applied sciences

MDPI

Article

Simulation and Optimization of an Intelligent Transport System Based on Freely Moving Automated Guided Vehicles

Ladislav Rigó [1], Jana Fabianová [1,*], Ján Palinský [1] and Iveta Dočkalíková [2]

[1] Institute of Logistics and Transport, Faculty of Mining, Ecology, Process Control and Geotechnology, Technical University of Kosice, Letná 9, 042 00 Košice, Slovakia; ladislav.rigo@tuke.sk (L.R.); jan.palinsky@student.tuke.sk (J.P.)
[2] Department of Bachelor's Studies, College of Logistics, Palackého 1381/25, 750 02 Přerov, Czech Republic; iveta.dockalikova@vslg.cz
* Correspondence: jana.fabianova@tuke.sk

Abstract: AGV-based intra-company transport systems are indispensable in the manufacturing industry of Industry 4.0. Designing the systems involves determining AGV movement paths that are predefined dynamically or adjusted based on real-time events. This study focuses on the simulation and optimization of an intelligent transport system. The aim is to create a system model with freely moving AGVs controlled based on the requirements of production facilities. The simulation model was designed in the Tecnomatix Plant Simulation environment. A fictional case study with a flexible manufacturing system was used. Specific methods have been developed for AGV operation, control, and dynamic product handling. The initial simulation model served as the basis for optimization. Model optimization, performed using a genetic algorithm, aimed to maximize production volume while minimizing the number of AGVs. Simulation results showed that AGV movements were dynamically adjusted based on real-time machine requests, and the optimal configuration of AGVs achieved a production volume that was significantly higher than the initial setup. This study demonstrates a new approach to modeling AGV traffic systems emphasizing real-time dynamic adjustments of AGV paths. The findings contribute to integrating intelligent transport systems into production processes, and this study provides valuable insights for future investigation in this area.

Keywords: simulation; optimization; freely moving AGV; genetic algorithm

check for updates

Citation: Rigó, L.; Fabianová, J.; Palinský, J.; Dočkalíková, I. Simulation and Optimization of an Intelligent Transport System Based on Freely Moving Automated Guided Vehicles. *Appl. Sci.* **2024**, *14*, 7937. https://doi.org/10.3390/app14177937

Academic Editor: José Machado

Received: 29 July 2024
Revised: 25 August 2024
Accepted: 2 September 2024
Published: 5 September 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Intelligent Logistics Systems (ILSs) play a crucial role in the revolution of Industry 4.0 by leveraging advanced technologies to enhance supply chain management and logistics operations. Logistics operations are cardinal in manufacturing systems. Today, designing production systems is practically impossible without using computer simulation. Simulation techniques have proven themselves to be one of the most practical methodologies available to investigate and evaluate manufacturing issues [1]. Simulation techniques provide a virtual environment to model, analyze, and optimize production systems before implementing changes in the real world. Additionally, production planning and control play a key role in enabling manufacturers to gain visibility and control over all manufacturing activities. Simulation models are usually used when complex or complicated systems and processes must be analyzed [2]. In relation to this, ref. [3] conducted an empirical survey on large-scale modeling and simulation of physical systems to investigate research needs and current challenges.

Through advanced modeling, scenario testing, and visualization tools, simulation programs provide a comprehensive framework for optimizing production processes. Optimization can be carried out manually, based on experiments and testing scenarios, or, on the other hand, by using various optimization tools. The authors of [4] studied the

integration of simulation and optimization for the design and improvement of facility layouts. They reviewed a few alternative layout designs, taking into account production and logistics constraints. Based on the Recursive Optimization–Simulation Approach, ref. [5] developed a methodology to provide effective decision support for integrated production planning and scheduling. Optimizing human–robot task allocation using a simulation tool was presented in [6]. Mathematical and simulation-based optimizations were applied in the research in [7]. While the mathematical model was formulated to find an optimal production–distribution plan, the simulation model verified the feasibility of the plan. Similarly, in [8], a mathematical model was proposed to optimize the production processes, with a simulation model to test the alternative facility designs. The study in [9] used a discrete event simulation model to test investment alternatives regarding their impact on selected production indicators. The authors of [10] performed experiments to optimize the production program using the FlexSim simulation program and the OptQuest to obtain a cost-effective solution in a short time. Applications of Tecnomatix Plant Simulation 2402 software to optimize the production system can be found in [11–13]. It is important to mention that the integration of simulation models with virtual reality holds significant importance in the implementation of Industry 4.0. The article in [14] deals with using the Tecnomatix Plant Simulation 2402 software to visualize, analyze, and optimize the modeled production–assembly process in connection with virtual reality. On the other hand, some researchers focused on the development of optimization tools. For instance, the authors of [15] developed the open-source software JSOptimizer 1.0 that can be used to optimize simulation models of complex engineering systems built with JaamSim.

While effective, manual optimization can be time-consuming and may not always yield the global optimum. It is also necessary to emphasize that this approach is highly dependent on the executor's expertise and the specific problem at hand. The second method of optimization is using optimization tools that are commonly integrated into simulation programs. Using optimization tools such as genetic algorithms (GAs) and Experiment Manager (EM) introduces a more automated and exploratory approach. In industry, optimization mainly concerns the production process. For instance, a study by [16] utilized the Experiment Manager to set up scenarios and experiments for optimizing manufacturing processes. Similarly, by employing advanced simulation techniques, including EM and GA, the authors of [17] proposed a methodology to ensure the smooth realization of robust and efficient adaptive manufacturing systems. Ref. [18] proposed a hybrid genetic algorithm that utilized a genetic algorithm to perform a global search and the particle swarm optimization algorithm to perform a local search. The semi-physical simulation technology of digital twins was utilized by [19] to achieve the optimization of the production line using the heuristic balance method. To improve precast production scheduling, ref. [20] developed a hybrid model based on the genetic algorithm and discrete event simulation. The study in [21] applied a genetic algorithm to support decision-making and operational production planning using the Tecnomatix Plant Simulation 2402 software. The paper in [22] proposed an integrated simulation as a holistic decision support tool.

Optimization encompasses both manufacturing and service processes, such as intra-company transportation and storage. Many scholars have carried out research related to the optimization of the automated guided vehicle (AGV) transport system. The deployment of AGVs has significantly increased in recent years, driven by the intensification of automation and the adoption of Industry 4.0 principles. In one study, the authors of [23] presented a flexible framework to simulate transport systems based on AGVs. The framework used to perform the simulation also serves to implement the different policies and algorithms in the global control system. The study in [24] proposed an improved GA for omnidirectional AGV path planning that incorporates an ant colony algorithm (ACO). In the multi-AGV system, each collision avoidance decision has an impact on the efficiency of the system. To optimize the system's fitness regarding decisions, ref. [25] used the particle swarm optimization algorithm. Similarly, ref. [26] presented an easily adaptable method for collision and deadlock resolution in a multi-AGV transport system. A two-stage traffic

scheduling scheme for a multi-AGV system with the use of a GA is presented in the paper by [27]. The paper in [28] introduced an initial model to demonstrate a possible approach for modeling the operation of freely moving AGVs by discrete event simulation. The AGV-hybrid task allocation problem in the assembly line was solved by [29], and real-time AGV scheduling on a flexible shop floor for material handling was addressed in [30].

When designing and optimizing an AGV transport system, it is necessary to consider various limitations regarding space, the number of AGVs, the loading area, and battery capacity. A model with the goal of minimizing the AGV path according to the AGV road network situation was solved by a genetic algorithm in [31]. Ref. [32] presented a scheduling solution that aimed to minimize the maximum completion time for the AGV scheduling problem in an intelligent warehouse. In addition, papers [33] and [34] studied the flexible job shop scheduling problem with a limited number of AGVs and proposed an improved genetic algorithm (IGA) to minimize makespan. The study in [35] proposed a dynamic scheduling method for task assignment and path optimization of AGVs to prevent collisions. The paper by [36] proposed a simulation-based multi-AGV scheduling procedure with limited buffer capacity and battery charging. In ref. [37], the authors designed an intelligent controller of a hybrid AGV while they used genetic algorithms to optimize the speed control strategy to improve efficiency and save energy.

