al. [125]	DynCNET Protocol	Manage dynamical assignments for AGVs	A novel framework
Wang et al. [127]	Bargaining-game-based scheduling approach	Multiple objectives: 1) makespan; 2) machine workload; 3) energy consumption.	A novel framework
Zhang et al. [21]	CPS-based smart control model	Minimise the processing time and maximise the AGVs delivery network utilisation	A novel framework
Xue et al. [129]	Reinforcement learning approach	Minimise the job delay and makespan	AI
Zhao et al. [128]	A* algorithm	Solve the scheduling and conflict-free routing problem	AI
Sahin et al. [151]	A Prometheus^{TM} based model	Scheduling machines and AGVs simultaneously	A novel framework
Xu et al. [152]	Double-level hybrid genetic algorithm and ant colony optimisation (DLH-GA-ACO)	Minimise the production time and minimise AGV numbers	AI
Erol et al. [153]	A negotiation mechanism-based multi-agent scheduling method	Scheduling machines and AGVs simultaneously	A novel multi-agent system (MAS) based model

Continued on next page

Table 2.5 – *Continued from previous page*

Authors/Ref	Scheduling Algorithm	Objective	Mathematical model
Chawla et al. [154]	Grey wolf optimisation (GWO) algorithm	Multiple Objectives: 1) AGV workload; 2) AGV travel time	Heuristic search
Zheng et al. [155]	Hormone regulation -based approach (HRA)	Optimise makespan	Hormone-regulation mechanism
Liu et al. [156]	Combination of a unidirectional graph and A* algorithm	Planning conflict-free path for multi-AGVs.	Topological map model
Huang et al. [157]	Combination of admissible and non-admissible heuristic functions based A* scheduling method	Minimise the makespan and maximise machines utilisation	heuristic search

Table 2.6: The literature of simulation-based scheduling approaches

Authors/Ref	Scheduling Algorithm	Objective	Mathematical model
Viharos et al. [131]	Heuristic algorithm	Minimise the total manufacturing time	Simulation model
Fu et al. [158]	Combination of DES, fractional factorial design (FFD), and response surface methodology	Determine the AGVs requirement in the manufacturing system.	Simulation model
Chan et al. [159]	Fuzzy-logic approach	Multiple objectives: 1) machines blocks; 2) processing time; 3) processing steps; 4) average flowtime.	A novel framework
Lin et al. [130]	GA-based local search with considering optimal computing budget allocation (L-GAOCBA)	Scheduling machine and AGV simultaneously with a low computation cost.	Simulation model

Table 2.7: The advantages and disadvantages of three types of scheduling methods

Types	Advantages	Disadvantages
Offline scheduling	• Able to solve complex scheduling problems; • Low CPU computing cost;	• Inflexibility; • Only can handle deterministic system; • Need overall system information; • Limited by executing time; • Unable to handle the unexpected behaviours;
Online scheduling	• Enable to solve the dynamic problem; • Enable to generate solution at the running time;	• Require powerful CPU/GPU performance; • Limited to complex scenarios; • Need a large amount of training data; • Reduce the response time;
Simulation-based scheduling	• Enable to simulate the real factory scenarios; • Enable to re-schedule AGVs during running time; • High accuracy of scheduling model;	• Need pre-design the system modelling; • Require the powerful CPU/GPU performance

In the literature on the scheduling algorithm for AGVs, the authors mostly focus on an optimisation solution which is able to improve the performance of AGV deliveries, for example, reduce the travel distance, minimise the AGV running time, and minimise the job makespan. However, they do not prove this optimised result is a global optimisation result and there is no clear evidence to show that by using their optimisation methods, a global optimisation solution can be found within an acceptable time. Thus, it may confuse the readers when they try to use these methods to search for a global optimisation solution.

Specially, In the literature on offline scheduling, the algorithm is easy to execute and is able to handle multiple AGV situations, but due to its offline limitation, it cannot easily handle different scenarios when the shop floor system changes. In the literature on online scheduling, most algorithms are designed for low throughput and small systems with a few stations. And it could cause more computing power when applied to a large-scale system. Furthermore, the authors mostly use a simulation or lab-based scenario to evaluate their algorithms, thus it is difficult to guarantee these algorithms perform similarly in the real industrial shop floor environment.

2.3.7 The challenges of current AGV scheduling approaches

As discussed in previous sections, a large number of research efforts have focused on scheduling multiple AGVs-based shop floor logistics. Nevertheless, there is limited literature on scheduling the AGVs tasks by considering the real-time manufacturing processes. Although, many AGV manufacturers developed fleet management software to manage multiple AGVs by monitoring location and status, prioritising delivery tasks, generating conflict-free routes, and providing connectivity with other machines. They still lack the data collection and analysis ability through which the AGV system can execute delivery tasks optimally.

In general, the offline-based scheduling algorithm was primarily investigated for handling static shop-floor environments. The offline scheduling method optimises AGV performance by preplanning logistics schedules. However, the offline method cannot handle unexpected situations, including machine breakdowns, human interruption, and new order demands.

More recently, with the increased flexibility of shop floor logistics and needs of customised products, the heuristic-search and AI algorithm-based online scheduling approach has been considered for dynamic scheduling of AGVs taking into account data collected from the shop floor in real-time. The online solution can optimise the production schedule during manufacturing execution and takes into account unpredicted events. It can continuously update the optimal solutions as the operational information becomes available in the real-time process, which is very supportive for scheduling the dynamic AGV delivery tasks. However, the mathematics model-based on-line method increases the CPU loads by converting the operation information into the analytics model during running time, also it has limitations in reproducing the real-time manufacturing processes.

Because DT technology can be used to duplicate the physical assets to provide an optimal scheduling solution by modelling, simulating and analysing real-time manufacturing processes as part of the online scheduling solution, the DT-based scheduling method has been considered recently to schedule AGVs in shop floor logistics.

2.4 DT-based shop floor logistics optimisation

2.4.1 The introduction of DT technology

With the increasing complexity of shop floor manufacturing processes, the DT-based approach has been applied recently for analysing the shop floor manufacturing systems. The Digital Twin, a conceptual model for Product Lifecycle Management (PLM), was first presented by Michael Grieves in 2002 [160]. Digital twin refers to a high-fidelity virtual representation of the physical objects built by a computer system through physical machine data and models to simulate the machines real-time operations [161]. The digital twin connects with the physical automation system using sensors, IoTs and real-time communication technology [162] to access and monitor live operation data. Thus, in the virtual world, the complex production process can be simulated, analysed, and predicted to optimise the product lifecycle and manufacturing process in a closed-loop [163]. In Figure 2.7, the digital twin is demonstrated as a virtual duplication of a physical shop floor.

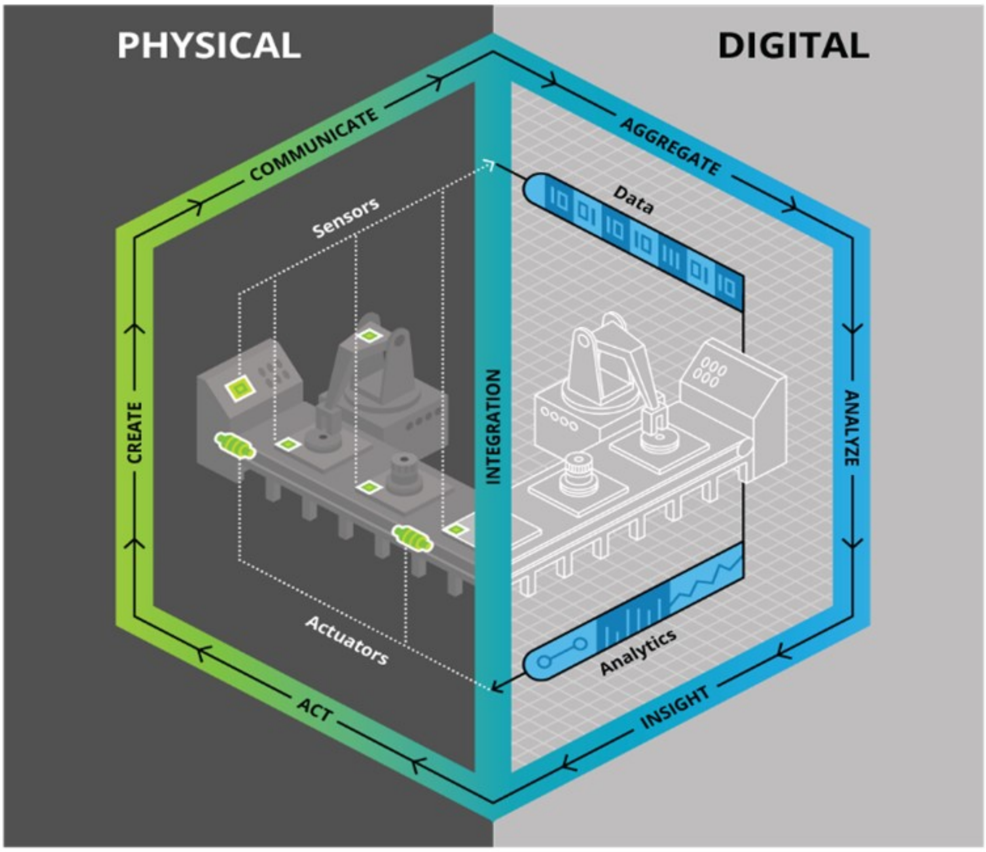

Figure 2.7: The example of the digital twin in smart factory[164]

As shown in Figure 2.7, the actuators and smart sensors were implemented into the production line to monitor real-time machine information. The data is passed to virtual models for duplicating the physical shop floor to combine the physical manufacturing system thoroughly with its digital world.

The DT technology can help the manufacturing systems to optimise the production process by supporting real-time monitoring, operation analysis, and production performance predictions. Thus, it has gained a wide range of attention in the manufacturing sector [163]. As shown in Figure 2.8, the DT converges the physical and virtual shop floor, including machine status, robotics behaviour, and logistics, to collect comprehensive information and provide optimised decisions for the shop floor service systems.

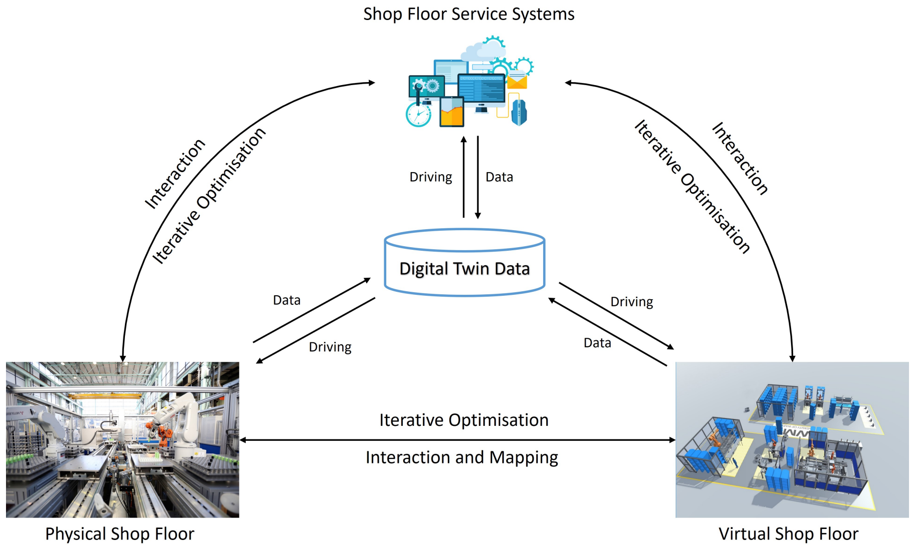

Figure 2.8: The Digital Twin models in shop floor

2.4.2 DT technology in scheduling AGVs

Digital Twin technology has been researched for shop floor logistics in the last few years [165] because of its benefit of investment cost reduction by using simulation to analyse the shop floor logistics and predict potential bottlenecks. Bottani et al. [166] exploited a cheap and suitable solution to optimise the AGV schedules by creating a DES-based DT. The DES model simulates various scenarios of the AGV systems, including the validation of suitable scheduling solutions and evaluation of the AGVs decision-making ability. However, due to the proposed communication method (i.e., Machine-to-machine (M2M) logic) constraints in the real shop floor, the author only focused on the DT model of AGVs without considering the overall manufacturing process.

Similarly, Gyulai et al. [167] proposed a DES-based shop floor logistics system to analyse AGV capacity planning and dispatch policy, which was successfully deployed on the shop floor with the AGVs working as expected. Although the model and the simulation of the manufacturing system cost more time to design and develop relative to alternative approaches, it could assess alternative candidate solutions without disturbing the physical manufacturing system.

Yan et al. [168] designed a DT model of the shop floor logistics to validate their proposed optimisation algorithm. The improved GA method was proposed to solve the Flexible Job Shop Problem (FJSP) on the shop floor and the DT model was built to verify the proposed algorithm. However, the DT technology was only used to verify the optimisation results, without the interaction with the actual shop floor logistics.

DT framework [169] supports the bidirectional communications between the physical and digital systems. A DT model of shop floor systems shows: 1) the capability of offering a significant user-friendly environment to analyse manufacturing systems; 2) the importance of bidirectional communication during manufacturing process monitoring; 3) the capability of enabling the operator to monitor and manage the manufacturing system remotely.

Although the literature concludes the importance of DT on the shop floor for the manufacturing process and logistics scheduling, most of the existing works are focused on the framework, concept, and solution validation. They are seldom applied the DT within the shop floor manufacturing process and AGV-based logistics in practice. Actions need to be considered towards implementing an enhanced DT for AGV-based logistics and the manufacturing process in order to carry out real-time optimisation of overall shop floor logistics [165]:

1) IIoT and data hub-based data-sharing platforms on the shop floor need to be implemented to enable real-time physical process monitoring and data gathering abilities for AGV systems throughout the whole manufacturing process.

2) A data analytics module needs to be developed for analysing machine actions and duplicating the physical behaviours for the DT model.

3) An optimisation and smart decision-making module needs to be developed and integrated with the DT model to provide an optimal solution for the shop floor logistics and manufacturing processes.

4) The engineering tool for creating a DT model of a physical shop floor environment needs to be considered, for example, DES, continuous simulation and monte carlo simulation.

2.4.3 DES

Due to the benefits of DES tools, such as the ability of powerful simulation, inexpensive scenario analytics, and dynamic process modelling [170], the DES tools [171] are studied in the manufacturing sector by integrating with physical shop floor data for initial investment analysis, production pre-planning, system performance prediction and AGV-based shop-floor logistics scheduling [172].

DES models can analyse and simulate the physical shop floor manufacturing systems, and can also be combined with a mathematical algorithm to optimise the manufacturing process [173]. The main benefit of DES is the opportunity to carry out the study of factory layout and production throughput analysis without the physical factory implementation [174]. DES can be integrated with physical shop floor systems and applied to a decision support system for improving the manufacturing system KPIs [175].

For example, Khan [176] combined the WITNESS Horizon simulation with the data-driven reengineering framework to analyse and verify the system changes. Later, Khan et al. [177] optimised the production process by integrating the WITNESS Horizon-based DES with Khan-Hassan-Butt (KHB) methodology. DES predicted that there was a 20% increase in the manufacturing process output by using this method. Similarly, Avventuroso et al. [178] proposed a simulation-based approach to design and analyse medical device production using AnyLogic. The decision-making was able to provide better options for this large-scale production set-up and design. Dorota et al. [179] used FlexSim to simulate the sleeves production line and combined it with Value Stream Analysis (VSA) to optimise the process and improve the sleeves production performance. Vysocky [180] designed a simulation model using WITNESS Horizon software to evaluate optimisation solutions in materials flow. Integrating the DES with data analytics allows the manager to evaluate the potential benefits of various solutions by predicting the production performance. Thus, many studies are combining the data analysis with DES [181] to improve the efficiency of the production process and to reduce the costs of physical shop floor implementation [182].

Furthermore, many other DES vendors have been working in the manufacturing sectors for system and process optimisation, for example, Siemens Plant Simulation, Simo and ARENA [183–187]. There is a comparison of DES-based engineering tools shown in Table 2.8

Table 2.8: The comparison list of DES software

Software	Typical application	Connectivity	Optimization	Comments
Arena [188]	Industry applications for analysing the system	OptQuest, VB,	OptQuest	**Pros:** User-friendly. **Cons:** Limited learning resource; Hard to debug the models.
AnyLogic [189]	Multi-method simulation tools	Excel, Access, Database, OptQuest, Java/DLL	OptQuest	**Pros:** Database connectivity. **Cons:** Need programming (JAVA) skills to customise the simulation.
FlexSim [190]	3D modelling-based simulation	Excel, Database, C++ based developing, OLE and ActiveX	OptQuest	**Pros:** AGV simulation capability; 3D model based simulation. **Cons:** Need programming skills to use FlexSim Scripts and process flow logic design.
Simio [191]	Object-oriented modelling	.Net programming, Excel, Access, SQL server	Multi-objective and pattern frontier optimisation	**Pro:** User-friendly. **Cons:** The learning curve is too steep.
Siemens Plant Simulation [192]	PLM planning and visualisation	MATLAB, Excel, Autocad, OPC UA	GA, Neural networks and branch and bound optimisation algorithm	**Pros:** easy to link with real-time data server. **Cons:** The learning curve is too steep.
SimEvents [193]	Optimise the event-driven operations	MATLAB/ SIMULINK	Hybrid system models	**Pro:** Powerful optimisation library and engine; **Cons:** Need programming skills; No 3D modelling capability.
WITNESS Horizon [194]	Production process predication and multi-objective optimisation	Excel, Database, OLE,WITNESS Action Language	experimenter with Hill Climb, Adaptive Thermostatically SA, Six Sigma algorithm	**Pros:** Easy to link with external optimisation engine. **Cons:** Missing AGV features; 3D visualise capability only.

Providing good connectivity with external software, stable APIs, and a user-friendly design environment, the WITNESS Horizon software is a well-established, capable, option to model, simulate and optimise shop-floor scenarios. WITNESS is a discrete event simulation software developed by Lanner Group Ltd. It is widely applied for simulating manufacturing processes to predict system performance and support the factory planning optimisation [195]. It can model the different elements in the aspect of discrete elements (e.g., parts, machines, vehicles, tracks, etc.) and continuous elements (e.g., fluids) and simulate their working status, including busy, processing, idle, blocked, set-up time, and waiting for labour. Moreover, via the "Object Linking and Embedding (OLE) Automation Server", the WITNESS software can be controlled by external optimisation engine by using WITNESS Commands Language (WCL).

2.5 Chapter summary and research gap

In this chapter, the application of AGVs on the shop floor has been reviewed. The literature review shows that the applications of AGV systems have rapidly increased for materials handling. It indicates the importance of the integration of an AGV system into shop floor logistics, production process planning and scheduling, and manufacturing performance optimisation.

Furthermore, the AGV scheduling and planning methodologies have been discussed and compared critically in the context of offline, online and, especially, simulation-based methods. The literature review shows that, in the last decade, many AGV suppliers and manufacturing companies have researched the integration of AGVs with manufacturing systems by developing the fleet manager, scheduling multi-AGV-based shop floor logistics.

Additionally, for improving the KPIs of the manufacturing process, for example, just-in-time, OEE, energy costs, etc., the simulation-based method was considered to provide more accurate optimisation results by simulating the physical shop floor manufacturing systems. In particular, DT technologies enable duplication of a physical system into a virtual world by monitoring and analysing the operational data. Thus, the DT technologies and the DT-based AGV scheduling methods have been reviewed and categorised in Section 2.4.2. As one of the keys enabling technology of the smart factory, the digital twin was widely researched from the perspectives of 3D modelling, product lifecycle management, factory pre-planning, and manufacturing scenario visualisation. Also, DES tools, as a cyber twin to implement the DT systems, are reviewed and compared analytically in Section 2.4.3.

This chapter presents the current approaches for integrating the AGVs with manufacturing systems on the shop floor and has comprehensively compared the scheduling methods for multiple AGVs. Moreover, DT-based AGV scheduling methodologies have been reviewed. This approach will be adopted and implemented in this research for AGV-based shop floor logistics optimisation. The research gaps identified through the literature review provided in this chapter are summarised below:

1) Lack of a generic solution to support the integration of AGVs with the automation systems on the shop floor. There is the need for a generic data-sharing platform that supports the communication between the AGV system with work stations and management systems, such as the PLCs, robots, energy monitors, sensors, shop floor database, MES and ERP. Moreover, this solution should be able to support the operational information communication between the high-level control systems (MES, ERPs, etc.), AGV-based shop floor logistics, and the work stations in real-time.

2) Lack of a methodology to schedule the AGV-based shop floor logistics for raw materials delivery, work-in-process delivery and finished products collection whilst considering the real-time manufacturing process. Currently, the offline AGV scheduling methods are based on the static factory environment, which cannot address unexpected events occurring in the manufacturing process, such as new production orders, machine breakdowns, and product defects. Online AGV scheduling methods mainly use mathematical models to describe the physical shop floor scenario, with limitations inadequately duplicating the real-time manufacturing process.

Thus, the SAMS methodology, which will be described in the next chapter, is conceived by the author to achieve the optimisation and integration of AGV-based shop floor logistics with the manufacturing process.

Chapter 3

Methodology

3.1 Introduction

The literature review has illustrated the challenges of the current shop floor logistics. They lack a generic solution to support the integration of AGVs into the shop floor logistics, and they have limitations in scheduling AGVs based on real-time manufacturing process data.

In this chapter, an innovative methodology, SAMS, is presented to enable seamless integration of the AGVs system with the manufacturing process on the shop floor and carry out real-time scheduling optimisation. The two main innovations and contributions of this methodology are:

1. Integration of AGV-based shop floor logistics with a manufacturing process. SAMS methodology supports integrating multiple AGVs with shop floor manufacturing processes and associated IT systems. The proposed system architecture allows communication among the AGV-based shop floor logistics, automation systems (e.g., robotics, PLCs, autonomous stations, etc.), and upper-level control systems (e.g., Advanced Planning and Scheduling (APS), MES, ERP, etc.) through a central data-sharing platform to increase the flexibility, interoperability and traceability of manufacturing process. Also, it can monitor the AGV status and real-time manufacturing operation information, and supports intelligent decision-making by considering the system KPIs

requirements.

2. DT-based real-time scheduling of AGVs for shop floor logistics. The SAMS methodology optimises and predicts the real-time manufacturing system performance by optimising the fleet size of AGVs, scheduling of AGVs, and the charging plan of AGVs. A DES model is developed and integrated with the proposed data-sharing platform to create a digital twin of the shop floor for real-time manufacturing process monitoring and production performance prediction. Moreover, through this digital twin model, SAMS can detect abnormalities, such the machine breakdown, AGV breakdown, tools defects, and new customer requirements to re-generate scheduling strategies for AGV-based shop floor logistics.

3.2 Methodology overview

There are many requirements of the smart factory, for instance, real-time manufacturing process monitoring, flexible logistics systems, JIT production delivery and intelligent decision-making. To meet these requirements, an innovative SAMS architecture, as shown in Figure 3.1, is proposed to integrate the AGV-based shop floor logistics with manufacturing processes and high-level management systems (e.g., MES, APS, ERP, etc.)

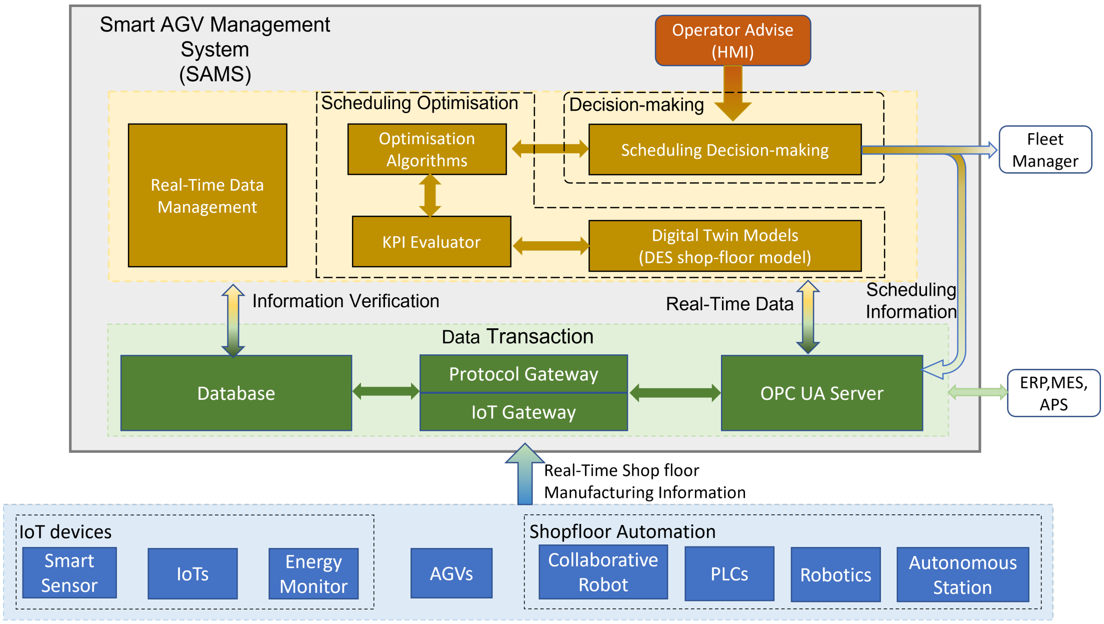

Figure 3.1: Architecture of Smart AGV Management System

SAMS architecture consists of the M2M communication platform which enables the cyber-physical integration, real-time interoperability among the work stations, AGVs, and system controllers, the DT-based optimisation which supports the AGV delivery task scheduling, and the smart decision-making which analyses and selects the optimised AGV schedules for the fleet manager. In this way, the SAMS architecture is able to support shop floor logistics schedules by analysing information of AGVs and automated stations and simulating AGV allocation scenarios.

The main innovation in the SAMS system is the development of the data transaction module and the DT-based scheduling optimisation module. In the data transaction module, the OPC UA Server is built to subscribe to the real-time operation information including the machine working status, takt time, AGV delivery tasks and operation status. Also, the real-time information can be stored in the database through the specific gateway (i.e., Structured Query Language (SQL) gateway, IoT gateway and OPC Data Access (OPC DA) driver) for further data analytics. Through this data transaction module, the AGV system can be integrated into the current manufacturing system with a low engineering cost.

In the scheduling optimisation module, the DT model is created to capture the information from the manufacturing system and duplicate the shop floor scenario for supporting the AGV-based shop floor logistics optimisation. Meanwhile, the DT model is able to feed back the optimised AGV scheduling strategy to the physical shop floor and act on the workflow of shop floor logistics. From this module, a group of AGV scheduling strategies are generated.

In the decision-making module, the operator is able to decide the best suitable scheduling strategy from a list of optimised strategies for the AGV system based on the operator's experience and the manufacturing requirements. Take an example, there are three different scheduling strategies provided by the DT-based optimisation module. Strategy-I aims to minimise the overall energy cost of AGVs; Strategy-II aims to maximise the balance of all AGVs; and Strategy-III aims to maximise the number of shipped products. If the priority objective of the shop floor for the current shift is to maximise the number of shipped products, the operator is able to choose Strategy-III manually through the HMI. Also, if the objective is changed in a different shift, the scheduling strategy can be updated through the decision-making module.

To be clear, in the SAMS, the optimisation focuses on finding near-optimal scheduling strategies which can support the smooth integration of AGVs with the manufacturing process to reach the production KPIs, rather than searching for a global optimisation solution. During the running manufacturing process, time is essential and searching for a global optimisation solution costs time, and even it may cause the failure of optimisation because of the late reaction.

After applying the SAMS architecture on a shop floor, the real-time shop floor manufacturing process optimisation is developed as a closed-loop. SAMS optimises the shop floor logistics along with the whole manufacturing process to improve the production performance. The operation mechanism is shown in Figure 3.2.

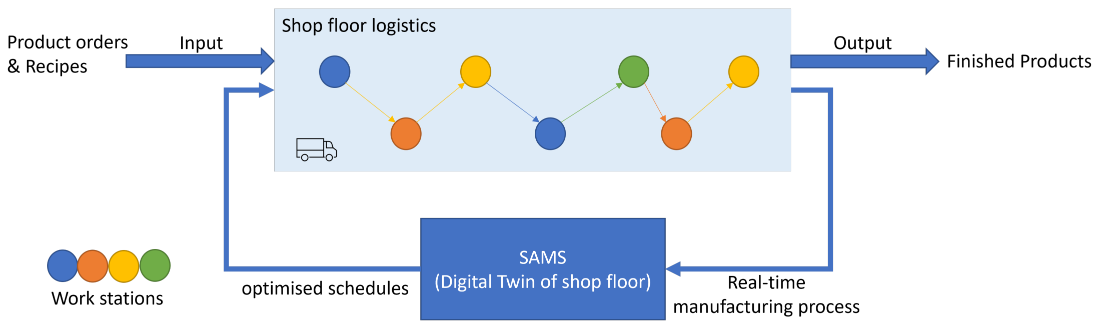

Figure 3.2: The operation mechanism of SAMS

In this operation mechanism system, SAMS collects and monitors shop floor data to create a digital twin environment. Also, SAMS allows analysing and predicting the physical production performance and automatically adjusting schedules and configuration solutions. An overview of the SAMS shop floor logistics schedule optimisation approach is depicted in Figure 3.3.

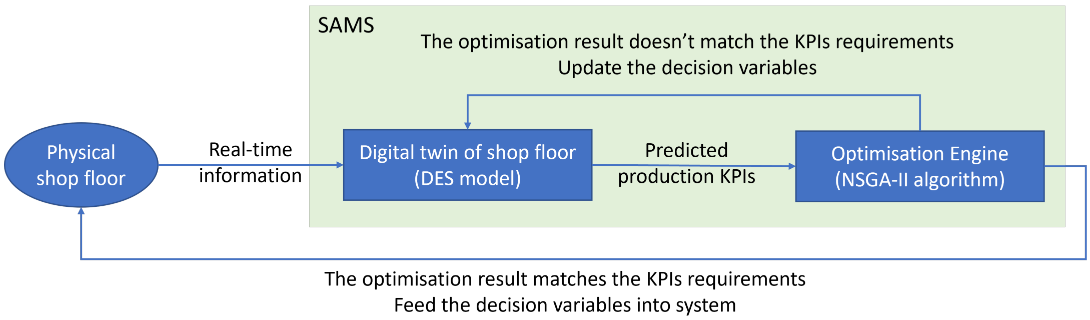

Figure 3.3: The running procedure of SAMS

The digital twin-based DES model simulates the manufacturing process and AGV-based shop floor logistics to predict the production KPIs (e.g., the utilisation of AGVs and work stations, the delivery JIT performance and the overall AGV energy consumptions) by analysing the different decision parameters, including the number of AGVs, AGV dispatching time, and the AGV charging threshold. If the predicted KPIs do not meet KPI requirements, the optimisation engine will generate new schedules by changing decision parameters and pass to the DT model for evaluation. Once the production KPIs meet with these KPIs requirements which are defined by operators, the related decision parameter will send to shop floor MES and fleet manager of AGVs by SAMS. The optimisation procedure starts when the real-time information is provided by the physical shop floor.

To implement the SAMS methodology, three main modules are developed, namely the data transaction module, the DT-based optimisation module, and the decision-making module, which are introduced specifically in the following sections.

3.3 Data transactional module

3.3.1 Introduction

In this section, the data transaction module is described. This module provides a real-time, consistent and robust inter-communication among the physical shop floor manufacturing systems, including work stations, AGVs, fleet manager, and IT/OT infrastructures, with the DT-based scheduling optimisation module along with the whole manufacturing process. In this module, the OPC UA server is used as a data-sharing platform to subscribe the operation information from shop floors and publish the optimised schedules for AGV systems and shop floor MES. The data transaction module as a middleware supports the real-time shop floor operation information monitoring for SAMS, which also enables the optimisation and integration of AGV systems with the manufacturing process on the shop floor.

3.3.2 Framework of data transactional module

In the data transaction module, an aggregation server is developed by using the OPC UA protocol to enable the data transaction between the cyber and physical systems. Using this aggregation server, the physical devices can publish the real-time status to update operation information, and the manufacturing IT system (e.g., MES, ERP and fleet manager of AGVs.) can subscribe to the optimised scheduling strategies for shop floor logistics scheduling and manufacturing process optimisation. The OPC UA based data transaction architecture is shown in Figure 3.4.

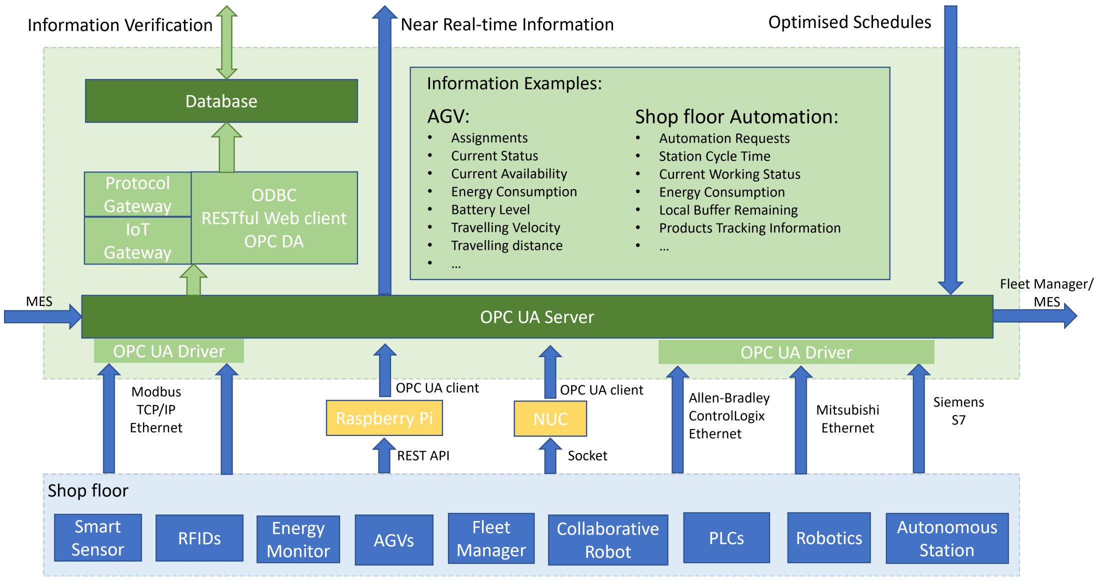

Figure 3.4: The framework of OPC UA-based data transactional module

The shop floor consists of various equipment (such as PLCs, robots, AGVs, smart sensors, RFIDs and energy monitors), which publish their operation data to this OPC UA server. However, the connection between the PLCs and OPC UA server is not direct. They usually require vendor dependent engineering tools, like APIs, clients and hardware drivers. Specifically, drivers are needed for various PLCs vendors, such as Siemens, Mitsubishi, Allen-Bradley ControlLogix Ethernet, and Modbus Transmission Control Protocol/Internet Protocol (TCP/IP) Ethernet. For the fleet manager, AGVs and collaborative robots, due to the limitation of the APIs, a middleware device (e.g., NUC, Raspberry Pi and single-board Personal Computer (PC)) is necessary for developing the OPC UA client to support the communication with the OPC UA server. To enable the connectivity between these devices and the OPC UA server, in this thesis, the KEPServerEX-based OPC UA server [196] is used to support data communication between PLCs, robotics and IoTs on the shop floor. It is a simply OPC UA server that provides an interface to various communication protocols.

The data required to carry out optimisation can be categorised into AGV related information, work stations related information and products manufacturing information. The details of the data are provided in Table 3.1, 3.2 and 3.3.

Table 3.1: The AGV related information

Name	Type	Description	Example
Number of AGVs	Integer	The number of AGVs on the shop floor.	5 (units)
AGV loaded/ unloaded energy consumption (AH/hr)	JSON String	The energy consumption of AGVs when it is loaded or unloaded.	[{”AGV_ID”:1, ”LoadedEnergy”:20.0, ”UnloadedEnergy”:7.0}]
AGV travel time between stations(s)	JSON String	The time AGV spends between different work stations.	[{”Start_Station_ID”: 1, ”Destination_Station_ID”: 2, ”Duration”: 40.5}]
AGV capacity	JSON String	The loading capacity of each AGV	[{”AGV_ID”:1, ”Capacity”: 1, “ProductType”: “18650”, “Loading_unloadingTime”: 10.0, ”BatteryCapacity”:120.0, ”IdleEnergy”:5.0, ”ChargingRate”:35.0}]
AGV idle energy consumption (AH/hr)		The energy consumption of AGVs when it is idle.	
Battery capacity of AGV (AH)		The maximum battery capacity of each AGV	
AGV loading/ unloading time (s)		The time AGV spends on loading or unloading products.	
AGV Charging rate (AH/hr)		The charging rate of each AGV	

Table 3.2: The work stations related information

Name	Type	Description	Example
Number of work stations	Integer	The number of work stations on the shop floor	20 (units)
Map of station type and ID	JSON String	Define the function of every work stations	{Station:[{” Type” : ”Welding”, ” ID” : 1}, {” Type” : ”Assembly”, ” ID” : 2}]}
Cycle time of work stations (s)	JSON String	The time of work station spends to manufacturing a product.	{[{”ID” :1, ”CycleTime” : 80.5, ”Status” : ”Idle”}, {”ID” : 2, ”Cycle” : 50.9, ”Status” : ”Busy”}]}
Status of work stations	JSON String	The status of work stations, including idle, busy, block, breakdown.	

Table 3.3: The products manufacturing related information

Name	Type	Description	Example
Order Pool	JSON String	The order queue from MES.	{[{"Order_ID":1, "Product-Type",18650}, {"Order_ID":2, "Product-Type",26650}]}
Process Recipe	JSON String	The sequence to manufacture a product defined by MES	{Recipe:[{"ProductType" : "18650", "StationType": "Lega-cyLoop","sequence":1}, {"ProductType" : "18650", "StationType": "Launch", "se-quence":2}, {"ProductType" : "18650", "StationType": "Weld-ing","sequence":3}]}

The AGV related information is fed to the OPC UA through Modbus TCP/IP protocol and REST API over Hypertext Transfer Protocol (HTTP) from the fleet manager and individual AGVs, and the work stations related information is collected from the station PLCs. The products manufacturing related information, such as the order pool and the process recipes, are accessed from MES. The optimised solution, such as the prioritised order, AGV delivery schedules and charging plans, are published to fleet manager and shop floor MES via this OPC UA-based data transaction module.

Furthermore, to comprehensively analyse the shop floor manufacturing data, predict the production KPIs, such as the delivery JIT, utilisation of AGVs and work stations, and the overall energy consumption, a database based on the Microsoft (MS) SQL server is built in the data transaction module to store these shop floor historical operation data. The AGV, work stations and products order related historical and real-time information are stored in this database by using JavaScript Object Notation (JSON) formatted payload and following pre-defined data types which are shown in the above tables. Moreover, these historical data are used to verify the accuracy of the DES model. Specifically, with the recorded operation information such as the AGV delivery schedules, work station cycle times, and AGV energy consumption, the DES model is able to predict production KPIs. Afterwards, the recorded production KPIs and DES predicted results can be cross-checked to validate the DES accuracy.

The data transaction module enables real-time operation information monitoring and historical data storage. Thus, to convert the data between different communication protocols, the data transaction module is implemented with: 1)a REST API to publish the AGV missions including delivery tasks and charging threshold to its fleet manager. 2) an Open Database Connectivity (ODBC) interface that allows creating a connection with the MS SQL server. 3) an OPC UA client to subscribe to the updates from the OPC UA server and publish the changes to other SAMS modules.

In summary, this OPC UA-based data transaction module works as an intermediate layer to connect the shop floor manufacturing systems with the optimisation module. In this way, It can collect and monitor the real-time shop floor operation information for the DT-based optimisation module to optimise the shop floor logistics. Moreover, it can transfer the optimised logistics schedules and order sequences to the fleet manager of AGVs and MES to keep the manufacturing process productive and efficient.

3.4 DT-based optimisation module

3.4.1 Introduction

This section presents the DT-based optimisation module. The aim of this module is to optimise the AGV-based shop floor logistics during the manufacturing process by simulating and analysing the shop floor manufacturing systems.

In this module, the DES-based simulation modelling software is used to create a DT model of the shop floor environment for simulating and predicting the production process and KPIs, such as the utilisation of AGVs, delivery JIT performance and energy consumption. Moreover, a hybrid optimisation method combining an NSGA-II algorithm and a DES model is implemented to find the suitable logistics schedules and the best AGV charging threshold. The framework of this optimisation module is depicted in Figure 3.5.

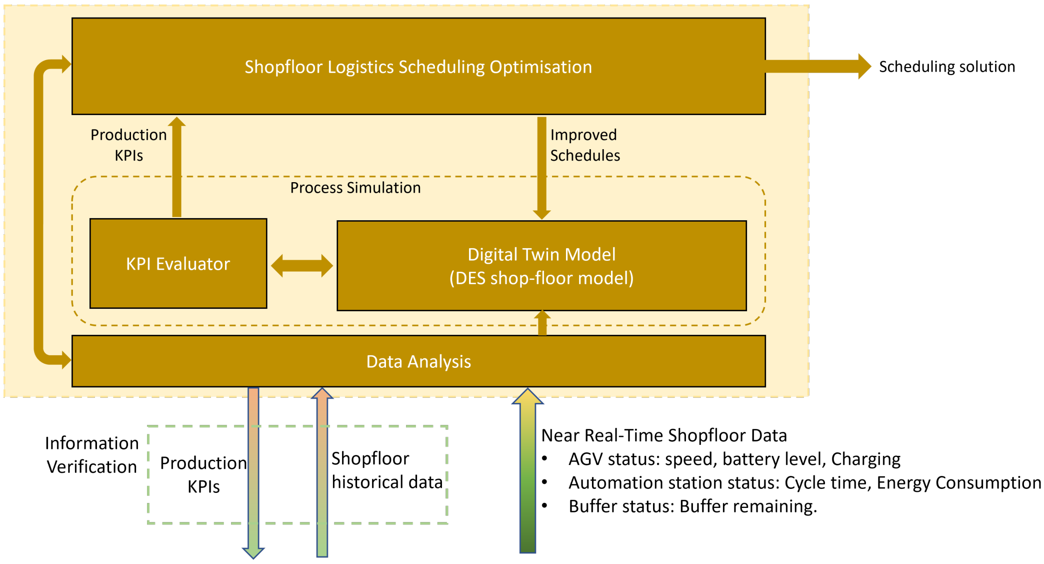

Figure 3.5: The framework of the optimisation module

The optimisation module employs the DT technology to replicate the physical shop floor manufacturing process and support the scheduling decision-making, such as raw material delivery schedules, work-in-process product delivery schedules and finished goods collection schedules. This module is composed of three main components: data analysis, process simulation and shop floor logistics optimisation. These three parts are integrated seamlessly to provide near-optimal scheduling strategies by monitoring, analysing and simulating the real-time manufacturing process.

