UNIWERSYTET TECHNOLOGICZNO-PRZYRODNICZY
IM. JANA I JĘDRZEJA ŚNIADECKICH W BYDGOSZCZY
ZESZYTY NAUKOWE NR 263
TELEKOMUNIKACJA I ELEKTRONIKA 18 (2014) 33-45

TASK SCHEDULING ALGORITHMS FOR MULTI ROBOT ENVIRONMENT

Sandra Śmigiel[1], Fatih Zungor[2], Hunkar Purtul[3], Inanc Inaloz[4]

[1] University of Technology and Life Sciences,
Faculty of Telecommunications, Computer Science and Electrical Engineering,
al. prof. S. Kaliskiego 7, 85-789 Bydgoszcz, Poland
[a] sandra.smigiel@utp.edu.pl

[2] Sakarya University

[3] Karadeniz Technical University

[4] Mustafa Kemal University

Summary: In recent years, the problem of multi robot task allocation and scheduling is the subject of many research activities. The key of this problem is to allocate proper number of tasks for each robot and schedule the optimal task sequence for each robot. In order to minimize the processing time for robots, were designed algorithms of the sample task scheduling. Determination of the optimal path for robots, taking into account the process of task scheduling, may be an effective strategy to manage and control of tasks in a environment. This paper presents the management and control works of robots in the logistics environment (warehouse), which will be realized by the soft real time systems. The aim of the authors was to develop and compare the optimal algorithms management works systems of multi-robot in warehouse, in terms of number of customer served and consumption energy. To validate the effectiveness of the proposed approach and simulation have been made. The results show that the proposed approach can be an effective approach in the design of optimized multi robots task allocation and scheduling scheme, especially for the Queued Task Algorithm, with 1-element queueCapacity and Nearest Task Algorithm.

Keywords: task scheduling, robot control algorithm, multi-agent system, warehouse.

1. INTRODUCTION

The problems of task scheduling and allocation of resources is accompanied almost every creative of human activity. They are a significant element of the decision-making processes of many issues [1]. These include systems: control and planning of production, supply management and marketing, organization of transport, management of resources, etc. The problem of task scheduling and resource allocation, is one of the basic issues attracted many researchers affection, relating to the field of distributed robot systems. Distributed robot system combines the intelligent planning and control, which consider event-based motion planning and control [3, 5].

Real time systems are those in which the logical correctness or correctly performed calculations are equally important as to keep up with the issues taking place in the surrounding real world. The real-time systems are characterized by the concurrency and conditions of timing. The existence of concurrency is related to the necessity of readiness to handle a multiple, concurrent appearing in any of the time points issues [4]. There are three types of the real time systems: Hard Real Time system, Soft Real Time System and Firm Real Time System. In hard real-time systems, the importance of meeting a task execution deadline is a central issue for correctness and reliability of such systems. In hard real-time systems, the importance of meeting a task execution deadline is a central issue for correctness and reliability of such systems. In this systems missing of task is absolute deadline leads to the catastrophic situation. Whereas in soft real time systems is sporadic acceptable to timeout of issues, so the deadline is tolerable and the system continues to operate. In addition to the hard and soft real time system, is acceptable the existence of firm real-time systems, where the individual requirements of real-time, can have a hard and soft character. In the case of real-time systems desired effect is their continuous operation, since their launch to the removal phase [4].

In recent years, many papers focus on issues related to the control of robots and managing, which affects of many areas. One of these ways is the automation of storage processes [3], in which tested the environment, where the robots transported the tasks in the virtual storage. In this article, has been prepared the task scheduling and allocation of resources in management planning of warehouse, which was based on soft real time systems. Considering the problem of scheduling in soft real time systems, use planning via robots, is must to use two fundamental concepts: task and resources. The task consists of performance the sequence of operations, in which each requires the involvement of specific resources. Analysis of the soft real time systems must to remember that, despite their characteristics, must be limited of delinquency. Order a given task cannot wait for performed. Characteristics of soft real time systems can make to creation of artificial reality, which will be presented in this article [2, 6].

The main purpose of the authors was to develop and compare the path planning of robots and the task scheduling algorithms for robots used for transport in virtual magazines.

