The Development of a Multi-Agent System for Controlling an Autonomous Robot

Natalia Axak [1], Mykola Korablyov [1] and Matvii Ushakov [1]

[1] *Kharkiv National University of Radio Electronics, Nauky Ave. 14, Kharkiv 61166, Ukraine*

Abstract
A control system for autopilot transport robots in warehouses has been developed to solve the problem of transportation, loading and unloading of the goods. The system is implemented on the basis of an organizational model of the mobile robots control process using a multi-agent approach. The proposed model describes the interaction of agents, the functions and the amount of work they perform, as well as the set of admissible states. The implemented algorithm for finding the shortest path in the virtual world of a warehouse to find needed targets. The developed system effectiveness has been demonstrated using computer simulations.

Keywords
multi-agent system, path planning, autonomous robot, work planning

1 Introduction

Currently, large companies (joint stock companies) merged into several companies tend to have central warehouses, where the products arrive for storing for a certain time. Any warehouse processes at least three types of material flows: input, output and internal. The presence of an input flow means the need to unload the transport, check the quantity and quality of the arrived cargo. The output flow necessitates loading on transport or leave for production, internal - the need to move the cargo in the warehouse. In general, the complex of operations contains the following sequence: unloading of transport, acceptance of goods, placement for storage, release of goods from storage locations, intra-warehouse movement of goods.

Any warehouse is faced with the problem of warehousing and cargo handling. Symptoms of errors in managing warehouse processes are: lack of space, ineffective use of equipment and technical resources, high costs of storage and handling of goods, poor quality of customer service. One of the tasks to increase the efficiency of the warehouse is to improve the organization of processes and technology for performing work. The solution of technical problems includes the development of algorithms for the effective placement of goods at storage locations, movement of goods in the warehouse, picking routes.

To solve the problem of internal transportation - movement of goods in the warehouse, a multi-agent control system for autopilot transport robots in warehouse premises is proposed, which will increase the efficiency of using a warehouse space without equipment downtime and queues.

2 Literature review

The warehouse topic as well as mobile robot controls at the warehouse, one of the most promising topics. When controlling several robots, it may be necessary to coordinate the actions of agents and distribute the amount of work between them. One robot can perform certain tasks, while some of them

ICTERI-2021, Vol I: Main Conference, PhD Symposium, Posters and Demonstrations, September 28 – October 2, 2021, Kherson, Ukraine
EMAIL: natalia.axak@nure.ua (N. Axak); mykola.korablyov@nure.ua (M. Korablyov); matvii.ushakov@nure.ua (M. Ushakov)
ORCID: 0000-0001-8372-8432 (N. Axak); 0000-0002-8931-4350 (M. Korablyov); 0000-0003-0230-1555 (M. Ushakov)

may be too complex to perform, for example, spatially separated tasks. Earlier the interaction of several robots was carried out using centralized and distributed methods. Centralized methods strived to provide a turnkey solution; however, such methods were time-consuming and computationally expensive due to the serious complexity of the specific problem [1, 2]. Auction techniques have been used as efficient, decentralized methods for coordinating multi-robots [3]. A common environment for routing multi-robots were proposed, as well as analysis of trading rules for various team purposes. A strategy for assigning a dynamic task for a heterogeneous team of multi-robots in real scenarios were proposed in [4]. This strategy was expected to be effective, scalable to solve problems of increasing complexity with minimal intervention. For this purpose, a basic auction was chosen, which was implemented to solve the problem of control of multi-robots, which needs coordination and organization of a set of subtasks. A simple auction strategy was chosen that provides linear costs associated with increasing the number of robots that make up a team.

The interaction and the distribution of work among the robots is receiving increasing attention. To solve the problems of coordinating mechanisms of the multi-robots in [5], attention is paid to the problem of the distribution of multi-robotic tasks. In [6], the question of the negotiation approach is considered, which is based on the simultaneous division of the task in heterogeneous multi-robotic systems. In addition, the issue of how to expand the bilateral protocol of negotiations to more than two parties is considered. The protocol is being tested in numerical simulations with various scenarios and applied to three real-life missions. In [7], the special attention paid to the problem of effective coordination of the multi-robots. Typically, a multi-robot coordination strategy proposes a centralized approach in which each robot (agent) is responsible for its own planning. The key advantage of centralized approaches is that they can generate global optimal plans. While most distributed approaches can overcome the barriers inherent in the centralized approach, they can only create suboptimal plans because they cannot take full advantage of the information available to all team members.

