A Multi-Level Architecture for Solving the Multi-Robot Task Allocation Problem Using a Market-Based Approach

Ali B. Bahgat, Omar M. Shehata, and El Sayed I. Morgan
Multi-Robot Systems (MRS) Research Group, German University in Cairo, Cairo, Egypt
Email: mrs.lab.guc@gmail.com

***Abstract*— Several applications have been implemented since the advent of mobile robots. Such applications include, but are not limited to, performing search and rescue missions and optimizing the material handling process in automated warehouses. This high interest in the field of cooperative Multi-Robot Systems is due to their robustness and flexibility, in addition to their high performance and affordability. In this study, an algorithm for the Multi-Robot Task Allocation problem has been developed to incubate the two well-known approaches for system architectures which are centralized and decentralized approaches. The algorithm utilizes the advantages of both approaches in a Multi-level structure and uses an economic, Market-Based Approach, to solve the problem. Several attributes have been considered and encapsulated in the robots and tasks such as the energy levels of the robots, which play an important role in solving the task allocation problem. The algorithm has been tested on three different known environments with different complexities and sizes. Several scenarios have been tested by varying the number of robots and tasks. Results show high performance of the algorithm and its applicability in solving the Multi-Robot Task Allocation Problem.**

***Index Terms*—multi-robot systems, cooperative robots, auctioning, market-based approach, multi-level architecture, multi-robot task allocation**

I. INTRODUCTION

The emergence of affordable mobile robots opened several research directions. These research directions aimed to utilize the flexibility and robustness of the newly introduced technology and extend the applications of autonomous systems to diverse domains. These domains included the usage of collaborative robots in search and rescue missions [1], industrial automation, surveillance and security, warehouse material handling [2], etc.

One of the major complex problems researchers face during execution of any of the above-mentioned applications is the task allocation problem [3]. The Multi-Robot Task Allocation (MRTA) problem aims to find an optimal or suboptimal solution for the following question: “Which robot will be responsible to handle which task such that the overall cost of the team is to be minimized?” Previous research in this field shows that there are two main approaches for this problem [4]. One approach is to use a centralized agent. The advantage of this approach is that it provides an optimal solution to the MRTA problem. The disadvantage of this approach is that any computation error or hardware malfunction in the centralized agent could result in a complete failure of the system. It could also be untraceable and slow for large scaled problems.

The other approach is to fully distribute the system. In this approach, each robot is responsible to carry out its own calculations to choose the task that it sees fit. The advantage of this approach is that there is not a single point of failure to the system and is better in terms of computation time. However, the main disadvantage is that it provides suboptimal solutions compared to the centralized approach.

In this article, a hierarchical architecture that utilizes both central and distributed approaches integrated together in a multi-level structure is proposed. The upper level is the centralized agent (This computer), while the lower level is the distributed system (The team of simulated robots) that uses an economic market-based approach to solve the MRTA problem.

The remainder of the paper is organized as follows:

Section 2 presents the previous work in the field of Multi-Robot Systems and the MRTA problem, followed by Section 3, which introduces the problem formulation and the algorithm design. In Section 4, the results and discussion, the selected scenarios, and the evaluation metrics are described. Finally, the conclusion and future work are summarized in Section 5.

II. BACKGROUND

Research in the field of collaborative mobile robots has emerged approximately four decades ago since the work published in 1978 by Beneš, J. and Kolář, P. in using artificial intelligence for solving the routing problem for multiple rovers in an unknown area [5]. In the following years, a lot of new domains of research has been opened and investigated in this field.

In [6], a scalable multi-robot task allocation algorithm has been implemented. The authors were motivated by

Manuscript received November 7, 2018; revised August 29, 2019.

doi: 10.18178/ijmerr.9.2.293-298

the Material Handling Problem (MHP) in modern warehouses. The MRTA problem was modeled as an instance of the Capacity-constrained Vehicle Routing Problem (cVRP) which is known to be NP-hard. The study presents a heuristic, called nearest neighbor based Clustering And Routing (nCAR) which proved to have better execution time compared to the already available state-of-the-art heuristics. The algorithm was tested on several scenarios.

