COMPUTER RESEARCH AND MODELING
2026 VOL. 18 NO. 2 P. 423–438
DOI: 10.20537/2076-7633-2026-18-2-423-438

MODELS IN PHYSICS AND TECHNOLOGY

UDC: 519.8

Simulation of fully automated warehouse with deep storage racks

A. V. Khelvas[a], **K. K. Pankratov**, **T. S. Afanasenko**[b], **Sh. M. Gadzhimirzayev**[c], **A. A. Saidov**[d], **R. A. Pashkov**[e], **S. A. Strelnikova**[f]

Moscow Institute of Physics and Technologies,
9 Institutskii per., Dolgoprudnii, 141701, Russia

E-mail: [a] hel@cos.ru, [b] afanasenko.ts@phystech.edu, [c] gadzhimirzaev.shm@phystech.edu, [d] saidov.aa@phystech.edu, [e] ruslan.pashkoff@gmail.com, [f] strelnikowa.sa@phystech.edu

Received 21.05.2025.
Accepted for publication 19.02.2026.

This article presents a model of a fully automated warehouse with deep storage racks designed for boxed goods storage. The study focuses on optimizing warehouse operations through discrete multiagent simulation of shuttle movements for pallet loading and unloading tasks. The authors investigate various product placement strategies, including the Nearest Channel Positioning Algorithm (NCPA), Most Empty Channel Group Placement (MECGP), and Most Filled Channel Group Placement (MFCGP), while analyzing optimal routing schemes for the given warehouse topology.

A key contribution is determining the optimal number of shuttles to maximize warehouse throughput. Simulation results demonstrate that increasing the number of robots beyond 15 does not significantly improve efficiency due to increased route collisions. The study also examines 24-hour warehouse occupancy dynamics, revealing optimal storage utilization levels.

The developed model enables performance evaluation and optimization of task distribution among robots to minimize order processing time. Future research directions include implementing machine learning techniques to further enhance warehouse management systems.

Keywords: robotic warehouse, boxes placement optimization, multiagent simulation

Citation: *Computer Research and Modeling*, 2026, vol. 18, no. 2, pp. 423–438.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2026 Т. 18 № 2 С. 423–438
DOI: 10.20537/2076-7633-2026-18-2-423-438

Ки&М

МОДЕЛИ В ФИЗИКЕ И ТЕХНОЛОГИИ

УДК: 519.8

Моделирование полностью роботизированного склада со стеллажами глубокого хранения

А. В. Хельвас[a], **К. К. Панкратов, Т. С. Афанасенко**[b], **Ш. М. Гаджимирзаев**[c], **А. А. Саидов**[d], **Р. А. Пашков**[e], **С. А. Стрельникова**[f]

Московский физико-технический институт,
Россия, 141701, Долгопрудный, Институтский пер., д. 9

E-mail: [a] hel@cos.ru, [b] afanasenko.ts@phystech.edu, [c] gadzhimirzaev.shm@phystech.edu,
[d] saidov.aa@phystech.edu, [e] ruslan.pashkoff@gmail.com, [f] strelnikowa.sa@phystech.edu

Получено 21.05.2025.
Принято к публикации 19.02.2026.

В данной статье рассматривается модель полностью роботизированного склада с глубокими стеллажами, предназначенного для хранения коробочных товаров. Основное внимание уделено оптимизации работы склада за счет дискретного мультиагентного моделирования движения шаттлов, выполняющих задачи по отгрузке и размещению коробок. Авторы исследуют различные стратегии размещения товаров в зонах склада, включая алгоритмы NCPA (Nearest Channel Positioning Algorithm), MECGP (Most Empty Channel Group Placement) и MFCGP (Most Filled Channel Group Placement), а также анализируют оптимальные схемы маршрутизации для заданной топологии.

Ключевым аспектом работы является определение оптимального количества шаттлов, обеспечивающего максимальную производительность склада. Результаты моделирования показывают, что увеличение числа роботов свыше 15 не приводит к значительному росту эффективности из-за учащения коллизий на пересечениях маршрутов. Кроме того, исследована динамика заполнения склада в течение 24 часов, что позволило выявить оптимальный уровень загруженности хранилища.

Разработанная модель позволяет не только оценивать производительность склада, но и оптимизировать распределение задач между роботами, минимизируя время обработки заказов. В перспективе планируется внедрение методов машинного обучения для дальнейшего улучшения управления складскими процессами.

Ключевые слова: роботизированный склад, оптимизация размещения коробок, мультиагентное моделирование

Introduction

Automated box-type warehouses are becoming increasingly common worldwide. The main customers for such warehouses are large retail chains, as well as manufacturers in the food industry, electronics, furniture, household goods, and other sectors.

These warehouses typically receive goods on mono-pallets (where all boxes on a pallet are identical), depalletize them (separate them into individual boxes), and then move these boxes one by one to storage racks using stacker cranes or lifts and mobile robots (shuttles). On the output side, the warehouse system assembles mixed pallets (consisting of different boxes) according to orders from retail stores or external customers.

The key performance indicators for such a robotic warehouse system are:

- Throughput (the number of boxes moved from racks to palletizing per hour/day);
- Storage density of boxes (usually measured in cubic meters of goods or a number of boxes per square meter of warehouse area or cubic meter of storage volume);
- Parameters of box stacking on mixed pallets (percentage of pallet space utilization, average pallet height, average number of boxes per pallet, etc.);

This work is based on a presentation at the 5th IFSA Winter Conference on Automation, Robotics, and Communications for Industry 4.0/5.0 (ARCI'2025) [Pankratov et al., 2025].

