Development of a task-oriented, auction-based task allocation framework for a heterogeneous multirobot system

TAN EIJYNE¹, RISHWARAJ G² and PONNAMBALAM S G³*,

¹Core Consulting PLT, 50100 Kuala Lumpur, Malaysia
²Immagine Sdn. Bhd., Jalan SS6/2, Level 3, Kelana Parkview Tower 1, 47301 Petaling Jaya, Selangor, Malaysia
³Faculty of Manufacturing and Mechatronics Engineering Technology, University Malaysia Pahang, 26600 Pekan, Pahang, Malaysia

e-mail: eijyne.tan@coreconsulting.asia; rishwaraj.gengarajoo@123rf.com; sgponnambalam@gmail.com

MS received 17 July 2019; revised 9 December 2019; accepted 3 February 2020

Abstract. A multirobot system has cooperative team of robots designed to enhance efficiency of its operations. One of the critically investigated problems of multirobot system is the multirobot task allocation (MRTA) issue. The main objective of MRTA is to assign tasks to the most suitable robot based on its functions and capability as well as availability. In this paper, a task-oriented, auction-based task allocation framework is presented and tested through simulations and real-world experiments. The developed framework consists of a novel heuristic-based task allocation algorithm and communication module. It is implemented in a multirobot system, allowing tasks to be dynamically assigned to the robots as they achieve given tasks. The implemented framework shows robustness in its flexibility to the task and environment requirements such as resource and energy requirements and size of the environment. The framework involved a task allocation algorithm, which consists of bid generation and bid selection process, and a TCP/IP-based client-server communication module. The results from both simulations and real-world experiments matched, producing optimum results in task allocation.

Keywords. Multi-robot systems; task allocation; auction-based system.

1. Introduction

Since the advancement of robotics in the modern society with robots’ reliability and high work quality [1, 2], robots are being used together as a system, providing advantage compared to a single robot [3]. Multirobot system (also known as MRS) resembles a cooperative task force, enhancing efficiency in operations such as search and rescue [4], automation and manufacturing [5], environmental monitoring [6], and even in healthcare facilities to assist patients [7]. To further improve efficiency, heterogeneous multirobot systems are introduced, allowing various tasks to be executed by a single MRS.

One of the main studies in heterogeneous multirobot system is multirobot task allocation (MRTA). The main objective in MRTA is to assign tasks to the most suitable robot. Some of the main components, which are highly dependent on the outcome, considered as one of the problems in MRTA are task requirements and environment parameters. With this, a task-oriented, auction-based task allocation algorithm is developed, tested and implemented to assign tasks to the most capable robot in a heterogeneous multirobot system depending on components suiting the robots’ and its application and environment. This provides an opportunity for easy adaptability to different environment, enabling the assignment of task to the best capable robot in the multirobot system.

A typical study on MRTA is highly and in some cases solely focuses on the efficiency of the task allocation algorithm and is usually implemented in virtual environments only. This paper present further expands to the study in MRTA framework with the development of a novel heuristic-based, task-oriented auction algorithm and a communication module framework. Furthermore, the paper also highlights the implementation of the developed framework which is replicated in an actual physical environment for real-life application.

2. Related work

Algorithms in MRTA problems can be modelled with criteria from three different areas, which are task model, solution model, and magnitudes for cost functions [8]. Besides that, the relation of tasks to robots can also be
classified based on their interdependencies [9]. In the recent MRTA studies between 2013 and 2017, different algorithms were introduced and implemented in [10–19].

Local distributed task allocation (LDTA) was introduced in [10] where if allocation is not satisfactory, the tasks are adjusted until reaching a prescribed proportion. Repeated greedy auction algorithm was introduced in [11] where tasks can be presented dynamically in groups and each robot can do at most one task in each group. Self-organizing map approach was applied in [12] by using the Euclidian distance to assign target to the winning node. A resource-based algorithm using Multiple-Choice Hungarian method introduced in [13] includes interrelationship between task costs and resource-based interactions among robots when shared resources occur. Distributed Hungarian method with linear objective functions is introduced in [14].

