ANADOLU ÜNİVERSİTESİ

Bilim ve Teknoloji Dergisi A-Uygulamalı Bilimler ve Mühendislik
Cilt: 14 Sayı: 2 2013
Sayfa: 173-182

ARAŞTIRMA MAKALESİ /**RESEARCH ARTICLE**

**Uğur GÜREL [1], Osman PARLAKTUNA [2], Nihat ADAR [1], Selçuk CANBEK [1]**

# EN YAKIN KOMŞU SEZGİSELİ YAKLAŞIMI İLE MARKET TABANLI GÖREV DAĞITIMI

## *ÖZ*

Çok-robotlu uygulamalarda çalışılan ana konulardan biri görevleri robotlara dağıtmak ve bu dağıtılmış görevleri kullanan robotlar için etkin rotalar inşa etmektir. Bu makalenin esas amacı piyasa temelli görev dağıtım mimarisi aracılığıyla görevleri dağıtmak ve bilinen bir iç mekânda çok-robotlu sistemler için çarpışmanın olmadığı rotalar inşa etmektir. Görev dağıtımı için piyasa temelli mimari inşa edilmiş ve dört adet çoktürel gezgin robottan oluşan bir örnek takıma uygulanmıştır. Çalışmada, görevler iki kısıt uyarınca dağıtılmıştır: robotların toplam iş yapma sürelerini en aza düşürmek ve gruptaki her robotun görev robot uyum değerini en yükseğe çıkarmak. Bu kısıtlar bir karar parametresinin değeri değiştirilerek sağlanmıştır. Önerilen yöntemin etkinliğini göstermek için bu kısıtlar içinde karşılaştırmalar yapılmıştır. Rotaları inşa etmek için En Yakın Komşu sezgiseli ve Dijkstra'nın en kısa yol algoritmasının birleşimi kullanılmıştır. Çakışmayan rotaları inşa edebilmek için çarpışmaları saptayan ve çözen bir yöntem geliştirilmiştir. Çalışmada erkinler arasındaki iletişim için Açık Erkin Mimarisi (Open Agent Architecture) kullanılmıştır. Ek olarak önerilen yöntemin uygulanabilirliğini kanıtlamak için MobileSim platformunda benzetimler yapılmıştır.

**Anahtar Kelimeler:** Çok-erkinli, Görev dağıtımı, Market tabanlı, Yol planlaması, Çarpışmadan kaçınma.

# TASK ALLOCATION IN MARKET-BASED APPLICATIONS BY THE NEAREST NEIGHBOR HEURISTIC

## *ABSTRACT*

One of the main subjects that is studied in multi-robot applications is allocating tasks to robots and constructing objective-efficient tours for the robots using these allocated tasks. The main purpose of this paper is to allocate tasks via market-based task allocation architecture and to construct collision-free routes for multi-robot systems in a known indoor environment. For task allocation, a market-based architecture is constructed and applied to an example of a team of four heterogeneous mobile robots. In the study, tasks are allocated according to one of two constraints: minimizing the makespan of the robots and maximizing the robot task matching value of each robot in the group. These constraints are achieved by changing the value of a decision parameter. To show the effectiveness of the proposed method, comparisons are made within these constraints. To construct path, a combination of the Nearest Neighbor heuristic and Dijkstra's shortest path algorithms is used. To construct non-conflicting paths, a method that detects and solves collisions is developed. In the study, for agent communication Open Agent Architecture is used. Additionally, simulations on the MobileSim platform are conducted to verify the feasibility of the proposed method.

**Keywords**: Multi-agent, Task allocation, Market-based, Path planning, Collision avoidance.

[1] Eskişehir Osmangazi University, Faculty of Engineering and Architecture, Department of Computer Engineering, 26480, Eskişehir, Turkey,E-mail: ugurel@ogu.edu.tr, nadar@ogu.edu.tr, selcuk@ogu.edu.tr
[2] Eskişehir Osmangazi University, Faculty of Engineering and Architecture, Department of Electrical & Electronics Engineering, 26480, Eskişehir, Turkey,E-mail: oparlak@ogu.edu.tr

## 1. INTRODUCTION

In the mobile robot domain, many applications require robustness, efficiency and more than one task in parallel. To accomplish these requirements, first a proper and flexible architecture has to be constructed. Agent-based architectures are quite preferable for these types of architectures. Distributing each sub task to a software agent increases the robustness of the architecture because it may not affect whole system if a software agent fails. Only when this failed agent's ability disappears in the system, and the system may continue to work with the remaining agents' abilities.