High storage density is achieved through the proposed concept of deep storage channels. Unlike most existing systems where racks contain only one box per unit length (or a small number — 2 or 3 identical boxes), the presented model consists of racks with deep channels containing multiple different boxes.

Such a system requires optimization of box placement by type and management of a swarm of mobile robots (shuttles) performing multiple tasks in the warehouse simultaneously.

The optimization goal is to maximize the availability of each product at any given time. In the simplest case this means that at least one box of each product should be first in line to exit its channel. A more correct approach involves optimizing the total processing time for a certain number of outgoing pallets with given characteristics. For different warehouses, the properties of pallet sets may vary significantly.

General approaches to warehouse modeling are presented in works [Baker, Canessa, 2009; Chayaphum et al., 2019].

Figure 1 shows the arrangement of boxes in channels for a warehouse described above. The channels represent classic queues of N_{CH}^{k} boxes. Boxes of a certain type are in different "phases" — in different parts of channels, from entrances to the exits.

Outgoing pallets can have varying numbers of boxes. The distribution of box quantities for a real retail warehouse is shown in Fig. 3.

Literature review

Using the ProModel simulation language, the article [Macro, Salmi, 2002] developed a universal warehouse storage model. The model was successfully applied to analyze the capacity of warehouse spaces and rack efficiency in a medium-volume warehouse with a small number of stock keeping units (SKUs) and a medium-volume warehouse with a large number of SKUs. The model is scalable and can be modified to simulate any warehouse configuration, including selective racks, floor storage, push-back, flow-through, drive-in, and drive-through racks.

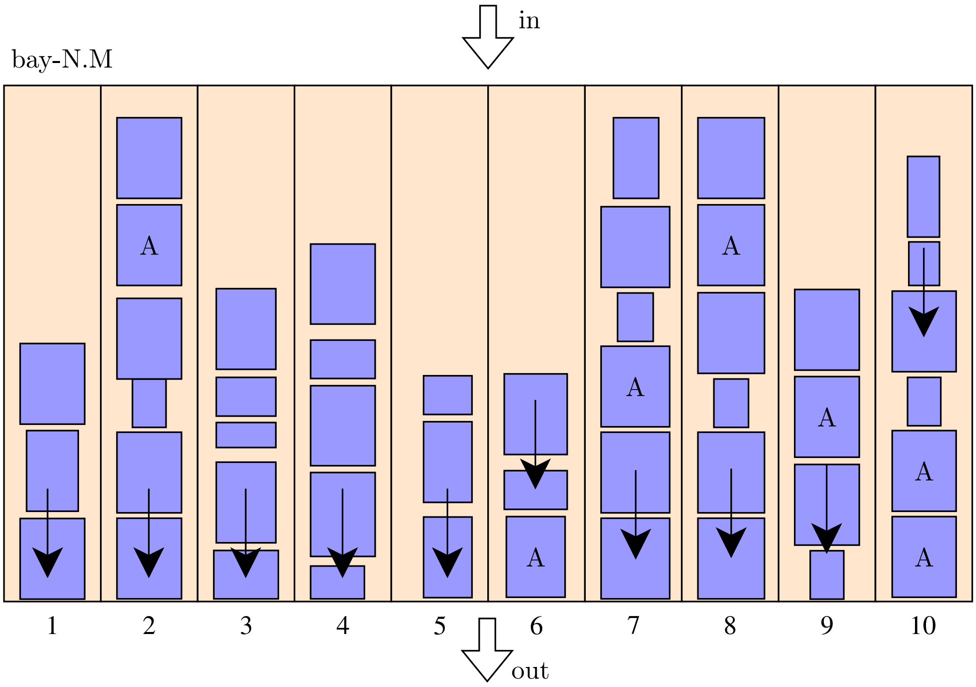

Figure 1. Placement of boxes in channels

The article [Saderova et al., 2022] examines the modeling of individual warehouse activities. The modeled activities include receiving goods, inspecting goods, and storing goods in the warehouse system on the inbound side. On the outbound side, these activities include order picking, packaging, and shipping. The main goal is to present a created simulation model for selected warehouse activities — goods receipt. This operation involves unloading goods from trucks, moving them to a physical receiving point where quantitative accounting is performed, and quality control of goods (goods acceptance) is carried out. Analysis of this activity preceded the creation of the simulation model under operating conditions. The study was carried out by observing and measuring forklift work cycles to unload pallets from trucks. The resulting analysis data were input data for the simulation model.

An important aspect of the described multiagent warehouse systems (including for scaling) is ensuring efficient operational movement planning. In this case, multiple agents move within a specified topology (represented by a graph) with a constraint on the number of agents occupying a single section simultaneously. The goal of modeling is to achieve maximum system performance and eliminate collisions (deadlock situations where a robot needs to be switched to manual control).

The work [van Gemund et al., 2010] thoroughly examines the problem of operational order delivery planning under uncertainty for an automated multiagent system. The authors present routing algorithms that avoid conflicts when multiple agents move.

In [Goodwin et al., 2010], the authors study the influence of forecasts on decision quality in planning warehouse operations.

The work [Gilya-Zetinov et al., 2019] addresses the problem of three-dimensional packing of boxes on pallets, which belongs to the category of NP-complete problems. Various heuristic methods for its solution are proposed, including a layer-by-layer approach and the use of a genetic algorithm. A method for evaluating packing efficiency using percolation and stability indicators is also developed. Experiments demonstrated that the accuracy of box dimension specifications has a minor impact on packing quality.

