

Article

Dynamic Multi-Objective Auction-Based (DYMO-Auction) Task Allocation

Uthman Baroudi ¹, Mohammad Alshaboti ¹, Anis Koubaa ^{2,3,*} and Sahar Trigui ⁴

¹ Computer Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia; ubaroudi@kfupm.edu.sa (U.B.); alshaboti.it@gmail.com (M.A.)

² Computer Science, Prince Sultan University, Riyadh 12345, Saudi Arabia

³ CISTER, INESC-TEC, ISEP, Polytechnic Institute of Porto, 4249-015 Porto, Portugal

⁴ National School of Computer Science, University Campus of Manouba, Manouba 2010, Tunisia; sahar.trigui@coins-lab.org

* Correspondence: akoubaa@psu.edu.sa

Received: 15 April 2020; Accepted: 28 April 2020; Published: 8 May 2020



Abstract: In this paper, we address the problem of online dynamic multi-robot task allocation (MRTA) problem. In the existing literature, several works investigated this problem as a multi-objective optimization (MOO) problem and proposed different approaches to solve it including heuristic methods. Existing works attempted to find Pareto-optimal solutions to the MOO problem. However, to the best of authors' knowledge, none of the existing works used the task quality as an objective to optimize. In this paper, we address this gap, and we propose a new method, distributed multi-objective task allocation approach (DYMO-Auction), that considers tasks' quality requirement, along with travel distance and load balancing. A robot is capable of performing the same task with different levels of perfection, and a task needs to be performed with a level of perfection. We call this level of perfection quality level. We designed a new utility function to consider four competing metrics, namely the cost, energy, distance, type of tasks. It assigns the tasks dynamically as they emerge without global information and selects the auctioneer randomly for each new task to avoid the single point of failure. Extensive simulation experiments using a 3D Webots simulator are conducted to evaluate the performance of the proposed DYMO-Auction. DYMO-Auction is compared with the sequential single-item approach (SSI), which requires global information and offline calculations, and with Fuzzy Logic Multiple Traveling Salesman Problem (FL-MTSP) approach. The results demonstrate a proper matching with SSI in terms of quality satisfaction and load balancing. However, DYMO-Auction demands 20% more travel distance. We experimented with DYMO-Auction using real Turtlebot2 robots. The results of simulation experiments and prototype experiments follow the same trend. This demonstrates the usefulness and practicality of the proposed method in real-world scenarios.

Keywords: multi-robot system (MRS); multi-robot task allocation (MRTA); auction-based task allocation; task quality; online decision making

1. Introduction

Robotic networks get more and more proliferation in industrial and scientific applications. Robotics network attracts researchers' attention because of its computational, sensing, communications, and movement capabilities. Robotics network is suitable for a wide range of applications where human intervention is limited or denied, such as search and rescue operations, surveillance, logistic, and humanitarian demining. Besides, such networks can lend themselves easily to applications where there are economic benefits such as farming or production line applications. The advantages of

using the robotics network including, but are not limited to, the flexibility of modifying the robotics network to match different application scenarios, the robustness of multi-robot systems against failure, and parallelism operation, which leads to time-efficient system [1].

A team of robots can map a specific area and identify unsafe zones. It can allocate tasks either by itself if the robots are equipped with appropriate sensors or with the help of an external system, such as wireless sensor networks (WSN). The task can be carrying goods in a warehouse application, cleaning in a cleaning operation scenario, or finding victims in a surveillance application. In many situations, task locations and requirements are not known beforehand because they emerge as needed in the area. Hence, offline task assignment approaches are not a feasible solution. Therefore, an on-line distributed task allocation approach is required for such dynamic scenarios.

Several works have investigated this problem as a multi-objective optimization (MOO) problem and proposed different approaches to solve it including heuristic methods [2–8]. The existing MRTA approaches can be classified into three major categories: centralized approaches, market-based approaches, and behavior-based approaches. Centralized methods are suitable for a small number of robots in a static environment. They suffer from a single point of failure and high communication overhead, and they respond slowly to local changes [9]. In contrast, fully distributed approaches, such as behavior-based approaches (e.g., [10,11]), are robust to failures, flexible, and require less computational and communication resources. However, they work on local optimal solutions, which do not necessarily aggregate to produce the globally optimal solution, and thus, they yield suboptimal solutions. Other approaches also include knowledge-based coordination where a set of rules are embedded into the agent that determines its behavior [12].

On the other hand, the market-based/auction-based approach, one of the most popular algorithms of MRTA, is considered to be the mid-point between fully centralized and fully distributed approaches. It works similarly as an auction process in the market, where an auctioneer opens an auction, and then bidders submit their offers, with the auctioneer granting the item to the highest bidder. In the market-based approach, the bid is computed by each robot as a function of its utility in performing a task. There are primarily two common types of auctions: single-task auctions and combinatorial auctions. In a single-task auction, there is one task, and it is granted to the highest bidder. In a combinatorial auction, multiple tasks are offered in one auction, and bidders can bid on any subset of the offered tasks based on the robots' decision.

The problem of the single-task approach is that it does not consider the synergy among tasks, which leads to a suboptimal solution. Synergy is a term used to describe the relationship between tasks in terms of their locations [9]. Tasks have positive synergy if the total cost of execution by one robot is less than the total cost if they are executed by more than one robot. In contrast, because the combinatorial auction allows robots to bid on a subset of tasks, robots can bid on tasks that have positive synergy, and thus, it produces a better solution. However, the combinatorial auction requires tremendous computational resources because the number of possible bundles increases exponentially with the number of tasks [13]. Sequential single-item (SSI) auctions [14] are an alternative to combinatorial auctions. In SSI auctions, robots bid on all unallocated tasks by the minimum cost increase in the sum of the minimum path to visit the assigned tasks. The robot with the smallest bid wins a task, and then, the bidding process begins again for the remaining unallocated tasks. Then robots compute the minimum path to their tasks and move accordingly [15].

This paper targets the applications in which robots are capable of performing the same task with different levels of accuracy/perfection (we call it quality) and tasks emerge with desired quality to be performed. Besides, the quality requirement system will try to balance the load distribution among robots and minimize the total travel distance. Examples of such systems are given in detail in Section 3.2.

The contributions of this work are summarized as follows. First, we present an on-line distributed multi-objective task allocation approach (DYMO-Auction), which assigns the tasks emerging from a dynamic scenario. DYMO-Auction is distributed in the sense that there is no pre-defined auctioneer

for all tasks. A robot that receives a task announcement is the auctioneer for this task, and the next task announcement may have different auctioneer. The task assignment process takes into consideration the task's quality requirement as well as load balancing and travel distance. The important novelty of this work is that we define a utility function to consider the four competing metrics, namely the cost, energy, distance, type of tasks. To the best of our knowledge, the considered utility metric was not addressed in other prior works.

Second, we mathematically formulate the concept of quality in the allocation process (to the best of the authors' knowledge; this is the first work that considers quality in this context), see Section 3.3. Third, we provide a comprehensive simulation study of the proposed DYMO-Auction approach and benchmark with the offline task allocation approach (i.e., sequential single-item auction SSI). Finally, we validate the simulation results through real-world experimentation and deployment on Turtlebot robots.

The remainder of the paper is structured as follows. Section 2 presents an overview of the literature for distributed multi-objective task allocation. In Section 3, we provide a detailed description of the proposed approach, including the target system model, real-world application examples, task's quality representation, the proposed auction process, and the cost function. We define the performance evaluation metrics in Section 4. Simulation setup and results are presented in Sections 5 and 6, respectively. Then, we validate the DYMO-Auction approach using real-world experiments in Section 8. Finally, we conclude this paper with the main findings and future research directions in Section 9.

2. Related Works

This section provides the most relevant works that have been studied considering distributed task allocation in multi-robot networks. We focus on the factors used for the allocation process and the scenarios, whether static or dynamic, and we provide a summary table that compares the differences between DYMO and prior works.