3.4.2 Data analysis

The data analysis section collects the real-time operational data from the shop floor, analyses their distribution trend, and feeds them into the process simulation section and optimisation section for real-time shop floor logistics optimisation. The data analysis section collects the shop floor historical data from the database located in the data transaction module. This data is used to analyse the distribution trend of work stations cycle time and the energy consumption of AGVs which are applied to generate the DT model of the shop floor manufacturing system in the process simulation section.

To subscribe to the real-time information from AGVs and work stations on the shop floor, the data analysis section needs access to the OPC UA server located in the data transaction module. A python-based OPC UA client is created in the data analysis section. At the same time, a subscription mechanism is built up to monitor the updates of AGV information including the running status, charging status and energy consumptions, and the updates of work stations status including cycle time and working status, as well as the changes of line-side buffer status.

Also, a python-based SQL client is developed to collect the shop floor historical data from the database established in the data transaction module. In the data analysis section, the one-sample Kolmogorov-Smirnov Test(KS Test) [197] has been used to assess the distribution trend of these data. The one-sample KS Test is a nonparametric test to analyse the relationship between the hypothesis Cumulative Distribution Function (CDF) and the data set CDF. For instance, based on the historical distribution trend of machine cycle time, AGV energy consumptions and charging rate, they are analysed to follow a normal distribution. As a result, the data analysis section combines the real-time and historical data to predict the distribution trend of the machine cycle time, AGV energy consumptions and charging rate during the manufacturing process. Then, the data analysis section feeds this predicted operation information, including AGV operational data, order information and machine operational status, into the DT model.

To update the real-time and predicted manufacturing process information for the DT model, a python-based 'OLE Automation Controller' is created. Thus, the data analysis section can provide real-time manufacturing information, including the process information and system interruptions, to keep the accuracy of DT simulation.

Meanwhile, the data analysis section monitors the operation information of the shop floor by subscribing to the OPC UA server during the manufacturing process. Once a machine or AGV breaks down, the abnormalities can be detected and then the data analysis section feeds the changes into the DT model to update the digital environment. Also, when MES generates a new product order list, the data analysis section can send it to the optimisation section for re-scheduling shop floor logistics to improve the production KPIs like JIT performance, energy consumptions and utilisation of working stations.

3.4.3 Optimisation problem description

In this section, a battery pack assembly line scenario is carried out to explain the optimisation problem. In this scenario, the raw materials and work-in-progress products are transported between stations by multiple AGVs following the specific manufacturing recipes. It is notable that each product should be passed through every station, which is instructed by the battery pack assembly recipe. It is presumed that each product has a requested arrival time for every work station. Thus, the JIT means a product has been delivered to the right work station at the requested time without earliness or lateness. The two main objectives of real-time AGV schedules optimisation are: 1) to maximise the performance of JIT (i.e., minimise the error of earliness/lateness); 2) to minimise the overall AGV energy consumptions.

A schematic diagram of the battery assembly is illustrated in Figure 3.6. This scenario consists of five stages workstations: 1)Station I, where the battery pack frames are placed initially by an operator; 2)Station II, the battery cells are assembled into a pack by using industrial robots connected via a conveyor system; 3) Station III, the battery pack is welded at this station; 4) Station IV, the battery pack is inspected for quality check by a camera-based vision system; 5) Station V, the battery pack is packed and stored at this station. All the AGVs are waiting at the Parking Area for new delivery tasks. In the AGV recharging procedure, the AGV need to drive to the charging station once the battery level is lower than the threshold. Additionally, AGVs park at the Parking Area after completing the last delivery task. The empty battery packs are delivered by AGV from Launch Station and are assembled through each station following the pre-define assembly recipe until they are delivered to Packing Station. The robotics insert the battery cells at Legacy Loop, and once the battery cell runs out, the AGVs are requested to deliver a new pack of battery cells from the warehouse. The battery packs are collected and transported to the next station via AGVs, depending on the schedule generated by the optimisation module. For the AGV travelling in the shop floor area, the working area map is pre-defined in the AGV control system, and the collision-free routing is continuously supervised and planned by a control system embedded in the AGV system.

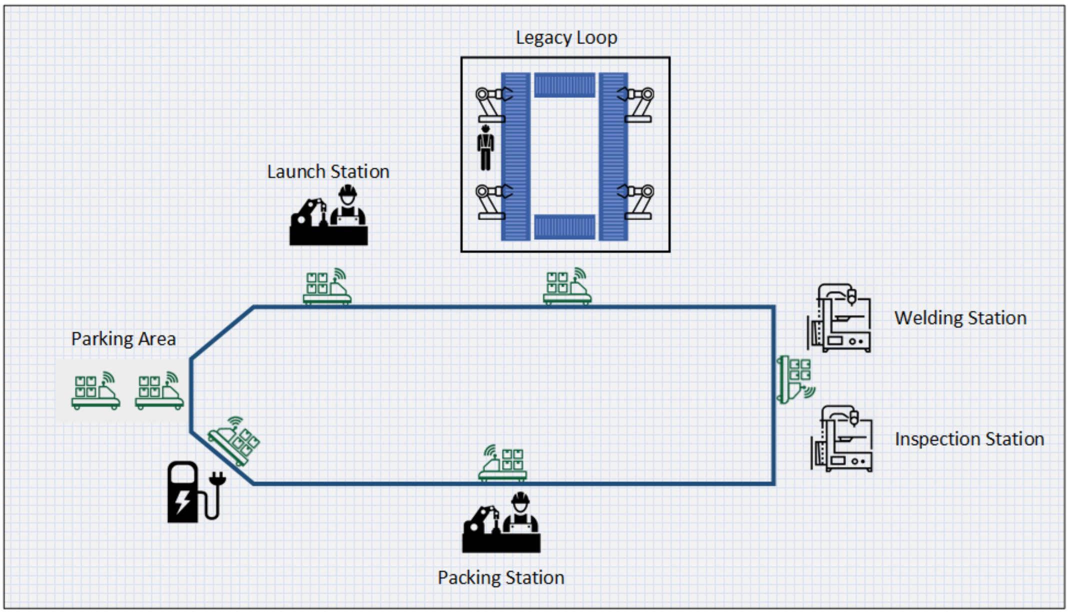

Figure 3.6: A schematic of battery assembly line model

3.4.4 Optimisation problem formulation

The explanation of mathematics notations for the shop floor logistics problems is given in Table 3.4

Table 3.4: The list of notations

Notation	Description
Sets	
S	Set of work stations
T	Set of tasks
N	Set of AGVs
Indices	
s	Index of work station, $s \in \{1, 2, ..., S\}$
t	Index of production task, $t \in \{1, 2, ..., T\}$
n	Index of AGV, $n \in \{1, 2, ..., N\}$
Parameters	
Q_{n_o}	The weight of AGV n without load
Q_{ijnt}	The weight of AGV n with loads,
	when travels between station i and j for task t
α	Earliness cost penalty coefficient
β	Lateness cost penalty coefficient
PT_{ts}	Processing time of task t at working station s
C_i	Delivery completion time of task t
R_i	Delivered request time of task t
S_{ts}	Starting time of task t at working station s
D_{ts}	Completion time of task t at working station s
dis_{ij}	Distance between work station i and j, also, $i \neq j$
r_t	Release time of the task t into the system
Decision Variables	
M_{ts}	1 if work station s working on task t, otherwise 0
X_{ijnt}	1 if AGV n travels between work station i and j
	for task t, otherwise 0

The two main objectives functions are considered in this mathematical model, defined as below:

- Objective function 1: to minimise the cost of total AGV delivery earliness (i.e., the time of AGV waiting at the working station because of the delivery early arrival) and lateness (i.e., the time of working station waiting for the material because of the delivery late arrival).

$$f_1 = \sum_{i=1}^{|T|} \alpha \cdot max\{0, R_i - C_i\} + \sum_{i=1}^{|T|} \beta \cdot max\{0, C_i - R_i\} \tag{3.1}$$

  In the formulation, the earliness and lateness penalty cost vary in the different scenarios, and the values are decided based on the situation and the KPI requirements of the manufacturing system. However, in general, the lateness has more impacts on manufacturing performance because it may stop the overall production process.

- Objective function 2: to minimise the overall energy consumption of AGVs, and it is considered via the execution states and travelling distances. The formulated as below:

$$f_2 = \sum_{i=1}^{S}\sum_{j=1}^{S}\sum_{t=1}^{T}\sum_{n=1}^{N} dis_{ij} X_{ijnt} F(Q_{n_o} + Q_{ijnt}) \tag{3.2}$$

  Where $F(Q_{n_0} + Q_{ijnt})$ means the energy consumption rate relationship with loads of AGV at a unit travel distance.

To describe constrains in the shop floor logistics and assembly process, the objectives are bound by the following constraints:

$$S_{t(s+1)} \geq D_{ts}, \quad t = 1, ..., T, s = 1, ..., S \tag{3.3}$$

$$S_{ts} - S_{(t-1)s} \geq PT_{(t-1)s}, \quad t = 1, ..., T, s = 1, ..., S \tag{3.4}$$

$$S_{t1} \geq r_t, \quad t = 1, ..., T \tag{3.5}$$

$$max\{\sum_{t=1}^{T} X_{ijnt}\} = 1, \quad i = 1, ..., S, j = 1, ..., S, n = 1, ..., N \tag{3.6}$$

$$X_{ijnt}, M_{ts} \in 0, 1, i = 1, ..., S, j = 1, ..., S, t = 1, ...T, s = 1, ..., S, n = 1, ..., N \tag{3.7}$$

In these constraint equations, Constraint (3.3) ensures that, for an individual task, the task cannot be started before it has been completed in the previous station. Constraint (3.4) ensures that a machine can only process one task at a time. Constraint (3.5) enforces that, for any task, the first assembly process only can be started after the raw product has been delivered to the shop floor. Constraint (3.6) emphasises that an AGV can only execute one delivery task at a time. Constraint (3.7) shows the binary nature of decision variables.

3.4.5 Optimisation problem assumptions

The assumptions made to carry out optimisation are listed below:

- The manufacturing recipes are pre-defined in the shop floor database, and AGV follows the recipe to deliver products or materials.
- The product order information is continuously published to the optimisation module in real-time.
- A machine only can process one job at the same time.
- Considering the fast charging period and the lifetime of the AGV battery, the charging threshold of AGV has to be assigned between 20% to 80%.
- AGV loading capacity is fixed and limited.
- The AGV fleet has enough capacity to cover all delivery jobs.
- The AGV cannot be requested when the remaining battery level cannot support the AGV to complete the new task and travel back to charging stations.
- The AGV only can execute one delivery job at a time.

3.4.6 Process simulation

In the scheduling optimisation module, a process simulation section is developed to duplicate the physical shop floor environment using digital twin technology. The process simulation section simulates the shop floor manufacturing process by combining the real-time operation information with a DES model, which provides an opportunity to replicate and analyse the complex shop floor logistics and production processes, especially for the prediction of manufacturing system KPIs.

Manufacturing KPIs are a group of quantifiable metrics that manufacturing companies usually use to assess their overall manufacturing performance [198]. Commonly used shop floor KPIs include the utilisation of AGV, utilisation of work stations, energy consumption, workload balance, JIT performance of shop floor logistics, finished product quality, etc.

In the process simulation section, the DT model of the physical manufacturing process is created using the DES tool (i.e., WITNESS Horizon). It provides predicted KPIs based on the real-time production process information passed by the data analysis section. These KPIs are fed to the optimisation section to carry out schedule optimisation by optimising AGV dispatch time and the charging threshold of AGVs.

The information required to develop the DES model is shown in Figure 3.7. The system configuration parameters include: i) number of work stations; ii) number of AGVs in shop floor logistics; iii) distance between different work stations, AGV parking area, charging stations and shop floor warehouse; iv) mapping of work station and job types; v) manufacturing recipes. These parameters are pre-defined in the simulation model. The real-time manufacturing process data include: i) cycle time of each work station for various product variants; ii) status of each work stations, (e.g., breakdown, execution and idle); iii) AGV travel time between different work stations; iv) energy consumption rate of AGVs during different states (i.e., idle, loaded and unloaded); v) charging rate of AGVs; vi) order pool from MES. These real-time operation information are constantly updated from the shop floor through the OPC UA server whose data scan rate is up to 100Hz and are fed to the DES model after being processed by the data analysis section.

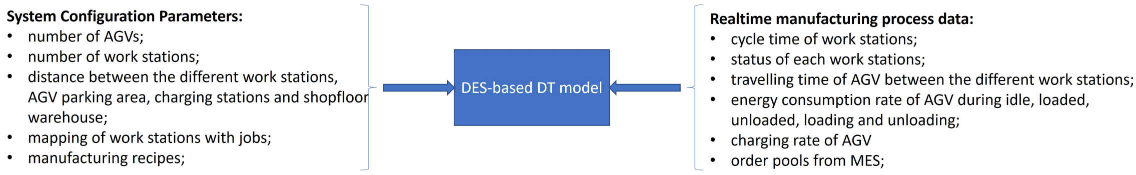

Figure 3.7: The information details for the DES-based DT model

The real-time operational information fed to the DES model and the simulation of the physical manufacturing process significantly increase the accuracy of the performance prediction. To predict the process data, like the cycle time of work stations, energy consumption and charging rate of AGV, the historical and real-time data from the database (via data transaction module) are analysed by the data analysis section. For example, the cycle time of the work station is analysed following a normal distribution curve [199], and the distribution parameter (i.e., mean and standard deviation) is calculated to predict the cycle time which is updated to the DES model.

To evaluate the performance of the simulated shop floor manufacturing process, the KPI evaluator is implemented through a group of equations, shown in Table 3.5:

Table 3.5: The KPIs for optimisation

KPIs	Equations	Description
Shipped products	$Number\ of\ finished\ products - Failed\ products$	The total number of good products.
JIT cost of shop floor logistics	Referred from Equation (3.1)	The cost of JIT with consideration of earliness and latenes.
Energy consumption of AGVs	Referred from Equation (3.2)	The total energy consumption of AGVs under various actions.

Once the simulation of the DES model is completed, these equations are triggered to calculate the KPIs of this simulated process, and these KPIs are published to the optimisation section for further evaluation and optimisation.

3.4.7 Optimisation section

3.4.7.1 NSGA-II-based optimisation method

The meta-heuristic based optimisation algorithm is widely used in the scheduling problem [200]. The NSGA-II [201], an evolutionary optimisation algorithm, is developed based on GA [202] and NSGA [203]. It aims to solve multiple objective problems with an acceptable computational complexity (O(MN^2), where M is the number of objective and N is the size of population). NSGA-II was proposed in 2002 by Deb et al. [201] with a fast computational and elitist capability. Also, NSGA-II shows better performance when solving several optimisation problems, especially a two-objective optimisation problem, compared to other multi-objective evolutionary algorithms (MOEAs). NSGA-II has been applied in many manufacturing scheduling problems recently to find the near-optimal or optimal solutions efficiently [142, 204–210]. For example, Akbar and Irohara [205] presented a modified NSGA-II variant by using different decoding schemes to minimise the overall makespan and maximise the balance of the workload. They concluded that the NSGA-II is the fittest algorithm to solve the scheduling problems compared to other methods, like SPEA2. Also, Souier et al. [210] applied NSGA-II into an AGV real-time routing problem in FMS to maximise system reliability and minimise system deadlocks considering the workload and machine utilisation. The authors claimed that NSGA-II provided better performance in routing optimisation compared with that of classical GA. Similarly, Bandyopadhyay and Bhattacharya [211] proposed that NSGA-II could be used for machine schedules to address three objectives: 1) minimise the tardiness costs; 2) minimise the deterioration costs; 3) minimise the overall makespan. Their simulation results showed high efficiency of NSGA-II. Yusoff et al. [212] presented a summary of NSGA-II advantages in the machine's operations, including the traditional and modern machine operation, and concluded its reliable capability of simultaneous multi-objective optimisation.

Please note, the DES software provides a built-in optimiser which tests the different parameter changes of the model. And it can indicate the optimisation solution based on the objective function from the model builder. However, some of the built-in optimisers use an exhaustive algorithm to find the best solution, which causes a large compute complexity in a large-scale system. And most of the built-in optimisers could not handle multiple-objective optimisation. The optimised results have to be analysed by external data analytics software. Moreover, the built-in optimiser and the DES model have to be run on the same PC. Thus, the optimiser cannot be executed parallelly in another powerful PC or server. It could limit the efficiency of the optimisation process in a large-scale shop floor manufacturing system.

Therefore, in the optimisation section, the NSGA-II is selected to maximise the JIT performance of logistics and minimise the overall AGVs energy consumption by optimising dispatch time of AGVs for material delivery/collection and charging threshold. Taking the complexity of shop floor manufacturing operations into consideration, the mathematical model cannot fully reflect shop floor scenarios. However, the DT-based DES model can duplicate the shop floor environments and predict the production KPIs. Therefore, in this thesis, the fitness functions of NSGA-II are calculated by using a DES-based process simulation engine. The flowchart for generating fitness function results is shown in Figure 3.8. In the initialisation step, the DES model is set up with pre-defined manufacturing process parameters and shop floor manufacturing process scenarios. Once the initial run is executed, the system configuration parameters and updated real-time machine and AGV information are fed into the models. When the scenario simulation is complete, the predicted cost of JIT and energy consumptions of AGVs are calculated by the DES-based fitness function and then these KPIs are returned to the optimisation section for the current generation evaluation. The overview of process steps for the modified NSGA-II optimisation method is given in Figure 3.9.

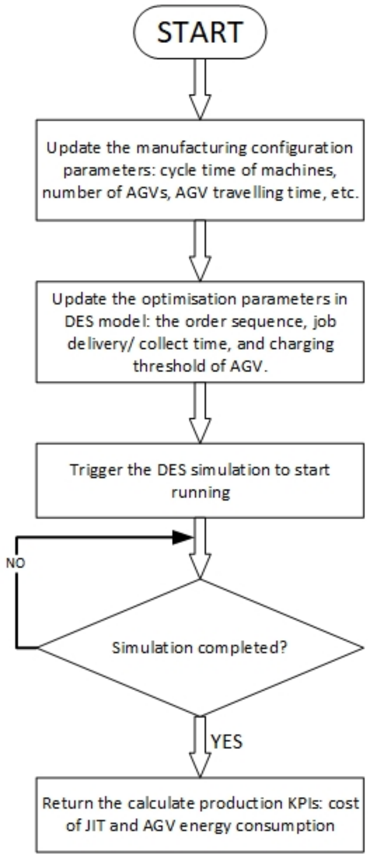

Figure 3.8: The workflow for DES-based fitness function

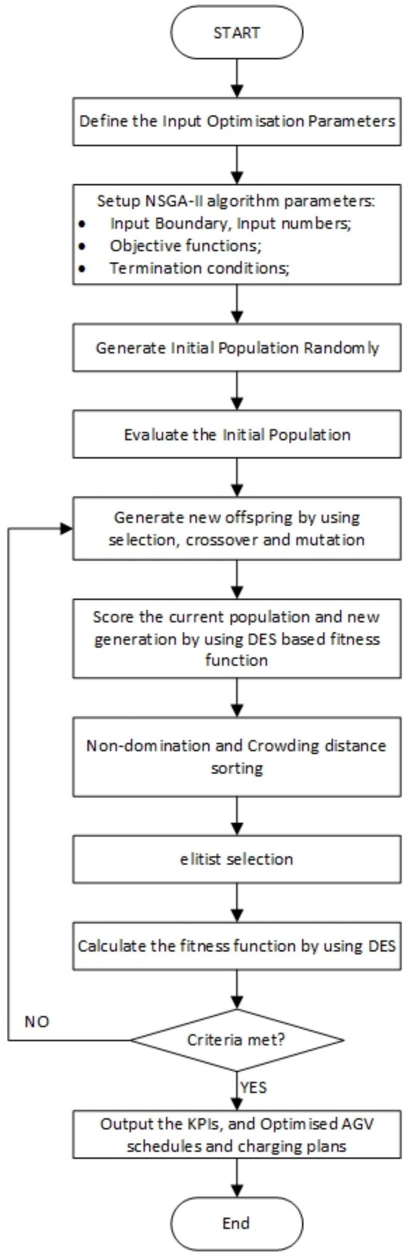

Figure 3.9: The workflow for modified NSGA-II

3.4.7.2 Proposed NSGA-II

The NSGA-II-based optimisation mechanism is presented in this section for scheduling AGV dispatching time and charging thresholds. In the proposed algorithm, the fitness functions are: 1) The cost of JIT delivery. 2) The energy consumption of AGVs. These function results are obtained from the DES-based process simulation section. The pseudo-codes for obtaining the fitness function results are shown in Table 3.6.

Table 3.6: The pseudo-codes of the fitness function

Algorithm 1 Obtain the manufacturing system KPIs from the DES-based process simulation engine
Input: The real-time manufacturing process information and the decision variables
Output: The manufacturing process KPIs: cost of JIT and cumulative energy consumption of AGVs
Pseudo-code of the Fitness Function
Initialise the system parameters: cycle time, AGV travel time, energy costs
Update the decision variables: AGV calling time of each station, charging threshold
Starting the DES simulation
while (the DES simulation is running) **do**
Check the DES simulation status
end while
f1 $<-$ cost of delivery JIT
f2 $<-$ energy consumption of AGVs
Return f1, f2

The algorithm variables and constraints, including the decision parameters, the variable number, boundaries, fitness functions and termination conditions are defined. The initial population is then generated based on pre-defined constraints and is evaluated using fitness functions. The parents are selected by sorting the non-domination Pareto ranking and crowding distance. Following this, the new generation is reproduced by the selection, elitism, crossover and mutation processes. The KPIs of the manufacturing system, like the cost of JIT delivery and energy consumption of AGVs, are the objectives to be optimised. The algorithm is terminated once reaching the target value is reached. Also, the algorithm stops reproducing when the maximum number of generations is reached. The pseudo-code of the proposed NSGA-II algorithm is shown in Table 3.7.

Table 3.7: The pseudo-codes of the proposed NSGA-II

Algorithm 2 Proposed NSGA-II
1: Initialise the optimisation parameter, algorithm variables and constraints.
2: Initialise the populations.
3: Evaluate the initial populations by using **Algorithm 1**
4: **while** (fitness function results do not reach the stopping conditions) **or** (iteration < MaxGeneration) **do**
5: Children < − Population(select, crossover, and mutation)
6: Score (Population, Children) by using **Algorithm 1**
7: Non-domination front ranking and crowding distance sorting
8: Selected < − Select the parents from sorted Pareto Front and Crowding distance
9: Evaluate the Selected Population fitness by using **Algorithm 1**
10: **end while**
11: Output the best solutions: AGV dispatch time and charging threshold of AGVs

3.4.7.3 Initialising Algorithm Parameters

Each generation is split into two segments: the AGV dispatch time and charging thresholds of AGV. Figure 3.10 shows two examples of the population structure.

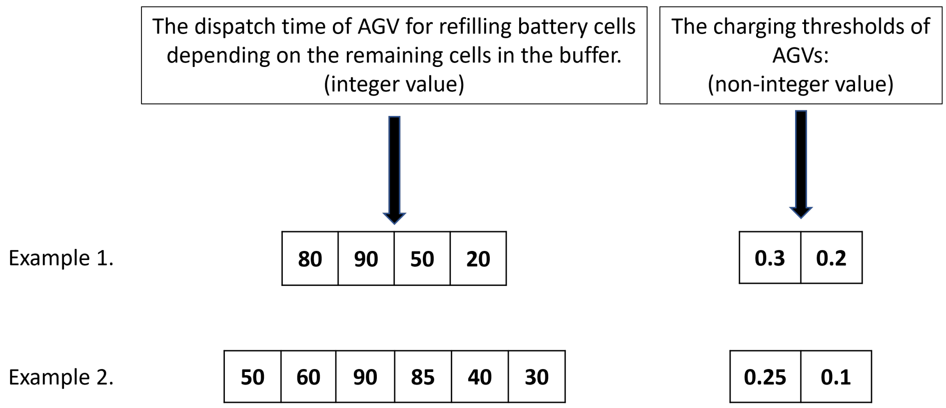

Figure 3.10: Two examples of the population structure

In example 1, the manufacturing system includes four local battery cell buffers, and example 2 consists of six local battery cell buffers. The left-hand side array illustrates the AGV dispatch time for each buffer depending on its remaining battery cells, for example, in this scenario, 80 indicates that the first local battery cell buffer requests a new pack of battery cells delivery when only 80 battery cells are left. The right-hand side array represents the charging thresholds for AGV, which comprises two parts: 1) Alert level, for instance, 0.3 means when AGV battery percentage is lower than 30%, the AGV is called to re-charge if the charging station has free space, otherwise the AGV can carry on its delivery tasks. 2) Alarm level, for example, 0.2 means when AGV battery percentage is lower than 20%, the AGV must be re-charged and cannot carry on any new delivery task.

3.4.7.4 Initialising population

The first population is created by using a uniform distribution based random generator. The first part of the population are consecutive integers starting from 1, and the population size is considered by the number of local buffers. The second part contains two non-integer values, and they are between 0.2 and 0.8. Also, the Alert level is always higher or equal to the Alarm level. The NSAG-II algorithm constraints, including the number, the types and the boundaries of variables, are set up depending on these conditions.

3.4.7.5 Generate the child population

The new generation is created using selection, crossover, and mutation [201].

1) Selection: The parents are selected using the stochastic universal selection algorithm (see [213]) to produce the child population.

2) Crossover: In this step, the two groups of selected parents are combined to generate the next generation. The crossover genes are chosen randomly from parents, and the coordinate of these genes are the same in both parents. The crossover fraction decides the number of crossover children population. Here is an example to show the crossover strategy in Figure 3.11.

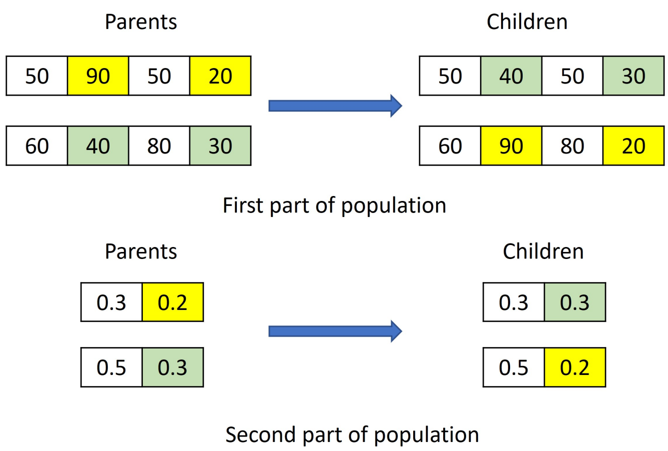

Figure 3.11: The example of crossover strategy

3) Mutation: The mutation procedure keeps genes diverse while producing offspring. The Gaussian distribution ([214]) is used for generating a mutation child population from parents' genes. Figure 3.12 shows an example of a mutation.

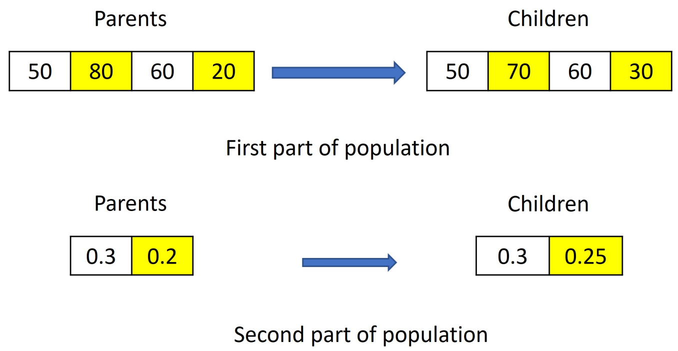

Figure 3.12: The example of mutation strategy

3.4.7.6 Fast non-dominated and crowding distance sorting

In this step, all objective results of the current population are calculated, and the non-dominated [215] solutions are considered.

1) The population and child population are combined and scored for all fitness functions.

2) Identify all different Pareto fronts. The lower rank Pareto front solutions are better or equal to those in the higher rank Pareto fronts. Moreover, the lower rank Pareto front has a higher chance to be selected as the new parent population. An example of Pareto front-ranking is shown in Figure 3.13.

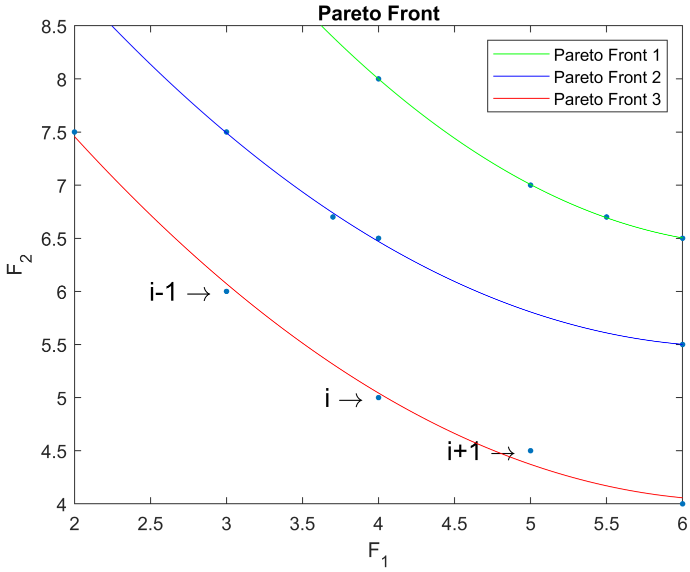

Figure 3.13: An example of Pareto front-ranking

3) Select the population from the lower rank Pareto front solutions until the population size is over the limitation. Then, reduce the population of the last selected Pareto front by crowding selection, for instance, in Figure 3.13. The Pareto front 1 has the lowest rank, and Pareto Front 3 has the highest rank. There are four solutions in Pareto front 1, four in Pareto front 2, and five in Pareto front 3. If the population size is ten, The Pareto front 1 and the Pareto front 2 are selected as new parents. The Pareto front 3 population will be reduced by using crowing distance sorting to meet the population size requirement.

4) Crowding Distance Sorting. Firstly, the closeness of each solution to its nearest neighbours in the same Pareto front is sorted. For example, in Figure 3.13, the closeness of solution i in Pareto Front 3 is:

$$i_d = \sum_{j=1}^{n}(|f_j^{i+1} - f_j^{i-1}|) \tag{3.8}$$

where, i_d is the closeness of individual i, f_j^{i+1} and f_j^{i-1} are fitness function (f_j) values of the individual $i+1$ and $i-1$. While, for the individual at the extreme position, the crowding distance is assigned as infinity.

Then, these closenesses are sorted. To keep the diversity of the population, the selection starts from the individual with a higher crowding distance until the population size is reached.

5) Elitism. The new population is generated from selected the Pareto front with lower rank and individuals with higher crowding distance. The elitism schema is described in Figure 3.14.

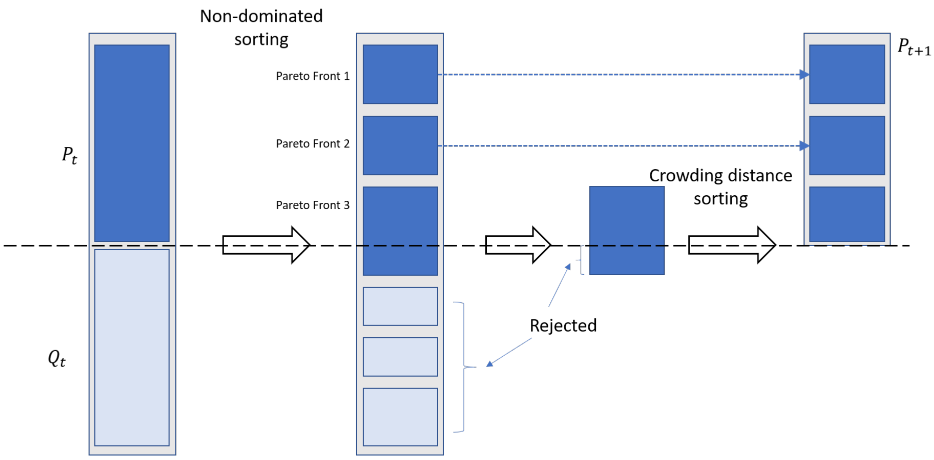

Figure 3.14: The elitism schema of the NSGA-II [201]

3.4.7.7 Evaluation and iteration

The generated population are evaluated by the fitness function that is calculated using the DT model. Producing the new generations is terminated when the production KPIs meet the manufacturing system requirement or the generation number reaches the pre-defined maximum generation.

3.4.8 Communication structure of these three sections

In the optimisation module, the AGV-based shop floor logistics deliveries are scheduled or rescheduled by combining an NSGA-II algorithm with a DT model considering the real-time manufacturing process. Firstly, the process scenarios and system configuration parameters are configured in the DES tool to create the production process simulation model. Secondly, the initial schedule plans and real-time shop floor information are updated in the DES model. Then, predicted production KPIs are generated using the input schedules and sent to the optimisation section. An NSGA-II algorithm is used in the optimisation section to optimise the shop floor logistics performance by maximising the JIT delivery performance and minimising the energy consumption of AGVs simultaneously.

Additionally, when an interruption happens during the manufacturing process, for instance, a machine breakdown, AGV breakdown, or a new customer order occur on the shop floor, it is updated into the process simulation section via the data analysis section. Then the process simulation engine triggers the re-schedule mechanism by updating the DES model and optimisation conditions to reduce the effects of the interruptions on the overall manufacturing process. The data flow between the process simulation section and optimisation section is illustrated in Figure 3.15.

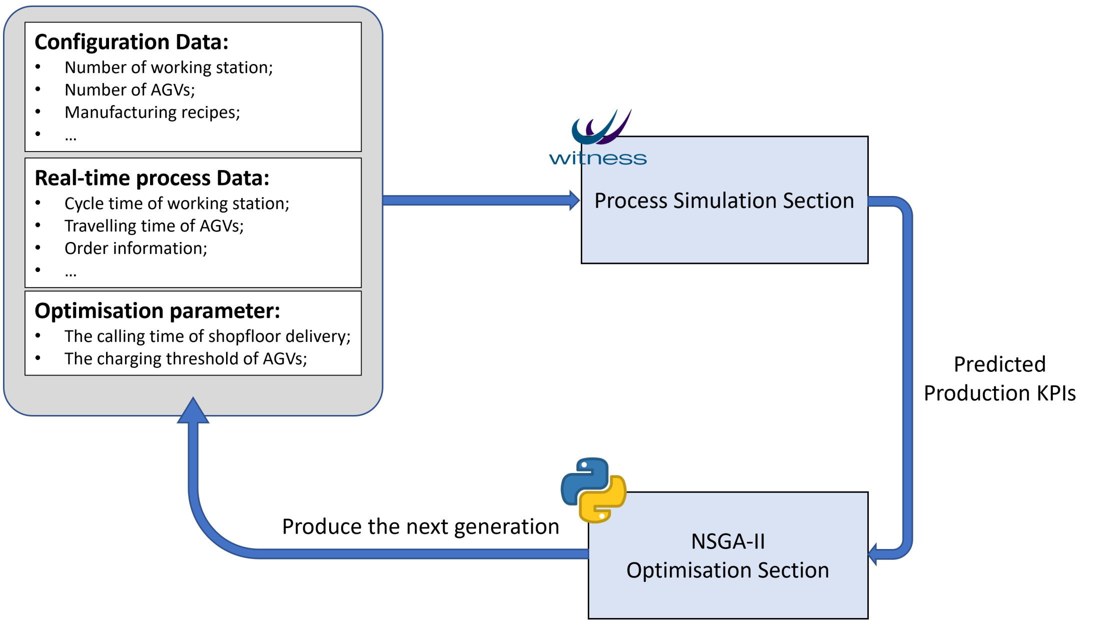

Figure 3.15: The data flow between the process simulation and optimisation

The communication architecture for data sharing between these three sections is shown in Figure 3.16.

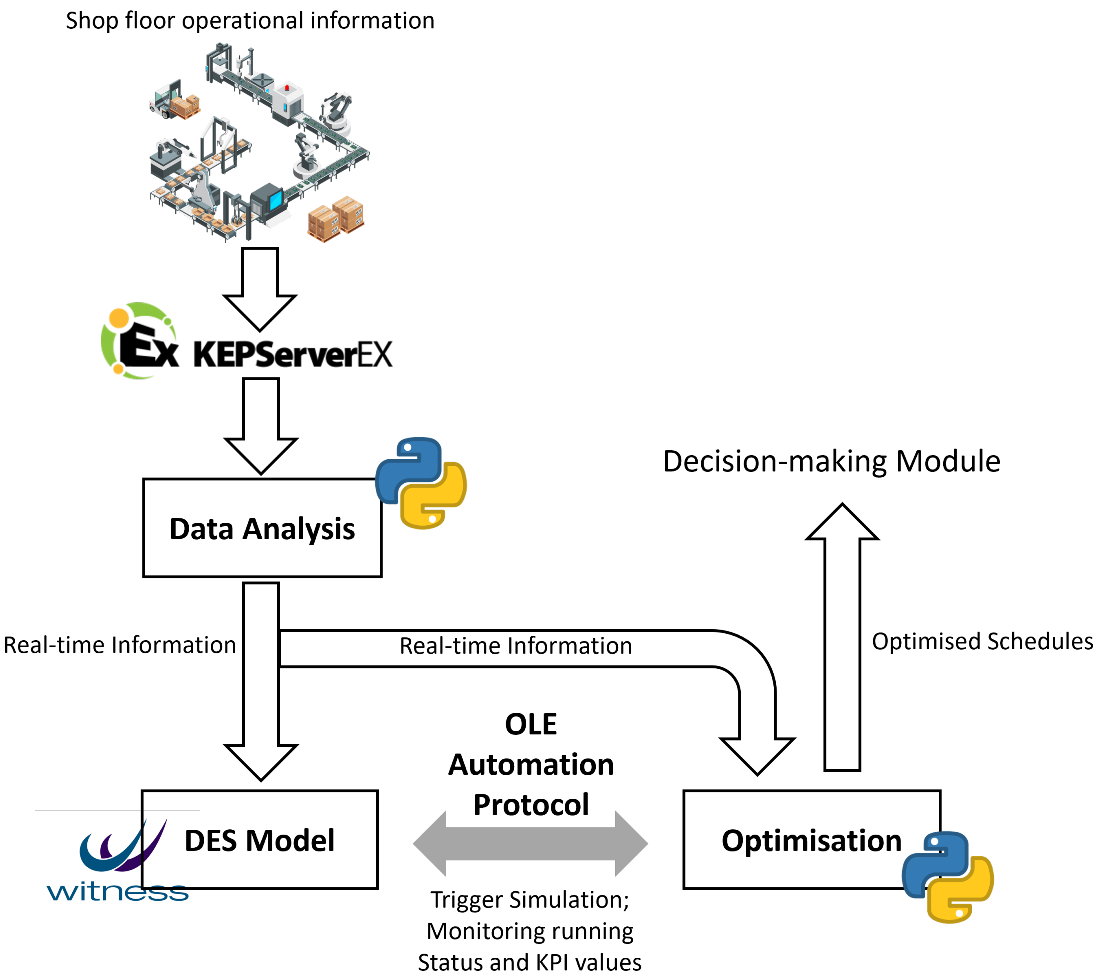

Figure 3.16: The communication architecture between three sections

The DT-based DES model simulates the real-time shop floor production process and provides the predicted production KPIs for the optimisation using OLE automation protocol [216]. The OLE automation server transfers the factory configuration parameters, real-time production process information, predicted KPIs and WCL commands between the process simulation and optimisation. In the optimisation section, the traditional NSGA-II algorithm is modified by integrating with the DT-based process simulation model to provide the optimised scheduling strategies for Decision-making module.

3.5 Decision-making module

The Decision-making module is the decision element of SAMS to analyse the optimised scheduling strategies received from the optimisation module. It combines the system KPIs requirement or operator's advice to generate the final optimal schedule for controlling the AGVs delivery via the fleet manager and feeding the manufacturing process plans into the system management through the OPC UA server. The overview of the decision-making module is shown in Figure 3.17.

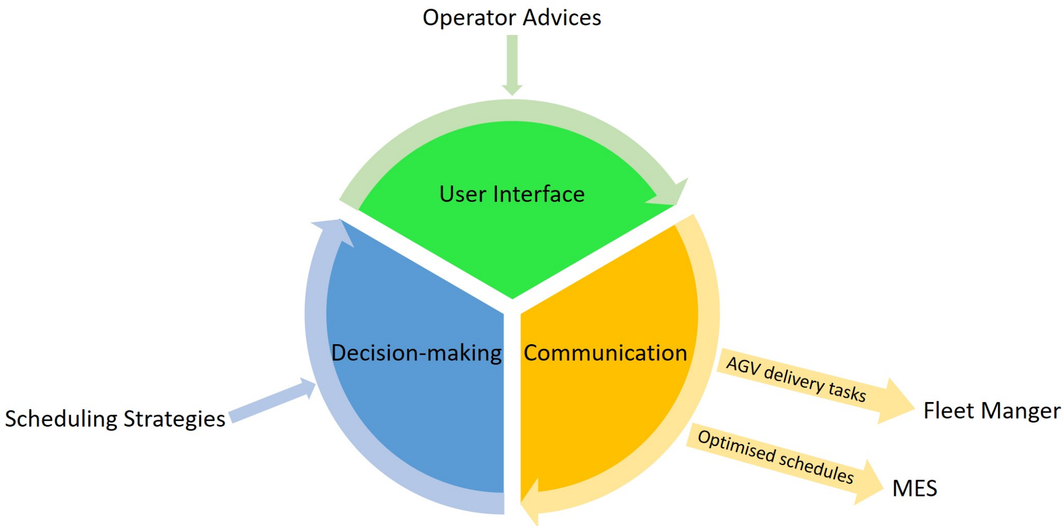

Figure 3.17: The overview of the Decision-making module

As shown in Figure 3.17, the decision-making module consists of three functional parts:

1) User interface creates the interaction between the operator and SAMS to enable access the human advice for improving the manufacturing system performance;

2) Decision-making combines the requirement of KPIs and optimised schedules to generate a suitable solution for AGVs and shop floor manufacturing systems.