Modeling and optimizing AGV-based transport systems necessitates alignment with real operational conditions. An essential condition is the determination of the movement path. The existing literature extensively addresses AGVs operating along predefined paths. Some studies focus on freely navigating AGVs within specified points, where the exact path is not exactly specified. However, a research gap persists concerning intelligent AGV systems, where AGV movement responds dynamically to device requests without predefined directions or paths. Developing a simulation model and optimizing such AGV transport systems present a significant challenge. This study addresses this gap by presenting a simulation of an intelligent AGV transport system using the Tecnomatix Plant Simulation 2402 software. This paper outlines the solution methodology (Section 2), details model creation and AGV system programming (Section 3), and demonstrates system optimization through the genetic algorithm tool (Section 4), illustrated via a fictional case study.

2. Materials and Methods

2.1. Case Study Description

AGV-based transportation systems are commonly utilized to deliver materials to individual production facilities or workstations along production lines. These systems necessitate timely and precise preparation of materials for processing in the required quantities. Minimizing excessive inventory is crucial, ideally achieving Just-In-Time (JIT) deliveries. When production is stable and uniform, supplying workstations is relatively straightforward. In that case, AGVs can follow a predefined path or move freely among fixed points. However, in flexible production systems where operations vary in sequence or duration, or if the numbers and locations of serviced workstations change, optimization of workstation supply becomes a much more complex issue.

The solution to the problem of servicing the abovementioned flexible manufacturing system is represented using a fictitious case study. This production system is designed for manufacturing two distinct products, referred to as Part 1 and Part 2, and consists of ten workstations (Figure 1). For better clarity within the model, the products Part 1 and Part 2 are color-coded (Part 1—red; Part 2—black). Four different operations are performed by ten devices organized into four machine groups (MG1-MG4) based on the operations conducted.

Each product undergoes three operations across three different machines. The black part (Part 1) is processed sequentially on machines from groups MG1, MG2, and MG4, while the red part (Part 2) is machined on machines from groups MG1, MG3, and MG4. Each group may contain a varying number of machines, each with different processing

times. Transport between machines is facilitated by an appropriate number of AGVs, which are dispatched based on machine requests.

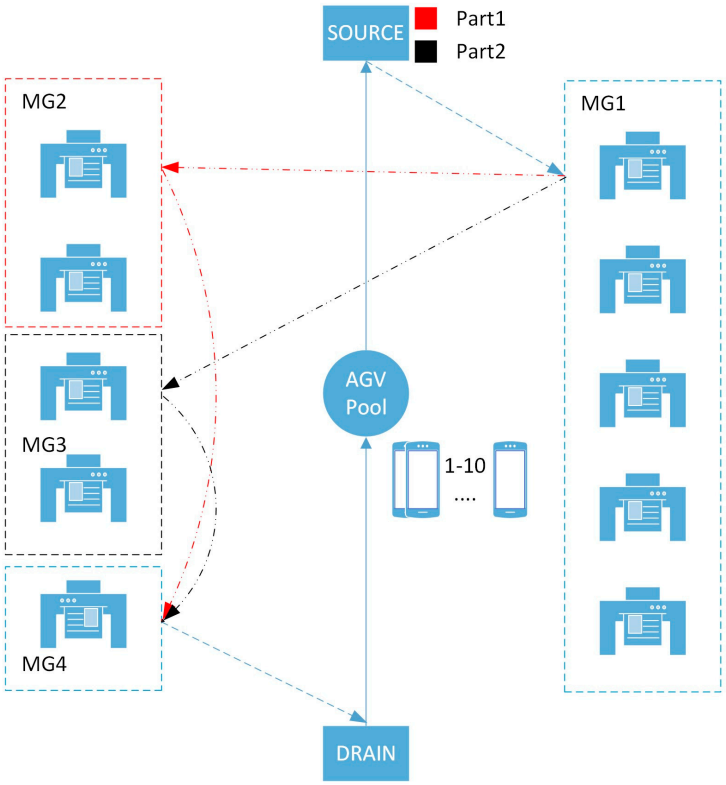


Figure 1. The scheme of the workplace.

The AGV transport system ensures not only the delivery of materials to individual workstations but also the transfer of products between workstations according to the technological process and the final transfer of finished products to the warehouse. For modeling and simulation purposes, we selected the SIEMENS Tecnomatix Plant Simulation program (version 2302). For the subsequent optimization of the AGV system, the optimization tool Genetic Algorithm is used, which is part of the Tecnomatix 2402 software.

2.2. Research Problem Definition

Based on the above, it was necessary to decide on the appropriate way to manage AGVs. In Tecnomatix Plant Simulation, two primary methods for AGV control are commonly utilized: following a precisely defined track or using a method and points (markers). Although the movement between the markers is free, the AGV follows the predetermined points (Figure 2a,b).

Both methods are insufficient for our case, as they significantly restrict the variability required to accommodate changes in the environment (e.g., the number and location of machines). Additionally, alterations in the processing time interval would result in time discrepancies for the affected machines. To circumvent these limitations, we employed an intelligent AGV system controlled using methods (in the SimTalk programming language) and dynamic dispatching of free AGVs based on calling via a table (Figure 2c).

The second task involves optimizing the AGV transport system to determine the optimal number of AGVs required to service the workstations efficiently. The primary parameter monitored in this optimization is the total number of parts produced. The objective is to identify the optimal number of AGVs based on several factors, like the number of production devices, the type of equipment, the operating time required for machining, and the layout of machines in the workplace.

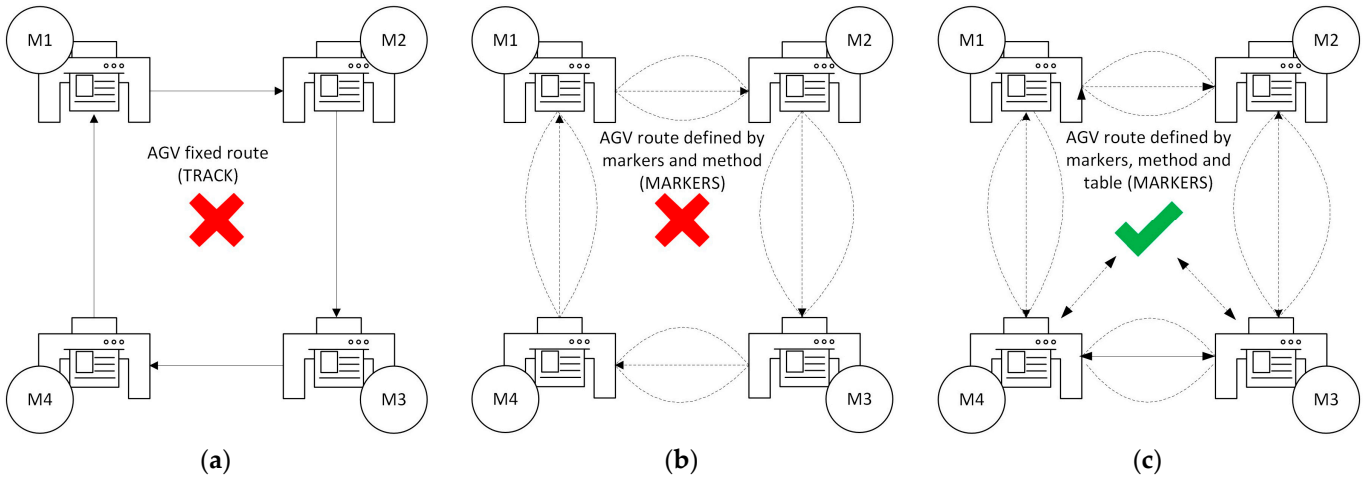


Figure 2. Ways of moving AGVs: track (**a**); markers (**b**); freely moving (**c**).

2.3. Problem-Solving Methodology

The methodology for solving the movement of freely moving AVGs is illustrated in Figure 3. Based on the problem definition in Section 2.2, the methodology consists of four sequential steps. Firstly, it is necessary to analyze the problem of intra-company transport and choose a suitable type of transport system. The choice of transport system is based on the attributes of the production system. In the next step, selecting a suitable simulation program and creating a simulation model of the production system is necessary. In our case study, the simulation program Tecnomatix Plant Simulation version 2302 is applied, and the fictitious production system described in detail in Section 2.1 is modeled. The third step is defining the movement of AGVs, both from the point of view of the AGV's movement path determination and then from the methods defining this movement. The last step is the optimization of the transport system. The optimization goal is to maximize production volume while minimizing the number of AGVs.