2. ENVIRONMENT DESCRIPTION

For the purpose of testing algorithms for control and management work of robots, was created the virtual environment. It is simulate a virtual storage environment. In this article, the working environment is divided into fields. Each fields can be occupied by: a shelf, a distribution point, and the obstacle for robot or customer. The controls and management works of robot requires division created the environment for two areas: robot area and client area (fig. 1). Robot areas is a place where the robots can move and carry out the outsourced tasks. Client area is a place intended for clients. The common element for these areas are the distribution points.

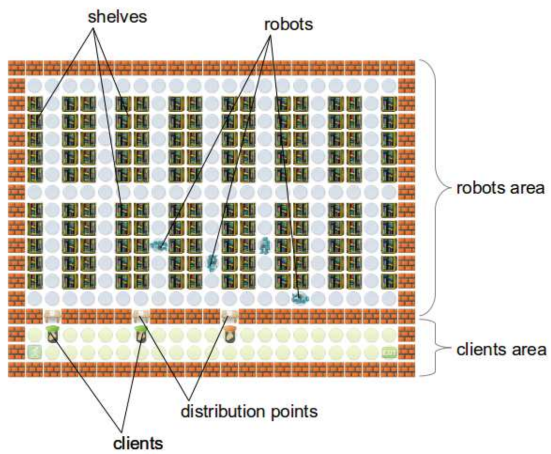

Fig. 1. Environment

The tasks in a virtual storage environment, for control and management of work robots, involve multiple segments of actions. All segments are logically connected or dependent upon each other logically and temporally. Designing such systems involves two issues: determining the sequence of actions and planning the actions them-self. Both the forward issues: task scheduling and action planning, requires a combination and co-operation two areas: robots and clients. Therefore, the problem of designing a robotic system requires the creation of a some level problem.

In created the environment the customers comes to warehouse, to download and/ or put one or more resources. The customer, when enter to warehouse, shall be referred to the distribution point, where delay the resources, if the customer has another task to delay, and later ordered the resources, which would like to download. If the customer will be served, then he could leave the storage. The basis for proper operation warehouse is the suitable number of customers, located in the warehouse in relation to the number of available distribution points. Each of resources has particular location in warehouse. The task of the robots is to transport the resources between the distribution point and the shelves. One robot at one time can transport a maximum of one resources

The submitted environment is divided into steps. At each step, both the robot and the client can: move forward one square, to make a 90 degree turn, to get/ put a resource to/ from a distribution point or to/ from shelf, in the case robots or wait. In addition, the customer, on single field, at one time, can place an order for the necessary resources, and also input/ output the warehouse (provided that stands on the appropriate field). In the case of robots, any activity via waiting, at one time during one step consumes a unit of energy.

The aim of authors was to present some of algorithms, for control and management of robots, in virtual warehouse. Each of algorithms was assigned to another robots. The operation of algorithms for each robots was divided into smaller groups: Main Algorithm, Routing Algorithm and Task Choosing Algorithm. The principle of the algorithms is follows: performed the single-step by the robot, cause that the simulator queries algorithm for a decision. The algorithm returns the decision and when it is possible, performs the action. In addition, for all instances of algorithms, there is one central Agent Coordinator. Agent Coordinator has full knowledge about the environment. Function of Agent Coordinator is mainly control and manage of tasks. The task is called a transport the specified resource from point A to B, that is, from the distribution point on the shelf or vice versa. Each of order or a delivery of a specific resource by a client generates a task, which then placed on the list of tasks to be performed. Placed on the list of tasks can be reserved by the robot control algorithms. There is no limit to the amount of task reserved for one algorithm, at one time. Reservations causes to moves the task from the list of tasks to perform to the list of reserved task. Detection of the Agent Coordinator performed tasks (when the resource will be delivered to the destination), what causes that the tasks are moved from the list of tasks reserved to the list of tasks performed.

In this article, conducted tests were performed for 2 sizes environment (Fig. 2): medium (2 rows of 7 columns of shelves) and large (3 rows of 10 columns of shelves). Each of test took 10,000 steps. In addition, for each study was set the maximum time customer service.