The most common factor used in assessing a task allocation process is the traveled distance. Hence it can be formulated as the traveling salesman problem [16]. Other works focused on resource consumption besides the traveled distance factor, such as [2], which proposed a distributed resource-oriented auction algorithm that considers the resources that a robot consumes while it executes the task. The authors argue that not including this factor can affect the task execution because the winning robot may run out of resources while executing the task. Mi et al. [3] suggested integrating mobile sensors and robots to perform the assigned tasks. Mobile sensors are responsible for locating and identifying the tasks, whereas robots are responsible for performing the tasks. When a sensor detects a task, it sends a request to the robots; the closest robot will be the coordinator for that task and will forward the request with a time limit to other robots. Each robot that has a vacancy will participate, computing its utility based on cost, energy, distance, type of task and sending a reply. Then, the coordinator will assign the task to the robot with the highest utility. Sheng et al. [4] used a bidding model to select an appropriate robot to discover the unknown area. They included a nearness measure (as a communication link measure) in the utility function, which computes the distance between the current robot and its neighbors. The use of the nearness measure keeps robots close to each other and reduces the total travel distance and discovery time. Elango et al. [5] proposed workload balancing by decomposing tasks into clusters based on the total travel distance in each cluster and the distance between tasks in that cluster. Then, the auction-based approach is used to assign robots to the task clusters. However, for task clustering, each robot must compute its cost for executing all combinations of tasks, which is a complex process, mainly if there are a large number of tasks. Gong et al. [6] proposed a combinatorial auction model based on a genetic algorithm (GACA). The GA searches through all combinations of tasks and robots to obtain the best solution based on distance. A hunting task is given as a mission for the team of robots. They compared combinatorial auction versus single-item auction and found that in such scenarios, combinatorial auction outperforms single-item auction in terms of time. However, it requires high computation compare to the single item

auction. Gerkey and Mataric [7] proposed MURDOCH, a fault-tolerant distributed protocol based on the auction method for multi-robot coordination. The central concept is that the winner obtains the task with a contract to finish it within a time window. The Auctioneer is responsible for monitoring the progress of the task; if it discovers a failure or insufficient progress, it can terminate the contract and announce this task in a new auction process. Similar failure recovery mechanisms, re-assigning tasks based on environmental changes, also presented in [17]. Some work has also been performed in multi-objective task allocation, such as the work by [8]. They used an auction-based approach to solve multiple objectives: the remaining energy after task execution, the total time for completion of the task, and the priority of the task. A team of robots is divided into various clusters, each of which consists of a coordinator besides other robots. The coordinator is responsible for arranging the auction and distributing task information. Robots are heterogeneous, and for each task, some are eligible to participate in the auction based on the task requirements. A summary of the related works and proposed approaches are provided in Table 1 [18].

Table 1. Summary of the related works and the proposed method in terms of distance (D), load (L), quality (Q), dynamicity (D), synergy (SY) objectives [18].

Multi-Objective					
Reference	D	L	Q	DY	SY
[2]	Yes	No	No	Yes	No
[3]	Yes	No	No	Yes	No
[4]	Yes	No	No	No	No
[5]	Yes	Yes	No	No	Yes
[6]	Yes	No	No	No	No
[7]	Yes	No	No	Yes	No
[8]	Yes	No	No	Yes	No
Proposed approach	Yes	Yes	Yes	Yes	Yes

Motivated by the lack of methods that can be applied to applications where tasks and robots quality are needed to be considered (see Section 3.2). We proposed DYMO an auction-based task allocation approach that considers task quality requirements along with load balancing and travel distance.

In this work, we target task allocation for single-task, single-robot, multi-robot (ST-SR MR) systems applications [19]. In these applications, each robot can perform one task at a time, and each task requires only one robot. Our approach can be considered a variant of an SSI auction with two main differences. First, the proposed DYMO-Auction considers task quality requirements besides load balancing and travel distance. Second, it works online in a dynamic scenario (i.e., tasks assign whenever they emerge on the area) without any prior information about the incoming task.

3. Dynamic Multi-Objective (DYMO) Auction-Based Task Allocation

We propose quality-aware DYMO-Auction for MRTA problems in dynamic scenarios. DYMO-Auction selects an auctioneer randomly for each task to avoid a single point of failure. Initially, all robots are deployed in the area of interest, either randomly or for a purpose, as in [20]. The tasks appear in the area over time and must be assigned on time. A task may be discovered either by mobile sensors or by a static WSN [21,22]; either way, we assume that the details of a newly emerged task can reach the robotic network (i.e., can contact any robot in the network). The auctioneer (the receiver) opens an auction for the discovered task, and other robots bid on that task. Then, the auctioneer assigns the task to the robot with the minimum cost. A robot R_i bids on a task T_j using the bidding rules that are designed to maximize the quality satisfaction, distribute the workload among the available robots, and minimize the total traveled distance. The winner robot starts visiting and executing the assigned tasks A_i , and it can also participate in the upcoming auction processes.

3.1. System Model

We consider the following system characteristics:

- The robots are randomly deployed in the area of interest A .
- The tasks appear uniformly within the area of interest A .
- The number of robots m is less than the number of tasks n , i.e., $(m < n)$.
- The order of the tasks' execution is not important; thus, a robot is free to perform assigned tasks based on the travel distance, not their assignment order.
- A task T_j demands a quality level q_j for its performance. The task quality level is uniformly distributed over the entire range, from the minimum to the maximum quality level. The range of the quality level is defined based on the application requirements (e.g., three quality levels would be low, medium, and high).
- Task discovery may be performed either by a separate system, such as mobile sensors/a WSN or by the robots themselves. In both cases, we assume that only one robot receives the details of a given task randomly.
- Task's details don't change after discovery process.

3.2. Application Example

We consider a scenario of imaging objects (plants) for a disease diagnostic system as a real example in which the proposed approach can be implemented. A team of robots equipped with appropriate sensors integrated with a remote sensing system, such as that proposed in reference [23], can be used in a useful farming scenario application. This application includes two modes: remote sensing and near-range sensing. Remote sensing aims to detect and diagnose any unhealthy symptoms in an area of interest, such as diseases, weeds, and pests. If any infection is detected, the remote sensing advertises a new task for the near-range sensing robots, which are equipped with appropriate sensors, such as a camera, thermography, chlorophyll fluorescence, and hyperspectral sensors. The remote sensing set of robots provides necessary information about a task, such as its location, the number of resources required (fertilizers/pesticides), and the required quality. In such a case, a robot's quality is the quality of its camera resolution. In contrast, a task's quality requirement is the resolution of the image required (since different diseases need different image resolutions to be detected).

A cleaning process is another example in which there are different spots; some are easy to clean (low quality), whereas others are difficult (high quality). A team of robots is equipped with various tools/materials based on their cleaning capabilities.

Quality is a term used to represent the preference that a robot and task that have the same quality level be paired. In other words, we prefer a task with quality level q_j to be executed by a robot with a quality level q_i such that q_i is equal or close to q_j . It is also possible for a task to be executed by a robot with a different quality level, either higher or lower. In the first case, where a robot's quality level is higher than a task's quality level ($q_i > q_j$), some resources will be wasted from the robot's perspective (e.g., image storage size, energy consumed by a high-quality camera, transmission bandwidth required for a high-quality image), whereas from the task's perspective, its requirements have been satisfied (the image is captured with a resolution greater than what is required). For the second case, where a low-quality robot executes a higher-quality task ($q_j > q_i$) there is no waste in resources. However, the task's requirements are not completely satisfied (i.e., the image has a resolution lower than what is required). To differentiate between a task type and task quality level, we can state that there are different quality levels for the same task type.

The robots use our proposed approach to assign each farming task to a robot that attempts to achieve: satisfaction of the image quality required (the quality level), load balance and minimizes the travel distance.

3.3. Quality Settings

Assume the task’s quality requirements levels are defined by $Q_T = \{a_1, a_2, a_3, \dots, a_k\}$, from the very low quality level a_1 to the very high a_k , based on the application. To utilize robots and satisfy tasks’ quality, each task has to be assigned to a robot where the difference between robot’s quality and task’s quality is minimized. For example, if we have tasks with quality requirements from 1 to 9, and only three robots with quality 2, 5, and 8, then tasks that require quality 1, 2, or 3 should be assigned to the robot with quality 2, similarly with the other tasks, see Figure 1.