In [8], the problems of performing dynamic tasks by multi-robots are formalize, using state transitions, represented through the behavior tree. A structure with distributed algorithms is proposed for communication between robots, negotiation protocols and coordination using a priority mechanism. In [9], the hybrid software architecture is proposed which supports both adaptability and reactivity of the robots behavior in dynamic environments. The structure of the multi-agent system for an autonomous robot is developed, in which different roles of agents interact to achieve different functional capabilities.

The task assignment and the path planning are the two main problems faced by the robots with many tasks. The study [10] proposed a methodology for optimizing task allocation and route planning in the case of multiple start and end points, where each robot starts and ends at the same warehouse. Since planning the trajectory of several robots in dynamic environments is a difficult task, in [11], the most appropriate path plan was developed to achieve the goal through overcoming the obstacles encountered by service robots. For this, navigation based on particle swarm optimization proposed.

In [12], the interaction of the robot and the environment in navigation tasks was presented, in which robots have no idea about their habitat and planning functions; instead, the active environment is responsible for these aspects. This was done by creating cells for traffic management in the corresponding area. Cells interact with each other to manage information about the environment and build instructions on how the robots move. In [13], a multi-robot path planning algorithm proposed using Deep Learning combined with the Convolutional Neural Network (CNN) algorithm. Deep Q-Learning used to strengthen the learning algorithm in combination with the CNN algorithm, which is necessary for efficient situation analysis. CNN analyzes the exact situation using information about the image of the environment, and the robot moves based on the situation analyzed using the Deep Q-Learning. In [14], the authors investigated the behavior of the system during joint hunting with several robots. They developed a mathematical model to represent the problem of cooperative hunting involving multiple robots. Dynamic path planning problems for indoor mobile robots are NP-Hard. Route planning algorithms usually suggest the following: the mobile robot will achieve its goals in the shortest time, overcoming the minimum distance with the least energy consumption, as well as finding a free route in an environment in which dynamic and static objects are located, including a larger number of robots. Analysis of the path planning problem for mobile robots showed that genetic algorithms [16, 17, 18] and a popular path-finding algorithms A * and D * [13, 19, 20, 21, 22] are most often used.

Thus, the studies of the architectures of control systems for the autonomous robots and approaches for their implementation have advanced and have shown significant success in modeling and developing a complex software system for robots. However, there are certain limitations and disadvantages, and therefore such methods and algorithms need to be improved and improved. The agent-based approach is advisable to use for the development of new models

The aim of the work is to develop a multi-agent control system for autopilot transport robots to solve the problem of transporting goods in warehouses, which is capable of searching. To achieve this goal, it is necessary to solve to develop an organizational model of the mobile robots control process, its components and their interaction.

3 Organizational model of multi-agent system for control autonomous robot

To control transport robots in warehouses, we will use the agent-group-role (AGR) organizational concept [15]. The advantage of the organizational concept is the ability to model the complex system behavior, which helps to verify the designed system. By a group we mean a team, capable of achieving the goal autonomously and consistently, with minimal control actions. The aspects are: D_Q– finding the minimum route in the environment in which dynamic and static objects are located; D_D– distribution of a work in the coordinated joint activity of agents-robots. We represent the multi-agent system *MAS* in the following form:

$$MAS = < A, V, C >, \tag{1}$$

where $A = \{A_{i,k_m}\}$ ($i = \{exe, init, coord\}, m = 1,2,3$) – the set of agents, k_1 – the number of agents in the subset A_{exe}, k_2 – the number of agents in the subset A_{init}, k_3 – the number of agents in the subset A_{coord}, V – the virtual world of the warehouse, consisting of a two-dimensional lattice of rectangular patches, C – the connections between agents and the environment: the agents exchange messages.