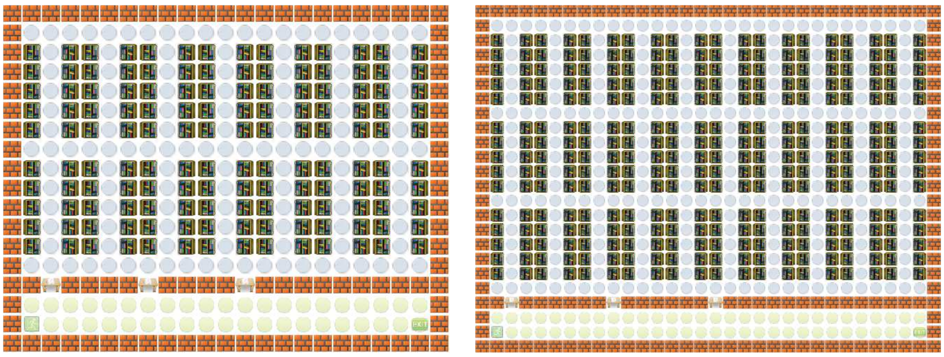

Fig. 2. a) medium size environment, b) large size environment

2.1. MAIN ALGORITHM

Main Algorithm (Fig. 3) is responsible for managing the operation of the robot. In every step of the simulation the Main Algorithm is asked about the decision. It has an instance Routing Algorithm and Task Choosing Algorithm.

Figure 3 shows the principle operation of the Main Algorithm: first algorithm checks whether it has assigned task. If the robot does not have an assigned tasks, then try to get it from the Task Choosing Algorithm. If this fails, then returns a decision taken by the Routing Algorithm. If the robot transports the resource, the Main Algorithm checks, whether the robot is on a good position. If this is confirmed, then appoint the delivery place and queries the Routing Algorithm about a decision. If the robot does not transport the resources, it sets a target loading place and queries the Routing Algorithm about a decision.

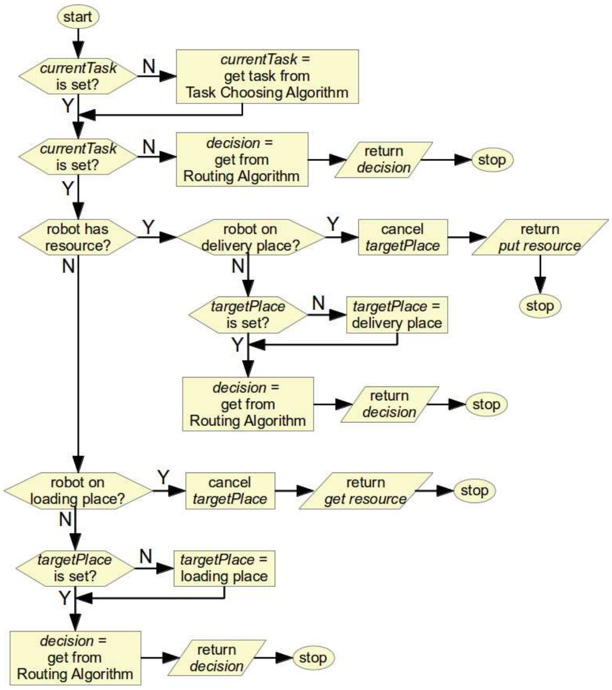

Fig. 3. Main Algorithm

2.2. ROUTING ALGORITHM

Routing Algorithm (Fig. 4) works as follows: if the target is set, then algorithm determines the shortest path to destination, have regard to the obstacles (shelves, walls, other robots). Avoiding obstacles is a result of the use of Algorithm of Dijkstry. If the robot does not have an assigned target, it returns a decision wait. In addition, the Routing Algorithm has a mechanism, called deadlock preventer. If the robot does not move for 20 simulation of steps, the Routing Algorithm returns a decision random to move.

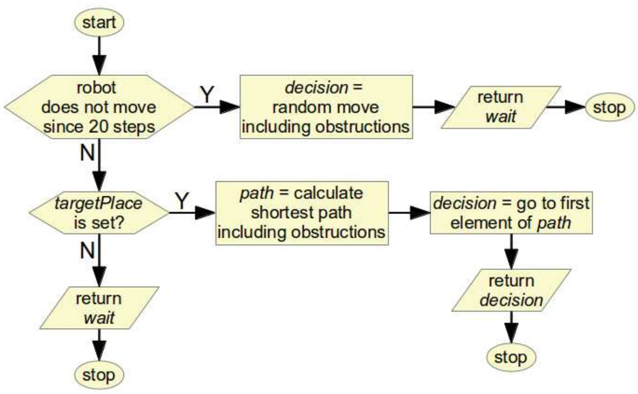

Fig. 4. Routing Algorithm

2.3. TASK CHOOSING ALGORITHM

Task Choosing Algorithm is responsible for the selection of tasks for the robot. In this article presents 3 algorithms of this type: Simple Algorithm Task, Task Nearest Algorithm [3], and also new solution, the most complex Queued Task Algorithm.