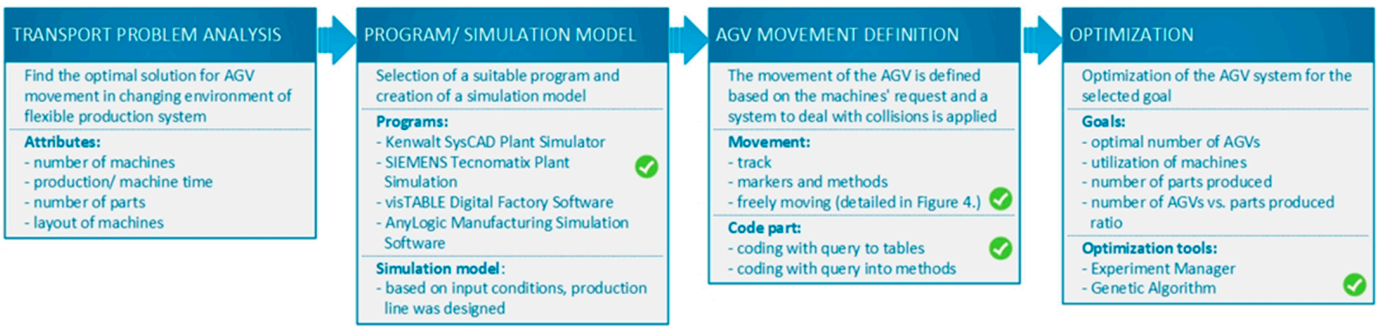


Figure 3. Methodology for problem solving.

The flowchart detailing the sequence of steps for controlling AGV movement based on equipment requirements is presented in Figure 4. It graphically highlights the activities of individual equipment groups MG1–MG4. Subsequent sections elaborate on this methodology, presenting algorithms and methods that govern the operation of the intelligent AGV transport system.

It must be noted that the primary issue with the free movement of AGVs is the occurrence of collisions. Despite the capabilities of the Tecnomatix Plant Simulation program, it does not provide a means to set conflict zones for mutual AGV avoidance effectively. Consequently, collisions are an inevitable part of the simulation. Although the AGV can be configured with two collision zones, it is only possible to define deceleration for each zone. To address this, the problem of AGV collisions and the subsequent need for mutual avoidance is managed by slowing down the AGV movement. This deceleration simulates

the time required for AGVs to maneuver around one another. If an AGV detects another AGV within a distant collision zone, it will decelerate slightly. If it enters a near-collision zone, it will decelerate significantly.

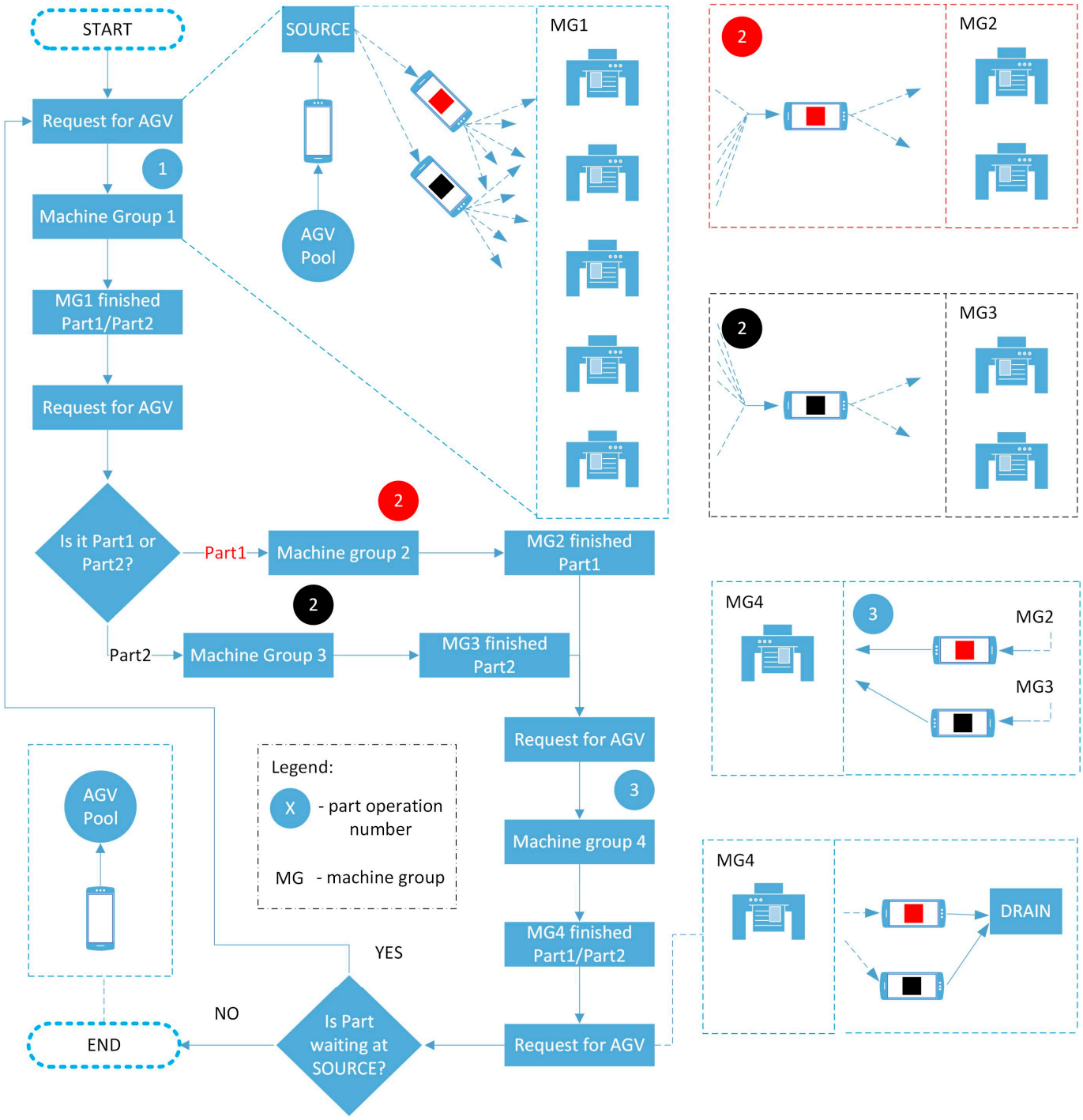


Figure 4. Scheme of procedure for freely moving AGV transport system.

3. Model Creation in Tecnomatix Plant Simulation

Because of the model's complexity, the creation of it involves several distinct phases. First, the location, type, and number of individual machines are determined. Subsequently, the number and variety of parts are specified, ensuring an accurate representation of the production process. Following this, methods for AGV operation, taking into account the specific requirements of the production environment, are developed. The main methods

within the model are then defined. Finally, the optimization tool, a genetic algorithm, is configured to enhance the system's performance, ensuring efficient and effective operation. This multi-step approach allows for a comprehensive and robust simulation model.

Figure 5 presents a planning view of the entire model in the Tecnomatix environment. The subsequent sections provide a detailed description of the creation process, the configuration of individual elements, and the method codes used to manage activities within the model.