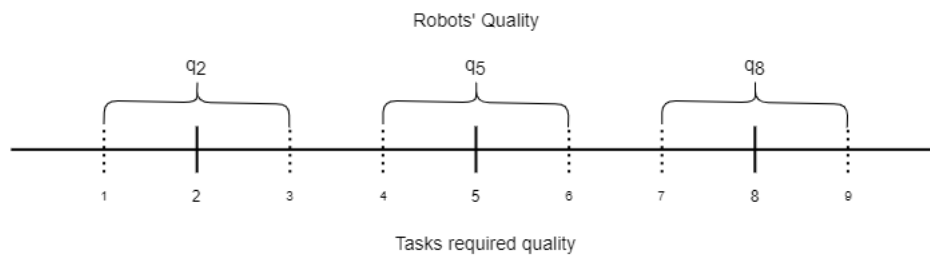


Figure 1. Ideal tasks assignment based on quality.

To generalize, assume the robots’ quality levels are given by $Q_R = q_1, q_2, q_3, \dots, q_z$; then we propose the following equation to interrelate the robots’ quality levels to the tasks’ quality levels:

$$q_i = (i - 1) * \beta + \alpha, \tag{1}$$

where β is the range of task qualities that can be fulfilled by each robot’s quality level, α is used to compute q_i , and they are computed as follows:

$$\beta = \frac{k}{z} \tag{2}$$

$$\alpha = \frac{\beta + a_1}{2}, \tag{3}$$

where k and z are the number of task quality requirement levels and the number of robot quality levels, respectively. As a proof of concept, we assume nine levels of quality requirements for the tasks ($k = 9$), represented by numbers from a_1 (very low quality) to a_9 (very high quality). Using Equation (1), the robot qualities are as follows: $Q_R = 2, 5, 9$, whereas $\beta = 9/3 = 3$ and $\alpha = (3 + 1)/2 = 2$.

3.4. Objectives

The objectives we aim to achieve in this approach are three-fold:

1. Maximize the quality satisfaction (minimize the sum of differences between robot quality and assigned task quality).

$$\min_{\forall i \in \{1,2,3,\dots,m\}} (l_i * q_i - \sum_{j=1}^n q_j),$$

where l_i is the number of tasks assigned to a robot R_i .

2. Minimize the sum of traveled distance (TTD) per robot.

$$\min_{\forall i \in \{1,2,3,\dots,m\}} TTD(R_i, l_i).$$

3. Balance the load between all robots.

$$\min_{\forall i \text{ and } j \in \{1,2,3,\dots,m\}} \sum |l_i - l_j|, i \neq j.$$

3.5. Bidding Rules

Bidding rules are the rules for robots to follow for allocating the tasks, and they reflect the objectives of DYMO approach. Assume that a robot R_i with quality level q_i is located at position p_i , and a task T_j quality level q_j is located at position p_j . The robot R_i bids on a task T_j using the following three bidding rules, which are derived from the aforementioned objectives.

3.5.1. Bid Based on Quality Satisfaction

A robot would bid with the difference between its quality level q_i and a task's quality level q_j .

$$q_{ij} = |q_i - q_j|.$$

This metric will ensure that a robot with a quality level equal or close to the task's quality level will win the task. This approach will maximize the quality satisfaction objective.

3.5.2. Bid Based on Workload

Let l_i be the number of current tasks that have been assigned to robot R_i . Then, the robot will bid with the current number of assigned tasks l_i . For instance, if robots R_1, R_2 and R_3 have the following assigned task lists: $l_1 = 3, l_2 = 3$ and $l_3 = 2$, respectively. Then, robot R_3 will win the new task because it has the minimum bid ($l_3 = 2$). This rule ensures that the robot that has been assigned a minimum number of tasks will have an opportunity to win the new task.

3.5.3. Bid Based on Distance

A robot R_i located at position p_i will bid on task T_j located at position p_j by computing the minimum distance between either its current location p_i or the position p_k of any unaccomplished task to the new task location p_j as described in Equation (4).

This rule will guarantee that the synergy between tasks is considered to minimize the traveled distance. Synergy is a term used to describe the relationship between tasks; the tasks have a positive synergy if the total cost of executing all of them by one robot is less than if a different robot performs each task. We consider the spatial synergy in this work. If the total traveled distance for visiting two tasks using one robot is less than the total traveled distance if different robots visit each task, then these tasks have positive synergy, and vice versa; see Figure 2.

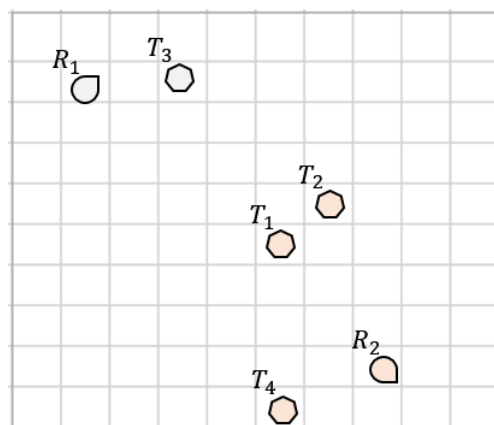


Figure 2. Synergy example: the optimal assignment is to assign T_3 to robot R_1 and T_4 to robot R_2 because they have negative synergy. Because T_1 and T_2 have positive synergy, they must be performed by one robot, R_2 .

The proposed DYMO-Auction algorithm considers tasks' spatial synergy. This ensures that if a robot wins a task T , then it is more likely to win other tasks that are close to it. Therefore, a robot

R_i located at position p_i bids on a task T_j located at position p_j by the minimum of (a) the distance from its location (p_i) and the task's location (p_j); (b) the minimum distance between the task's location (p_j) and any of the robot's assigned but uncompleted tasks' location (p_k), where the robot's assigned tasks are recorded in a special list denoted by l'_i . For example, in Figure 2 if the task T_1 is assigned to the robot R_2 , following that, the robot R_2 bids on the next task T_2 by the distance between T_1 and T_2 . Hence, the traveled distance cost that a robot will bid based on is given by Equation (4).

$$d_{ij} = \text{MIN}(|p_i - p_j|, |p_k - p_j|) \forall T_k \in l'_i. \quad (4)$$

3.6. Auction Process

The auction process is the main component of the market-based task allocation approach; robots cooperate via an explicit negotiation. The proposed method uses a low-cost one-round single auction, meaning that a task is assigned to the robot with the lowest bid in one round. Robots bid by their cost to execute the task. The auction process consists of four sequential steps.

1. Task advertisement: The auction process starts once a robot discovers a new task or receives an advertisement message from an agent outside the system (e.g., a remote-sensing system in automated farming applications). It is worth noting that any robot can act as an auctioneer. The auctioneer is not pre-selected; instead, for each new task, the sensor system that detects a new task, it selects a robot randomly. The task advertisement message includes all the details of the task (i.e., location, quality level, and the required resources). The robot that receives the task's details will consider itself the auctioneer for that particular task.
2. Auction announcement: Then, the auctioneer computes its bid for the task and announces a new auction by broadcasting an auction message *auctionMsg* to its neighbors. The auction announcement message contains all the details of the current tasks. The auction remains open for a sufficient amount of time to allow bidders to send their bids.
3. Bid submission: Once a robot receives an auction announcement message, it computes its bid based on the proposed cost function and submits it to the auctioneer. Unlike the conventional auction-based bidding process, in which a robot computes its bid based only on the distance. The proposed approach considers the distance, quality satisfaction, and workload balance. A bid is a scalar value computed using the cost function and represents a robot's cost to execute the task. The cost function design is explained in the next section.
4. Closing the auction and winner selection: During the auction, the auctioneer maintains a list, *BidsList*, of all current bids and the associated bidders' IDs; then, once the auction time ends, it selects the bidder with the minimum bid (cost) from the bids list *BidsList* and assigns it the current task.

Figure 3 contrasts the state machine of a robot during the auction process. Initially, robots start in an idle state, and if a robot receives a task advertisement, it moves to the auctioneer state. In the auctioneer state, a robot computes its bid and opens a new auction for the task, as shown in Algorithm 1. All robots that receive the *auctionMsg* change from the idle state to the bidder state and run Algorithm 2, in which they compute their cost and send it to the auctioneer. The auctioneer then selects the winning bidder and sends it the task assignment message *winMsg*. After the task has been assigned, all robots, both the auctioneer and bidders, return to the idle state again.