2.3.1. SIMPLE TASK ALGORITHM

Simple Task Algorithm (Fig. 5) reserves the first task, which is on the list of tasks to be performed, designated by the Agent Coordinator. This means, that the tasks are performed by robots, in the order in which they were originally created. There is no risk, that any customer will be waiting too long (as compared to the others).

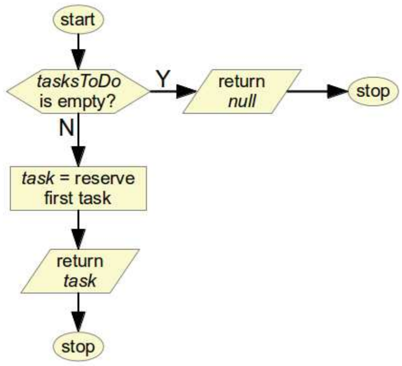

Fig. 5. Simple Task Algorithm

2.3.2. NEAREST TASK ALGORITHM (NTA)

Nearest Task Algorithm for proper operation uses the findNearest function (Fig. 6). Function of findNearest uses in the calculation parameter position, to determine the location of the robot. Then, search the tasks, which the distance between the start point and the position is shortest.

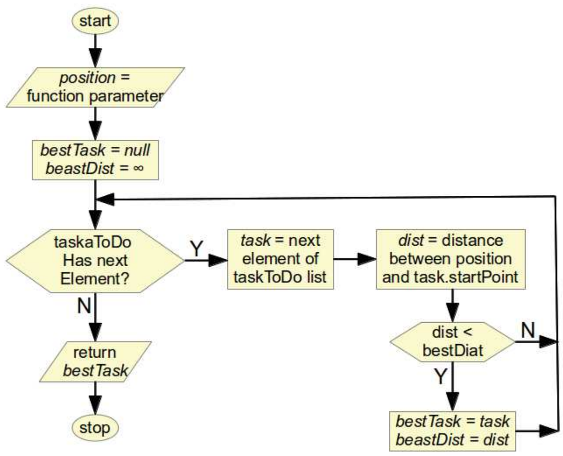

Fig. 6. FindNearest function diagram

Nearest Task Algorithm works as follows (Fig. 7): if the target is set, then algorithm use the function of findNearest, to determine the tasks, in which the start point is close to the current position of the robot.

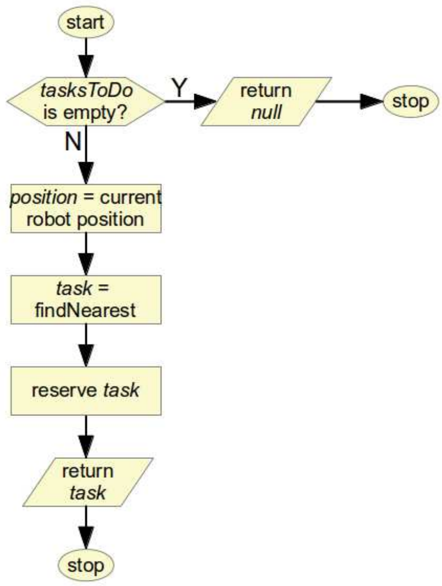

Fig. 7. Nearest Task Algorithm

2.4. QUEUED TASK ALGORITHM (QTAx)

Queued Task Algorithm (Fig. 8) is extended version of Nearest Task Algorithm. Proper operation of this algorithm requires the use of findNearest function. Queued Task Algorithm uses to works the taskQueu, size queueCapacity. This queue stores a few reserved tasks. Queued Task Algorithm works as follows: if the queue is empty, the Queued Task Algorithm reserves task, which the start point is close to the current position of the robot. Then, the algorithm reserves the next task, where distance between of the next point and to the end of the preceding point, was the shortest. In this article, was tested some of variants algorithm with the queueCapacity, equal to 2, 3, 4 (QTA2, QTA3, QTA4).