In the literature, multi-robot task allocation (MRTA) has been given significant attention. The solutions to MRTA problems are addressed in the literature as centralized (Brumitt and Stenz (1998); Caloud et al. 1990), distributed (Ostergaard et al. 2001), and hybrid (Dias and Stenz 2002; Ko et al. 2003). In centralized approaches, all the plans are made by a single robot called the planner, and all the data used to build an optimal plan are gathered by the planner. However because it is performed by only one robot, any failure of the planner causes the whole team to fail. In the distributed approach, every team member of the robot team is responsible for its own plan generation based on its local information and states (Kaleci et al. 2010). There are several multi-robot architectures for a MRTA problem in a distributed approach. ALLIANCE (Parker 1998) is one of the leading studies in the literature. Another approach, developed by Sandholm, is based on contract net protocol (Sandholm 1993). In the study, market-based negotiation is applied based on the contract net protocol rules. Gerkey and Mataric have used a publish/subscribe communication methodology for their task allocation via distributed negotiation problem (Gerkey and Mataric 2000). In their studies, Hu et al. find a solution for an underwater box-pushing scenario. The box-pushing task consists of subtasks conducted by consecutively executing a series of behaviors; a market-based task allocation method is used to allocate these sub tasks to the robots via explicit communications (Hu et al. 2011). Ling and Stentz address environmental coverage with incomplete prior map information using multiple robots. A marked-based approach is used to allow robots to share particular sections of the map to more evenly divide the work (Ling and Stentz 2011). Khamis et al. cover a market-based solution to complex task allocation for mobile surveillance systems in their studies. Complex tasks are defined as the tasks that can be decomposed into subtasks. In the study, a comparison between different methods for a large number of robots and large areas is made (Khamis et al. 2011).

In the mobile robot domain, task allocation is an important subject. Parlaktuna et al. find a solution for an m-robot n-task allocation problem using multi-travelling salesman problem (m-TSP) heuristics (Parlaktuna et al. 2007). The aim of the m-TSP is to find tours for travelers such that the total distance travelled is minimized and all nodes are visited exactly once. Yu et al. describe a methodology to find a globally sub-optimal path for the robot group working on certain tasks by using a genetic algorithm method (Yu et al. 2002). Additionally, Guo and Parker propose a distributed and optimal motion planning algorithm for multiple robots (Guo and Parker 2002). Zu et al. used a depth-first search algorithm to find a path for a multi-robot system (Zu et al. 2002). Hasgul et al. in their studies built a project-oriented framework for a multi-robot task scheduling problem (Hasgul et al. 2009). Sariel and Balch address efficient bids for task allocation in their studies (Sariel and Balch 2006).

As a result, a combination of a market-based task allocation system and the Nearest Neighbor heuristic are used to allocate and determine the execution order of tasks in the study. The Nearest Neighbor heuristic is used to order tasks more efficiently during the auction, because the nearest Neighbor is one of the well-known heuristics used to solve m-TSP (Laporte 1992). The experiment's results showed the effectiveness of the proposed methodology. The rest of the paper is organized as follows. In section 2, the proposed agent architecture is explained. In section 3, the proposed methodology and the results of the application are presented. In the final section, conclusions and planned future work is explained.

## 2. AGENT ARCHITECTURE

In the proposed architecture, all the communications between agents are made via the Open Agent Architecture (OAA) structure. OAA is a framework for integrating a community of heterogeneous software agents in a distributed environment (Cheyer and Martin 2001). In Figure 1, the communication infrastructure is shown. In OAA, communication occurs through Interagent Communication Language (ICL) messages. ICL messages contain the sender, the receiver, and the message information in the message body.

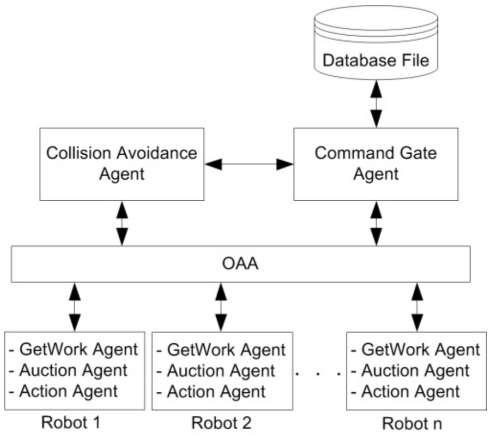

Figure 1. Communication Architecture via OAA

In the proposed architecture there are six different agents. These agents are named CommandGate, GetWork, Auction, Action, CollisionAvoidance, and MobileRobot itself. Only the MobileRobot agent is a physical agent; the other five agents are software agents (Guzzoni et al. 1997). Software agents named GetWork, Auction, and Action exist in each mobile robot, but CommandGate and CollisionAvoidance agents are unique in the architecture. The function of each agent is discussed below (Gürel et al. 2013).