A combinatorial auction mechanism has been implemented for robot exploration in an unknown terrain task in [7]. Combinatorial auctions differ from single-item auctions in the bidding strategies of the robot. Synergies between tasks are taken into consideration as robots are set to bid on a bundle of tasks instead of single-task bidding, changing the problem from the widely known single-task and single-robot (ST-SR) to a multi-task and single-robot (MT-SR) assignment problem [8]. Different combinatorial auctioning strategies were studied and compared to each other as well as to the single-item auction bidding strategy.

Swarm intelligence has been used to solve the MRTA of a Multi-Agent System (MAS) formed by a group of Unmanned Aerial Vehicles (UAVs) in [9]. A central agent has been used to create the tasks whilst the MRTA was solved using a decentralized approach by the agents themselves. The authors present three complementing algorithms that form a new method aiming to increase the amount of performed tasks.

In [10], a comparative study between a market-based approach and an optimization-based approach for solving the MRTA problem was held. The algorithms have been tested on a team of heterogeneous robots and a set of heterogeneous tasks. The results showed that the optimization-based approach outperformed the market-based approach in terms of the best allocation and the computation time.

Ahmed Hussein et al addressed the MRTA for search and rescue missions in [11]. A generic market-based approach was proposed and quantitatively evaluated through simulation and real experimentations using heterogeneous Khepera-III mobile robots. The results showed high performance of the proposed algorithms and their applicability in search and rescue missions.

Dong-Hyun Lee investigated a resource based task allocation for Multi-Robot Systems (MRS) in [12]. Robots monitor their energy levels and they go for the nearest charging station to refill their batteries when their energy levels fall below a certain, predetermined, value.

Contract Net Protocol (CNP) was used in [13] to solve the MRTA problem using a market-based approach. The aim was to allocate tasks to vacuum cleaning robots that need to cooperate for cleaning an area that is beyond the capabilities of a single robot. Results showed that using the CNP protocol alone is not sufficient to provide acceptable solutions for the MRTA problem.

In [14], the MRTA problem was addressed on a larger scale. The team consisted of self-driving vehicles. The objective was to introduce a hybrid optimization-based approach to solve the problem of multiple intelligent vehicles requests allocation as an instance of MRTA problem, to find an optimized solution as per the objective function. A comparative study is conducted between the obtained results and the already available suboptimal results.

III. Methodology

A. Problem Formulation

A map with a known terrain, a predetermined size, and exact obstacles' locations is introduced as a binary occupancy grid. A binary occupancy grid is represented as a matrix of zeros and ones. A cell is marked as obstacle free if its value is equal to zero and is marked as an obstacle if the cell value is equal to one. The map is processed and converted to a directed graph to ease up any path planning algorithm for the robots' motion. A directed graph is one type of graphs that deals with each cell in a matrix as an individual component that has some attributes. Each cell is connected with neighboring nodes or cells with a path of a specific distance. In this study, the distance between each cell and its neighboring cell is 1 meter. Diagonal distances are not considered because the robot is not allowed to commute diagonally in the workspace. The map contains main pieces of information that must be known before being submitted to the central agent for solving the MRTA problem. The pieces of information are encapsulated in the following four categories:

1. Layout information and restrictions:

Every map has its own specifications such as its size, obstacles' locations, and the nature of the obstacles whether they are static or not. The map could also have a set of rules restricting the robots behavior, such as one-way paths and the points at which the robots are allowed to change their directions. In this study, the size of the map is known as well the locations and nature of the obstacles. Obstacles considered are only static ones. Map restrictions are not considered in this study.