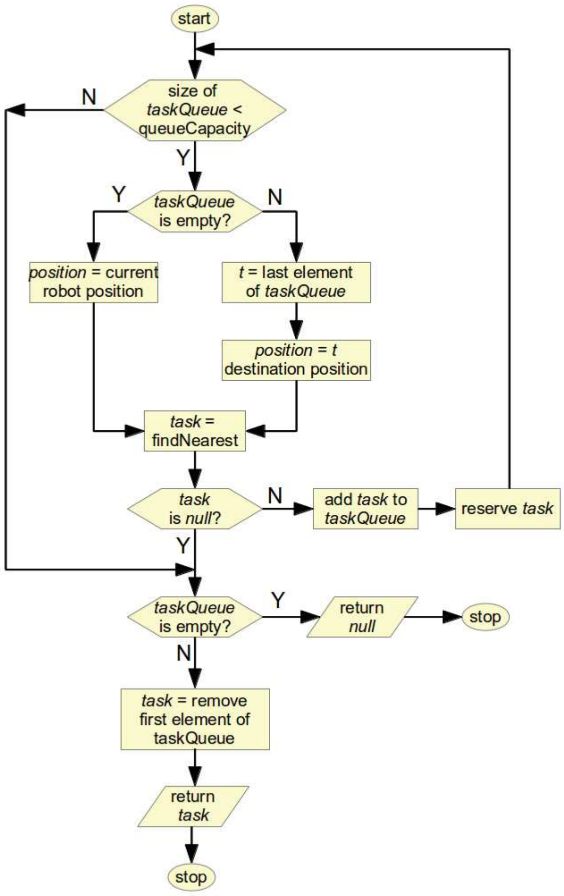

Fig. 8. Queued Task Algorithm

3. RESULTS

Figure 9 shows the results of examination of the proposed algorithms in terms of number of serviced clients as a function of the number of robots. The graphs show that, the results for a small amount of distribution point are similar. The best results were obtained for Nearest Task Algorithm and Queued Task Algorithm, with 2-element queueCapacity. In addition, the worst results were obtained for the Simple Task Algorithm, for a large number of distribution points.

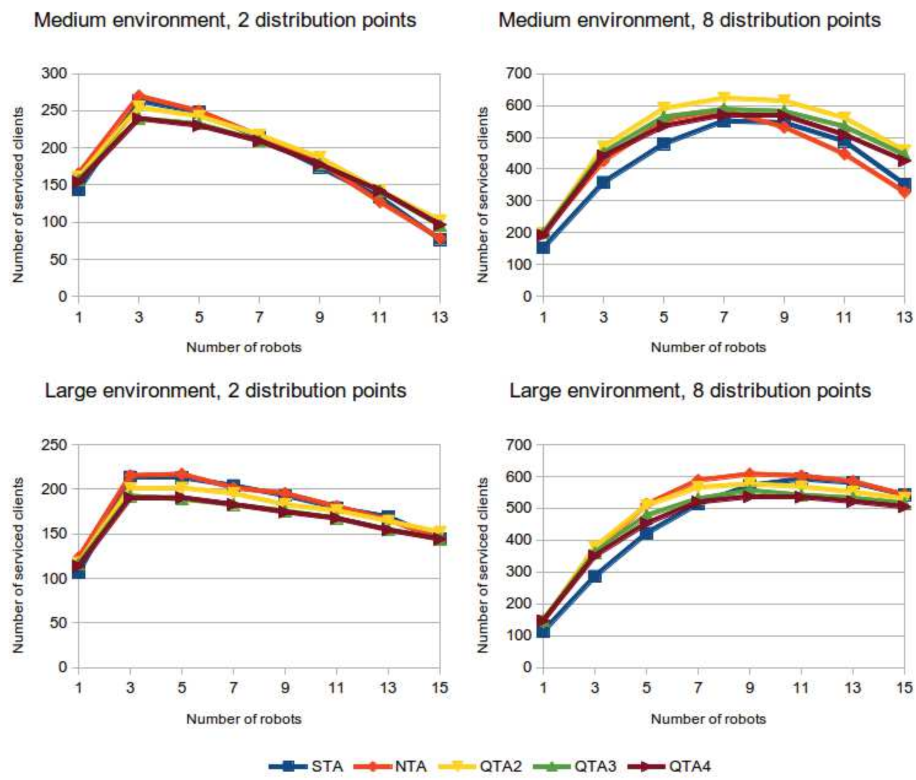

Fig. 9. The results of examination of the proposed algorithms in terms of number of serviced clients as a function of the number of robots

Figure 10 shows the results of examination of the proposed algorithms in terms of average client service time as a function of the number of robots. The results for medium environment shows a similar relationship for the tested algorithms. Whereas differences exist in the case of algorithms in large environment. In terms of a small number of distribution points the best works were obtained for Nearest Task Algorithm. In addition, for a large points of distribution with 2-element queueCapacity the best works were obtained for Queued Task Algorithm.