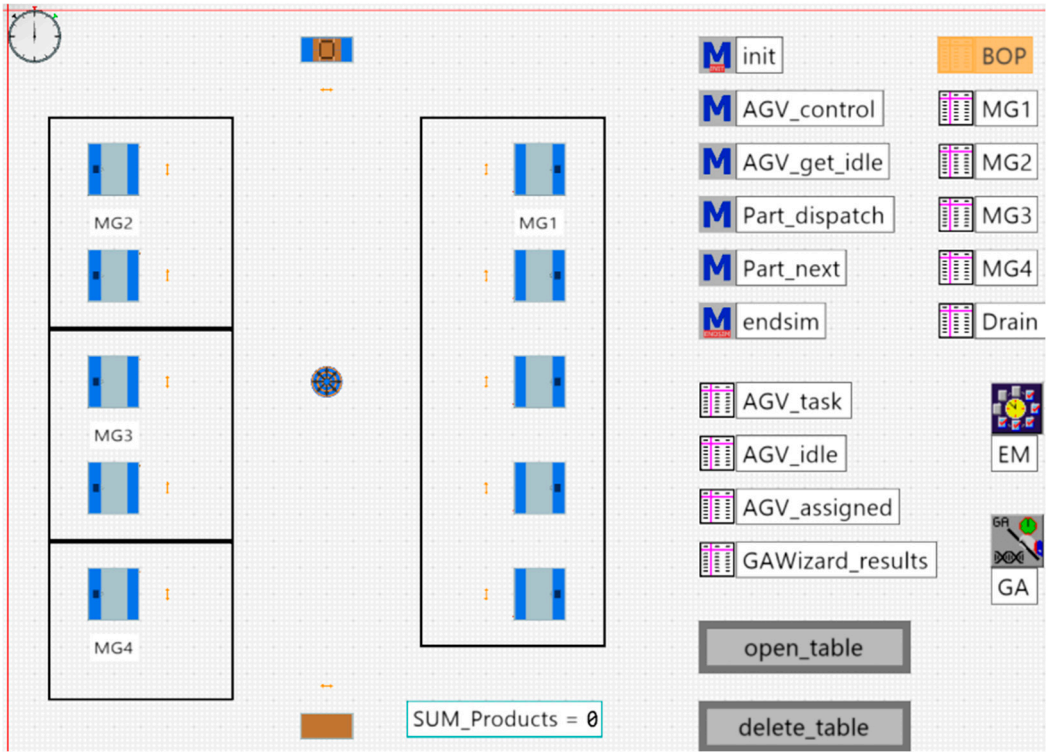
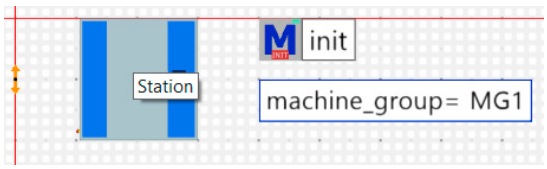


Figure 5. Model of workplace and AGV transport system in Tecnomatix environment.

3.1. Station Settings

The essential elements in the model are the workstations, with one type of workstation employed throughout. While this may suggest simplicity, the workstations vary significantly in their parameters. Each workstation is created as a separate "frame" to allow for greater modularity. This design choice enables the simultaneous activation of two initialization methods: the Init method for the station frame, as illustrated in Figure 6a, and the Init method for the overall model seen in Figure 6b.



(**a**)

```
//message as available
wait 0.001
var mg:object
//create path based on root frame and variable machine_group
mg:=root.extendPath(machine_group)
//make an entry
mg[1,mg.yDim+1]:=self.~
```

(**b**)

Figure 6. Station frame (**a**); Init method code for stations (**b**).

The workstation frame comprises several components. They are the workstation (machine), the marker (a point from which the AGV picks up or delivers the part), the initialization method (Init method), and a machine group variable (machine_group variable).

In the workstations, within the User-defined section, an additional method titled "onExit" has been created to specify the actions to be taken upon the completion of part

processing (Figure 7). This method delineates the subsequent steps for the part once the machining operation is finished. To implement this method, the "onExit" method is added to the Exit field in the Controls section (self.onExit). This ensures that the system accurately follows the defined procedures, such as moving the finished part to the next workplace, updating the production status, or informing the AGV about a ready delivery.

```
var stationGroupList:object
//call dispatch
@.op_no+=1 //increase operation
root.&Part_dispatch.executeNewCallChain(@)
//wait until the part was moved

waituntil @.~ /= ?
//make an entry in the right machine group table
stationGroupList:=root.extendPath(machine_group)
stationGroupList[1,stationGroupList.yDim+1]:=?.~
```

Figure 7. The code of the method "onExit" for workstations.

The variable machine_group is defined as a string, with the value ranging from MG1 to MG4, indicating the machine group to which each workstation belongs. For the Source, the Source frame includes the configuration of the marker position and a table containing information about the parts (refer to Figure 8). This table provides essential details about the parts to be processed, such as type, processing time, and other relevant attributes.

	object 1	real 2	integer 3
string	MU	Frequency	Number
1	.UserObjects.Part1	60.00	1
2	.UserObjects.Part2	40.00	1

Figure 8. Part table located in Source frame.

Within the Source, the table is referenced directly in the "Attributes" section, ensuring that specific information is accurately accessed and utilized during the simulation.

The Source element also includes a "User-defined" method named "onExit". The method "onExit" is triggered from the "Controls" section through the "Entrance" field. Upon completion of the processing steps, the part is subsequently transferred to the storage area, represented by the Drain. The Drain is configured as a "frame" and includes a Marker and an "Init" method. Additionally, the Drain is equipped with a User-defined method called "onEntry", which is activated in the Controls section within the Entrance field (self.onEntry). The codes of "onExit" method of Source, "Init" and "onEntry" method of Drain are presented in Figure 9.

```
root.Part_dispatch(@)
```

(**a**)

```
root.drain[1,1]:=?.~
```

(**b**)

```
root.drain[1,1]:=self.~
```

(**c**)

Figure 9. Codes of the methods: Source_ onExit (**a**); Drain_Init (**b**); Drain_ onEntry (**c**).

3.2. Part Settings

Given that the model incorporates two types of parts with differing processing requirements, it is necessary to define in a table which part will be processed on which machine or machine group. In the BOP (Bill of Process) table (Figure 10), the parts (Part 1 and Part 2) are specified. For each part, a nested table is included under "x" in the "table" column, detailing the processing times and the machine groups where each part will be machined (Figure 11).

	string 0	table 1
string	Part	BOP
1	Part1	x
2	Part2	x

Figure 10. BOP table.

	string 1	string 2	time 3	time 4	string 5
string	Step	description	setup time	processing time	machine group
1	op1		1:00.0000	12:00.0000	MG1
2	op2		1:00.0000	3:00.0000	MG2
3	op3		1:00.0000	1:00.0000	MG4
4					Drain

(**a**)

	string 1	string 2	time 3	time 4	string 5
string	Step	description	setup time	processing time	machine group
1	op1		1:00.0000	12:00.0000	MG1
2	op2		1:00.0000	4:00.0000	MG3
3	op3		1:00.0000	1:00.0000	MG4
4					Drain

(**b**)

Figure 11. BOP nested tables for Part1 (**a**); for Part2 (**b**).

3.3. AGV Settings

For the AGV, two methods and one parameter are configured in the "User-defined" section. The first method, AGV_bypass (Figure 12), simulates the time required for an AGV to bypass other AGVs by decelerating. This method utilizes conflict zones and is defined in the "Controls" section under the "Distance" field (self.AGV_bypass). In the more distant conflict zone (orange), the AGV will decelerate to 70% of its original speed. In the nearby conflict zone (red), the AGV will decelerate to 20% of its original speed.

```
param IsInsideZone : boolean, ZoneNumber : integer
-- param DistanceIsBelowLimit: boolean

var numMUInZone1 = ?.getSafetyZoneContent(1).dim
var numMUInZone2 = ?.getSafetyZoneContent(2).dim

if (numMUInZone1 > 0) then
    // Drive very slowly - other AGVs in zone #1
    ?.Speed := ?.Origin.speed * 0.2
    if ?.curveSpeed /= -1
        ?.curveSpeed := ?.Origin.curveSpeed * 0.2
    end
    // Show load bay in red
    self.~._3D.getGraphic("default", [6, 2]).MaterialDiffuseColor := makeRGBValue(255, 0, 0)
elseif (numMUInZone2 > 0) then
    // Drive a little slower - other AGVs (only) in zone #2
    ?.Speed := ?.Origin.speed * 0.7
    if ?.curveSpeed /= -1
        ?.curveSpeed := ?.Origin.curveSpeed * 0.7
    end
    // Show load bay in orange
    self.~._3D.getGraphic("default", [6, 2]).MaterialDiffuseColor := makeRGBValue(255, 128, 0)
else
    // Drive with original speed - no other AGVs in my safety zones
    ?.Speed := ?.Origin.Speed
    if ?.curveSpeed /= -1
        ?.curveSpeed := ?.Origin.curveSpeed
    end
    // Show load bay in gray
    self.~._3D.getGraphic("default", [6, 2]).MaterialDiffuseColor := makeRGBValue(210, 210, 210)
end
```

Figure 12. AGV_bypass method code of AGV.