2. Robots:

Each robot has a minimum set of attributes that are carefully measured and taken into account for solving the MRTA problem. These set of attributes could be but are not limited to:

a- Robot ID number   b- Velocity
c- Current Location   d- Energy Level

All robots are initially deployed with a 100% energy level and discharge with a constant discharge rate that is directly proportional to the total travelled distance by the robot. Other attributes are needed that complement the main attributes such as:

a- Distance Travelled   b- Completed Tasks
c- Current Task   d- Current Status

Distance Travelled accounts for determining whether the robot in hand is overloaded compared to the rest of the robot team. It is also useful for visualizing results and measuring the individual and team performance.

Current Task holds the information related to the task that the robot is currently executing such as its nature and location.

Completed Tasks is a list that holds inside the history of the robot and reveals all the tasks that this specific robot executed. This is useful to know how much did this robot contribute in achieving the overall objective of the team.

Current Status is the attribute that reveals the availability of the robot or not. The value could hold any of the following values:

a- Available   b- Battery Low
c- Committed   d- Charging

Other values could be added, but are out of scope of this study. These attributes could be added to indicate, for example, whether the robot needs its periodic maintenance or not.

3. Tasks:

Tasks are the duties that should be assigned to each robot. Each task has the following characteristics:

a- Task ID   b- Nature
c- Location   d- Status
e- Nearest Charging Stations

Nature of the task indicates what exactly the robot is supposed to do when it is committed to the execution of a task. In this study, the robots are required to only *visit* the locations of the allocated tasks. However, the nature of the task may vary according to the application.

Status of the task is concerned with whether this task has been completed or not. A task may be *available*, *allocated*, or, *completed*. *Available* means that the task has not yet been allocated to any of the robots. *Allocated* means that the task has been assigned to one of the team members but it is still in the execution phase. *Completed* means that the task has been already assigned to one of the robots and that it has been successfully executed.

Nearest Charging Stations is a list containing the nearest charging stations to this task in ascending order. Having this attribute linked to every task simplifies the decision that the robot should take concerning which charging station is it supposed to visit for recharging.

4. Charging Stations:

Charging stations are the places where the robots recharge their batteries. Each charging station has an *ID*, *location* and a *capacity*. The *capacity* of the station indicates how many robots this charging station can accommodate at the same time.

In this article, each charging station is designed to carry a maximum of two robots at the same time.

B. Algorithm Perspective

The MRTA problem is known to be one of the most complex problems to solve. It needs a robust algorithm to be able to handle most of the cases that might occur whilst solving the problem. The algorithm should monitor the change in the situation of both the map and the robots' status.

The controller cannot handle the expense of solving the whole problem in a single iteration because it will not be able to predict the behavior of the robots and may not consider their updated locations when assigning new tasks to them. Hence, the controller is able to solve the problem by assigning one and only one task to each *available* robot per iteration. The status of each robot and each task is then checked and based upon the updated information, the next batch of tasks are assigned and so on until there are not any unexecuted tasks left in the market. The detailed algorithm is shown in Algorithm 1 and a description of its functions is discussed afterwards.

Algorithm 1: Market-Based Approach Using a Multi-Level Architecture

---

Input: tasks list *tasks*, robots list *robots*, charging_stations list *charging_stations*, map matrix *p*, map directed graph *DG*
Output: robots, tasks, status summary

Bidding List *bidding_list*
Available Tasks *availTasks*
Available Robots *availRobots*
Allocation Summary *allocation_summary*
Best Allocation *best_allocation*
availTasks ← tasks
while (!*isEmpty*(*availTasks*)) **do**
  bidding_list ← getBids(*availRobots, availTasks, p, DG*)
  allocation_summary ← initialAllocation(*bidding_list, availTasks, availRobots*)
  best_allocation ← negotiate(*allocation_summary, availRobots, robots, availTasks, tasks, p, DG*)
  [*robots, tasks, charging_stations]* ← update(*robots, tasks, charging_stations*)
  status_summary ← Report(*robots, tasks*)
end