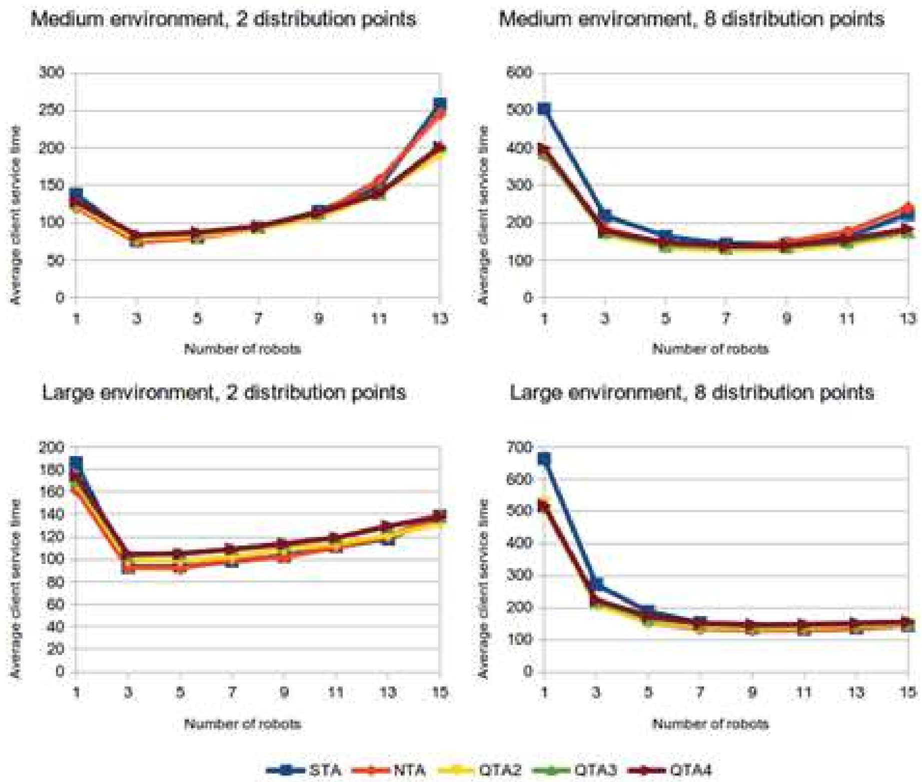

Fig. 10. The results of examination of the proposed algorithms in terms of average client service time as a function of the number of robots

Figure 11 shows the results of examination of the proposed algorithm in terms of average robot energy consumption as a function of the number of robots. Regardless of the size of the environment, in most cases for created algorithms were obtained the similar results. The exception for this analysis is large environment, where for a large number of distribution point definitely the best proved to be the Task Queued Algorithm, with 4-element queueCapacity.