3) Communication creates a connection with the fleet manager of AGVs and OPC UA server to assign tasks for shop floor logistics and production processes. The details are explained below session.

1) User interface

Firstly, a dashboard panel is developed, it shows the real-time manufacturing process and scheduling strategies generated from the optimisation module, including: 1) status of AGVs, 2) status of work stations, 3) current KPIs of the overall manufacturing process, 4) the list of suggested AGV scheduling strategies. This information update rate is set up to every 1 second, which can help operators to understand the manufacturing process in real-time.

Secondly, a manual control panel is implemented in the user interface. The two features are achieved in this control panel:

(i) The debug feature. The operator can manually manipulate the AGVs movement and process the delivery tasks, including creating a new task, skipping and deleting existing tasks for debugging the AGV-based logistics or production processes.

(ii) The scheduling suggestion feature. Based on the real-time manufacturing process and the working experiences, the operator can manually select a suitable scheduling solution from a list of scheduling strategies.

The functions of the user interface are shown in Figure 3.18

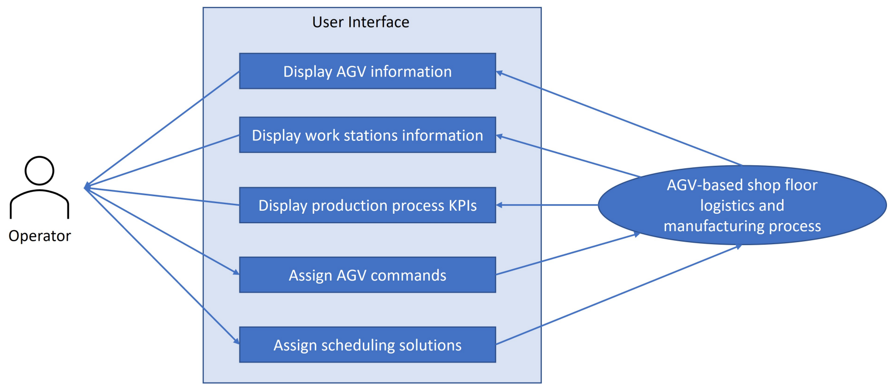

Figure 3.18: Functions of the user interface

2) Decision-making

In the decision-making section, the ruled-based decision engine is created. The rule definition is shown in Table 3.8

Table 3.8: The example of rule definition

Condition	Action	Description
A list of scheduling strategies are generated	Select one optimal solution for AGV systems	One suitable solution for AGVs need to be selected.
User input is detected	Select one solution following the operator instruction	The operator can select a suitable solution based on their experience.
User input timeout	Select one solution based on defined KPIs requirements	The system will select one solution automatically.
One scheduling solution is selected	Send the scheduling to the AGV fleet manager	The scheduling will be encoded to AGV tasks and sent to the fleet manager.

The workflow for the decision-making is shown in Figure 3.19

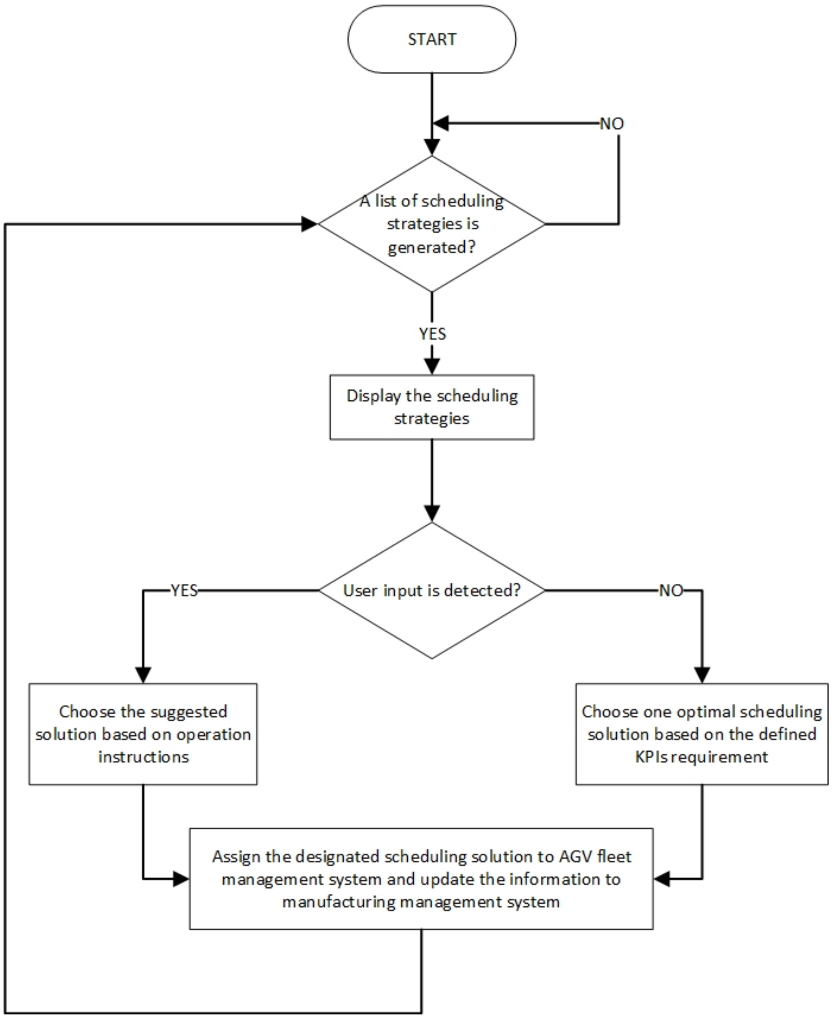

Figure 3.19: The workflow of decision-making

3) Communication

The communication part is implemented to handle the delivery message with the fleet manager of AGVs and update the manufacturing process information with system management through the OPC UA-based data transaction module.

Thus, the two-communication mechanism is applied in this part:

(i) **REST API Client.** The API of the fleet manager contains a large group of API end-points, which is designed by using the HTTP protocol to request the data or post a new mission. Thus, a REST API client is implemented to connect with the fleet manager of AGVs for posting the delivery task at the right time.

(ii) **OPC UA Client.** It enables communications between the decision-making module with OPC UA server. This client publishes the optimised schedules for system management to enable production traceability via the OPC UA-based data transaction module.

3.6 Chapter summary

This chapter presents a methodology to integrate and optimise AGV-based shop floor logistics with the manufacturing process. Various functional modules of SAMS, including the data transaction module, optimisation module and decision-making module, are presented in detail. SAMS is integrated with MES/APS and the fleet manager of AGVs to monitor real-time manufacturing process information and provide scheduling strategies. In the SAMS, the delivery time and charging threshold of AGVs can be optimised and be updated automatically when e.g. the production demand change. The communication among SAMS, shop floor work stations and MES is carried out using the OPC UA server. The interoperability of the OPC UA server allows SAMS to track the real-time manufacturing process information, such as status of work stations, distribution trend of work stations' cycle time, the energy consumption of AGVs, tasks of AGVs, and progress on production.

Furthermore, the hybrid of NASG-II and DT-based optimisation method is elucidated. The DT model duplicates the physical shop floor manufacturing process by updating attributes of model elements, and the NASG-II algorithm provides a multi-objective optimisation capability with an acceptable computation complexity. The presented optimisation module generates a list of AGV scheduling strategies considering the real-time manufacturing process to optimise the shop floor JIT delivery and the overall energy consumption. Especially when the manufacturing system is interrupted, a new group of delivery schedules can be re-generated from the optimisation section. Alternatively, a human operator can choose an optimal scheduling strategy through the build-in user interface and broadcasted system performance dashboard. Once the scheduling scheme is decided, the delivery tasks will be assigned to the fleet manager of AGVs, and the updated schedule will be sent to MES.

Chapter 4

Case Study

4.1 Introduction

In this chapter, the proposed SAMS methodology is validated through a case study. The aim of the proposed methodology focuses on optimising and integrating the AGVs-based shop floor logistics with the manufacturing process. The case study describes the implementation of the proposed methodology into a battery assembly process. Then the scheduling of multiple AGVs-based shop floor logistics, especially the dispatching time and charging threshold of AGVs, is optimised by using the proposed methodology.

4.2 Case study background

This case study was implemented on the Integrated Manufacturing & Logistics (IML) demonstrator built by the Automation Systems Group (ASG) in the International Manufacturing Centre (IMC) engineering hall. It is used for demonstrating a cylindrical-cell-based battery pack assembly process, including packing battery cells into modules, followed by battery pack assembly, spot welding, and inspection. Also, the AGV systems are developed to execute the work-in-progress delivery and collection in this shop floor logistics. Thus, this system has the modularity and reconfigurability to adapt to multiple automation stations for demonstrating Industry 4.0 based technologies, including CPS, DT, robotics, virtual engineering, PLC auto-code generation, AGV-based logistics, and cloud-based engineering services. In this case study, the SAMS software application integrates the AGVs with the automatic battery assembly process and optimises the AGVs delivery schedules.

The overall battery assembly process contains five different types of work stations plus AGV-based shop floor logistics:

1) Launch station. The battery assembly process starts from this station. Operators are instructed via HMI to assemble the trays and update RFID tags in preparation for the battery assembly process. Once a pallet tray is placed on a trolley, it is ready to be delivered to legacy loop through AGV;

2) Legacy loop. A roller-based conveyor system is used to transport the pallet passing through four robot stations. The first two robot stations insert the battery cells for battery modules, after which a pick-and-place unit fits the lids to modules, and then the modules are assembled into battery packs using a collation robot;

3) Welding station. The robot in this station is fitted with a welding tool. Different welding processes are performed depending on the battery pack's RFID tag information;

4) Inspection station. A camera-based inspection head is mounted on this robot to analyse the quality of the battery packs via visualisation. The station is shown in Figure 4.1;

5) Packing station. This stores the finished products;

6) AGV. MiR100-AGV, which supports the shop floor logistics, is shown in Figure 4.1. It provides collection and delivery services for work stations.

The IML deployed various legacy and agile manufacturing systems, for example, conveyor system-based tradition cellular manufacturing [217] and AGVs-based shop floor logistics.

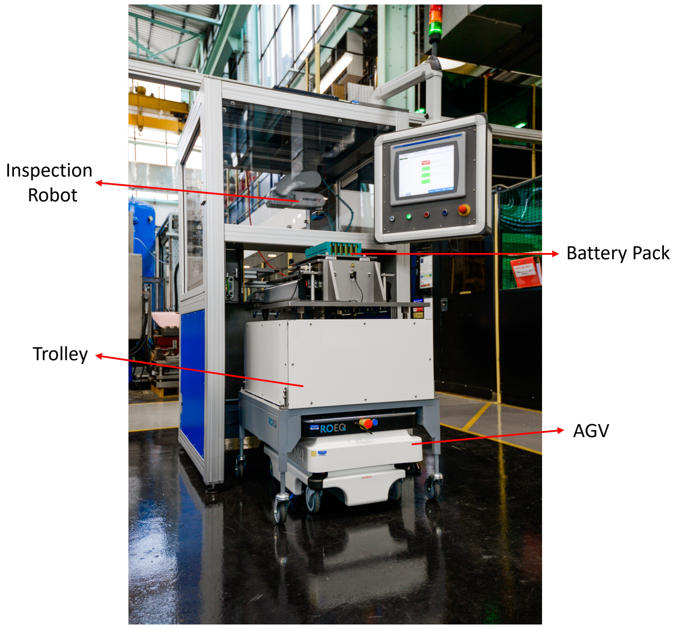

Figure 4.1: The inspection station

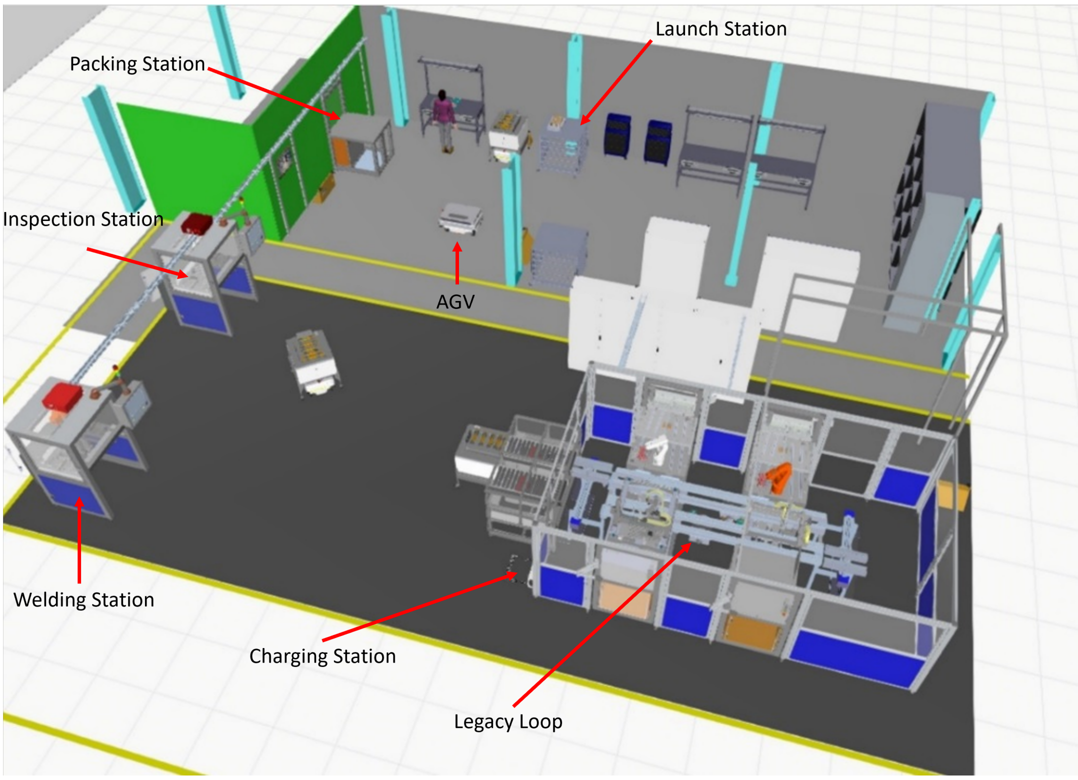

Figure 4.2: The overview of the IML demonstrator

Two types of battery cells (i.e., 18560 and 26650) can be assembled into battery packs during the battery assembly process. Each battery pack includes six battery modules, which are collated by a robot in the legacy loop. Also, MiR100 AGVs are integrated to deliver/collect products between work stations. They are differential-drive autonomous indoor vehicles and have the capability of self-navigation, dynamic route plan and obstacle avoidance via SLAM-based navigation and onboard sensors, such as laser scanners, odometry sensors, Inertial Measurement Unit (IMU) and 3D cameras.

The layout of the IML is shown in Figure 4.2. The automated battery assembly process starts from the Launch Station where the battery brackets are placed via a manual operator, then the brackets are transported to the Legacy Loop where the battery cell is inserted, and the battery module is assembled by robot stations. Once the battery module is assembled, they are transported to the welding station for connecting battery cells and to the inspection station for a final quality check. Finally, the battery pack are transported to Packing Station. The process of the automated battery assembly in the IML has been recorded in the link: https://vimeo.com/387503412?embedded=true&source=vimeo_logo&owner=96799118.

During the battery assembly process, the proposed optimisation method is executed to optimise the overall performance of AGV-based shop floor logistics. Specifically, the dispatch time of AGVs for battery cells and pack delivery/collection, the charging threshold of AGVs are optimised with consideration of the real-time battery assembly process information for improving the JIT performance of shop floor logistics, minimising the overall energy consumption of AGVs and maximise the number of shipped battery packs. The input and output variables of the proposed optimisation module are shown in Figure 4.3.

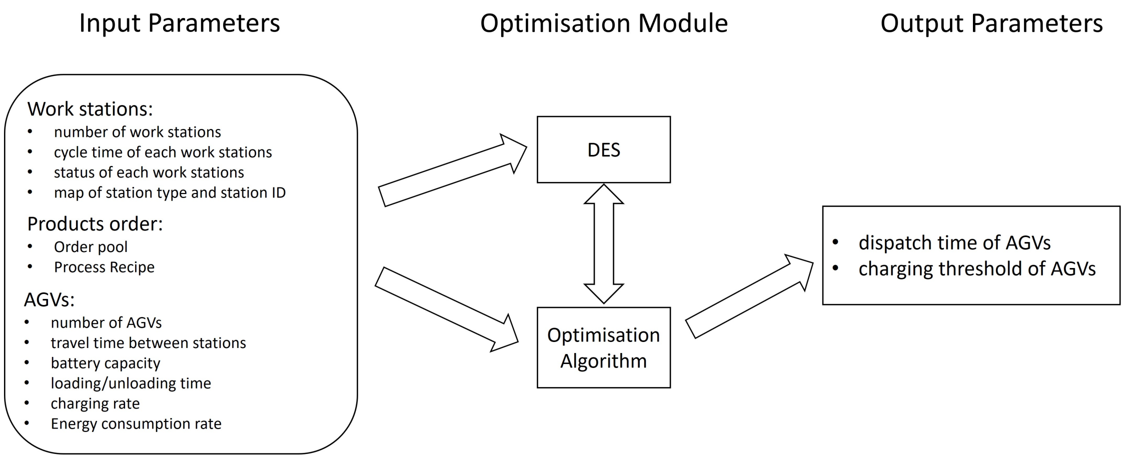

Figure 4.3: Input and output parameters of the optimisation model

4.3 Case study implementation

4.3.1 SAMS software application implementation

The SAMS software application and the database server are deployed on a PC with $Intel^R$ $Core^{TM}$ with 32GB RAM and I9 18-core 3.0 GHz processors. This PC is connected with work stations and AGVs through a WIFI-enabled route. The data sharing between these elements is shown in Figure 4.4.

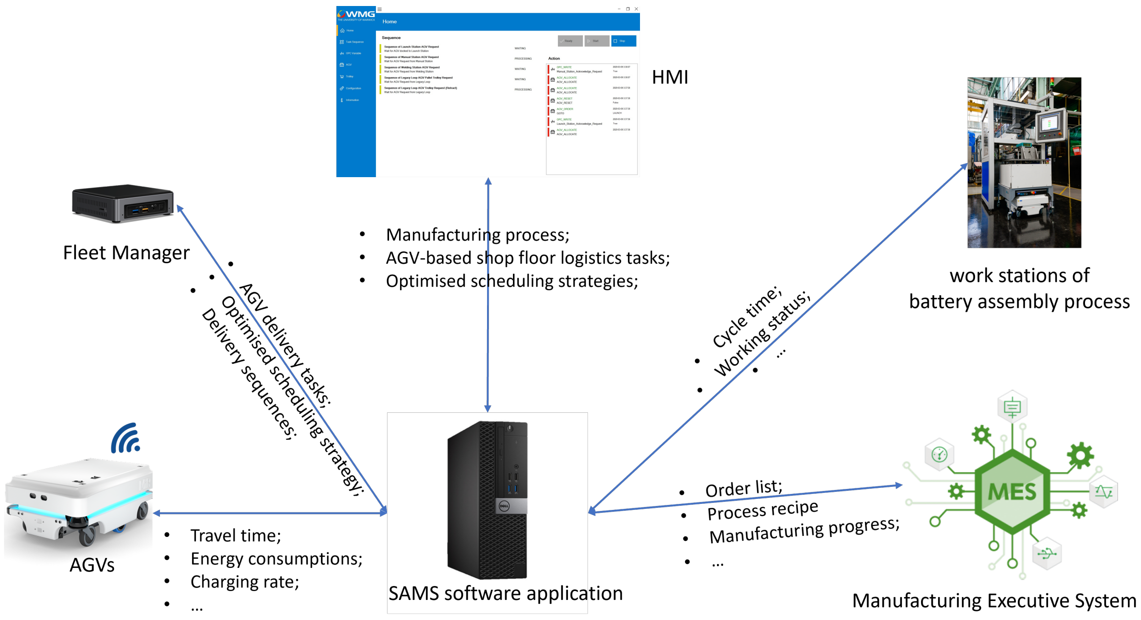

Figure 4.4: The information sharing of this case study

These elements are connected to the same local network, thus, the SAMS is able to subscribe to the real-time shop floor logistics and manufacturing process information and to support the AGVs-based shop floor logistics scheduling when necessary, which is detailed in Table 3.1, 3.2, 3.3. Also, the sequence of the communication between these elements is described in Figure 4.5.

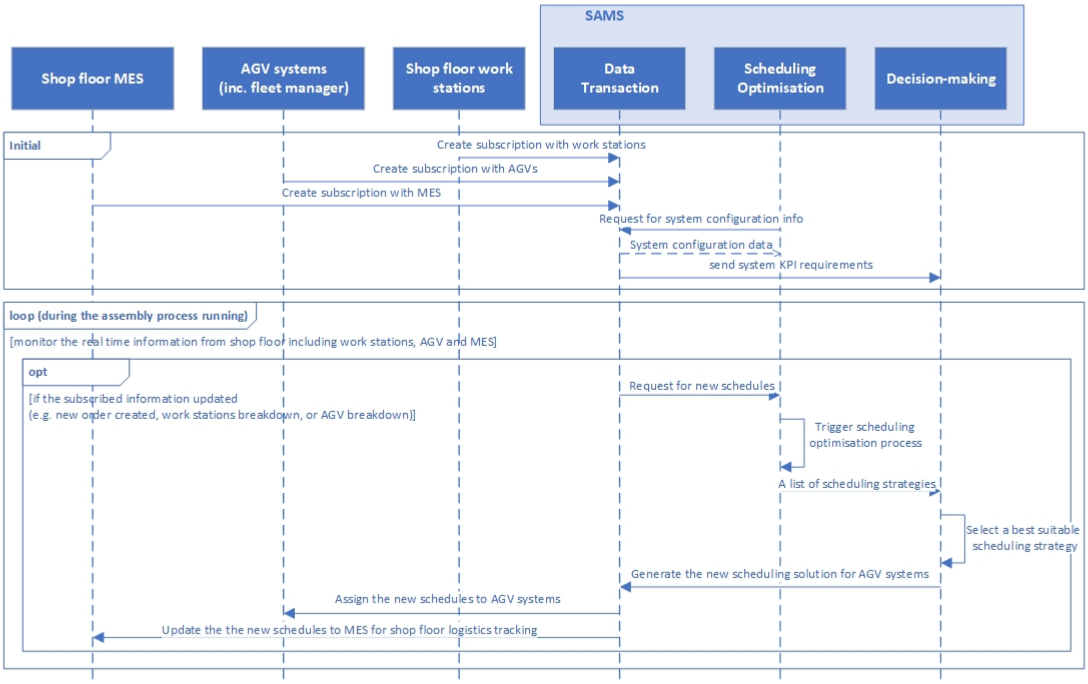

Figure 4.5: The Unified Modelling Language diagram

In this Unified Modelling Language (UML) activity diagram, the communication sequence of SAMS and shop floor automation systems has been illustrated. To monitor the real-time battery assembly process, the subscriptions between SAMS and work stations, AGV systems and shop floor MES are created first. When the abnormality is detected by SAMS, it will trigger the rescheduling process to generate a list of new scheduling strategies for AGV systems.

4.3.2 Database implementation

To store shop floor layouts, pre-defined process recipes and analysed real-time operation, information of work stations, MES and AGV is stored in a SQL server-based database, the database schema is shown in Figure 4.6.

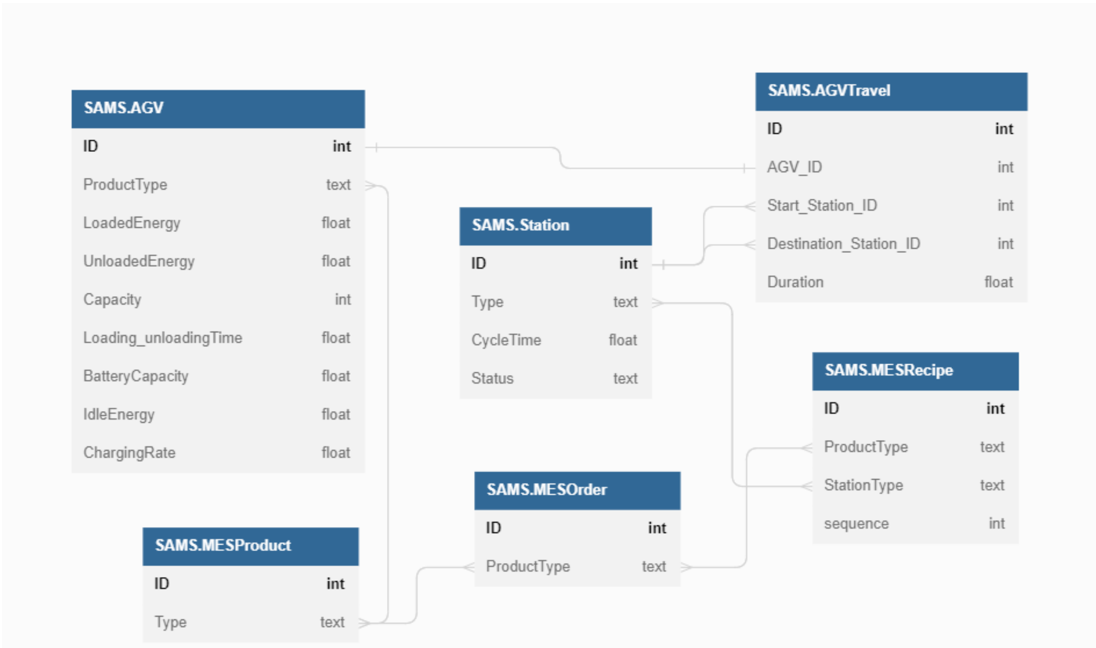

Figure 4.6: The database schema

Based on this schema, the SAMS database is created. Specifically, the cycle time of the work stations and the energy-related information (including charging time, energy consumption rate) of the AGV are used by the data analysis section to analyse the distribution trend. This information will be fed into the DES model to simulate the real-time battery assembly process.

4.3.3 Real-time operational data collection implementation

The real-time operational data is collected through the KEPSeverEX-based OPC UA server. The real-time operation information of the work stations (shown in Figure 4.7 and Table 4.1), including work station status, actuator state, PLC status, safety information, AGV docking information and RFID-based battery pack tracking data, and the AGV information (shown in Figure 4.8), including the location, destination, running status and battery level are subscribed by this server.

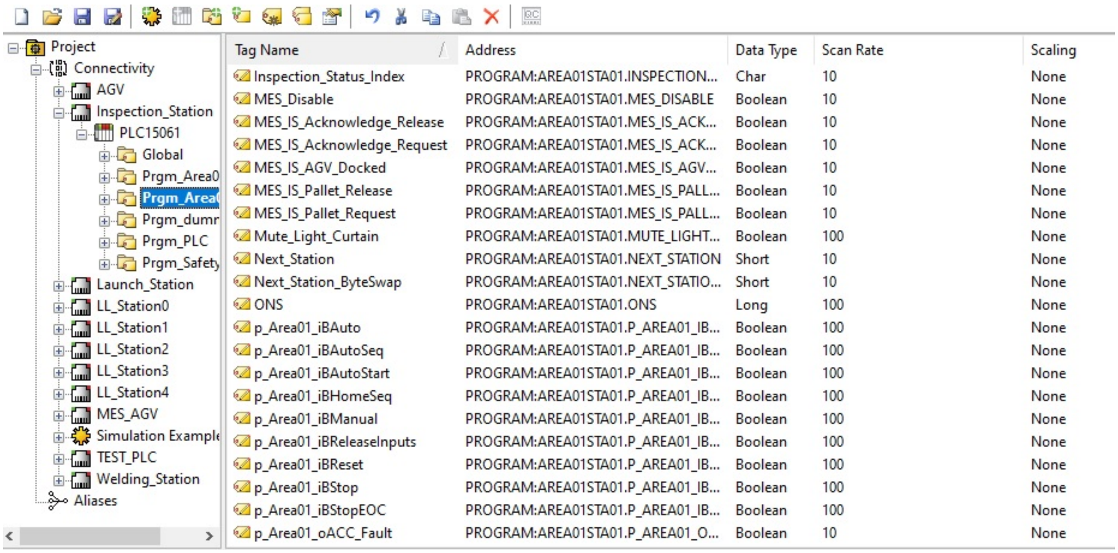

Figure 4.7: The real-time data of work stations

Table 4.1: Key tags information of inspection stations

Tags	Type	Description
MES_IS_Pallet_Request	Bool	The request signal from IS
MES_IS_Pallet_Release	Bool	The release signal from IS
MES_IS_Acknowledge_Request	Bool	Acknowledge signal from MES
MES_IS_Acknowledge_Release	Bool	Acknowledge signal from MES
Trolley_At_Dock	Bool	AGV arrivals signal
Next_Station	Integer	The next station for AGV
Global_Part_Status	Integer	Assembled battery pack status
Inspection_Status_Index	Integer	Inspection Station Status

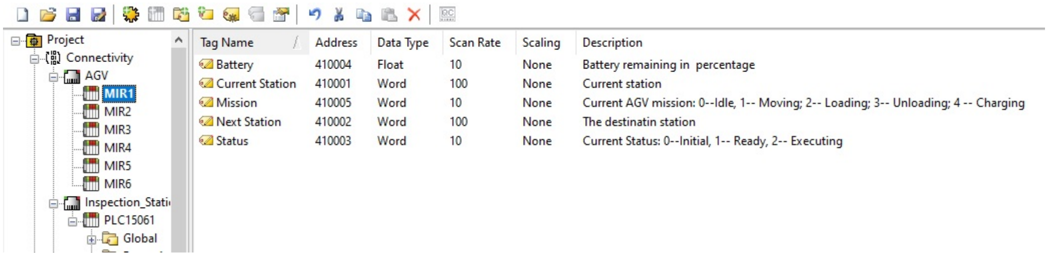

Figure 4.8: The real-time data of AGVs

In Table 4.1, the key tags information for the inspection station is shown. For the other work stations (i.e. launch stations, legacy loop stations, welding stations, packing stations), similar operational information is subscribed. In Figure 4.8, the tags of one AGV are illustrated, and the same operational information from other AGVs are considered. For this real-time information from the work stations and AGVs, the cycle time of stations, work status of stations, travel time of AGVs, status of AGVs, energy consumption rate and charging rate of AGVs can be analysed for simulating the real-time battery assembly process.

4.3.4 DES model implementation

In this case study, the WITNESS Horizon software [218] is used to duplicate the physical shop floor manufacturing process and AGV-based shop floor logistics for generating predicted production KPIs for the optimisation section. The WITNESS Horizon supports the process modelling, analysis and optimisation, and can help the engineering cost-effectively pre-plan the shop floor layout [219]. The WITNESS Horizon provides a user-friendly interface in which the operator can customise the manufacturing environment and production process efficiently. Also, because of the WCL [220] protocol, the system database, real-time data server and optimisation software can interact with WITNESS Horizon software for better decision-making in the shop floor planning and real-time logistics scheduling.

To build the DES model of the battery assembly process, the following assumptions are considered:

1) The shop floor layout and the work stations' locations are fixed, which means route dynamic

planning is not considered.

2) An AGV can only perform one delivery task at one time.

3) In this case study, the battery assembly process time was set to one single shift which is 8 hours of continuous operation.

4) In this case study, the two levels of charging threshold were specified for the AGVs. If the battery percentage of AGV is lower than the alert threshold, it can carry on with a new task or drive to the charging station depending on the station occupancy and demand list. However, if the battery percentage is lower than the alarm threshold, the AGV is required to drive to the charging station and cannot be assigned a new task.

5) For the energy consumption of AGVs, based on the current electrical price in west midlands, per KWH electricity costs 19.6p.

$$The\ cost\ of\ energy = energy\ consumption\ of\ AGV * 0.196 \tag{4.1}$$

6) Both earliness and lateness of battery delivery could cause the energy waste of the work stations and AGVs. Thus, the penalty of earliness and lateness (referring to equation 3.1) can be set by considering the energy waste of work stations and AGVs. In this case study, the work stations' power is 4.4KW (220V and 20A) and AGVs' power is 0.36KW (24V and 15A). Thus, the cost of JIT is calculated:

$$The\ cost\ of\ JIT = earliness\ time * 0.36 + lateness\ time * 4.4 \tag{4.2}$$

7) The cycle time of work station, the parameters of AGVs, including the travel time between stations, energy consumption rate, charging rate, and unloading/loading time are predicted through the combination of historical data in the database and the real-time information from the OPC UA server.

8) This case study contains 4 launch stations, 4 battery cell buffers, 4 legacy loops, 4 welding stations, 4 inspection stations, 4 packing stations, 6 AGVs and 4 charging stations. Please

note, the cycle time of legacy loops is two to three times as long as other work stations, which could cause an unbalanced workload at work stations. In this case study, the objectives are focused on minimising the cost of JIT, energy consumption and maximising shipped battery packs. Nevertheless, the overall workload balance of work stations will be considered in further research.

The WITNESS Horizon-based DES model of the battery assembly process is shown in Figure 4.9. Additionally, the KEPServerEX scanning rates are set up as 10ms for subscribing to the updates of physical systems. Thus, it can inform the operational information changes to the DES model for updating timely. In the DES model, the priority of the delivery task is designed based on the recipe of the battery assembly process, which means the priority sequence of this case study is packing station, inspection station, welding station, legacy loop, and launch stations. For example, when the packing station and inspection station request one AGV at the same, this AGV will be assigned to travel to the packing station firstly. It is notable that the simulation process will terminate if the decision variables do not fit with the simulation during the optimisation process, and the production KPIs and these decision variables will be neglected.

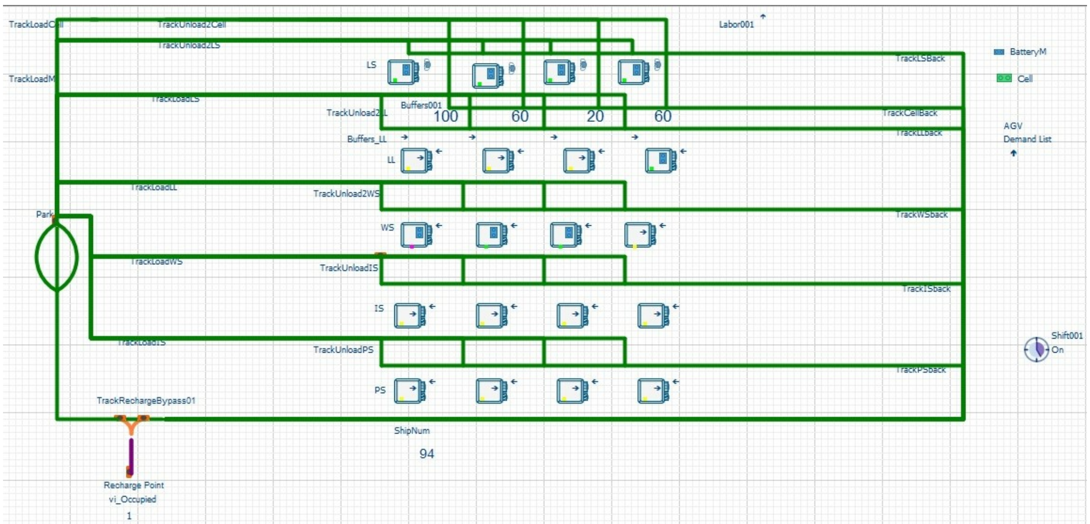

Figure 4.9: The WITNESS model of battery pack assembly process

In this figure, five stages of work stations are designed, including four Launch Stations (LS), four Legacy Loops (LL), four Welding Stations (WS), four Inspection Stations (IS) and four Packing Stations (PS). They are shown as blue blocks in the model. The AGVs are drawn in orange/black blocks and the green tracks are their travelling path. The “AGV demand list” which is on the right side of the model means the real-time tasks for AGVs.

4.3.5 Optimisation model implementation

In the optimisation, the hybrid NSGA-II combines the NSGA-II algorithm and the DES-based process simulation. The hybrid NSGA-II is run with the physical battery assembly process concurrently to find the near-optimal AGV dispatch time for each work station. According to Figure 4.9 and the layout of the shop floor, the configuration of the optimisation problem consists of:

1) The number of decision variables: 22 variables are assigned, including different AGV dispatch time for 16 work stations and 4 battery cell buffers, 1 AGV charging alert threshold and 1 AGV charging alarm threshold.

2) The number of objectives: 3 objectives, including the number of the shipped battery packs, the overall energy consumption of AGV (Equation 4.1)s and the cost of JIT (Equation 4.2) are considered.

3) The number of constraints: 1 condition – the charging alert threshold needs to be constrained, and it should be larger or equal to the charging alarm threshold.

4) The type of decision variables: The decision variables are separated into 2 groups. The AGV dispatching time for each battery cell buffer is in an integer type, while the AGV dispatching time for each work station are in a float type. The charging thresholds are the percentage of the remaining battery in the AGV, they are in a float type.

5) The objective function: It is a function to control the DES model and collect the production KPIs from simulation. In this function, the decision variables are transferred to the DES model as model elements inputs.

6) The optimisation direction: maximise or minimise the objective functions.

7) The stopping criteria: the optimisation process terminated when it reaches the maximum

number of generations which is set to 100 in this case study. A definition of the optimisation model by using Python is shown in Figure 4.10.

```
###### Set up optimisation problem model############################
problem = Problem(22, 3, 1) ###22 decision variables, 3 objectives, 1 constraints

problem.types[0:4] = Integer(40,80) ### Integer variables: AGV calling time of battery cell buffer

problem.types[4:20] = Real(0.5,0.8)### Float variables: AGV calling time of automation stations
problem.types[20:22] = Real(0.2,0.8) ### Float variables: Charging threshold of AGVs

problem.constraints[:] = ">=0"  ### Constraints function are larger/equal than 0
problem.function = Fitness_Function ### Set up Objective Function:Process Simulation (DES)
problem.directions[0] = Problem.MINIMIZE ### objective are being minmised
problem.directions[1] = Problem.MINIMIZE ### objective are being minmised
roblem.directions[2] = Problem.MAXIMIZE ### objective are being maxmised

##### Start the NSGAII ########################
algorithm = NSGAII(problem)
start_time = time.time()
print("Optimisation Start....")
algorithm.run(100)
end_time = time.time()
print("time_elapsed: " + str(end_time - start_time))
```

Figure 4.10: The configuration of the optimisation problem

To define the objective function of the NSGA-II, the "Fitness_Function" is created to set up the parameters in the DES model, trigger the simulation and subscribe to the prediction KPIs results. Firstly, the WITNESS client is defined to build the connection with the WITNESS model, and the SQL client is created to enable data access from the SQL database. The codes are shown in Figure 4.11.

```
### Inital WINTESS client ##############################
wb1 = win32com.client.GetObject(Class="WITNESS.wcl")
wb1.Begin()


sql = SQLClient.SQLClient()
Init_Machine(wb1, sql)
```

Figure 4.11: The initialisation of the objective function

Then, the work station information including cycle time, working status and the number of work stations are initialised through the data from the database and fed into the WITNESS model via its client. The part of codes of work station information initialisation are shown in Figure 4.12.

```python
def Init_Machine(wb1, sql):
    ##########Setup Configuration Data####################
    cycle_time = sql.machine_read()
    ############Working Station Cycle time Initial##############
    LS_cycle = cycle_time('Launch')
    LL_cycle = cycle_time('Legacy')
    IS_cycle = cycle_time('Inspection')
    WS_cycle = cycle_time('Welding')
    PS_cycle = cycle_time('Packaging')

    for i in range(1,machine_num + 1):
        wb1.Action("LS_Cycle(1,"+ str(i) + ") = " + str(LS_cycle[i]))
        wb1.Action("LL_Cycle(2,"+ str(i) + ") = " + str(LL_cycle[i]))
        wb1.Action("IS_Cycle(1,"+ str(i) + ") = " + str(IS_cycle[i]))
        wb1.Action("WS_Cycle(2,"+ str(i) + ") = " + str(WS_cycle[i]))
        wb1.Action("PS_Cycle(2,"+ str(i) + ") = " + str(PS_cycle[i]))
```

Figure 4.12: The initialisation of work stations information for the WITNESS model

After the work station information is configurated, the optimisation parameter including the dispatching time of AGV for each station and the charging threshold will be updated into the WITNESS model. The codes of the setup for the optimisation parameter are shown in Figure 4.13.

```
############Setup Optimisation Parameter######################

#1. Setup AGV dispatching time########################
for i in range(1,machine_num):
    variable = "Cell_Call(" + str(i) + ") = "
    wb1.Action(variable + str(Cell_calling[i-1]))
    variableA = "LS_Call(" + str(i) + ") = "
    wb1.Action(variableA + str(LS_call[i-1]))
    variableB = "LL_Call(" + str(i) + ") = "
    wb1.Action(variableB + str(LL_call[i-1]))
    variableC = "WS_Call(" + str(i) + ") = "
    wb1.Action(variableC + str(WS_call[i-1]))
    variableD = "IS_Call(" + str(i) + ") = "
    wb1.Action(variableD + str(IS_call[i-1]))

# #2. Setup Charging threshold of AGVs####################

wb1.Action("Charging_threshold(1) = " + str(charging_threshold[0]))
wb1.Action("Charging_threshold(2) = " + str(charging_threshold[1]))
```

Figure 4.13: The setup of optimisation parameters for the WITNESS model

Once the pre-setting step is completed, the WITNESS model will be triggered to start the simulation. Meanwhile, the simulation status will be monitored to figure out when the simulation is completed. Please note, in this study, if the simulation time is longer than 5 seconds, the simulation will be terminated. The simulation usually takes 3 seconds, and the simulation time could be longer than normal, due to the wrong combination of optimisation parameters. The codes of this part are shown in Figure 4.14.

```
### Starting the witness simulation###################
start_time = time. time()
wb1.Batch()
time.sleep(1)
sim_status = wb1.ModelStatus

### Check the simulation termination condition: ###############
### (simulation completed) or (time elapses long than 5s)######
while (sim_status == 1):
    sim_status = wb1.ModelStatus
    end_time = time. time()
    time_elapsed = (end_time - start_time)
    if time_elapsed > 5:
        print("Simulation Fault...")
        return [999,999,0], [-1]
    continue

print("Simulation Done...")
```

Figure 4.14: The codes for starting and monitoring the simulations

When the simulation is completed, the predicted KPIs, including the cost of JIT, the energy consumption of AGVs and the number of shipped products, will be collected and returned to the NSGA-II optimisation function block. Also, the relationship between the charging alert threshold and the charging alarm threshold will be returned as a second parameter for comparison with constraint conditions. The data analytics after the simulation is shown in Figure 4.15.

```
########### data analytics after simulation#################
late = wb1.variable("Lateness")
early = wb1.variable("Earliness")
energy = wb1.variable("Energy_cost")
ship = wb1.variable("Ship_Num")

##############Caculate JIT cost#############################
a = 0.24
b = 3.3
JIT = (a * early + b * late)
##############################################################

return [JIT, energy, ship], [charging_threshold[0] - charging_threshold[1]]
```

Figure 4.15: The data analytics after the simulation

Specifically, the simulation of the automated battery assembly process can be impacted by the information of work stations, AGVs, and the combination of the dispatching time of AGVs for each station. Therefore, it is important to create an error handler to deal with the abnormality in the objective function. The codes are shown in Figure 4.16.

```
except Exception as ex:
    print("ERROR")
    print(ex)
    time.sleep(1)
    wb1.Begin()
    return [999,999,0], [-1]
```

Figure 4.16: The error handler

In this error handler, the objective function will return the infinity value (Note: it is a large number, 999, in the code) for the cost of JIT and energy consumption of AGVs, and 0 for the shipped battery pack. And -1 will be returned for comparison with the constraint condition. Because in the definition of the optimisation problem, the constraints are defined to be greater than 0. Thus, this group of results will be neglected when choosing feasible solutions.