Human-like behaviour-based algorithm was introduced in [15], where robots have a set of behaviours that take control the task of other robots if needed. Ant colony optimization algorithm implemented in [16] involved a cooperative colony where reward or penalty signal was given for favorable or unfavorable response. A heuristic based task allocation algorithm was introduced in [17] which focuses on minimizing transmission time and task execution time. Heuristic approach with genetic algorithm was implemented for task allocation as seen in [18]. Other methods involved clustering such as Sandholm algorithm with K-means clustering was adapted in [19]. A signal propagation model was used in [20] involving robot decision making based on signal emission and its propagation. Distributed artificial intelligence was also incorporated for realistic decision making in [21]. An improved particle swarm optimization method was developed in [22] with better computation time for task allocation. Growing cost of task is another aspect that should be considered too as noted in [23] where a binary max-sum optimization technique is designed to address this issue. The minimization of time demand of MRS is also focused in [24] with the usage of bacterial memetic algorithm to overcome this problem.

2.1 MRTA base methods

It has been observed that the main approaches for MRTA consisted of auction, Hungarian method or the Greedy method. Based on the comparative study in [25], the auction method is more favoured compared to Hungarian and Greedy method as seen in figure 1 in terms of overall cost and time.

2.2 Auction-based methods

With auction algorithm as base algorithm, it can be seen in [25–37] that the algorithm can be combined with other factors to improve overall performance and to suit different applications.

Centralized auction-based task allocation was proposed in [26], resulting in balanced task distribution using coalition trees. Decentralized auction-based algorithm in [27] is more efficient in task throughput, communication overhead and energy efficiency. Decentralized, auction and consensus-based task allocation was presented in [28] where best solution is found through fulfilling constraints with increased computational cost, however preserving polynomial nature. Distributed auction-based task allocation in [29] can maximize robot payoff using modified payoff function based on task price. Another distributed auction algorithm in [30] involves each robot bid for a task, producing near optimal solutions. Resource-oriented decentralized auction method in [31] includes resource in bidding to maximize task completion ratio. Probability based bid generation in [32] gives a higher priority in resource availability to complete tasks.

Parallel auction was introduced in [25], involving cost change with respect to auction time with logarithmic-based incremental function, resulting in optimized time and costs. A simultaneous, descending auction-based approach was introduced in [33], where new task can distract other robots from current tasks and currently executed task can be replaced with other robots.

Reconfigurable system can also be carried out, involving additional bids which are merge bid and split bid [15] depending on which is more profitable. Combinatorial auction method was introduced in [34] where evaluation of the best robot was made, improving utilization rate of the robots. Besides that, priority-based assignment was implemented in [35] where tasks are allocated using auction-based and neighbor heuristic approach, increasing average makespan linearly.

Target hunting approach was introduced in [36] where with strengthened learning, robots have self-interest and mutual trust. Multi-objective optimization (MOO) with genetic algorithm and Pareto optimality was implemented in [37], involving minimizing total completion time and maximizing energy level and task priorities. Clustering approaches such as the stochastic clustering auctions (SCA) were introduced to improve global cost of task allocation in [38], involving stochastic transfer between different task clusters, based on heterogeneity, allowing trade-off between optimality and computational requirements. Artificial capital market approach was also introduced in [39] for task allocation with robots obtaining benefits from investment.

3. Problem statement

With research on related works, it can be concluded that the feasible method for MRTA is auction method due to its time efficiency compared to other methods. Throughout the studies conducted with auction-based method, in majority of the journals studies, the authors reported their findings
on the algorithms’ testing and performance in a simulated environment. This shows a lack of real experiment implementation of their developed algorithm with robots and their communication in a physical environment.

Therefore, the main objectives presented in this paper are:

- Development of a task allocation framework.
- Development of a communication module.
- Implementation and testing onto a physical heterogeneous multirobot system.

The following sections in this paper are organized as follows. Section 4 details the development of the framework involving algorithm and communication module. Section 5 details the testing of the developed framework through simulations, including two simulation modules and summary. Section 6 details the experiments of the developed framework implemented in a multirobot system.