The second method, AGV_work, manages the movement of the AGV (Figure 13). Additionally, the AGV_time parameter specifies the time required to load the part. These

methods ensure control over AGV operations, enabling simulation of AGV interactions and part handling.

```
param start,destination:object

//drive to start (empty)

self.~.isIdle:=false

self.~.setRoute([start.marker])
//wait for arriving
waituntil self.~.destinationWasReached

//load part
wait self.~.AGV_time

if start.class.name="Source"
    start.source.cont.move(self.~)
elseif start.class.name="Station" or start.class.name="Drain"
    start.station.cont.move(self.~)
else
    debug
end

//drive to destination (loaded)
self.~.setRoute([destination.marker])
//wait for arriving
waituntil self.~.destinationWasReached

//load part
wait self.~.AGV_time

if destination.class.name="Station" or destination.class.name="Drain"
    self.~.cont.move(destination.station)
else
    debug
end
//finish job
//move to AGVPool
self.~.setRoute([AGVPool])
//wait for arrive
waituntil self.~.destinationWasReached

self.~.isIdle:=true

AGV_get_idle
```

Figure 13. AGV_work method code of AGV.

3.4. Main Methods and Tables

To develop a functional simulation model, it is essential to establish the main methods. This involves creating method codes that specify the actions to be performed, the tables for data entry, and subsequent data retrieval by other methods. The main methods in the model include the following:

Init (Figure 14): At the beginning of each simulation, the main initialization method clears the contents of the "Machine group" and "AGV" tables. Subsequently, the method calls free AGVs from the AGVPool and initiates the execution of the AGV_control method.

AGV_control (Figure 15): This method waits for data to be written to the AGV_task and AGV_idle tables. The AGV_idle table lists the available AGVs that can be assigned to tasks from the AGV_task table, which involves transporting components from point A (Source) to point B (destination). At the end of the method, the method indicates that the transfer is executed according to the code stored in the AGV (AGV_work).

AGV_get_idle (Figure 16): This method identifies when an AGV becomes free and available for use again.

Part_dispatch (Figure 17): After processing a part, this method calls a free AGV and simultaneously initiates the Part_next and AGV_control methods.

```
//delete contents of machine groups
MG1.delete
MG2.delete
MG3.delete
MG4.delete
AGV_task.delete
AGV_idle.delete

AGVPool.getAssignedAGVsTable(AGV_assigned)

AGV_idle[1,1]:=agvpool.getIdleAgv

&AGV_control.executeNewCallChain
```

Figure 14. Main Init method code of model.

```
//wait for job and idle AGV
waituntil AGV_task.yDim>0 and AGV_idle.yDim>0
var start,destination:object
var AGV:object

start:=AGV_task[1,1]
destination:=AGV_task[2,1]

AGV_task.cutRow(1)

AGV:=AGV_idle[1,1]
AGV_idle.cutRow(1)
//assign the job to the AGV

AGV.AGV_work(start,destination)
```

Figure 15. AGV_control method code.

```
var m : object :=.Models.Model
var tb : object     := AGV_assigned
var tb2 : object    --:=
var r,f: integer
------------------------------------------------

for var i := 1 to tb.ydim

    if tb[1,i].isidle and tb[1,i].empty
        AGV_idle[1,1]:=tb[1,i]
        return

    end


next
```

Figure 16. AGV_get_idle method code.

Part_next (Figure 18): This method determines, through the BOP table, to which machine the component should be moved next.

Additionally, the model incorporates the Endsim method (Figure 19), which is not critical for the model's functionality but is utilized to record results from the genetic algorithm into the custom "GAWizard_results" table. The model also includes an "open_table" button, which opens the specified table, and a "delete_table" button that removes data from the table.

```
param part:object

//get next machine
var machine:object


while true
    machine:=Part_next(part)
    if machine=void
        wait 5
    else
        exitLoop
    end
end

//create AGV job
AGV_task.writeRow(1,AGV_task.yDim+1,part.~.~,machine)

AGV_get_idle

AGV_control
```

Figure 17. Part_dispatch method code.

```
param part:object
-> object

//read list of possible stations
var station:object
var stationGroup:string
var stationGroupList:object


//read the successor from BOP
stationGroup:=BOP[1,part.name][5,part.op_no+1]
stationGroupList:=root.extendPath(stationGroup)
//at least one station of the group need to be receptive --> entry
if stationGroupList.yDim>0

    //make a decision
    //lowest utilization
    stationGroupList.setCursor(1,1)
    stationGroupList.minAttr({1,1}..{1,*},"statWorkingPortion")
    station:=stationGroupList[1,stationGrouplist.cursorY]

    //cut Row
    stationGroupList.cutRow(stationGrouplist.cursorY)

    return station
else
    return void
end
```

Figure 18. Part_next method code.

```
//record the results of GAWizard
GAWizard_results[1,GAWizard_results.Ydim+1] :=Drain_.Station.NumIn
GAWizard_results[2,GAWizard_results.Ydim]:= AGVPool.Amount
```

Figure 19. Endsim method code of model.

As indicated by the above presented methods, data are consistently written to and retrieved from corresponding tables within the model. These tables are named AGV_task, AGV_idle, AGV_assigned, MG1, MG2, MG3, MG4, Drain, and GAWizard_results (Figure 4). Each method interacts with these tables to ensure accurate data management and operational flow.

The model also includes a display named "SUM_Products", which tracks the number of components transported to the warehouse (Drain). This display is configured for observation (Mode: Watch), with the observation path set to "Drain_.Station.NumIn".

4. Results

4.1. Results of Simulation (Basic Configuration with Three AGVs)

The simulation model, thus prepared, is now ready for testing and optimization. The model in the simulation process is presented in Figure 20. The primary objective of model testing is to verify the free movement of the AGV based on the requirements of the operated devices. Additionally, the settings and programming of other model elements, as well as their interactions, will be thoroughly examined. The model will be tested with a fixed number of AGVs (three AGVs), and the simulation will run continuously for a period of 8 h.