The article [Soloviev, Nomerchuk, 2023] considers a model for processing requests and distributing tasks among agents. A queueing system with scaling capabilities is presented. In this work (as in the warehouse presented later), an order (a pallet in our case) contains boxes of different SKUs. Several robots can process this order simultaneously. A robot's route is the path from a storage location to a loading point. The distribution of tasks among robots and the creation of a head server (which manages robots) were also considered. The described server bears a significant load as it assigns tasks to each robot, so many computational processes must be handled. Based on experiments and further analysis, the author demonstrated the effectiveness of these warehouse systems and proposed algorithms.

The work [Wurman et al., 2008] describes a warehouse management system that significantly improves productivity by making autonomous agents carry mobile racks with many stored goods. This work describes the first commercially available large-scale autonomous robotic system of its time.

The study [Maxemchuk, 2005] proposes a simple agent routing algorithm in Ad Hoc networks. An Ad Hoc network is a decentralized wireless network. This network has no permanent structure. Ad Hoc network devices connect on the fly, thereby changing a system topology. Each device (in our case, an agent) can exchange data via messages with other devices (other agents), without knowing in advance which device on the other side to send messages to, so that this information is developed dynamically. An interesting feature of the presented research is that agents do not require global network information for routing and exchange messages through nearby agents. The simplicity of this algorithm remains an advantage even in modern systems with frequent dynamic state changes (e. g. in warehouse systems).

The work [Rangel, 2019] describes a movement algorithm based on ant colonies and pheromone allocation. The advantage of this model is that it is almost completely autonomous.

The Cooperative A* algorithm is described, for example, in [Silver, 2005]. The Cooperative A* algorithm is a combination of A* and Cooperative Pathfinding algorithms. The idea is that, unlike simple pathfinding using A*, this algorithm considers information about other agents that may be encountered along the robot's path. Thus, this algorithm helps avoid path conflicts among agents. In the case of conflict, conflicting robots send messages to each other and, based on information exchange, reconfigure their routes.

The paper [Oitzman, 2021] describes an automated warehouse based on autonomous mobile robots (AMRs). The goal of the AMR application is to increase warehouse system productivity. The author describes the implementation of robots with lidars and their integration with a Warehouse Management System (WMS).

Warehouse description for modeling

To study the basic properties of the proposed warehouse configuration and develop approaches to optimal product placement and robot management, a model warehouse configuration with parameters listed in Table 1 was used.

Table 1. Warehouse parameters

	Parameter	Value
1	Size (sections)	5×5
2	Levels	1
3	Boxes per channel	10
4	Channels per section	10
5	Total storage locations	2500
6	Number of SKUs	150
7	Input conveyors	1
8	Output conveyors	1
9	Pallet size	1200×800

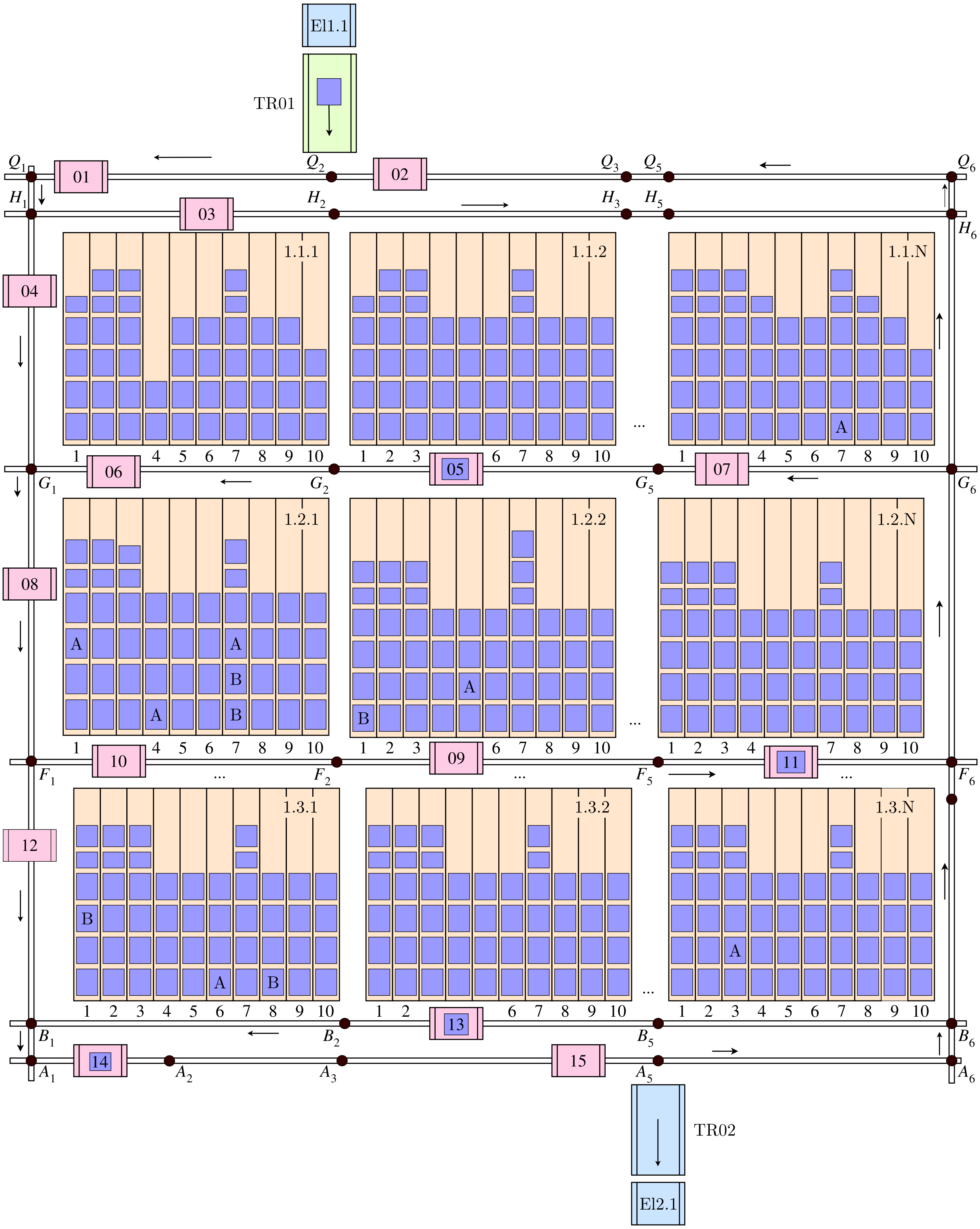

Figure 2. The warehouse geometry

The warehouse geometry is shown in Fig. 2.