When the integration of the proposed method and physical process is completed, the optimisation module can update the real-time shop floor information into the DES model, and start searching the near-optimal AGV delivery schedules by cross-checking with the predicted production KPIs. As soon as the delivery schedule and AGV charging plans are decided, they are transferred to the MES and fleet manager of AGVs. As a result, a closed-loop-based optimisation system is built to minimise the logistics cost of JIT, the overall energy consumption of AGVs and maximise the number of shipped battery packs. The optimisation working flow of the proposed methodology is summarised in Figure 4.17.

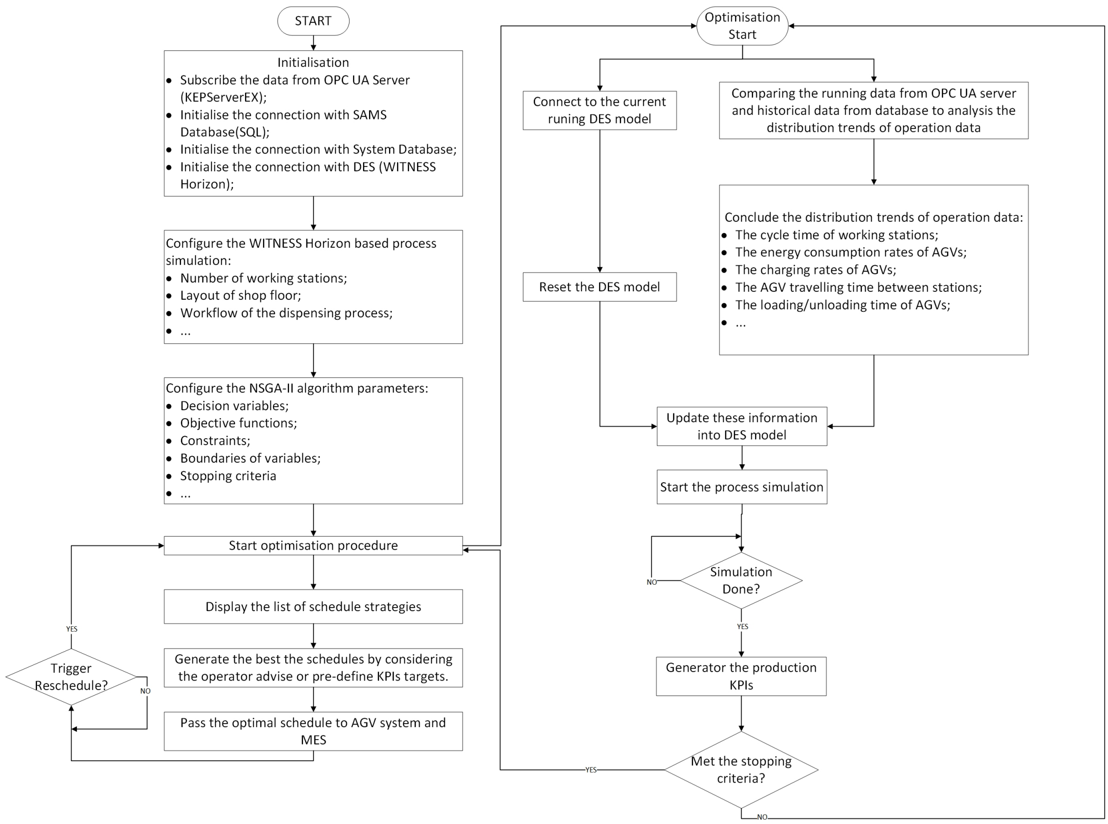

Figure 4.17: The working flow of the proposed methodology

4.3.6 Decision-making implementation

The battery assembly process contains five types of standalone work stations, and their requests are analysed and responded simultaneously. The overall process recipe, shown in Figure 4.18, is complex and challenging to follow without the proposed methodology. However, in this case study, the proposed method with interoperability and real-time decision-making capability plays a significant role in supporting the integration of AGV systems with the battery assembly process.

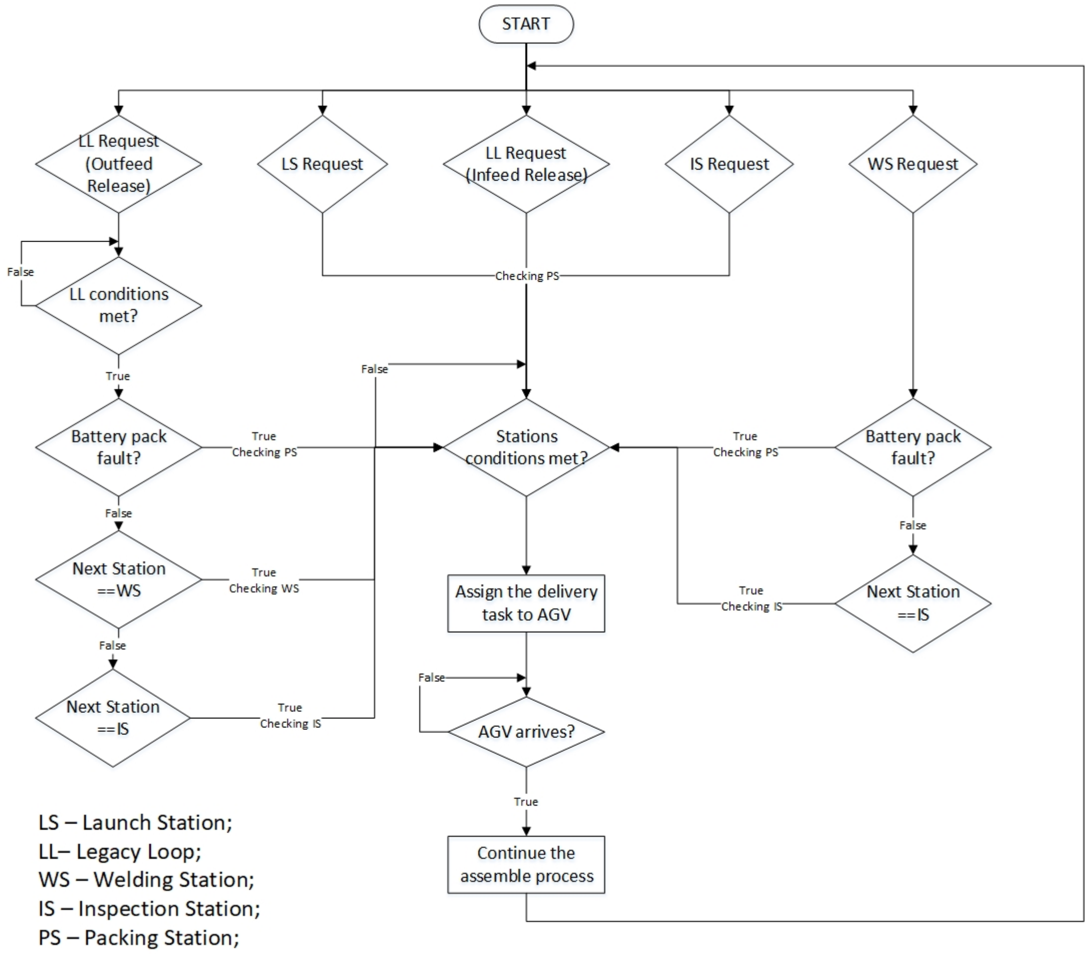

Figure 4.18: The workflow of the battery assembly process

To follow this battery assembly process recipe, shown in Figure 4.18, the JSON format-based logic rules are defined, shown in Figure 4.19.

Figure 4.19: JSON format-based logic rules

4.3.7 User interface implementation

The user interface is a part of the decision-making module displayed on a monitor which is connected to the SAMS software application-embedded PC. Through this user interface, the operator can monitor the real-time battery assembly process and shop floor logistics status through the “OPC Variable”, shown in Figure 4.20. The left side shows the information of the stations and AGVs. The right-side circles mean the current state of its linked information. Additionally. the red circle is false, and the green circle is true. For example, the “Legacy Loop Pallet Trolley Request” is a battery trolley request signal from Legacy Loop, and the red circle shows the current state is false, which means the Legacy Loop does not request the battery trolley.

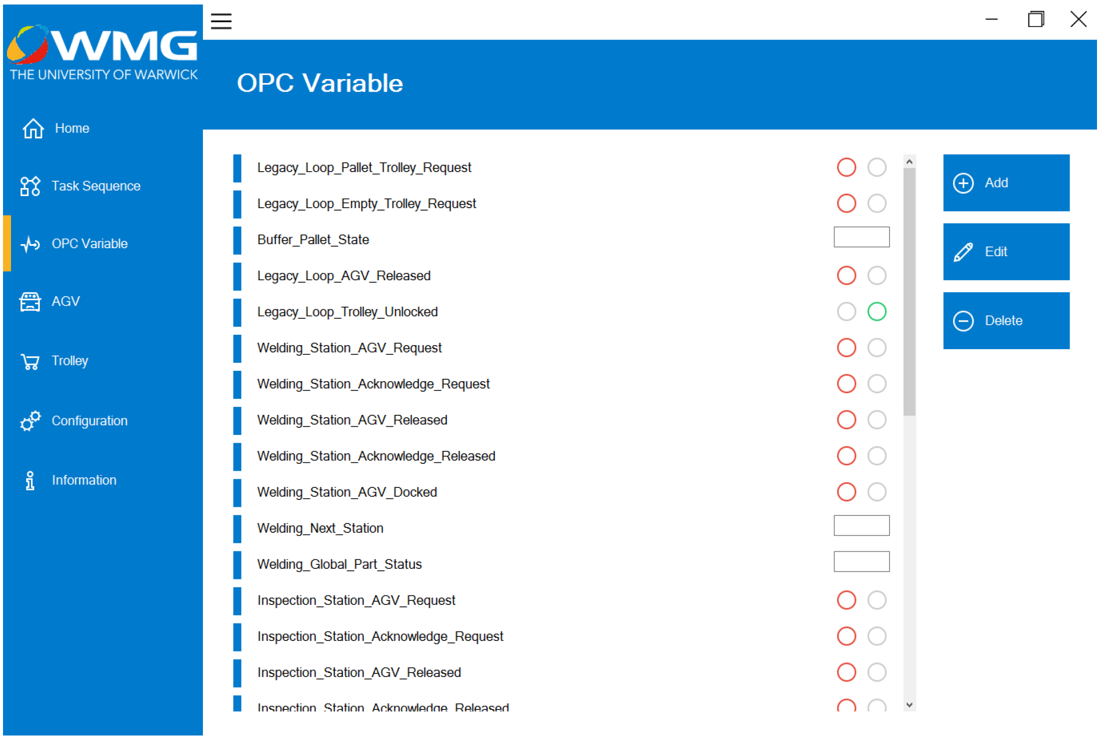

Figure 4.20: The battery assembly process monitoring panel of the user interface

Furthermore, the operator can manually control the AGVs for training or debugging the AGV-based shop floor delivery service via the "AGV" section, shown in Figure 4.21. In the "AGV Manual Control" session, the name of the AGV, actions, and destinations can be selected separately to manually operate the AGVs. Also, the current state of AGV, including current position, destination, allocated status, loading status, availability and in-working status, can be monitored in this section.

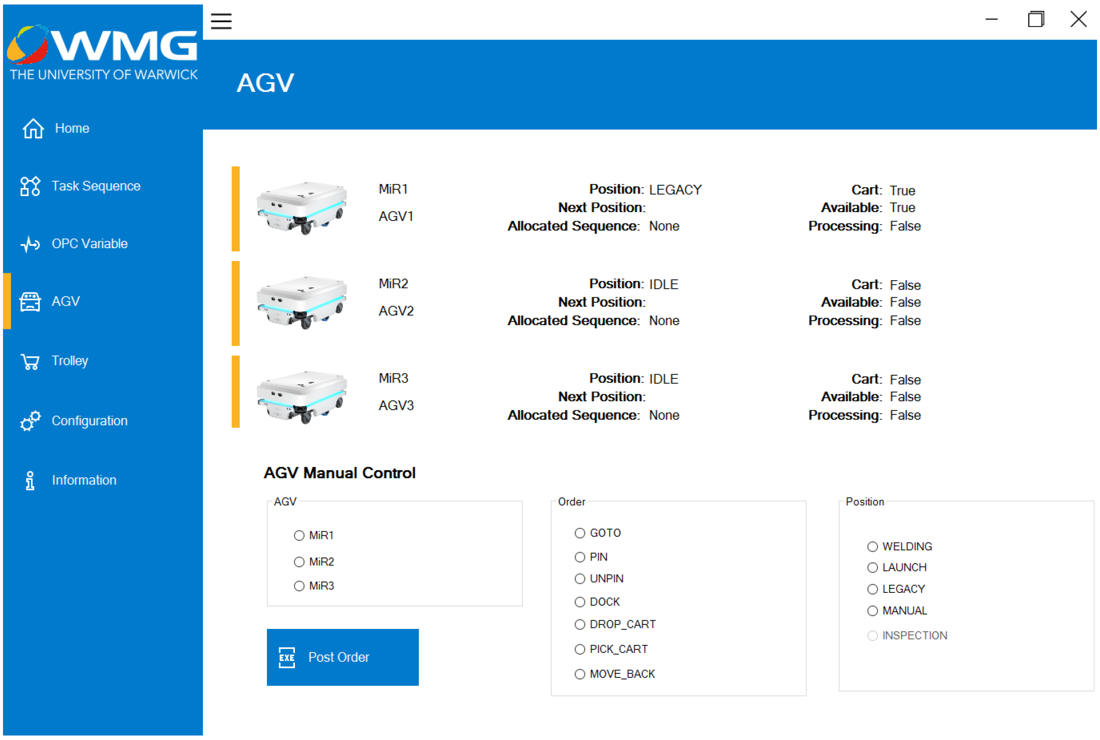

Figure 4.21: The manual control panel for shop floor logistics

4.4 Case study results

4.4.1 Pre-phasing data analysis result

In the data analysis section, the real-time operational information of each work station, including its PLCs, buffer sensors, RFIDs and actuators, are subscribed and analysed to understand its work status and cycle time. For example, as shown in Figure 4.22, the real-time operational data of the inspection station are subscribed. The data analysis section calculates its cycle time and stores this in the database.

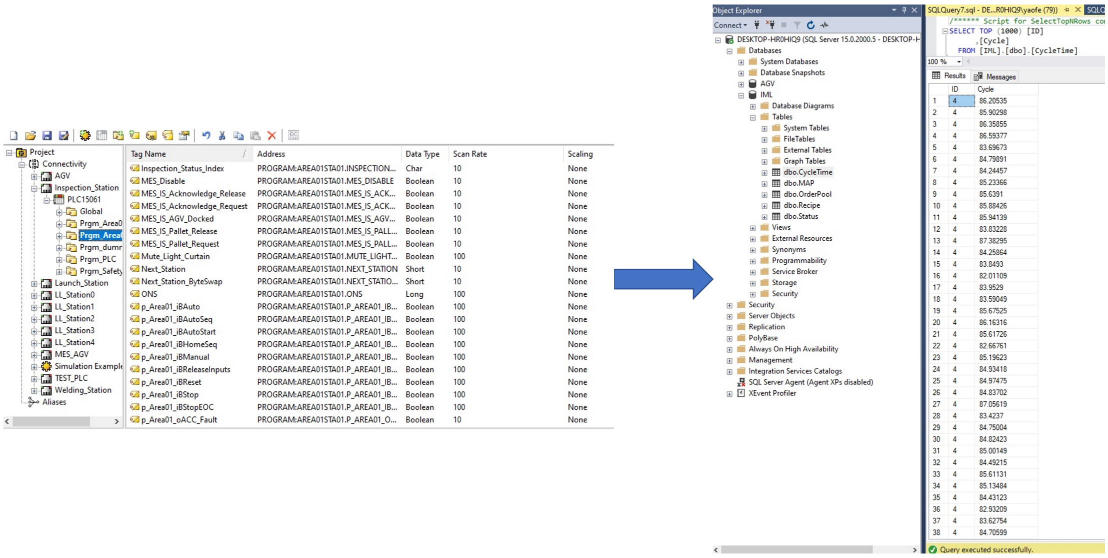

Figure 4.22: The process of analysing the cycle time of the inspection station

In Figure 4.23, the histograms of the cycle time for the inspection station and welding station is shown. A normal distribution is concluded to represent the cycle time of these stations based on the historical data from the database:

$$CT \sim \mathcal{N}(\mu, \sigma^2) \tag{4.3}$$

Where the μ is the average of cycle time (CT), and the σ means the standard deviation of these cycle times. The μ and σ are different in the different distribution trends. In this case study, these parameters are analysed for every work stations, and they are fed into the DES model. The database includes more than 5000 sampling data for each work station.

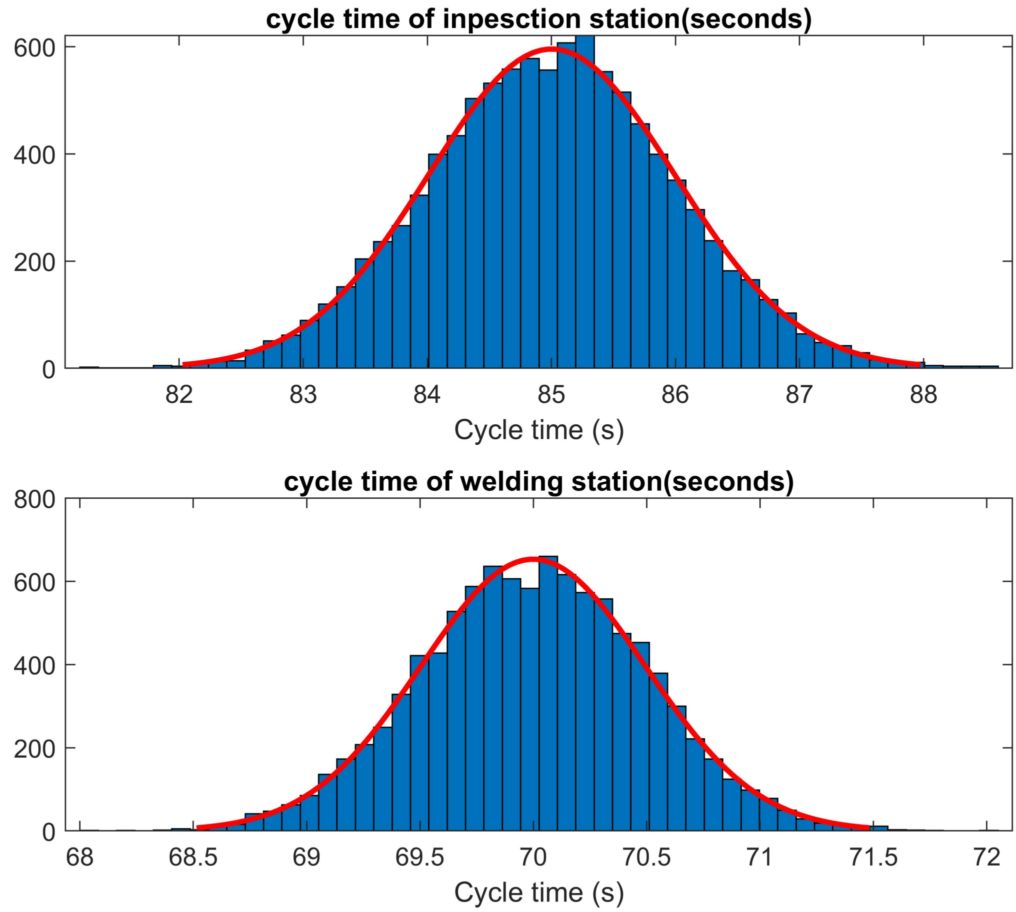

Figure 4.23: The distribution trend of cycle time (inspection station and welding stations)

4.4.2 Real-time information battery assembly process

To visualise the real-time battery assembly process, including the working status of stations and the state of AGVs, a user interface is designed, shown in Figure 4.24. It helps operators to track the sequence of the battery assembly process. Also, the operational information, including the tags of OPC UA, availability of AGVs, and the steps of battery assembly, is shown in this interface for process monitoring.

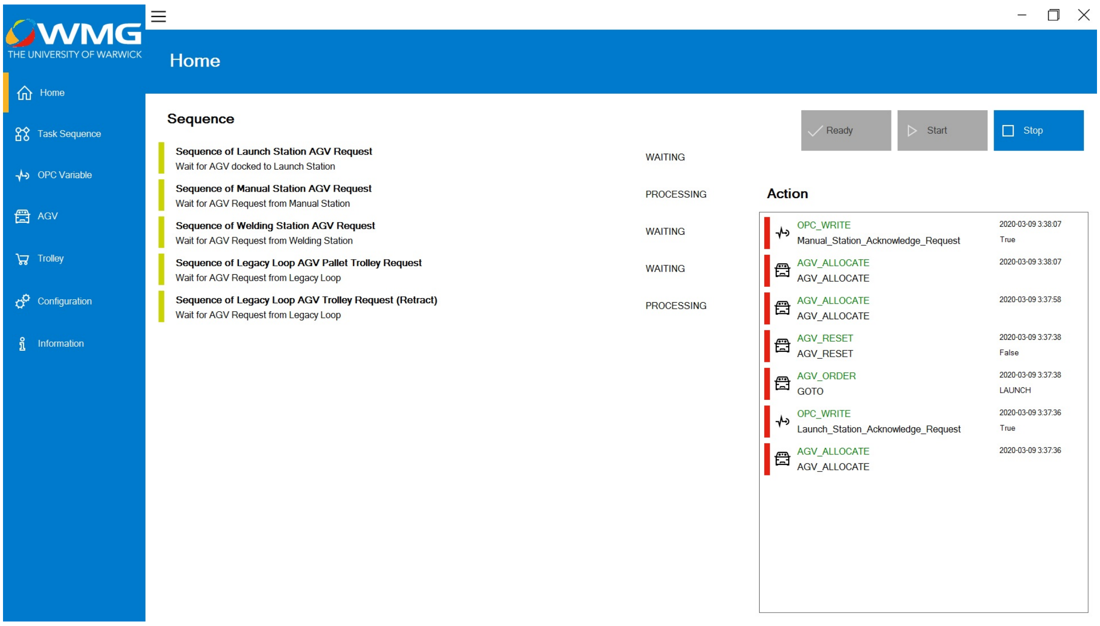

Figure 4.24: The real-time process of the battery assembly

4.4.3 Optimisation results

Based on the design of the optimisation section, the optimisation algorithm converges after around 100 times simulation runs. Thus, the AGV scheduling optimisation process takes 5 $\pm$ 1 mins. According to the assumption in section 4.3.4, the overall battery assembly process is set as 8 hours based on the time scale of the DES model. The results of optimisation are displayed in Figure 4.25. To determine the relationship among the three objectives, one example of the relationship between the cost of JIT and the overall energy cost of AGVs is shown in Figure 4.26.

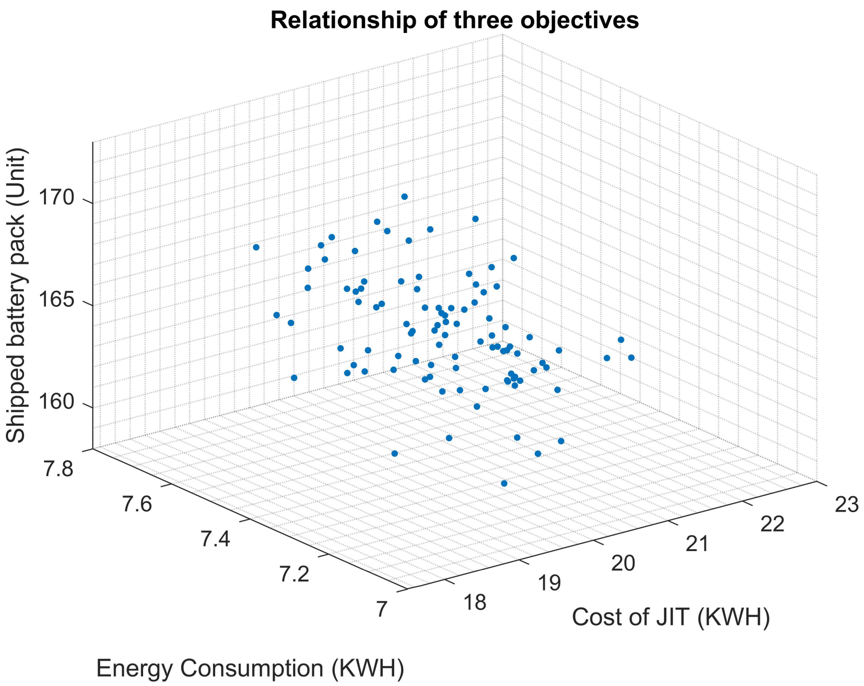

Figure 4.25: The optimisation results of three objectives

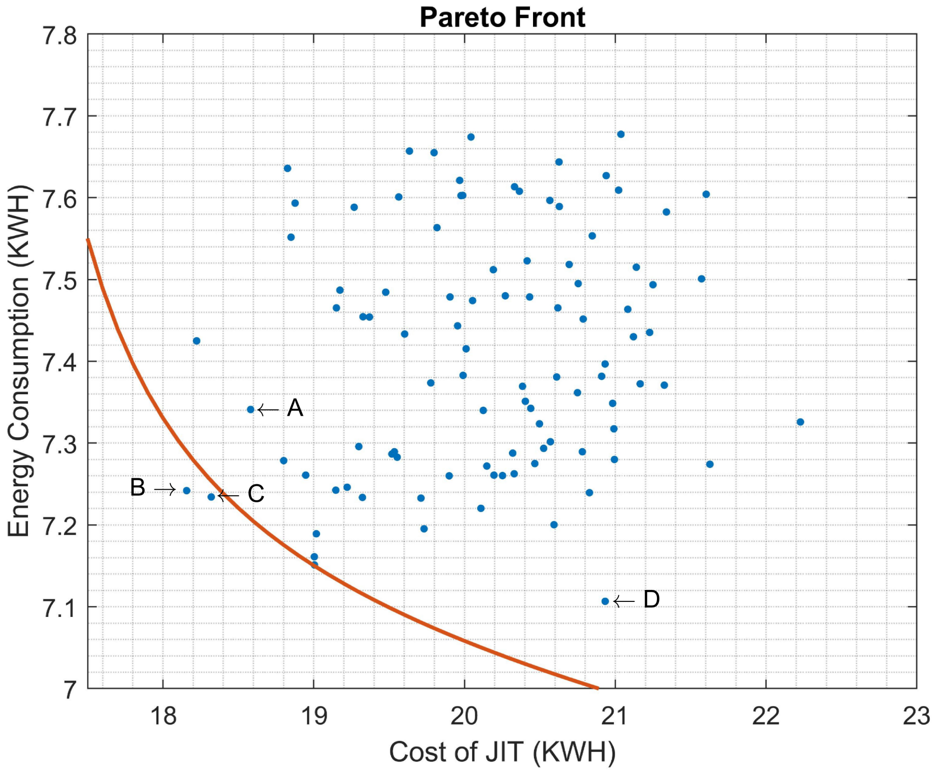

Figure 4.26: The Pareto Fronts

In this Pareto Front figure, each blue dot represents a feasible schedule strategy, and the red line means an example of a Pareto-efficient frontier. For example, Point A is not on the Pareto frontier, because it is dominated by Point B and C. Therefore, considering the minimal energy consumption and cost of JIT, the strategies of Points B and C are better than those of Point A. After the optimisation of the scheduling optimisation module, this group of feasible solutions will be sent to the decision-making module. Through this module, the best suitable scheduling strategy can be chosen for the shop floor logistics based on the operator's advice or the pre-defined system KPI requirements. For instance, the operator considers the minimisation of the AGV energy consumption, so the strategy of Point D can be selected for scheduling the AGVs delivery tasks. This Pareto Front only shows the relationship between the cost of JIT and the energy consumption of AGV. If the shipped number of the battery pack needs to be considered, the new Pareto Front can be extracted from Figure 4.25.

4.5 Chapter summary

In this case study, a five-stage work stations-based battery assembly scenario is presented. The proposed methodology is implemented through five processes, including creating a database, building up an OPC UA server, designing a DES model, developing an optimisation model and structuring the rules-based decision. The results show that the proposed methodology is able to analyse the real-time operational information of work station and AGV and provide them to the DES model for process simulation. Moreover, The SAMS software application is able to visualise the real-time battery assembly process for operators and it can optimise the delivery scheduling and charging threshold for AGVs with consideration of multiple objectives.

Chapter 5

Evaluation of the Proposed Methodology

5.1 Introduction

In this chapter, the proposed SAMS methodology is evaluated through comparisons of its integration capability with the traditional integration method and its optimisation capability under different system disturbances. The case study chapter has validated that the proposed methodology is able to support the integration and optimisation of AGV-based shop floor logistics in a battery assembly process. This chapter firstly compares the integration capability of the proposed methodology with the traditional integration method. Then, this chapter evaluates the optimisation performance of the proposed methodology under three different system disturbances. The results show the proposed methodology is able to optimise the dispatching time and the charging threshold of AGVs to improve the overall performance of manufacturing systems.

5.2 Evaluation of the integration capability of the proposed methodology

In the case study, demonstrated in Section 4.3.3, the KEPSeverEX-based OPC UA server is applied to integrate the AGV-based shop floor logistics and manufacturing processes. Through this server, the SAMS software application is able to monitor the real-time operational data of the overall shop floor.

With the traditional integration method, to understand the operation information of work stations, a connection gateway to access the data from their PLCs, sensors and actuators is necessary. This case study contains five different types of automation stations which are controlled by PLCs and robotics, including Siemens, Schneider, Rockwell, SMC robotics and Festo gantry systems. Thus, to build this connection gateway, different communication protocols, including ProfiNet, EtherNet/IP and Modbus, need to be considered. It would require developing a gateway for integrating the battery assembly system with AGVs. However, because of the capability of the proposed methodology, the steps for system integration and configuration have been significantly reduced. Operators only need basic OPC UA knowledge to link the work stations information with the OPC UA server through tags. Comparison details of the steps and skills requirements are shown in Tables 5.1 and 5.2.

Table 5.1: The steps and skill requirements of the traditional integration method

Steps for integration	Skill requirements
Develop an Software Development Kit (SDK) to share the data with AGVs	Advanced programming and AGV skills
Configure the PLC tag which contained the related machine information	Basic PLC skills
Configure the PLC communication function modules	Advance PLC skills
Develop PLC function blocks which can publish the data to the AGV system	Advanced programming and PLC skills
Develop a client to integrate AGV with a specific PLC vendor	Advanced programming and PLC skills

Table 5.2: The steps and skill requirements of the SAMS integration method

Steps for integration	Skill requirements
Develop an SDK to share the data with AGVs	Advanced programming and AGV skills
Define the OPC UA server tags which contained the related machine information	Basic OPC UA skills
Develop an OPC UA client to integrate AGV with the OPC UA server	Advanced programming skills

In the traditional integration method, because of the changing of the PLC controllers and robotics in different manufacturing systems, the connection gateway needs to be re-developed based on the new systems to integrate AGV systems with different manufacturing processes. Thus, the integration steps, including configuring PLC tags, developing the PLC communication function module and developing the PLC programme to publish data, need to work through repeatedly. And the time and cost of system integration could be increased if the new system becomes more complex with more PLCs and robotics.

However, compared with the traditional integration method, the SAMS integration method dramatically reduces the complexity of the system integration process and it requires a significantly lower level of skills. Moreover, when the shop floor layout and manufacturing scenario are updated, the proposed generic integration does not require re-developing the communication gateway and relevant function blocks. It only needs to re-link the machine information with SAMS with a low engineering cost.

5.3 Evaluation of the optimisation capability of the proposed methodology

5.3.1 Evaluation criteria and method

In order to evaluate the optimisation performance of the proposed methodology, three main production KPIs are considered, which are shown in Table 5.3. The details of the calculation formulations are explained in Table 3.5. The control variable method is applied. Two scenarios are analysed in the following sections. All variables, including the number of work stations, AGVs, and charging stations, the battery assembly recipes and process time, of the two cases are the same except for the application of SAMS.

Table 5.3: The quality criteria for evaluation

Criteria	Description
Cost of JIT	The punctuality of the AGV-based shop floor logistics delivery
AGV energy consumption	The overall energy cost of AGVs during the manufacturing process
Number of shipped products	The number of finished products

From the optimisation result of Figure 4.26, the JIT cost and the energy consumption of AGV conflict with each other. in this way, the Euler distance method is involved here to find suitable solutions from a group of feasible results, shown in Figure 4.25, for scheduling the AGVs in the battery assembly process.

$$Dis(f_1, f_2) = a \cdot (\frac{f_x^1 - f_{min}^1}{f_{max}^1 - f_{min}^1})^2 + b \cdot (\frac{f_x^2 - f_{min}^2}{f_{max}^2 - f_{min}^2})^2 \tag{5.1}$$

Where the $Dis\,(f_1,\ f_2)$ means the Euler distance between two objective functions, and the minimal Euler distance is the optimal solution. The f_{max}^1 and f_{min}^1 are the maximum and minimum value of the first objective function, and the f_{max}^2 and f_{min}^2 are the maximum and minimum values of the second objective function. The a and b are the gain weights of the two objective functions. For example, in the manufacturing company, the JIT and the shipped of good quality products are usually more important than the energy consumption, the gain weights of these criteria can be adjusted to higher than others to find a suitable solution. The value of the a and b are decided at the decision-making module depending on the overall production KPI requirements or decisions from the human operators. The final optimal solution can be chosen from a list of feasible solutions and then be passed to the fleet manager and the MES.

5.3.2 Evaluation results of optimisation performance

The production performances with and without applying the proposed methodology are compared in three aspects, the cost of JIT, energy consumption (EC) of AGVs and the required number of shipped battery packs. And their related financial costs are calculated following Equation 4.1. The specific data of three scenarios are detailed in Table 5.4. As is mentioned in Section 4.3.4, 6 AGVs, 4 charging stations are used to build the shop floor logistics in two situations. In scenario I, the “first come, first service (FCFS)” scheduling principle is considered, which means the AGV will deliver one task at a time and the task that has been waiting for the longest is delivered first. In scenario II, a list of feasible solutions is generated by SAMS (See Figure 4.25). Then, considering the JIT cost, the energy consumption of AGVs and the number of shipped battery packs via Equation 5.1. The results below are selected.

Table 5.4: Comparison of three criteria with and without applying the proposed method

	Scenario I (using FCFS)	Scenario II (using SAMS)	
		Solution I	Solution II
Cost of JIT(KWH)	22.536 (£4.42)	20.83(£4.08)	18.80(£3.68)
EC (KWH)	7.40(£1.45)	7.24(£1.41)	7.28(£1.43)
battery pack (Unit)	161	161	161

Scenario I shows a high cost of JIT, which means the working stations waste time and energy waiting for AGV work-in-process delivery instead of working on assembly battery packs. However, after scheduling the AGV system with the proposed methodology, Scenario II is able to show different solutions with a better performance in the cost of JIT and the AGV energy consumption when the same number of battery packs are assembled. For example, in this table, Solution I shows the result with the minimised the EC of AGVs, and Solution II shows the results with the minimised cost of JIT. Thus, the operator can select the suitable one for the AGV-based shop floor logistics via SAMS.

Another comparison example is given in Table 5.5. It is assumed that 161 units of battery packs are required for assembling per day. the normal scenario without SAMS still requires 6 AGVs and 4 charging stations. While the number of AGV and charging stations reduce following the scheduling strategy generated by the proposed methodology, thereby helping to reduce the initial investment cost of the shop floor logistics.

Table 5.5: Comparison of with and without applying the SAMS

	Number of AGV	Number of charging station
Without SAMS	6	4
with SAMS	5	2

5.3.3 Evaluation results of abnormality handling performance

To evaluate the disturbance handling capability of the proposed methodology, three common abnormalities are discussed in the following section. For Disturbance I and II, it is assumed that 161 units of battery packs are required for assembling per day.

Disturbance I – AGV breakdown

In this disturbance situation, one AGV in the parking area is intentionally shut down during the battery assembly process. The SAMS identifies the abnormality by monitoring the information on the OPC UA server. Thus, it triggers the re-scheduling procedures by updating the AGV number in the DES model and re-generating the AGV delivery schedules depending on the current assembly process. Finally, it generates a new group of scheduling strategies within 5 minutes along with the running assembly process. The comparison results of the scenario with and without applying SAMS are depicted in Table 5.6.

Table 5.6: Comparison results under AGV breakdown situation

	Scenario I (without SAMS)	Scenario II (with SAMS)
Cost of JIT(KWH)	23.83(£4.67)	19.99 (£3.92)
EC (KWH)	6.48 (£1.27)	6.28 (£1.23)
battery pack (Unit)	150	161

In Scenario I, the cost of JIT has increased, which means the shop floor logistics has been significantly affected by the shortage of AGVs; and the number of the finished battery pack are reduced. While in Scenario II, after applying the SAMS, the threshold of AGV has been modified to a lower level to adapt to the new situation, which keeps AGV working for a long period before travelling to the charging station. Also, the dispatch time of AGV for each station are adjusted to reduce the waiting time of the work station. Thus, the assemble performance still can keep on a better level with a similar cost of JIT and a relatively lower cost in AGV energy consumption, resulting in the finished battery packs meeting the required number.

Disturbance II – work station breakdown

Another common disturbance of the battery assembly process is the work station breakdown. In this disturbance case, the fourth machine of launch stations, legacy loops, welding stations, inspection stations and packing stations are intentionally taken out of service. After the battery assembly process starts for 1 hour and 25 minutes, the machines break down, and they are out of service for 3 hours and 20 minutes. During the battery assembly process, the new scheduling solutions are generated by the proposed methodology to meet the daily battery pack requirement, 161 units. The comparison results are shown in Table 5.7.

Table 5.7: Comparison results under machines breakdown situation

	Scenario I (without SAMS)	Scenario II (with SAMS)
Cost of JIT(KWH)	16.79 (£3.29)	9.323 (£1.83)
EC (KWH)	7.09 (£1.38)	6.84 (£1.34)
battery pack (Unit)	154	161

In scenario I, the machine breakdown causes an impact on the number of the shipped battery pack which reduces to 154. However, in scenario II, the dispatch time of AGV for each station has been rescheduled to help the work station to reduce its waiting time, which reduces the impact of machine breakdowns. Thus, the proposed method is able to complete the required number of completed battery packs, when compared with the case without machine breakdowns.

Disturbance III – New order requirement

The daily demand for the battery pack could be changed if the customer requests more products. In this disturbance case, the battery pack of one day shift (8 hours) is increased to 170. The shop floor with SAMS software application can easily update its assembly pace by changing the AGV dispatching time, while the system without SAMS cannot adapt to this demand increases. The comparison results are shown in Table 5.8.

Table 5.8: Comparison results facing new order requirement

	Scenario I (without SAMS)	Scenario II (with SAMS)
Cost of JIT(KWH)	22.54 (£4.42)	18.16 (£3.56)
EC (KWH)	7.40 (£1.45)	7.24 (£1.41)
battery pack (Unit)	161	170

From the results, the scenario I cannot handle the increase in the number of the battery pack. However, in scenario II, the new scheduling strategies are applied for the AGVs, in which the charge thresholds are lower than normal to keep AGV continuous working for a longer period, and the new dispatch time of AGV leads to shorter waiting of work stations. In this way, it helps to increase the speed of the overall assembly process. Thus, the proposed method is able to complete the new order requirement with a slight increase in energy consumption.

Please note, the results for the cost of JIT, and EC of AGVs in the different system distribution only shows the assembly performance under its disturbance situation. The solutions cannot be compared across different distribution situations, because the parameters of AGVs and work machines are different. For instance, the cost of JIT, in the Disturbance II—work station breakdown, is the least one, but it doesn't mean this system has better performance, it is because less work station is involved when calculating the cost of JIT. And, actually, this system has a less throughput of shipped battery packs compared to the throughput of the normal systems.

5.4 Chapter summary

In this chapter, the SAMS capability of the integration and scheduling optimisation for AGV-based shop floor logistics has been evaluated by comparing them with the traditional solutions' under different situations. As discussed in Chapter 2, there is a lack of a generic solution to integrate the AGVs with the manufacturing process, and current scheduling methods are difficult to optimise the AGV-based shop floor logistics with the real-time manufacturing process. Therefore, the SAMS architecture is created to fill these gaps. To sum up, this chapter validates and evaluates the performance of SAMS as below:

1. The integration evaluation shows that the proposed methodology can support the integration of the AGV system into the shop floor with higher productivity compared to the traditional integration method. With the proposed methodology, the redeployment of the shop floor logistics into different manufacturing processes becomes much easier, with a potentially significant reduction in engineering costs.

2. The optimisation evaluation validates the real-time AGV scheduling capabilities by optimising the AGV dispatching time and charging threshold to minimise the cost of delivery JIT, overall AGV energy consumption and maximising the number of shipped battery packs. Moreover, the proposed methodology re-schedules the AGVs to help the battery assembly process meet its daily target even when facing system disturbances.