## 2.1. CommandGate

The responsibility of this agent is to control and serialize concurrent multiple accesses to the database file. The database file stores tasks and auction data. Other queries or data store commands are processed via the CommandGate with special command sentences that are defined in the detailed explanation of each agent.

## 2.2. GetTask

This agent is responsible for gathering unallocated tasks from the database file via the CommandGate with a special command sentence. The form of the sentence is as follows:

NewTsk, Robot_Id

When this command is received by the CommandGate, it sends back the task information to the robot that sent the GetWork command. The structure of the answer is given below:

NewTsk, Task_Id, X_Pos_of_the_Task, Y_Pos_of_the_Task, Requirements_of_the_Task

If there is no unallocated task, an information message is sent back to the robot.

## 2.3. Auction

This Agent is responsible for managing the auction process for task allocation. With this agent, a mobile robot can hold an auction and receive bids from other robots or bid on an open auction process. Auction process start with the auction agents' special command sentence that asks if there is any unallocated task. To ask this, following sentence is used:

NewTSK, Robot_Id

All the auction process data are stored in the database file. To store a bid value the following sentence is used:

NewAuc, Robot_Id, Task, Cost

## 2.4 Action

This agent is responsible for mobile robot movements. It calculates the current position of the robot and gives necessary movement commands such as go to, hold, grip, release gripper, etc. to achieve assigned tasks.

## 2.5. CollisionAvoidance

During the experiments, the map in Figure 2 is used. In the map E denotes the edges, N denotes the position of the nodes and MR denotes the initial position of the mobile robots.

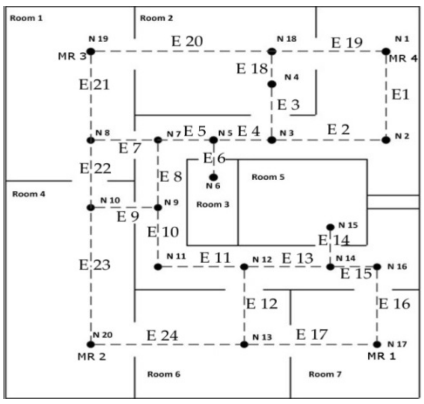

Figure 2. Map used in the study

The CollisionAvoidance agent is responsible for solving collisions on the paths of the robots. In the study, an edge is defined as the shortest path segment that connects two nodes $\{N_i, N_{i+1}\}$ such that there are no other nodes that exist in the path segment. Our main assumption is that a collision occurs on an edge when two or more robots try to pass through the same edge during the same time period. To determine the collisions, edge-time graphs for every robot are formed, as seen in Figure 3, and checked whenever there is an overlap between time slots.

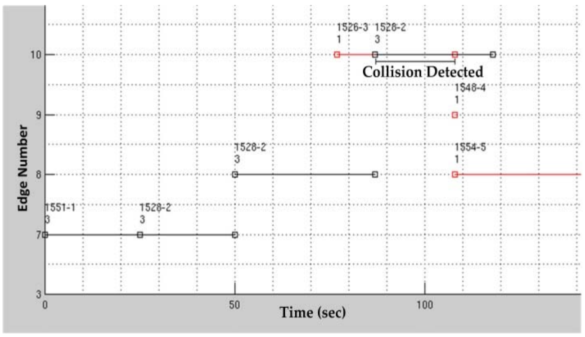

Figure 3. Edge-Time Graphic with Collision.

In the edge-time graphic, horizontal lines denote the time window that the mobile robot is going to occupy the related edge. The small squares located at the beginning and at the end of the horizontal line are used to emphasize the start and the end of the time window. The number located above the first small square of the time window shows the id of the robot that will occupy the related edge. The number above the robot id shows the task id.