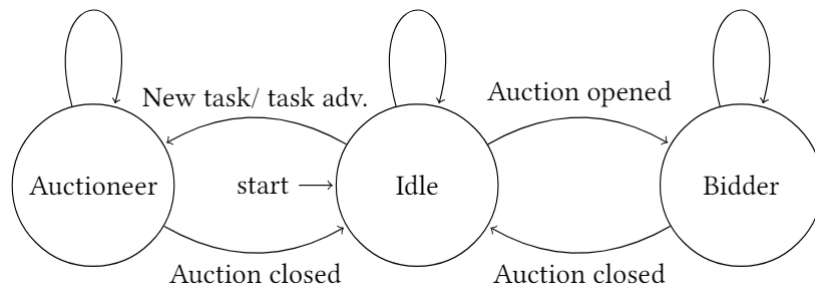


Figure 3. A robot state diagram in during auction process.

Algorithm 1: Auctioneer algorithm

input :New task announcement
output: Assigning a task to the highest bidder

- 1 AuctioneerBid \leftarrow ComputeMyBid();
- 2 AppendToBidsList (AuctioneerId,AuctioneerBid) ;
- 3 BroadcastAuctionMsg();
- 4 AuctioneerTimer \leftarrow t sec ;
- 5 **while** !IsAuctionEnds() **do**
- 6 **if** ReceiveBidMsg() **then**
- 7 AppendToBidsList (bidValue,BidderID) ;
- 8 winnerBidder \leftarrow GetHighestBidder();
- 9 SendTaskAssignMsg(winnerBidder);

Algorithm 2: Bidder algorithm

input : Auction message, Task Assignment message
output: Bid for task, append assigned task to the task list

- 1 In parallel with other procedures;
- 2 **if** AuctionMsgReceived() **then**
- 3 MyBid \leftarrow ComputeMyBid() ;
- 4 sendBidMsg (MyBid,auctioneerID);
- 5 **else if** TaskAssignMsgReceived() **then**
- 6 AppendToTaskList (NewTask);

3.7. Cost Function

Because we have three bidding rules, each of which considers an objective to satisfy, there is a need to combine these bids into a single bid to be sent to the auctioneer. For this purpose, we normalize the values and use the weighted sum model to compute the final bid. We normalize each value to its maximum expected value as follows. Assuming (a) the maximum distance between any two points in the area is D (diagonal of the area), (b) the maximum deviation between a robot quality level and task quality level is ΔQ , and (c) the maximum expected tasks for each robot is L . We use these values to normalized the objective as follows:

$$\bar{d}_{ij} = \frac{d_{ij}}{D}$$

$$\bar{q}_{ij} = \frac{q_{ij}}{\Delta Q}$$

$$\bar{l}_i = \frac{l_i}{L}.$$

Therefore, the cost value for a robot R_i to execute a task T_j is given by the following equation:

$$C_{ij} = W_d * \bar{d}_{ij} + W_q * \bar{q}_{ij} + W_l * \bar{l}_i, \tag{5}$$

where W_d, W_q , and W_l are the weights that sum up to one. The cost is proportional to the distance, the absolute difference between the task quality requirement and the robot quality, and the current load. Hence, all three terms have a positive sign.

We have tested the dependency of all factors (distance, quality, load). Firstly, we will analyze the relationship between the distance and quality factors, hence, we assume $W_l = 0$. Later, we will include W_l when we discuss the impact of synergy on the overall performance. Costs computed based on the distance factor alone lead to insufficient quality satisfaction because there is no relation between the task quality and being close or far from robots. Therefore, the distance and quality factors are independent factors. A test has been conducted to determine the appropriate weights of the distance and quality factors, i.e., W_d and W_q , respectively ($W_d + W_q = 1$) given that the robots' quality levels are three (low, medium, high).

Figure 4 shows the relationship between the traveled distance and quality satisfaction with respect to changes in W_d . The best quality satisfaction occurs when $W_q = 1$ (i.e., $W_d = 0$); however, this is the worst-case for travel distance. As expected, the travel distance decreases as W_d increases, whereas the departure from quality satisfaction increases. The point at which the two objectives intersect (i.e., $W_d = 0.7$ and $W_q = 0.3$) will be the optimal setting for both. These weights are used in the cost function in our simulation experiments.

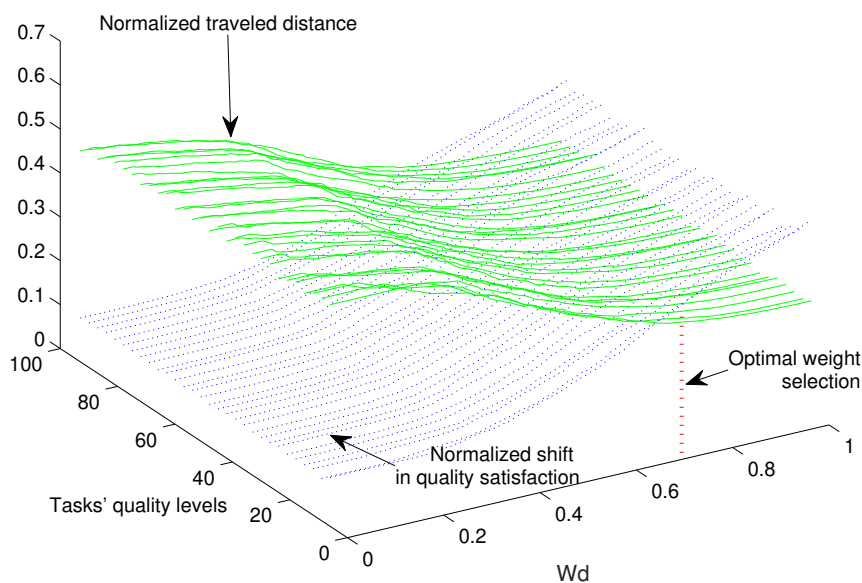


Figure 4. Effect of increasing the value of W_d on the traveled distance and the drift of quality with respect to different tasks' quality levels.

Figure 5 shows that the optimal weights change with respect to the available number of robots' quality level. As the number of robots' quality level increases, the weights of distance gets close to 0.5. This is expected as the normalized value of the difference in the distance between a robot and a task is almost the same as the difference in quality level, as we have a large number of quality level for robots' and tasks'.

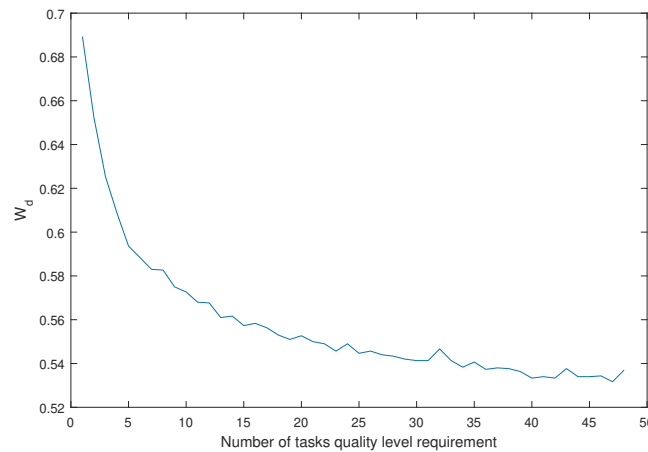


Figure 5. Effect of increasing the number of robots' quality levels on the weight W_d .

Although considering task synergy reduces the total traveled distance, it has a side effect on the workload balance objective. In Figure 6, robot R_1 won all of the tasks, whereas robot R_2 won nothing (i.e., using Equation (4), $(d_{1x} < d_{2x} \forall x = 1, 2, 3, 4)$). We call this problem the 'jump to the middle' problem (JMP). The JMP occurs because once the robot R_1 wins a task located close to the middle of the area of interest, it becomes closer to more new incoming tasks. Because of the synergy definition in Equation (4), it consequently wins an increasing number of tasks, leading to a compromised load-balancing objective. Therefore, we had to re-design the cost function in Equation (5) to include the workload balance objective, and we recomputed the weights associated with each objective.

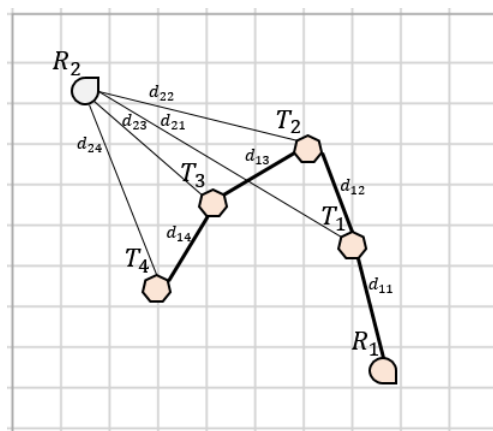


Figure 6. Synergy side effect: R_1 jumped to the middle of the area, and hence, it occupied all tasks.