Chapter 6

Conclusion

6.1 Introduction

In this chapter the research gaps, research contributions and recommendations for future work are discussed. At the beginning of the research, the author observed that AGVs offer the potential to increase the efficiency of product delivery in shop floor logistics. However, it is currently not easy to effectively and efficiently integrate an AGV system with existing manufacturing processes, and to manage and optimise the AGV tasks in the context of a real-time manufacturing process. Thus, the objectives of this thesis are outlined as below:

1. Identify the current challenges of AGV systems working in shop floor manufacturing systems and understand the research gaps.
2. Develop a generic framework that can support information sharing among work stations, AGVs and operators to integrate AGV systems with the current manufacturing system on the shop floor.
3. Design a scheduling optimisation methodology to optimise the number of AGVs, the scheduling of AGV delivery and the charging plans considering multiple objectives: i) to maximise the performance of delivery JIT, which means minimise the cost of lateness

and earliness; ii) to minimise the overall energy cost of AGV systems; and iii) to maximise the number of shipped products.

4. Develop a software application and implement it into the shop floor manufacturing system to evaluate and identify its capabilities and performance.

In summary, Objective 1 identifies the research gaps and clarifies the current drawbacks of integrating and managing multiple AGVs with the manufacturing system. A novel SAMS methodology is proposed to achieve Objective 2 and Objective 3, which is the main innovative contribution of this research. Moreover, Objective 4 validates and evaluates the performance of the SAMS methodology via a case study.

6.2 Summary of research gaps

Regarding **Objective 1**, the literature shows the rapid increase in the application of AGVs to shop floor manufacturing systems, their potential advantages and current limitations. AGV systems can significantly improve the flexibility, reconfigurability, modularity and safety of the shop floor manufacturing environment through dynamic point-to-point delivery and autonomous driving, instead of the fix-path conveyor systems and manual forklifts. However, the literature, Section 2.2 shows that current integration solutions mainly focus on the physical interaction between AGVs and work stations. Whilst basic integration is now typically achieved, current AGV systems have difficultly in responding effectively to the real-time manufacturing system requirements.

Section 2.3 shows that the integration and scheduling methods have been explored and developed for many years in both academic and industrial fields to promote cooperation between AGV based shop floor logistics and automation systems. However, Section 2.3.7 concludes that the current AGV scheduling applications struggle to effectively optimise the tasks of the AGVs and the real-time manufacturing process simultaneously.

Based on the literature, the following research gaps have been identified:

1. The lack of a generic integration framework to support the cooperation between AGVs and

manufacturing systems. This framework should not only monitor and assign the tasks for AGVs but also allow the data and control information interaction among AGV systems, the high-level system manager and automation systems during the manufacturing process.

2. The lack a scheduling optimisation method to organise the delivery schedules of AGVs based on the real-time manufacturing process. This method needs to schedule the delivery of multiple AGVs by considering the running information of the manufacturing process for improving the production KPIs. Also, it should be able to react to unexpected cases/interruptions during the manufacturing process by re-scheduling the tasks of AGVs.

6.3 Research contributions

In this thesis, the author has presented a novel architecture, Smart AGV Management System, to support the integration of AGV systems with manufacturing systems and to optimise the scheduling of AGVs' delivery tasks during the manufacturing process, which is described in Chapter III. The main innovations and contributions of this research are summarised below and in the related sections in this thesis, which have been referred for evidence:

1. With regard to **Objective 2**, a generic framework is proposed to support the integration of AGV-based shop floor logistics with the manufacturing process. This contribution has been achieved by:

    - The proposed SAMS methodology. In this methodology, the OPC UA server-based data transaction module is implemented. This module allows data sharing among the AGVs, work stations, IoT enabled devices (e.g., energy monitors, RFIDs, sensors) and MES (Section 3.2, Figure 3.1, Section 3.3, Figure 3.4, Section 4.3.3).
    - Through the proposed methodology, the AGVs are able to be generically and seamlessly integrated with the current manufacturing system on the shop floor more easily and with higher productivity comparing to the traditional integration methods (Section 5.2).

2. With regard to **Objective 3**, a real-time scheduling method is proposed to optimise the

AGV delivery tasks during the manufacturing process. This contribution has been achieved by:

- An optimisation method proposed combining the DES-based DT model and the NSGA-II algorithm (Section 3.4, Figure 3.5).
- Through this method, real-time manufacturing information is collected to simulate the real-time manufacturing process, optimising the performance of overall manufacturing systems (Section 4.3.4).

3. Considering **Objective 4**, a SAMS software application is developed and implemented into the shop floor for supporting the optimisation and integration of AGVs with manufacturing processes. This contribution has been demonstrated and evaluated by

   - The SAMS software application is developed to support the integration of AGVs with a battery assembly process (Section 4.3).
   - The proposed methodology has the potential to rapidly reduce integration time comparing to the traditional integration method (Section 5.2).
   - The proposed methodology is able to schedule the AGV dispatch time and charging threshold by considering the overall manufacturing KPIs (Section 4.4.3).
   - The proposed methodology can generate new scheduling strategies for AGVs when facing system disturbances (Section 5.3).

6.4 Future work

Although the objectives have been successfully addressed and the integration and optimisation performance of the SAMS architecture have been validated through the case study, there are still some areas that need further research.

1. The next step of work is to further develop the SAMS software application, such as the user interface, application design and test runs under different operating systems.

2. To implement the SAMS into different industrial factory environments. In this way, the feasibility and functionality of the SAMS software application can be further evaluated and can be adapted to various requirements on the shop floor.

3. To improve the accuracy of the digital twin model, the complex event processing (CEP) engine can be considered in further research. The CEP engine can support the analysis and detect the pattern of machine operation events, thereby providing more accurate machine cycle time prediction for the digital twin model.

4. A reinforcement learning-based optimisation algorithm should be considered in future research. Each action of AGV delivery affects the overall manufacturing process performance. Thus, the reinforcement learning method can be used to schedule the AGV delivery tasks for optimising production KPIs.

5. A web server-based information sharing function should be implemented. This function would allow operators and customers remote access to relevant real-time SAMS system information.

References

[1] D. Culler and J. Long, "A prototype smart materials warehouse application implemented using custom mobile robots and open source vision technology developed using emgucv," *Procedia Manufacturing*, vol. 5, pp. 1092–1106, 2016.

[2] P. Udvardy and K. Széll, "Advanced navigation of automated vehicles in smart manufacturing," *Acta Technica Napocensis*, vol. 60, no. 1, pp. 11–14, 2019.

[3] A. V. Barenji, R. V. Barenji, D. Roudi, and M. Hashemipour, "A dynamic multi-agent-based scheduling approach for smes," *The International Journal of Advanced Manufacturing Technology*, vol. 89, no. 9-12, pp. 3123–3137, 2017.

[4] "Mir —automate your internal transportation — mobile industrial robots." https://www.mobile-industrial-robots.com/. (Accessed on 12/21/2021).

[5] "Mobile robot — omron, uk." https://industrial.omron.co.uk/en/products/mobile-robot. (Accessed on 12/21/2021).

[6] J. Mehami, M. Nawi, and R. Y. Zhong, "Smart automated guided vehicles for manufacturing in the context of industry 4.0," *Procedia manufacturing*, vol. 26, pp. 1077–1086, 2018.

[7] G. Czifra, Z. Molnár, *et al.*, "Covid-19 and industry 4.0," *Res. Pap. Fac. Mater. Sci. Technol. Slovak Univ. Technol*, vol. 28, no. 46, pp. 36–45, 2020.

[8] L. T. Blessing and A. Chakrabarti, *DRM: A design reseach methodology.* Springer, 2009.

[9] T. L. IQ, "Agv and amr market for logistics, manufacturing, healthcare, disinfection, retail inventory management, security inspection, agriculture, hospital assistance, indoor &

outdoor delivery, tele-presence & tele-operation, and data platforms remote sensing - forecast to 2026," tech. rep., Research and Markets, 2021.

[10] "Connected papers — find and explore academic papers." https://www.connectedpapers.com/. (Accessed on 12/05/2022).

[11] G. Chryssolouris, *Manufacturing systems: theory and practice*. Springer Science & Business Media, 2013.

[12] O. T. Baruwa and M. A. Piera, "A coloured petri net-based hybrid heuristic search approach to simultaneous scheduling of machines and automated guided vehicles," *International Journal of Production Research*, vol. 54, no. 16, pp. 4773–4792, 2016.

[13] E. Hozdić, "Smart factory for industry 4.0: A review," *International Journal of Modern Manufacturing Technologies*, vol. 7, no. 1, pp. 28–35, 2015.

[14] D. Hrušecká, R. B. Lopes, and E. Juřičková, "Challenges in the introduction of agvs in production lines: Case studies in the automotive industry," *Serbian Journal of Management*, 2019.

[15] J. Theunissen, H. Xu, R. Y. Zhong, and X. Xu, "Smart agv system for manufacturing shopfloor in the context of industry 4.0," in *2018 25th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, pp. 1–6, IEEE, 2018.

[16] D. Kolberg, J. Hermann, F. Mohr, F. Bertelsmeier, F. Engler, R. Franken, P. Kiradjiev, M. Pfeifer, D. Richter, M. Salleem, *et al.*, "Smartfactorykl system architecture for industrie 4.0 production plants," *SmartFactoryKL, Whitepaper SF-1.2*, vol. 4, 2018.

[17] T. Gregor, M. Krajčovič, and D. Wiecek, "Smart connected logistics," *Procedia Engineering*, vol. 192, pp. 265–270, 2017.

[18] L. Ribas-Xirgo, J. M. Moreno-Villafranca, and I. F. Chaile, "On using automated guided vehicles instead of conveyors," in *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*, pp. 1–4, IEEE, 2013.

[19] A. Fellan, C. Schellenberger, M. Zimmermann, and H. D. Schotten, "Enabling communication technologies for automated unmanned vehicles in industry 4.0," in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 171–176, IEEE, 2018.

[20] M. Čech, P. Wicher, R. Lenort, T. Malčic, J. David, D. Holman, D. Staš, and J. Záruba, "Autonomous mobile robot technology for supplying assembly lines in the automotive industry," *Acta logistica*, vol. 7, no. 2, pp. 103–109, 2020.

[21] Y. Zhang, Z. Zhu, and J. Lv, "Cps-based smart control model for shopfloor material handling," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1764–1775, 2017.

[22] "The daily routine of autonomous robots." https://www.volkswagenag.com/en/news/stories/2018/01/the-daily-routine-of-autonomous-robots.html. (Accessed on 09/13/2021).

[23] "Warehouse automation solutions — toyota material handling uk." https://toyota-forklifts.co.uk/automation/automated-solutions/. (Accessed on 09/13/2021).

[24] "Daimler relies on flexible autonomous guided transport system — kuka ag." https://www.kuka.com/en-gb/industries/solutions-database/2020/01/agv-solution$_biw - trucks. (Accessedon$09/13/2021).

[25] "How ford optimized its internal logistics with 3 mir robots — mobile industrial robots." https://www.mobile-industrial-robots.com/en/industry/automotive/how-ford-optimized-its-internal-logistics-with-3-mir-robots/. (Accessed on 09/13/2021).

[26] "Vertical wing assembly." https://www.foriauto.com/Our-Products/Automated-Material-Handling/High-Capacity-AGVs/Vertical-Wing-Assembly. (Accessed on 09/13/2021).

[27] "Kuka robotics case study at aviation — kuka ag." https://www.kuka.com/en-de/industries/solutions-database/2016/07/solution-robotics-airbus. (Accessed on 09/13/2021).

[28] "An intelligent fleet for the audi a8." https://www.grenzebach.com/en/company/news-press/detail/an-intelligent-fleet-for-the-audi-a8/. (Accessed on 09/13/2021).

[29] "Why did dell choose geek+'s picking solutions to upgrade its warehouse." https://blog.geekplus.com/case-studies/why-did-dell-choose-geeks-picking-solutions-to-upgrade-its-warehouse. (Accessed on 09/13/2021).

[30] "Download — dematic north america." https://www.dematic.com/en-us/downloads-and-resources/case-studies/download/?map=1id=1728. (Accessed on 09/13/2021).

[31] C. Cronin, A. Conway, A. Awasthi, and J. Walsh, "Flexible manufacturing using automated material handling and autonomous intelligent vehicles," in *2020 31st Irish Signals and Systems Conference (ISSC)*, pp. 1–6, IEEE, 2020.

[32] C. Cronin, A. Conway, and J. Walsh, "State-of-the-art review of autonomous intelligent vehicles (aiv) technologies for the automotive and manufacturing industry," in *2019 30th Irish Signals and Systems Conference (ISSC)*, pp. 1–6, IEEE, 2019.

[33] "Motion control solutions — kollmorgen — industrial servomotors servo drives ac dc motors." https://www.kollmorgen.com/en-us/. (Accessed on 01/02/2022).

[34] J. Böllhoff, J. Metternich, N. Frick, and M. Kruczek, "Evaluation of the human error probability in cellular manufacturing," *Procedia CIRP*, vol. 55, pp. 218–223, 2016.

[35] "Human error is more common cause of unplanned downtime in manufacturing than any other sector, according to new research — business wire." https://www.businesswire.com/news/home/20171106006370/en/Human-Error-is-More-Common-Cause-of-Unplanned-Downtime-in-Manufacturing-Than-Any-Other-Sector-According-to-New-Research. (Accessed on 01/04/2022).

[36] A. Radziwon, A. Bilberg, M. Bogers, and E. S. Madsen, "The smart factory: exploring adaptive and flexible manufacturing solutions," *Procedia engineering*, vol. 69, pp. 1184–1190, 2014.

[37] F. Yao, A. Keller, M. Ahmad, B. Ahmad, R. Harrison, and A. W. Colombo, "Optimizing the scheduling of autonomous guided vehicle in a manufacturing process," in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, pp. 264–269, IEEE, 2018.

[38] L. S. Dalenogare, G. B. Benitez, N. F. Ayala, and A. G. Frank, "The expected contribution of industry 4.0 technologies for industrial performance," *International Journal of Production Economics*, vol. 204, pp. 383–394, 2018.

[39] D. Li, Å. Fast-Berglund, and D. Paulin, "Current and future industry 4.0 capabilities for information and knowledge sharing," *The International Journal of Advanced Manufacturing Technology*, vol. 105, no. 9, pp. 3951–3963, 2019.

[40] J. Wan, J. Hong, Z. Pang, B. Jayaraman, F. Shen, *et al.*, "Ieee access special section editorial: Key technologies for smart factory of industry 4.0," *IEEE Access*, vol. 7, pp. 17969–17974, 2019.

[41] J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," *Manufacturing letters*, vol. 3, pp. 18–23, 2015.

[42] V. Rudtsch, J. Gausemeier, J. Gesing, T. Mittag, and S. Peter, "Pattern-based business model development for cyber-physical production systems," *Procedia CIRP*, vol. 25, pp. 313–319, 2014.

[43] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn, and K. Ueda, "Cyber-physical systems in manufacturing," *Cirp Annals*, vol. 65, no. 2, pp. 621–641, 2016.

[44] B. Engelmann, S. Schmitt, E. Miller, V. Bräutigam, and J. Schmitt, "Advances in machine learning detecting changeover processes in cyber physical production systems," *Journal of Manufacturing and Materials Processing*, vol. 4, no. 4, p. 108, 2020.

[45] C. Blume, S. Blume, S. Thiede, and C. Herrmann, "Data-driven digital twins for technical building services operation in factories: A cooling tower case study," *Journal of Manufacturing and Materials Processing*, vol. 4, no. 4, p. 97, 2020.

[46] P. J. Mosterman and J. Zander, "Industry 4.0 as a cyber-physical system study," *Software & Systems Modeling*, vol. 15, no. 1, pp. 17–29, 2016.

[47] P. Leitão, A. W. Colombo, and S. Karnouskos, "Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges," *Computers in industry*, vol. 81, pp. 11–25, 2016.

[48] D. Mourtzis, M. Doukas, and D. Bernidaki, "Simulation in manufacturing: Review and challenges," *Procedia Cirp*, vol. 25, pp. 213–229, 2014.

[49] S. Haag and R. Anderl, "Digital twin–proof of concept," *Manufacturing Letters*, vol. 15, pp. 64–66, 2018.

[50] C. Cimino, E. Negri, and L. Fumagalli, "Review of digital twin applications in manufacturing," *Computers in Industry*, vol. 113, p. 103130, 2019.

[51] E. Glaessgen and D. Stargel, "The digital twin paradigm for future nasa and us air force vehicles," in *53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA*, p. 1818, 2012.

[52] V. D. Majstorovic and R. Mitrovic, "Industry 4.0 programs worldwide," in *International Conference on the Industry 4.0 model for Advanced Manufacturing*, pp. 78–99, Springer, 2019.

[53] K. Chaika, "Research challenges of industry 4.0," *Industry 4.0*, vol. 5, no. 1, pp. 16–20, 2020.

[54] K. Furmans, Z. Seibold, and A. Trenkle, "Future technologies in intralogistics and material handling," in *Operations, Logistics and Supply Chain Management*, pp. 545–574, Springer, 2019.

[55] "Bespoke solutions - mechtech automation group — bespoke automation." https://mechtech.group/bespoke-solutions/. (Accessed on 01/04/2022).

[56] "Meet omron's family of autonomous mobile robots — omron, uk." https://industrial.omron.co.uk/en/misc/dm/autonomous-mobile-robots?utm$_source = googleutm_medium = cpcutm_campaign = SEA_mobile - robots_EMEAutm_term = omron (Accessedon$01/04/2022).

[57] "Customized solutions with mirgo — mobile industrial robots." https://www.mobile-industrial-robots.com/mirgo/. (Accessed on 01/04/2022).

[58] R. Sell, E. Väljaots, T. Pataraia, and E. Malayjerdi, "Modular smart control system architecture for the mobile robot platform," *Proceedings of the Estonian Academy of Sciences*, vol. 68, no. 4, pp. 395–400, 2019.

[59] B. Scholten, *The road to integration: A guide to applying the ISA-95 standard in manufacturing*. Isa, 2007.

[60] L. Monostori, "Cyber-physical production systems: Roots, expectations and r&d challenges," *Procedia Cirp*, vol. 17, pp. 9–13, 2014.

[61] E. Trunzer, A. Calà, P. Leitão, M. Gepp, J. Kinghorst, A. Lüder, H. Schauerte, M. Reifferscheid, and B. Vogel-Heuser, "System architectures for industrie 4.0 applications," *Production Engineering*, vol. 13, no. 3, pp. 247–257, 2019.

[62] M. Saqlain, M. Piao, Y. Shim, and J. Y. Lee, "Framework of an iot-based industrial data management for smart manufacturing," *Journal of Sensor and Actuator Networks*, vol. 8, no. 2, p. 25, 2019.

[63] R. Harrison, D. A. Vera, and B. Ahmad, "A connective framework to support the lifecycle of cyber–physical production systems," *Proceedings of the IEEE*, vol. 109, no. 4, pp. 568–581, 2021.

[64] A. W. Müller, I. Grangel-González, and C. Lange, "An ontological view of the rami4. 0 asset administration shell.," in *KEOD*, pp. 165–172, 2020.

[65] E. Barnstedt, H. Bedenbender, M. Billmann, B. Boss, E. Clauer, M. Fritsche, K. Garrels, M. Hankel, O. Hillermeier, M. Hoffmeister, *et al.*, "Details of the asset administration shell," *Tech. Rep.*, 2018.

[66] S. Cavalieri, S. Mulé, and M. G. Salafia, "Opc ua-based asset administration shell," in *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, vol. 1, pp. 2982–2989, IEEE, 2019.

[67] M. Schleipen, S.-S. Gilani, T. Bischoff, and J. Pfrommer, "Opc ua & industrie 4.0-enabling technology with high diversity and variability," *Procedia Cirp*, vol. 57, pp. 315–320, 2016.

[68] M. Hankel and B. Rexroth, "The reference architectural model industrie 4.0 (rami 4.0)," *ZVEI*, vol. 2, no. 2, pp. 4–9, 2015.

[69] S.-H. Leitner and W. Mahnke, "Opc ua–service-oriented architecture for industrial applications," *ABB Corporate Research Center*, vol. 48, no. 61-66, p. 22, 2006.

[70] W. Dai, Y. Song, Z. Zhang, P. Wang, C. Pang, and V. Vyatkin, "Modelling industrial cyber-physical systems using iec 61499 and opc ua," in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, pp. 772–777, IEEE, 2018.

[71] I. Seilonen, T. Tuovinen, J. Elovaara, I. Tuomi, and T. Oksanen, "Aggregating opc ua servers for monitoring manufacturing systems and mobile work machines," in *2016 IEEE*

21st international conference on emerging technologies and factory automation (ETFA), pp. 1–4, IEEE, 2016.

[72] P. F. S. De Melo and E. P. Godoy, "Controller interface for industry 4.0 based on rami 4.0 and opc ua," in *2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4. 0&IoT)*, pp. 229–234, Ieee, 2019.

[73] K. Zhou, T. Liu, and L. Zhou, "Industry 4.0: Towards future industrial opportunities and challenges," in *2015 12th International conference on fuzzy systems and knowledge discovery (FSKD)*, pp. 2147–2152, IEEE, 2015.

[74] K. T. Park, Y. W. Nam, H. S. Lee, S. J. Im, S. D. Noh, J. Y. Son, and H. Kim, "Design and implementation of a digital twin application for a connected micro smart factory," *International Journal of Computer Integrated Manufacturing*, vol. 32, no. 6, pp. 596–614, 2019.

[75] V. Kuts, G. E. Modoni, T. Otto, M. Sacco, T. Tähemaa, Y. Bondarenko, and R. Wang, "Synchronizing physical factory and its digital twin through an iiot middleware: a case study.," *Proceedings of the Estonian Academy of Sciences*, vol. 68, no. 4, 2019.

[76] Y. Lu, K. C. Morris, S. Frechette, *et al.*, "Current standards landscape for smart manufacturing systems," *National Institute of Standards and Technology, NISTIR*, vol. 8107, p. 39, 2016.

[77] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, "Smart factory of industry 4.0: Key technologies, application case, and challenges," *Ieee Access*, vol. 6, pp. 6505–6519, 2017.

[78] Q. Liu, J. Chen, Y. Liao, E. Mueller, D. Jentsch, F. Boerner, and M. She, "An application of horizontal and vertical integration in cyber-physical production systems," in *2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pp. 110–113, IEEE, 2015.

[79] T. Stock and G. Seliger, "Opportunities of sustainable manufacturing in industry 4.0," *Procedia Cirp*, vol. 40, pp. 536–541, 2016.

[80] "Centralized control with mir fleet — mobile industrial robots." https://www.mobile-industrial-robots.com/solutions/mir-applications/mir-fleet/. (Accessed on 01/04/2022).

[81] "Agv intelligent management system - casun." https://casun-agv.com/agv-intelligent-management-system/. (Accessed on 09/13/2021).

[82] "Max control system — max agv." https://maxagv.com/max-management-system/. (Accessed on 09/13/2021).

[83] "Control software - autoguide mobile robots." https://www.agmobilerobots.com/control-software/. (Accessed on 09/13/2021).

[84] "Bagware - software control solutions — alstef group." https://alstefgroup.com/baggage-handling/software/bagware/. (Accessed on 09/13/2021).

[85] "Fleet control software ds navios — ds automotion." https://www.ds-automotion.com/en/technology/ds-navios/?L (Accessed on 09/13/2021).

[86] "Sgv manager - automated material handling system - jbt." https://www.jbtc.com/automated-systems/products-and-applications/products/software-and-services/software/sgv-manager. (Accessed on 09/13/2021).

[87] "Mir fleet — mobile industrial robots." https://www.mobile-industrial-robots.com/en/solutions/robots/mir-accessories/mir-fleet/. (Accessed on 09/13/2021).

[88] "Otto fleet manager." https://ottomotors.com/fleet-manager. (Accessed on 09/13/2021).

[89] "Fleet management supervisor software — seegrid." https://seegrid.com/fleet-central/supervisor/. (Accessed on 09/13/2021).

[90] "Fleet manager — omron, uk." https://industrial.omron.co.uk/en/products/fleet-manager. (Accessed on 09/13/2021).

[91] B. Huang, R. Jiang, and G. Zhang, "Search strategy for scheduling flexible manufacturing systems simultaneously using admissible heuristic functions and nonadmissible heuristic functions," *Computers & Industrial Engineering*, vol. 71, pp. 21–26, 2014.

[92] V. Kaplanoğlu, C. Şahin, A. Baykasoğlu, R. Erol, A. Ekinci, and M. Demirtaş, "A multi-agent-based approach to dynamic scheduling of machines and automated guided vehicles (agv) in manufacturing systems by considering agv breakdowns," *International Journal of Engineering Research & Innovation*, vol. 7, no. 2, pp. 32–38, 2015.

[93] Q.-V. Dang, I. Nielsen, K. Steger-Jensen, and O. Madsen, “Scheduling a single mobile robot for part-feeding tasks of production lines,” *Journal of Intelligent Manufacturing*, vol. 25, no. 6, pp. 1271–1287, 2014.

[94] A. I. Corréa, A. Langevin, and L.-M. Rousseau, “Scheduling and routing of automated guided vehicles: A hybrid approach,” *Computers & operations research*, vol. 34, no. 6, pp. 1688–1707, 2007.

[95] Y.-L. Cheng, H.-C. Sen, K. Natarajan, C.-P. Teo, and K.-C. Tan, “Dispatching automated guided vehicles in a container terminal,” in *Supply chain optimization*, pp. 355–389, Springer, 2005.

[96] T. F. Abdelmaguid, A. O. Nassef, B. A. Kamal, and M. F. Hassan, “A hybrid ga/heuristic approach to the simultaneous scheduling of machines and automated guided vehicles,” *International journal of production research*, vol. 42, no. 2, pp. 267–281, 2004.

[97] L. Qiu and W.-J. Hsu, “Algorithms for routing agvs on a mesh topology,” in *European Conference on Parallel Processing*, pp. 595–599, Springer, 2000.

[98] L. Qiu and W.-J. Hsu, “Routing agvs on a mesh-like path topology,” in *Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No. 00TH8511)*, pp. 392–397, IEEE, 2000.

[99] M. De Guzman, N. Prabhu, and J. Tanchoco, “Complexity of the agv shortest path and single-loop guide path layout problems,” *International Journal of Production Research*, vol. 35, no. 8, pp. 2083–2092, 1997.

[100] M. S. AKTURK and H. YILMAZ, “Scheduling of automated guided vehicles in a decision making hierarchy,” *International Journal of Production Research*, vol. 34, no. 2, pp. 577–591, 1996.

[101] Ü. Bilge and G. Ulusoy, “A time window approach to simultaneous scheduling of machines and material handling system in an fms,” *Operations Research*, vol. 43, no. 6, pp. 1058–1070, 1995.

[102] D. Y. Lee and F. DiCesare, “Scheduling flexible manufacturing systems using petri nets and heuristic search,” *IEEE Transactions on robotics and automation*, vol. 10, no. 2, pp. 123–132, 1994.

[103] H. W. Jing and H. S. Ying, "Route planning of automated guided vehicles," in *Proceedings of the Intelligent Vehicles' 94 Symposium*, pp. 479–484, IEEE, 1994.

[104] N. Jawahar, P. Aravindan, S. Ponnambalam, and R. Suresh, "Agv schedule integrated with production in flexible manufacturing systems," *The International Journal of Advanced Manufacturing Technology*, vol. 14, no. 6, pp. 428–440, 1998.

[105] F. Taghaboni and J. Tanchoco, "A lisp-based controller for free-ranging automated guided vehicle systems," *International Journal of Production Research*, vol. 26, no. 2, pp. 173–188, 1988.

[106] R. J. Gaskins and J. M. Tanchoco, "Flow path design for automated guided vehicle systems," *International Journal of Production Research*, vol. 25, no. 5, pp. 667–676, 1987.

[107] S. C. Daniels, *Real-time conflict resolution in automated guided vehicle scheduling.* The Pennsylvania State University, 1988.

[108] L. Qiu and W.-J. Hsu, "Conflict-free agv routing in a bi-directional path layout," in *Proceedings of the 5th international conference on computer integrated manufacturing*, vol. 1, pp. 392–403, Citeseer, 2000.

[109] L. Qiu and W.-J. Hsu, "A bi-directional path layout for conflict-free routing of agvs," *International Journal of Production Research*, vol. 39, no. 10, pp. 2177–2195, 2001.

[110] H. Fazlollahtabar and M. Saidi-Mehrabad, "Methodologies to optimize automated guided vehicle scheduling and routing problems: a review study," *Journal of Intelligent & Robotic Systems*, vol. 77, no. 3, pp. 525–545, 2015.

[111] C. Oboth, R. Batta, and M. Karwan, "Dynamic conflict-free routing of automated guided vehicles," *International Journal of Production Research*, vol. 37, no. 9, pp. 2003–2030, 1999.

[112] G. Demesure, M. Defoort, A. Bekrar, D. Trentesaux, and M. Djemai, "Navigation scheme with priority-based scheduling of mobile agents: Application to agv-based flexible manufacturing system," *Journal of Intelligent & Robotic Systems*, vol. 82, no. 3-4, pp. 495–512, 2016.

[113] G. Demesure, M. Defoort, A. Bekrar, D. Trentesaux, and M. Djemai, "Decentralized motion planning and scheduling of agvs in an fms," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1744–1752, 2017.

[114] D. B. M. Fontes and S. M. Homayouni, "Joint production and transportation scheduling in flexible manufacturing systems," *Journal of Global Optimization*, vol. 74, no. 4, pp. 879–908, 2019.

[115] H. Fazlollahtabar, "Scheduling of multiple autonomous guided vehicles for an assembly line using minimum cost network flow," *Journal of Optimization in Industrial Engineering*, vol. 11, no. 1, pp. 185–193, 2018.

[116] A. I. Corréa, A. Langevin, and L.-M. Rousseau, "A scheduling and conflict-free routing problem solved with a hybrid constraint programming/mixed integer programming approach," *Les Cahiers du GERAD ISSN*, vol. 711, p. 2440, 2005.

[117] T. Nishi, Y. Hiranaka, and I. E. Grossmann, "A bilevel decomposition algorithm for simultaneous production scheduling and conflict-free routing for automated guided vehicles," *Computers & Operations Research*, vol. 38, no. 5, pp. 876–888, 2011.

[118] S. Gelareh, R. Merzouki, K. McGinley, and R. Murray, "Scheduling of intelligent and autonomous vehicles under pairing/unpairing collaboration strategy in container terminals," *Transportation Research Part C: Emerging Technologies*, vol. 33, pp. 1–21, 2013.

[119] Q.-V. Dang and I. Nielsen, "Simultaneous scheduling of machines and mobile robots," in *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pp. 118–128, Springer, 2013.

[120] A. Mehrabian, R. Tavakkoli-Moghaddam, and K. Khalili-Damaghani, "Multi-objective routing and scheduling in flexible manufacturing systems under uncertainty," *Iranian Journal of Fuzzy Systems*, vol. 14, no. 2, pp. 45–77, 2017.

[121] M. Mousavi, H. J. Yap, S. N. Musa, F. Tahriri, and S. Z. Md Dawal, "Multi-objective agv scheduling in an fms using a hybrid of genetic algorithm and particle swarm optimization," *PloS one*, vol. 12, no. 3, p. e0169817, 2017.

[122] M. Mousavi, H. Yap, S. Musa, and S. Dawal, "A fuzzy hybrid ga-pso algorithm for multi-objective agv scheduling in fms," *a a*, vol. 1, no. 1, p. 1, 2017.

[123] Y. Liu, S. Ji, Z. Su, and D. Guo, "Multi-objective agv scheduling in an automatic sorting system of an unmanned (intelligent) warehouse by using two adaptive genetic algorithms and a multi-adaptive genetic algorithm," *PloS one*, vol. 14, no. 12, p. e0226161, 2019.

[124] J. Jin and X.-H. Zhang, "Multi agv scheduling problem in automated container terminal," *Journal of Marine Science and Technology*, vol. 24, no. 1, p. 5, 2016.

[125] D. Weyns, N. Boucké, T. Holvoet, and B. Demarsin, "Dyncnet: A protocol for dynamic task assignment in multiagent systems," in *First International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2007)*, pp. 281–284, IEEE, 2007.

[126] R. G. Smith, "The contract net protocol: High-level communication and control in a distributed problem solver," in *Readings in distributed artificial intelligence*, pp. 357–366, Elsevier, 1988.

[127] J. Wang, Y. Zhang, Y. Liu, and N. Wu, "Multiagent and bargaining-game-based real-time scheduling for internet of things-enabled flexible job shop," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2518–2531, 2018.

[128] X. Zhao, H. Liu, S. Lin, and Y. Chen, "Design and implementation of a multiple agv scheduling algorithm for a job-shop.," *International Journal of Simulation Modelling (IJSIMM)*, vol. 19, no. 1, 2020.

[129] T. Xue, P. Zeng, and H. Yu, "A reinforcement learning method for multi-agv scheduling in manufacturing," in *2018 IEEE International Conference on Industrial Technology (ICIT)*, pp. 1557–1561, IEEE, 2018.

[130] J. T. Lin, C.-C. Chiu, and Y.-H. Chang, "Simulation-based optimization approach for simultaneous scheduling of vehicles and machines with processing time uncertainty in fms," *Flexible Services and Manufacturing Journal*, vol. 31, no. 1, pp. 104–141, 2019.

[131] A. B. Viharos and I. Németh, "Simulation and scheduling of agv based robotic assembly systems," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1415–1420, 2018.

[132] H. Rashidi and E. P. Tsang, "A complete and an incomplete algorithm for automated guided vehicle scheduling in container terminals," *Computers & Mathematics with Applications*, vol. 61, no. 3, pp. 630–641, 2011.

[133] H. Fazlollahtabar, M. Saidi-Mehrabad, and E. Masehian, "Mathematical model for deadlock resolution in multiple agv scheduling and routing network: a case study," *Industrial Robot: An International Journal*, 2015.

[134] P. Udhayakumar and S. Kumanan, "Task scheduling of agv in fms using non-traditional optimization techniques," *International Journal of Simulation Modelling*, vol. 9, no. 1, pp. 28–39, 2010.

[135] Y. Yang, M. Zhong, Y. Dessouky, and O. Postolache, "An integrated scheduling method for agv routing in automated container terminals," *Computers & Industrial Engineering*, vol. 126, pp. 482–493, 2018.

[136] S. Huang, C. Brown, and G. Hu, "Shop floor agv assignment optimization with uncertain request arrival," in *IIE Annual Conference. Proceedings*, pp. 1115–1120, Institute of Industrial and Systems Engineers (IISE), 2017.

[137] S. S. Sankar, S. Ponnambalam, and M. Gurumarimuthu, "Scheduling flexible manufacturing systems using parallelization of multi-objective evolutionary algorithms," *The International Journal of Advanced Manufacturing Technology*, vol. 30, no. 3-4, pp. 279–285, 2006.

[138] M. Nageswararao, K. Narayanarao, and G. Ranagajanardhana, "Simultaneous scheduling of machines and agvs in flexible manufacturing system with minimization of tardiness criterion," *Procedia Materials Science*, vol. 5, pp. 1492–1501, 2014.

[139] D. S. Sanches, J. da Silva Rocha, M. F. Castoldi, O. Morandin, and E. R. R. Kato, "An adaptive genetic algorithm for production scheduling on manufacturing systems with simultaneous use of machines and agvs," *Journal of Control, Automation and Electrical Systems*, vol. 26, no. 3, pp. 225–234, 2015.

[140] H. E. Nouri, O. B. Driss, and K. Ghédira, "Simultaneous scheduling of machines and a single moving robot in a job shop environment by metaheuristics based clustered holonic multiagent model.," in *ICAART (2)*, pp. 51–62, 2016.

[141] Y. He, K. E. Stecke, and M. L. Smith, "Robot and machine scheduling with state-dependent part input sequencing in flexible manufacturing systems," *International Journal of Production Research*, vol. 54, no. 22, pp. 6736–6746, 2016.

[142] M. M. Tavakoli, S. M. Sajadi, and S. A. S. Aghili, "Proposing a new mathematical model and a meta-heuristic algorithm for scheduling and allocating automated guided vehicle," *International Journal of Mathematics in Operational Research*, vol. 13, no. 2, pp. 202–218, 2018.

[143] M. Zhong, Y. Yang, Y. Dessouky, and O. Postolache, "Multi-agv scheduling for conflict-free path planning in automated container terminals," *Computers & Industrial Engineering*, vol. 142, p. 106371, 2020.

[144] C. Wang, C. Jin, and Z. Li, "Bilevel programming model of low energy consumption agv scheduling problem at automated container terminal," in *2019 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE)*, pp. 195–199, IEEE, 2019.

[145] G. Li, B. Zeng, W. Liao, X. Li, and L. Gao, "A new agv scheduling algorithm based on harmony search for material transfer in a real-world manufacturing system," *Advances in Mechanical Engineering*, vol. 10, no. 3, p. 1687814018765560, 2018.

[146] W.-Q. Zou, Q.-K. Pan, and M. F. Tasgetiren, "An effective discrete artificial bee colony algorithm for scheduling an automatic-guided-vehicle in a linear manufacturing workshop," *IEEE Access*, vol. 8, pp. 35063–35076, 2020.

[147] F. Zhang and J. Li, "An improved particle swarm optimization algorithm for integrated scheduling model in agv-served manufacturing systems," *Journal of Advanced Manufacturing Systems*, vol. 17, no. 03, pp. 375–390, 2018.

[148] V. Chawla, A. Chanda, and S. Angra, "Scheduling of multi load agvs in fms by modified memetic particle swarm optimization algorithm," *Journal of Project Management*, vol. 3, no. 1, pp. 39–54, 2018.

[149] C. Chen, L. K. Tiong, and I.-M. Chen, "Using a genetic algorithm to schedule the space-constrained agv-based prefabricated bathroom units manufacturing system," *International Journal of Production Research*, vol. 57, no. 10, pp. 3003–3019, 2019.

[150] S. Riazi, T. Diding, P. Falkman, K. Bengtsson, and B. Lennartson, "Scheduling and routing of agvs for large-scale flexible manufacturing systems," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pp. 891–896, IEEE, 2019.

[151] C. Sahin, M. Demirtas, R. Erol, A. Baykasoğlu, and V. Kaplanoğlu, "A multi-agent based approach to dynamic scheduling with flexible processing capabilities," *Journal of Intelligent Manufacturing*, vol. 28, no. 8, pp. 1827–1845, 2017.

[152] W. Xu, S. Guo, X. Li, C. Guo, R. Wu, and Z. Peng, "A dynamic scheduling method for logistics tasks oriented to intelligent manufacturing workshop," *Mathematical Problems in Engineering*, vol. 2019, 2019.

[153] R. Erol, C. Sahin, A. Baykasoglu, and V. Kaplanoglu, "A multi-agent based approach to dynamic scheduling of machines and automated guided vehicles in manufacturing systems," *Applied soft computing*, vol. 12, no. 6, pp. 1720–1732, 2012.

[154] V. Chawla, A. Chanda, and S. Angra, "The scheduling of automatic guided vehicles for the workload balancing and travel time minimi-zation in the flexible manufacturing system by the nature-inspired algorithm," *Journal of Project Management*, vol. 4, no. 1, pp. 19–30, 2019.

[155] K. Zheng, D. Tang, A. Giret, M. A. Salido, and Z. Sang, "A hormone regulation–based approach for distributed and on-line scheduling of machines and automated guided vehicles," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 232, no. 1, pp. 99–113, 2018.

[156] C. Liu, J. Tan, H. Zhao, Y. Li, and X. Bai, "Path planning and intelligent scheduling of multi-agv systems in workshop," in *2017 36th Chinese Control Conference (CCC)*, pp. 2735–2739, IEEE, 2017.

[157] B. Huang, R. Jiang, and G. Zhang, "Heuristic search for scheduling flexible manufacturing systems using multiple heuristic functions," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pp. 178–187, Springer, 2014.

[158] J. Fu, J. Zhang, G. Ding, S. Qin, and H. Jiang, "Determination of vehicle requirements of agv system based on discrete event simulation and response surface methodology," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, p. 0954405421995602, 2021.

[159] F. T. Chan, H. K. Chan, and A. Kazerooni, "Real time fuzzy scheduling rules in fms," *Journal of Intelligent Manufacturing*, vol. 14, no. 3, pp. 341–350, 2003.

[160] M. W. Grieves, "Virtually intelligent product systems: digital and physical twins," 2019.

[161] W. Kuehn, "Digital twins for decision making in complex production and logistic enterprises," *International Journal of Design & Nature and Ecodynamics*, vol. 13, no. 3, pp. 260–271, 2018.

[162] Q. Qi and F. Tao, "Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison," *Ieee Access*, vol. 6, pp. 3585–3593, 2018.

[163] F. Tao and M. Zhang, "Digital twin shop-floor: a new shop-floor paradigm towards smart manufacturing," *Ieee Access*, vol. 5, pp. 20418–20427, 2017.

[164] "Industry 4.0 and the digital twin technology — deloitte insights." https://www2.deloitte.com/us/en/insights/focus/industry-4-0/digital-twin-technology-smart-factory.html. (Accessed on 12/21/2021).

[165] C. Zhuang, T. Miao, J. Liu, and H. Xiong, "The connotation of digital twin, and the construction and application method of shop-floor digital twin," *Robotics and Computer-Integrated Manufacturing*, vol. 68, p. 102075, 2021.

[166] E. Bottani, A. Cammardella, T. Murino, and S. Vespoli, "From the cyber-physical system to the digital twin: the process development for behaviour modelling of a cyber guided vehicle in m2m logic," *XXII Summer School Francesco TurcoIndustrial Systems Engineering*, pp. 1–7, 2017.

[167] D. Gyulai, J. Bergmann, A. Lengyel, B. Kádár, and D. Czirkó, "Simulation-based digital twin of a complex shop-floor logistics system," in *2020 Winter Simulation Conference (WSC)*, pp. 1849–1860, IEEE, 2020.

[168] J. Yan, Z. Liu, C. Zhang, T. Zhang, Y. Zhang, and C. Yang, "Research on flexible job shop scheduling under finite transportation conditions for digital twin workshop," *Robotics and Computer-Integrated Manufacturing*, vol. 72, p. 102198, 2021.

[169] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital twin in manufacturing: A categorical literature review and classification," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1016–1022, 2018.

[170] S. K. Fahl, "Benefits of discrete event simulation in modeling mining processes," 2017.

[171] C. G. Cassandras, S. Lafortune, *et al.*, *Introduction to discrete event systems*, vol. 2. Springer, 2008.

[172] I. F. Vis, "Survey of research in the design and control of automated guided vehicle systems," *European Journal of Operational Research*, vol. 170, no. 3, pp. 677–709, 2006.

[173] R. Davidrajuh, "Developing a new petri net tool for simulation of discrete event systems," in *2008 Second Asia International Conference on Modelling & Simulation (AMS)*, pp. 861–866, IEEE, 2008.