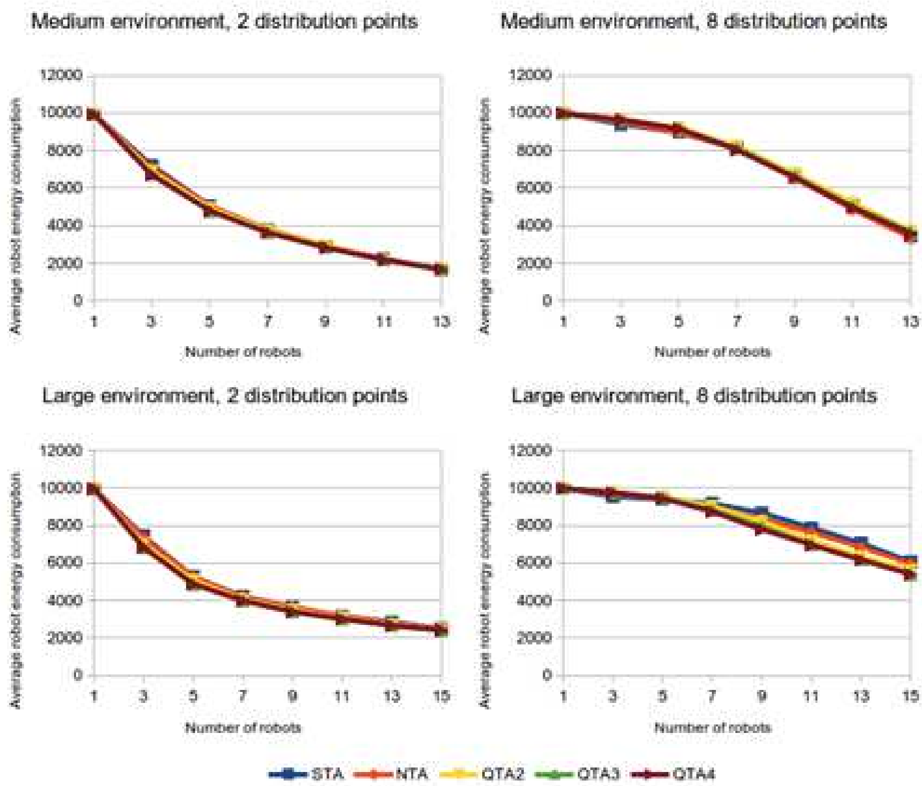

Fig. 11. The results of examination of the proposed algorithm in terms of average robot energy consumption as a function of the number of robots

Figure 12 shows the results of examination of the proposed algorithms in terms of robot energy consumption per serviced client as a function of the number of robots. The tests confirmed no dependence on the choice of tasks, in relation to the consumed energy, needed for customer service. The exception for this analysis is Simple Task Algorithm, which for a small amount of robots were obtained the worst results. The opposite relationship received the Queued Task Algorithm with 2-element queueCapacity, which works the best for a large number of robots.

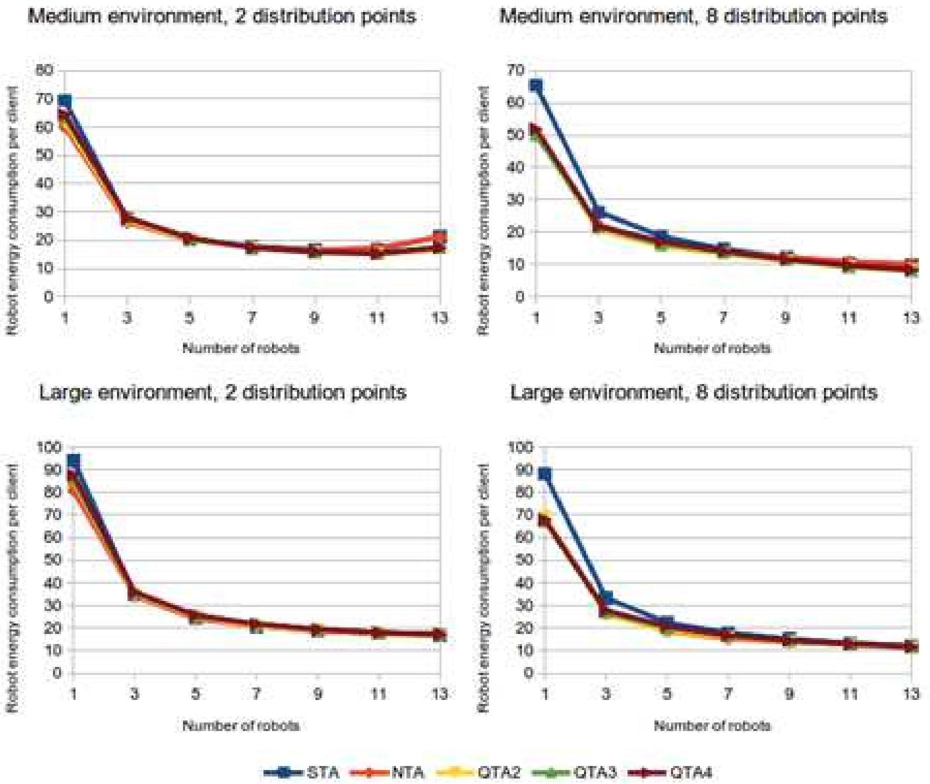

Fig. 12. The results of examination of the proposed algorithms in terms of robot energy consumption per serviced client as a function of the number of robots