Because we do not prefer one objective over the others, we set each associated weight to $1/3$, i.e., $W_d = W_q = W_t = 1/3$. Then, we kept the weight of $W_l = 1/3$ and the remaining $2/3$ divided between W_d, W_q proportionally based on the results from Figure 4 such that $W_d = 0.7 * 2/3 = 0.47$, $W_q = 0.3 * 2/3 = 0.23$. Hence, the final weighted sum of the cost function will be as in Equation (6).

$$C_{ij} = 0.47 * \bar{d}_{ij} + 0.23 * \bar{q}_{ij} + 0.3 * \bar{l}_i. \tag{6}$$

We assume that the application doesn't require a specific order of task execution, as is stated in the system model (Section 3.1). Therefore, a robot will visit the assigned tasks using the minimum route, which is a variation of the traveling salesman problem (TSP). We used the nearest neighbor (NN) algorithm [24], which requires a low computational requirement. In the NN algorithm, a robot orders its tasks in descending order based on their distance from its current location and then executes them one after another.

4. Performance Evaluation

To measure the performance of the proposed DYMO-Auction approach, we tested it in different scenarios. We also compared our proposed method with the SSI auction task allocation, in which it is assumed that all information about the tasks is known, that all tasks are announced in each auction sequentially, and one task will be assigned in each auction.

The following performance metrics have been used to evaluate the proposed methods:

1. Quality satisfaction, which measures the quality satisfaction of the tasks. If the tasks have been assigned to robots with similar qualities, the average quality level for all assigned tasks for each individual robot will be nearly equal to the robot's quality level, and it is computed as follows:

$$R_q^i = \frac{\sum_{j=1}^l q_j}{l},$$

where l is the number of tasks assigned to robot R_i , and q_j is a task quality requirement.

2. Load balance, which indicates whether the total load (the number of tasks) has been divided equally among the available robots or not. The optimal load balancing is when all robots execute the same number of tasks.

$$R_L^i = \left\lceil \frac{n}{m} \right\rceil$$

where n is a number of tasks, and m is the number of robots.

3. Sum of the total traveled distance (TTD), which is the sum of the distance traveled by a robot to accomplish all of its assigned tasks. The distance traveled by a robot R_i that is assigned k number of tasks is computed as follows:

$$TTD = \sum_{i=1}^n \sum_{j=1}^k d_{ij},$$

where d_{ij} is the distance between the robot R_i 's location and the nearest task T_j , computed using Equation (4).

5. Simulation Setup

We validated our approach by evaluating its performance using KheperaIII robots on the Webots simulator [25] (Version 8.3.0). The KheperaIII robot [26] was used because it is well known and commonly used in such experiments; it is a differential wheeled robot with dimensions of 13 cm in diameter and 7 cm in height, with a ring of nine infrared (IR) distance sensors, which are used to detect obstacles.

The controllers of the robots are written in the C programming language, and it includes our proposed approach, DYMO-Auction, as well as a proportional-integral-derivative (PID) controller [27], which controls the robot navigation. The PID controller is a feedback control system that continuously calculates the difference between the desired goal and measured current state. A robot uses the PID controller to navigate to a task's location (desired goal), always computing the error (the difference between the current robot location and task location) and minimizing it. The PID controller uses odometry to estimate the current robot location and utilizes the IR sensor ring to avoid obstacles while driving a robot towards a task's location.

Initially, we deployed three KheperaIII robots that were assigned quality levels represented by numbers (2, 5, 8) as low, medium, and high quality, respectively. Tasks emerge with a wide range of quality level requirements; we represent these levels from very low to very high-quality level requirements with numbers from 1 to 9, respectively. Robots and tasks are given a unique color based on their quality level as follows: gray for the low-quality level, blue for the medium quality level, and red for the high-quality level. Figure 7 illustrates the schematic diagram of the network for the simulation experiments.

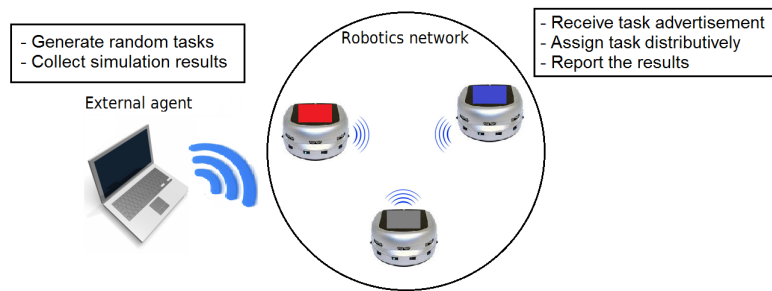


Figure 7. Network schematic diagram of the simulation experiments.

The robots form a robotic network, and they are responsible for receiving tasks from an external agent and then allocating those tasks among themselves. While the robots work on the assigned tasks, they report their status (quality satisfaction, load, traveled distance) to the external agent, which can then be used to generate the simulation results.

Table 2 shows the experimental parameters. The simulation starts by deploying the robots uniformly in the area. Then, tasks are generated following exponential inter-arrival times, uniform locations, and quality requirements. A task advertisement message is produced with each newly generated task in the area. A random robot receives the task advertisement message and considers itself the auctioneer for that task; then, the auction process occurs. The task advertisement message is sent by an external agent located outside the team of robots. It includes the details of the task (location and quality level requirement).

Table 2. Simulation parameters.

Parameter	Value
Number of robots	3, 6, 9, 12
Number of tasks	(12, 24, 36, 48)
Robot’s quality (q_R)	(2, 5, 8)
Task’s required quality (q_T)	(1, 2, 3, 4, 5, 6, 8, 9)
Area size	50 m × 50 m
Task inter-arrival mean time ($1/\lambda$)	80 s
Task execution mean time ($1/\mu$)	140 s
Task’s required resources	(1, 2, 3, 4, 5, 6, 8, 9, 10)
PID parameters K_p, K_i, K_d	(2, 0.01, 0.04)
W_d, W_q, W_l	0.46, 0.21, 0.33
Number of Iterations	35

We have tested the proposed approach with different numbers of tasks (12, 24, 36, and 48 tasks). We repeated each experiment 35 times to achieve a 95% confidence level. Video footage of a nine-task experiment is available [28].

6. Simulation Results

The proposed DYMO-Auction and SSI auction both converge to satisfy the quality requirement for the tasks with an increase in the number of tasks, as shown in Figures 8 and 9. Tables 3 and 4 show that the average quality requirements of the tasks that are executed by the high-quality robots deviate in a negative direction, which indicates that the robot performs tasks that require a quality lower than that of the robot. Alternately, the opposite occurs with the robot with low quality when its tasks’ average quality requirements deviate in a positive direction, meaning that the robot performs tasks with higher quality requirements. These deviations become smaller with an increase in the number of tasks because there will be more tasks of every quality level. Noteworthy, we feed SSI with all information about the upcoming tasks and their quality requirements, while DYMO gets task details as they emerge.

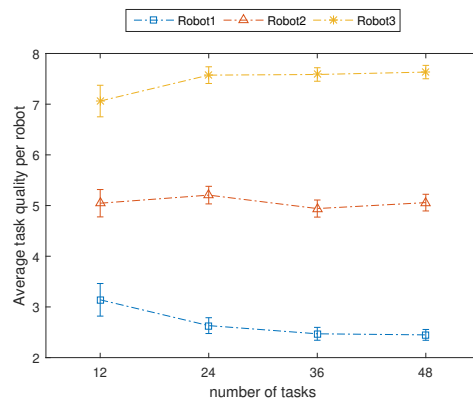


Figure 8. Quality satisfaction using a distributed multi-objective task allocation approach (DYMO-Auction).