Shuttle movement occurs on a rail system with two permitted directions: “North – South” and “West – East”. The rail system is described by a directed graph where movement along each edge is permitted in only one direction. A shuttle can be either on an edge or at a graph node. Only one shuttle can be on a graph edge at a time.

The modeling uses a pallet array obtained from a real set of pallets from a retail warehouse.

The duration of picking a box from a channel exit or loading it into a channel entrance is $\Delta T_1 = \Delta T_2 = 3s$.

Channel replenishment is done with mono-pallets based on the event "remaining boxes of SKU type less than $N_{SKU}^{\min} = N_{SKU}^{\max} \times 0.15$".

The formation of a task array for box movement by robots is not the subject of optimization in this task, but an accurate description of the data structure describing multiple tasks for robots represents an independent scientific result.

The simulator's output is a list of events for loading/unloading boxes and a separate report on pallet formation/disassembly.

Event types output in the box loading/unloading report:

1 — movebox2channel — moving a box from an input pallet to a channel;

2 — movebox2pallete — moving a box from a channel to an output pallet loader;

3 — moveboxCh2Ch — moving an interfering box from a channel exit to the entrance of another channel.

Event types output in the pallet processing report:

4 — startDepalletize — start of depalletizing;

5 — finishDepalletize — end of depalletizing;

6 — startPalletize — start of palletizing;

7 — finishPalletize — end of palletizing.

In this task, we begin modeling shuttle movement. It is necessary to develop a data structure for multiple storage racks (bay) with a FIFO box extraction algorithm.

A box interfering with access to a required box not at the channel exit is moved to the entrance of some channel. This adds a task to the task list.

Initial warehouse loading is done by filling the warehouse through depalletizing incoming pallets for each SKU. Initial warehouse filling assumes depalletizing one pallet for each SKU.

The following exceptional situations are handled in the software model implementation:

out-of-free-places — no available locations for placing boxes in channels;

no-boxes-for-SKU — the required SKU is not present in the warehouse space.

Input data description

A data set (https://disk.yandex.ru/d/LOPkNGkVvgEPBw) containing descriptions of 436 real mixed-SKU pallets was used as initial data input. This data set was truncated to contain only 100 SKU to use in simulations of a smaller warehouse model.

The distribution describing the number of boxes per input (single-SKU) pallet is shown in Fig. 3.

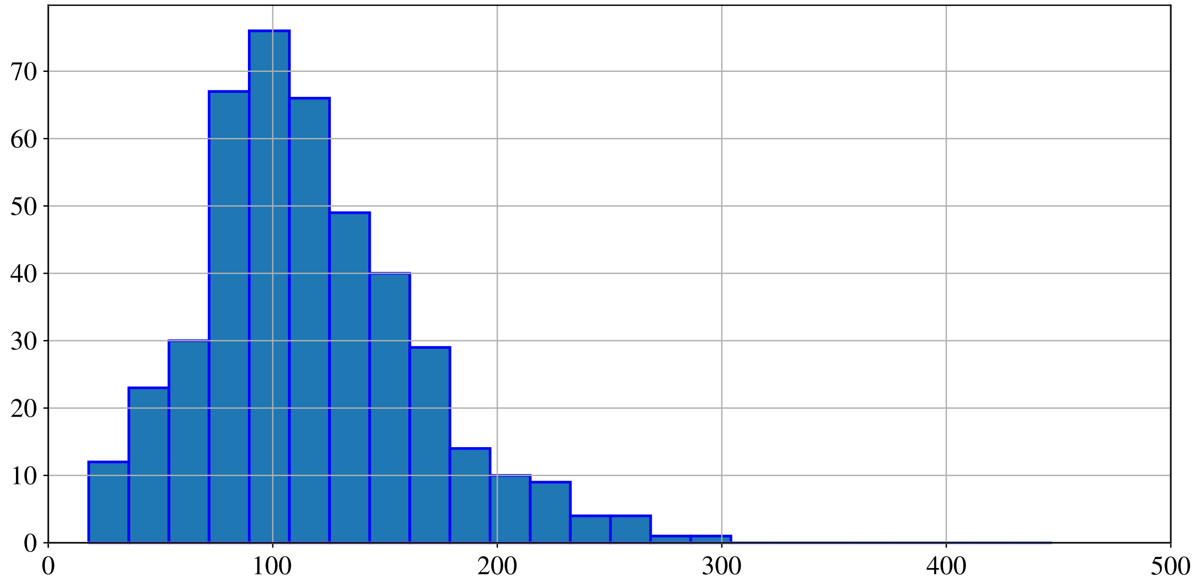

Figure 3. Distribution of number of boxes per pallet

Simulation results format

The simulation results are log files of the following form.