[174] G. S. Fishman, *Discrete-event simulation: modeling, programming, and analysis*. Springer Science & Business Media, 2013.

[175] M. Saez, F. P. Maturana, K. Barton, and D. M. Tilbury, "Real-time manufacturing machine and system performance monitoring using internet of things," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 4, pp. 1735–1748, 2018.

[176] M. A. A. Khan, J. Butt, H. Mebrahtu, H. Shirvani, and M. N. Alam, "Data-driven process reengineering and optimization using a simulation and verification technique," *Designs*, vol. 2, no. 4, p. 42, 2018.

[177] M. A. A. Khan, J. Butt, H. Mebrahtu, H. Shirvani, A. Sanaei, and M. N. Alam, "Integration of data-driven process re-engineering and process interdependence for manufacturing optimization supported by smart structured data," *Designs*, vol. 3, no. 3, p. 44, 2019.

[178] G. Avventuroso, M. Silvestri, and E. M. Frazzon, "Additive manufacturing plant for large scale production of medical devices: A simulation study," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1442–1447, 2018.

[179] D. Stadnicka and D. Antonelli, "Application of value stream mapping and possibilities of manufacturing processes simulations in automotive," *FME Transactions*, vol. 43, no. 4, pp. 279–286, 2015.

[180] T. Vysockỳ, L. Nováková-Marcinčinová, and E. Nováková-Marcinčinová, "Simulation as a designed tool for material flow analysis by means of witness horizon," in *MATEC Web of Conferences*, vol. 137, p. 01020, EDP Sciences, 2017.

[181] A. Greasley and J. S. Edwards, "Enhancing discrete-event simulation with big data analytics: A review," *Journal of the Operational Research Society*, vol. 72, no. 2, pp. 247–267, 2021.

[182] J. Butt, "A strategic roadmap for the manufacturing industry to implement industry 4.0," *Designs*, vol. 4, no. 2, p. 11, 2020.

[183] F. EDITION, "Discrete-event system simulation,"

[184] B. S. Kumar, V. Mahesh, and B. S. Kumar, "Modeling and analysis of flexible manufacturing system with flexsim," *International Journal of Computational Engineering Research*, vol. 5, no. 10, pp. 1–6, 2015.

[185] I. Paprocka, W. Kempa, K. Kalinowski, and C. Grabowik, "Estimation of overall equipment effectiveness using simulation programme," in *IOP conference series: materials science and engineering*, vol. 95, p. 012155, IOP Publishing, 2015.

[186] G. Zaayman and A. Innamorato, "The application of simio scheduling in industry 4.0," in *2017 Winter Simulation Conference (WSC)*, pp. 4425–4434, IEEE, 2017.

[187] A. Steinemann, J. Taiber, G. Fadel, K. Wegener, and A. Kunz, "Adapting discrete-event simulation tools to support tactical forecasting in the automotive industry," *CoDesign*, vol. 9, no. 3, pp. 159–177, 2013.

[188] "Arena simulation software — arena simulation software united kingdom." https://www.rockwellautomation.com/en-gb/products/software/arena-simulation.html. (Accessed on 01/04/2022).

[189] "Anylogic: Simulation modeling software tools & solutions for business." https://www.anylogic.com/. (Accessed on 01/04/2022).

[190] "3d simulation modeling and analysis software — flexsim." https://www.flexsim.com/. (Accessed on 01/04/2022).

[191] "Simulation, production planning and scheduling software — simio." https://www.simio.com/. (Accessed on 01/04/2022).

[192] "Tecnomatix plant simulation foundation - siemens digital industries software." https://trials.sw.siemens.com/tecnomatix-plant-simulation-foundation/?stc=ukdi100006ef$_id = Cj0KCQiA_c -$

$OBhDFARIsAIFg3ex0mzBVkT7vK2_pIS2kJZtWQAOU8KMt9TpmWEV \quad -$
$s3aT1oF8MPIylLIaAhWPEALw_wcB \quad : \quad G \quad : \quad ss_kwcid \quad =$
$AL!463!3!555856794632!p!!g!!plant\,(Accessedon01/04/2022)$.

[193] "Simevents - matlab & simulink." https://uk.mathworks.com/products/simevents.html. (Accessed on 01/04/2022).

[194] "Witness simulation modeling software — lanner." https://www.lanner.com/en-us/technology/witness-simulation-software.html: :text=Consistently (Accessed on 01/04/2022).

[195] Z. I. Majdoub Hassani, A. El Barkany, I. El Abbassi, A. Jabri, and A. M. Darcherif, "Planning and scheduling of production system in conditioning line: industrial application, optimizationand simulation approach," *Management and Production Engineering Review*, vol. 10, 2019.

[196] M. Solutions, "Kepserverex v6 opc server," 2017.

[197] V. W. Berger and Y. Zhou, "Kolmogorov–smirnov test: Overview," *Wiley statsref: Statistics reference online*, 2014.

[198] I. S. I. 22400-2, "Automation systems and integration–key performance indicators (kpis) for manufacturing operations management–part 2: definitions and descriptions," International Standard Organization (ISO) Geneva, 2014.

[199] Y. Zhao, X. Xu, H. Li, and Y. Liu, "Stochastic customer order scheduling with setup times to minimize expected cycle time," *International Journal of Production Research*, vol. 56, no. 7, pp. 2684–2706, 2018.

[200] V. Anbumalar and R. C. Sekar, "Methods for solving cell formation, static layout and dynamic layout cellular manufacturing system problems: A review," *Asian Journal of Science and Technology*, vol. 6, no. 12, pp. 2107–2112, 2015.

[201] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[202] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* MIT press, 1992.

[203] L. Wang, A. H. Ng, and K. Deb, *Multi-objective evolutionary optimisation for product design and manufacturing.* Springer, 2011.

[204] H. Wang, Y. Fu, M. Huang, G. Q. Huang, and J. Wang, "A nsga-ii based memetic algorithm for multiobjective parallel flowshop scheduling problem," *Computers & Industrial Engineering*, vol. 113, pp. 185–194, 2017.

[205] M. Akbar and T. Irohara, "Nsga-ii variants for solving a social-conscious dual resource-constrained scheduling problem," *Expert Systems with Applications*, vol. 162, p. 113754, 2020.

[206] L. Yue, H. Fan, and M. Ma, "Optimizing configuration and scheduling of double 40 ft dual-trolley quay cranes and agvs for improving container terminal services," *Journal of Cleaner Production*, vol. 292, p. 126019, 2021.

[207] D. Yang, X. Zhou, Z. Yang, Q. Jiang, and W. Feng, "Multi-objective optimization model for flexible job shop scheduling problem considering transportation constraints: A comparative study," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8, IEEE, 2020.

[208] Z. Xu, X. Ming, M. Zheng, M. Li, L. He, and W. Song, "Cross-trained workers scheduling for field service using improved nsga-ii," *International journal of production research*, vol. 53, no. 4, pp. 1255–1272, 2015.

[209] J. Mou, L. Gao, X. Li, Q. Pan, and J. Mu, "Multi-objective inverse scheduling optimization of single-machine shop system with uncertain due-dates and processing times," *Cluster Computing*, vol. 20, no. 1, pp. 371–390, 2017.

[210] M. Souier, M. Dahane, and F. Maliki, "An nsga-ii-based multiobjective approach for real-time routing selection in a flexible manufacturing system under uncertainty and reliability constraints," *The International Journal of Advanced Manufacturing Technology*, vol. 100, no. 9, pp. 2813–2829, 2019.

[211] S. Bandyopadhyay and R. Bhattacharya, "Solving multi-objective parallel machine scheduling problem by a modified nsga-ii," *Applied Mathematical Modelling*, vol. 37, no. 10-11, pp. 6718–6729, 2013.

[212] Y. Yusoff, M. S. Ngadiman, and A. M. Zain, "Overview of nsga-ii for optimizing machining process parameters," *Procedia Engineering*, vol. 15, pp. 3978–3983, 2011.

[213] R. G. Baldovino, M. G. A. C. Bautista, A. U. Aquino, E. J. Calilung, E. Sybingco, and E. P. Dadios, "Ga optimization of coconut sugar cooking process: a preliminary study using stochastic universal sampling (sus) technique," in *Proceedings of the 9th International Conference on Computer and Automation Engineering*, pp. 346–349, 2017.

[214] V. Cicirello, "Genetic algorithm parameter control: Application to scheduling with sequence-dependent setups," in *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, pp. 136–143, 2016.

[215] A. Ghosh and M. K. Das, "Non-dominated rank based sorting genetic algorithms," *Fundamenta Informaticae*, vol. 83, no. 3, pp. 231–252, 2008.

[216] J. Warchulski and M. Warchulski, "Process automatization of creating 3d objects in autocad software," *Mechanik*, vol. 91, no. 7, pp. 564–566, 2018.

[217] G. Ferrari and A. Corinna, *Application of an AGV system for material handling to a cellular manufacturing environment: the Valeo case.* PhD thesis, Master's Thesis, Politecnico di Torino, Torino, Italy, 2018.

[218] L. GROUP *et al.*, "Witness simulation software," 2003.

[219] P. Barlas and C. Heavey, "Automation of input data to discrete event simulation for manufacturing: A review," *International Journal of Modeling, Simulation, and Scientific Computing*, vol. 7, no. 01, p. 1630001, 2016.

[220] K. A. Gore, S. Jalwadi, and S. Natarajan, "Systematic efficiency improvement by optimizing the assembly line using witness simulation software," 2018.

Appendix A

Publications

Publication 1:

Yao, F., Keller, A., Ahmad, M., Ahmad, B., Harrison, R. and Colombo, A.W., 2018, July. Optimizing the scheduling of autonomous guided vehicle in a manufacturing process. In 2018 IEEE 16th International Conference on Industrial Informatics (INDIN) (pp. 264-269). IEEE.

Publication 2:

Yao, F., Alkan, B., Ahmad, B. and Harrison, R., 2020. Improving just-in-time delivery performance of IoT-enabled flexible manufacturing systems with AGV based material transportation. Sensors, 20(21), p.6333.

Optimizing the Scheduling of Autonomous Guided Vehicle in a Manufacturing Process

Fengjia Yao*, Alexander Keller*†, Mussawar Ahmad*, Bilal Ahmad*, Robert Harrison*, Armando Walter Colombo†
* Warwick Manufacturing Group, University of Warwick, Coventry, UK
{f.yao.2, a.keller, mussawar.ahmad, b.ahmad, robert.harrison@warwick.ac.uk}
† Institue for Industrial Informatics, Automation and Robotics (I^{2}AR), University of Applied Sciences Emden-Leer, Emden, Germany
{awcolombo@technik-emden.de, awcolombo@ieee.org}

***Abstract*—Autonomous Guided Vehicles (AGVs) are considered as one of the key enablers of smart factories which make possible smart and flexible transportation of pallets and material on shop-floor. However, existing AGV fleet management solutions often suffer from poor integration with real-time manufacturing operations information systems, which negatively affects scheduling of AGVs. To exploit the full potential of AGVs in achieving just-in-time (JIT) transportation, there is a need for intelligent AGV fleet management system which not only integrate with manufacturing information technology (IT) and operational technology (OT) but also provide prediction for the shop-floor logistic based on real-time manufacturing operations information to optimize scheduling of AGVs. This paper presents an approach for a Smart AGV Management System (SAMS), which combines the real-time data analysis and digital twin models that can be deployed within complex manufacturing environments for optimized scheduling. For a proof of concept, a case study of a line side supply of components to a manual assembly station is presented.**

***Keywords*—Smart AGV Management System, Smart Factory, Digital Twin, shop-floor logistics scheduling, Real-time data analytics, Prediction**

I. Introduction

The smart factory represents an ongoing evolution from traditional factories to a fully connected, flexible and reconfigurable systems that can learn, self-adapt and self-optimize in real or near real-time to frequently changing product and production requirements. One of the key distinguishing feature of the smart factory is the 'lot size one' manufacturing.

For large volume customized production, such as automotive manufacturing, 'lot size one' implies that both line-side supply of auxiliary components and transportation of parts/pallets to stations will require greater flexibility and agility. This necessitates change in structure of traditional mass production systems, for example, several production stations, which are needed to produce a product, are not considered as one unit any more [1] [2]. Fig. 1 shows layout of a traditional production line and the smart factory.

In the smart factory, stations are considered as autonomous stand-alone units. Such autonomous units can work independently and can be added, removed and reconfigured without disrupting the production line. This allows a high degree of flexibility in terms of adapting factories to changing business requirements. However, this requires a flexible shop-floor

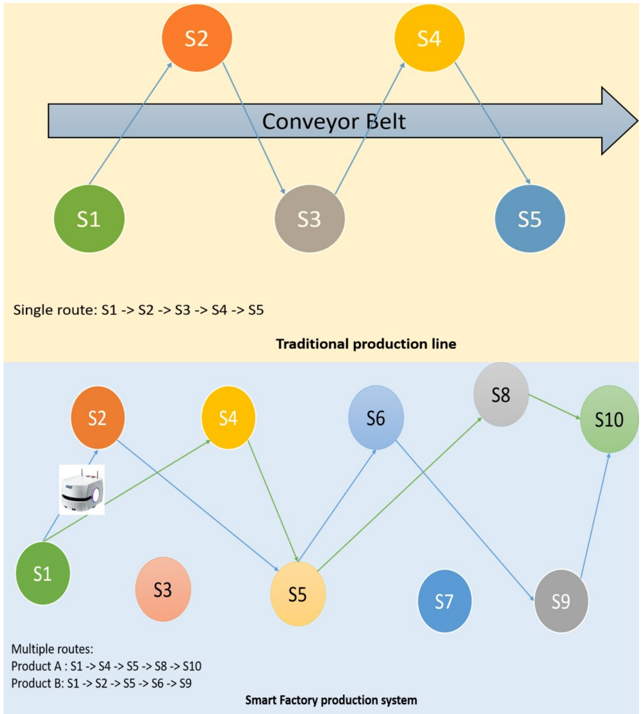

Fig. 1. Illustration of the traditional production line and Smart Factory production system [2]

logistic which could be dynamically optimised in runtime with changing requirements for material and pallet transportation.

AGVs are widely considered as one of the key enablers of flexible logistic on shop-floor [2]. They can move products and materials with no pre-defined routes. A large number of commercially available AGVs provide self-guided navigation system to find their routes to get to target workstations.

In the manufacturing industry, AGVs have to work in an integrated manner with both humans and machines. This makes communication between AGVs, machines and operators of a significant importance, which enables AGVs to work with foresight and optimise scheduling based on specific circumstances. Even though fleet managers based AGV systems have been applied in manufacturing for more than half a century [3], existing fleet managers primarily focus on the AGV routing and localisation without consideration of the use of digital twin models and real time manufacturing operations information to optimise scheduling for just in time delivery of materials.

This paper presents an approach for a Smart AGV Management System (SAMS), which combines the real-time data analysis and digital twin models that can be deployed within complex manufacturing environments for optimised scheduling. SAMS collect real-time manufacturing operations data such as throughput rate, availability and utilisation of both production stations and AGVs to predict and optimise schedule for material delivery. For a case study, the approach is used for optimisation of material delivery schedule for manual assembly station of a small scale battery assembly line.

II. State of Art

A. Digital Twin

The concept of a digital twin was introduced in 2003 at University of Michigan Executive Course on Product Lifecycle Management (PLM) by Dr. Michael Grievers [4]. In the context of manufacturing systems, a digital twin is a computer based virtual representation of physical systems built up from machine data models and simulation models [5]. Digital twins can be connected to physical assets that enable live monitoring as well as prediction and dynamic optimisation of performance of manufacturing processes [6] [7].

In the context of this paper, digital twin models provide an opportunity, not just for the planning phase, but also to predict material delivery schedule based on real time data from the shop floor to optimise scheduling of the AGV fleet. By doing this, the manufacturing processes will become more efficient, as the AGV scheduling will have already been optimised using the connected digital twin.

The use of digital twin models for AGVs has been proposed by a number of researchers. For example, in 2017, Bottani [8] has presented a concept for AGVs being implemented in a shop floor logistics environment using digital twin and showed that this is effective for simulating AGVs and optimizing schedules. According to CEIT, the use of digital twin for AGVs will affect up to 80% of the costs of layout and mapping preparations [9].

B. Fleet Mangers for AGVs

Fleet managers are commonly used for supervisory control and scheduling fleet of AGVs in industrial environments. They are mainly aimed at reducing transportation time of parts and products by optimising routes and allocation of appropriate AGV to a task. For example, an agent-based fleet manager for AGV operation control was designed in 2008 [10]. In this fleet manager, an algorithm was included to find the conflict-free and minimum time path under the AGV path networks. Simulation capabilities to evaluate various scenarios within this fleet manager offers an efficient and validated solution for AGVs running in large and complex flow networks. A warehouse based AGV fleet manager model was presented by Dimitrios [11], the approach can help AGVs to be embedded into existing manufacturing systems and optimize intra-logistics operations (e.g. less damage of products, and optimized energy consuming). A cloud robotics architecture-based fleet manager was proposed in [12]. In this architecture, the robotics and factory sensor information were consolidated to provide the live views of the whole warehouse to the AGVs.

Recently, advanced fleet managers have been developed by a number of AGV manufacturers. For example, Kollmoegen fleet manager by Comau [13] uses multiple navigation systems. The route planning is based on the laser system and the fix marks (like tapes, magnetic tags, and reflectors) as these help to provide the highly precise routes. In Swisslog, KUKA [14] and BA Systèmes [15], the fleet manager provides the real-time routing and job scheduling by looking at the factors of the environment such as the required destination and the traffic on routes in order to reduce the overall travel time. In the SGV manager, JBT [16], and the AGV supervision system, Gebo Cermex [17], the operation and travel routing historical data analysis is carried out to improve the AGVs performance in manufacturing processes. Dematic [18] has AGV management software which includes the benefit of being able to choose the most suitable AGV for the job by looking at the work flow and re-evaluating assignments. The Vehicle Manager (VM) from Savant Automation [19] includes manufacturing operation data collection and historical data based processing. VM can process the devices(sensors, data input terminals or PLCs) available information for assigning the AGVs tasks.

To meet the smart factory requirements, there is still a lack of capabilities to collect and analyse real time data from machines and factory information systems to connect with the respective Digital Twins.

C. Summary

Although various advantages of Digital Twin based manufacturing systems have been described, there remains a lack of Digital Twin based tools and methodology, to support scheduling prediction and decision making, in production lines.

The cooperation between AGVs and the automation machines, human operators and factory information systems is vital. Even though AGVs supervised by fleet managers already deployed in manufacturing since long, the fleet manager related literature focus more on the AGV traffic flow system without consideration of the build to sequence, and JIT issues or the operation information from production environment.

Furthermore, from the Smart Factory perspective, the data analytics is important in the production environment. Present fleet managers still lack the technologies that could enable the monitoring of real-time information about the manufacturing process.

III. Methodology

In order to meet the Smart Factory requirements in shop floor logistics, such as real-time operation information analysis, working process prediction, JIT, an AGV system architecture with SAMS as a core is proposed in this paper as shown in Fig. 2

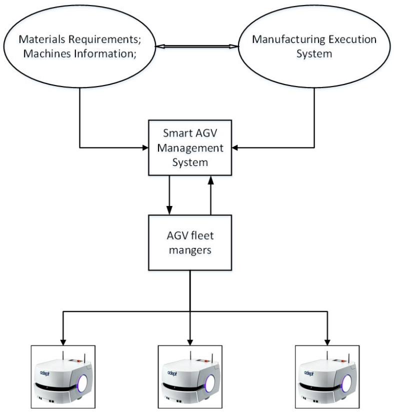

Fig. 2. The architecture of AGV system

In this architecture, SAMS is a key part and has the following characteristics to deliver its objectives: interoperability, cooperation, information integration, decentralized organisation, dynamic execution, cyber-physical integration, and human-software-hardware cooperation [20]. The software based SAMS consists of RFID technology, IoT environment, fault tolerance, real-time SCADA, information prediction, and digital twin.

The key novelty proposed in this paper is the data analysis and connectivity with digital twins. The modules of SAMS are shown in Fig. 3. SAMS consists of six main modules: communication module, data storage module, data processing module, Digital Twin module, decision making module and human machine interface module.

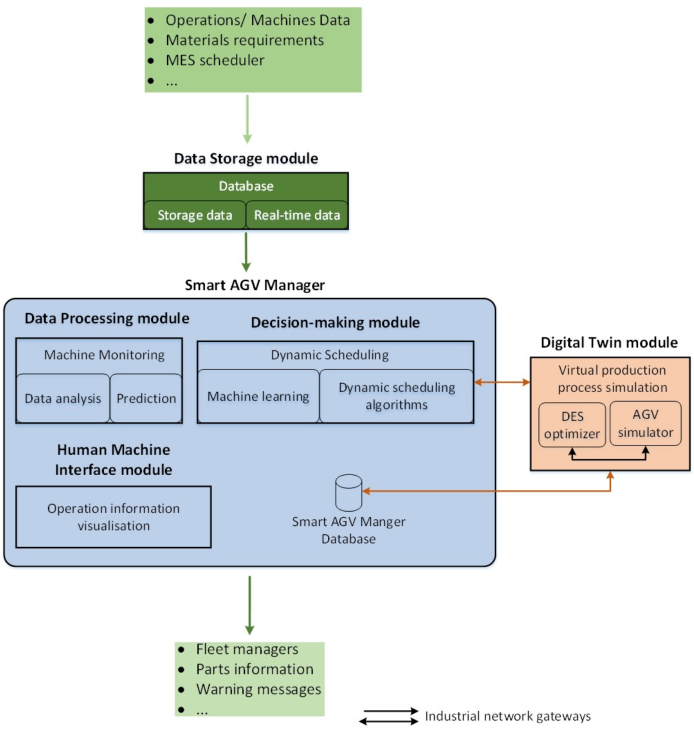

Fig. 3. The architecture of Smart AGV Management System

1) **Communication module.** This module is designed based on industrial network protocols. This allows connectivity with the shop floor local network to collect information from shop floor devices / machines (e.g. robot arms, PLCs, energy monitors, IoT devices). The collected data is then converted to Smart AGV Manager readable data format through this module. For example, the communication between SAMS and a weighing scale system is based on ModBus TCP/IP protocol, and shared on the OPC Unified Architecture (OPC UA) based platform. The sensors data, AGVs missions, and execution commands are transferred and updated through this module.
2) **Data Storage module.** This module is designed to provide a database for both knowledge and information. The shop floor machines (e.g. smarts sensors, PLCs, IoT devices) and operation data is stored. This data is accessed by other modules of SAMS.
3) **Data Processing module.** This module is implemented based on the real-time data analysis and prediction technology, in which the operation information is sensed, monitored, analysed and evaluated. For example, a weighing scale is monitored and the quantity of materials is updated in every half a second, and the RFID tags equipped materials boxes are monitored to check that a right material is presented to a station.

   In future, the authors plan to include machine break time, maintenance time, and tools replacement time as additional considerations for material usage and delivery prediction.
4) **Digital Twin module.** This module has two parts (DES optimiser and AGV simulator) for building the digital production lines in the virtual world to simulate the manufacturing process before being deployed into the real factories. The DES optimizer is for prediction simulation which represents the real life production flow and process, and depends on the time-based engine for scheduling processes. The AGV simulator is for analysing and optimising schedule and the number of AGVs.
5) **Decision-making module.** This module works as a brain of SAMS. It collects the sensors information from database in the data storage module every second, and presents a visualisation of the quantity of materials for operators. Based on the rate of materials usage, the operator working cycle time is predicted. Meanwhile, through comparison with the AGV travelling time, predicted cycle time, and ERP requirements, the dynamic threshold is generated as per Equation 1, for triggering AGV calls.

$$N_T = PR \cdot N_{diff} \cdot T_{AGV} \quad (1)$$

Where N_T is the threshold value for triggering AGV delivery call, PR is the data polling rate from server, N_{diff} is the value differences between two times data polling, also the PR and N_{diff} is regarded as the production cycle time. T_{AGV} is the AGV travelling time from warehouse to designated position, which is considered to be different during difference labour working periods, including the shop floor peak time, break time, and off working time. In the future research, to consider multi-AGVs, the priority of the assignments and AGV arrangement is to be taken into account.

6) **Human Machine Interface module.** In this module, requirements of materials are fed back to the operator, which is assigned by the upper level execution systems. The parts tracking information and warning messages are shown on the dashboard of the operations human machine interface (HMI) based on the data processing module analysed results. For example, in this paper, the different colour codes are shown by RGB-LED ring for reminding and warning. The HMI is developed in MATLAB, which displays the remaining materials in real-time (refreshing every second), dynamic threshold value, and the operator guidance related instructions. This interface is for visualising SAMS assignments and operation process information.

SAMS helps to integrate the different physical systems (e.g. sensors, weighing scales, AGVs) into the same workspace, and cooperate with each other. It not only focuses on the AGV assignment scheduling, but also contributes to the manufacturing process synchronisation and machine to machine (M2M) communication. SAMS based AGV system will play a key role in the smart factory which is expected to run autonomously.

IV. USE CASE

The use case is based on a manual assembly station of a battery manufacturing line. Manual assembly stations often have varying cycle time and often suffer from unsteady material replenishments. To address this, an AGV and Smart AGV Manager is deployed to delivers batteries just in time to a manual assembly station of the automation system demonstrator, at the University of Warwick.

A. AGV Used for Demonstration

In this use case, an Omrons AGV (Pioneer LX) was used to achieve the integration with a manufacturing processes. The Omron Adept mobile robot is an autonomous vehicle, which is designed to execute self-navigation, obstacle avoidance and self-dynamic route planning [21]. The mapping technology was implemented by Mapper3 software based on simultaneous localization and mapping (SLAM) algorithms. So the AGV firstly travels around the factory to update the unknown environment and track its location at the same time. This map can be used for all the AGVs which are deployed into the same factory, during routes and paths planning. Also, the forbidden areas can be configured and manipulated by the operator.

B. Weighing Scale System

A microcontroller based weighing scale system has been developed to monitor number of batteries, check material identification (RFID), and relay this information to the Smart AGV Manager through data storage module.A compression load cell is used to weigh the battery box. This single point load cell (model 1042) is manufactured by Huntleigh[1], the maximum loading is 3 kg, the resolution is 1 g, and the output voltage is linear to the input weight. The weighing scale consist of four components the load cell with high-resolution Analog-to-Digital Converter (ADC), Automatic Identification (Auto-ID) with RFID, display, and data processing. This system is based on NXP 1769 microcontroller manufactured by NXP[2], which collects the materials weights from load cell and converts to a string format. An industrial PC polls and publish the data through an RS-485 based interface every 0.5 second to the OPC UA server, which is the database in this use case. The data include material weight, box ID, and cycle time information.

C. Communication and Information Sharing of the AGV System with Weighing Scale System

In this use case, the OPC Unified Architecture (OPC UA) embedded KEPServer software is used. It supports the Modbus TCP/IP Ethernet which, is a commonly used communication channel for IoT devices. The smart sensors data, the real-time AGV status and manual station operation information for SAMS system are channelled through OPC UA. The communication between the different components used in this study is shown in Fig. 4.

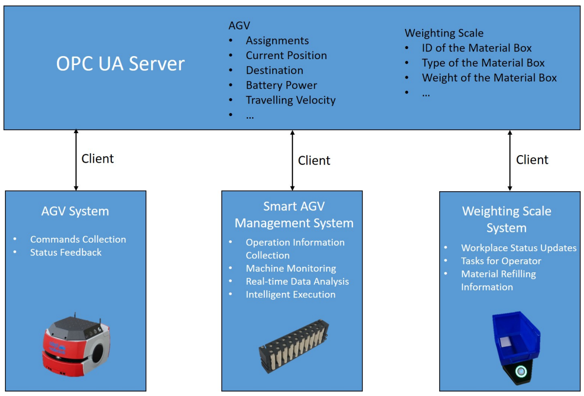

Fig. 4. The architecture of AGV system's communication

D. Experiment

The AGV system is set up in advance based on the demonstrator areas and connected with AGV client, which is running on a PC, via Wi-Fi. The battery weighing scale system is installed at the manual station, and connected with OPC UA server through Ethernet.

Firstly, the weight of a battery box and a single cell is recorded by the weighing scale system, which is used to calculate the number of batteries in the box in runtime. The AGV

[1] http://www.vpgtransducers.com/

[2] https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/lpc-cortex-m-mcus/lpc1700-cortex-m3/512kb-flash-64kb-sram-ethernet-usb-lqfp100-package:LPC1769FBD100

travelling time between two stations is trained for computing the AGV trigger time. The Smart AGV Manager uses real-time data and recorded training information to calculate operator working cycle time and set a dynamic threshold for triggering battery delivery task.

The ERP requirements based on the number of cells required per module is assigned to Smart AGV Manager. The experimental setup is illustrated in Fig. 5.

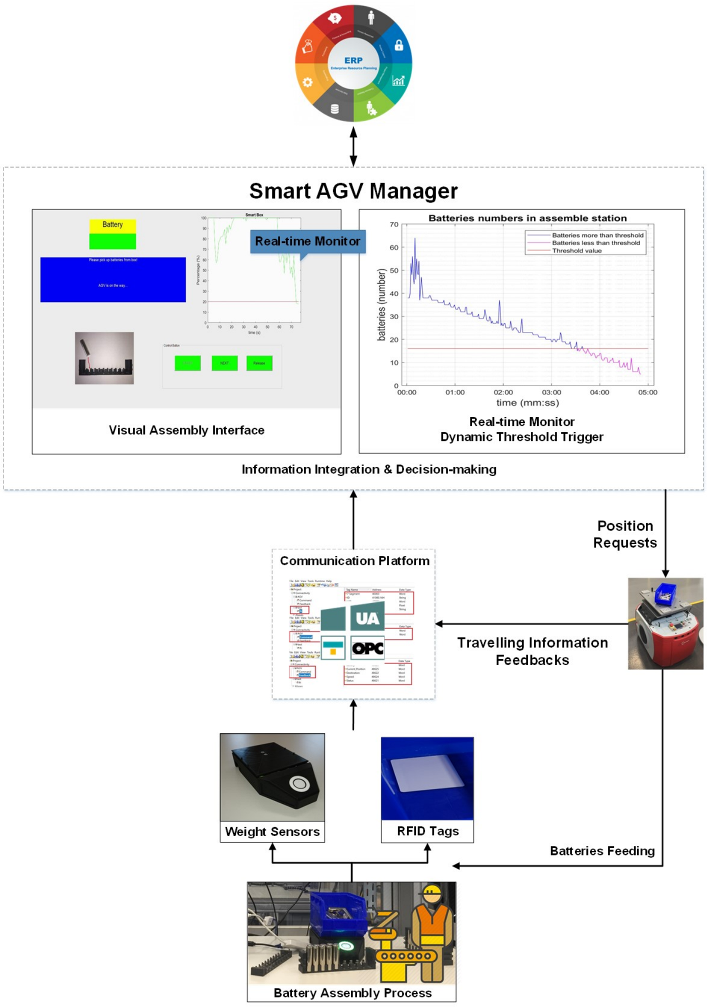

Fig. 5. Experimental Setup

As shown in Fig. 5 the battery assembly process is carried out at the manual station. When the number of cells in the box reaches the dynamic threshold, which is calculated by data processing module in SAMS, the AGV is triggered for replenishing cells. At the same time, AGV feedbacks the travelling information (e.g. velocity, x-y related coordinates, powers, destinations) to Smart AGV Manager. When AGV arrives at the manual station, it wait for the operator replacing the empty batteries box on the platform and loading the full battery box on the weighing scale. Once the task is completed, the AGV travels back to the warehouse with empty box.

V. Conclusion and Future

The paper presents an approach for optimising the scheduling of AGVs system and close integration with a manufacturing process using SAMS. The use case shows integration of the Smart AGV Manager based control system integrating with IoT devices, AGV, and manual assembly process. The presented approach can improve the manufacturing performance and efficiency by optimising line side supply of material.

In the future, the experimental setup will be extended to integrate multiple AGVs with machines (e.g. robot arm, PLC, smart sensors), other IoT devices and the digital twin to evaluate the performance of this approach using more complex AGV assignments.

Acknowledgement

This work is supported by the U.K. Engineering and Physical Sciences Research Council (EPSRC), through the Knowledge Driven Configurable Manufacturing (KDCM) research project under the Flexible and Reconfigurable Manufacturing Initiative.

References

[1] Bartodziej, Christoph Jan. "The concept industry 4.0." The Concept Industry 4.0. Springer Gabler, Wiesbaden, 2017. 27-50.
[2] Wang, Shiyong, et al. "Implementing smart factory of industrie 4.0: an outlook." International Journal of Distributed Sensor Networks 12.1 (2016): 3159805.
[3] Ullrich, Gnter. "The History of Automated Guided Vehicle Systems." Automated Guided Vehicle Systems. Springer, Berlin, Heidelberg, 2015. 1-14.
[4] Grieves, Michael. "Digital twin: Manufacturing excellence through virtual factory replication." White paper (2014).
[5] White Paper - Digital Twins for IoT Applications. 2018. Digital Twins for IoT Applications. [ONLINE] Available at: http://www.oracle.com/us/solutions/internetofthings/digital-twins-for-iot-apps-wp-3491953.pdf. [Accessed 6 April 2018].
[6] Coronado, Pedro Daniel Urbina, et al. "Part data integration in the Shop Floor Digital Twin: Mobile and cloud technologies to enable a manufacturing execution system." Journal of Manufacturing Systems (2018).
[7] Qi, Qinglin, and Fei Tao. "Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison." IEEE Access 6 (2018): 3585-3593.
[8] Bottani, E., et al. "From the Cyber-Physical System to the Digital Twin: the process development for behaviour modelling of a Cyber Guided Vehicle in M2M logic." XXII Summer School Francesco TurcoIndustrial Systems Engineering (2017): 1-7.
[9] Furmann, Radovan, Beta Furmannov, and Dorota Wicek. "Interactive design of reconfigurable logistics systems." Procedia engineering 192 (2017): 207-212.
[10] Srivastava, Sharad Chandra, et al. "Development of an intelligent agent-based AGV controller for a flexible manufacturing system." The International Journal of Advanced Manufacturing Technology 36.7-8 (2008): 780.
[11] Bechtsis, Dimitrios, et al. "Industry 4.0: Sustainable material handling processes in industrial environments." Computer Aided Chemical Engineering. Vol. 40. Elsevier, 2017. 2281-2286.
[12] Cardarelli, Elena, et al. "Cooperative cloud robotics architecture for the coordination of multi-AGV systems in industrial warehouses." Mechatronics 45 (2017): 1-13.
[13] Kollmorgen. 2018. COMAUS FIRST AGV: A FLEXIBLE SOLUTION FOR THE SMART FACTORY. [ONLINE] Available at: https://www.kollmorgen.com/en-gb/service-and-support/knowledge-center/success-stories/comau-s-first-agv-a-flexible-solution-for-the-smart-factory/. [Accessed 6 April 2018].

[14] KUKA AG. 2018. KUKA.NavigationSolution. [ONLINE] Available at: https://www.kuka.com/en-gb/products/mobility/navigation-solution. [Accessed 6 April 2018].
[15] BA Systèmes. 2018. SUPERVISION / FLEET MANAGEMENT SOFTWARE. [ONLINE] Available at: https://www.basystemes.com/en/agvs/supervision/. [Accessed 6 April 2018].
[16] JBT - SGV Manager. 2018. SGV Manager. [ONLINE] Available at: http://www.jbtc.com/automated-systems/products-and-applications/products/software-and-services/software/sgv-manager. [Accessed 6 April 2018].
[17] AGV supervison system. 2018. AGV SUPERVISION SYSTEM: AGV MANAGER. [ONLINE] Available at: http://www.gebocermex.com/equipment/intralogistic-systems/automated-guided-vehicles/agv-supervision-system. [Accessed 6 April 2018].
[18] AGV management software for transport management and traffic control of automated guided vehicles. 2018. AGV management software for traffic control & transport management. [ONLINE] Available at: http://www.egemin-automation.com/en/automation/material-handling-automation_ha-solutions_agv-systems_agv-software-systems/management-software. [Accessed 6 April 2018].
[19] System_VM. 2018. System Management Controls for AGV/AGC Systems. [ONLINE] Available at: https://www.agvsystems.com/wp-content/uploads/2013/02/System_VM.pdf. [Accessed 6 April 2018].
[20] Liu, Mingzhou, et al. "Intelligent assembly system for mechanical products and key technology based on internet of things." Journal of Intelligent Manufacturing 28.2 (2017): 271-299.
[21] Banwell, Paddy. "Adept Lynx Platform Users Guide." Adept Technology, Inc 5960: 11970-000.

sensors

MDPI

Article

Improving Just-in-Time Delivery Performance of IoT-Enabled Flexible Manufacturing Systems with AGV Based Material Transportation

Fengjia Yao [1], Bugra Alkan [2,*], Bilal Ahmad [1] and Robert Harrison [1]

1 Warwick Manufacturing Group (WMG), University of Warwick, Coventry CV4 7AL, UK; fengjia.yao@warwick.ac.uk (F.Y.); b.ahmad@warwick.ac.uk (B.A.); robert.harrison@warwick.ac.uk (R.H.)
2 Department of Computer Science and Informatics, London South Bank University, London SE1 0AA, UK
* Correspondence: alkanb@lsbu.ac.uk

Received: 18 September 2020; Accepted: 30 October 2020; Published: 6 November 2020

Abstract: Autonomous guided vehicles (AGVs) are driverless material handling systems used for transportation of pallets and line side supply of materials to provide flexibility and agility in shop-floor logistics. Scheduling of shop-floor logistics in such systems is a challenging task due to their complex nature associated with the multiple part types and alternate material transfer routings. This paper presents a decision support system capable of supporting shop-floor decision-making activities during the event of manufacturing disruptions by automatically adjusting both AGV and machine schedules in Flexible Manufacturing Systems (FMSs). The proposed system uses discrete event simulation (DES) models enhanced by the Internet-of-Things (IoT) enabled digital integration and employs a nonlinear mixed integer programming Genetic Algorithm (GA) to find near-optimal production schedules prioritising the just-in-time (JIT) material delivery performance and energy efficiency of the material transportation. The performance of the proposed system is tested on the Integrated Manufacturing and Logistics (IML) demonstrator at WMG, University of Warwick. The results showed that the developed system can find the near-optimal solutions for production schedules subjected to production anomalies in a negligible time, thereby supporting shop-floor decision-making activities effectively and rapidly.

Keywords: internet-of-things; flexible manufacturing systems; shop-floor logistics; industry 4.0; autonomous guided vehicles; decision support systems

1. Introduction

In today's highly competitive and uncertain manufacturing environment, agility and flexibility are two key factors that manufacturing systems need to possess to operate optimally and to adapt to manufacturing disturbances with minimal human intervention. Along with the recent advancements in Industry 4.0 and related technologies, a rapid configuration of manufacturing systems can be achieved through the dynamic planning of shop-floor logistics, real-time optimisation of manufacturing schedules and customised production requirements [1–3]. In this context, autonomous guided vehicles (AGVs) became an appropriate enabler to perform versatile jobs in manufacturing shop-floors. In recent years, AGVs are increasingly deployed on shop-floors to replace human labour for material handling and/or transportation jobs with an uncompromised performance [4]. This is due to their ability to help increase the manufacturing efficiency and productivity owing to their flexibility and agility [5].

In Flexible Manufacturing Systems (FMSs) with AGV based material transportation, the due time of AGVs including their earliness and lateness is significantly important in satisfying both the expected overall takt time and production cost [6,7]. Earliness leads AGVs to wait in idle, whereas lateness puts human operators and machines in temporary wait state which results in loss of production [8]. To overcome such a challenge, an optimal dispatch time of AGVs including both start time of operations for jobs at each machine in production stages and precedence relation constraints is required [9]. The previous literature concluded that the efficiency of AGV fleet management highly depends on the selection of dispatching and routing mechanisms as well as the overall integration of the AGV and machine schedules [10]. The overall integration of AGV and machine scheduling dramatically increases the complexity of FMS scheduling, as it does not only involve the job operation sequencing, but also the assignment of material handling tasks to corresponding AGVs by considering the arrival and departure time of vehicles [11–13]. This is particularly difficult as a consequence of nature in predicting the AGV transportation times as the conflicts and interferences among AGVs often cannot be neglected. As a result, there is an increasing need for IT tools to schedule/reschedule FMSs based on the integrated machine and AGV operations to rapidly respond to various manufacturing disruptions to operate in an optimal manner [14–16].

This paper presents the Smart AGV Management System (SAMS) aiming to integrate real-time shop-floor monitoring and analytics systems with production schedules of machines and AGVs to support shop-floor decision-making activities during the event of manufacturing disruptions. The SAMS and its system architecture were initially proposed in [17]. In this paper, we extend the SAMS architecture by adding a set of novel decision-support capabilities. Towards this aim, an architectural decision-support layer is designed and developed to support shop-floor decision-making at the event of manufacturing disruptions (e.g., machine breakdowns). The SAMS architecture includes a discrete event simulation (DES) model as the digital replica of the FMS under consideration, in which field-level Internet-of-Things (IoT) enabled production data are streamlined and used to enhance the accuracy of the operational behaviours of the entities defined within DES models. In the proposed framework, the production schedule is produced based on both the real-time demand information and resource status information with the help of a Mixed-Integer Nonlinear Programming (MINLP) using Genetic Algorithm (GA) integrated with the DES model. The proposed system can actively sense and transfer production abnormality information to the production management system, such that a rescheduling instruction can be released as a response action. The proposed system is deployed in the Integrated Manufacturing Logistics (IML) demonstrator developed by Automation Systems Group (ASG) at WMG, University of Warwick. The IML is a full-scale FMS integrating logistics with manufacturing operations. This system showcases Industry 4.0 methods, and encompasses both new production systems and legacy equipment within a series of advanced manufacturing scenarios, which is being used for both research and training with a range of industrial partners. The implementation of this research is expected to increase the productivity and flexibility for manufacturing systems by improving shop-floor decision-making efficiency.

The rest of the paper is structured as follows. Section 2 reviews the related literature on the offline and online FMS scheduling approaches and outlines the research gaps. Section 3 presents the overall architecture of the proposed decision support system and data communication protocols. Section 4 details the integrated shop-floor scheduling optimisation approach. Section 5 presents the implementation of the proposed decision support system on the IML demonstrator, and discusses the results and the validity of the approach. Section 6 concludes the paper and outlines the future work.

