Modular Framework for Crowd Simulation “Menge” from a Production Warehouse Simulation Perspective

Irving Bruno López, Rafael Saldaña, and Gilberto Rivera

Abstract The topic of warehouse optimization is a critical and popular topic within the scientific literature [1, 4, 5, 7]. Particularly, order-picking optimization presents several profound difficulties: modeling, analysis, and data representation. Althoug these are necessary and obvious requirements for any research project, the visualization is an important aspect that has its challenges and complications. Menge is an open-source platform for crowd simulation. It provides an extensible common for research and development. In the present study, it is shown how the simulation platform known as Menge was adapted as a possible complement to the optimization process for the simulated visualization of its results. This chapter presents a solution using Menge to separate the modeling for the routing algorithm from visualization and simulation programming. The purpose of this work is to establish a starting point of industrial simulation in collaboration with algorithms for order-picking optimization in warehouses.

Keywords Production Warehouses · Menge simulation · Order-Picking Policies

1 Introduction

There are publications in the scientific literature that deal with the problem of Order picking; however, a few of them include the display of the simulation of the routes suggested by optimization algorithms. Some examples to model production warehouses for simulation are reported by [1], as shown in Fig. 1.

Because creating a simulation model is a separate and exhausting job, [6] proposes a method to create effective simulation models. As shown in Fig. 2.

I. B. López · G. Rivera (✉)
Universidad Autónoma de Ciudad Juárez, Av. Hermanos Escobar Omega, 32410 Cd Juárez, Chihuahua, Mexico
e-mail: gilberto.rivera@uacj.mx

R. Saldaña
Instituto Tecnológico de Ciudad Juárez, Chihuahua, Mexico

A. Ochoa-Zezzatti et al. (eds.), *Technological and Industrial Applications Associated with Intelligent Logistics*, Lecture Notes in Intelligent Transportation and Infrastructure,
https://doi.org/10.1007/978-3-030-68655-0_16

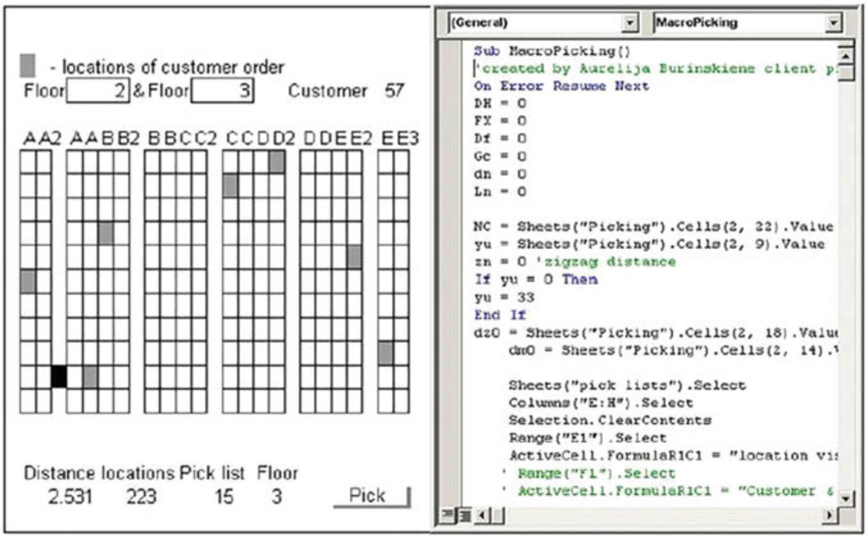

Fig. 1 Simulation model:simulation of order and routing algorithm [1]

A micro-simulation model for an intelligent logistics platform is published by [4] to optimize the transportation tasks and, consequently, reduce costs. These authors establish a flowchart for common logistics of a receiving area and, from this flowchart, a mathematically and visually characterized model is created to display different optimized routes for the order-picking problem. The model obtained is shown in Fig. 3.

Similarly, other authors have reported the creation of tools to create layout models for measurable and verifiable results. Some examples of the latter are the studies published by Sharp and Krueger [5]. A model using Petri nets is shown in Fig. 4.

Defining the layout of production warehouses is critical because warehouse productivity depends on the item locations within the warehouse. The simulation of the picking routes helps to determine the design and control of the warehouse. This is achieved by interrelating three aspects: the type of systems, the layout, and the policies to control all operational processes [7].