Box movement report format:

```
time, boxID, eventtype, stackID, channelID, palletID
23, 345, 2, 18.12.8, 6, 286
24, 435, 2, 18.11.2, 3, 286
```

Pallet processing report format:

```
time, palletID, palletTypeID, eventtype
23, 231, 4, 4
24, 382, 4, 7
```

where EventID is the event code, Time is the event TimeStamp, BotID is the shuttle code, Command is the type of action performed, StartPoint is the starting section code, EndPoint is the ending section code, BoxType is the box SKU, Channel is the channel code involved in the action, and TrID is the conveyor code involved in the action.

Warehouse model

A warehouse simulation model has been developed.

The solution architecture is shown in Fig. 4.

Robotic warehouse management is performed by the robot management system 1.2 (RMS) and a set of optimization components 1.1 Robots AI.

The class diagram of the core model is shown in Fig. 5.

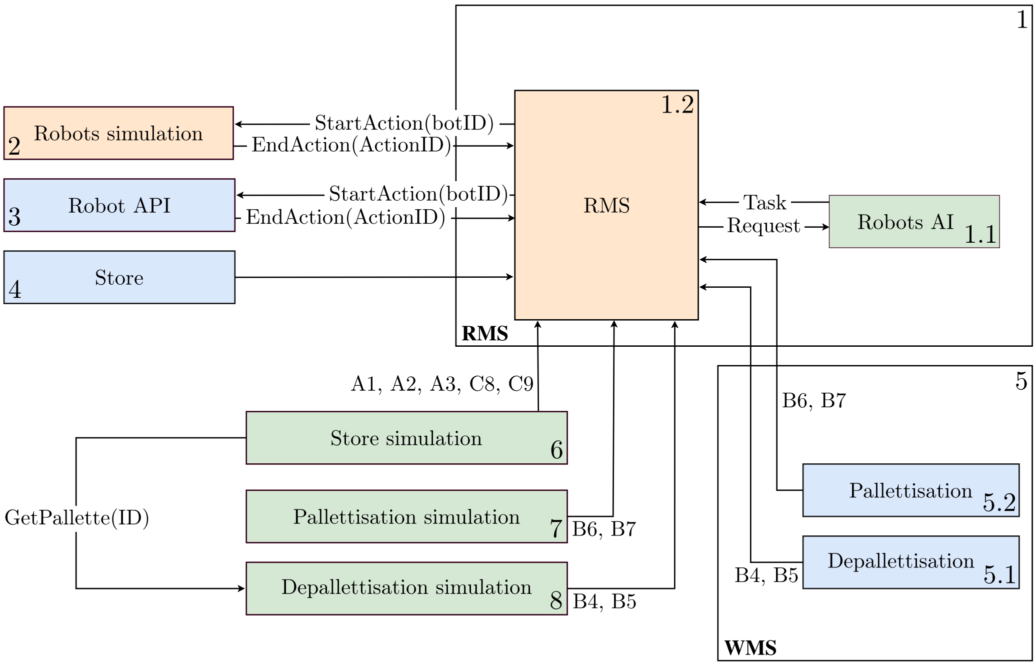

Figure 4. Solution architecture

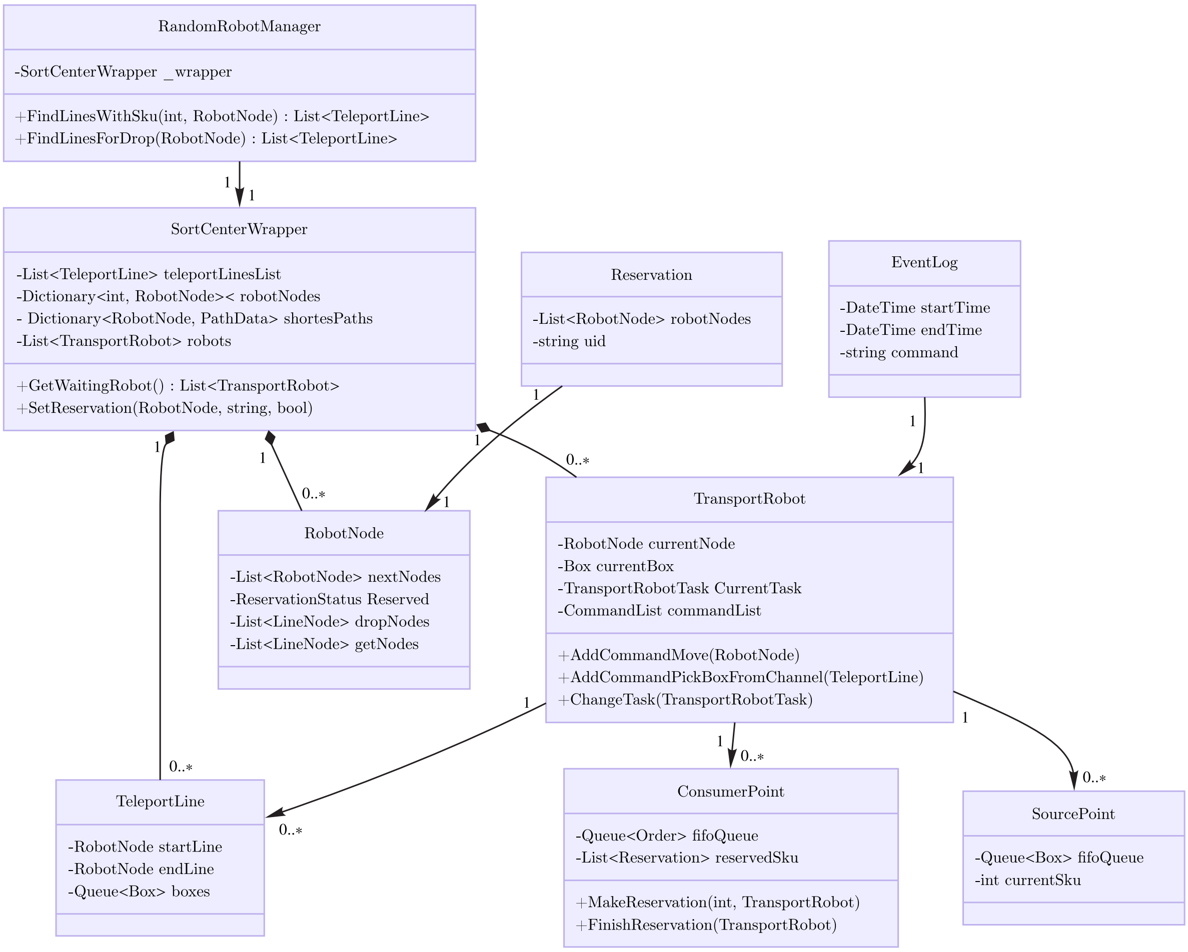

Figure 5. Enter caption

Class diagram of the model

Class Diagram of the Model is shown in Fig. 5.

Here *SortCenterWrapper* represents the main class that manages all components of the warehouse model.

Key properties of the class:

- *teleportLinesList* — a list of lines (FIFO queues) for temporary box storage.
- *robotNodes* — a graph of nodes (intersections, loading/unloading points).
- *shortestPaths* — cached shortest paths between nodes (Dijkstra's algorithm).
- *robots* — a list of all robots in the system.

Class methods:

- *GetWaitingRobot()* — returns robots in standby state.
- *SetReservation()* — reserves nodes to prevent collisions.

The *TransportRobot* class describes the model of a mobile robot that performs box transportation tasks.

Key properties of the class:

- *currentNode* — the current node where the robot is located.
- *currentBox* — the box the robot is carrying.
- *CurrentTask* — the current task (*MoveBoxToLine* or *MoveBoxToPaletize*).
- *commandList* — a queue of commands to execute.

Methods:

- *AddCommandMove()* — adds a movement command to a node.
- *AddCommandPickBoxFromChannel()* — adds a command to pick up a box from a line.
- *ChangeTask()* — changes the robot's task.

The *TeleportLine* class represents a FIFO queue for temporary box storage between operations.

Key properties of the class:

- *startLine* — the starting node of the line (for adding boxes).
- *endLine* — the end node of the line (for picking up boxes).
- *boxes* — a queue of boxes (Queue⟨Box⟩).

The *ConsumerPoint* class describes a palletizing point where robots deliver boxes for pallet assembly.

Key properties of the class:

- *fifoQueue* — a queue of palletizing orders.
- *reservedSku* — a list of reserved SKUs (to prevent conflicts).

Methods:

- *MakeReservation()* — reserves an SKU for a specific robot.
- *FinishReservation()* — completes the reservation.

The *SourcePoint* class describes an unloading (depalletizing) point where robots pick up boxes.
