Task allocation in multi-robot system using resource sharing with dynamic threshold approach

Nayyer Fazal, Muhammad Tahir Khan, Shahzad Anwar, Javaid Iqbal, Shahbaz Khan

1 Department of Mechatronics Engineering, UET Peshawar, Peshawar, Pakistan, 2 College of Electrical and Mechanical Engineering, NUST, Islamabad, Pakistan, 3 Institute of Manufacturing Engineering Management, University of Engineering and Applied Sciences, Swat, Pakistan

* nayyerfazal@uetpeshawar.edu.pk

Abstract

Task allocation is a fundamental requirement for multi-robot systems working in dynamic environments. An efficient task allocation algorithm allows the robots to adjust their behavior in response to environmental changes such as fault occurrences, or other robots’ actions to increase overall system performance. To address these challenges, this paper presents a Task Allocation technique based on a threshold level which is an accumulative value aggregated by a centralized unit using the Task-Robot ratio and the number of the available resource in the system. The threshold level serves as a reference for task acceptance and the task acceptance occurs despite resource shortage. The deficient resources for the accepted task are acquired through an auction process using objective minimization. Despite resource shortage, task acceptance occurs. The threshold approach and the objective minimization in the auction process reduce the overall completion time and increase the system’s resource utilization up to 96%, which is demonstrated theoretically and validated through simulations and real experimentation.

1. Introduction

Task Allocations are considered in a range of applications, such as path exploration [1], path planning [2], foraging [3], tracking [4], and transportation [5], and have also been applied to many domains like industrial engineering [6]. With the increase in demand for multi-robot cooperation in complex and uncertain environments, the significance of the Task Allocation in Multi-Robot systems, i.e., to determine an efficient task assignment to improve the system efficiency [7] and task completion time [8–11] has increased. A challenge in using teams of robots is the cooperation of robots to perform tasks while optimizing one or more objectives. Each application has different requirements for Task allocation, as in search and rescue operations, the survivors need to be assisted within specified deadlines, where the objectives are to maximize the number of survivors and minimize the average waiting time of their rescue. Similarly, tasks in the manufacturing process require task completion with different constraints.
The optimization objectives usually include completing all the tasks in the shortest possible time. As opposed to the manufacturing environment, the search and rescue operations normally deal with the unstructured environment, highly dynamic conditions, and limited resources. Hence the Task Allocation techniques and objective function requirements may vary with application and different frameworks. For example, a Centralized system offers a clear chain of command, a focused vision, and quick implementation using the global information and optimization-based heuristics to generate global solutions for task allocation [12–14]. However, it lacks the agility about robustness, scalability, and flexibility in dynamic circumstances.

On the other hand, distributed control is generally complex to implement and lacks global knowledge of the system’s current status. Hence to attain the benefits of different frameworks, this study aim to develop a decentralized system. In a decentralized approach there is no single point where the decision is made. Instead, each node decides its own behaviour and the overall behavior is an aggregate response of all the node’s response, as shown in Fig 1 for demonstration.

To develop a decentralized system a dynamic Adaptive Inception Value (AIV(i)) is introduced here, which is generated by accumulating the entire system status by aggregating the number of tasks, number of robots, and the faults occurrence or resource absence in the system. The robot uses this aggregated value as a reference for task acceptance. When a robot encounters a task and the resource deficiency relevant to the detected task is not more than the acceptable deficiency level that is determined through AIV(i), the robot becomes an auctioneer to acquire the deficient resources. The task allocation and completion times are further improved by incorporating objective minimization in the bidding process. The auctioneer can settle its main concern of either time minimization or the distance travelled by the robot by minimizing the corresponding objective value.

Extensive work has been performed in the field of task allocation in multi-robot systems; however the focus is always on the bid calculation, and the winner selected by the auctioneer, also the tasks are supposed to be assigned by a centralized auctioneer to the robot that matches the task criterion which increases the computation time of task allocation. Moreover, the auctioneer selection, the resource absence, or resource failure and its effect on auctioneer selection has never been addressed. This paper aims to develop a methodology for task allocation which can easily adapt the changes with less computation time for task allocation and allows any robot in the system to become an auctioneer if its deficiency is below the threshold level as settled by the central monitoring. The proposed approach reduces the computation time of task allocation and, consequently, task completion time and the system’s communication burden.

The remaining of the paper is organized as follows, different techniques used for task allocation and the gap analysis are presented in Section 2. To overcome the gap, the proposed

![Fig 1.](https://doi.org/10.1371/journal.pone.0267982.g001)
methodology is proposed in Section 3. To validate the performance of the proposed method, simulation and real experimentation have been performed in Section 4, whereas the results obtained and validated with benchmark techniques are given in Section 5. Finally, the conclusion of the proposed method and results and the future work are given in Section 6.