Menge is a cross-platform, extensible, modular framework for simulating pedestrian movement in a crowd. Menge creates abstractions for related problems and provides a plug-in architecture so that a novel simulator can be dynamically configured by connecting built-in and bespoke implementations of solutions [3]. The use of Menge is intended to simplify the work required to visualize the routes resultant from optimization algorithms in production warehouses.

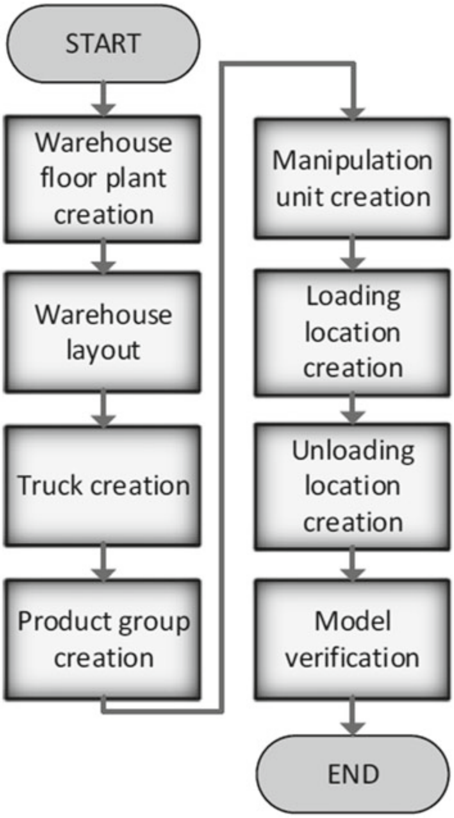

Fig. 2 Simulation model creation procedure [6]

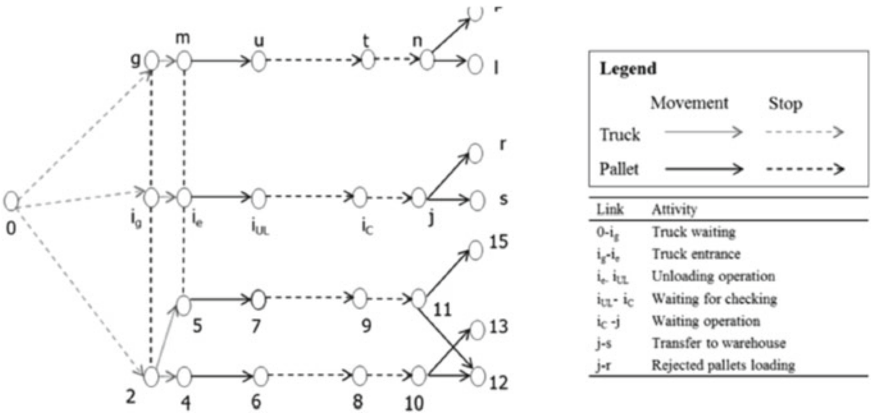

Fig. 3 Receiving Area model [4]

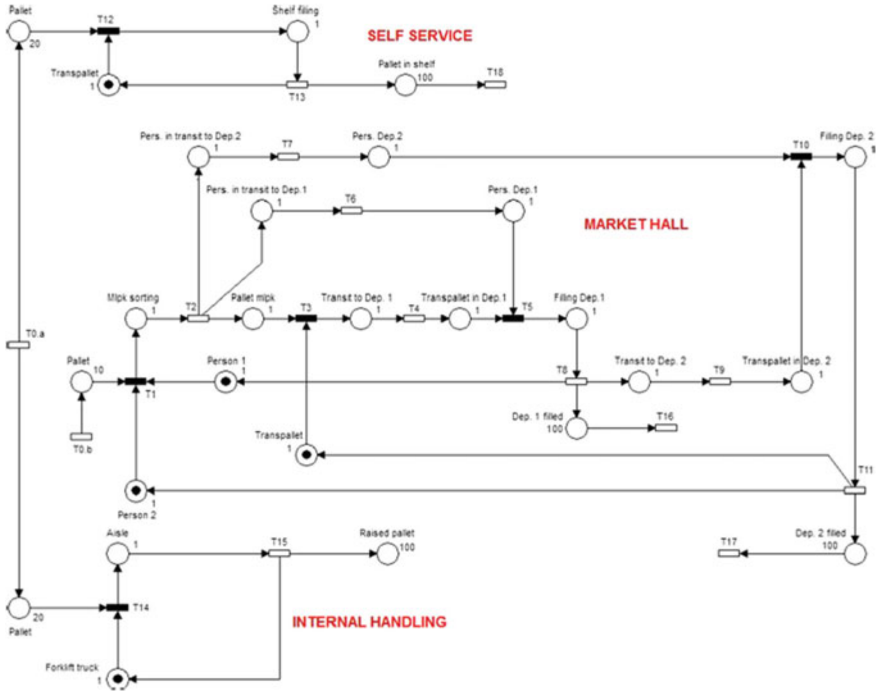

Fig. 4 The Petri Net model of a logistic system [5]