As it can be seen in Figure 3, there is collision between robot 1 and robot 3 with the tasks 1526 and 1528, respectively. Because robot 1 uses edge 10 between 77.5 sec and 108.75 (sec) for task 1526 and robot 3 uses the same edge between 87.5 and 118.75 sec, there is a collision. To solve the collision, both an alternative route cost and a delay cost are calculated. An alternative route cost is calculated as the length of the new generated route without the edge where the collision occurs. Delay cost is calculated as the length of the time axis that is overlapped for both robots. The lower-cost solution is selected and applied to the edge-time plan. In this example because its cost is lower, the delay solution is selected and calculated delay is added to the edge-time graph. After the delay is added, the new edge-time graphic is shown in Figure 4, and as it shows there is no longer a collision.

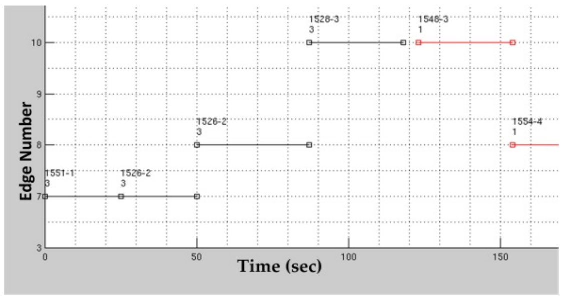

Figure 4. Edge-time graphic after collision is solved.

The block diagram and flow chart of each agent in a mobile robot are shown in Figure 5.

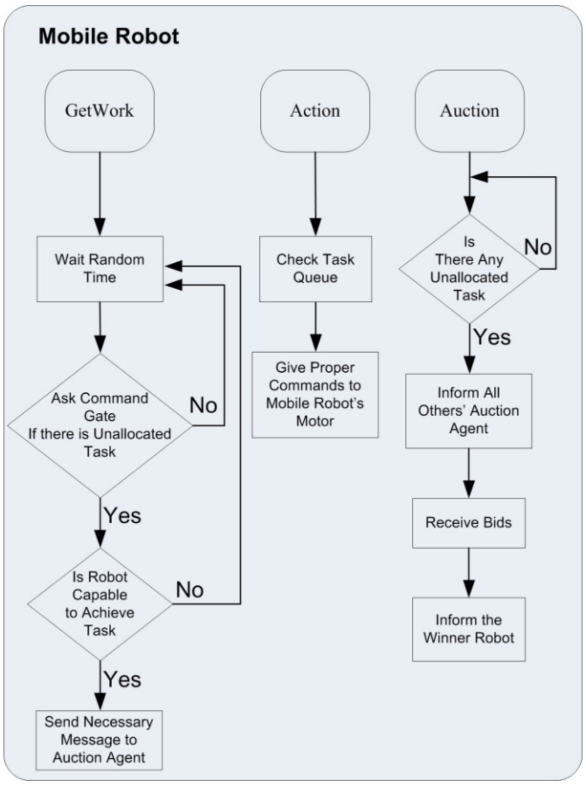
Mobile Robot
GetWork
Action
Auction
Wait Random Time
Ask Command Gate If there is Unallocated Task
No
Yes
Is Robot Capable to Achieve Task
No
Yes
Send Necessary Message to Auction Agent
Check Task Queue
Give Proper Commands to Mobile Robot's Motor
Is There Any Unallocated Task
No
Yes
Inform All Others' Auction Agent
Receive Bids
Inform the Winner Robot

Figure 5. Agents present in a mobile robot

## 3. TASK ALLOCATION, TOUR CONSTRUCTION AND SHORTEST PATH PROBLEMS, NEAREST NEIGHBOR HEURISTIC

In recent years, market-based approaches have become more and more popular as a means to solve task allocation problems. One of the main reasons behind this popularity is that by using market-based approaches, a user can combine the advantages of both centralized and distributed systems. With this approach, each member of the robot team builds its own plan using the robot's local information. In addition, mobile robot can participate in auctions to take unassigned tasks in the market. In market-based applications, there are two main roles. The first is the auctioneer (Auc$_{(i)}$, i=1,2, …n), and the second is bidder (Bid$_{(i)}$ i=1,2, …n). With the auctioneer role, a mobile robot can hold auctions for unallocated tasks and with the bidder role can submit bids for the unallocated tasks. A bid value for the task is calculated by using the distance that the robot travels between the nodes. The steps of the task allocation, task ordering, and collision avoidance processes are given in Figure 6.