2. Related work

The literature on task allocation for multi-robot systems (MRS) is very broad [15,16]. Task allocation mechanisms aim to allocate tasks to the appropriate robot in the most optimal way based on one or more objectives. In [15], a complete organization of task allocation problems is delivered, which is repeatedly used in the task allocation literature. Usually, two approaches are employed to solve task allocation problems in literature: market-based approaches and optimization-based approaches [17,18]. The existing work on each of these two approaches is explained as follows:

**Optimization approach**

The optimization-based approach is usually employed in a centralized system where one centralized point can access the global information and find the optimal global solution for task allocation. This type of NP-hard optimization problem is generally solved using metaheuristics or heuristics. The optimization algorithms that are frequently in use for task allocation are Ant Colony Optimization [19], Simulated Annealing [20], and Genetic Algorithms [21].

A broader review is presented in [22] on different techniques and algorithms of optimization. A team of heterogeneous or homogeneous robots can cover more area and could be more robust to failures than a single all-purpose robot [23,24]. The dynamics of collaboration between the robot and the workers in a shared workplace and its impact on task allocation are key parameters for optimal solutions [25]. The collaboration among many robots has been employed in [26] to make a coalition of all the targets for the sequence in which UAVs need to be assigned to the task to get a minimal time of completion. Sometimes a single optimal allocation plan may not work efficiently as the uncertain emergencies may frequently occur during real execution processes of multi-robot cooperation, and the predefined optimal task allocation techniques may not work as expected. To tackle different forms of failures, outstanding work has been performed. For the disputation of resources, the interference issue is addressed when multiple robots contend for the same resource [27]. An ontology-based mechanism is proposed in [28] for the fluctuating coupling affairs. Still, these techniques can only handle specific types of measures. An optimal assignment strategy is desired, such that the scheme can switch to another solution when it cannot continue anymore. Usually, more robots are available for limited tasks, but the problem arises with robots and resource constraints and the tasks are time-bound. The multiple optimal solutions are normally employed in manufacturing and have highly valued practical applications. Second, the uncertain failures in the system and the physical and energy constraints impact the long-term execution. Also, multiple solutions with similar or the same quality are important for the flexibility and robustness of a system. Hence, as premeditated by a centralized unit, the multiple optimal solutions can act as the prior information for a distributed multi-robot system to improve the cooperative work and reduce the chances of deadlock and resource conflict.

The Task allocation based on optimization techniques works well for simple and small systems but usually lacks optimality for larger and complex systems. The number of tasks and robots increase the computational complexity exponentially. The user usually limits the number of iterations for an optimization solver to bound the execution time of the process to compute more task allocations in the limited time, which allows a faster response to dynamic
changes. By limiting the optimization time, scarcer solutions are explored by the algorithm. This prevents the algorithm from fully exploring the search space, resulting in depraved optimization and giving sub-optimal solutions to more complex and larger systems. The decentralized market-based task allocation approaches could be used as an alternative that is used in this paper as a solution.

**Market based approach**

The computational time of the Market-based approach may scale better with the problem complexity and overcome the limitations of optimization-based approaches. The competence of auction-based algorithms has been proved in [29,30]. A market-based approach is a decentralized optimization that uses an auction method with bids to attain an assignment. Generally, an auctioneer announces the task to all the robots, and in response, the robots compute bids on tasks and send it back, and the auctioneer allocates the task to the robot with the lowest bid. This very basic auction-based protocol is called the CNET protocol [31]. Other variations to this protocol are deeply investigated, such as parallel single-item auctions, where every robot computes a bid for each task. The auctioneer assigns all tasks at once. The solutions are expected to be highly sub-optimal since it does not account any interaction between tasks [32]. Another method is the combinatorial auctions where every robot computes a bid for each subset of the task on offer. By considering the possible combination of tasks in the bidding process, an interaction is established [32]. The combinatorial auction method usually produces solutions that are close to optimal but need a very high computation time. A compromise between the parallel and combinatorial auctions is a Sequential single-item auction for computational effort and optimality. The auction continues for several rounds, and in each round, a task is assigned to a robot. The Sequential single-item auctions, in general, do not guarantee to find an optimal solution. However, synergies are exploited between tasks.

The most challenging problem in multi-robot systems is to optimally assign robots to tasks that optimize the overall system performance and employ more than one robot for a task. Multi-robot task allocation is a complex problem for heterogeneous robots with different capabilities and unreliable resources that are to perform tasks with different constraints and requirements in an optimal way and to attain the benefits of both centralized and distributed frameworks. Hence, the authors here use the market-based approach to select an optimal solution proposed by the bidder robots that minimizes the given objectives. The proposed approach also selects the auctioneers in the system using the system status as a criterion which may generate multiple parallel auctions in the system.

**3. Methodology**

In our proposed approach for task allocation using resource sharing, each robot has to serve transportation services. The overall approach is termed a decentralized approach where each robot is independent to decide, but the resulting system behaviour is the accumulation of all robot’s response as shown in Fig 1(a). An accumulative value is generated by the centralized unit used as reference value for task acceptance by the robots. Although it is arguable to refer to the proposed system as centralized or decentralized, the authors of this paper termed it a decentralized system.