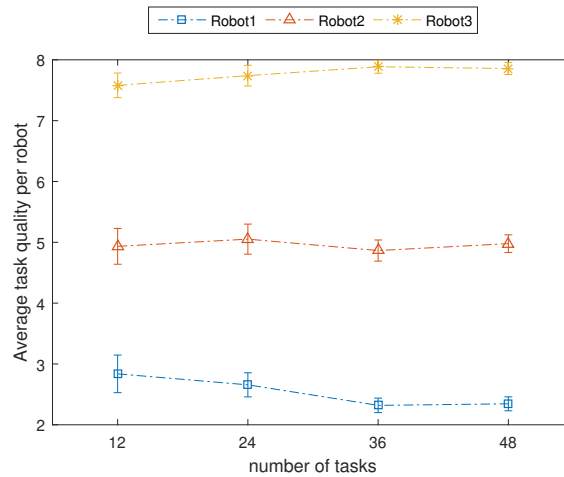


Figure 9. Quality satisfaction using an sequential single-item approach (SSI) auction.

Table 3. Percentage deviation from the quality satisfaction using DYMO-Auction.

High	−11.7263	−5.3375	−5.1787	−4.5837
Medium	−0.9240	−4.1220	+1.1860	−1.1600
Low	+57.0250	+31.5650	+23.4900	+22.4300

Table 4. Percentage deviation from the quality satisfaction using SSI.

High	−5.2438	−2.488	−1.4275	−1.7900
Medium	+1.3240	−1.0440	+2.7080	+0.4400
Low	+41.8550	+32.8350	+15.9600	+17.2350

The load-balancing objective has been met with a deviation from the optimal (the load for all robots is equal) by up to $\pm 2.5\%$ for the SSI auction and up to $\pm 5\%$ for DYMO-Auction as shown in Figures 10 and 11. The deviation in both methods is due to the competition between the load-balancing objective and the other two objectives, which may compromise the load objective. However, the SSI auction deviates less because it has a full view of all tasks ahead, which is not the case for DYMO-Auction.

In terms of the total traveled distance per robot, this objective has a significant weight in the cost function ($Wd = 0.46$). Robots also attempt to select the minimum route to visit/execute all assigned tasks, which further decreases the total traveled distance. The total traveled distance increases by 20% using DYMO-Auction compared to an SSI auction, as shown in Figure 12. An SSI auction requires global

information about the tasks from the beginning, whereas our method assigns the tasks dynamically as they appear.

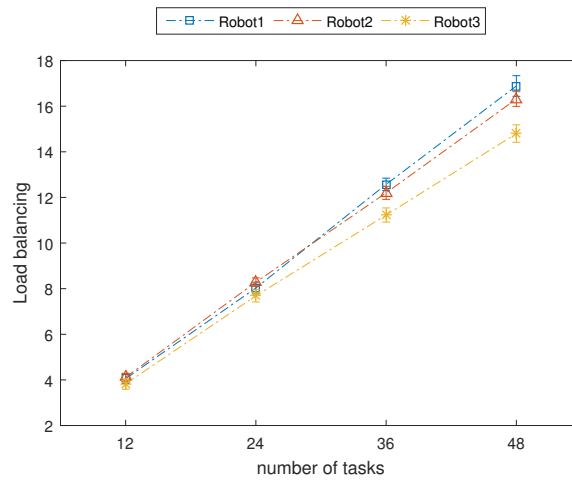


Figure 10. Average number of tasks executed per robot using DYMO-Auction.

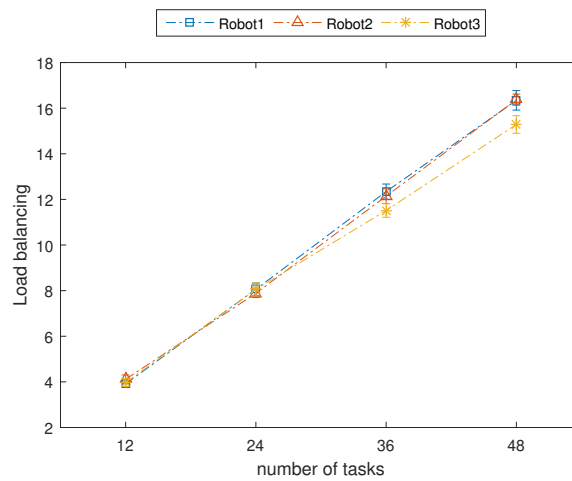


Figure 11. Average number of tasks executed per robot using an SSI auction.

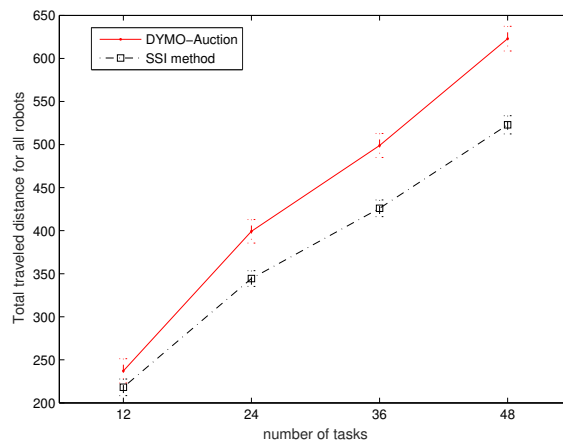


Figure 12. Total traveled distance comparison.

7. Comparison of DYMO-Auction with the FL-MTSP

Overview of the FL-MTSP: The FL-MTSP approach is a centralized solution designed to solve the multiple-depot multiple traveling salesman problem. The solution uses the fuzzy logic algebra to combine two objectives: the total traveled distance by all the robots and the maximum traveled distance by any robot. The solution is composed of two main phases: the assignment phase and the tour construction phase. In the first phase, the inputs of the fuzzy logic system were computed and combined into a single fuzzy metric, reducing the problem to a single objective optimization problem. Then, based on the fuzzy logic system output, each target will be assigned to the robot with the minimum output value. After allocating all targets, an improvement process starts trying to balance the workload among the robots. If the number of targets won by a robot is larger than the ratio of the number of targets to the number of robots, the farthest target is selected and added to the nearest robot. For the tour construction phase, a TSP solver (i.e., TSP_LKH solver [29]) was used. A detailed description of the FL-MTSP is provided in [16].

Results: As shown in Figure 13, the deviation of the load balancing objective for the FL-MTSP is up to 35% (case of 36 tasks). Even though the FL-MTSP is an off-line approach (i.e., the tasks are known from the beginning), this deviation is reasonable as the main objective of the FL-MTSP is to provide a trade-off between the total traveled distance and the maximum traveled distance by any robot. However, for the DYMO-Auction, load balancing is one of the optimized objectives.

We also compare the DYMO-Auction with the FL-MTSP in terms of the total traveled distance. This objective increases by 45% using DYMO-Auction, as shown in Figure 14. For FL-MTSP, the TTD has a significant impact on the assignment as the goal of this approach is to provide a trade-off solution where only two objectives (i.e., the total distance and the maximum distance) compete. However, for DYMO-Auction, three objectives compete to generate a solution. It is also worth noting that FL-MTSP requires global knowledge about the tasks from the beginning.

Figure 15 shows the average task quality per robot. It is clearly shown that the task quality deviates in a negative direction, and in a positive direction. For each robot, the deviation becomes higher with the increase in the number of tasks because there will be more tasks at every quality level. This means that the tasks assignment is independent of the quality of the task. However, for the DYMO-Auction (Figure 8), the deviation of the average task quality per robot is small. This is because the tasks' quality is one of the objectives to be optimized.

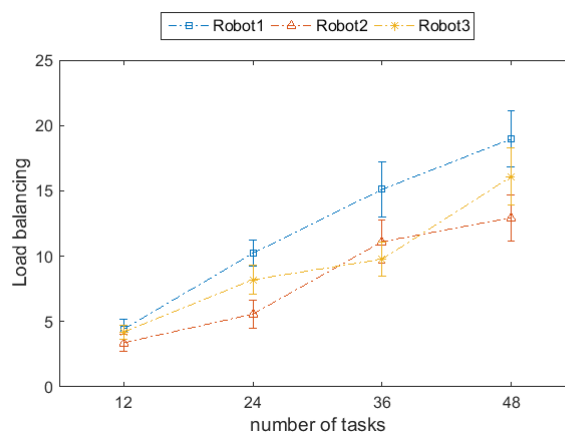


Figure 13. Average number of tasks executed per robot using FL-MTSP.