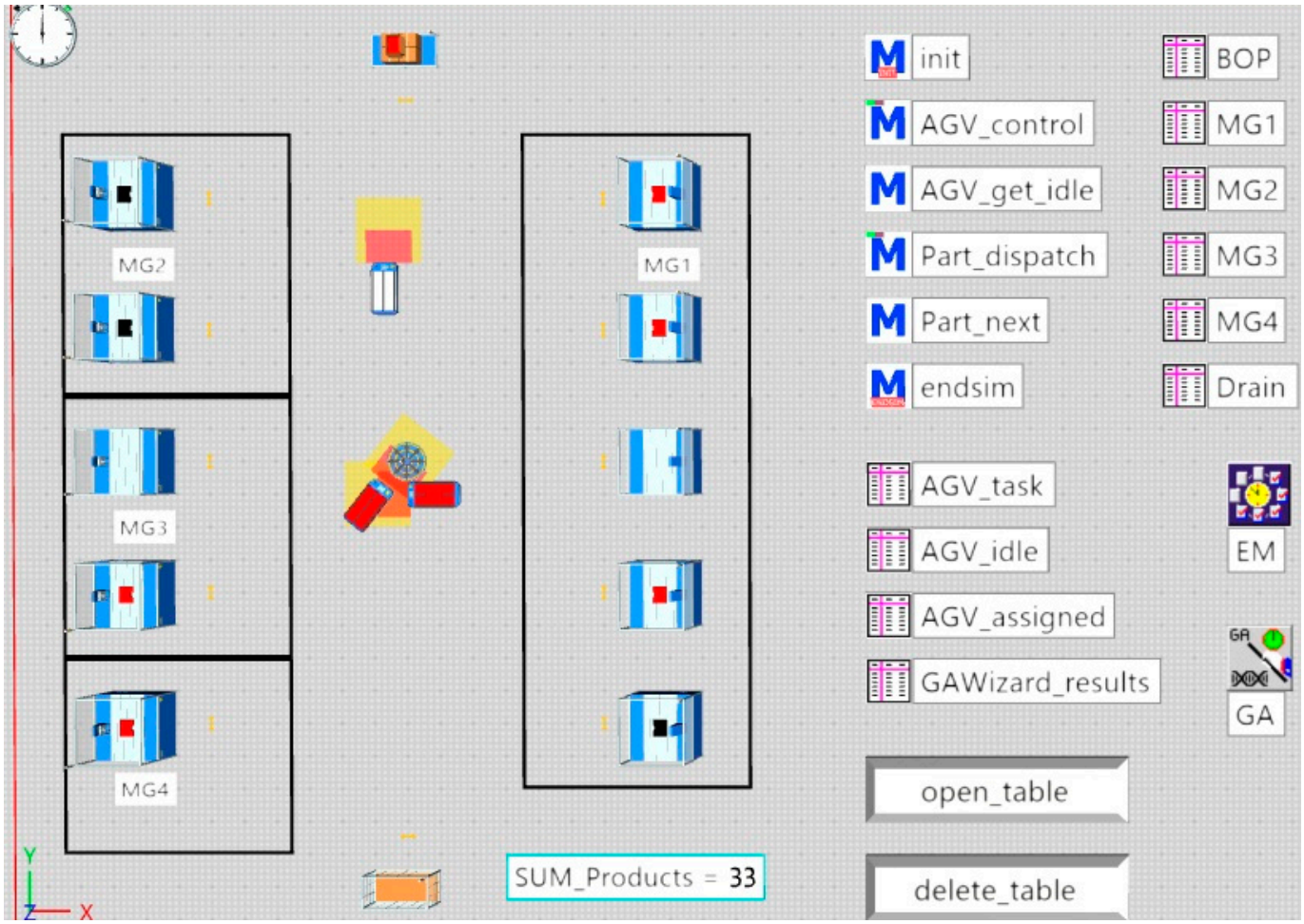
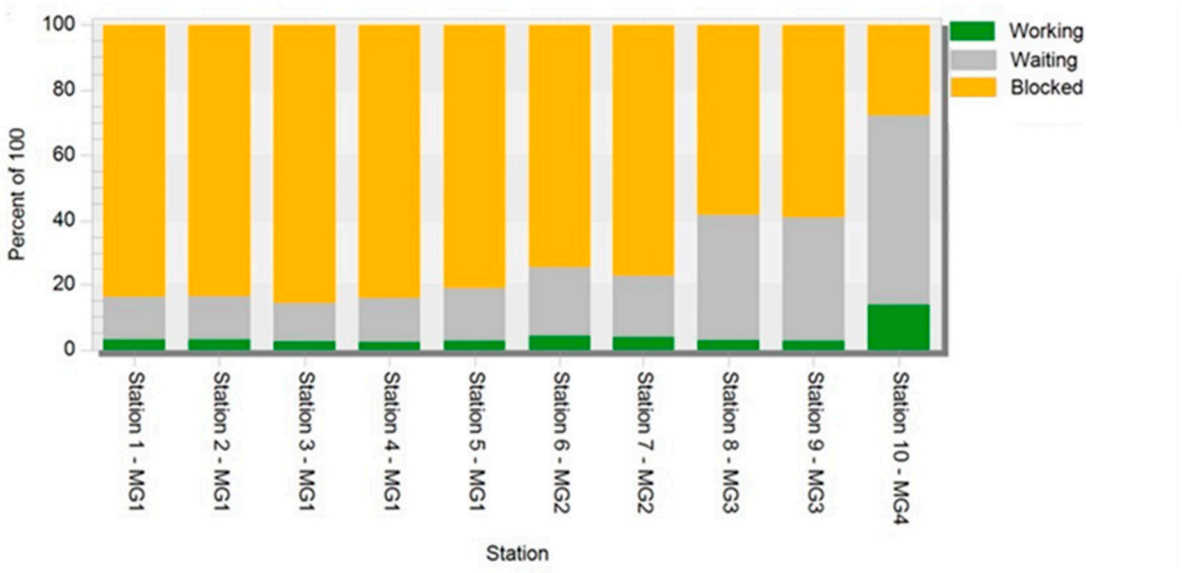


Figure 20. Simulation of workplace operation with three AGVs.

The output of this testing phase will include an assessment of the AGV transport system and the utilization rates of the workstations, which may serve as secondary objectives for subsequent optimization. The initial configuration of the model acts as the baseline for optimization. By simulating this setup, we will obtain crucial data to accurately estimate the parameters and constraints for the optimization tool, the genetic algorithm.

The simulation results for the initial model with three AGVs, along with the statistics of the workstation utilization, are presented in Figure 21. The total output is 395 products per 8 h, consisting of 236 units of Part 1 and 159 units of Part 2. The workstations exhibited a maximum utilization rate of 18%, with a significant portion of time spent blocked. This indicates that the workplace operation in this configuration is highly inefficient. In addition to interoperation transport, production logistics or inappropriate equipment placement can also contribute to the low utilization of the workstation.



Simulation time:8:00:00.0000

Object	Name	Mean Life Time	Throughput	TPH	Production	Transport	Storage	Value added	Portion
Drain_.Station	Part1	12:46.8543	236	30	89.19%	10.81%	0.00%	3.91%	
Drain_.Station	Part2	9:37.6746	159	20	87.53%	12.47%	0.00%	5.19%	

Cumulated Statistics of the Parts which the Drain Deleted

Figure 21. Simulation results for the initial model with three AGVs.

4.2. Results of Optimization via Genetic Algorithm

The optimization of interoperation transport within the current model will be conducted using the genetic algorithm tool. The primary objective of this optimization is to enhance the AGV transport system to maximize production volume while minimizing the number of AGVs required. A secondary effect of the optimization is to increase the efficiency of production equipment utilization.

The genetic algorithm settings are configured within the GAWizard functionality. Figure 22 displays the context window for the GAWizard functionality (left), the optimization parameter table (top right), and the fitness calculation table (bottom right).

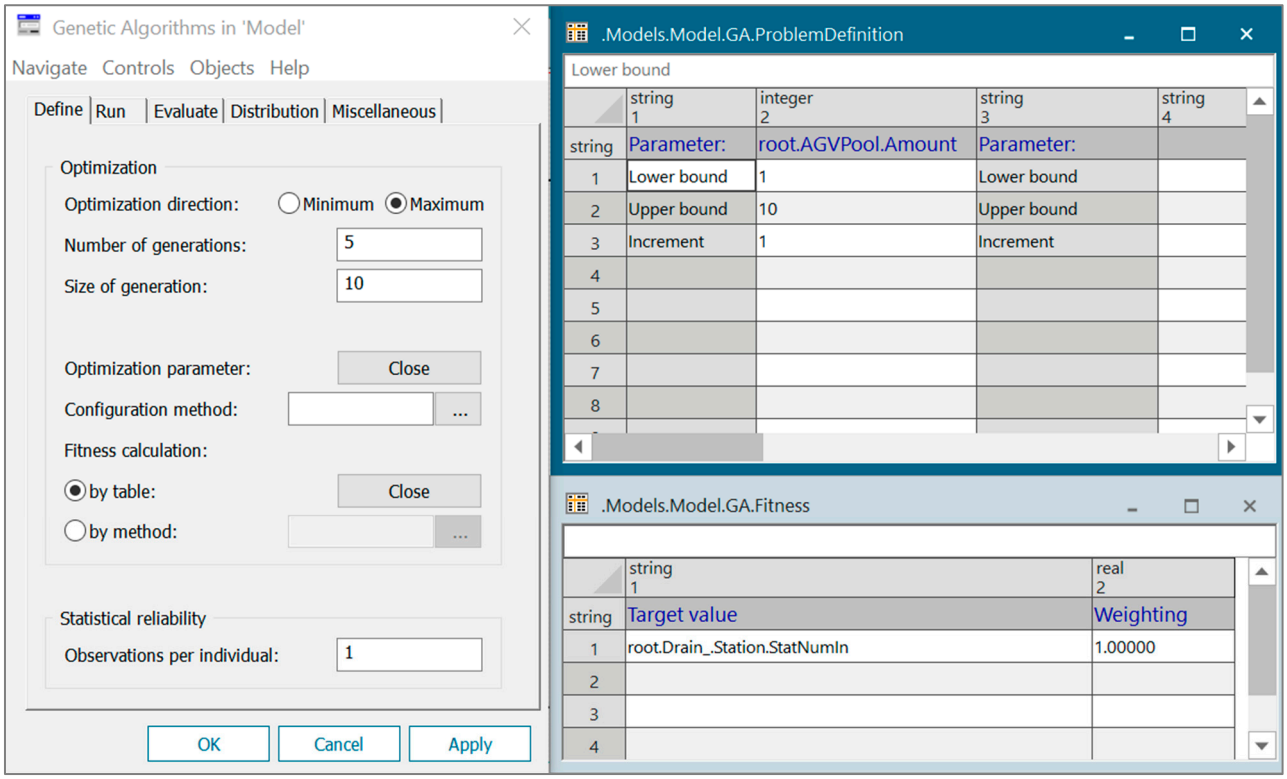


Figure 22. Configuration of GAWizard.