The tasks are accepted by the robots based on reference/ threshold value (AIV(i)) set by the centralized unit whereas the deficient resources are acquired from other robots using Market-based approach.

The different auction principles for the Market-based approach are,
I. In a Parallel single-item auction, each robot computes its bid for all the tasks, and the auctioneer allocates all the tasks at once.

II. In a combinatorial auction, each robot computes a bid for every subset of tasks on-demand. By considering the combination between the tasks in bidding process a combined effect is produced.

III. The sequential single-item auction is a compromise between combinatorial and parallel auctions in computational effort and optimality. One resource sharing request is granted in each round. Literature [32–34] establishes that the sequential single-item auction gives best trade-off between simplicity and optimality.

Whenever there is a shortage of resources relative to the encountered task and the deficiency of the robot is within the frontiers of $AIV(i)$, the robot becomes an auctioneer and announces the resource shortage to the system’s robots. Then, the bidder robots send the Sealed first-price auctions [35], where all robots place their bids secretly without the knowledge of other robot’s bids, and the auctioneer selects the best bid.

### 3.1 Task robot encounter

The system starts with a specific number of resources, the number of tasks, and the number of robots. Then, the robots randomly search for a task. When the robot has detected a task, it evaluates the task with its available resources and compares its deficiency with the acceptable deficiency level settled by the Centralized unit, as depicted in Fig 2. The tasks are considered as purely transportation tasks where each robot should detect and recognize the task and place it at the designated location. Table 1 gives a list of abbreviations used in the methodology.

Fig 3 shows the mapping of resource requirement of each task to the resources of a robot which identifies the unavailable resources of the robot and may demand other robots in the system provided it satisfies the criteria, described in the forthcoming section.

---

**Fig 2.** Generalized methodology for the proposed resource sharing mechanism.

https://doi.org/10.1371/journal.pone.0267982.g002
The resources of the $n^{th}$ robot is represented with $r_n$, given as

$$r_n = \{r_1, r_2, \ldots r_L\}$$  \hfill (1)

$r_L$ denotes the maximum number of working/operational resources of the robot.

The Resource Requirement matrix of the task is represented as given in Eq (2)

$$\hat{r} = \{\hat{r}_1, \hat{r}_2 \ldots \hat{r}_M\}$$  \hfill (2)

Each subtask has a different resource requirement. When a robot detects a task, it maps its available resources with the resource requirement of the task, as shown in Fig 3. Where each task has several Subtasks, represented in Eq (3)

$$T = \{\tau_1, \tau_2 \ldots \tau_p\}$$  \hfill (3)

The fitness of the robot to perform the subtask of the detected task is 1 if the resource relevant to the subtask is available and "0" otherwise, represented in Eq (4)

$$f(R_n, \tau_{p_{k}}) = \begin{cases} 1 & \text{if } r_{\tau_{p_k}} \in r_k \\ 0 & \text{Otherwise} \end{cases}$$  \hfill (4)

where $\tau_{p_{k}}$ represents the $p^{th}$ subtask of task $k$, and $r_{\tau_{p_k}}$ represents the resource required for $p^{th}$ subtask of the $k^{th}$ task.

The accumulative fitness of $n^{th}$ robot for the $k^{th}$ task is computed using Eq (5)

$$F_{nk} = \sum_{p=1}^{p} f(R_n, \tau_{p_{k}})$$  \hfill (5)

For the same AIV($i$), each Task-Robot encounter has a different fitness value. The robot computes the fitness value for each task-robot encounter and each encounter accumulates and gives an aggregate status of the system. Once the fitness is evaluated, the robot computes its scarcity level relative to the task, which is the difference between the fitness value and resource

![Fig 3. Mapping of subtask’s requirement with available resources of the robot.](https://doi.org/10.1371/journal.pone.0267982.g003)
requirement. A generalized task-robot encounter is depicted in Fig 4.

\[ \zeta = |F_{nk} - \tilde{r}| \]  

(6)

The \( \zeta \) (scarcity) categorizes the subsequent stage of the task, explained in the forthcoming, Task Evaluation section.

### 3.2 Task evaluation

For task evaluation, the robot compares its scarcity level with the \( AIV(i) \), which serves as a threshold level for the instant \( i \), defined by the central unit.

The Central Unit computes the \( AIV(i) \) by considering the Task-Robot ratio and the number of operational resources in the system. The initial Task-Robot proportion is given as Eq (7)

\[ TR_i = \frac{r_i}{R_{io}} \]  

(7)
Where the average resources per robot are computed as in Eq (8), the $r_{\text{in}}$ represents the initial total resources in the system

$$\mathcal{Z} = \frac{r_{\text{in}}}{R_{\text{in}}} \tag{8}$$

When there is a change in the total number of resources that might occur due to resource failure, the average resources per robot, named Operational robots, changes. The Operational Robots are calculated using Eq (9)

$$R_{\text{opr}} = \frac{r_{\text{opr}}}{\mathcal{Z}} \tag{9}$$

This $R_{\text{opr}}$ helps determine the total effective operational robots in the system and revises the Task-Robot ratio.