The composition of the *MAS* is fixed and is a set of heterogeneous (by functions, i.e. performing different functions) agents $N = \sum k_m$, known total amount of the works $R^{A_{exe,k_1}} \geq 0$, which needs to be performed. Identified types of the agents $\{A_{i,k_m}\}$, with certain characteristics that reflect the tasks assigned to them.

The states of the agents include their functions F and fixed volumes of work $R^{A_{i,k_m}}$

$$S = < F, R^{A_{i,k_m}} >. \tag{2}$$

Then

$$\sum_{j \in N} R^{A_{i,k_m}} A_{i,k_m} w_j \to \min_{w_j \in \{0;1\}}, \tag{3}$$

where value $w_j = \begin{cases} 0, \text{ if the agent is not working} \\ 1, \text{ if the agent is working} \end{cases}$,

$$\sum_{j \in N} R^{A_{i,k_m}} w_j \geq R^A. \tag{4}$$

This means that the total amount of work does not exceed the capabilities of all agents. To distribute the amount of work between agents, denote the amount of work performed by a particular agent as J_{i,k_m}. Then we have a continuous problem:

$$\sum_{j \in N} A_{i,k_m} J_{i,k_m} \geq R^A \to \min_{J_{i,k_m}}. \tag{5}$$

When condition (4) is satisfied, the agents should be ordered and sequentially loaded to the maximum until the entire amount of work is distributed.

To ensure distributed cooperation between agents, an organizational model is proposed, consisting of the following components (Fig. 1):

- a set of a fixed number of agents that determines the composition of the group;
- a system structure describing the relationship between group members and their interactions;
- the states of the agents include: the functions and the scope of a work performed by the agents, as well as sets of allowed states;
- the awareness of the agents – an information from external and internal parameters.

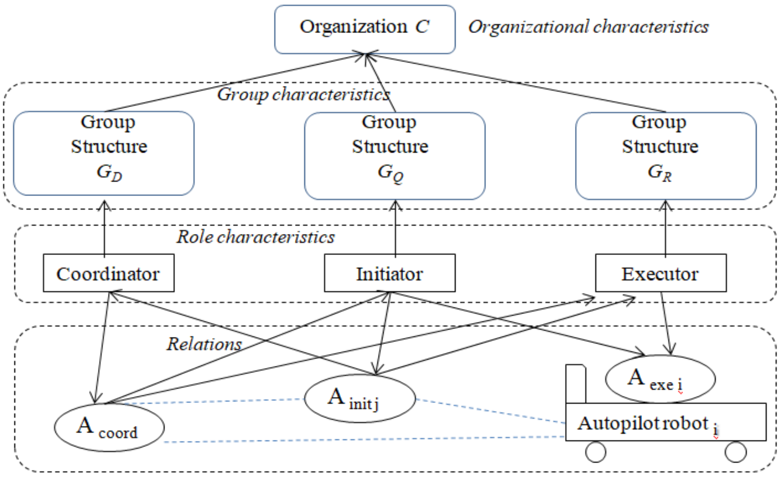

Figure 1: The structure of the organizational model for the process of control an autonomous robots

The organizational model is presented in groups $G = G_D, G_Q, G_R$, where G_D – the distribution; G_Q – the quest; G_R – the resolution. In the G_D group, work is distributed among agents. The coordinator agent A_{coord} distributes the work among the executing agents and orders A_{exe} agents according to the degree of their workload. The G_Q group searches for the minimum route. The initiating agent A_{init} determines the location of the free executing agent A_{exe} and the coordinates of the placement of the target cargo P. In the G_R group, tasks are solved: A_{exe}, agents, having received tasks from A_{init} move along the constructed route to the target cargo P and move it to the destination.

The dynamic characteristic S_D of organization C allows you to control the observance of the D_Q, D_D *aspects to achieve the goals*

$$S_D = \forall\tau : TIME\ state(\iota,\tau,C) \models complacent(conjiuction(D_Q, D_D)),$$

where $state(\iota,\tau,C) \models complacent(conjiuction(D_Q, D_D))$ determines that the condition $state(\iota,\tau,C)$ at the moment τ in the organization C in trace ι there is a characteristic $complacent(conjiuction(D_Q, D_D))$ with predicate |=, denoting the correspondence relationship between state and state characteristic.

