Multi-agent Simulation for Scheduling and Path Planning of Autonomous Intelligent Vehicles

Kader Sanogo[1,2](✉), M'hammed Sahnoun[3], and Abdelkader Mekhalef Benhafssa[1]

[1] CESI LINEACT, EA 7527, Angouleme Campus, 16400 La Couronne, France
ksanogo@cesi.fr
[2] ENSAM, 75013 Paris, France
[3] CESI LINEACT, EA 7527, Rouen Campus, S. E du Rouvray, 76800 Rouen, France
msahnoun@cesi.fr

Abstract. Autonomous and Guided Vehicles (AGVs) have long been employed in material handling but necessitate significant investments, such as designating specific movement areas. As an alternative, Autonomous and Intelligent Vehicles (AIVs) have gained traction due to their adaptability, intelligence, and capability to handle unexpected obstacles. Yet, challenges like optimizing scheduling and path planning, and managing routing conflicts persist. This study introduces a simulator tailored for AIV scheduling and path planning in various production systems. The simulator supports both predictive, where paths are predetermined, and dynamic scheduling, with real-time optimization. Paths are determined using Dijkstra's method, ensuring AIVs use the shortest route. When path-sharing conflicts arise, a multi-criteria priority system comes into play, and its impact on the makespan is assessed. Experimental results highlight the advantage of AIVs over AGVs in most scenarios and the simulator's efficiency in generating effective schedules, incorporating the priority management system.

Keywords: Simulation · AIV · Job-shop scheduling · FMS · Multi-agent System · Industry 5.0

1 Introduction

Research in unmanned ground vehicles has been done for several decades and is continuously creating advancements and capabilities [4]. For more than a decade, AGVs have proven their effectiveness in material handling tasks in manufacturing workshops or logistic warehouses [8]. However, AGV installation is expensive, as it requires modifying the workshop's layout by defining dedicated movement areas [9]. Since AGVs are guided robots, any modification to the workshop's layout requires updating their map and dedicated environment. To alleviate these problems, more intelligent, flexible, and collaborative mobile robots, namely AIVs, are increasingly being used [4].

J.-L. Guisado-Lizar et al. (Eds.): SIMUtools 2023, LNICST 519, pp. 195–205, 2024.
https://doi.org/10.1007/978-3-031-57523-5_15

Unlike AGVs, AIVs do not require dedicated areas in the workshop and can navigate around static and dynamic obstacles, including human operators. Hence, AIVs are relevant in Industry 5.0, which refers to a human-centered industry where humans are working alongside robots and smart machines [5]. They can provide several advantages over traditional transportation methods, such as increased efficiency, flexibility, and safety [4,9]. However, the deployment of AIVs presents some challenges [9], such as the need to efficiently schedule AIVs, plan their paths carefully, and resolve conflicts that may occur in routing.

In this paper, a simulator for scheduling and path planning of AIVs in job-shop production systems is presented. The simulator can be used to simulate both advanced and dynamic scheduling. For predictive scheduling, the AIVs plan their paths based on an optimized schedule that is generated offline. For dynamic scheduling, their paths are planned based on a real-time optimization algorithm that is integrated into the simulator. The simulator uses a path-planning method based on Dijkstra's method for finding the shortest path for AIVs. Routing conflict resolution is based on a multi-criteria system, and the influence of each criterion on the makespan is studied.

Job shop scheduling problems with mobile robots handling materials have been extensively studied by researchers. Indeed, Bilge *et al.* [3] have considered the problem as a simultaneous scheduling of machines and vehicles. They propose four layouts in the literature, each consisting of four machines and one load/unload station, and transportation tasks are carried out by two AGVs. Taking this work as background, Ham [7] proposes a constraint programming approach to solve the Job-Shop Scheduling Problem (JSSP) with AGV-transport. He considers both machines and AGVs as constrained resources. Abderrahim *et al.* [1] tackle the JSSP with automated transportation tasks, treating workstations and vehicles as resources, and employ a Variable Neighborhood Search (VNS) algorithm to optimize makespan by scheduling both manufacturing and transportation tasks.