Key properties of the class:

- *fifoQueue* — a queue of boxes for unloading.
- *currentSku* — the current SKU being processed.

The *RobotNode* class describes a graph node representing a robot route point.
Key properties of the class:

- *nextNodes* — a list of adjacent nodes.
- *Reserved* — node reservation flag.
- *dropNodes/getNodes* — nodes for dropping/picking up boxes.

The *RandomRobotManager* class describes a control system that implements a robot control algorithm with random elements.
Class methods:

- *FindLinesWithSku()* — finds a channel with a given SKU.
- *FindLinesForDrop()* — finds a channel for dropping a box.

The *Reservation* class manages information about node reservations for specific robots.
Key properties of the class:

- *robotNodes* — a list of reserved nodes.
- *uid* — the ID of the owner robot.

The *EventLog* class provides system event logging for analytics and debugging.
Key properties of the class:

- *startTime/endTime* — event timestamps.
- *command* — command type (e. g., start_movebox2tr).

A basic algorithm of SKU distribution across channels together with a “greedy” algorithm for assigning shuttles to tasks is implemented in the *RandomRobotManager* class and solves the following problems:

1. Optimization of shuttle movements through the graph.
2. Optimization of shuttle movements when working with FIFO queues.
3. Handling resource access conflicts.
4. Adaptation to changing conditions.

The goal of the developed solution is to minimize the processing time for a set of outgoing pallets.

The mathematical model adopts the following constraints:

- Channel queues are limited (subRowNumber) and operate as FIFO queues.
- Different shuttles cannot occupy the same node simultaneously.
- Shuttles cannot start moving to a node if it is occupied.
- The palletizing order is strictly determined by the FIFO queue.

The algorithm is called after each model simulation step.

The algorithm uses the following main data structures:

- Shuttle status *TransportRobotState*.
- Shuttle task *TransportRobotTask*.
- Shuttle assignment queue for palletizing — FIFO.
- Cached shortest path matrix (shortestPaths), calculated using Dijkstra's algorithm for all node pairs.

The simulation model includes the warehouse model 6, the palletizing system model 7, and the warehouse digital twin 8.

Let us describe the algorithm's operation sequence.

When a free shuttle is available, it is assigned the next task from the task queue. A shuttle is considered free if its current status is "Waiting" (CurrentState = TransportRobotState.Waiting).

If there are depalletizing tasks in the queue, they are assigned to the robot. If there are no depalletizing tasks in the queue — the robot is assigned a task to take a box from a channel with the specified SKU type for palletizing (TakeBoxFromChannel).

To improve the algorithm's performance at this stage, two heuristics are used:

The first prohibits entry to lines A and B for shuttles with a position in the palletizing queue $N_Q > 6$.