Each request for the delivery of the target product P is served after a certain time delay (Fig.2).

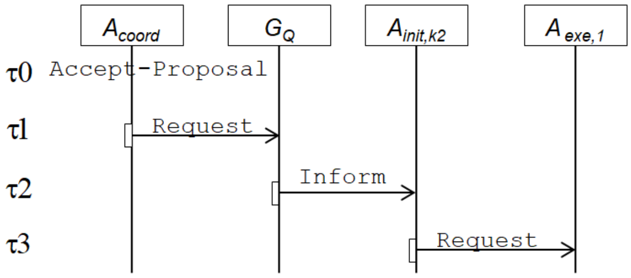

Figure 2: Free Agent Service Request Example $A_{exe,1}$

If at time $\tau 1$ the agent A_{coord} issues a request to move the product P with coordinates (x_P, y_P) to the group G_Q then at time $\tau 2$ after $\tau 1$ the group G_Q issues possible routes to the agents $A_{init,k2}$ at time $\tau 3$ after $\tau 2$ the agent $A_{init,j}$ *(j=1,…,k₂)* ends a message to the free agent $A_{exe,j}$ *(j=1,…,k₁)*, located at the coordinates (x_A, y_A), the route to the product P with coordinates (x_P, y_P).

Temporal formalization is expressed as a dynamic characteristic in TTL (with a free variable to trace ι):

∀ ι,τ, *Aexe* ∃ *Acoord*
[state(ι,τ1,output(*Acoord*)) |=
communication_from_to(request_to move product (xp,yp), *Acoord*, G_Q) ⇒
[∃τ2 > τ1
state (ι,τ2,output(G_Q)) |=
communication_from_to(inform_routes, G_Q, *Ainit*)] ⇒
[∃τ3 > τ2
state (ι,τ3,output(*Ainit*)) |=
communication_from_to(route {(xA,yA)-> (xp,yp)}, *Ainit*, *Aexe,1*)] ⇒

3.1 Second algorithm based on negotiations (SAN)

For the most efficient distribution of the works, the agents interact with each other through negotiations. During the negotiation process, the most suitable performer for each job is identified. The negotiation process in the distribution of a work consists of the following components: the negotiation object and the negotiation protocol – a set of rules according to which the agents interact.

The subject of negotiation implies a range of issues on which an agreement must be reached. The negotiations take place on the stage of distributing D_D work between the agents. In the course of the negotiations, the agent-initiator A_{coord} is looking for a potential the executor agent $A_{exe,k1}$ to perform some work. The object of the agent negotiations at the stage of a work distribution will be the distance between the autopilot robot and the product that needs to be moved. The movement of goods P is characterized by the maximum duration t_P, after which its movement becomes irrelevant. That is, if the executing agent $A_{exe,k1}$ searches for the product P longer than t_P, then his work will be meaningless. Figure 3 shows the function $f(t_P)$, which describes the dependence of the equilibrium value on time.

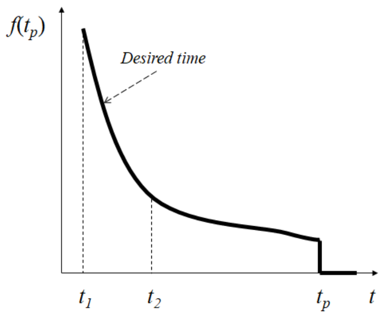

Figure 3: Balancing function

The set of admissible values of Y is the set $(t, f(t_P))$, that satisfy the condition

$$(t, f(t) \in Y, \text{if } \left(\left(f(t) = \min\left(f(t_1), f(t_2), \ldots f(t_n)\right) \wedge (f(t) \le f(t_p)\right)\right).$$

The negotiation protocol is shown in Figure 4. The A_{coord} agent sends messages to the executing agents $A_{exe,k1}$. The message contains the coordinates of the target location and the final point of delivery. In the responses from the executing agents there is a load indicator, (for example, *IsValid=false* – the agent is 100% loaded) and its coordinates (x_A, y_A). Based on these responses, the A_{init} agent builds D_Q routes and sends them to the A_{coord} agent. Agent A_{coord} distributes work D_D: a message is sent to agent $A_{exe,j}$ $(j=1,\ldots,k_1)$ with the maximum value of the function $f(t_P)$ and the load indicator и $IsValid=true$. The rest of the agents receive a message that the executor agent for work has already been found.