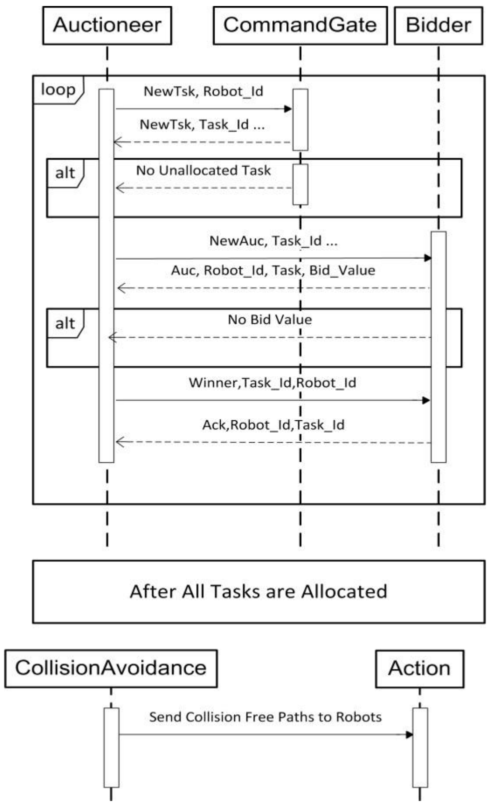

Figure 6. Sequence diagram for the auction and collision avoidance process

### 3.1. Predefined Tasks

In the study, tasks are defined with two parameters; the task node and the requirement word. The task node defines the spatial coordinates of the task. The requirement word is a three-character word that defines the robot abilities which are necessary for this particular task. In a requirement word, each character represents an ability that the robot must have to bid for the task. By using this requirement word, four different types of tasks are named in the study. Tasks and their requirement words are defined in Table 1:

| | | Abilities | | |
|---|---|---|---|---|
| **Task Name** | Requirement Word | Mobility and Localization | Gripper | Camera |
| **Visiting Node** | 100 | 1 | 0 | 0 |
| **Surveillance** | 101 | 1 | 0 | 1 |
| **Carry** | 110 | 1 | 1 | 0 |
| **Carry with Camera** | 111 | 1 | 1 | 1 |

Table 1. Requirement words

### 3.2. Announcement of Tasks

An announcement of unallocated tasks is made by the robot that has the auctioneer role. To announce a task, the task id, the spatial coordinates of the task, and the requirement word of the task have to be passed to the other robots. This announcement is made via an auction agent that is defined in the communication framework section. After the announcement of the task, an auctioneer waits for a predefined time period to receive bids from the robots, and chose the robot that has the best (in this study the lowest) bid and announces the winner to every robot. The winning robot adds this task to its task queue and sends task information to the collision avoidance agent.

### 3.3. Bidding Process:

After the announcement of the task, each robot checks whether it can achieve the task. To make this decision, a robot compares its ability word and the announced task's requirement word. The ability word is a special word that contains the robot's abilities. This is a three-character word consisting of ones and zeros. Similar to the requirement word at robot's ability word, if the value is one it means that the robot has that ability, and if it is zero the robot does not have that particular ability. A robot with adequate abilities calculates the bid value. To calculate the bid value, first the robot inserts the task virtually into its task list and sorts the tasks from nearest to farthest to obtain the execution order. While sorting the tasks the Nearest Neighbor heuristic is used. Basically, this heuristic can be summarized as: start from a start node, choose the nearest node as the next node and continue until there is no un-ordered task. The steps of the heuristic are listed below.

**Step 1:** Set the mobile robot's current node as start node

**Step 2:** Find the shortest edge connecting the current node and an unvisited node n

**Step 3:** Set current node to n

**Step 4:** Mark n as visited

**Step 5:** If there is no unvisited node then terminate

**Step 6:** Go to step 2

After the sorting process, a robot knows the execution order of its tasks, including the task that is on auction. To calculate the bid value of the task, the combination of the path lengths that is calculated by Dijkstra's shortest path algorithm (Dijkstra 1959) and robot task matching value of Robot k ($RTMV_k$) will be used. The robot task matching value is a metric that calculates the capability difference between the robot ability (RA) and the task requirement (TR). The $RTMV_k$ is calculated by using equations 1 and 2. Equation 1 is used to find the robot ability and task requirements pairs such that the j^th character of ability $RA_k(j)$ word is greater than $TR_k(j)$ word.

$$f(RA_k(j), TR_k(j)) = \left\{ \begin{array}{ll} 1 & \text{if } RA_k(j) > TR_k(j) \\ VLV & \text{if } RA_k(j) < TR_k(j) \\ 0 & \text{if } RA_k(j) = TR_k(j) \end{array} \right\} \quad (1)$$

In equation 1, a very large value (VLV) is defined as a value greater than the greatest possible RTMV value. To calculate $RTMV_k$ of Robot $_k$, equation 2 is used. Equation 2 is the count of pairs that are defined by equation 1 between the robot ability and the task requirements word.

$$RTMV_k = \sum_{i=1}^{3} f(RA_k(j), TR_k(j)) \quad (2)$$

In the study, tasks are allocated according to two constraints: minimizing the maximum makespan of the robot and maximizing the robot task matching value of the robot in a mobile robot group. Three different methods are proposed. Detailed explanations are given below.