The instantaneous value of the task has also been modified in Eq (10), as some of the tasks might have ended with completion, which results in changing the total number of tasks at the instant $i$ as $T_i$. For Normalization,

$$\frac{T_i}{T_{\text{max}}} = \frac{r_i}{R_{\text{opr}}} \tag{10}$$

Hence using Eq (10), the central monitor updates the Adaptive Incentive Value (AIV),

$$AIV(i) = \frac{T_i \mathcal{Z}}{T_{\text{max}}} \tag{11}$$

The $AIV(i)$ accumulates the entire system status into a single dynamic parameter updated with each fault occurrence or task completion in the system. The $AIV(i)$ is used as a reference value for task acceptance. The robot is allowed to accept the task, despite resource shortage and, can become an auctioneer to acquire the deficient resources through resource sharing when the deficiency level does not exceed the reference $AIV(i)$. Adopting this approach ensures realistic resource sharing with reasonable resource switching among the robots, as excessive switching may also degrade system performance. The $AIV(i)$ computed by the centralized unit is globally known to all the robots and computes the acceptable deficiency level of the robot for the encountered task as given in Eq (12)

$$\text{Def}_{k} = \bar{r}_k.AIV(i) \tag{12}$$

Eq (12) gives the acceptable deficiency level for a task. The decision making is explained using Eqs (6) and (12).

The AIV affects the system in the following way,

With a higher Task-Robot ratio, the task acceptance rate is high, which increases resource sharing in the system in order to increase the overall number of task completion and a greater number of robots become auctioneers.

Whereas when Task-Robot ratio is lower and with fewer faults reported in the system, the robot focuses on improving the quality of performance of the task by accepting tasks with a higher fitness value hence engaging fewer numbers of robots in performing a task.

As an explanatory example, at a value of $AIV(i) = 0.2$ and a task requirement $\bar{r}_k = 10$, the robots can accept the task if 80% of the resources relevant to the detected task and acquire the remaining 20% using resource sharing through the auction process. The actual deficiency of the robot relevant to the detected task is calculated using Eq (6).
3.3 Decision making

i. When \((\zeta \geq Deff_k)\), the robot ignores the task as the robot’s deficiency to the relevant task is less than the acceptable deficiency level and the robot searches for another task.

ii. When the robot’s resources match the resource requirement of the task and \((\zeta = 0)\), Eq (6), the robot performs the detected task.

iii. When \(\zeta \leq Deff_k\), the robot’s deficiency to the detected task is less than or equal to the acceptable deficiency level of the task, the robot completes the detected task by becoming an auctioneer and requesting other robots in the system to share their surplus resources.

3.4 Auction process

3.4.1 Bidder’s role. To participate in the auctions, the robot needs to know its location, its local task list, its resource engagement, and the resource requirement of the newly detected task. The robot computes the cost in terms of time and sends their bids using two of the rules available in [36], the MiniSum and MiniMax. The MiniSum computes the extra cost for utilizing its resources with the newly announced task in its task list where the task is inserted in the task list using the optimal insertion heuristic [37]. For this purpose, the robot first computes the cost for the resource usage, for the tasks that are already assigned as \(C_1\) and the cost with the newly announced task as \(C_2\) and figures out its difference \((\beta_{ms} = C_2 - C_1)\).

Whereas MiniMax computes the total cost for the already assigned task and the newly announced task with no marginal cost. Consequently, the MiniSum tends to minimize the total path cost of overall robots, whereas the MiniMax reduces the total time of tasks execution. Both approaches have their paybacks in certain situations. Once these bids are calculated the total bid is sent to the auctioneer using a linear combination of both as given in the following Equation

\[
\beta_{tot} = (1 - \alpha)\beta_{mm} + \alpha\beta_{ms}
\]  

(13)

To prioritize the effect of one of these as per requirement, another tunable parameter \(\alpha\) is introduced here. When the Task-Robot ratio is higher than a specific ratio and the tasks need to be completed within a specific time, then \(\alpha = 0\), whereas when there are fewer tasks in the system, and the goal is to reduce the total cost, then \(\alpha = 1\). When both the objectives are minimized simultaneously the value of \(\alpha\) is set at 0.5.

3.4.2 Auctioneer’s role. The rule for the bid selection by the auctioneer can alter [16]. The simple Lowest Bid has been used here where the auctioneer computes the bids sent by all the bidders and accept the resource offered by the robot which has sent the lowest bid. A combination of our auctioneer selection rule and the Lowest Bid principle minimizes the total cost and the total time in which all the tasks are executed by selecting optimal solutions.