A map with fixed locations and number of tasks *(n)*, and fixed locations and number of robots *(m)* is considered. The algorithm begins by assigning all tasks as *available* and then, it proceeds with the higher level controller. All robots are set to be *available*. The robots begin submitting bids on the available tasks using the *getBids* function. Their bids totally depend on the distance between the robot's current position and the task it is trying to win. The bids are calculated using the formula:

$$bid_{ij} = \sum_{i=1}^{n} \sum_{j=1}^{m} d_{ij} \tag{1}$$

The robot's motion is restricted in four unique directions; *North*, *East*, *South*, and *West*. Path planning algorithms must be proposed and tested. In this article, two different, optimization based, path planning algorithms have been proposed: **B**readth-**F**irst **S**earch (BFS) [15] and the Dijkstra algorithms [16].

The auction design is not time restricted as the auctioneer keeps receiving bids from the robots until all the bids from all *available* robots on all *available* tasks are submitted. After the robots submit the bids, they wait for their assigned tasks from the higher level controller. This initial allocation is received as a list named *allocationSummary* in a format as the one shown in Table I.

TABLE I. ALLOCATION SUMMARY LIST FORMAT

Allocated Task$_j$	Robot$_i$	Bid$_{ij}$

The *allocation Summary* list is created after opening all the submitted bids by the robots. It assigns one task per one robot according to what robot is seen as the fittest for this specific task. The task on which the robot submitted its least bid is awarded to this robot. This is illustrated by a two robots (R1 and R2) and two tasks (T1 and T2) scenario as shown in Figs. 1 and 2.

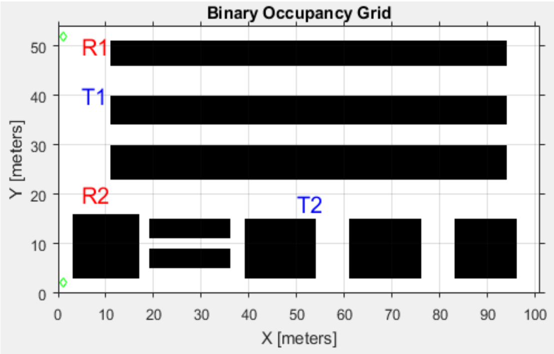

Figure 1. A two robots and two tasks scenario in Map 1

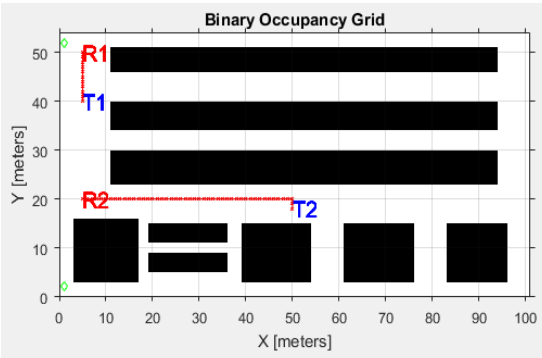

Figure 2. T1 is allocated to R1 and T2 is allocated to R2

After the robots receive their allocation summary, they begin investigating if there are better ways for carrying out the task. In this phase, robots begin negotiating with each other about the tasks they have been awarded in an economic point of view. This happens by revealing the awarded tasks along with the bids that made them win this task. After the information is shared locally within the group, robots having clashes with other robots begin negotiating. Meaning that if two robots have been awarded the same task because both of them have submitted their lowest bid on this particular task, they both agree that the robot who has submitted the lowest bid on this task at the first place will carry it out while the other must forfeit this task and search for its second best task within the available tasks in the market.

After the robots agree and settle upon their assigned tasks, they begin carrying out the tasks that they are committed to by planning their paths using any of the path planning algorithms and following the trajectory of the path. When all the robots finish the tasks in hand, their attributes are updated as well as the attributes of the tasks. Afterwards, a summary of the status of all robots and tasks is created and sent back to the central agent in order to resolve the MRTA problem for the next *m* tasks.