**Full Task Repelling (FTR):** In this method, a robot rejects all tasks that have different abilities. The robot offers bids only for the tasks that perfectly match its abilities. The aim of the full task repelling (FTR) method is allocate tasks to robots that have a perfect robot ability and task requirement match. In FTR, robots cannot bid if there is a difference between a robot's ability word and a task requirement word; therefore, with this allocation method tasks are allocated to robots by maximizing the robot task matching value.

**Semi Task Repelling (STR):** In this method, robots can offer bids even there is no perfect match. Using this type of task allocation, a robot can bid for the task, but while calculating the bid value there is an additional cost if there is difference between a robot's ability word and a task requirement word. The aim of this allocation method is to allocate tasks based on both the robot task matching value and distance of the task.

**Non Task Repelling (NTR):** With this method, any robot can offer a bid to any task if the robot's ability word is suitable for the task. The aim of this task allocation method is allocate tasks only according to the distances of the tasks.

The Robot $_k$ Bid value for the task is calculated as:

$$Robot_k Bid = \left( 1 + RTMV_k \right)^{\tau_k} \times STPL_k \quad (3)$$

First, the sorted task path length ($STPL_k$) of the announced task is calculated. To calculate this value the following steps are applied.

**Step 1:** Find the execution order of the announced taskk in the robot's task queue.

**Step 2:** Find the Dijkstra’s shortest path length of each task individually, starting from the first task until this announced taskk in the queue.

**Step 3:** Sum each tasks’ path length

In equation 3, τk is used as a decision parameter between minimizing average makespan and maximizing robot task matching value. If τk increases, the system allocates tasks according to maximizing the robot task matching value, and if τk goes to zero the system will allocate tasks to minimize average makespan. For different type of task allocation methods, values for τk and the conditions for RTMVk are listed in Table 2.

| Method Name | $\tau_k$ | Condition |
|---|---|---|
| **Full Task Repelling (FTR)** | Any Positive Value | $RTMV_k = 0$ |
| **Semi Task Repelling (STR)** | $\tau_k = 1$ | $RTMV_k > 0$ |
| **Non Task Repelling (NTR)** | $\tau_k = 0$ | $RTMV_k > 0$ |

Table 2. Parameter values and conditions for the proposed methods

## 3.4. Collision Detection and Avoidance

Collision Detection and Avoidance*:* In the study, it is observed that there are some collisions in the robots' paths. To solve these collisions, a method is developed. The collision detection and avoidance is conducted by the CollisionAvoidance agent.

## 3.5. Announce New Task Execution Plan

After trying to solve all collisions, new edge-time graphs are formed, and these plans are passed to the related robots via our agent framework for the execution of the tasks.

## 4. IMPLEMENTATION OF PROPOSED AGENT ARCHITECTURE AND EXPERIMENT RESULTS

To show the effectiveness of the proposed method, several experiments are performed on the MobileSim simulator platform (Adept MobileRobots 2013). The ability word of the four mobile robots is listed in Table 3.

| | Ability Word |
|---|---|
| **Mobile Robot 1** | 100 |
| **Mobile Robot 2** | 101 |
| **Mobile Robot 3** | 110 |
| **Mobile Robot 4** | 111 |

Table 3. Ability word of the mobile robots

During the experiments, forty tasks are generated randomly using a roulette wheel method, according to the possibilities listed in Table 4.

| Requirement world of the task | Possibility % |
|---|---|
| **100** | 75 |
| **101** | 10 |
| **110** | 10 |
| **111** | 5 |

Table 4. Probability ratios that are used in the wheel roulette method

The experiments are repeated ten times, and the average value of a robot’s makespan according to three different task allocation methods is listed in Table 5.

| | FTR | STR | NTR |
|---|---|---|---|
| **Experiment** | Makespan | Makespan | Makespan |
| **1** | 438 | 437 | 290 |
| **2** | 475 | 417 | 373 |
| **3** | 550 | 479 | 303 |
| **4** | 495 | 396 | 323 |
| **5** | 617 | 522 | 368 |
| **6** | 604 | 453 | 494 |
| **7** | 629 | 490 | 501 |
| **8** | 500 | 420 | 299 |
| **9** | 555 | 380 | 415 |
| **10** | 593 | 483 | 473 |
| **Average MakeSpan** | **546** | **448** | **384** |