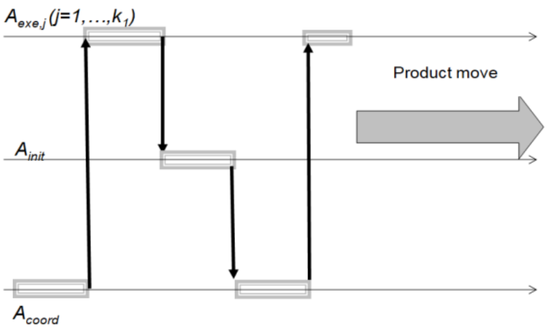

Figure 4: Negotiation protocol

The main purpose of agent $A_{init,k2}$ is to find the best path for agent $A_{exe,k1}$ from its location (x_A, y_A) to product P with coordinates (x_P, y_P). Figure 5 shows a map with open cells on which the autopilot robot can move; closed cells, inaccessible for movement; starting point; end point of movement of the autopilot robot.

For each node, the following values are determined: G – distance traveled, H – remaining distance, $F=G+H$ – total distance. After defining an F – value for each node, the nodes are sorted in descending order of that value. Obviously, a node with a value of $F=3$ is more suitable.

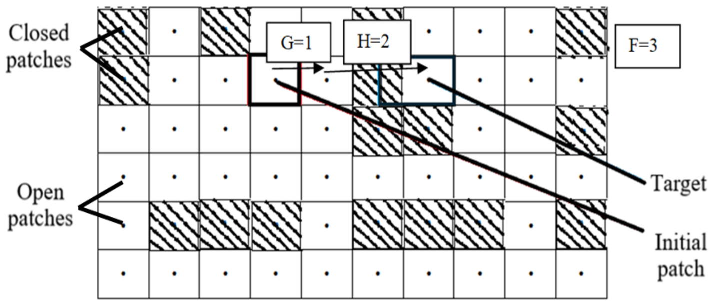

Figure 5: The virtual world of the warehouse

When the next node is reached, it is necessary to recalculate the value of the current node and determine a new direction (Figure 6.a). The H value is not recalculated. But situations may arise when the nodes have the same characteristic values, such situations are called dead ends. If such a situation is detected, then control goes to the next level.

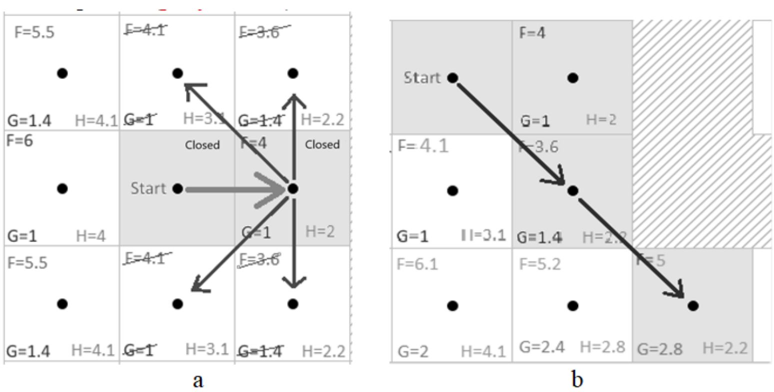

Figure 6: a - Selecting a new cell (node); b - Move to the next node

The path is built from the parent nodes contained in each node. If there is no parent, then such node is considered the initial one. Each time the current cell is changed, the indicators for neighboring nodes are recalculated. After each step, the cycle repeats (Figure 6.b).

When moving from one node to another, the previous data is saved, which can be used to build a return path (Figure 7). When the current position of the node coincides with the final one, the iteration over the nodes ends and the reverse path is built. Also, to find the optimal path, the wave tracing algorithm was used. Its idea is as follows: there are 2 points - start and finish.