3.5 Task-robot architecture

We have two entities in Resource Sharing based task allocation: the auctioneer and the bidder. Here we introduce two task-robot architectures, one which assumes the bidding rule algorithm, and the other which determines the winner.

3.5.1 Bidder algorithm. The region of the announcement is generally considered for large-scale systems. In the proposed approach, when a robot becomes an auctioneer, the auctioneer announces the task to all robots in the system as shown in Fig 5. In the case of very large scale systems, this may require many unnecessary computations as those robots
That is very far from the auctioneer and needs to compute the bids. However, in the case of sharing the processing capabilities of the robot, the distance between the robots seldom matters.

### 3.5.2 Auctioneer’s algorithm

The role of the auctioneer can be assigned on be token-based or time-based mechanism. The authors here have used the $AIV(i)$ as a reference for the dynamic criterion for auctioneer selection which is announced by the centralized unit and changes in the system status. If the robot’s deficiency, relevant to the detected task is less than or equal to the acceptable deficiency level $Def_{i}^{slack}$, computed through $AIV(i)$, the robot becomes an auctioneer and requests deficient resources depicted in Fig 6.
4. Experimentation

To validate the proposed mechanism, a manufacturing warehouse was formulated with heterogeneous multi robots that were supposed to detect and recognize the task and transport it to the goal location. Tasks of different colours (red, green, blue) and different shapes as shown in Fig 5 were introduced for the robots to classify and preserve them at different warehouse locations. Each robot was supposed to have capabilities for task detection and transportation, such as RGB sensor, Camera, image recognition, task organization to the goal location, the Gripper, and the Bumper for pushing the task. Furthermore, random resource absence and faults were introduced to see their effect on task allocation and performance. Resource sharing was classified into two categories.

I. One which needed physical movement of the robot to share its resources such as RGB sensor, Camera, Gripper and Bumper

II. The other which did not need any physical movement for sharing resources such as image recognition and task organization, where resources could be shared remotely

Each robot had a certain battery level and local task list that contained the tasks assigned to the robots. The Task-Robot ratio was also varied for different scenarios.

**Parameters used**

The parameters listed in Table 2 are selected for both the simulations and real experimentation. The same parameters were considered for comparison with baseline techniques.
The number of tasks were varied from 3–36, with number of robots 4 for simulation and 3 for real experimentation to validate the fact that the performance depends on the task-robot ratio and not only the total robots in the system. The performance remained the same when the tasks were increased beyond 36. The camera frame rate was set at 10Hz in both the real and simulation setup because a very high frame rate could increase the image processing load and could introduce additional delays in the process.

4.1 Simulation experiment

We validated our proposed method by testing its efficiency using a Pygame simulator with 4 robots in the environment with Python API added to interact with the environment, as shown in Fig 7. All the robots were capable of communicating and interacting with each other, and each robot had the capabilities to become the bidder or the auctioneer. The Red, Green, and Blue rectangular shapes represent the goal locations.

The experimental scenario is clearly shown in Fig 7 and 7a shows the initial position of the robots. After some time \(R_4\) encounters a robotic arm failure whereas \(R_3\) encounters an RGB failure. \(R_4\) initiates an auction because its deficiency level is less than the acceptable deficiency level. All the robots send their bids. The lowest bid is sent by \(R_3\) hence the partially completed task is allocated to \(R_3\).

4.2 Real experimentation

The onboard processor, Raspberry Pi, fully executed the implementation framework to reveal the task allocation mechanism as shown in Fig 8. Each robot was capable of communicating with other robots in the system through a WiFi module installed on the Raspberry Pi using the communication protocol TCP/UDP. The internal structure of the real robot along with the virtual viewer is shown in Fig 9 for clear description of the internal resources of the robot.

Using serial communication, an Arduino connected to the Raspberry Pi controlled the Motor Driver L298.

To perform the real experimentation, a server that served as a centralized monitor was created on quad processor Intel i7 7700HQ, 16 GB RAM, and NVIDIA GeForce GTX 1080 GPU using the Socket library Python, with all the robots connected to the server using Multithreading. The Graphical User Interface (GUI) was visualized using the VNC viewer to perceive the status of each robot and tasks in the system to generate a global view of the system, as shown in Fig 8.