Table 5. Experiment results if collisions are ignored

The results that are listed at Table 5 are obtained for a map that has enough large corridors that no collisions occur, but in real life it is not possible; there exist collisions. As we explained before, collisions are detected by the CollisionAvoidance agent, which tries to solve them. After collisions are solved, we obtain the results in Table 6.

| | FTR | | STR | | NTR | |
|---|---|---|---|---|---|---|
| **Experiment** | **Average Makespan** | **Detected Collisions** | **Average Makespan** | **Detected Collisions** | **Average Makespan** | **Detected Collisions** |
| **1** | 438 | 0 | 476 | 4 | 295 | 3 |
| **2** | 481 | 8 | 479 | 16 | 424 | 10 |
| **3** | 727 | 4 | 668 | 29 | 316 | 2 |
| **4** | 518 | 6 | 447 | 12 | 439 | 11 |
| **5** | 723 | 16 | 605 | 16 | 378 | 2 |
| **6** | 743 | 9 | 492 | 10 | 515 | 4 |
| **7** | 691 | 8 | 724 | 24 | 634 | 7 |
| **8** | 547 | 8 | 486 | 12 | 370 | 12 |
| **9** | 593 | 6 | 521 | 11 | 471 | 8 |
| **10** | 598 | 4 | 664 | 19 | 541 | 28 |
| **Average** | **605.9** | **6.5** | **556.2** | **16.55** | **438.3** | **9.33** |

Table 6. Experiment results if collisions are considered

In Table 6, all three methods' results are given. If we look at the results that are at Tables 5 and 6, at both tables there is a decreasing trend from Full Task Repelling (FTR) to Non Task Repelling (NTR) in the average makespan row. The main reason for this decreasing is, at Full Task Repelling (FTR) method tasks are allocated to the robot that has perfect match between the robot ability and task requirement word. That's why longer paths with perfect robot task match are chosen. But at Non Task Repelling (NTR) method tasks are allocated according to the shorter paths without considering robot task match. Because of this difference in NTR method produce shorter makespan with robot ability waste and opposite to this FTR produce longer makespan with perfect robot task match. And Semi Task Repelling (STR) method produce values between NTR and FTR. These three methods are shown in Figure 7.

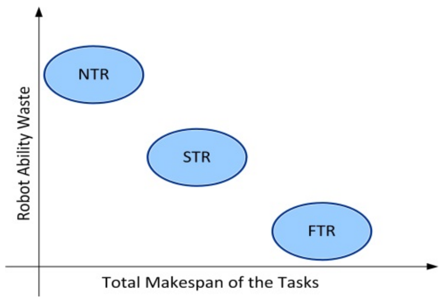

Figure 7. Three task allocation methods

The average makespan values are lower at table 5 because in Table 5 collisions are not taken into account, but in Table 6 collisions are solved either by applying alternative paths or inserting delays in the robots' task execution plans, that causes increase in makespan of the robot.

## 5. CONCLUSION

In the study, a market-based architecture is constructed and tested for a task allocation problem by using a Mobilesim simulation platform.

In the proposed architecture, each mobile robot contains an auctioneer agent in the group, so that every robot can host an auction or bid for the open auction. The proposed auction architecture is decentralized in terms of the auction process. A decentralized auction allows us to use heterogeneous robot decisions (i.e., while one robot forces an ability word, the other can force the shortest path without ability restriction). Because all data are stored in the database file and access to this file is controlled via the CommandGate agent, the architecture is centralized in terms of data storage.

Tasks are allocated to the most suitable robot according to two constraints: minimizing the average makespan of the mobile robots and maximizing the robot task matching value by changing the decision parameters. The experiment results show that by adjusting the τk parameter, the proposed method solves the task allocation problem while considering the collisions and the constraints. In the future, it is planned that different decision parameter τk values will be tested for each robot, to allocate tasks with heterogeneous constraints.

## REFERENCES

Brumitt B. and Stenz A. (1998). Grammps: a Generalized Mission Planner for Multiple Mobile Robots. *In Proceedings of the IEEE International Conference on Robotics and Automation.*

Caloud P., Choi W., Latombe J., Pape C. Le, And Yim M. (1990). Indoor Automation with Many Mobile Robots. *In Proceedings of the IEEE International Workshop on Intelligent Robotics and Systems (IROS).*

Cheyer A., and Martin D. (2001). The Open Agent Architecture. *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 4, no. 1, 143-148.