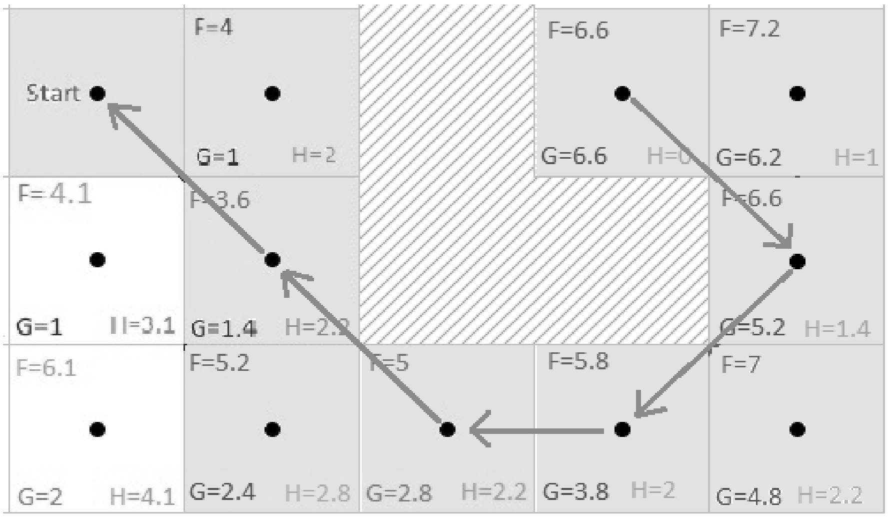

Figure 7: Built path

The object can move in 4 directions: up, down, left, right. When the algorithm starts, the step value equal to 1 is entered in the neighboring cells. At the next increment of the cycle, the step value is increased by another 1. After filling the cells of the first step, the neighboring cells of the first step will be filled, they will already contain the value 2. It is important that the cell hasn’t already been passed before or should not be a wall (impassable). The cycles are repeated until the finish line is reached. The exit from the main loop is the value of the start cell that is not equal to -1 or when there is no solution (the number of steps bigger than the number of cells).

Figure 8 shows the 4 stages of the wave tracing algorithm.

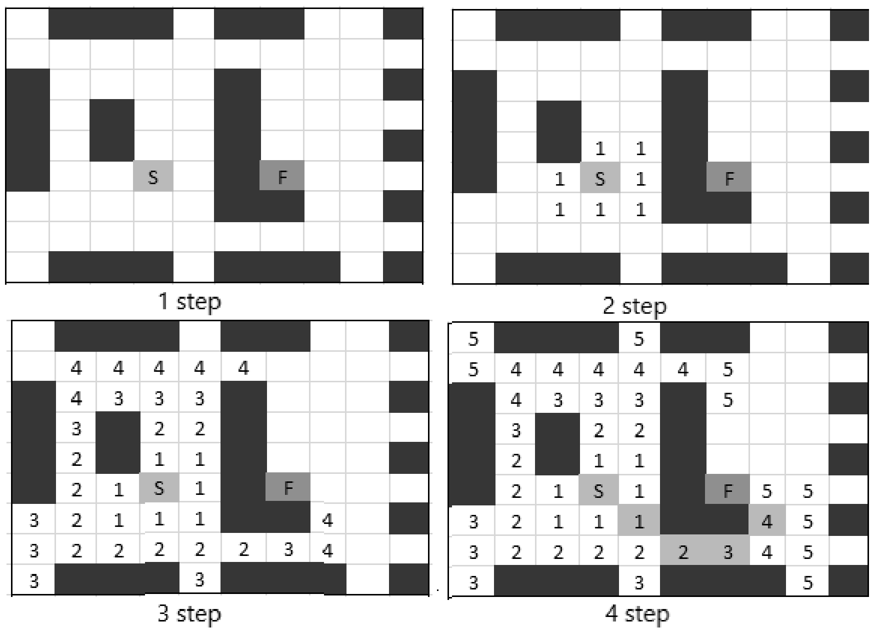

Figure 8: Demonstration of the wave algorithm

Point S indicates the location of the object (starting point), F - the position where the target is, the numbers are indicating number of steps.

4 The simulation of a multi-agent system for control an autonomous robot

The experiments were conducted on Intel Core i3-7100, 3.9 GHz, OS Microsoft Windows 12, Unity 3D and C#. When the application starts, the scene is loaded, on which the map is initialized (Fig. 9), the agents of the autopilot warehouse robots, and the station are created.