4. CONCLUSION & FUTURE PLANS

Analysis of the results of examination shows that depending on the size of the environment and the number of available distribution points, there is an optimum number of robots. Detailed analysis revealed the coexistence algorithms, depending of the effectiveness of the two algorithms: the Queued Task Algorithm, with 1-element queueCapacity and Nearest Task Algorithm. Analysing the task scheduling for multi-robot environment, there was no relationship between efficiency and selection tasks. The differences occurred in the field of size of the environment. Much better results were obtained for medium environment, which as presented in the case of the Nearest Task Algorithm and Queued Task Algorithm, with 2-element queueCapacity. Increasing the size of the queue, Queued Task Algorithm in most cases did not increase their efficiency. In addition to the situation, in which the considered amount of energy, which is used to performed the tasks. Regards to this factor, were been designated the dependence of queue length for consumed energy. Reduced energy of consumption was observed with increasing length of the queue. Analyzing algorithms, which are presented in this article, was determined the course of action for authors. The aim of the authors, in further work will be to create a mechanism to prevent situation, when the customer must wait for service (receive a task). In this paper, this situation was not presented, but be aware that it can occur, for example when the start of point the tasks will be very remote. The next steps, should be tested the algorithms in terms of the distribution times of customer service.

BIBLIOGRAPHY

[1] He H., Shi L., 2010. The Application and Research of System Structure for Mobile Robot Path Planning Based on Multi-Agent, International Conference On Computer Design And Applications (ICCDA 2010), pp. 523-526.
[2] Hu X., 2008. Clonal Selection based Mobile Robot Path Planning, Proceeding of the IEEE, International Conference on Automation and Logistics, pp. 437-442.
[3] Ledziński D., Marciniak T., Maszewski M., Boroński D., 2015. Robot Actions Planning Algorithms in Multi-Agent System, Solid State Phenomena 2015, vol. 223, pp. 221-230.
[4] Kumar P., Sharma R.K., 2013. A Novel Task Scheduling Algorithm for Real-Time Systems, IEEE International Conference on Communication and Signal Processing, pp. 995-998.
[5] Yuan P., Moallen M., Patel R.V., 2004. A Real-Time Task-Oriented Scheduling Algorithm for Distributed Multi-Robot Systems, Proceedings of the IEEE, International Conference on Robotics & Automation, pp. 2562-2567.
[6] Zhang Y., Parker L.E., 2013. Multi-Robot Task Scheduling, IEEE International Conference on Robotics and Automation (ICRA), pp. 2992-2998.

ALGORYTMY PLANOWANIA ZADAŃ W ŚRODOWISKU WIELU ROBOTÓW

Streszczenie

Na przestrzeni ostatnich lat problem poprawnego harmonogramowania zadań, w środowisku wielu robotów, stał się przedmiotem wielu prac badawczych. Kluczem do tego problemu jest przydzielenie odpowiedniej ilości zadań do zrealizowania dla każdego robota oraz zaplanowanie optymalnej sekwencji ich wykonania. W celu minimalizacji czasu dostarczenia zasobu do punktu docelowego, zaprojektowano algorytmy planowania zadań. Proces ustalenia optymalnej ścieżki dla robota, biorąc pod uwagę sposób doręczenia zasobu, stał się dobrą strategią do zarządzania pracą robotów w wielu środowiskach. Przedstawione w niniejszym artykule algorytmy dotyczą procesu zarządzania i kontroli działań wykonywanych w środowisku logistycznym (magazyn), oparte na systemach miękkich czasu rzeczywistego. Celem autorów było opracowanie i porównanie algorytmów zarządzania w środowisku wielu robotów, pod względem liczby obsłużonych klientów oraz ilości zużytej energii na jej wykonanie. Aby potwierdzić skuteczność proponowanego podejścia, stworzono środowisko na którym przeprowadzono symulacje. Uzyskane wyniki wskazują, że proponowane podejście jest prawidłową drogą do uzyskania optymalizacji w planowaniu zadań i sekwencji ich wykonania, zwłaszcza dla algorytmu Queued Task Algorithm i Nearest Task Algorithm.

Słowa kluczowe: planowanie zadań, algorytmy sterowania robotami, systemy wieloagentowe, magazyn.