2. Literature Review

In this study, by mainly following the taxonomy proposed by [4], applied methods on the FMS scheduling are grouped into two, i.e., (*i*) offline methods and (*ii*) online (real-time-based) methods. Offline methods are used to schedule FMS operations based on the entire production planning, in which all product components are assumed to be available prior to the start of the production. Online (real-time-based) methods, in contrast, aim at scheduling manufacturing operations at the execution phases, in which shop-floor scheduling decisions are required as the manufacturing system's status changes. Applied methods on the offline scheduling can be further divided into the following categories: (*i*) the exact methods, (*ii*) heuristics, and (*iii*) simulation-based methods [18]. The exact solution methods aim at achieving the global optimum. [19]. Demesure et al. [20] proposed an AGV navigation approach for FMSs based on the combined use of a motion planner and a priority-based negotiation algorithm. Fontes and Homayouni [21] addressed the integrated scheduling of machines and AGVs in an FMS. In their approach, the FMS scheduling problem is approached using a novel mixed-integer linear programming model, where chained decisions for both machines and AGVs are connected through the completion time-constraints. Fazlollahtabar [22] proposed an AGV scheduling optimisation approach based on the minimum-cost network flow (MCF) algorithm. The approach optimises weighted completion time of tasks for each short-term window by formulating the problem of task and resource assignment as an MCF problem during each short-term scheduling.

Heuristics and meta-heuristics-based search methods are often used in scheduling of FMSs. Dang and Nielsen [23] presented a genetic algorithm-based scheduling optimisation approach for AGV based FMSs. Nageswararao et al. [24] proposed a scheduling approach simultaneously optimising both machine and AGV schedules, based on the implementation of binary particle swarm optimisation approach and vehicles assignment heuristic utilising the rebuts factor maximization function and mean tardiness. Huang et al. [25] proposed an AGV scheduling strategy using both admissible and non-admissible heuristic functions and a production-specific search scheme. The approach is aimed at minimising the makespan and maximising the average machine utilisation and tested on a set of randomly generated FMSs generated using Petri nets. In a similar study, Baruwa and Piera [4] proposed an AGV scheduling strategy evaluating all possible AGV scheduling scenarios without the imposition of a specific dispatching rule. The strategy is based on a hybrid heuristic search method, called any-time layered search (ALS), optimising the AGV schedules based on both the makespan and the exit time of the last job of the system. Sanches et al. [26] propose a simultaneous production schedule optimisation approach for both machines and AGVs using an adaptive genetic algorithm minimising the makespan with low running time. Mehrabian et al. [8] developed a two-objective mathematical programming model, i.e., due dates and processing time, integrating flow shop scheduling and AVG routing in an FMS. The model is studied using two meta-heuristics algorithms, i.e., non-dominated Sorting Genetic Algorithm, and a multi-objective particle swarm optimisation approach. Mousavi et al. [27] proposed a mathematical AGV scheduling model integrated with evolutionary algorithms to optimise the task scheduling of AGVs with the objectives of minimizing makespan and number of AGVs while considering the AGVs' battery charge. Zhong et al. [28] investigated an integrated scheduling problem of a multi-AGV based system with conflict-free path planning using a Hybrid Genetic Algorithm-Particle Swarm Optimization (HGA-PSO) algorithm. Rahman et al. [29] proposed a meta-heuristics-based scheduling approach to minimise the cycle time and total tardiness in a robotic assembly line with multiple AGVs. Wang et al. [30] aimed at improving energy consumption and production efficiency of AGV transportation using a bi-level heuristic algorithm. Liu et al. [31] proposed a multi-objective mathematical optimisation model based on the combination of two Adaptive Genetic Algorithms (AGA) and a Multi-Adaptive Genetic Algorithm (MAGA).

Online (real-time) scheduling approaches allow manufacturing companies to dynamically schedule their production systems to match the desired customer demands promptly. These approaches are, in general, time-constraint methods in which a limited amount of computation time is provided to generate a set of optimal scheduling solutions [4]. Please note that these methods can be either static or dynamic. Weyns et al. [32] developed a dynamic task assignment protocol, called DynCNET, allowing a flexible task assignment approach that can cope with the operational system dynamics. The proposed protocol is an extension of contract net protocol, CNET (see [33]), allowing AGVs' task assignments dynamically. Another approach, proposed by Chan et al. [34], is a real-time expert system for scheduling parts in an FMS based on two fuzzy-logic based decision-making/selection rules. Wang et al. [35] proposed a multi-agent-based real-time scheduling architecture, called MARS, for IoT-enabled FMSs. The MARS allows dynamic scheduling based on the coordination of real-time status of AGVs carried out by "*bargaining-game-based negotiation mechanism*" and optimises scheduling targets, such as the makespan, the critical machine workload and the total energy consumption. Zhang et al. [36] developed a cyber-physical system based smart production control model for shop-floor material handling and transportation. TF et al. [37] proposed a reinforcement learning-based method for dynamic multi-AGV flow-shop schedules aiming at minimising both the average job delays and the total makespan. Zhao et al. [38] developed a dynamic scheduling system for multi-AGV based smart factories. Sahin et al. [39] developed a multi-agent-based expert system with agent-to-agent communication and negotiations for simultaneous scheduling of both machines and AGVs in a manufacturing system operating under dynamic manufacturing constraints. Their system is based on the Prometheus methodology (see [40]), and is modelled in the JACK agent-based systems development tool. Xu et al. [41] developed an intelligent logistics scheduling model and execution method for AGVs. Their approach is based on the mode of "request-scheduling-response", and is integrated with Internet-of-Things (IoT) systems to meet the shop-floor demands in real time. The solution method is based on the combined use of a double-level hybrid genetic algorithm and ant colony optimisation (DLH-GA-ACO).

The literature review showed us that many research works are aiming to optimise FMS production schedules with and without considering production uncertainties and abnormalities such as machine breakdowns and sudden customer demand changes. In general, most of these studies investigate FMS schedules based on a static factory environment, thereby providing offline FMS scheduling approaches. The exact solution approaches can be very promising in finding the global optimum; however, they can be computationally very costly due to the vehicle routing problem being proven to be NP-hard [42]. Heuristics-based can be considered as useful tools; in particular, production performance is the main priority in terms of completion time [43]. Nevertheless, these methods have problems with trapping in local minima and equilibrium attraction. Meta heuristics optimisation algorithms, on the other hand, can be a useful solution for this, as these methods involve mechanisms to avoid getting trapped in local minima. Simulation-based approaches offers what-if analyses that can be used to select the best solution among alternatives. The online (real-time) solution methods are very helpful in solving dynamic AGV routing problems. These methods continuously update the solution space as more information exposed or available in real time. Table 1 summarises the literature review.

Table 1. A summary of the related literature review.

Type	Examples	Strengths	Weaknesses
Offline scheduling	[4,8,20,20–31,44–57]	Handles scheduling complexity Low CPU overloads	Inflexibility Deterministic behaviours Requires task arrival information Subjected to a limited execution time
Online scheduling	[32–35,37–39,41]	Handles unpredictable workloads	Reduced utilisation of resources CPU overloads are harder to detect

3. Smart AGV Management System (SAMS)

In this section, the Smart AGV Management System (SAMS) is presented for real-time scheduling optimisation for both AGVs and machines within an FMS. The decision support system connects the Integrated Manufacturing and Logistics (IML) demonstrator rig to DES models, and enables the collection and monitoring of real-time operational information, and prediction and optimisation of the job schedules for: manufacturing processes, and materials delivery and product collection activities. Moreover, the proposed system implements the allocation of AGVs in different workstations, including legacy production loops, standalone autonomous stations and manual operations stations, in shop-floor logistics under the smart factory background. An overview of the SAMS architecture is depicted in Figure 1. In this section, the digital layer of the SAMS is introduced in detail, while other two layers are briefly discussed. Please note that a detailed information about the physical and data-transaction layers can be found in [17].

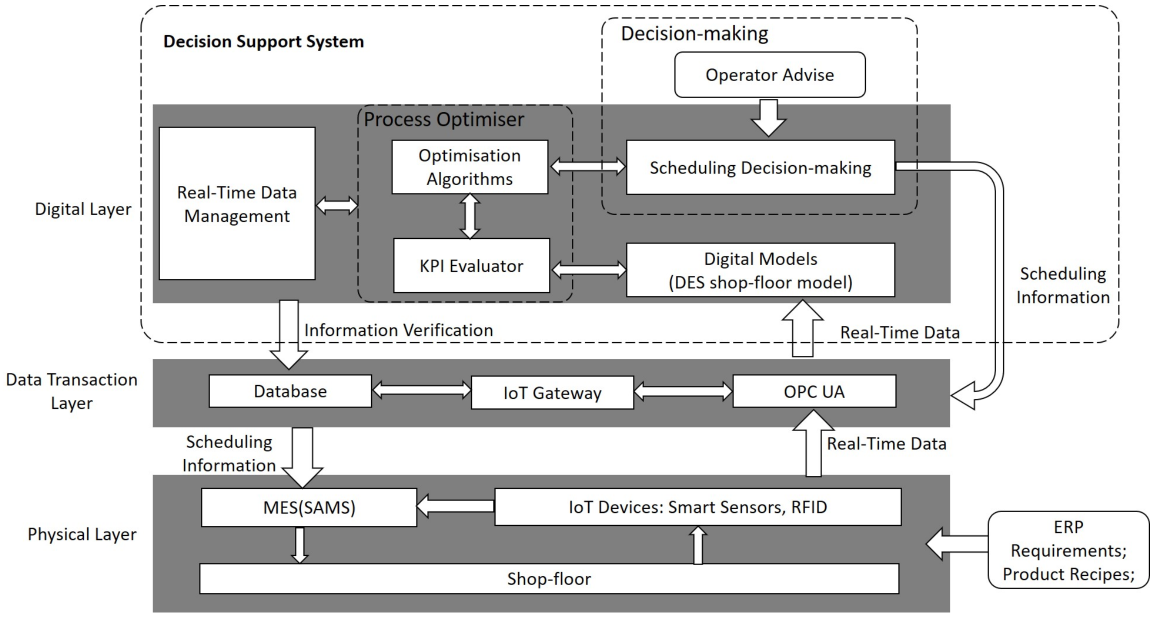

Figure 1. The SAMS architecture.

3.1. Physical and Data-Transaction Layers

The bottom two layers of the SAMS are solely responsible for collecting and transferring the real-time production data from IoT-enabled sensing devices in the manufacturing shop-floor to digital layer and vice versa. In the physical layer, two levels of monitoring are considered, i.e., workstation level and system level. Within these monitoring levels, IoT devices, including energy monitor and smart buffer sensors, are implemented on the machines, and three kinds of information, i.e., energy consumption information, machine status information and cycle-time information, are collected from the IoT-enabled field-devices within the production line. This information includes: cycle-time of each machine job, time of each transportation job between two machines, cycle-time for each loading and unloading operation, AGV charging time, AGV energy consumption in each transportation job, status of machines and AGV (i.e., breakdown, run). Cycle time for both stations and system is sensed through the RFID system, whereas machine status information, and products tracking information for different monitoring levels are captured directly from the function blocks (FB) employed within the Programming Logic Controllers (PLCs). Moreover, machine energy consumption information is directly collected from IoT-enabled smart energy meters.

In the SAMS, real-time data sharing between system modules is based on the OPC-UA protocol (see [58,59]). The OPC-UA is a machine to machine (M2M) communication protocol enabling both connectivity and interoperability among different physical and digital components. The real-time data sharing allows the SAMS to monitor and analyse the operational information from shop-floor devices and machines, such as: robots, PLCs, AGVs, and other IoT-enabled field devices, through the industry network. As an example, battery cell buffers based on the IoT-enabled weight scale are monitored, and the quantity of battery cells is updated into OPC-UA server in real time. In addition, battery packs equipped with an RFID tag are tracked by the SAMS to auto-correct the AGVs transporting in real-time. The SAMS database is created in a data transaction layer for storing shop-floor machines and operation data, such as: machine cycle time, AGV energy consumption, and production life-cycle information. The collected data can also be accessed by other supervisory systems for further production key performance indicator (KPI) assessments.

3.2. Digital Layer

Manufacturing KPIs are a set of metrics that can be used by manufacturing enterprises to evaluate the success of their manufacturing operations in meeting the performance targets [60]. These metrics include but are not limited to cost, flexibility, energy, (just-in-time) JIT material delivery performance, quality, etc. In the SAMS, the digital layer is mainly developed for the prediction of production KPIs based on a real-time data management system and a DES model coupled with KPI evaluation schemes and heuristics optimisation algorithms.

The real-time data management system is developed as a software plug-in updating operational DES parameters using the real-time production data stored within a time-series database. Currently, the developed system updates the following information within the DES model: (*i*) cycle time information for each manufacturing process, (*ii*) AGV travelling time and (*iii*) AGV energy consumption for each material transfer event, (*iv*) the charging time for each AGV, and (*v*) the demand. Although this approach provides a noticeable increase in prediction accuracy of DES models, it is planned as a future work to replace the real-time data management system with a complex event processing (CEP) engine to provide a better resolution in identifying and anticipating the relationships between the shop-floor events. The DES model uses the historical data captured from the physical layer to define individual operational parameters represented as a probability distribution function (PDF). It also receives the real-time status information of both machines and AGVs from the corresponding PLCs through the OPC-UA connection. Currently, two types of status information are defined, i.e., available and not available. The KPI evaluator sub-module is embedded within the DES Model describing the definitions and algorithms for the real-time production KPIs. These KPIs can be published into a MATLAB optimiser add-on for further evaluation through the OLE Automation Controller communication protocol.

In this research, DES models are built in the WITNESS Simulation Software [61]. The WITNESS DES tool helps engineers to model, analyse and optimise manufacturing processes, so that they can make decisions under a risk-free environment [62]. In general, the WITNESS Simulation Software can build customised manufacturing systems and production processes, and can be connected by external software and databases remotely through WITNESS Command Language (WCL) [63]. It is currently used by various manufacturing companies. For example, Ford UK integrates this software into its assembly line, and has achieved a 10% increase in the production capacity [64]. The WITNESS is capable of generating and analysing production KPIs, such as average material flow time, production cycle time and average AGV energy consumption. In this research, the DES simulations are performed to obtain the production KPIs streamlining into the optimisation engine through OLE Automation Protocol [65]. The OLE Automation Server acts as a data-interface, where commands and the data are transmitted between the WITNESS Simulation Software and the optimisation engine. The communication architecture is depicted in Figure 2.

The optimiser module is responsible for scheduling and re-scheduling both machine and AGV tasks based on evolutionary optimisation algorithms, KPI predictions and real-time resource status information. In the scheduling/rescheduling process, first, the real-time resource status information stored in the time-series database is checked, and the corresponding values are updated within the DES model. Then, a new scheduling instruction is released based on the KPI values obtained from the DES model prioritising the JIT material delivery performance. A mixed-integer Genetic Algorithm (GA) is used in the optimisation of the shop-floor logistics by minimising the JIT error and AGV energy consumption at the same time. Moreover, when a manufacturing disruption occurs, e.g., machine breakdown, the rescheduling mechanism will be triggered to reduce the influence of the disruption, thereby improving the overall production efficiency. In the proposed approach, the decision-making and optimisation modules cooperate to generate the optimal scheduling strategies, and to feed back to the manufacturing execution system (MES) located in the physical-layer. The Decision-making module mainly focuses on the dynamic scheduling strategies under varying production requirements. In such a way, production KPIs predicted by the DES model are evaluated by managers with respect to requirements before being deployed into the MES.

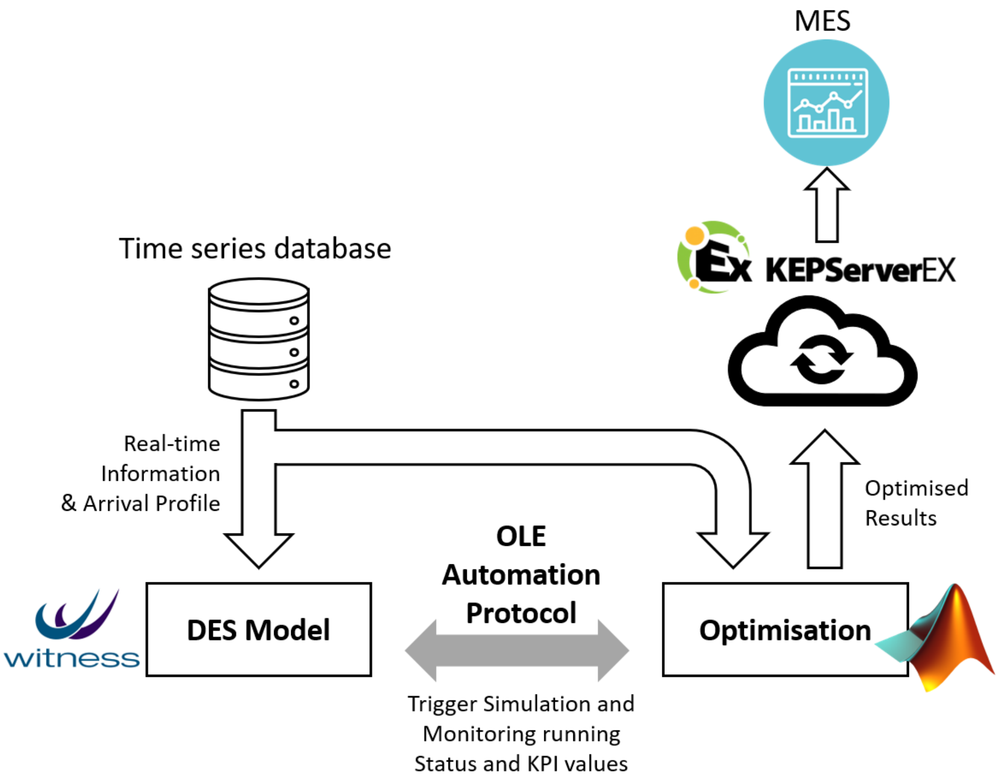

Figure 2. The real-time data communication architecture.

3.3. Shop-Floor Decision-Support

The decision support module integrates the SAMS to the existing Products Order System and MES to provide a real-time decision-support functionality during the production process. In the SAMS, the AGV scheduling and production sequences are generated and updated automatically depending on the pre-configured KPI priorities or the manufacturing station change. The integration between the existing systems and the SAMS architecture is done via OPC-UA machine to machine (M2M) communication protocol. The Products Order System used in the experiments is developed by the ASG at WMG, University of Warwick. The implementation details and architecture of this system will be the focus of a future manuscript. The SAMS receives the products order information and customer request updates from the Products Order system and uses this information along with real-time production data to generate a set of production schedules. On the other hand, the OPC-UA connects the SAMS with the MES to monitor the real-time machine states and to track the production processes. The system monitors the real-time production performance, e.g., run-time energy consumption, deviations in process cycle times and overall tardiness. When production abnormalities occur, the SAMS releases a re-scheduling scheme by considering the current machine utilisation and pre-defined KPI targets, such as machines working balance, the average energy consumption and Just-in-Time material delivery performance. The optional scheduling strategies can be chosen by the decision support system about the targeted system KPIs. Alternatively, managers can choose an optimal scheduling strategy through the application HMI and broadcasted KPI dashboards. Once the optimal strategy is selected, the job schedule is sent to the MES system for its execution. Please note that the interoperability of the decision support system allows it to access the system database/server directly. The overview of decision support components of the SAMS is shown in Figure 3.

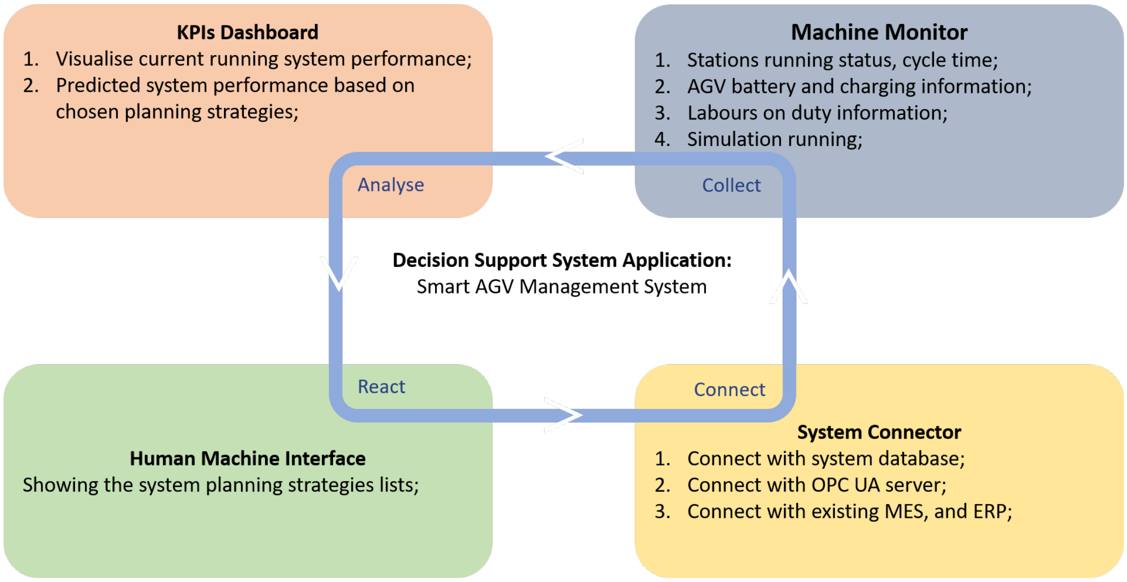

Figure 3. The main components of the decision support system.

4. Optimisation Approach

In this section, a flow-shop problem is prepared for the IML demonstrator's factory logistics. The IML is composed of several stages in which machines in the same stage perform identical manufacturing operations. The raw products follow a specific production sequence, and are transported between stages through a number of AGVs. Products are delivered into the packaging area as they are packed as a final product. Please note that each product must go through all production stages one by one in order to finish the entire assembly. It is assumed that every job has a pre-defined due time, and a JIT delivery error occurs if the job is completed after or before its due date (i.e., earliness and lateness). The objective of the problem is to find the near-optimal production schedules including both machines and AGVs that can minimise the total earliness/lateness cost as well as overall energy consumption of AGV operations, simultaneously.

A schematic representation of the presented shop-floor logistics problem is given in Figure 4. The IML shop-floor has a tiered flow-shop layout consisting of several stages: including AGV docking area, warehouse, packing area, and work machines area, etc. All AGVs are waiting in the docking area for delivery tasks. Depending on the battery status, AGVs can be recalled back to the docking area for battery recharging. In addition, the AGV parks at the docking station after the completion of the last delivery job if no further jobs are available to the AGV. Raw products are distributed to stations from the warehouse via AGVs, and they are processed through every machine stage until they are delivered to the packing area. These products are transported from one station to another through AGVs based on the delivery schedules generated by the SAMS. AGVs use predefined paths between shop-floor areas, and collisions within each path are continuously monitored and avoided by a supervisory control system.

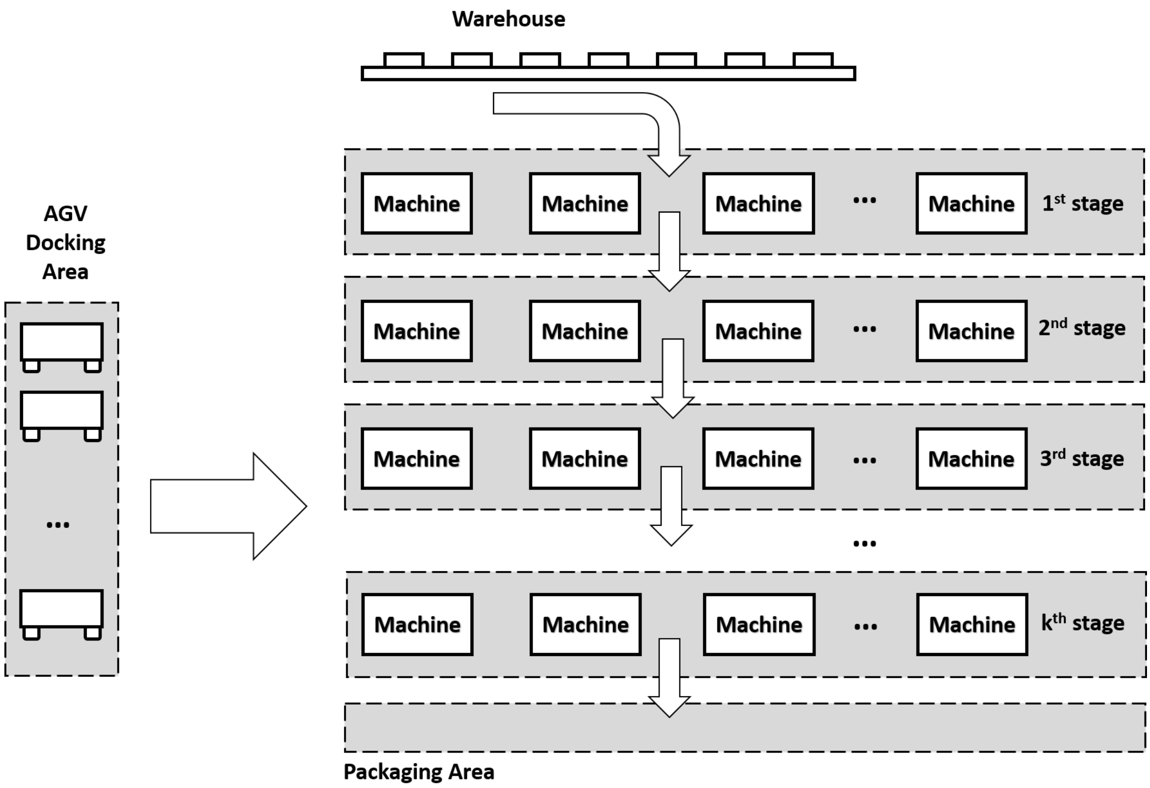

Figure 4. A schematic for the IML shop-floor logistics problem.

4.1. Problem Formulation

The mathematical notations for the presented shop-floor logistics problems are given in Table 2. The established mathematical model composed of two objective functions, described as follows.

$$Min(f) = \{f_1, f_2\} \tag{1}$$

- Objective function 1: aims to minimise the total cost associated with the earliness and lateness of the scheduled jobs, and formulated as below.

$$f_1 = \sum_{i=1}^{|T|} \alpha max\{0, d_i - C_i\} + \sum_{i=1}^{|T|} \beta max\{0, C_i - d_i\} \tag{2}$$

  Please note that the authors report based on their project experiences from seat and car manufacturing projects that, overall manufacturing performance, in general, tends to be more affected by the lateness of the jobs. Hence, it is often penalised more than the earliness of the jobs. However, the penalty costs for both earliness and lateness should be configured based on the factory and user requirements.
- Objective function 2: stands for the minimisation of the total energy consumption associated with the AGV loading and cumulative travel distances, and formulated as follows:

$$f_2 = \sum_{i=1}^{S} \sum_{j=1}^{S} \sum_{t=1}^{T} \sum_{n=1}^{N^{AGV}} dis_{ij} X_{ijnt} F(Q_{n_o} + Q_{ijnt}) \tag{3}$$

  where $F(Q_{n_o} + Q_{ijnt})$ represents the energy consumption rate related to AGV weights and travel distance.

Table 2. Notations.

Notation	Description
Sets	
S	Set of stations
T	Set of production jobs
N^{AGV}	Set of AGVs
W	Set of workstages
S_w	Number of stations in stage w
Indices	
s	Index of station, $s \in \{1, 2, ..., S\}$
t	Index of production job, $t \in \{1, 2, ..., T\}$
n	Index of AGV, $n \in \{1, 2, ..., N^{AGV}\}$
w	Index of workstage, $w \in \{1, 2, ..., W\}$
s_w	Index of station in stage w, $s_w \in \{1, 2, ..., S_w\}$
Parameters	
Q_{n_o}	The weight of no load AGV n
Q_{ijnt}	The weight of AGV n loaded, when travelling between station i and j for job t
α	Earliness cost penalty coefficient
β	Lateness cost penalty coefficient
PT_{tsw}	Processing time of job t allocated to s in stage w
d_t	Due date of job t
C_t	Completion date of job t
S_{tsw}	Starting time of job t at station s in stage w
D_{tsw}	Completion time of job t at station s in stage w
dis_{ij}	Distance between station i and j, also, $i \neq j$
r_t	Release time of the job t into the system
Decision Variables	
M_{tsw}	1 if machine s_w working on job t, else 0
X_{ijnt}	1 if AGV n travels between station i and j for job t, else 0

- These objectives are subjected to the following constraints:

$$\begin{gathered} S_{ts(w+1)} \geq D_{tsw}, \\ t = 1, ..., T,\ w = 1, ..., W, s = 1, ..., S_w \end{gathered} \tag{4}$$

$$\begin{gathered} S_{tsw} - S_{(t-1)sw} \geq PT_{(t-1)sw}, \\ t = 1, ..., T, w = 1, ..., W, s = 1, ..., S_w \end{gathered} \tag{5}$$

$$max\{\sum_{s=1}^{S_w} M_{tsw}\} = 1, \quad t = 1, ..., T, w = 1, ..., W \tag{6}$$

$$S_{ts1} \geq r_t, \quad t = 1, ..., T, s = 1, ..., S_1 \tag{7}$$

$$\begin{gathered} max\{\sum_{t=1}^{T} X_{ijnt}\} = 1, \\ i = 1, ..., S, j = 1, ..., S, n = 1, ..., N^{AGV} \end{gathered} \tag{8}$$

$$\begin{gathered} X_{ijnt}, M_{tsw} \in 0, 1 \\ i = 1, ..., S, j = 1, ..., S, s = 1, ..., S_w, \\ t = 1, ...T, w = 1, ..., W, n = 1, ..., N^{AGV} \end{gathered} \tag{9}$$

In the above equations, constraint (4) is used to ensure that the precedence relations between stages of a job for every AGVs is not breached. Constraint (5) ensures that multiple jobs cannot be performed by a machine at a time. Constraint (6) is used to fulfil the requirement that a job cannot be performed more than one machine in a stage. Constraint (7) enforces the time difference between start time of machine in the first stage and the release time of the jobs that are assigned to them must be equal or greater zero. Constraint (8) ensures that an AGV cannot perform more than one material transportation task at a time. Constraint (9) states the variables' binary nature.

4.2. Assumptions

The following are the assumptions in formulating the model:

- The parameters of machines, including: set up time and processing time are known and based on continuously updated historical production data;
- The parameters of AGVs, including: energy consumption rate, battery capacity and travelling speed are known and based on continuously updated historical production data;
- The demand information is continuously updated in real time;
- Machine output buffers have a fixed capacity limit;
- The AGV fleet capacity is enough to cover all transportation jobs;
- The AGV will not be called by the machine when the machine output buffer is empty.

4.3. Genetic Algorithm Based Solving Method

A meta-heuristics algorithm is widely applied for searching the global optimal solution for scheduling problems [66]. In this article, a mixed-integer GA, which is one of evolutionary optimisation algorithms imitating the natural selection and genetics [67], is chosen to search the near-optimal machine jobs sequence and the AGV distribution rules for battery assembly processes performing within the IML. The GA has been used to solve a wide variety of combinatorial optimisation problems and obtained optimal or near-optimal results efficiently. The GA examples for FMS scheduling optimisation problems include: [68–73]. The data-flow between optimisation module and the DES model is given in Figure 5. The flow chart of GA-based optimisation approach consisting of the following steps is shown in Figure 6. In the proposed approach, the arrival products sequence and AGV distribution rules for each arrival parts are the input for the DES model, whereas production KPIs are considered as outputs.

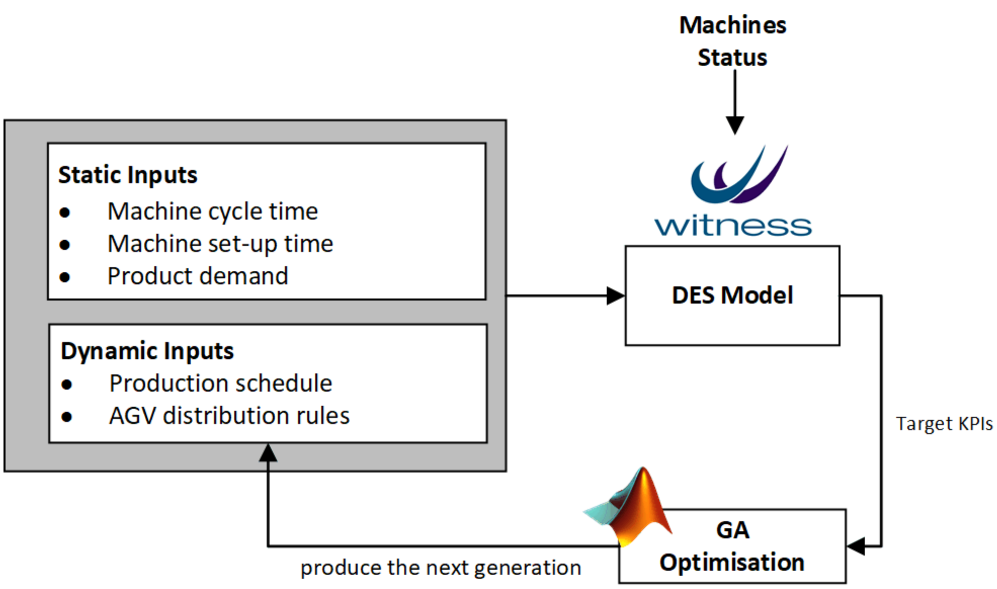

Figure 5. The data-flow between the optimisation module and the DES model.

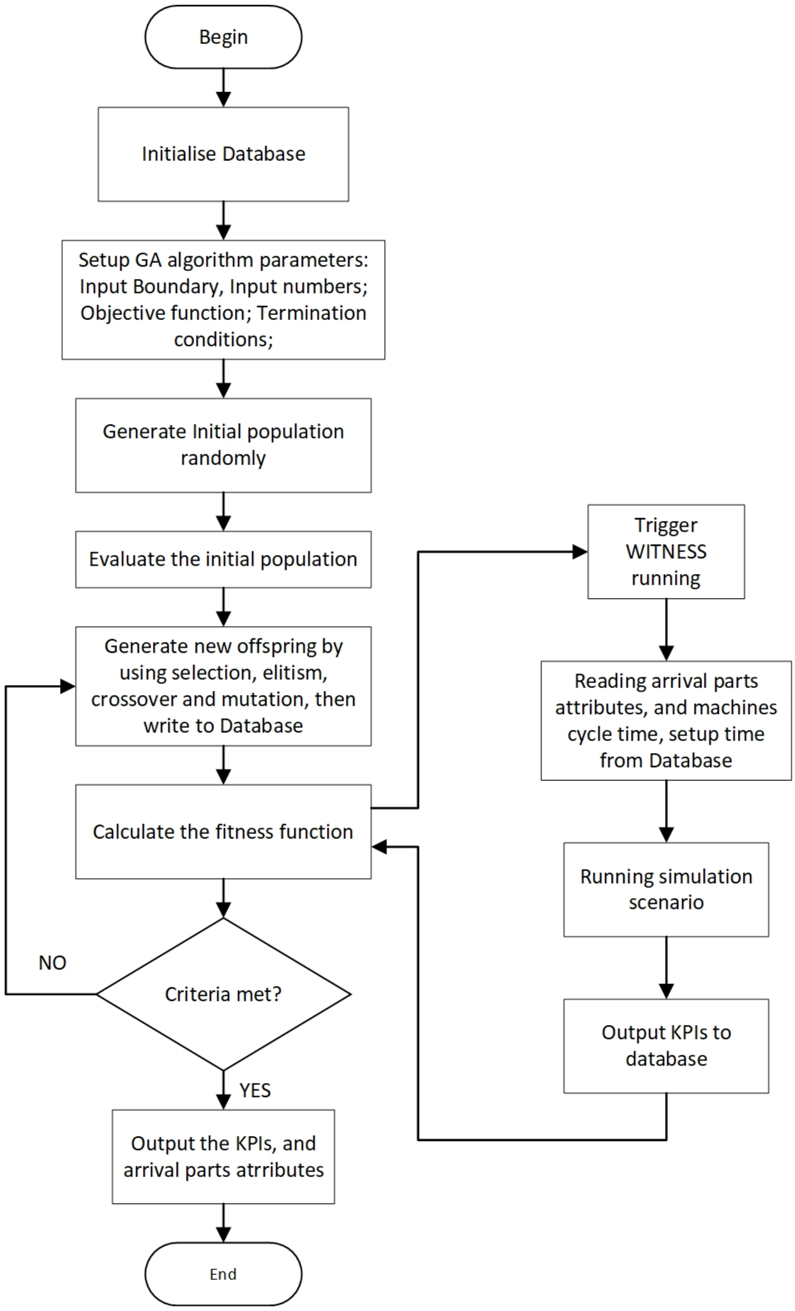

Figure 6. The flow-chart of the approach.

4.4. Genetic Algorithm

This section presents a GA based optimisation method for AGV and machine jobs schedules in FMSs. The GA method is based on the approach proposed in [74]. In the optimisation approach, fitness function is considered to include: shop-floor processing time, AGV energy consumptions, and machine utilisations mainly derived from the DES simulation. First, a group of initial population is created by the GA algorithm, which are then evaluated through the fitness functions. Following this, a new generation population is created through the selection, crossover, and mutation processes, in which the elitists of current generation are passed to the next populations. The manufacturing processes KPIs: just-in-time performance and cumulative AGV energy consumption are defined as objectives to be improved. The algorithm also stops when the maximum number of generations or number of stall generations are reached. The detail of the GA based optimisation method's pseudo code is shown in Algorithm 1.

Algorithm 1 Genetic Algorithm pseudo-code.

Pseudo-code of the GA

```
Initialise the populations;
Evaluate the initial population through fitness function;
for (iteration < MaxIteration) do
    while (not meet the stopping criteria) do
        Select the elitists for next generation;
        Crossover
        Mutation
    end while
    Evaluate the new population fitness;
end for
Output the best solutions;
```

4.4.1. Initialising Parameter

In this article, each generation is separated into two segments representing the product sequence and AGV distribution strategies. Figure 7 shows the population structures of two examples. The first example includes a system consisting of three products and four work stages, each having four identical workstations performing operations for three different arrival products, whereas the second example consists of three work stages, five production jobs, and four identical workstations in each stage. The left-hand side in Figure 7, the encoding rule represents non-integer optimisation parameters that are used to define the product sequence to be released from the warehouse. According to this rule, the product sequence is determined based on weighted cumulative cycle times of product variants. This is characterised by cycle times of each product variant at each machine stage and corresponding machine stage weight coefficients. The right-hand side represents the AGV task distribution sequence to be followed by AGVs. This dictates AGVs to transport materials from one stage to another by following the encoding rule.

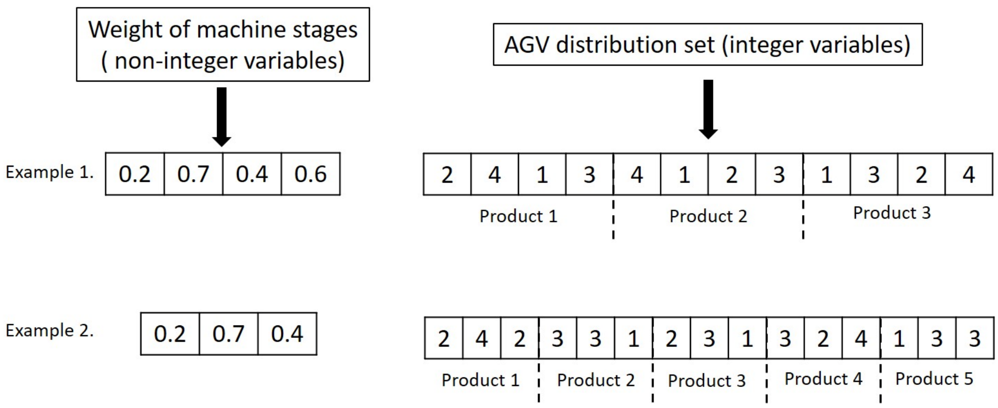

Figure 7. Two examples of the population structure.

4.4.2. Initialising Population

The initial population is generated based on the uniform random generator. The first part of variables is in the range of 0 to 1, and the size of their population is considered to be equal with the number of machine stages. In the second part, the size is taken as equal to the product of machine stages number and arrival products number, and the values are limited by the station number in each stage. Therefore, the lower bounds, upper bounds, the number of variables, and the list of integer values are set up to meet these constraints.

4.4.3. The New Generate Population Generating

The new generations are produced by using selection, elitism, crossover, and mutation.

- Selection: The stochastic universal selection strategy (see [75]) is used to select parents for producing the next generators. In the stochastic uniform selection, all parents are laid on a line. The algorithm follows the line, and moves to the next point at an equal step size. At each movement, the algorithm chooses the current point as the parent for the next generation. The first step is also a uniform random number, which is smaller than the step size.
- Elitism : All the individuals are sorted based on the fitness values. The first N_e (Equation(10)) best individuals are chosen and passed to the next generation directly. This step guarantees that the best fitness values can survive in the next generation:

$$N_e = 5\% * PopulationSize \tag{10}$$

- Crossover: Crossover is generated by combining the two parents together. The genes from parents are chosen randomly for crossover, and genes coordinates are the same for both parents, and the crossover children population is specified by the crossover fraction P_c. These rules are applied into both parts of parents. Figure 8 shows an example of crossover strategy.

Figure 8. An example of crossover strategy.

- Mutation: Mutation is also an important way to create the next generation in GA for genes diversity. The algorithm generates the mutation children from the parents' genes by choosing a random number from the Gaussian distribution (see [76]). An example of mutation is demonstrated in Figure 9.

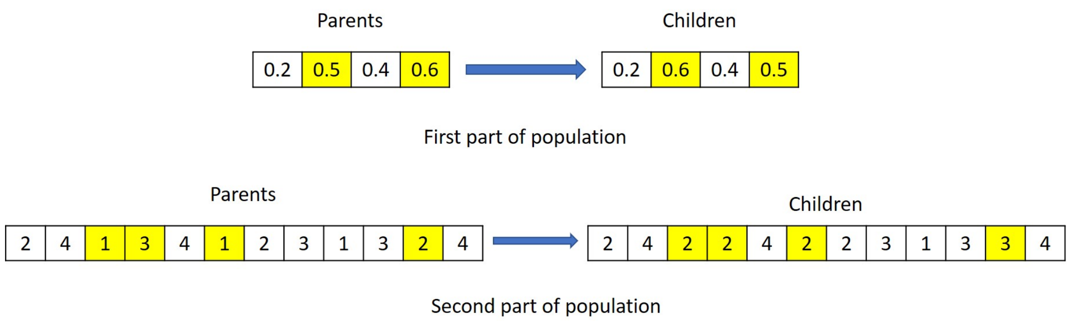

Figure 9. An example of mutation strategy.