In recent years, simulation has been used to address various challenges in FMS [2,13]. Simulation is valuable for identifying phenomena that may not have been apparent during theoretical modeling stages [14]. Moreover, some constraints are difficult to model, so simulation is an alternative to overcome this problem. For instance, in [17], the authors used simulation to demonstrate the difference between the simulated and the theoretical schedule in a simple example. Recently, [16] have introduced a simulation approach to solve the Flexible Job-Shop Scheduling Problem (FJSSP) with transportation tasks, a more difficult problem than JSSP with transportation tasks. In [15], the authors developed a multi-agent simulation for the FJSSP, focusing on AGV collision avoidance and testing its influence on AGV fleet and makespan. They then enhanced this approach to simulate predictive and dynamic schedules, incorporating collision avoidance and deadlock resolution algorithms [14]. Current research emphasizes AGV simulation, neglecting the vital role of AIV implementation and management in Industry 5.0

This paper's tackles this problem by proposing the following contributions:

- Address job-shop production systems with transportation tasks carried out by AIVs.

- Proposes a decentralized method for AIV path planning inspired by Dijkstra's method.
- Propose, as well, a decentralized method for managing priorities in AIVs routing conflicts.
- Present an integrated simulation approach for job-shop scheduling optimization with a decentralized AIV's fleet management.

The remainder is organized as follows. Section 2 is dedicated to the problem description, where the job-shop scheduling, path planning, and collision avoidance problems are presented. The study backbone is contained in Sect. 3. Section 4 presents experiments conducted and discusses the results. Finally, a conclusion ends this paper.

2 Problem Description

2.1 JSSP and FJSSP

The JSSP is the problem of sequencing a set of jobs $\mathcal{J} = \{J_1, J_2, ..., J_I\}$ to be processed on a set of machines $\mathcal{M} = \{M_1, M_2, ..., M_M\}$ in a job-shop organization. Each job J_i is composed of a sequence $(O_{i1}, O_{i2}, ..., O_{in})$ of operations to be performed consecutively. The operation O_{ij}, which means operation j of job i, can be performed only on the machine $M_k \in \mathcal{M}$ with the processing time τ_k. Furthermore, a machine can only perform one operation at a time, and preemption of operations is not allowed. However, in FJSSP, an operation O_{ij} is performed by a machine M_k within a subset of eligible machines $\mathcal{M}_{ij} \subset \mathcal{M}$ $(1 \leq card(\mathcal{M}_{ij}) \leq card(\mathcal{M}))$. We have complete flexibility when $card(\mathcal{M}_{ij}) = card(\mathcal{M})$. Otherwise, it is a partial flexibility.

Job transportation between two machines is performed by a single-load AIV. A transportation task is denoted by $T_{i,j}$, which means the transportation of the job J_i to the machine selected to perform the operation $O_{i,j}$. Task preemption is not allowed, i.e., we can not interrupt a task once it starts. Besides, it is assumed that each machine M_k has an input and an output buffer, respectively B_k^I and B_k^O, for storing jobs before processing and after processing. It is also assumed that all the jobs are stored in a load/unload (L/U) station at the beginning/end of the execution.

2.2 Path Planning and Collision Avoidance

Path planning involves finding a suitable path for robots to move between two locations [10]. Several parameters can be taken into account, such as distance, duration [12], risk of collisions and deadlocks [6], and energy consumption [11].

In our case, we have developed a path planning method that searches for either the shortest path or the fastest path. The environment is modeled by an undirected connected graph in which each point of interest (machine, stock, corner, and intersection) is represented by a node, and the edges are the corridors linking them. When a task is assigned to the robot, it plans its path based on this graph.

Collisions are avoided locally by the robots themselves. Indeed, when two robots meet, thanks to our priority management system, one stops and gives way to the other as presented in Fig. 1. Once the path is clear, the robots continue as normal. The difference with our previous work is that no direction of travel is imposed, so AIVs can meet face-to-face. However, the algorithms developed in [14] remain valid and are used to support the priority management system.

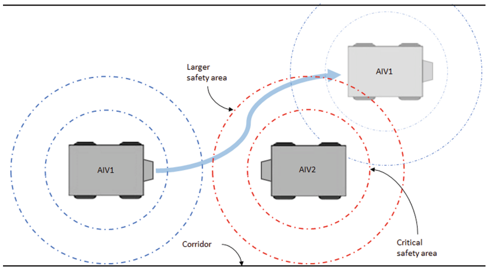


Fig. 1. AIVs collision avoidance mechanism

To ensure safety and prevent collisions between robots, each robot employs two safety radii: a larger radius for obstacle detection and speed reduction, and a smaller radius for immediate stopping. Upon detecting another robot, the priority management system takes over. Within the larger radius, both robots halve their speed. Inside the smaller radius, one robot stops, while the prioritized robot initiates a go-around maneuver, further reducing its speed. Once the path is clear, both robots gradually accelerate back to their cruising speed of 1 m/s.