2 Materials

The equipment (software and materials) used for implementing the simulations are the following: Computer equipment to run the simulation, Operating System: Windows 10 Home Single Language 64-bit x64-based processor, System Manufacturer: Lenovo, Processor: AMD Ryzen 3 2200U with Radeon Vega Mobile Gfx 2.50 GHz, Memory: 8 GB RAM and OS Memory: 7.78 GB RAM.

2.1 Software

Visual Studio 2017 Community Edition. Menge: A framework for modular pedestrian simulation for research and development, free code [3].

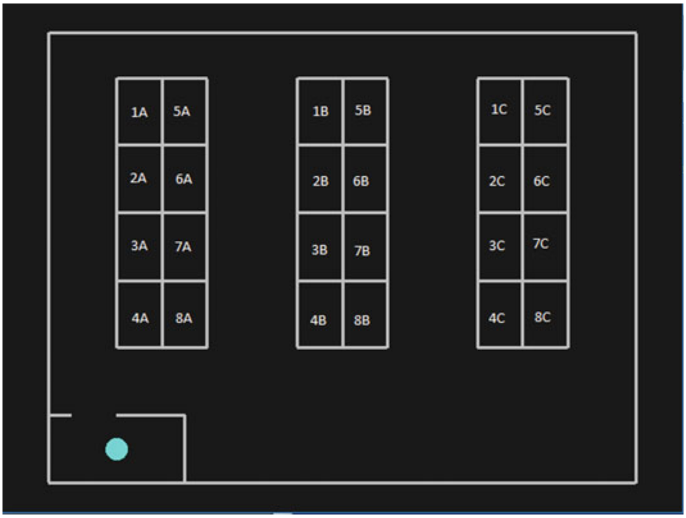

Fig. 5 Menge Scenario for a small production warehouse

2.2 Scenarios

The scenario is represented by a warehouse with three racks and one block. Each rack has eight available spaces or locations, four per aisle. A single picker is represented as an agent, which will explore the picking aisles of the warehouse following a specific list of collection points or predetermined goals (Fig. 5).

Some orders were generated with products located at the points 4A, 5A, 4B, 7B, 1C, and 6C, each of which will create a picking node. In this case, in the layout, there are eight crossing nodes (nodes CN1 to CN8) and a single node (DN1) of departure and arrival for the picking agent. Figure 6 shows the representative sequence of the warehouse modeled in Menge with the picking, crossing and destination nodes. The picking nodes are represented in green, the crossing nodes in red and the destination with a blue node, respectively.

2.3 Simulation

The simulation was performed using Menge software by modifying an open-source example called steerbench¥curves.xml for setting multiple and consecutive goals and another open-source example called globalNavSwap-roadmap.xml [2] to determine a roadmap as a guide for the simulation agents. These examples were studied and adapted to the conditions of the scenario as well as the objectives to be achieved during the simulation of an order-picking activity in production. The way of use Menge is like a black box. The Menge documentation shows how to adapt the setting