The genetic algorithm optimization parameters are configured as follows: the optimization direction is set to maximize, the number of generations is three, the size of each generation is ten, and there are three observations per individual.

Detailed results from the genetic algorithm optimization process are presented in Figure 23. The genetic algorithm identified the optimal solution in 5 min and 49 s, determining that the optimal number of AGVs is five. Figure 22 presents the five best simulation outputs relative to the specified objective. The "Fitness" column reflects the average production volume derived from three observations for each simulation run. The optimal solution, which involves equipping the workplace with five AGVs, results in a production volume of 431 pieces. This optimal configuration was achieved by the second generation, as depicted in Figure 24, in which the vertical axis represents the number of pieces and the horizontal axis represents the generation number.

	table 0	real 1	integer 2	table 3	table 4	table 5
string	Individual	Fitness	ID	Chromosomes ->	Observations	AGVs amount
1	Gen 2 Ind 6	431.333333333333	16	Chrom 1	Fitness 16	5
2	Gen 1 Ind 4	430	4	Chrom 1	Fitness 4	9
3	Gen 1 Ind 8	430	8	Chrom 1	Fitness 8	7
4	Gen 2 Ind 13	430	23	Chrom 1	Fitness 23	10
5	Gen 1 Ind 2	429.666666666667	2	Chrom 1	Fitness 2	6
6						

Figure 23. Genetic algorithm optimization results.

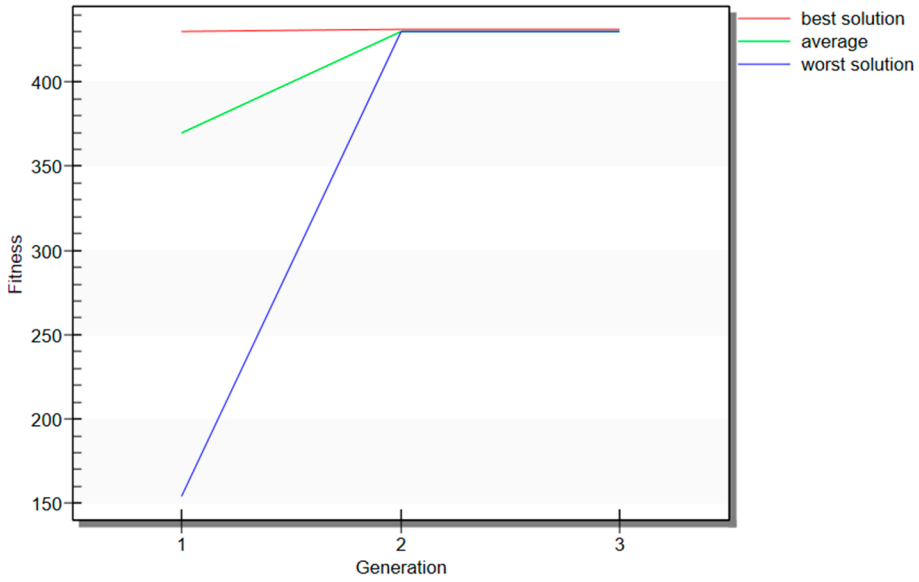


Figure 24. Genetic algorithm performance graph.

5. Conclusions

Intra-company transport systems based on AGVs are widely employed in modern manufacturing environments. Designing such systems also involves determining the movement of the AGVs, which can follow predefined tracks or specific points (markers). A special issue arises with the implementation of freely moving AGVs, where their navigation is dynamically adjusted based on concrete events, e.g., the requirements of other devices in the system.

This study presented a comprehensive methodology for the simulation and optimization of an intelligent transport system that utilized freely moving AGVs. Traditional approaches often prefer predefined paths for AGVs. Nevertheless, a predefined path limits flexibility in dynamic production environments. For comparison, we can mention some research that has addressed various problems related to AGV systems in the Tecnomatix environment. For instance, the study in [28] focused on the pathfinding problem, while the study in [29] explored the hybrid AGV task allocation problem on a predefined path. Also, the study in [30] examined the real-time scheduling of AGVs on a defined path. On the other hand, papers [25–27] dealt exclusively with the path conflict problem. There is no research described in the scientific literature that deals with an AGV motion modeling approach without a predefined path. Our research addresses this gap by implementing a simulation model where AGV movements are dynamically adjusted based on real-time requests from production devices. The research contribution is a detailed description of the methodology, including codes for this specific AGV free movement.

We employed a fictional case study featuring a flexible production system manufacturing two distinct parts. The manufacturing process required part transportation between workstations. The complexity of this situation required the development of specific codes for methods of the AGVs' work, bypass of AGVs, AGV control, and product dynamic handling. The simulation of the initial model, configured with three AGVs, served as the baseline for optimization. Through the initial model simulation, we gained data to estimate the parameters and constraints for the optimization via the genetic algorithm. The AGV transport system optimization objective was to maximize production volume while minimizing the number of AGVs required. The optimal configuration, involving five AGVs, achieved a production volume that was significantly higher than the initial setup.

This study highlighted the role of intelligent AGV systems in flexible manufacturing systems, emphasizing the importance of real-time adjustments to AGV paths. This research advances the theoretical understanding of AGV-based intra-company transport systems by addressing the constraints of traditional AGV routing approaches. The limitations of this

research are that this study utilized a fictional case study with a limited scope, focusing on a specific production and transport scenario. The study was restricted to addressing the movement of AGVs without extending the analysis to other critical dimensions such as economic aspects, capacity utilization, or long-term efficiency. While the findings contribute valuable insights, they serve as a foundation for future investigations incorporating these additional aspects. The methodology can serve to encourage further progress in intelligent transport systems and their integration into production systems. From a managerial view, by implementing AGV systems that adapt to real-time production needs, managers can better align their transport logistics with overall production goals. However, it is important to consider other factors that, although beyond the scope of the current study, are crucial for long-term success.

Author Contributions: Conceptualization, L.R. and J.F.; methodology, L.R. and J.F.; validation, L.R., J.F. and J.P.; formal analysis, L.R. and J.F.; investigation, L.R., J.P. and J.F.; resources, J.F.; writing—original draft preparation, J.F., L.R. and I.D.; writing—review and editing, I.D. and J.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Acknowledgments: This paper was developed within the projects KEGA_010ŽU-4/2023, KEGA_005TUKE-4/2022, VEGA 1/0674/24, APVV-21-0195, and the operational program Integrated infrastructure with ITMS: 313011T567.