Some robots' energy levels will decline beneath the specified threshold specified by the system. Once this happens, the robot with low energy level sets its status to be unavailable and refrain from participating in the next auction round. Instead, it begins moving to the nearest charging station in order to recharge its battery. The robot keeps out of service for one bidding round and returns in the next bidding round when the auctioneer reopens the auction. The process is repeated until all tasks are completed.

IV. Experiments and Results

A. Evaluation Metrics

In this section, the proposed algorithm is evaluated based upon three different aspects — (i) average processing time (seconds), (ii) total distance travelled, and (iii) task execution time (seconds). The time taken by the robots to execute the tasks is measured per bidding round. This means that all robots must wait until the execution of the longest current task before entering the new bidding round. This results in losing some time for some of the robots that have to stand still in an idle status until the whole team is done with the tasks in hand.

The dynamics and the kinematics of the robots are out of scope of this study. This implies ignoring the angular velocity of the robots when changing directions. Also, the task allocation process is considered of minimal time and hence, not calculated when calculating the time taken by the robots to execute the tasks.

In testing out the algorithm, three different maps with different sizes and complexities have been tried out — (i) small map (Map 1), (ii) medium map (Map 2), and (iii) large map (Map 3). Each map is introduced as a .PNG image which is analyzed using image processing techniques by the algorithm. The small map is a 101*54 pixels map, the medium map is a 106*102 pixels map, and the large map is a 200*150 pixels map. The number and locations of robots and tasks were varied from one test case to another. All maps were tested with 2, 4, 8, and 16 robots against 1, 2, 4, 8, 16, 32, and 48 tasks and 2 charging stations. The robots are assumed to move with a constant velocity of 1.5m/s and their batteries are designed to last for only five minutes (450 meters). A robot changes its status from "available" to "battery low" whenever its battery consumes 50% of its capacity. All possible combinations have been considered when mixing and matching the number of robots and tasks. All experiments are carried out on a standard computer (Intel "Sandy Bridge" generation, i5 processor with dual core and 8GB memory).

B. Comparative Study

A comparative study between the three maps is held in terms of the evaluation metrics discussed before. The results are displayed in Table II using BFS and in Table III using the Dijkstra algorithm.

TABLE II. Results of the MRTA Using BFS Path Planning

	# robots	# tasks	Total distance (m)	Execution time (s)	Processing time (s)
Map 1	16	48	1154	94	21.982949
Map 2	16	48	1199	94.67	48.654827
Map 3	16	48	2086	230.67	134.31610

TABLE III. RESULTS OF THE MRTA USING DIJKSTRA PATH PLANNING

	# robots	# tasks	Total distance (m)	Execution time (s)	Processing time (s)
Map 1	16	48	1154	94	8.29401033
Map 2	16	48	1199	94.67	18.8017864
Map 3	16	48	2086	230.67	51.7022466

These results are based upon randomly selected initial starting positions of the robots and tasks. The results may be affected if the initial locations of the robots and tasks are altered. Fig. 3 shows the performance of the MRS with respect to the distance travelled by the team of mobile robots in Map 3 using the BFS path planning algorithm. Fig. 4 illustrates the performance of the MRS with respect to the tasks' execution time in Map 3 using the BFS algorithm. Fig. 5 compares the computational time taken by the processor to solve the MRTA problem in Map 2 when the Dijkstra algorithm was used.