In the experiment, shown in Fig 9, the robot \(R_3\) was considered to have a Physical resource absence. More specifically, the robotic arm/Gripper was observed whereas \(R_1\) encountered an

| Parameter                  | Value     |
|----------------------------|-----------|
| Simulation time step       | 0.002s    |
| Camera frame rate          | 10Hz      |
| Camera resolution          | 640’640   |
| Number of tasks            | 3–36      |
| Area                       | 6m’6m     |
| Goal location              | 3         |
| Velocity of robot          | 0.20m/s   |
| Robots in real experimentation | 3       |
| Robots in simulation       | 4         |

https://doi.org/10.1371/journal.pone.0267982.t002
Fig 7. Simulation Scenario (a) Initial Position (b) \textit{R}_4$ encounters a robotic arm failure and \textit{R}_3$ encounters an RGB failure (c) \textit{R}_3$ is selected by \textit{R}_4$ for its lowest bid for robotic arm sharing (d) \textit{R}_3$ completes the task which was detected by \textit{R}_4$.

https://doi.org/10.1371/journal.pone.0267982.g007
RGB failure. The robot $R_3$ detected the task with RGB and recognized it using the camera and image recognition capability and planned to take it to the goal location but did not have the gripper to transport it to the goal location. A request was generated by $R_3$ which was received by $R_2$ and $R_1$. Both the robots sent their bids for sharing their capabilities for the required task whereas the lowest bid was sent by $R_1$, which was declared as the winner.

**Validation and benchmarking**

The results are validated with benchmark Lee et al. [38], the task allocation with Deep Q-Learning incorporation Dai et al. [39], and Decentralized Task Allocation Ryck et al. [40]. We aim to minimize the total cost of executing tasks to complete more tasks with limited resources and task completion despite faults in robot’s capabilities, also decreasing the system’s communication burden. To achieve both purposes, the tunable parameter $\alpha$ was set at 0.5.
The cost is calculated in seconds considering the distance between the two points and the robot’s velocity \((s/v)\) and the time it needs to release its occupied resources. Random faults were generated in the robot’s capabilities which changed the net operational robots in the system, as in Eq (9). Also the Instantaneous Task-Robot ratio as given in Eq (10) is utilized to validate the proposed mechanism.

5. Results and discussion

5.1 Time consumed in completing all the tasks

The time required to complete the total tasks with different objectives is calculated as shown in Fig 10.

As per the user constraint or the task deadline requirement, the value of \(\alpha\) is updated. When the deadline is the main concern, the MiniMax objective is employed by incorporating the value of \(\alpha = 1\) or closer to 1. Hence, the robots that can share their resources with minimum time send their bids, reducing unnecessary computations in the system. When the time deadline is not a concern the value of \(\alpha = 0.5\) is employed which increases the overall number of task completion and the distance travelled by the robots. However, when the Task-Robot ratio was increased to 9, an exponential increase in time consumption was observed as shown in Figs 11 and 12.

The time required by the benchmark techniques of task allocation Ryck at al [40], Dai et al. [39], Lee et al [38] with the same objective value achievement is shown in Fig 11.

The proposed approach improved the overall time completion time while considering both objectives minimization. Although the task performed by the system is simply provided all the sensor measurements and the actuators are working ideally and there are no external disturbances. However, a small difference in the simulated and real experimentation was observed in target position estimations due to slight difference in movement of robot and transportation. The difference in simulated and real experimentation values for various task-robot ratios was 4.5–5% which can be observed in Figs 11 and 12. An exponential increase in total time was observed when the task-robot ratio was increased to 9 due to vigorous sharing of resources.

5.2 Resource utilization

Efficient resource utilization is an important parameter in evaluating a methodology. In the proposed mechanism, each experiment introduced an overall 5% resource failures or absence
of the required resources to analyze resource utilization. Around 15% higher resource utilization was observed when only the soft resources failures occurred in the system and there was no hard resource failure in the system. However, enhanced resource utilization was perceived in both types of resource failures compared to the already developed techniques, as shown in Figs 13 and 14.

Fig 11. Simulated results for total completion time for various tasks-robot ratios.

https://doi.org/10.1371/journal.pone.0267982.g011

Fig 12. Time required for various task-robot ratio with real experimentation.

https://doi.org/10.1371/journal.pone.0267982.g012
In the Resource-Based Task allocation Dai et al. [38], the robot’s resources were considered for task allocation however, the robots consumed most of the time in resource management, and the resource filament was rendezvous, whereas in the proposed mechanism, the deficient

Fig 13. Simulation results for resource utilization for various task-robot ratio.
https://doi.org/10.1371/journal.pone.0267982.g013

Fig 14. Real experimentation for resource utilization for various task-robot ratios.
https://doi.org/10.1371/journal.pone.0267982.g014
resources could be attained from an optimal location which increased the resource utilization and also the mission did not need to be reorganized.

Also, in Dai et al. [38] when a robot was partially damaged during task execution, the task was declared undone and needed to be reallocated to another robot the same applied to Ryck et al [40], Dai et al. [39]. Contrary to this in the proposed approach, when the robot was partially damaged, it could request for only the failed resource without re-announcing the entire task, provided the deficiency was not less than the AIV(i). The highest resource utilization of 96% was observed when the Task-Robot raised to 9 in simulation experimentation however during real experimentation there was a slight difference and the utilization reduced to 92%.