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

References

1. Jeon, S.M.; Kim, G. A Survey of Simulation Modeling Techniques in Production Planning and Control (PPC). *Prod. Plan. Control* **2016**, *27*, 360–377. [CrossRef]
2. Kuncova, M.; Zajoncova, M. Discrete Event Simulation Usage to Model and Optimize the Production Line. *MM Sci. J.* **2018**, *2018*, 2325–2332. [CrossRef]
3. Schweiger, G.; Nilsson, H.; Schoeggl, J.; Birk, W.; Posch, A. Modeling and Simulation of Large-Scale Systems: A Systematic Comparison of Modeling Paradigms. *Appl. Math. Comput.* **2020**, *365*, 124713. [CrossRef]
4. Zúñiga, E.R.; Moris, M.U.; Syberfeldt, A.; Fathi, M.; Rubio-Romero, J.C. A Simulation-Based Optimization Methodology for Facility Layout Design in Manufacturing. *IEEE Access* **2020**, *8*, 163818–163828. [CrossRef]
5. Vieira, M.; Moniz, S.; Gonçalves, B.S.; Pinto-Varela, T.; Barbosa-Póvoa, A.P.; Neto, P. A Two-Level Optimisation-Simulation Method for Production Planning and Scheduling: The Industrial Case of a Human–Robot Collaborative Assembly Line. *Int. J. Prod. Res.* **2022**, *60*, 2942–2962. [CrossRef]
6. Bänziger, T.; Kunz, A.; Wegener, K. Optimizing Human–Robot Task Allocation Using a Simulation Tool Based on Standardized Work Descriptions. *J. Intell. Manuf.* **2020**, *31*, 1635–1648. [CrossRef]
7. Ji, J.; Chiadamrong, N. Integrating Mathematical and Simulation Approach for Optimizing Production and Distribution Planning With Lateral Transshipment in a Supply Chain. *Int. J. Inf. Syst. Supply Chain Manag.* **2022**, *15*, 1–30. [CrossRef]
8. Rouyendegh, B.D.; Tirkolaee, E.B.; Utku, D.H. The Evaluation and Improvement of the Production Processes of an Automotive Industry Company via Simulation and Optimization. *Sustainability* **2023**, *15*, 2331. [CrossRef]
9. Vrecko, I.; Kovac, J.; Rupnik, B.; Gajsek, B. Using Queuing Simulation Model in Production Process Innovations. *Int. J. Simul. Model.* **2019**, *18*, 47–58. [CrossRef]
10. Smagowicz, J.; Szwed, C.; Berlec, T. An Assortment–Quantity Optimization Problem in Printing Industry Using Simulation Modelling. *Sustainability* **2024**, *16*, 1693. [CrossRef]
11. Malega, P.; Gazda, V.; Rudy, V. Optimization of Production System in Plant Simulation. *Simulation* **2021**, *98*, 295–306. [CrossRef]
12. Bambura, R.; Sujová, E.; Čierna, H. Utilizing Computer Simulation to Optimize Furniture Production System. *BioResources* **2020**, *15*, 6752–6765. [CrossRef]
13. Istokovic, D.; Perinic, M.; Vlatkovic, M.; Brezocnik, M. Minimizing Total Production Cost in a Hybrid Flow Shop: A Simulation-Optimization Approach. *Int. J. Simul. Model.* **2020**, *19*, 559–570. [CrossRef]
14. Trebuna, P.; Pekarcikova, M.; Duda, R.; Svantner, T. Virtual Reality in Discrete Event Simulation for Production–Assembly Processes. *Appl. Sci.* **2023**, *13*, 5469. [CrossRef]

15. Katsios, D.; Xanthopoulos, A.S.; Koulouriotis, D.E.; Kiatipis, A. A Simulation Optimisation Tool and Its Production/Inventory Control Application. *Int. J. Simul. Model.* **2018**, *17*, 257–270. [CrossRef]
16. Butrat, A.; Supsomboon, S. A Plant Simulation Approach for Optimal Resource Utilization: A Case Study in the Tire Manufacturing Industry. *Adv. Prod. Eng. Manag.* **2022**, *17*, 243–255. [CrossRef]
17. Grznár, P.; Burganová, N.; Mozol, Š.; Mozolová, L. A Comprehensive Digital Model Approach for Adaptive Manufacturing Systems. *Appl. Sci.* **2023**, *13*, 10706. [CrossRef]
18. Maghawry, A.; Hodhod, R.; Omar, Y.; Kholief, M. An Approach for Optimizing Multi-Objective Problems Using Hybrid Genetic Algorithms. *Soft Comput.* **2021**, *25*, 389–405. [CrossRef]
19. Zhao, R.; Zou, G.; Su, Q.; Zou, S.; Deng, W.; Yu, A.; Zhang, H.; Zhao, R.; Zou, G.; Su, Q.; et al. Digital Twins-Based Production Line Design and Simulation Optimization of Large-Scale Mobile Phone Assembly Workshop. *Machines* **2022**, *10*, 367. [CrossRef]
20. Wang, Z.; Hu, H.; Gong, J. Framework for Modeling Operational Uncertainty to Optimize Offsite Production Scheduling of Precast Components. *Autom. Constr.* **2018**, *86*, 69–80. [CrossRef]
21. Bojic, S.; Maslaric, M.; Mircetic, D.; Nikolicic, S.; Todorovic, V. Simulation and Genetic Algorithm-Based Approach for Multi-Objective Optimization of Production Planning: A Case Study in Industry. *Adv. Prod. Eng. Manag.* **2023**, *18*, 250–262. [CrossRef]
22. Bleicher, F.; Duer, F.; Leobner, I.; Kovacic, I.; Heinzl, B.; Kastner, W. Co-Simulation Environment for Optimizing Energy Efficiency in Production Systems. *CIRP Ann.* **2014**, *63*, 441–444. [CrossRef]
23. López, J.; Zalama, E.; Gómez-García-Bermejo, J. A Simulation and Control Framework for AGV Based Transport Systems. *Simul. Model. Pract. Theory* **2022**, *116*, 102430. [CrossRef]
24. Niu, Q.; Fu, Y.; Dong, X. Omnidirectional AGV Path Planning Based on Improved Genetic Algorithm. *World Electr. Veh. J.* **2024**, *15*, 166. [CrossRef]
25. Cao, X.; Zhu, M. Research on Global Optimization Method for Multiple AGV Collision Avoidance in Hybrid Path. *Optim. Control Appl. Methods* **2021**, *42*, 1064–1080. [CrossRef]
26. Zając, J.; Małopolski, W. Structural On-Line Control Policy for Collision and Deadlock Resolution in Multi-AGV Systems. *J. Manuf. Syst.* **2021**, *60*, 80–92. [CrossRef]
27. Liu, J.; Wang, Z.; Xu, Q.; Huang, Q. Path Scheduling for Multi-AGV System Based on Two-Staged Traffic Scheduling Scheme and Genetic Algorithm. *J. Comput. Methods Sci. Eng.* **2015**, *15*, 163–169. [CrossRef]
28. Skapinyecz, R.; Landschützer, C. Application of Freely Moving AGVs in a Discrete Event Simulation Environment: A First Model. *Adv. Logist. Syst.-Theory Pract.* **2022**, *16*, 5–23. [CrossRef]
29. Hu, Y.; Wu, X.; Zhai, J.; Lou, P.; Qian, X.; Xiao, H. Hybrid Task Allocation of an AGV System for Task Groups of an Assembly Line. *Appl. Sci.* **2022**, *12*, 10956. [CrossRef]
30. Hu, H.; Jia, X.; He, Q.; Fu, S.; Liu, K. Deep Reinforcement Learning Based AGVs Real-Time Scheduling with Mixed Rule for Flexible Shop Floor in Industry 4.0. *Comput. Ind. Eng.* **2020**, *149*, 106749. [CrossRef]
31. Liang, C.; Zhang, Y.; Dong, L. A Three Stage Optimal Scheduling Algorithm for AGV Route Planning Considering Collision Avoidance under Speed Control Strategy. *Mathematics* **2022**, *11*, 138. [CrossRef]
32. Cheng, W.; Meng, W. An Efficient Genetic Algorithm for Multi AGV Scheduling Problem about Intelligent Warehouse. *Robot. Intell. Autom.* **2023**, *43*, 382–393. [CrossRef]
33. Meng, L.; Cheng, W.; Zhang, B.; Zou, W.; Fang, W.; Duan, P. An Improved Genetic Algorithm for Solving the Multi-AGV Flexible Job Shop Scheduling Problem. *Sensors* **2023**, *23*, 3815. [CrossRef] [PubMed]
34. Liu, Q.; Wang, N.; Li, J.; Ma, T.; Li, F.; Gao, Z. Research on Flexible Job Shop Scheduling Optimization Based on Segmented AGV. *Comput. Model. Eng. Sci.* **2022**, *134*, 2073–2091. [CrossRef]
35. Yu, C.; Liao, W.; Zu, L. Dynamic Scheduling Optimization Method for Multi-AGV-Based Intelligent Warehouse Considering Bidirectional Channel. *Systems* **2023**, *12*, 9. [CrossRef]
36. Park, J.-S.; Kim, J.-W. Multi-AGV Scheduling under Limited Buffer Capacity and Battery Charging Using Simulation Techniques. *Appl. Sci.* **2024**, *14*, 1197. [CrossRef]
37. Sierra-Garcia, J.E.; Santos, M. AGV Fuzzy Control Optimized by Genetic Algorithms. *Log. J. IGPL* **2024**, jzae033. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.