4. Model development

The model development involved the algorithm itself and communication module. Algorithm development consisted of the task allocation algorithm which determined the best robot for the required task, whereas the communication module consisted of communication between the robots before, during and after task execution. Figure 2 depicts the system overview of the framework and figure 3 shows the network architecture detailing the framework’s components with functionality and behaviour.

4.1 Algorithm development

Auction-based task allocation algorithm is chosen as the base algorithm to be implemented. The structure involved

![Figure 1. Comparative performance between Greedy, Hungarian, and Auction algorithm in terms of overall cost and overall time.](image1)

![Figure 2. System overview where clients involved users and the multirobot system, server for task allocation and client communication.](image2)

![Figure 3. Network architecture of the developed multirobot task allocation framework.](image3)
agents providing their own cost function through bid generation which indicates their ability to accomplish the task, to a common auctioneer. The best bidder is selected, and the auctioneer assigns the bid interest to the winner. Figure 4 shows the flow of the basic auction process.

A novel heuristic-based task-oriented framework model is developed based on the auction-based algorithm. By including parameters required for the task and its environment, the algorithm is task-oriented. The auction algorithm involved bidding by the robots after task initialization. The developed algorithm involved bid generation and bid selection, where bid selection involves bid and parameter comparison.

4.1a Bid generation

In bid generation, a model equation is developed, to accept inputs of each robot and task and environment parameters and generate output as bid for each robot for bid selection. The input parameters (RED) in the algorithm development were: resource, energy, and distance. RED parameters involved actual parameters based on the robot and estimated parameters that are required for the task and environment. With this, three different equations, equation (1), equation (2) and equations (3), depicting Model A, Model B and Model C respectively, are made and tested before selecting the best equation that produced the expected outcome for bid selection. The variables involved are summarized in table 1 and figure 5 shows how the distance parameters are obtained.

\[
\text{Bid} = R \times \left\{ \left[ \frac{1}{1 + (e_{\text{act}} - e_{\text{est}})} \times e_{\text{fac}} \right] \times \left[ \frac{1}{1 + (d_{\text{est}} - d_{\text{act}})} \times d_{\text{fac}} \right] \right\}
\]

(1)

\[
\text{Bid} = R \times \left\{ \left[ \frac{1}{1 + (e_{\text{act}} - e_{\text{est}})} \times e_{\text{fac}} \right] \div \left[ \frac{1}{1 + (d_{\text{est}} - d_{\text{act}})} \times d_{\text{fac}} \right] \right\}
\]

(2)

\[
\text{Bid} = R \times \left\{ \left[ (e_{\text{act}} - e_{\text{est}}) \times e_{\text{fac}} \right] \times \left[ \frac{1}{1 + (d_{\text{est}} - d_{\text{act}})} \times d_{\text{fac}} \right] \right\}
\]

(3)

The focus in developing the model equations seen in (1)–(3) is to obtain the variation between actual and estimated parameters by quantifying them between 0 to 1. The purpose of introducing task parameter importance factor as multiplier to the difference between actual and estimated variables is to affect the overall bid value for bid selection. The factor distinguishes a robot’s bid from other robots’ if distance/energy is to be of more importance than energy/distance for the task.

4.1b Bid selection

With every participating robot in the auction generating a bid value, bid selection process is done through bid comparison and parameter comparison.

4.1c Bid comparison

To determine the better model equation, graph and value-based simulations are made to observe the outcome for each model equations. For graph-based simulation, actual

\[\text{Figure 4. Basic auction flowchart involving bid generation and bid selection.}\]