3 Methods

3.1 Multi-agent System (MAS)

The simulator is based on a MAS involving four (04) main agents: AIVs, machines, stocks, and jobs. The agents are interrelated as follows:

- AIVs pick/deliver jobs from/to stocks.
- Stocks store jobs before/after processing by machines.
- Machines process jobs (Fig. 2).

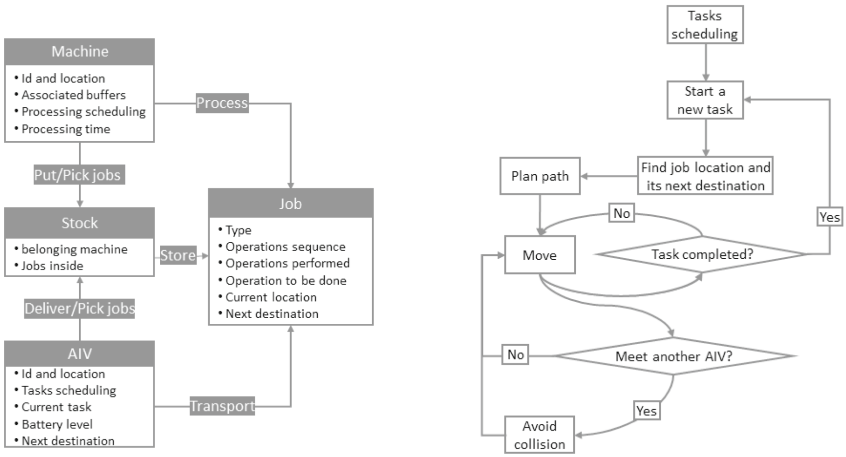
Table 1 summarize the assumptions made for each agent.

3.2 Framework

The multi-agent system (MAS) and the environment are simulated using NetLogo 6.2, a programming language and simulator designed for modeling and simulating systems with multiple interacting agents. Each agent is represented

Table 1. Assumptions for each agent's type

AIVs	Machines	Stocks	Jobs
Are independent of each other	Are independent of each other	Machine buffers have an identical limited capacity	Are independent of each other
Can transfer one and only one job at a time	Can perform one and only one operation at a time	L/U station capacity is unlimited	Can be processed on one and only one machine at a time
Job's load/unload time is included in transportation time	Setup times and breakdowns are ignored	Can store products as long as possible	Can be transported by one and only one AIV at a time
Plan path from task scheduling	Process jobs according to the scheduling		Must be processed according to the scheduling
Avoid collisions by their own	Are available at the beginning of the simulation		Preemption of operation is not allowed



(a) Multi-agent system

(b) Diagram of AIV performing tasks in the simulator.

Fig. 2. Simulation framework

as a "turtle" belonging to a specific "breed". Global parameters apply to the entire model, while breed-specific parameters are exclusive to that breed. In this simulation, AIVs, machines, jobs, and stocks are considered as four distinct breeds. Simulation time is measured in "ticks", with the assumption that 20 ticks represent one second. AIVs move along the corridors as shown in Fig. 3. The simulator interface is composed by:

– Sliders: for varying the number of jobs or transporters.
– Choosers: for selecting the problem instance, simulation environment, navigation type (shortest or fastest path), and priority type.

- Switches: for activating collision avoidance between robots and for recording simulation results.
- The monitor: for displaying simulation outputs.
- Buttons: The ***setup*** button initializes (or reinitializes) the simulation, while the ***go*** button launches it.
- The simulation environment represents the workshop layout. It displays a real-time visualization of the simulation.

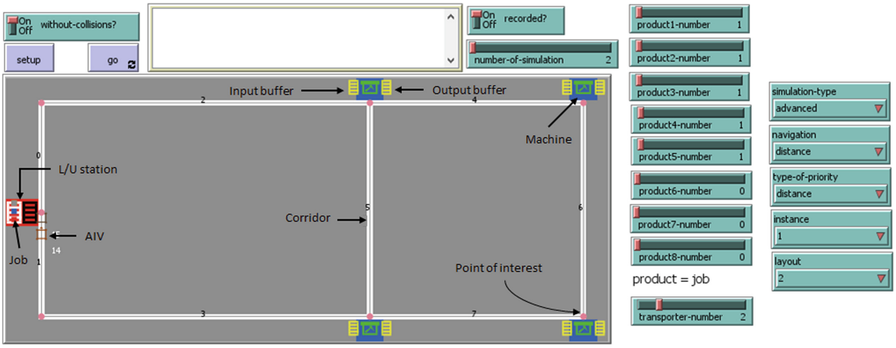


Fig. 3. Simulation interface

3.3 Simulation Model

In our model, the production and transportation tasks are not dependent, which means that AIVs can transfer some jobs while machines are processing others. However, for a job to be processed by a machine, it must first be transported by a robot. Therefore, the problem involves four subtasks:

- Vehicle scheduling: determines which jobs will be transported by a robot and the order in which the jobs will be transported.
- Machine scheduling: determines which jobs will be processed by a machine and the order in which the jobs will be processed.
- Vehicle routing: determines the path a robot will take while carrying out its task.
- Vehicles and machines synchronization: to ensure that precedence constraints and logical sequences are respected.

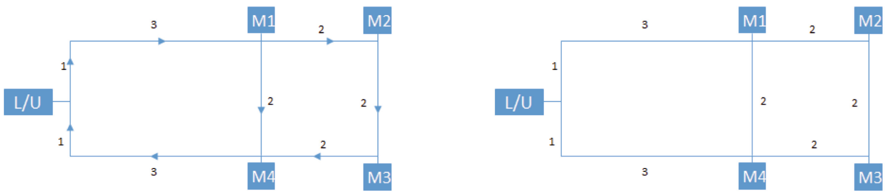
In the case of predictive scheduling simulation, transportation and production tasks are generated offline after an optimization process. The simulator therefore takes the result of this optimization as input and simulates it. However, in the case of dynamic scheduling simulation, the simulator is embedded with a dynamic scheduling algorithm. As a result, transportation and production tasks are generated dynamically, step by step, throughout the simulation.

3.4 Layout and Instances

Experiments are conducted using the well-known benchmark instances proposed by Bilge and Ulusoy [3]. They proposed four different layouts of the job shop, each consisting of a load/unload (L/U) station and four machines. The L/U station is used as a storage area for all jobs before they are processed (raw materials) and after they have been completed (finished products). Transportation tasks are carried out by two identical uni-charge AGVs. All transportation tasks begin and end at the L/U station. Each layout has a unique travel orientation, travel times, and L/U station and machine locations. For our part, we have replaced AGVs with AIVs. As a result, AIVs don't need to follow a unique travel orientation. They're intelligent enough to plan their own routes. This change has a direct impact on travel times, as shown in Table 2. Note that the values in Table 2b represent the minimum time required to travel between two locations.

Bilge and Ulusoy also proposed 10 different job sets, each of which consists of 5 to 8 jobs. Jobs are made up of several operations that must be performed on specific machines, and each operation has a corresponding processing time. The proposed test instances are denoted as "EX$\alpha\beta$", where α and β represent the job set and the layout, respectively. It is important to note that both travel times and processing times are measured in seconds (Fig. 4).

For this paper, we have limited our experiments to layout 2.



(a) Original layout 2 (b) Layout 2 with AIVs

Fig. 4. Layout 2

Table 2. Travel times

	L/U	M1	M2	M3	M4
L/U	0	4	6	8	6
M1	6	0	2	4	2
M2	8	12	0	2	4
M3	6	10	12	0	2
M4	4	8	10	12	0

(a) Layout 2: original travel times

	L/U	M1	M2	M3	M4
L/U	0	4	6	6	4
M1	4	0	2	4	2
M2	6	2	0	2	4
M3	6	4	2	0	2
M4	4	2	4	2	0

(b) Layout 2: AIVs travel times

3.5 Priority Management System

The priority management system is a rule-based system based on four criteria:

– Distance: The AIV closest to its destination has priority.
– Battery: The AIV with the lowest battery level has priority. In this case, the robots' battery levels gradually decrease as they perform their tasks. It is assumed that the battery level decreases at each simulation time step by 0.0025, 0.00125, and 0.0005 percent of the complete charge, respectively, when the AIVs are at cruising speed, when they reduce speed, and when they are at a complete stop. Nevertheless, robot batteries are initially 100% charged, and have sufficient energy to complete all their tasks.
– Starting time: The AIV that starts its current task earliest has priority.
– Random: AIVs draw a token at random to determine who has priority.