Dias M. B. and Stenz A. (2002). Opportunistic Optimization for Market-based Multirobot Control. *In Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems* 2714–2720.

Dijkstra E.W. (1959). A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*. Volume 1, Issue 1, 269-271.

Gerkey B.P. and Mataric M.J. (2000). Murdoch: Publish/Subscribe Task Allocation for Heterogeneous Agents. *In Proceedings of the Fourth International Conference on Autonomous Agents*. 203–204.

Guo Y. and Parker L. E. (2002). A Distributed and Optimal Motion Planning Approach for Multiple Mobile Robots. *In Proceedings of IEEE International Conference on Robotics & Automation* Washington DC 2612-2619.

Gurel U.,Adar N, Parlaktuna O. (2013). Priority-based Task Allocation in Auction-based Applications. *Innovations in Intelligent Systems and Applications (INISTA)*, IEEE International Symposium , 19-21 June 2013 , Albena Bulgaria.

Guzzoni D., Cheyer A. , Julia L., and Konolige Kurt. (1997). Many Robots Make Short Work Report of The Sri International Mobile Robot Team. *AI Magazine* Volume 18 Number 1 55-64.

Hasgul S., Saricicek İ., Ozkan M., Parlaktuna O. (2009). Project-oriented Task Scheduling for Mobile Robot Team. *Journal of Intelligent Manufacturing*, Volume 20, Issue 2, 151-158.

Adept MobileRobots. http://www.mobilerobots.com/Software/MobileSim.aspx.
Accessed at 15.10.2013 .

Hu Y., Wang L., Liang J., Wang T. (2011). Cooperative Box-pushing with Multiple Autonomous Robotic Fish in Underwater Environment. *Control Theory & Applications,* IET Volume: 5 Issue: 17 2015 – 2022.

Kaleci B., Parlaktuna O., and Ozkan M. (2010). Market-based Task Allocation Method using Exploratory Learning for Heterogeneous Multi-robot Team. *International Symposium on Innovations in Intelligent Systems and Applications (INISTA 2010)*. Kayseri, Turkey.

Khamis A. M., Elmogy A. M., Karray F. O. (2011). Complex Task Allocation in Mobile Surveillance Systems. *Journal of Intelligent & Robotic Systems.* Volume 64, Issue 1 33-55.

Ko J., Stewart B., Fox D., Konolige K. and Limketkai B. (2003). A Practical Decision-Theoretic Approach to Multi-robot Mapping and Exploration. *In Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems.* 2714–2720.

Laporte G. (1992). The Vehicle Routing Problem: an Overview of Exact and Approximate Algorithms. *European Journal of Operational Research.* Vol 59 345-358.

Ling X., Stentz, A. (2011). Market-based Coordination of Coupled Robot Systems. *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference*. 2784 – 2789.

Ostergaard, E.H., Mataric, M.J., Sukhatme, G. (2001). Distributed Multi-robot Task Allocation for Emergency Handling. *G.S.Intelligent Robots and Systems, Proceedings.* IEEE/RSJ International Conference on Volume: 2 821 - 826.

Parker L. E. (1998). Alliance: An Architecture for Fault-tolerant Multi-robot Cooperation. *IEEE* Transactions *on Robotics and Automation.* 14(2) 220–240

Parlaktuna O., Sipahioglu A., Yazıcı A. (2007). A Vrp-based Route Planning for A Mobile Robot Group. *Turk. Journal Electric Engineering*. 15 187-197

Sandholm T. (1993). An Implementation of The Contract Net Protocol based on Marginal Cost Calculations. *In Proceedings of the 12th International Workshop on Distributed Artificial Intelligence*. 256-262

Sariel S. and Balch T.R. (2006). Efficient Bids on Task Allocation for Multi-robot Exploration. *In Proc FLAIRS Conference.* 116-121.

Yu Z., Jinhai L., Guochang G., Rubo Z. and Haiyan Y., (2002). An Implementation of Evolutionary Computation for Path Planning of Cooperative Mobile Robots. *In Proc 4"' World Congress on Intelligent Control and Automation.* 1798-1802

Zu L., Tian Y. and Liu J. (2004). Algorithms of Task-allocation and Cooperation in Multi Mobile Robot System. *In Proc 5" World Congress on Intelligent Control and Automation, Hangzhou, P.R. China*. 2841-2845.