5.3 Communication overhead

Compared to other techniques, the proposed approach reduced the communication burden on the system, as the robots generated the requests only in case of resource absence and when the deficiency was under the roof of AIV(i).

A very slight difference (1–2%) in the simulated and real experimentation was observed in terms of communication overhead for each task -robot ratio in the system, as can be seen in Figs 15 and 16. The robot could not proceed further in case of resource failure Ryck et al [40], and the task needed to be re-announced and reallocated, whereas, in the proposed approach, a request was made only for the failed resources and only the robots having the required resource responded which reduced the communication burden on the entire system. Furthermore, in the task allocation proposed in Dai et al.[38] reduced communication burden; however it still had 16.7% more communication burden on the system as compared to the proposed approach, as shown in Fig 16.

![Fig 15. Simulation results for communication burden for various task-robot ratio.](https://doi.org/10.1371/journal.pone.0267982.g015)
Conclusion and future work

This paper presented an efficient Task Allocation algorithm that allowed the robots to adjust their behavior due to some fault occurrence, environmental changes, or other robot’s actions to increase the overall system performance. The resource sharing technique was incorporated in task allocation algorithm to compensate the resource shortage in certain robots. The robots accepted the tasks despite resource shortage provided the resource shortage was below the threshold level. Adaptive Incentive value, accumulated the information of entire system’s resources and the resource shortage notifications into a single numerical value which served as a criterion for task acceptance. Auctions were initiated with objective minimization, in case of resource shortage relevant to the accepted task, which reduced the overall completion time in case of resource shortage. The resource sharing using the dynamic global threshold approach and the objective minimization not only reduced the task execution time but also the search time for task. The resource utilization was increased due to sharing of the temporarily free resources of the robots along with this the centralized unit was aware of the available resources of the system The proposed approach was validated through simulations using the Pygame software. To further authenticate the proposed method real robot experimentation were rigorously performed.

The proposed approach can be very effective in many areas such as firefighting, surveillance, and patrolling. This approach is specially very beneficial in situations like the pandemic of COVID-19 when robots were deployed in hospitals and there was shortage of robots and their capabilities, and a minor fault in the robot capabilities could halt the entire process and the tasks could miss their deadline. Also a limited men power was available to fix the faults in robots. In such situations, the developed resource sharing can serve as the best solution to continue the tasks without disruption.
In the proposed approach, the path cost and the time required was assumed to be deterministic during the bidding phase. In the future, we will introduce the uncertainty and routing information in the bidding process because the time required to share the requested resource may be uncertain due to crowdedness or obstacles in the path. Further priorities would be introduced for different tasks to see their effect on the sharing mechanism.

Author Contributions

Conceptualization: Nayyer Fazal.
Data curation: Nayyer Fazal.
Investigation: Javaid Iqbal.
Methodology: Nayyer Fazal.
Project administration: Javaid Iqbal.
Software: Shahbaz Khan.
Supervision: Muhammad Tahir Khan.
Validation: Muhammad Tahir Khan.
Visualization: Shahzad Anwar.
Writing – review & editing: Shahzad Anwar, Shahbaz Khan.

References

1. Elizondo Leal J. C., Ramirez Torres J. G., Rodriguez Tello E., and Martinez Angulo J. R., “Multi-robot Exploration Using Self-Biddings under Constraints on Communication Range,” *IEEE Lat. Am. Trans.*, vol. 14, no. 2, pp. 971–982, Feb. 2016, https://doi.org/10.1109/TLA.2016.7437248
2. Rakshit P. et al., “Realization of an adaptive memetic algorithm using differential evolution and q-learning: A case study in multirobot path planning,” *IEEE Trans. Syst. Man, Cybern. Part A Systems Humans*, vol. 43, no. 4, pp. 814–831, 2013, https://doi.org/10.1109/TSMCA.2012.2226024
3. Lee D. H., Zaheer S. A., and Kim J. H., “Ad hoc network-based task allocation with resource-aware cost generation for multirobot systems,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 12, pp. 6871–6881, Dec. 2014, https://doi.org/10.1109/TIE.2014.2326987
4. Chang C. H., Wang S. C., and Wang C. C., “Exploiting Moving Objects: Multi-Robot Simultaneous Localization and Tracking,” *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 810–827, Apr. 2016, https://doi.org/10.1109/TASE.2015.2426203
5. Garcia Barrientos A., Lara Lopez J., Espinoza E. S., Hoyo J., and Valencia Palomo G., “Object Transportation Using a Cooperative Mobile Multi-Robot System,” *IEEE Lat. Am.,* vol. 14, no. 3, pp. 1184–1191, Mar. 2016, https://doi.org/10.1109/TLA.2016.7459597
6. Li H., Karray F., Basir O., and Song I., “A framework for coordinated control of multiagent systems and its applications,” *IEEE Trans. Syst. Man, Cybern. Part A Systems Humans*, vol. 38, no. 3, pp. 534–548, May 2008, https://doi.org/10.1109/TSMCA.2008.918591
7. Lee D. H., Zaheer S. A., and Kim J. H., “A Resource-Oriented, Decentralized Auction Algorithm for Multirobot Task Allocation,” *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 4, pp. 1469–1481, Oct. 2015, https://doi.org/10.1109/TASE.2014.2361334
8. Liu L. and Shell D. A., “Large-scale multi-robot task allocation via dynamic partitioning and distribution,” *Autow. Robots*, vol. 33, no. 3, pp. 291–307, Oct. 2012, https://doi.org/10.1007/s10452-012-9303-2
9. Wu D., Zeng G., Meng L., and Li L., “Gini coefficient-based task allocation for multi-robot systems with limited energy resources,” *IEEE/CAA J. Autom. Sin.*, vol. 5, no. 1, pp. 155–168, Jan. 2018, https://doi.org/10.1109/JAS.2017.7510385
10. Luo L., Chakraborty N., and Sycara K., “Provably-good distributed algorithm for constrained multi-robot task assignment for grouped tasks,” *IEEE Trans. Robot.*, vol. 31, no. 1, pp. 19–30, Feb. 2015, https://doi.org/10.1109/TRO.2014.2370831
11. Lu X., Wang L., Wang H., and Wang X., “Kalman filtering for delayed singular systems with multiplicative noise,” *IEEE/CAA J. Autom. Sin.*, vol. 3, no. 1, pp. 51–58, Jan. 2016, https://doi.org/10.1109/JAS. 2016.7373758