4.4.4. Evaluation and Iteration

The current generation population is evaluated by the fitness function. The iteration of creating new generations is terminated once the fitness performance meets the requirement, or the iteration number reaches the maximum iteration limits.

5. Case Study

The case study is implemented in the IML demonstrator at the University of Warwick. IML is designed as a discrete-part automation system assembling battery-packs for electric vehicles. The battery assembly process includes customised battery packs from a single battery cell, such as: 18650, 26650. IML deploys a variety of legacy and agile systems—a traditional conveyor based system represents traditional cellular manufacturing practice [77], while autonomous stations, connected by AGVs for battery pack welding and vision based inspection, represent an Industry 4.0 based example of responsive manufacturing. Figure 10 shows a section of the IML rig. The case study shows the optimisation methodology to improve the manufacturing performance of battery assembly process.

Figure 10. An example material transportation within the IML rig: An AGV is carrying battery cells to the Legacy Loop Assembly Machine in the Stage One where battery modules are assembled.

5.1. Overview of the Experiments

The case study describes a battery assembly process based on the IML demonstrator prototype. The production is modelled and simulated via a DES model. The model's input values are fed by the proposed optimisation model, and the predicted of KPI values are served as feedback, thereby indicating a closed-loop system for improving the battery assemble process JIT performance. The assembly system is separated into four stages, including the Legacy Loop Assembly stage, Welding Stage, Inspection Stage, and Packing Stage. Cycle time and machine tool changing time are predefined from the historical data from the IML demonstrator. In addition, AGV speed, AGV charging time, and AGV running time are assigned as AGV attributes based on the MiR100 servicing in the IML demonstrator. In stage One and Three, the battery cell insertion and nut assembly operations are carried out, respectively. These operations require raw materials such as module baskets and nuts.

Stages Two and Four perform welding and inspection processes, respectively. Materials between each stage are delivered and collected by AGVs. Customer orders are recorded by a web-based products order system. Once an order is issued, this information will be published to the OPC-UA server. In the OPC-UA server, the data from IML demonstrator rig, e.g., PLC registers and I/O, buffer sensors status, and product RFIDs information are recorded. When an order arrives, the decision support system optimises the arrival product sequence and AGV schedules. Once the system finishes the optimisation process, it will broadcast a list of optimised solutions and corresponding production KPIs on the system HMI which can be accessed by system managers or operators to manually choose the proper solution. As soon as a solution is selected, the decision support system will pass this information to the MES application, written in C language, to assign the defined task to corresponding working stations and MiR fleet manager.

In the experiment, 30 jobs are designed to be processed. These jobs are separated into 20 different categories. Each job has four processes, and each stage of the process has four parallel machines. In the experiments, the simulation run-time is set as 25,000 s. Please note that the simulation is forced to terminate when the time runs out, and KPI values will not be recorded. The DES model and embedded GA-based optimisation algorithm are concurrently run to find the Pareto-optimal design space. There are two stopping criteria for GA: (*i*) stop by reaching the maximum number of generations (1000) and (*ii*) stop by max stall generations (30). Moreover, the production target time is set as 4 h per shift, including 3.75 h (135,000 s) processing time and 0.25-h break time. The input parameters are separated into two parts: the first four indicate non-integer parameters, i.e., weights of each machine stage, which the arrival parts sequence can be derived from; the rest of 120 integer values are AGV distribution rules for every arriving part. They are converted as arrival parts attributions and transfer to the WITNESS simulation model. Moreover, the production KPIs are collected as outputs to the optimisation model.

To evaluate the re-scheduling performance of the proposed framework, machine breakdown scenarios were also introduced. In those experiments, after 4000 s of overall process time, two machines were intentionally shut down and process stops. The SAMS is expected to detect the abnormality by solely monitoring the PLC status and resources cycle times. Once the fault information is received by the SAMS, the re-schedule procedure starts. This process involves updating the DES model, executing the simulation for the remaining tasks, and re-allocates the tasks between system resources as soon as a re-schedule among the solution set is approved. After 40 min (2400 s) of hypothetical repair time, the broken machines were back to operation. The SAMS initiates a second re-scheduling process and feeds the new set of solutions into MES application for the approval.

The following assumptions were made during the experiments:

- The shop-floor layout and AGV routing paths were fixed.
- Charging threshold for AGV is set at 20%. If the battery level is lower than 20%, the AGV needs to park at the charging station for re-charge. When AGV battery is fully charged, it will be ready for the new task.

In the experiments, the initial machine parameters, including: setup time and cycle time, and AGV average speeds, non-stop travelling time, and charging time, are collected from the shop-floor through IoT-enabled data collection devices. For instance, the RFID tags are used for tracking battery pallets and calculating the commuting time between each station, energy monitors are attached at each workstation to collect the energy consumption, and PLC function blocks are programmed to calculate machines and robots cycle times. In addition, these data are fed into the OPC UA server through Modbus TCP/IP protocol. The DES model is implemented in WITNESS software, and the optimisation engine is achieved through MATLAB programming language. The experiments presented here are deployed on a PC with Intel(R) Xeon(R) with a 32 GB RAM and I7 8-core 3.8 GHz processors. Please note that the average time for each process simulation within the DES environment is recorded as 5 ± 1 s. Based on the experiments, the GA converges around 200 simulation runs. This indicates a total scheduling optimisation run about 11 ± 3 min (including decision-support and communication with MES). Please note that this is based on the experiments we carried out with simple machine breakdown scenarios at the IML.

Figure 11 shows the histograms of the job processing times for selected machine operations. A uniform normal distribution is selected to represent these job processing times based on the data stored in the time-series database:

$$PT \sim \mathcal{N}(\mu, \sigma^2)\,. \tag{11}$$

where the μ means the average processing time(PT), and σ means the standard deviation of these collected processing time. The μ and σ changes with different jobs. In this case study, both parameters (μ, σ) for each job are analysed, and then updated in the DES software. Please note that the time-series database includes more than 5000 sampling points for each operation.

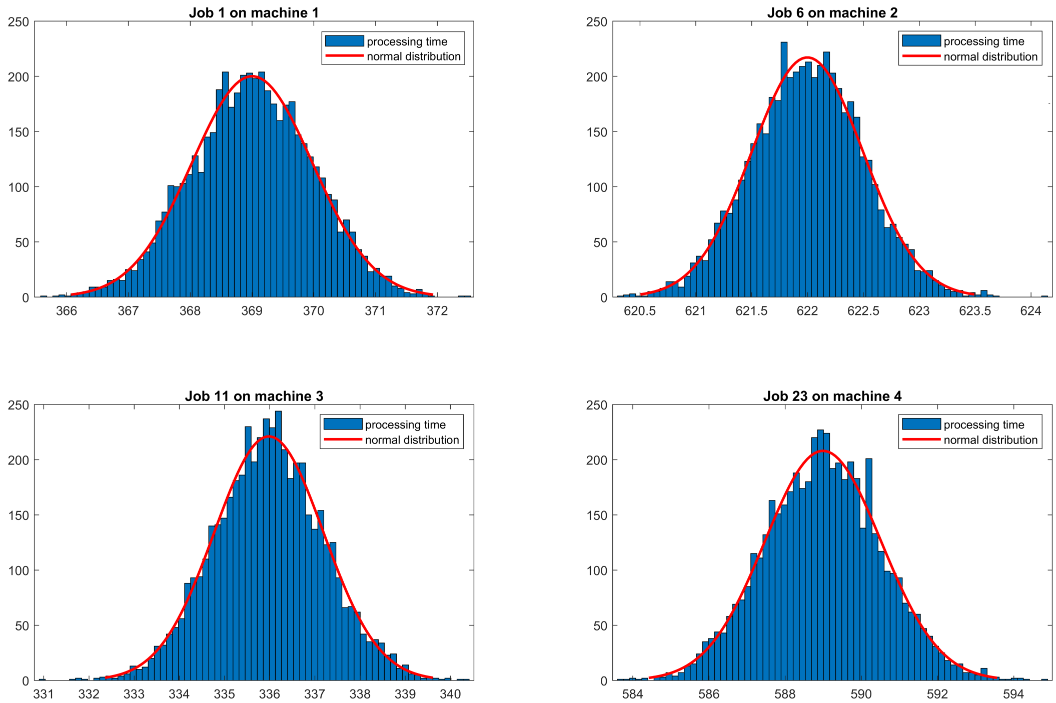

Figure 11. Example of job processing time distribution.

5.2. Results

The result for static job scheduling problem is given in Figure 12, showing the relationship of the tardiness of the material delivery and the average AGV energy consumption. It has been found that AGV energy consumption and JIT material delivery performance are two conflicting outputs. Hence, an optimal scheduling strategy is required. In this research, the relative Euler distance method is chosen to find the near-optimal solutions for AGV and machine jobs scheduling:

$$Dis(f_1, f_2) = \left(\frac{(f_x^1 - f_{min}^1)}{(f_{max}^1 - f_{min}^1)}\right)^2 + \left(\frac{(f_x^2 - f_{min}^2)}{(f_{max}^2 - f_{min}^2)}\right)^2 \tag{12}$$

In the equation given above, the $Dis(f_1, f_2)$ represents the Euler distance between two objective functions, and the minimum value is considered as the best solution in this paper. f_{max}^1 and f_{min}^1 represent the minimum and maximum value of 1st objective function, respectively, and f_{max}^2 and f_{min}^2 represent the minimum and maximum value of 2nd objective function. Once the solution parameters for the Euler distance are set, the best solution for machine jobs schedule and AGV distribution rules can be attained. In this way, multiple solutions can be provided based on different KPIs requirements, including AGV blocking time, machines utilisation balance, and parts waiting time in the buffer, etc. Figure 13 depicts the Gantt chart for the best solution including both machine and AGV schedules for static job scheduling experiments.

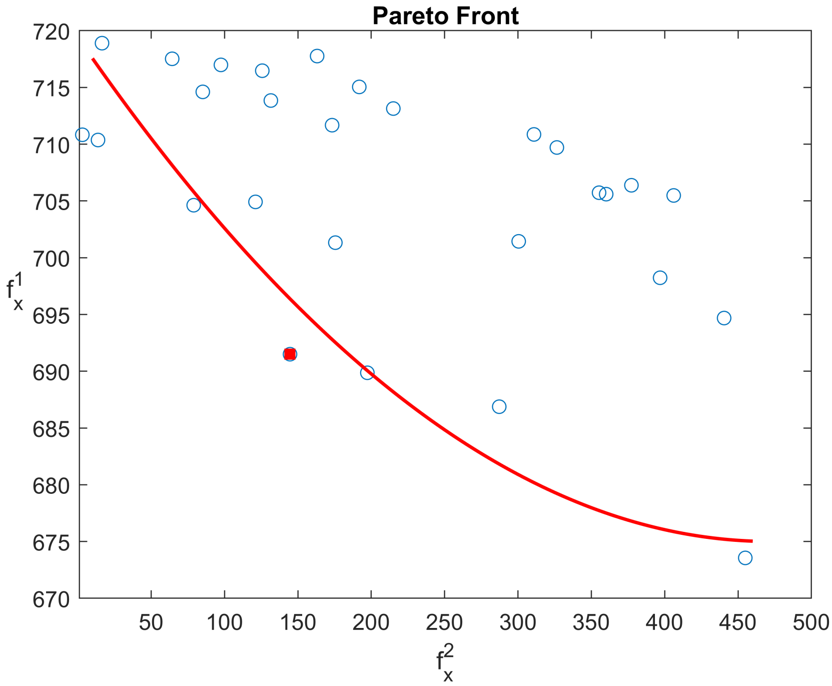

Figure 12. The Pareto Front.

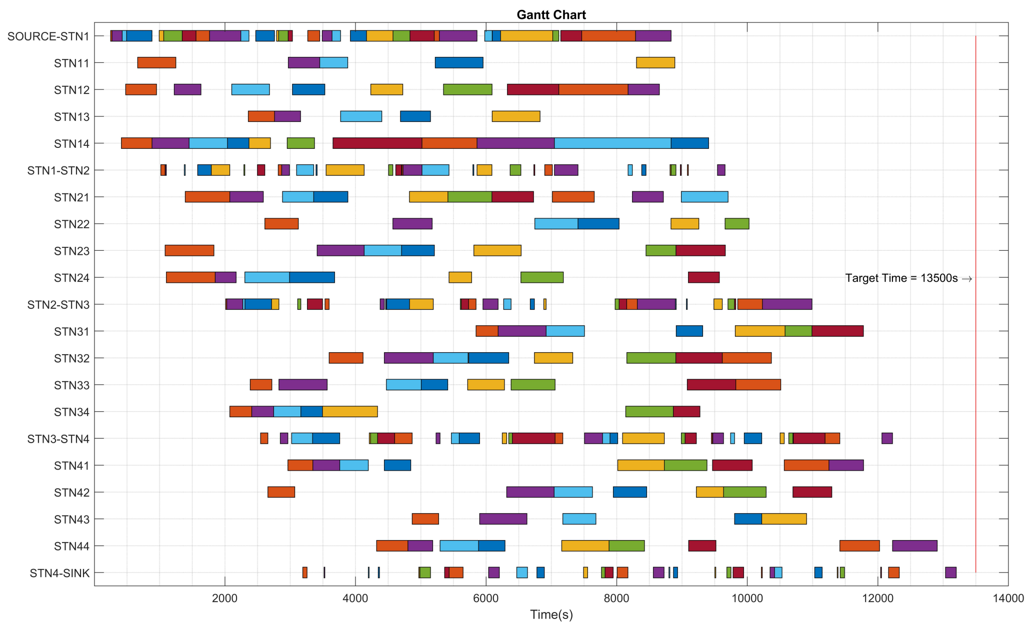

Figure 13. Gantt Chart (Normal/planned events).

To evaluate the efficiency of the proposed system, its performance is compared with a static First-In-First-Out (FIFO) and five Shortest Processing Time (SPT) based dispatching methods. In the FIFO-based dispatching approach, the first arrival product is delivered to the nearest machine, and be processed first. On the other hand, SPT-based method prioritises the product with the shortest processing time. Here, five different SPT-based scheduling rules, i.e., the SPT based on the cycle-time for each stage and the SPT based on the overall product cycle-time. The performance comparison is given in Table 3. According to the results, a large tardiness improvement is recorded for the proposed approach. It is also noted that a slight increase in AGV energy consumption (EC) performance is achieved.

Table 3. Comparison of the implemented scheduling approaches.

Solutions	**Normal Events**		**Two Machines Breakdown**	
	Tardiness	**EC**	**Tardiness**	**EC**
Proposed Scheduling	300.4484 (Earliness)	701.4404	218.6914 (Earliness)	704.9327
FIFO Scheduling	1575.7169 (Earliness)	577.4241	4657.8487 (Lateness)	701.1848
SPT based on 1st Stage	1103.9 (Earliness)	585.7565	14,968 (Lateness)	997.0096
SPT based on 2nd Stage	679.377 (Earliness)	607.6007	13,708 (Lateness)	923.2472
SPT based on 3rd Stage	1223.6 (Earliness)	612.8167	9681.7 (Lateness)	833.9795
SPT based on 4th Stage	1179.2 (Earliness)	613.1612	15,750 (Lateness)	1150.4
SPT based on overall Stage	1710.9 (Earliness)	576.6980	13,956 (Lateness)	944.900

Two production scenarios with manufacturing disruptions are also set up to evaluate the re-scheduling capability of the proposed approach. In these scenarios, the fourth machines in Stage 2 and Stage 3 are intentionally broken down. The breakdown is set from 4000 s to 6400 s, lasting for 40 min. Meanwhile, the re-scheduling strategies are generated by the SAMS to meet the JIT requirements with an acceptable AGV energy consumption rate. The results (Table 3) showed that the SPT and FIFO-based methods are unable to handle manufacturing interruptions, although they are capable of providing acceptable performance under normal operational conditions. The proposed approach is able to effectively re-schedule AGV and machine schedules subjected to production abnormalities, and provides a significantly better tardiness performance. Please note that all methods have similar results for AGV energy consumption rates.

5.3. Discussion

With the recent advancement in the Industry 4.0 systems and technologies, the decision-support systems became a vital enabler in ensuring global competitiveness of manufacturing enterprises. In the related literature, there are a few-number of works involving the simulation-based decision-support systems within the context of manufacturing systems engineering. Some examples include: [78–81]. Contrary to exact methods, the simulation-based approaches provide timely decisions due to reduced computational complexity. However, these methods are often criticised due to accuracy problems [82]. In this research, the SAMS architecture is modified to overcome this challenge. To minimise the prediction errors of the static DES models used within the SAMS, IoT-enabled historical data are streamlined into the DES models to enhance their prediction capabilities. In addition to this, an evolutionary optimisation algorithm (i.e., GA) is employed in a multi-objective optimisation problem to deal with the scheduling complexity while avoiding getting trapped in the local minima. The interoperability of the proposed system is demonstrated using OPC-UA industrial M2M communication protocol. Moreover, the decision-support capabilities of the approach are demonstrated on case studies where a set of near-optimal re-scheduling solutions are promptly provided to shop-floor managers upon the event of manufacturing disruption via a human–machine interface. The results showed that the proposed approach can help to improve the performance of the system in terms of just-in-time delivery performance, the average utilisation of the system resources, average queue times, and energy efficiency of AGV transportation.

The Lanner's WITNESS DES software provides an object-oriented modelling approach for AGV material transportation. The model has a pre-defined AGV routing topology that ensures that AGVs do not collide against each other. A deadlock consists of a model state in which the AGVs are simultaneously waiting for any other AGV to perform a task and no AGV can change its current state. Effectively, this locks the model, and prevents the completion of the simulation run. During the initial modelling stages, which involves identifying potential issues, we observed that the possibility of deadlock occurrence when an AGV tries to access to the storage locations. This is because the access to the storage location was done using a bi-directional path with a single unit capacity. This allows only one AGV to cross this path at any time with another AGV waiting on the other end of the path and there is no space for the first AGV to exit. It is important to note that this type of deadlock should be avoided during the simulation. To prevent this issue, we introduced two unidirectional paths across the routing topology. This, in fact, can be considered as a crude simplification of the real system. However, since the IML demonstrator under consideration has a very low number of AGVs, this situation rarely occurs in the real system. Therefore, it is assumed that the addition of two unidirectional paths in the simulation model has a negligible impact on the results. Please note that, for systems with complex layouts and/or a high number of AGVs, more sophisticated deadlock prevention algorithms and mechanisms should be employed. Some examples include: [83–87].

The proposed approach, however, has certain limitations that need to be addressed. Firstly, in its current form, the SAMS operates with limited data. As future work, to fully exploit the advantages of the concept of Big Data Analytics, more IoT-enabled data will be streamlined into the SAMS and a complex event processing engine will be employed to process those streams. This will provide a better understanding of the relationships among various shop-floor activities and will help to improve the predictive analytics capabilities of the approach. Another important limitation is the prediction errors arising due to the real-time behaviours of AGVs. The proposed SAMS provides a set of scheduling alternatives based on the simulation optimisation results. The selected schedule and corresponding AGV job assignments are then fed to the MES and further MiR fleet manager. The MiR fleet manager is an industrial control system for AGVs providing a collision-free routing with shortest travel times. The fleet manager assigns tasks to AGVs depending on their location, energy levels, etc. This manager has an in-built traffic control mechanism offering the coordination of critical zones with multiple robot intersections and hence providing a collision free routing. Additionally, MiR AGVs have collision sensors and in-built cartographer SLAM algorithms to prevent any real-time collision issues. AGVs can autonomously decide and manoeuvre outside of their pre-defined path to avoid any type of collisions. It is important to note that there might be differences in the AGV path since the WITNESS models have pre-defined routes unlike the MiR fleet manager. In the experiments, we observed a difference between completion time of shop-floor jobs and DES simulation results (up to 7.1%) because of logistics uncertainties. This limitation of the SAMS will be addressed as future work by employing a better information-mirroring mechanism between cyber and physical domains. The graphical user interface used in the SAMS decision-support system only broadcasts a list of solutions to be selected on the HMI screens. As future work, a new dashboard with varying visualisation options will be developed to provide a better decision-support to shop-floor decision-makers. Lastly, the communication between the proposed systems and MES and Enterprise Resource Planning (ERP) systems will be enhanced using web-services to provide a more industry-ready deployable solution.

6. Conclusions

In this paper, a decision-support system capable of providing multiple scheduling solutions as a response to manufacturing disruptions was introduced. The system uses IoT-enabled production data to enhance the accuracy of the digital replica of the FMS under consideration. In the event of a manufacturing disruption, the system automatically detects the production anomaly and releases a set of re-scheduling strategies aiming to satisfy both maximised just-in-time delivery performance and minimised AGV energy consumption on time. The system was tested on a real industrial case study, and the results showed that the system is helpful to managers for the decision-making at the operational level.

Author Contributions: Conceptualization, B.A. (Bugra Alkan); methodology, B.A. (Bugra Alkan); software, F.Y.; validation, F.Y.; formal analysis, F.Y.; investigation, F.Y.; resources, F.Y.; data curation, F.Y.; writing—original draft preparation, F.Y. and B.A. (Bugra Alkan); writing—review and editing, B.A. (Bugra Alkan); visualization, F.Y. and B.A. (Bugra Alkan); supervision, B.A. (Bugra Alkan), B.A. (Bilal Ahmad) and R.H.; project administration, B.A. (Bilal Ahmad) and R.H.; funding acquisition, B.A. (Bilal Ahmad) and R.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work presented in the paper is based on foundational work undertaken on the EPSRC funded Knowledge Driven Configurable Manufacturing project. The work is financially supported by Innovate UK through ISCF Manufacturing Made Smarter project DIALOG.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chryssolouris, G. *Manufacturing Systems: Theory and Practice*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
2. Ahmad, M.; Ahmad, B.; Harrison, R.; Alkan, B.; Vera, D.; Meredith, J.; Bindel, A. A framework for automatically realizing assembly sequence changes in a virtual manufacturing environment. *Procedia CIRP* **2016**, *50*, 129–134. [CrossRef]
3. Chinnathai, M.K.; Alkan, B.; Vera, D.; Harrison, R. Pilot to full-scale production: A battery module assembly case study. *Procedia CIRP* **2018**, *72*, 796–801. [CrossRef]
4. Baruwa, O.T.; Piera, M.A. A coloured Petri net-based hybrid heuristic search approach to simultaneous scheduling of machines and automated guided vehicles. *Int. J. Prod. Res.* **2016**, *54*, 4773–4792. [CrossRef]
5. Espinouse, M.L.; Pawlak, G.; Sterna, M. Complexity of scheduling problem in single-machine flexible manufacturing system with cyclic transportation and unlimited buffers. *J. Optim. Theory Appl.* **2017**, *173*, 1042–1054. [CrossRef]
6. Barenji, A.V.; Barenji, R.V.; Roudi, D.; Hashemipour, M. A dynamic multi-agent-based scheduling approach for SMEs. *Int. J. Adv. Manuf. Technol.* **2017**, *89*, 3123–3137. [CrossRef]
7. Alkan, B.; Bullock, S. Assessing operational complexity of manufacturing systems based on algorithmic complexity of key performance indicator time-series. *J. Oper. Res. Soc.* **2020**. [CrossRef]
8. Mehrabian, A.; Tavakkoli-Moghaddam, R.; Khalili-Damaghani, K. Multi-objective routing and scheduling in flexible manufacturing systems under uncertainty. *Iran. J. Fuzzy Syst.* **2017**, *14*, 45–77.
9. Fazlollahtabar, H.; Saidi-Mehrabad, M.; Balakrishnan, J. Mathematical optimization for earliness/tardiness minimization in a multiple automated guided vehicle manufacturing system via integrated heuristic algorithms. *Robt. Autom. Syst.* **2015**, *72*, 131–138. [CrossRef]
10. Behrje, U.; Himstedt, M.; Maehle, E. An Autonomous Forklift with 3D Time-of-Flight Camera-Based Localization and Navigation. In Proceedings of the 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, 18–21 November 2018; pp. 1739–1746.
11. Alkan, B.; Vera, D.A.; Ahmad, M.; Ahmad, B.; Harrison, R. Complexity in manufacturing systems and its measures: a literature review. *Eur. J. Ind. Eng.* **2018**, *12*, 116–150. [CrossRef]
12. Alkan, B.; Vera, D.; Ahmad, M.; Ahmad, B.; Harrison, R. Design evaluation of automated manufacturing processes based on complexity of control logic. *Procedia CIRP* **2016**, *50*, 141–146. [CrossRef]
13. Alkan, B.; Vera, D.; Ahmad, B.; Harrison, R. A Method to Assess Assembly Complexity of Industrial Products in Early Design Phase. *IEEE Access* **2018**, *6*, 989–999. [CrossRef]
14. Zhang, S.; Wong, T.N. Flexible job-shop scheduling/rescheduling in dynamic environment: A hybrid MAS/ACO approach. *Int. J. Prod. Res.* **2017**, *55*, 3173–3196. [CrossRef]
15. Alkan, B.; Harrison, R. A virtual engineering based approach to verify structural complexity of component-based automation systems in early design phase. *J. Manuf. Syst.* **2019**, *53*, 18–31. [CrossRef]
16. Chinnathai, M.K.; Al-Mowafy, Z.; Alkan, B.; Vera, D.; Harrison, R. A Framework for Pilot Line Scale-up using Digital Manufacturing. *Procedia CIRP* **2019**, *81*, 962–967. [CrossRef]

17. Yao, F.; Keller, A.; Ahmad, M.; Ahmad, B.; Harrison, R.; Colombo, A.W. Optimizing the Scheduling of Autonomous Guided Vehicle in a Manufacturing Process. In Proceedings of the 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), Porto, Portugal, 18–20 July 2018; pp. 264–269.
18. Eksioglu, B.; Vural, A.V.; Reisman, A. The vehicle routing problem: A taxonomic review. *Comput. Ind. Eng.* **2009**, *57*, 1472–1483. [CrossRef]
19. Braekers, K.; Ramaekers, K.; Van Nieuwenhuyse, I. The vehicle routing problem: State of the art classification and review. *Comput. Ind. Eng.* **2016**, *99*, 300–313. [CrossRef]
20. Demesure, G.; Defoort, M.; Bekrar, A.; Trentesaux, D.; Djemaï, M. Navigation scheme with priority-based scheduling of mobile agents: Application to AGV-based flexible manufacturing system. *J. Intell. Robt. Syst.* **2016**, *82*, 495–512. [CrossRef]
21. Fontes, D.B.M.; Homayouni, S.M. Joint production and transportation scheduling in flexible manufacturing systems. *J. Glob. Optim.* **2019**, *74*, 879–908. [CrossRef]
22. Fazlollahtabar, H. Scheduling of Multiple Autonomous Guided Vehicles for an Assembly Line using Minimum Cost Network Flow. *J. Glob. Optim.* **2018**, *11*, 185–193.
23. Dang, Q.V.; Nielsen, I. Simultaneous scheduling of machines and mobile robots. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 118–128.
24. Nageswararao, M.; Narayanarao, K.; Ranagajanardhana, G. Simultaneous scheduling of machines and AGVs in flexible manufacturing system with minimization of tardiness criterion. *Procedia Mater. Sci.* **2014**, *5*, 1492–1501. [CrossRef]
25. Huang, B.; Jiang, R.; Zhang, G. Heuristic Search for Scheduling Flexible Manufacturing Systems Using Multiple Heuristic Functions. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 178–187.
26. Sanches, D.S.; da Silva Rocha, J.; Castoldi, M.F.; Morandin, O.; Kato, E.R.R. An adaptive genetic algorithm for production scheduling on manufacturing systems with simultaneous use of machines and agvs. *J. Control Autom. Electr. Syst.* **2015**, *26*, 225–234. [CrossRef]
27. Mousavi, M.; Yap, H.J.; Musa, S.N.; Tahriri, F.; Dawal, S.Z.M. Multi-objective AGV scheduling in an FMS using a hybrid of genetic algorithm and particle swarm optimization. *PLoS ONE* **2017**, *12*, e0169817. [CrossRef] [PubMed]
28. Zhong, M.; Yang, Y.; Dessouky, Y.; Postolache, O. Multi-AGV scheduling for conflict-free path planning in automated container terminals. *Comput. Ind. Eng.* **2020**, *142*, 106371. [CrossRef]
29. Rahman, H.F.; Janardhanan, M.N.; Nielsen, P. An integrated approach for line balancing and AGV scheduling towards smart assembly systems. *Assembly Autom.* **2020**, *40*, 219–234. [CrossRef]
30. Wang, C.; Jin, C.; Li, Z. Bilevel programming model of low energy consumption AGV scheduling problem at automated container terminal. In Proceedings of the 2019 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE), Hangzhou, China, 20–21 April 2019; pp. 195–199.
31. Liu, Y.; Ji, S.; Su, Z.; Guo, D. Multi-objective AGV scheduling in an automatic sorting system of an unmanned (intelligent) warehouse by using two adaptive genetic algorithms and a multi-adaptive genetic algorithm. *PLoS ONE* **2019**, *14*, e0226161. [CrossRef] [PubMed]
32. Weyns, D.; Boucké, N.; Holvoet, T.; Demarsin, B. Dyncnet: A protocol for dynamic task assignment in multiagent systems. In Proceedings of the First International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2007), Cambridge, MA, USA, 9–11 July 2007; pp. 281–284.
33. Smith, R.G. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Trans. Comput.* **1980**, *C-29*, 1104–1113. [CrossRef]
34. Chan, F.T.; Chan, H.K.; Kazerooni, A. Real time fuzzy scheduling rules in FMS. *J. Intell. Manuf.* **2003**, *14*, 341–350. [CrossRef]
35. Wang, J.; Zhang, Y.; Liu, Y.; Wu, N. Multiagent and Bargaining-Game-Based Real-Time Scheduling for Internet of Things-Enabled Flexible Job Shop. *IEEE Internet Things J.* **2018**, *6*, 2518–2531. [CrossRef]
36. Zhang, Y.; Zhu, Z.; Lv, J. CPS-based smart control model for shopfloor material handling. *IEEE Trans. Ind. Inf.* **2017**, *14*, 1764–1775. [CrossRef]
37. Xue, T.; Zeng, P.; Yu, H. A reinforcement learning method for multi-AGV scheduling in manufacturing. In Proceedings of the 2018 IEEE International Conference on Industrial Technology (ICIT), Lyon, France, 20–22 February 2018.

38. Zhao, X.; Liu, H.; Lin, S.; Chen, Y. Design and implementation of a multiple agv scheduling algorithm for a job-shop. *Int. J. Simul. Modell. (IJSIMM)* **2020**, *19*, 134–145. [CrossRef]
39. Sahin, C.; Demirtas, M.; Erol, R.; Baykasoğlu, A.; Kaplanoğlu, V. A multi-agent based approach to dynamic scheduling with flexible processing capabilities. *J. Intell. Manuf.* **2017**, *28*, 1827–1845. [CrossRef]
40. Padgham, L.; Winikoff, M. *Developing Intelligent Agent Systems: A Practical Guide*; John Wiley & Sons: Hoboken, NJ, USA, 2005; Volume 13.
41. Xu, W.; Guo, S.; Li, X.; Guo, C.; Wu, R.; Peng, Z. A Dynamic Scheduling Method for Logistics Tasks Oriented to Intelligent Manufacturing Workshop. *Math. Prob. Eng.* **2019**, *2019*. [CrossRef]
42. Lenstra, J.K.; Kan, A.R. Complexity of vehicle routing and scheduling problems. *Networks* **1981**, *11*, 221–227. [CrossRef]
43. Defoort, M.; Kokosy, A.; Floquet, T.; Perruquetti, W.; Palos, J. Motion planning for cooperative unicycle-type mobile robots with limited sensing ranges: A distributed receding horizon approach. *Robt. Autom. Syst.* **2009**, *57*, 1094–1106. [CrossRef]
44. Corréa, A.I.; Rousseau, L.M.; Langevin, A. *A Scheduling and Conflict-Free Routing Problem Solved with a Hybrid Constraint Programming: Mixed Integer Programming Approach*; Groupe D'études et de Recherche en Analyse des Décisions, Springer: Berlin/Heidelberg, Germany, 2005.
45. Oboth, C.; Batta, R.; Karwan, M. Dynamic conflict-free routing of automated guided vehicles. *Int. J. Prod. Res.* **1999**, *37*, 2003–2030. [CrossRef]
46. Jin, J.; Zhang, X.H. Multi agv scheduling problem in automated container terminal. *J. Mar. Sci. Technol.* **2016**, *24*, 32–38.
47. Demesure, G.; Defoort, M.; Bekrar, A.; Trentesaux, D.; Djemai, M. Decentralized motion planning and scheduling of AGVs in an FMS. *IEEE Trans. Ind. Inf.* **2017**, *14*, 1744–1752. [CrossRef]
48. Nouri, H.E.; Driss, O.B.; Ghédira, K. Simultaneous Scheduling of Machines and a Single Moving Robot in a Job Shop Environment by Metaheuristics based Clustered Holonic Multiagent Model. In Proceedings of the 8th International Conference on Agents and Artificial Intelligence ICAART, Rome, Italy, 24–26 February 2016; pp. 51–62.
49. He, Y.; Stecke, K.E.; Smith, M.L. Robot and machine scheduling with state-dependent part input sequencing in flexible manufacturing systems. *Int. J. Prod. Res.* **2016**, *54*, 6736–6746. [CrossRef]
50. Mousavi, M.; Yap, H.; Musa, S.; Dawal, S. A fuzzy hybrid GA-PSO algorithm for multi-objective AGV scheduling in FMS. *Int. J. Simul. Modell.* **2017**, *16*, 58–71. [CrossRef]
51. Li, G.; Zeng, B.; Liao, W.; Li, X.; Gao, L. A new AGV scheduling algorithm based on harmony search for material transfer in a real-world manufacturing system. *Adv. Mech. Eng.* **2018**, *10*. [CrossRef]
52. Zou, W.Q.; Pan, Q.K.; Tasgetiren, M.F. An Effective Discrete Artificial Bee Colony Algorithm for Scheduling an Automatic-Guided-Vehicle in a Linear Manufacturing Workshop. *IEEE Access* **2020**, *8*, 35063–35076. [CrossRef]
53. Zhang, F.; Li, J. An Improved Particle Swarm Optimization Algorithm for Integrated Scheduling Model in AGV-Served Manufacturing Systems. *J. Adv. Manuf. Syst.* **2018**, *17*, 375–390. [CrossRef]
54. Chawla, V.; Chanda, A.; Angra, S. Scheduling of multi load AGVs in FMS by modified memetic particle swarm optimization algorithm. *J. Project Manag.* **2018**, *3*, 39–54. [CrossRef]
55. Chen, C.; Tiong, L.K.; Chen, I.M. Using a genetic algorithm to schedule the space-constrained AGV-based prefabricated bathroom units manufacturing system. *Int. J. Prod. Res.* **2019**, *57*, 3003–3019. [CrossRef]
56. Riazi, S.; Diding, T.; Falkman, P.; Bengtsson, K.; Lennartson, B. Scheduling and Routing of AGVs for Large-scale Flexible Manufacturing Systems. In Proceedings of the 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE), Vancouver, BC, Canada, 22–26 August 2019; pp. 891–896.
57. Hooker, J.N. Planning and scheduling by logic-based Benders decomposition. *Oper. Res.* **2007**, *55*, 588–602. [CrossRef]
58. Eckhardt, A.; Müller, S.; Leurs, L. An evaluation of the applicability of OPC UA Publish Subscribe on factory automation use cases. In Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Turin, Italy, 4–7 September 2018; Volume 1, pp. 1071–1074.
59. Ioana, A.; Korodi, A. OPC UA Publish-Subscribe and VSOME/IP Notify-Subscribe Based Gateway Application in the Context of Car to Infrastructure Communication. *Sensors* **2020**, *20*, 4624. [CrossRef]
60. Chryssolouris, G.; Efthymiou, K.; Papakostas, N.; Mourtzis, D.; Pagoropoulos, A. Flexibility and complexity: Is it a trade-off? *Int. J. Prod. Res.* **2013**, *51*, 6788–6802. [CrossRef]

61. Lanner Group. WITNESS Simulation Software. 2003. Available online: https://www.addlink.es/images/pdf/WITNESS_Simulation_Software_Paper_lr.pdf (accessed on 10 September 2020).
62. Barlas, P.; Heavey, C. Automation of input data to discrete event simulation for manufacturing: A review. *Int. J. Model. Simul. Sci. Comput.* **2016**, *7*, 1630001. [CrossRef]
63. Gore Kumkum, A.; Jalwadi, S.; Natarajan, S. Systematic Efficiency Improvement by Optimizing the Assembly Line Using Witness Simulation Software. Available online: https://journalnx.com/papers/20150547-optimizing-assembly-line.pdf (accessed on 10 September 2020).
64. Mensah, P.; Merkuryev, Y.; Longo, F. Using ICT in developing a resilient supply chain strategy. *Procedia Comput. Sci.* **2015**, *43*, 101–108. [CrossRef]
65. Vitanov, I.V.; Vitanov, V.I.; Harrison, D.K. Buffer capacity allocation using ant colony optimisation algorithm. In Proceedings of the 2009 Winter Simulation Conference (WSC), Austin, TX, USA, 13–16 December 2009; pp. 3158–3168.
66. Anbumalar, V.; Raja Chandra Sekar, M. Methods for solving cell formation, static layout and dynamic layout cellular manufacturing system problems: A review. *Asian Int. J. Sci. Technol.* **2015**, *6*, 2107–2112.
67. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*; MIT Press: Cambridge, MA, USA, 1992.
68. Jerald, J.; Asokan, P.; Saravanan, R.; Rani, A.D.C. Simultaneous scheduling of parts and automated guided vehicles in an FMS environment using adaptive genetic algorithm. *Int. J. Adv. Manuf. Technol.* **2006**, *29*, 584–589. [CrossRef]
69. Choi, H.R.; Park, B.K.; Lee, J.; Park, C. Dispatching of container trucks using genetic algorithm. In Proceedings of the 4th International Conference on Interaction Sciences, Busan, Korea, 16–18 August 2011; pp. 146–151.
70. Udhayakumar, P.; Kumanan, S. Task scheduling of AGV in FMS using non-traditional optimization techniques. *Int. J. Simul. Modell.* **2010**, *9*, 28–39. [CrossRef]
71. Umar, U.A.; Ariffin, M.; Ismail, N.; Tang, S. Hybrid multiobjective genetic algorithms for integrated dynamic scheduling and routing of jobs and automated-guided vehicle (AGV) in flexible manufacturing systems (FMS) environment. *Int. J. Adv. Manuf. Technol.* **2015**, *81*, 2123–2141. [CrossRef]
72. Grunow, M.; Günther, H.O.; Lehmann, M. Dispatching multi-load AGVs in highly automated seaport container terminals. In *Container Terminals and Automated Transport Systems*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 231–255.
73. Bose, J.; Reiners, T.; Steenken, D.; Voß, S. Vehicle dispatching at seaport container terminals using evolutionary algorithms. In Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, Maui, HI, USA, 7 January 2000; p. 10.
74. Deep, K.; Singh, K.P.; Kansal, M.L.; Mohan, C. A real coded genetic algorithm for solving integer and mixed integer optimization problems. *Appl. Math. Comput.* **2009**, *212*, 505–518. [CrossRef]
75. Baldovino, R.G.; Bautista, M.G.A.C.; Aquino, A.U.; Calilung, E.J.; Sybingco, E.; Dadios, E.P. GA Optimization of coconut sugar cooking process: a preliminary study using stochastic universal sampling (SUS) technique. In Proceedings of the 9th International Conference on Computer and Automation Engineering, Sydney, Australia, 18–21 February 2017; pp. 346–349.
76. Cicirello, V. Genetic algorithm parameter control: Application to scheduling with sequence-dependent setups. In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), New York, NY, USA, 3–5 December 2015; pp. 136–143.
77. Ferrari, G.; Corinna, A. Application of an AGV System for Material Handling to a Cellular Manufacturing Environment: The Valeo Case. Master's Thesis, Politecnico di Torino, Torino, Italy, 2018.
78. Vafeiadis, T.; Kalatzis, D.; Nizamis, A.; Ioannidis, D.; Apostolou, K.; Metaxa, I.; Charisi, V.; Beecks, C.; Insolvibile, G.; Pardi, M.; et al. Data analysis and visualization framework in the manufacturing decision support system of COMPOSITION project. *Procedia Manuf.* **2019**, *28*, 57–62. [CrossRef]
79. Sadeghian, R.; Sadeghian, M.R. A decision support system based on artificial neural network and fuzzy analytic network process for selection of machine tools in a flexible manufacturing system. *Int. J. Adv. Manuf. Technol.* **2016**, *82*, 1795–1803. [CrossRef]
80. Salama, S.; Eltawil, A.B. A decision support system architecture based on simulation optimization for cyber-physical systems. *Procedia Manuf.* **2018**, *26*, 1147–1158. [CrossRef]

81. Karlsson, I.; Bernedixen, J.; Ng, A.H.; Pehrsson, L. Combining augmented reality and simulation-based optimization for decision support in manufacturing. In Proceedings of the 2017 Winter Simulation Conference (WSC), Las Vegas, NV, USA, 3–6 December 2017; pp. 3988–3999.
82. Prajapat, N.; Tiwari, A. A review of assembly optimisation applications using discrete event simulation. *Int. J. Comput. Integr. Manuf.* **2017**, *30*, 215–228. [CrossRef]
83. Wu, N.; Zhou, M. Modeling and deadlock avoidance of automated manufacturing systems with multiple automated guided vehicles. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2005**, *35*, 1193–1202. [CrossRef]
84. Fanti, M.P.; Zhou, M. Deadlock control methods in automated manufacturing systems. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2004**, *34*, 5–22. [CrossRef]
85. Fanti, M.P. Event-based controller to avoid deadlock and collisions in zone-control AGVS. *Int. J. Prod. Res.* **2002**, *40*, 1453–1478. [CrossRef]
86. Luo, J.; Wan, Y.; Wu, W.; Li, Z. Optimal Petri-Net Controller for Avoiding Collisions in a Class of Automated Guided Vehicle Systems. *IEEE Trans. Intell. Transp. Syst.* **2019**. [CrossRef]
87. Banaszak, Z.A.; Krogh, B.H. Deadlock avoidance in flexible manufacturing systems with concurrently competing process flows. *IEEE Trans. Robt. Autom.* **1990**, *6*, 724–734. [CrossRef]