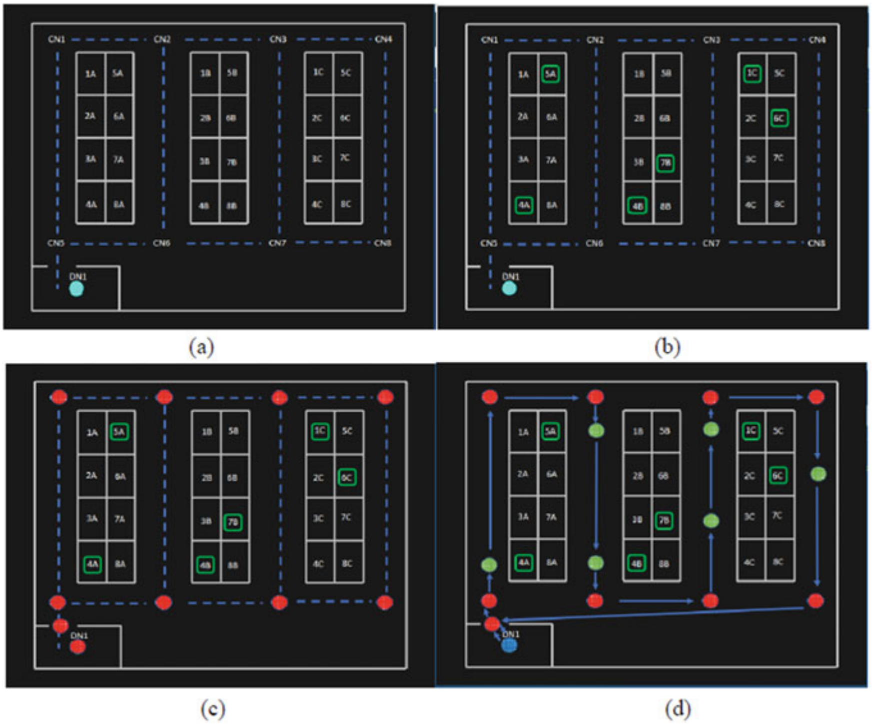

Fig. 6 Menge goals determination case based. **a** Cross nodes labeling. **b** Picking nodes locations. **c** Cross nodes model map. **d** Cross, picking and depot nodes for an order instance

files: the most relevant are the Scene Specification, Behavior Specification, and View Specification.

The obstacles of the scenario and the agents are defined in the Scene Specification file. The goals, as well as the trajectories, are established in the Behavior Specification file. The View Specification was not modified for this project. The mid points serve as a guide for the agent; avoiding having direct contact with the obstacles (following a specific notation in the file graph.txt). This grid of mid points is called roadmap. For this simulation, the S-shape policy was used. The final scenario was set as shown in Fig. 7.

For the agent to follow a pre-established route, it was established that the agent would always leave the depot towards the nearest crossing node (CN5 crossing node), then it will go to the nearest picking point until it reaches the next crossing point, when it arrives at the next picking aisle it will repeat the operation until it has visited all the picking points and, finally, go back to the delivery point. As observed, this simulation considers multiple objectives, which must be reached by the agent. These objectives are declared as a group of points in the behavior configuration file following this pattern: <Goal type="point" id="0" x="-10" y="-8.5"/> .

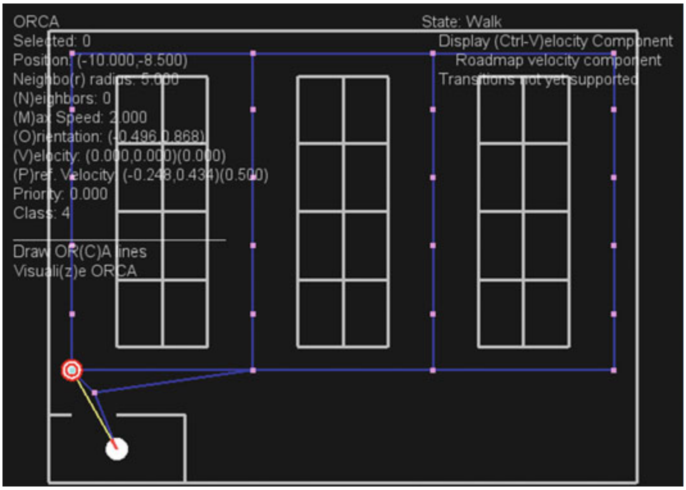

Fig. 7 The agent is heading towards the first goal (simulation parameters are shown)

A group of goals determined by the picking order used to validate the simulation is as follows:

```
<Goal type="point" id="0" x="-10" y="-8.5"/> <!--INICIO¥FIN-->
<Goal type="point" id="1" x="-11" y="-6"/> <!--INTERMEDIO-->
<Goal type="point" id="2" x="-12" y="-5"/> <!--PUNTO INTERMEDIO ABAJO A-->
<Goal type="point" id="3" x="-12" y="-2.5"/> <!--ESTANTE 4A-->
<Goal type="point" id="4" x="-12" y=".5"/> <!--ESTANTE 3A-->
<Goal type="point" id="5" x="-12" y="3.5"/> <!--ESTANTE 2A--> 5
<Goal type="point" id="6" x="-12" y="6.5"/> <!--ESTANTE 1A-->
<Goal type="point" id="7" x="-12" y="9"/> <!--PUNTO INTERMEDIO ARRIBA A-->
```