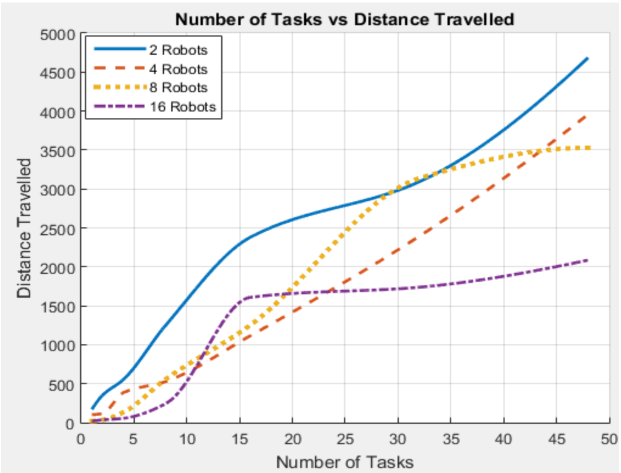

Figure 3. Based on Map 3 (BFS)

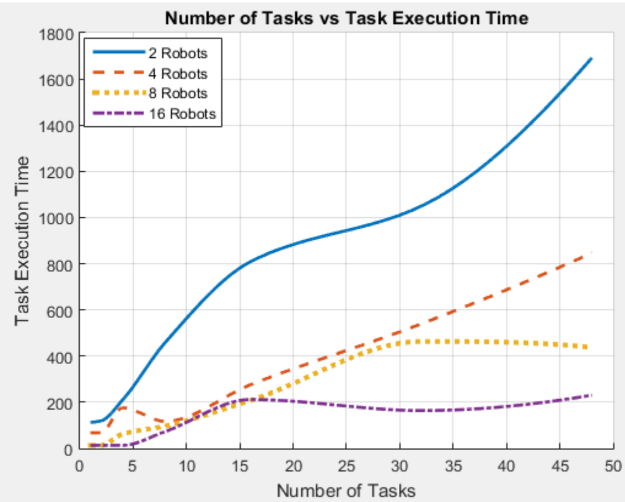

Figure 4. Based on Map 3 (BFS)

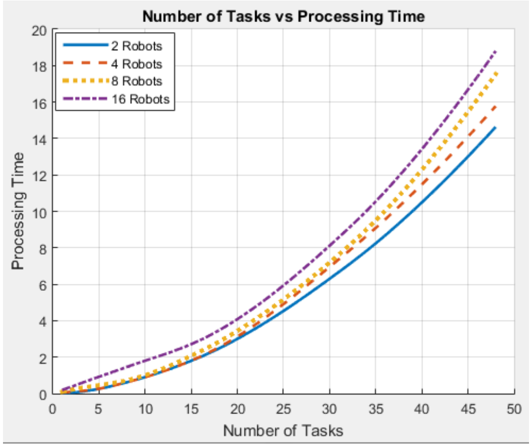

Figure 5. Based on map 2 (Dijkstra)

V. CONCLUSION

This study presented a market-based approach to solve the MRTA problem based upon a multi-level system architecture that utilizes the advantages of the centralized approach along with the distributed approach in a hierarchical fashion.

Results produced are optimal in terms of selecting the shortest path for every robot. Results show that as the size of the team increases, the overall distance travelled by the group is decreased. The task execution time is also decreased. However, increasing the size of the group comes on the expense of the processing time which increases with any extra group member. It is shown that the Dijkstra algorithm outperforms the BFS algorithm in terms of computation time. The results imply the applicability of the proposed algorithm to solve the MRTA problem with high performance and its robustness to any proposed map regardless of its complexity.

A. *Future Work*

Some limitations were found in the presented work. The first limitation is the time wasted by each robot when it has already finished its task and waiting for the rest of the team to execute their tasks in hand in order to enter the next bidding round. Another important aspect that should be considered is using combinatorial auctioning. In other words, synergies between tasks' locations should be taken into account when solving the MRTA problem. Implementing an intersection control algorithm is also mandatory to avoid clashes between robots while traversing the environment. More attributes could be added to the robots and tasks.

CONFLICT OF INTEREST

The authors declare no conflict of interest

AUTHOR CONTRIBUTIONS

Ali B. Bahgat conducted the research. Omar M. Shehata and Ali B. Bahgat wrote the paper. El Sayed I. Morgan made the comparative studies. Omar M. Shehata and El Sayed I. Morgan approved the final version of the paper.