\[\text{Figure 5. Illustration for obtaining } d_{\text{act}} \text{ and } d_{\text{est}}.\]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>availability of resources (i.e. 1 if available else 0)</td>
</tr>
<tr>
<td>d_{fac}</td>
<td>distance as task parameter importance factor</td>
</tr>
<tr>
<td>e_{fac}</td>
<td>energy as task parameter importance factor</td>
</tr>
<tr>
<td>e_{act}</td>
<td>actual energy (battery percentage) of the robot</td>
</tr>
<tr>
<td>e_{est}</td>
<td>estimated energy (battery percentage) required for the task</td>
</tr>
<tr>
<td>d_{act}</td>
<td>actual distance of robot current location to the task goal</td>
</tr>
<tr>
<td>d_{est}</td>
<td>estimated distance for the task which is the maximum possible distance of end of map to task goal (+/- 5 depending on task type)</td>
</tr>
</tbody>
</table>
parameters for the input range of the model equations are set as follows: $e_{\text{act}}$ is set from 30% to 100% with a step size of 5% and $d_{\text{act}}$ is set from 4 to 50 with a step size of 2. The estimated parameters, $d_{\text{est}}$ and $e_{\text{est}}$, are both set to 30.

With x-axis indicating number of iterations resulted on the fixed input range and y-axis indicating output bid values from the model equation developed, the trend of the output graphs for Model A and Model B in figure 6 are quite similar due to the division property of the energy term in the equation when denominator increases. For output graph for Model C, the amplitude increases at a constant defined rate of 0.8, showing higher consistency compared to the other two models. The fluctuations of all graph from positive to negative is due to the introduction of negative term when $d_{\text{act}} > d_{\text{est}}$.

Next, to further prove the consistency of the model equations, a more detailed simulation is done by generating a set of values. Three sets of distance and energy parameters are grouped together, indicating parameters for three robots for a single round of bidding. Five possible combinations of the robots’ actual parameters can be arranged as follows:

- Setting $d_{\text{act}}$ constant while increasing $e_{\text{act}}$
- Setting $e_{\text{act}}$ constant while increasing $d_{\text{act}}$
- Decreasing $e_{\text{act}}$ and decreasing $d_{\text{act}}$
- Decreasing $e_{\text{act}}$ and increasing $d_{\text{act}}$
- Increasing $e_{\text{act}}$ and decreasing $d_{\text{act}}$

With the possible combinations, the bid selection is first made to select the lowest bid while performing bid comparison and the agent corresponding to the lowest cost will be assigned to the task. Several other factors to be considered include when all bids are positive (i.e., $d_{\text{act}} < d_{\text{est}}$), all bids are negative (i.e., $d_{\text{act}} > d_{\text{est}}$), and bid for either one of the agent is either positive or negative (i.e., mix of positive and negative bids). With each row representing each robot’s values on actual energy, actual distance and bid value, figures 7 and 8 show portions of the results that highlight the difference between Models A, B and C for the same set of estimated parameters.

From figures 7 and 8, it is shown that a general pattern could not be found for both Model A and Model B, resulting in inconsistent results. Whereas for Model C, a general, consistent pattern of the outcome is found and conditions could be applied. From test simulation in both graphical and value form, it is concluded that the results of Model C are consistent as it shows a general pattern for all five possible combination patterns as indicated in the graph of figure 6 and value simulation results in figures 7 and 8. Therefore, Model C is used for bid generation and bid selection in the developed task allocation algorithm.

By arranging robot’s energy parameter in descending order and finding the overall general pattern of the robot’s distance parameter, bid decision rule table, shown in table 2, is developed for bid comparison to determine whether the selection is maximum or minimum bid. The bid decision rule table is designed for when $e_{\text{fac}} > d_{\text{fac}}$ and the logic is reversed, i.e., from minimum to maximum, when $d_{\text{fac}} > e_{\text{fac}}$ instead. The conditions for it is designed such that for example, if $e_{\text{fac}} > d_{\text{fac}}$, whichever robot that has the highest energy will be selected for the task, and if $e_{\text{fac}} < d_{\text{fac}}$, whichever robot that takes the shortest path to the target goal will be selected for the task. This is not always the case; thus, parameter comparison is introduced.

### 4.1d Parameter comparison

Another factor considered is the energy-distance relationship. To address this, parameter comparison is developed. In this process, the winning bid’s distance and energy parameters are compared with other robots’ to obtain a more optimum robot based on the system environment. To increase environment compatibility, short distance factor, long distance factor and large energy factor are introduced, where they can be modified based on its application and environment.