The second implements redirection of a robot to a row with a distance from the palletizing point proportional to its position in the palletizing queue if this position $N_Q > 6$. This measure aims to prevent congestion or balance the load.

Simulation results

Using the developed model, some algorithms for product placement in robotic warehouses and robot management were studied.

Nearest channel positioning algorithm (NCPA)

In the simplest case, a box from the depalletizing station is always taken to the nearest channel with spaces for additional boxes.

Most empty channel group placement algorithm (MECGP)

The second algorithm is as follows:

1. When receiving a product for placement, the algorithm looks for the channel with the minimum number of boxes.

2. When receiving a command to select a product, the algorithm looks for a channel where the required product is closest to the exit. If there are several such channels, the nearest one is selected.

Most filled channel group placement algorithm (MFCGP)

An alternative to the above MECGP algorithm, MFCGP suggests placing boxes not in the most empty but in the most filled channels.

Algorithm comparison

Numerical simulation results for NCPA, MECGP, and MFCGP algorithms using 12 robots for our model warehouse configuration are shown in Table 2.

Table 2. Comparison of NCPA, MECGP, and MFCGP algorithms

Algorithms	Service time	Number of transfers
NCPA	9.17:49:22	6739
MECGP	9.17:42:25	6889
MFCGP	11.08:31:15	7293

Optimizing robot quantity

Next, the model was used to test the hypothesis about the existence of an optimal number of warehouse robots (Fig. 6).

Research showed that, for $N_B > 15$, further increasing the number of robots does not improve performance. With further increases, collisions at intersections begin to grow, requiring a more intelligent movement management system.

Study of system dynamics

Figure 7 shows graphs of warehouse fullness over 24 hours of operation for $N_R = 2$, $N_R = 4$, $N_R = 8$, and $N_R = 16$ robots, respectively.

It is clearly visible that, during operation, the warehouse occupancy does not drop below the level of $\frac{N}{N_{\max}} = 0.5$.

It is important that some time is spent on the initial filling of the warehouse. For the selected warehouse model $T_0^1 = 12.2$ hrs, $T_0^2 = 5.5$ hrs, $T_0^3 = 3.2$ hrs, $T_0^4 = 3.0$ hrs, respectively.

An important conclusion is that the chosen upper limit of warehouse occupancy $\frac{N}{N_{\max}} = 0.8$, adopted in most real warehouses, is overkill.