REFERENCES

[1] R. Singh, R. Samkaria, A. Gehlot, and S. Choudhary, "Design and development of IoT enabled multi robot system for search and rescue mission," 2018.

[2] H. Ma, X. Wu, Y. Gong, Y. Cui, and J. Song, "A task-grouped approach for the multi-robot task allocation of warehouse system," in *Proc. 2015 International Conference on Computer Science and Mechanical Automation*, IEEE.s, 2015, October, pp. 277-280.

[3] M. Badreldin, A. Hussein, and A. Khamis, "A comparative study between optimization and market-based approaches to multi-robot task allocation," *Advances in Artificial Intelligence*, 2013, pp. 12.

[4] A. Hussein and A. Khamis, "Market-based approach to multi-robot task allocation," in *Proc. 2013 International Conference on Individual and Collective Behaviors in Robotics*, IEEE. 2013, pp. 69-74.

[5] J. Beneš and P. Kolář, "Automatic cooperative control of a group of mobiles," *Automatica*, vol. 14, no. 4, pp. 313-323, 1978.

[6] C. Sarkar, H. S. Paul, and A. Pal, "A scalable multi-robot task allocation algorithm," in *Proc. 2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018 May, pp. 1-9.

[7] M. Berhault, H. Huang, P. Keskinocak, S. Koenig, W. Elmaghraby, P. Griffin, and A. Kleywegt, "Robot exploration with combinatorial auctions," in *Proc. 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 2, pp. 1957-1962. IEEE. 2003, October.

[8] A. Khamis, A. Hussein, and A.Elmogy, "Multi-robot task allocation: A review of the state-of-the-art," in *Cooperative Robots and Sensor Networks 2015*. Springer, Cham. 2015, pp. 31-51.

[9] J. Schwarzrock, I. Zacarias, A. L. Bazzan, De Araujo Fernandes, R. Q., Moreira, L. H. and E. P. De Freitas, "Solving task allocation problem in multi unmanned aerial vehicles systems using swarm intelligence," *Engineering Applications of Artificial Intelligence*, 72, pp. 10-20,2018.

[10] A. Hussein and A. Khamis, "Market-based approach to multi-robot task allocation," in *Proc. 2013 International Conference on Individual and Collective Behaviors in Robotics (ICBR)*. IEEE. 2013, December, pp. 69-74

[11] A. Hussein, M. Adel, M. Bakr, O. M. Shehata, and A. Khamis, "Multi-robot task allocation for search and rescue missions," *Journal of Physics: Conference Series*,). IOP Publishing, vol. 570, no. 5, p. 052006, 2014.

[12] D. H. Lee, "Resource-based task allocation for multi-robot systems," *Robotics and Autonomous Systems*, vol. 103, pp. 151-161, 2018.

[13] A. Liekna, E. Lavendelis, and A. Grabovskis, "Experimental analysis of contract net protocol in multi-robot task allocation," *Applied Computer Systems*, vol. 13, no.1, pp. 6-14, 2012.

[14] A. Hussein, P. Marín-Plaza, F. García, and J. M. Armingol, "Hybrid optimization-based approach for multiple intelligent vehicles requests allocation," *Journal of Advanced Transportation*, 2018.

[15] D. C. Kozen, "The design and analysis of algorithms," Springer Science & Business Media. 2012.

[16] S. M. LaValle, *Planning Algorithms*, Cambridge university press, 2006.

Ali B. Bahgat is a research and teaching assistant at the Multi-Robots Systems (MRS) lab at the German University in Cairo (GUC), Egypt. His bachelor degree was in the field of medical robotics and all the related research was carried out at the GUC. He is currently pursuing his master's degree at the GUC as well. His work focuses mainly on investigating various system architectures for solving the Multi-Robot Task Allocation problem in several environments. He tries to explore and implement new algorithms based on optimization techniques and market-based mechanisms to compare the efficiency and robustness of the proposed algorithms.