There is a goal for each point of interest (picking node, crossing node or destination node); however, the route will be defined by the content of the orders. Only the picking points sorted by number and letter are selected and inserted into the Behavior.xml configuration file, relating the goals to the trajectories. In Menge, these trajectories are known as walks. Below, an example of the first walks created for this order-picking case is presented:

```
<State name="Walk" final="0" >
<GoalSelector type="explicit" goal_set="0" goal="2" />
<VelComponent type="road_map" weight="1.0" file_name="ruta.txt"/>
</State>
<State name="Walk2" final="0" >
<GoalSelector type="explicit" goal_set="0" goal="3" />
<VelComponent type="road_map" weight="1.0" file_name="ruta.txt"/>
</State>
<State name="Walk3" final="0" >
<GoalSelector type="explicit" goal_set="0" goal="7" />
<VelComponent type="road_map" weight="1.0" file_name="ruta.txt"/>
</State>
```

The results obtained for this agent's walk for an order of six items located in different locations were 3.6s consistently after 100 runs. The roadmap was defined using the file graph.txt for the simulation. For this simulation, 26 guide points and 29 interconnections were defined. The guide points were defined by specifying the number of points they would connect to and their location within the layout plane. The interconnections indicate which guide point connects to another. An example of this configuration is shown below:

```
26
1 -10 -8.5
3 -11 -6
3 -12 -5
...
29
0 1
1 2
1 13
...
```

The agent identifies its immediate target and orients its direction vector as directly as possible. If the visibility towards its assigned target does not allow a direct connection, the agent goes to the mid point furthest from the Roadmap and closest to the goal. The following set of images shows a simulation run. In Fig. 8, the agent goes to the node AN1; in Fig. 9, the agent is heading towards CN7 (Cross node 7) from PN4B (Picking node 4B).

In Fig. 10, the direction vector assigned is the one indicated by the yellow line. This line is automatically calculated by Menge as the fastest way between the agent and its destination considering the guide points of the route map. The agent uses the points of the route map only as a reference, does not strictly follow the blue lines of the route map unless they coincide with the direction between the current position and the next goal.

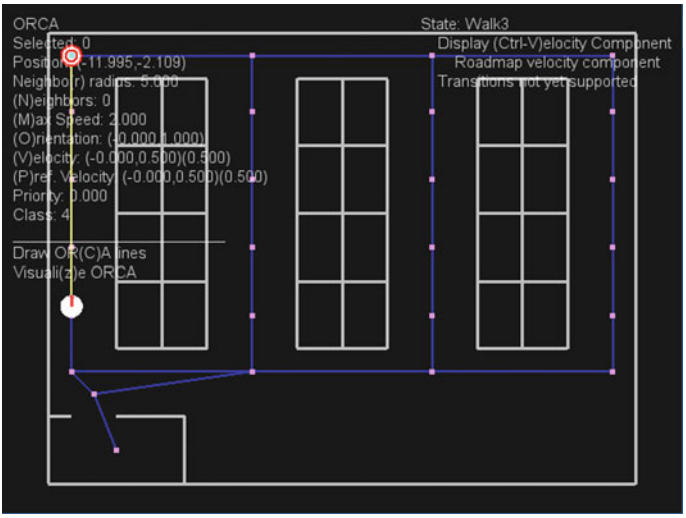

Fig. 8 The agent is heading towards AN1 (Cross node 1) from PN4A (Picking node 4A)

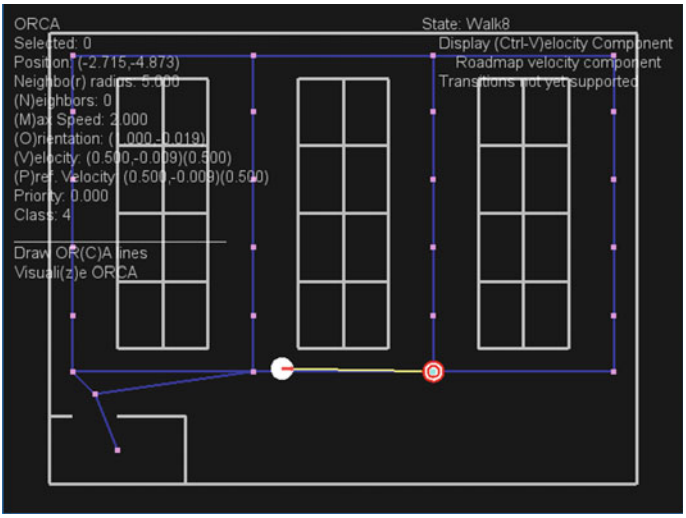

Fig. 9 The agent is heading towards CN7 (Cross node 7) from PN4B (Picking node 4B)