Figure 9: Simulation for scene with 2 warehouse robots

The location of the robots is defined in random way. The agent, that was created on station, corresponds to the agent-initiator, which distributes the work to other agents. The agents-executors are created for each warehouse robot communication with initiator and receive the assignment from it. On Fig. 10 it is shown that one of warehouse robots is doing its job and the others are in some standby mode.

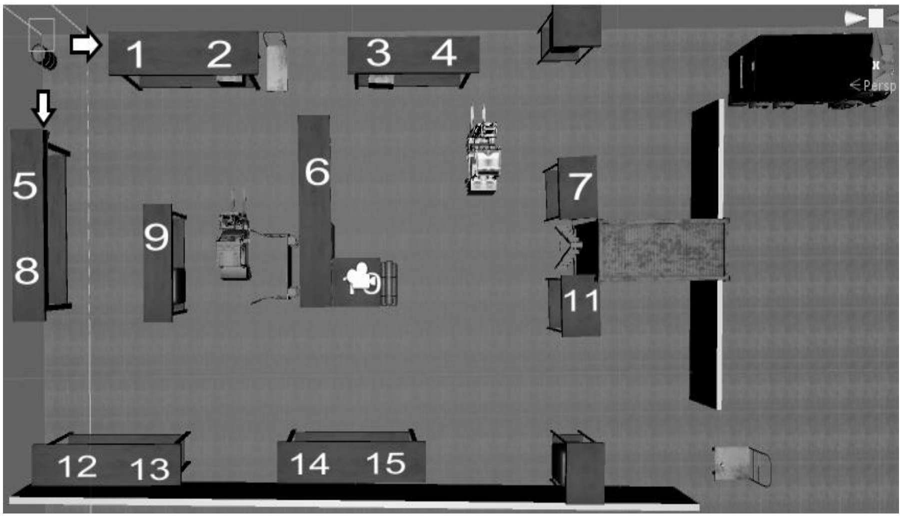

Figure 10: Warehouse scene by showing the coordinates of the placement of goods

After the assignments are received, the robot is transferred to the "Busy" state and the task is added to the queue of the tasks being executed. After that, the shortest path to the target is determined, and after loading the product, the calculation of the shortest path for shifting the cargo is performed. At the same time, the $A_{init,k2}$ agents communicate with each other in order to quickly resolve possible conflict situations (for example, when 2 robots are nearby). When the robot has completed its task, it is removed from the job queue. Then the robot goes into the "Waiting" state or starts performing tasks that are in the queue. In the course of the experiments, the results of modeling the control system for an autopilot transport robot in warehouses are compared, both with the use of the mechanism of agent negotiations for the wave and the A * algorithm, and without it. The tests were run by five robots generated at random starting positions. The placement of goods and the map, which was used to build the shortest path for each robot, was generated 1000 times. The test was repeated 1000 times. Table 1 shows the number of steps from the robot to the load, the search range, the time to build the route and the number of deadlocks that have arisen (the robots are at the same distance) both with the use of the agent negotiation mechanism for the wave and the A * algorithm, and without it.

Table 1
Simulation results.

Algorithm	Without using the mechanism of agency negotiations				Using the mechanism of agency negotiations			
	Steps to product	Search range	Time	Conflict situation	Steps to product	Search range	Time	Conflict situation
Algorithm A *	67	120	0,58	19	67	96	0,39	-
Wave algorithm	83	138	0,75	21	83	101	0,52	-

As is shown in Table 1 solving the problem of transportation along given coordinates using the mechanism of agency negotiations is carried out faster with a smaller search range and without the occurrence of deadlocks. Thus, the proposed multi-agent system shows the possibilities for creating agents with behavior models capable of changing existing connections and establishing new connections depending on the changes that have occurred in the scene. This opportunity is necessary when responding quickly to changes in the outside world. The initiating agent and the executing agents negotiate using messages. A script was developed to demonstrate the process of a successful task assignment.