- Short distance factor: Distance that is considered relatively short in the environment which do not have a huge impact on the energy parameter.
- Long distance factor: Distance that is considered relatively far in the environment, making a larger impact on the energy usage.
- Large energy factor: Energy that is considered large enough to affect the bid comparison process in selecting the other robot as the winner of the task if the other robot has the factor of energy larger than the current winning robot’s energy.

With the three factors, three important relationships are highlighted:

- Although a robot has a slightly larger distance than winning bid’s distance by short distance factor, the robot will be assigned as the new winner if its energy is still lesser than the former winning robot’s energy.
- If a robot’s energy is higher than the winning robot’s energy by the large energy factor, the robot will be assigned as the new winner if its distance may be larger or smaller than the winning robot’s distance by the short distance factor.
- If a robot’s distance is larger than the winning robot’s distance by the large distance factor, the robot will not win the bid.

The entire bid selection process can be summarized in figure 9 and the overall developed task-oriented, auction-based task allocation algorithm is shown in table 3.

### 4.2 Communication module development

In general, MRTA does not focus on communication module but this study considers the importance of it as well to improve efficiency by enabling robots in MRS to
Figure 6. Graph-based simulation results for (a) Model A, (b) Model B and (c) Model C.

Figure 7. Value simulation highlighted results for (a) Model A: shows increase (1st set) and decrease (2nd set) in bid value [not consistent] and (b) Model B: shows decrease (1st set) and increase (2nd set) in bid value [not consistent].

Figure 8. Value simulation highlighted results for Model C (a) shows increase (1st & 2nd set) bid in comparison to Model A result, (b) shows decrease (1st set) and increase (2nd set) in bid value, but gives the right result after applying conditions for bid < 1 and bid > 1.
perform frequent updates on its status before, during, and after task execution. The communication module for the task allocation framework comprises of robots, and users, who initiate the tasks, as clients and a server that connects each client with the task allocation algorithm. The communication module developed to link the task allocation algorithm to a multirobot system is based on TCP/IP client-server protocol. This protocol is chosen as it is application-oriented instead of development-oriented where lost packets during transmission would not be of concern when developing the communication module. The communication module on client and server can be summarized in the pseudocodes presented in tables 4 and 5.

Table 2. Bid Decision Rule Table.

<table>
<thead>
<tr>
<th>Possible Combinations</th>
<th>Positive Bid Rule</th>
<th>Negative Bid Rule</th>
<th>Mix of positive and negative bids</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$&lt; 1$</td>
<td>$&gt; 1$</td>
<td>$0 &lt; x &lt; 1$</td>
</tr>
<tr>
<td>e↓, d↓</td>
<td>max</td>
<td>min</td>
<td>max</td>
</tr>
<tr>
<td>e↓, d↑</td>
<td>max</td>
<td>max</td>
<td>min</td>
</tr>
</tbody>
</table>

**Figure 9. Summary of bid selection process.**

Table 3. Algorithm Pseudocode.

**Algorithm pseudocode**

while task not assigned do
  Check importance (d/e) and assign $e_w$ and $d_w$ accordingly
  Accept and store parameters from i no. of robots
  for $i ∈$ number of agents do
    parameterArr[i][0] = $e_w[i]$; parameterArr[i][1] = $d_w[i]$
  end for
  (i) Sort with the energy parameter in decreasing order:
    parameterArr = qSort(parameterArr, number of agents, sizeof(parameterArr))
  for $i ∈$ number of agents do
    Generate bid for an agent:
    bid[i] = findBid(parameterArr[i][1], parameterArr[i][0])
  end for
  (ii) Finding the general distance parameters:
    findPattern (parameterArr)
  (iii) Winner for the task is determined through bid selection:
    winnerAgent = winnerDet(bid[i] ... bid[i + n])
end while

Table 4. Algorithm Pseudocode (Client).

**Algorithm pseudocode (client)**

Initialize position
Connect to server
while system is on do
  Get robot’s status (location, resource and energy parameters)
  Send robot’s status to server
  if task received do
    while task not completed do
      Execute task
      Send robot’s status to server
    end while
  end if
end while

Table 5. Algorithm Pseudocode (Server).