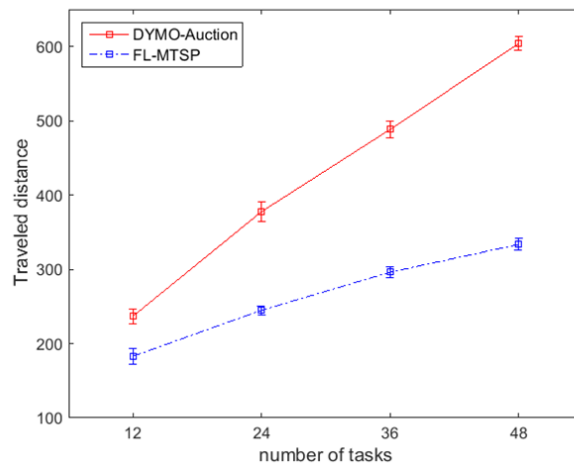


Figure 14. Comparison between DYMO-Auction and Fuzzy Logic Multiple Traveling Salesman Problem (FL-MTSP) in terms of total traveled distance.

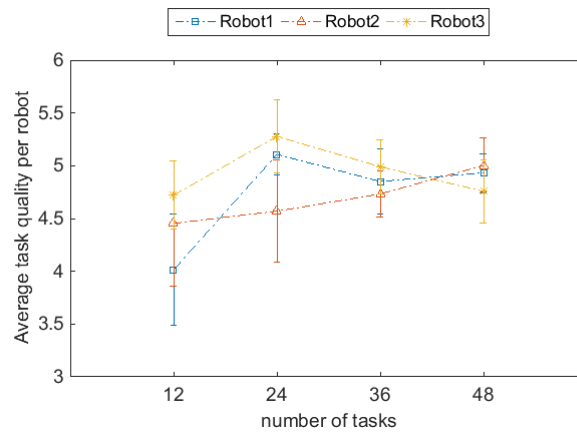


Figure 15. Quality satisfaction using FL-MTSP.

8. Experimental Setup and Results

We validated the proposed method via a real testbed experiment using three Turtlebot2 robots. We used a team of three Turtlebot2 mobile robots in our experimental testbed, as shown in Figure 16.

The Turtlebot2 is a $354 \times 354 \times 420$ -mm³ two-wheeled mobile base platform. It comes with a 3D sensor for obstacle avoidance and a notebook, which is used as the interface for communicating with the robot’s mobile base.

8.1. Experimental Setup

The three Turtlebot2 robots are denoted as $(R_0, R_1, \text{ and } R_2)$, and as we did in the Webots simulator, the robots have three quality levels: low, medium and high, respectively. Before the experiment, we specified 15 task locations in the experimental area ($5 \text{ m} \times 5 \text{ m}$). A task can appear randomly in any of these locations during the experiment, as shown in Figure 16.

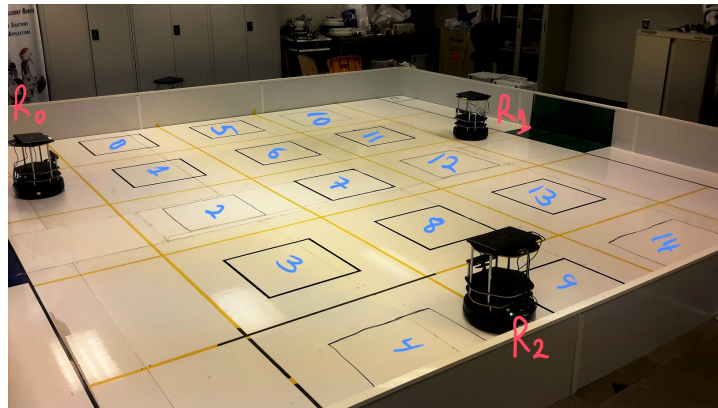


Figure 16. Experimental environment with three Turtlebot2 robots (R_0 , R_1 , and R_2) and 15 locations where a task can possibly appear.

A laptop is used to generate tasks randomly in the area over time (task quality requirement and location are variables) and send an advertisement message to a randomly selected robot (the auctioneer). At the end of each experimental run, the robots send statistical data to the laptop, including their average quality satisfaction, their number of assigned tasks, and their total travel distance. The experiment was repeated 11 times; each time, 21 tasks were generated with an exponential inter-arrival time with a mean of 100 s. The average robot task execution time was set to 230 s. The results were collected and displayed with a confidence level of 90%. Video footage of these experiments is available [30].

8.2. Experimental Results

To demonstrate the practicality of the proposed approach, we have performed the simulation under the same real experiment parameters and presented them side-by-side in Figures 17–19. The experimental and simulation results have no significant differences. Therefore we can say that the proposed method successfully balances between the three objectives, which indicates that all tasks were assigned on average to a robot that satisfied their required quality level as shown in Figure 17. Moreover, tasks are distributed among the available robots nearly equally, as shown in Figure 18. Consequently, robots travel the same distance on average, as shown in Figure 19.

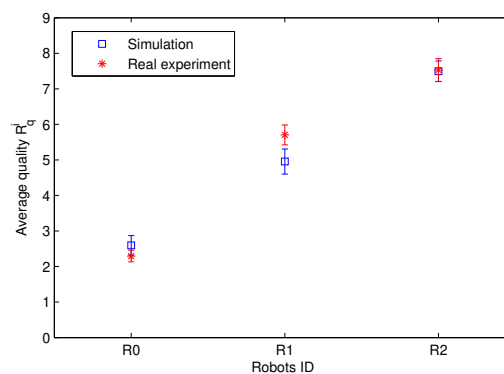


Figure 17. Average assigned task quality per Turtlebot2 robot. The quality levels low (2), medium (5), and high (8) are the quality levels of robots R_0 , R_1 , and R_2 , respectively.

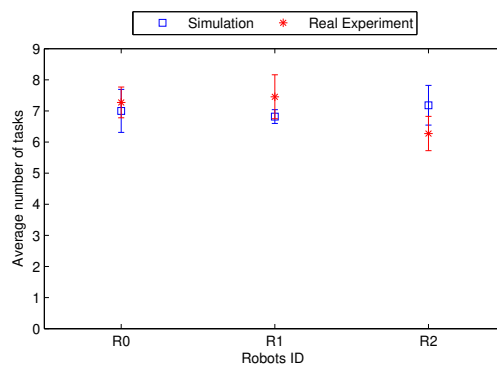


Figure 18. Average number of tasks assigned to each Turtlebot2 robot, where the total number of tasks = 21.

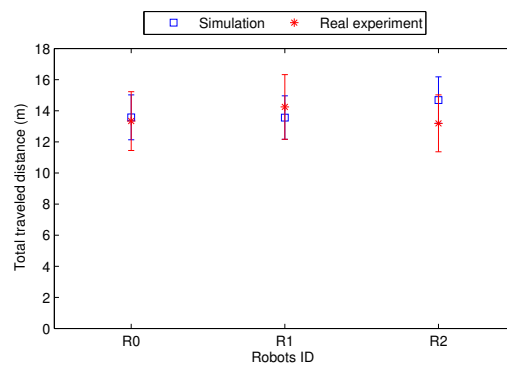


Figure 19. Average total travelled distance per robot (Turtlebot2).

9. Conclusions

In this work, we proposed the task quality as a new factor to be considered in the dynamic multi-robot task allocation (MRTA) problem to indicate the quality of executing a specific task. We also considered workload balancing and traveled distance in the assignment process. The MRTA problem was formulated as an auction process and behavioral process. DYMO-Auction approach was proposed as a distributed multi-objective task allocation approach for dynamic task allocation. We targeted the scenarios where tasks with a quality requirement appear dynamically during the mission. We assumed no prior information about the task and available robots, in particular, tasks location, quality requirements, and the current positions and loads on each robot. We aimed to achieve three objectives: maximize the quality satisfaction, balance the load among available robots, and minimize the total traveled distance. We addressed the MRTA problem as an auction process.

The DYMO-Auction approach combines these objectives using the weighted sum model (WSM) to compute the cost function. The weights were selected and generalized to scale. The proposed approach was tested against the sequential single-item approach (SSI) method. The SSI auction is presented with global information about the tasks and robots, whereas in DYMO-Auction, a task appears with its information dynamically over time. The proposed DYMO-Auction approach obtained comparable results to the SSI method in terms of quality satisfaction, load balancing, and traveled distance, given the limited information available compared to the SSI auction. We conducted testbed experiments for DYMO-Auction using real Turtlebot2 robots and performed the simulation under the same real experimental parameters; the results showed the same trend as the simulation results.