3.6 Experimental Protocol

For our experiments, we used the results obtained with the VNS method by Abderrahim *et al.* [1] presented in [14]. We are only interested in the results obtained considering collision avoidance. We then re-simulated these schedules by replacing AGVs with AIVs, while varying the priority management criterion to study its influence on makespan. We ran each simulation 50 times and recorded the average and the standard deviation. Layout 2 was chosen for the experiments because it presents several interesting challenges for the AIVs, such as deadlocks, intersections, and multiple path alternatives. These challenges provide opportunities to evaluate the performance of different priority strategies. Moreover, We investigated the impact of using shortest or fastest paths for AIV navigation on simulation results. We recorded the average time lost due to collision avoidance maneuvers on each path section (corridors) and used these averages to update the fastest path navigation method.

4 Results and Discussions

The results of the experiments are shown in Table 3. The first column lists the problem instances. The second column refers to the results in [14]. The other columns show the simulation results for the different priority criteria that were adopted. The results presented are the average of the makespan recorded after 50 runs, followed by the standard deviation in brackets.

Overall, AIVs outperform AGVs due to their ability to not follow pre-defined paths, which saves time. However, moving along the shortest path also increases the likelihood of encounters, necessitating collision avoidance maneuvers. Our measurements indicate an average collision avoidance time of 0.8 s for AIVs. Additionally, as noted in [14], lengthy waiting times can disturb other AIVs' activities, particularly evident in EX82, where robots experience significant delays before proceeding with subsequent tasks.

When comparing the different priority criteria, the distance criterion proves most efficient, allowing robots closer to their destinations to complete tasks promptly. This criterion gains further importance when considering deadline constraints for job deliveries. The battery criterion effectively prioritizes the AIV with the lowest battery level, enabling it to complete as many tasks as possible before depleting its energy. The starting time and random criteria, however, yield mixed results. The starting time criterion may prioritize a robot farther from its destination, and the random criterion may not always be relevant.

Furthermore, in general, navigation type has a negligible impact on the makespan. However, it was observed an increase in the number of avoided collisions with fastest path navigation compared to shortest path navigation as presented in Table 4. Similarly, some corridors became more collision-prone with fastest path navigation, because robots always choose the fastest path. In contrast, with shortest path navigation, robots choose a random path if multiple paths have the same length, which can help to distribute traffic more evenly. However, it is important to note that these experiments were conducted with only two AIVs and no external disturbances (moving obstacles or human operators) in order to comply with the benchmark instances [3]. The introduction of external disturbances and/or an increase in the robot fleet could produce different results.

Table 3. Experiment results

Instances	Results from [14]	Priority criteria			
		Distance	Battery	Starting time	Random
EX12	99	81.5 (1.7)	82.0 (1.3)	82.2 (1.6)	80.3 (2.2)
EX22	82.2	83.0 (0.0)	83.0 (0.0)	82.8 (0.0)	83.0 (0.0)
EX32	95	89.3 (0.1)	89.8 (0.3)	89.1 (0.2)	90.0 (0.3)
EX42	109	94.7 (0.8)	95.1 (0.7)	95.1 (1.0)	94.9 (1.7)
EX52	84	70.0 (0.4)	70.0 (0.4)	70.1 (0.4)	70.3 (0.5)
EX62	102.4	102.8 (0.0)	102.8 (0.2)	102.8 (0.1)	102.8 (0.0)
EX72	101	97.2 (0.3)	97.0 (0.3)	97.0 (0.2)	97.1 (0.3)
EX82	155.3	158.3 (1.2)	160.0 (1.0)	159.7 (0.9)	158.0 (0.8)
EX92	106.1	102.6 (1.7)	100.6 (0.7)	101.6 (0.8)	101.2 (1.3)
EX102	145	144.7 (0.1)	144.2 (0.4)	144.5 (0.3)	144.2 (0.4)

Table 4. Navigation type analysis

		Corridors of layout 2							
		0	1	2	3	4	5	6	7
nb. of collisions	Distance	96	404	0	0	159	81	0	873
	Time	105	395	0	0	0	500	0	1000
freq. of collisions	Distance	6%	25%	0%	0%	10%	5%	0%	54%
	Time	5%	20%	0%	0%	0%	25%	0%	50%

5 Conclusion

This paper addresses job-shop scheduling and Autonomous Intelligent Vehicles (AIV) path-planning problems through simulation. The transition from Autonomous and Guided Vehicles (AGVs) to AIVs solves several problems, such as the need for a dedicated environment. AIVs are more intelligent, flexible, and collaborative mobile robots that can navigate in spaces with mobile and/or unexpected obstacles. Therefore, their use is relevant in Industry 5.0, where humans and robots work together. Moreover, the results of the experiments conducted show that, in most of the cases, switching from AGV to AIV improves the makespan (more than 5 s on average).