**Algorithm pseudocode (server)**

Start server
while server is on do
  Receive robots’ status (location, resource and energy parameters)
  Update map representation record of positions of all robots in system
  if task received do
    Execute path planning for path step for each robot to target
    Execute task allocation to select best robot for task
    Notify winning robot to execute task
  end if
end while

(a) Two bidding robots

(b) Four bidding robots

Figure 10. Summary of Simulation Module 1.1 results for (a) and (b).
4.2a Client-server structure

The client communication for robots is developed in C programming language. They are connected to the server using TCP sockets connecting to the same wireless network as the server. The server communication is developed in Python programming language and implemented based on Twisted Matrix, an open source event-driven network communication programming framework licensed under MIT, which supported TCP/IP protocol with multithreading capabilities.

The communication flow between client and server comprises of 5 main communication flows and parameters involved in each flow in the task allocation framework consisted of the following:

- Robot to Server Communication: Location and resource update when a task is completed.
- Server to Robot Communication: Task parameters to robot that was awarded with task such as, task number and string of path coordinates to target.
- User to Server Communication: Presence of new tasks, including task parameters such as task number and its location and requirement with task importance factor.
• Server to Task Allocation Algorithm Communication:
Robots’ bidding information during the presence of new task.
• Task Allocation Algorithm to Server Communication:
Robot name and task number.

5. Model testing

Model testing is performed to test the algorithm by observing the outcome of random inputs from each robot and task. The simulation starts with generating random estimated parameters for a task, initiating the task. After setting the number of robots present in the MRS, the robots start the auction process by obtaining the actual parameters, in this simulation case the actual parameters are generated at random. The bid values of each robot are determined and based on the best bid from the algorithm, the robot corresponding to the best bid is selected as the winner for the auction, thus obtaining the task.

The test simulation consists of two simulation modules. The first module involves distance parameter as the importance factor, where $d_{\text{fac}}$ is set to be 0.8 and $e_{\text{fac}}$ is set to be 0.2. The second module involves energy parameter as the importance factor, where $e_{\text{fac}}$ is set to be 0.8 and $d_{\text{fac}}$ is set to be 0.2. For both simulation modules, short distance factor is set as 2, long distance factor is set as 3 and large energy factor is set as 5. Both sets show the effect of the different priority task to the algorithm on different number of robots. The expected outcome of all the simulation tests would be the consistency in obtaining the best robot for a certain task.

In both simulation modules with all generated robot parameters and estimated parameters at random required for task, each module involved 50 sets of results each for 2, 3, 4, and 5 bidding robots. The following subsections are based on the simulation modules where results are discussed. The results in the subsections highlight the key features of the outcome on the best robot for a task.

5.1 Simulation module 1: distance

In Simulation Module 1, distance is set as the task parameter importance factor. Based on the simulation results with different number of bidding robots, it is shown in figure 10 that regardless of the number of robots, the
robot with the shortest distance to the target is the best robot as it can reach the task target faster and therefore wins the task for tasks of distance importance. If there are multiple robots with same distance to target, the robot with the higher energy wins the task for tasks of distance importance, which is typically a better choice compared to robots with lower energy, as shown in figure 11.

5.2 Simulation module 2: energy

In Simulation Module 2, energy is set as the task parameter importance factor. Based on the simulation results with different number of bidding robots, it is shown in figure 12 that regardless of the number of robots, the robot with the higher energy wins the task for tasks of energy importance, which fulfills the purpose of the energy importance factor.

However, for tasks of energy importance, the robot’s energy relates to the target distance as well and it is not necessary for robot that has the highest energy winning the task all the time. As shown in figure 13, the robot with a much more shorter distance difference, by long distance factor of 3, and even though lower energy can win the task for tasks of energy importance. This is because larger travelling distance decreases energy. Robot with shorter distance and even though slightly lower energy, by large energy factor of 5, can win the task for tasks of energy importance as well. This is because the energy difference is not significant compared to the distance to target among the robots.