5 Conclusions

Autonomous solutions and distributed collaboration between agents provide high flexibility. In this case, feedback and coordination from the central coordinator is used to achieve high efficiency. Studies show that additional strategies can be developed to prevent deadlocks by improving agent decision-making and coordinator behavior.

On the basis of the proposed model, a multi-agent system of an auto-pilot warehouse robot has been developed, which distributes work between agents, and also searches for the shortest path of the autopilot robot from the current point to the target.

The information received by each individual robot is planned to be analyzed to obtain a more efficient solution in the future.

6 References

[1] P. Svestka, M.H Overmars, Coordinated path planning for multiple robots. Robotics and Autonomous Systems. 23 (1998).

[2] Z. Yan, N. Jouandeau, A. Cherif, A survey and analysis of multi-robot coordination. Journal of Advanced Robotic Systems. (2013).

[3] M. Lagoudakis, Auction-Based Multi-Robot Routing. Conference Paper (2005) 7–18.

[4] P. Garcia, Scalable task assignment for heterogeneous multi-robot teams. International journal of advanced robotic systems (2012) 1–10.

[5] B. Gerkey, A formal analysis and taxonomy of task allocation in multi-robot systems. The international journal of robotics Research (2004) 25–39.
[6] C. Rossi, Simultaneous task subdivision and allocation using negotiations in multi-robot system. International journal of advanced robotic system (2015) 15–19.
[7] B. Dias, TraderBots: a new paradigm for robust and efficient multirobot coordination in dynamic environment (2004) 15–25.
[8] Q. Yang et al., Self-reactive planning of multi-robots with dynamic task assignments. International Symposium on Multi-Robot and Multi-Agent Systems (2019) 89-91.
[9] S. Yang et al., Towards a hybrid software architecture and multi-agent approach for autonomous robot software. vol. 14. № 4 (2017) 1729881417716088.
[10] B. R. K. Mantha, B. García de Soto, Task allocation and route planning for robotic service networks with multiple depots in indoor environments. Computing in Civil Engineering: Data, Sensing, and Analytics (2019) 233-240.
[11] N. Baygin, M. Baygin, M. Karakose, PSO Based Path Planning Approach for Multi Service Robots in Dynamic Environments. International Conference on Artificial Intelligence and Data Processing (2018) 1-5.
[12] S. Kameyama, et al., Active Modular Environment for Robot Navigation. arXiv preprint arXiv:2102.12748, 2021.
[13] H. Bae, et al., Multi-robot path planning method using reinforcement learning. Applied Sciences. vol. 9. №. 15 (2019) 3057.
[14] Y. Li Song, C. X. Ma, Mathematical Modeling and Analysis of Multirobot Cooperative Hunting Behaviors. Hindawi Publishing Corporation. Journal of Robotics (2015).
[15] N.G. Aksak, Organizational view of the medical diagnostic multi-agent system Systems and means of artificial intelligence. No. 1 (2014) 8-10. (in Russian).
[16] X. Bajrami, A. Dërmaku, N. Demaku, Artificial Neural Fuzzy Logic Algorithm for Robot Path Finding, IFAC-PapersOnLine, vol. 48, No. 24 (2015) 123-127.
[17] S. Abinaya, et al., Hybrid genetic algorithm approach for mobile robot path planning. Advances in Natural and Applied Sciences, vol. 8, No. 17 (2014) 41-48.
[18] A. Lakhdari, N. Achour, Probabilistic Roadmaps and Hierarchical Genetic Algorithms for Optimal Motion Planning in Intelligent Systems in Science and Information, Springer International Publishing (2014) 321-334.
[19] A.Keselman, S. Ten, A. Ghazali, M. Jubeh, Reinforcement Learning with A* and a Deep Heuristic. arXiv 2018, arXiv:1811.07745.
[20] A. Botea, M. Müller, J. Schaeffer, Near optimal hierarchical path-finding. Game Dev. vol. 1. (2004) 1-30.
[21] M. Basem, et al., Modified A* Algorithm for Safer Mobile Robot Navigation. Proceedings of International Conference on Modeling, Identification & Control, 2013.
[22] F. Duchoň, et al., Path planning with modified A star algorithm for a mobile robot, Procedia Engineering, vol. 96 (2014) 59-69.