In future studies, the proposed method can be extended to form a generalized framework that considers task waiting time, task priority, task's details changes over execution time. Auction approaches may suffer from communication overhead. In particular, if the number of robots

is large. Challenges may also arise if there are a small number of bidders in the auction. As a solution, hop count can be used to expand or narrow the pool of auction process participants.

Author Contributions: Conceptualization, U.B., M.A. and A.K.; data curation, M.A. and S.T.; formal analysis, U.B.; funding acquisition, U.B.; investigation, U.B., M.A. and A.K.; methodology, U.B., M.A., A.K. and S.T.; project administration, U.B.; resources, U.B.; software, M.A. and S.T.; supervision, U.B. and A.K.; validation, U.B., M.A., A.K. and S.T.; Visualization, M.A. and S.T.; writing—original draft, M.A.; writing—review and editing, U.B., A.K. and S.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Plan for Science, Technology, and Innovation (MAARIFAH)—King Abdulaziz City for Science and Technology through the Science and Technology Unit at King Fahd University of Petroleum and Minerals (KFUPM), the Kingdom of Saudi Arabia, award project No. 11-ELE2147-4. In addition, Anis Koubaa would like to acknowledge the support by the Robotics and Internet-of-Things Lab of Prince Sultan University.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Koubaa, A.; Khelil, A. *Cooperative Robots and Sensor Networks 2014*; Springer: Berlin/Heidelberg, Germany, 2014; Volume 554.
2. Lee, D.H.; Zaheer, S.A.; Kim, J.H. A Resource-Oriented, Decentralized Auction Algorithm for Multirobot Task Allocation. *IEEE Trans. Autom. Sci. Eng.* **2015**, *12*, 1469–1481. [[CrossRef](#)]
3. Mi, Z.; Yang, Y.; Ma, H.; Wang, D. Connectivity preserving task allocation in mobile robotic sensor network. In Proceedings of the 2014 IEEE International Conference on Communications (ICC), Sydney, Australia, 10–14 June 2014; pp. 136–141.
4. Sheng, W.; Yang, Q.; Tan, J.; Xi, N. Distributed multi-robot coordination in area exploration. *Robot. Auton. Syst.* **2006**, *54*, 945–955. [[CrossRef](#)]
5. Elango, M.; Nachiappan, S.; Tiwari, M.K. Balancing task allocation in multi-robot systems using K-means clustering and auction based mechanisms. *Expert Syst. Appl.* **2011**, *38*, 6486–6491. [[CrossRef](#)]
6. Gong, J.; Qi, J.; Xiong, G.; Chen, H.; Huang, W. A GA based combinatorial auction algorithm for multi-robot cooperative hunting. In Proceedings of the 2007 International Conference on Computational Intelligence and Security, Harbin, China, 15–19 December 2007; pp. 137–141.
7. Gerkey, B.P.; Mataric, M.J. Sold!: Auction methods for multirobot coordination. *IEEE Trans. Robot. Autom.* **2002**, *18*, 758–768. [[CrossRef](#)]
8. Tolmidis, A.T.; Petrou, L. Multi-objective optimization for dynamic task allocation in a multi-robot system. *Eng. Appl. Artif. Intell.* **2013**, *26*, 1458–1468. [[CrossRef](#)]
9. Dias, M.B.; Zlot, R.; Kalra, N.; Stentz, A. Market-based multirobot coordination: A survey and analysis. *Proc. IEEE* **2006**, *94*, 1257–1270. [[CrossRef](#)]
10. Krieger, M.J.; Billeter, J.B. The call of duty: Self-organised task allocation in a population of up to twelve mobile robots. *Robot. Auton. Syst.* **2000**, *30*, 65–84. [[CrossRef](#)]
11. Agassounon, W.; Martinoli, A. Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems. In Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 3, Bologna, Italy, 15–19 July 2002; pp. 1090–1097.
12. Abukhalil, T.; Patil, M.; Sobh, T. A comprehensive survey on decentralized modular swarm robotic systems and deployment environments. *Int. J. Eng.* **2013**, *7*, 44.
13. Koenig, S.; Tovey, C.; Lagoudakis, M.; Markakis, V.; Kempe, D.; Keskinocak, P.; Kleywegt, A.; Meyerson, A.; Jain, S. The power of sequential single-item auctions for agent coordination. In Proceedings of the National Conference on Artificial Intelligence, Boston, MA, USA, 16–20 July 2006; Volume 21, p. 1625.
14. Lagoudakis, M.G.; Markakis, E.; Kempe, D.; Keskinocak, P.; Kleywegt, A.J.; Koenig, S.; Tovey, C.A.; Meyerson, A.; Jain, S. Auction-Based Multi-Robot Routing. *Robot. Sci. Syst.* **2005**, *5*, 98.
15. Lawler, E.L.; Lenstra, J.K.; Rinnooy-Kan, A.G.; Shmoys, D.B. *Traveling Salesman Problem*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 1985.

16. Trigui, S.; Cheikhrouhou, O.; Koubaa, A.; Baroudi, U.; Youssef, H. FL-MTSP: A fuzzy logic approach to solve the multi-objective multiple traveling salesman problem for multi-robot systems. *Soft Comput.* **2017**, *21*, 9351–7362. [CrossRef]
17. Sariel, S.; Balch, T. Real time auction based allocation of tasks for multi-robot exploration problem in dynamic environments. In *Proceedings of the AAAI-05 Workshop on Integrating Planning into Scheduling*; AAAI: Palo Alto, CA, USA, 2005; pp. 27–33.
18. Mohammed, A. Distributed Task Allocation in Multi-Robot Networks. Master's Thesis, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, 2017.
19. Gerkey, B.P.; Mataric, M.J. A formal analysis and taxonomy of task allocation in multi-robot systems. *Int. J. Robot. Res.* **2004**, *23*, 939–954. [CrossRef]
20. Sallam, G.; Baroudi, U. COVER: A Cooperative Virtual Force Robot Deployment Technique. In *Proceedings of the 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, Liverpool, UK, 26–28 October 2015; pp. 1308–1315.
21. Ahmed, A.A.; Al-Shaboti, M.; Al-Zubairi, A. An Indoor Emergency Guidance Algorithm Based on Wireless Sensor Networks. In *Proceedings of the 2015 International Conference on Cloud Computing (ICCC)*, Riyadh, Saudi Arabia, 26–29 April 2015; pp. 1–5.
22. Nasser, N.; Ali, A.; Karim, L.; Belhaouari, S. An efficient Wireless Sensor Network-based water quality monitoring system. In *Proceedings of the 2013 ACS International Conference on Computer Systems and Applications (AICCSA)*, Ifrane, Morocco, 27–30 May 2013; pp. 1–4.
23. Seelan, S.K.; Laguette, S.; Casady, G.M.; Seielstad, G.A. Remote sensing applications for precision agriculture: A learning community approach. *Remote Sens. Environ.* **2003**, *88*, 157–169. [CrossRef]
24. Johnson, D.S.; McGeoch, L.A. The traveling salesman problem: A case study in local optimization. *Local Search Comb. Optim.* **1997**, *1*, 215–310.
25. Webots. Commercial Mobile Robot Simulation Software. 2016. Available online: <http://www.cyberbotics.com> (accessed on 20 January 2020).
26. KheperaIII. K-Team Mobile Robotics. 2016. Available online: <http://www.k-team.com> (accessed on 20 January 2020).
27. O'Dwyer, A. *Handbook of PI and PID Controller Tuning Rules*; World Scientific: Singapore, 2009; Volume 57.
28. Robotics, K. Auction Based Task Allocation Using Webots Simulator. 2016. Available online: <https://www.youtube.com/watch?v=HURvQslTO8> (accessed on 20 January 2020).
29. Helsingaun, K. LKH. 2012. Available online: <http://www.akira.ruc.dk/~keld/research/LKH/> (accessed on 20 January 2020).
30. Robotics, K. Task Allocation Experiment Using Turtlebot2. 2016. Available online: <https://www.youtube.com/watch?v=SC-V6tRVdIo> (accessed on 20 January 2020).