5.3 Simulation summary

With 50 sets of simulation results for each simulation module, the consistency of the developed algorithm in obtaining the best robot for a task can be summarized through percentage consistency in figure 14, where percentage consistency is calculated through (4):

\[
\text{Percentage consistency} = \frac{\text{No. of Simulations with Expected Result}}{\text{Total No. of Simulations}} \times 100\%
\]

Based on figure 14, Simulation Module 2 has 100% percentage consistency and Simulation Module 1 has tolerable percentage consistency of 92% and above. This is due to the extension of the definition of tasks of distance importance where robots with higher energy and although slightly further or has lesser distance difference can win the task as well.

Consistency as performance comparison is only done internally based on the number of robots in the system.

<table>
<thead>
<tr>
<th>Task No.</th>
<th>Experiment 1 Reasoning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 &amp; 2</td>
<td>Robot closest to the task target won.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task No.</th>
<th>Experiment 2 Reasoning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Robot closest to the task target won.</td>
</tr>
<tr>
<td>2</td>
<td>Robot closest to the task target won.</td>
</tr>
<tr>
<td>3</td>
<td>Robot that was not occupied with task was eligible for bidding and won.</td>
</tr>
<tr>
<td>4</td>
<td>Robot with higher energy won the task though same distance to target with another robot.</td>
</tr>
<tr>
<td>5</td>
<td>Robot with lower energy and shorter distance than large distance factor won.</td>
</tr>
<tr>
<td>6</td>
<td>Robot that was not occupied with task was eligible for bidding and won.</td>
</tr>
<tr>
<td>7</td>
<td>Robot closest to the task target won.</td>
</tr>
</tbody>
</table>

Figure 17. Process of one of the Experiment 1 from (a) start to (d) end. (a) Moving to target (Task 1 and 2 in progress). (b) Reached target (Task 1 and 2 in progress). (c) Performing task (Task 1 and 2 in progress). (d) Tasks completed (Task 1 and 2 in progress).
not externally with other task allocation algorithms. This is due to the paper’s focus where task allocation can be studied not just the algorithm alone, but also including the communication framework for real live experiment testing. Therefore, the focus involved the overall framework which does not have precedent and hence no comparison is made.

6. Model experiment

Experiments are performed to implement the developed framework in a heterogeneous multirobot system comprising of 4 to 5 Khepera IV robots with different sensing capabilities and additional grippers, on a 2D 12 by 6 grid world structure. Three types of tasks are used in the

---

Figure 18. Process of Experiment 2 from (a) start to (j) end.
experiments shown in figure 15. Box pushing task involved pushing a box two steps away from box original location, whereas pick and place task involved retrieving an object and placing it at new location. Image capture task involved capturing image in the 4-cardinal directions of the target location.

The model experiment is divided into two sets. Experiment 1 is made to solely show the implementation of task allocation algorithm in a controlled environment to show the framework operations and key features of the algorithm. Experiment 2 is made to show a more practical application of the developed framework, with other factors such as obstacles in the system. The common flow for all the experiment resembles the network architecture that is developed for the task allocation framework. Figure 16 shows ongoing real live experiments on different environments.

Task list with reasoning on best robot selected for the task in Experiment 1 and Experiment 2 are summarized in table 6. Figures 17 and 18 show summary of physical process of the multirobot system when tasks are being initiated for both Experiment 1 and Experiment 2. Both experiments strive to show the outcomes for tasks of distance importance and tasks of energy importance.

7. Conclusion

The development of a task allocation framework for a heterogeneous multirobot system is successfully done through a task-oriented framework model developed based on the auction-based task allocation algorithm. The algorithm involved bid generation using a tested model equation and bid selection process through comparison of bid and parameter. The framework also includes the communication framework which allows task allocation to be more efficient by enabling robots in MRS to perform frequent updates on its status before, during, and after task execution. The framework is implemented in both simulations and real live experiments in a physical heterogeneous multirobot system. The results from both simulations and real live experiments matched, producing optimum results in task allocation. Further improvement can be made as future work, by implementing a more automated task initiation with lesser parameters using task database for task initiation which relates to the system environment.

Acknowledgements

This research is funded by University Malaysia Pahang research grant (Project ID: RDU1803139).

References