3 Conclusions and Future Work

Here, a notable difference with the most common approaches for the Menge simulation platform is presented. For a specific layout a few points are commonly determined for many people or agents in the simulation; but, in this case, a sequence

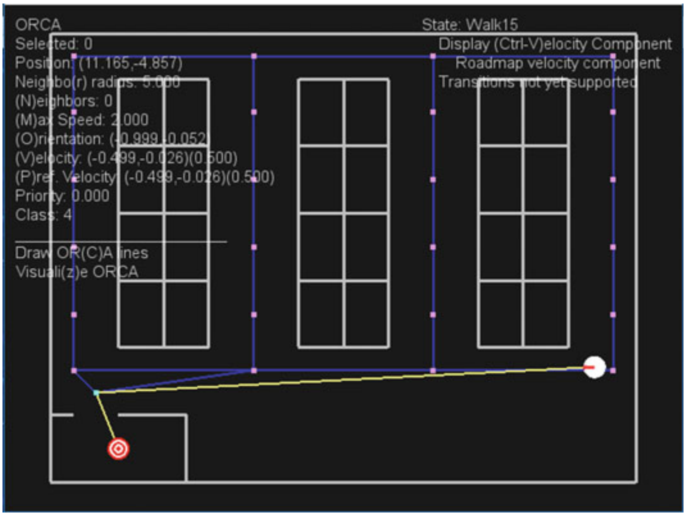

Fig. 10 The agent is heading towards destination DN1 (Depot node 1)

of points are determined for a single agent. This application shows the flexibility of parameter configuration that the Menge tool allows for a wide variety of applications [7].

By using Menge, it was possible to visualize a complete route for a randomly assigned picking order. Menge allows the user to see clearly and fast routes for different orders without the need to understand the optimization model. Menge simulations also do a visualization of other algorithms. The visualization of the simulated route coincided with the assigned route without abrupt changes of direction and traffic blockages. Once the terms used in the specification files are understood, Menge's flexibility allowed to quickly define the layout of the production warehouse, agent and route map. An important aspect that was not yet considered in this work is the lack of picking time for each product.

As future work, it would be proposed the creation of a visual interface that allows the following features:

(1) Visual definition of the warehouse layout.
(2) The definition of a folder where the picking orders to be simulated are automatically read.
(3) The inclusion of more picking agents.
(4) The definition of different capacities for different groups of agents.

We conclude that Menge is a tool that facilitates the visualization of routes and trajectories of agents, making an adequate complement for academics and researchers that are focused also on the optimization of routes and not only on the difficulties of visualization.

References

1. Burinskienė A, Davidavičienė V, Raudeliūnienė J (2018) Simulation and order picking in a very-narrow-aisle warehouse aisle warehouse. Econo Res-Ekonomska Istraživanja, 1–16. https://doi.org/10.1080/1331677X.2018.1505532
2. Curtis S, Best A, Manocha D (2013) MENGE
3. Curtis S, Best A, Manocha D (2016) Menge: a modular framework for simulating crowd movement 40: 1–40. https://doi.org/10.17815/CD.2016.1
4. Gattuso D, Cassone GC, Pellicano DS, Gattuso D, Cassone GC, A, D. S. P., Gattuso D (2017) A micro-simulation model for an intelligent logistics platform: specification and calibration results, 8312(April). https://doi.org/10.1080/16258312.2014.11517358
5. Gerini C, Sciomachen A (2018) Evaluation of the flow of goods at a warehouse logistic department by Petri Nets. Flexible Serv Manuf J. https://doi.org/10.1007/s10696-018-9312-3
6. Šaderová J, Marasová D, Galliková J (2018). Simulation as logistic support to handling in the warehouse: case study 7(1): 112–117. https://doi.org/10.18421/TEM71-13
7. Taylor P, Roodbergen KJ, Vis IFA, Jr, GDT (2014) Simultaneous determination of warehouse layout and control policies, (April 2015), 37–41. https://doi.org/10.1080/00207543.2014.978029