The simulator presented in this paper is a valuable tool for the study of AIV scheduling and path planning in job-shop production systems. It is able to generate efficient schedules for both predictive and dynamic scheduling, and the priority management algorithm is effective in resolving conflicts between AIVs. This work has a number of implications for the use of AIVs in production systems. First, the simulator can be used to evaluate the performance of different scheduling and path-planning algorithms. Second, AIVs ability to navigate in more complex workshop layouts can be evaluated. Third, it can prepare for the transition to Industry 5.0 by considering the human factor.

Future work will focus on expanding the experiments to other layouts and instances, as well as, improving the performance of the simulator by incorporating more features, such as a battery management system. Humans will be integrated as the fifth agent to study their impact on the scheduling of AIVs.

References

1. Abderrahim, M., et al.: Bi-local search based variable neighborhood search for job-shop scheduling problem with transport constraints. Optim. Lett. **16**, 255–280 (2020)
2. Benhafssa, A.M., et al.: Optimizing energy-conscious dynamic flexible job shop scheduling: Multi-agent simulation approach. In: 2021 1st International Conference on Cyber Management and Engineering (CyMaEn), pp. 1–6 (2021). https://doi.org/10.1109/CyMaEn50288.2021.9497301
3. Bilge, Ü., Ulusoy, G.: A time window approach to simultaneous scheduling of machines and material handling system in an fms. Oper. Res. **43**(6), 1058–1070 (1995)
4. Cronin, C., Conway, A., Walsh, J.: 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 (2019). https://doi.org/10.1109/ISSC.2019.8904920
5. Destouet, C., et al.: Flexible job shop scheduling problem under Industry 5.0: a survey on human reintegration, environmental consideration and resilience improvement. J. Manuf. Syst. **67**, 155–173 (2023). https://doi.org/10.1016/j.jmsy.2023.01.004
6. Drótos, M., et al.: Suboptimal and conflict-free control of a fleet of agvs to serve online requests. Comput. Ind. Eng. **152**, 106999 (2021)

7. Ham, A.: Transfer-robot task scheduling in job shop. Int. J. Prod. Res. **59**(3), 813–823 (2021)
8. Hu, H., et al.: Deep reinforcement learning based agvs real-time scheduling with mixed rule for flexible shop floor in industry 4.0. Comput. Ind. Eng. **149**, 106749 (2020). https://doi.org/10.1016/j.cie.2020.106749
9. Martin, L., et al.: Effect of human-robot interaction on the fleet size of aiv transporters in fms. In: 2021 1st International Conference on Cyber Management and Engineering (CyMaEn), pp. 1–5 (2021). https://doi.org/10.1109/CyMaEn50288.2021.9497273
10. Perez-Grau, F.J., et al.: Introducing autonomous aerial robots in industrial manufacturing. J. Manuf. Syst. **60**, 312–324 (2021)
11. Rapalski, A., Dudzik, S.: Energy consumption analysis of the selected navigation algorithms for wheeled mobile robots. Energies **16**(3), 1532 (2023). https://doi.org/10.3390/en16031532
12. Reith, K.B., Rank, S., Schmidt, T.: Conflict-minimal routing for free-ranging transportation vehicles in in-house logistics based on an a-priori lane design. J. Manuf. Syst. **61**, 97–111 (2021). https://doi.org/10.1016/j.jmsy.2021.07.019
13. Sahnoun, M., et al.: Optimization of transportation collaborative robots fleet size in flexible manufacturing systems. In: 2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO), pp. 1–5. IEEE (2019)
14. Sanogo, K., et al.: A multi-agent system simulation based approach for collision avoidance in integrated job-shop scheduling problem with transportation tasks. J. Manuf. Syst. **68**, 209–226 (2023). https://doi.org/10.1016/j.jmsy.2023.03.011
15. Sanogo, K., et al.: Multi-agent simulation for flexible job-shop scheduling problem with traffic-aware routing. In: Borangiu, T., et al. (eds.) Service Oriented, Holonic and Multi-agent Manufacturing Systems for Industry of the Future, pp. 573–583. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-99108-1_41
16. Xu, Y., et al.: A simulated multi-objective model for flexible job shop transportation scheduling. Ann. Oper. Res. **311**, 899–920 (2020)
17. Zaidi, L., Bettayeb, B., Sahnoun, M.: Optimisation and simulation of transportation tasks in flexible job shop with muti-robot systems. In: 2021 1st International Conference on Cyber Management and Engineering (CyMaEn), pp. 1–6. IEEE (2021)