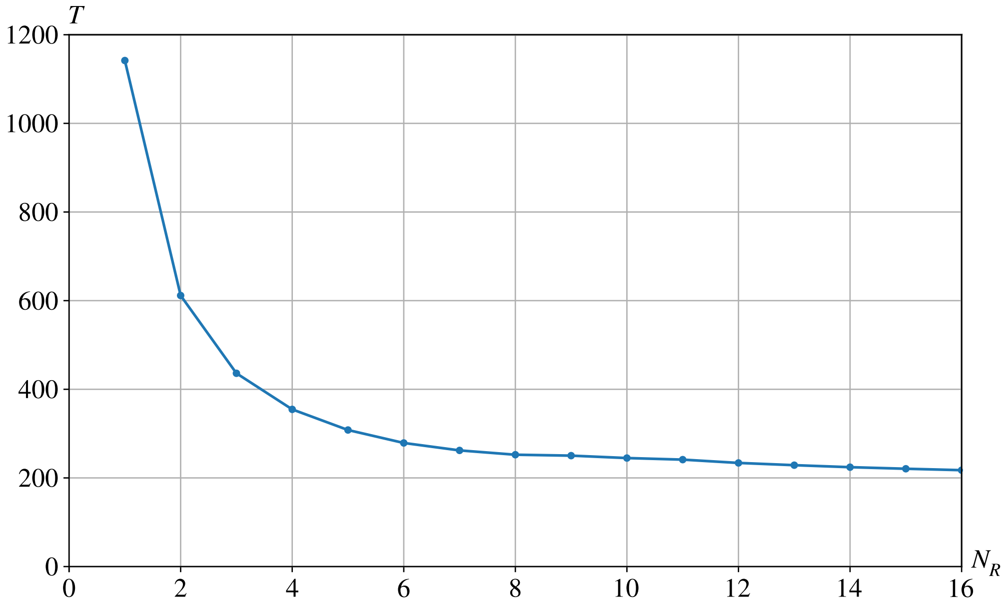

Figure 6. Total service time for different numbers of robots

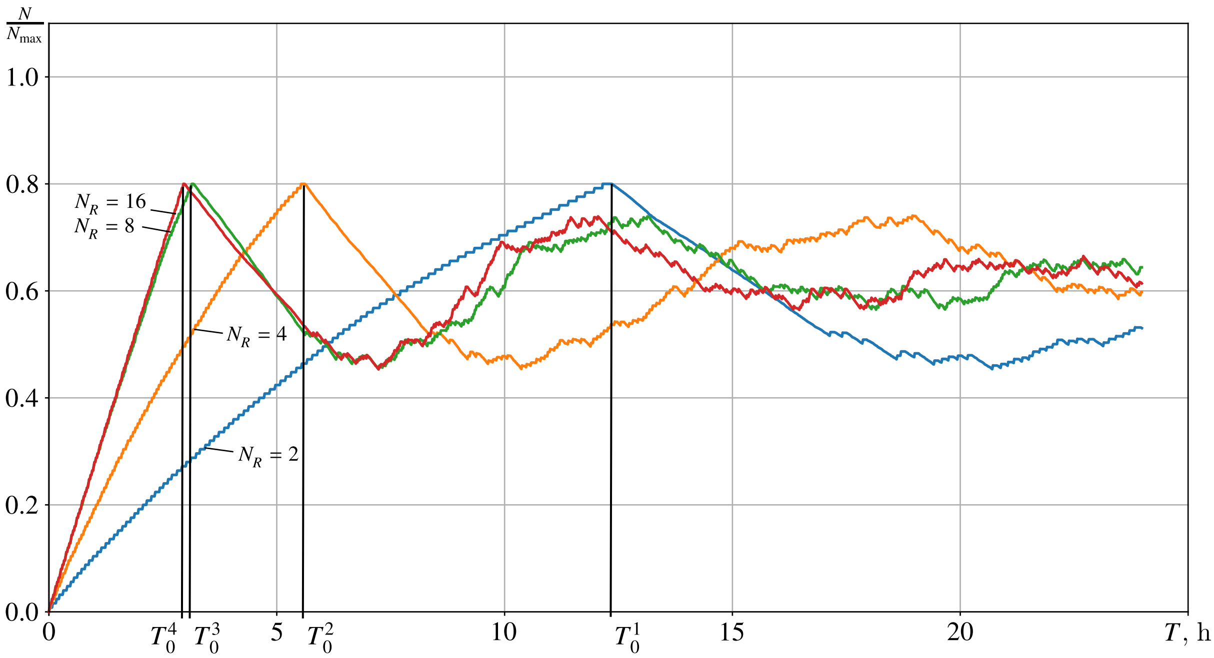

Figure 7. Warehouse fullness as a function of time

Conclusions

A new architecture for a fully robotic multilevel box storage warehouse has been proposed.

The developed and implemented simulation model allows evaluating and optimizing the performance of such warehouses.

The developed model allows:

- calculating the optimal number of robots in the warehouse;
- assessing the feasibility of daily shipping plans;

- comparing different approaches to product placement in fully robotic multilevel box storage warehouses.

The model also enables implementing algorithms for optimal product placement in the warehouse and task assignment to warehouse robots.

Main tasks for further research:

- Building full-scale multilevel models for real warehouses.
- Applying Reinforcement Learning methods for managing product distribution in the warehouse and task distribution among robots.
- Solving the optimization problem for pallet formation sequences and initial warehouse filling considering daily shipping plans.

References

Baker P., Canessa M. Warehouse design: A structured approach // European Journal of Operational Research. — 2009. — Vol. 193. — P. 425–436. — https://doi.org/10.1016/j.ejor.2007.11.045

Chayaphum A., Supsomboon S., Butrat A. The optimal number of reach trucks and order picker trucks in warehouse determining using simulation // Proceedings of the 2019 Research, Invention, and Innovation Congress (RI2C). — 2019. — P. 1–5. — https://doi.org/10.1109/RI2C48728.2019.8999936

van Gemund A., de Weerdt M., Witteveen C. Dealing with uncertainty in operational transport planning // Negenborn R. R., Lukszo Z., Hellendoorn H. (eds.) Intelligent Infrastructures. — Dordrecht: Springer Netherlands, 2010. — P. 349–375.

Gilya-Zetinov A., Demianova D., Khelvas A. Palletizing for full-automated warehouses on the genetics algorithm base // Proceedings of the International Conference on Engineering and Telecommunication (EnT) Proceedings; on Engineering, I.C.; (EnT), T., Eds. — IEEE, 2019. — P. 1–5. — https://doi.org/10.1109/EnT47717.2019.9030576

Goodwin P., Önkal D., Thomson M. Do forecasts expressed as prediction intervals improve production planning decisions? // European Journal of Operational Research. — 2010. — Vol. 3. — P. 195–201.

Macro J. G., Salmi R. E. Warehousing and inventory management: a simulation tool to determine warehouse efficiencies and storage allocations // Proceedings of the Proceedings of the 34th Conference on Winter Simulation: Exploring New Frontiers. Winter Simulation Conference, 2002 (WSC’02). — P. 1274–1281.

Maxemchuk N. Simple robotic routing in Ad Hoc networks // IEEE Xplore. — 2005. — Vol. 3. — P. 1–9.

Oitzman M. Logistics conference recap; Pitney Bowes on Autonomous Sorting // The Robot Report. — 2021. — https://www.therobotreport.com/amrs-pitney-bowes-on-autonomous-sorting/ (accessed: 20.05.2025)

Pankratov K., Kruchkou H., Samakhvalau A., Gadzhimirzayev S., Pashkov R., Razhkou D., Khelvas A. Digital twin of full-automated warehouse with deep storage racks // Proceedings of the 5th Winter IFSA Conference on Automation, Robotics & Communications for Industry 4.0/5.0 (ARCI’2025), 19–21 February 2025, Granada, Spain, 2025. — P. 172–179. — https://doi.org/10.13140/RG.2.2.13085.63208

Rangel T. Ant colony optimization algorithm // 2019. — Vol. 3. — P. 1–4.

Saderova J., Rosova A., Behunova A. Case study: the simulation modelling of selected activity in a warehouse operation // Wireless Netw. — 2022. — Vol. 28. — P. 431–440. — https://doi.org/10.1007/s11276-021-02574-6

Silver D. Cooperative pathfinding // Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. — 2005. — Vol. 1, No. 1. — P. 117–122.

Soloviev V., Nomerchuk A. A model for processing applications and distributing tasks for a robotic warehouse // Cybernetics and Programming. — 2023. — Vol. 3. — P. 144–157.

Wurman P. R., D'Andrea R., Mountz M. Coordinating hundreds of cooperative, autonomous vehicles in warehouses // AI Magazine. — 2008. — Vol. 3. — P. 60–67.