12. Jose K. and Prathar D. K., “Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods,” *Rob. Auton. Syst.*, vol. 80, pp. 34–42, Jun. 2016, https://doi.org/10.1016/j.robot.2016.02.003

13. K. F. E. Tsang, Y. Ni, C. F. R. Wong, and L. Shi, “A Novel Warehouse Multi-Robot Automation System with Semi-Complete and Computationally Efficient Path Planning and Adaptive Genetic Task Allocation Algorithms,” in *2018 15th International Conference on Control, Automation, Robotics and Vision, ICARCV 2018*, Dec. 2018, pp. 1671–1676, https://doi.org/10.1109/ICARCV.2018.8581092

14. Liu C. and Kroll A., “A centralized multi-robot task allocation for industrial plant inspection by using A∗ and genetic algorithms,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012, vol. 7268 LNAI, no. PART 2, pp. 466–474, https://doi.org/10.1007/978-3-642-29350-4_56

15. Nunes E., Manner M., Mitiche H., and Gini M., “A taxonomy for task allocation problems with temporal and ordering constraints,” *Rob. Auton. Syst.*, vol. 90, pp. 55–70, Apr. 2017, https://doi.org/10.1016/j. robot.2016.10.008

16. Sullivan N., Grainger S., and Cazzolato B., “Sequential single-item auction improvements for heterogeneous multi-robot routing,” *Rob. Auton. Syst.*, vol. 115, pp. 130–142, May 2019, https://doi.org/10.1016/j.robot.2019.02.016

17. F. Tang and L. E. Parker, “A complete methodology for generating multi-robot task solutions using ASyMTRe-D and market-based task allocation,” in *Proceedings—IEEE International Conference on Robotics and Automation, 2007*, pp. 3351–3358, https://doi.org/10.1109/ROBOT.2007.363990

18. Kharnis A., Hussein A., and Elmgoy A., “Multi-robot task allocation: A review of the state-of-the-art,” *Study. Comput. Intell.*, vol. 604, pp. 31–51, 2015, https://doi.org/10.1007/978-3-319-18299-5_2

19. Li X., Liu Z., and Tan F., “Multi-Robot Task Allocation Based on Cloud Ant Colony Algorithm,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Nov. 2017, vol. 10837 LNCS, pp. 3–10, https://doi.org/10.1007/978-3-319-70093-9_1

20. A. R. M. and Montan L., “Simulated annealing for multi-robot hierarchical task allocation with flexible constraints and objective functions,” 2006.

21. Elmaraghy H., Patel V., and Ben Abdallah I., “Scheduling of manufacturing systems under dual-resource constraints using genetic algorithms,” *J. Manuf. Syst.*, vol. 19, no. 3, pp. 186–201, Jan. 2000, https://doi.org/10.1016/s0278-6125(00)80011-4

22. De Ryck M., Versteyhe M., and Debrouwere F., “Automated guided vehicle systems, state-of-the-art control algorithms and techniques,” *Journal of Manufacturing Systems*, vol. 54. Elsevier B.V., pp. 152–173, Jan. 01, 2020, https://doi.org/10.1016/j.jmsy.2019.12.002

23. Bernardine Dias M., Zlot R., Kalra N., and Stentz A., “Market-based multirobot coordination: A survey and analysis,” *Proc. IEEE*, vol. 94, no. 7, pp. 1257–1270, 2006, https://doi.org/10.1109/JPROC.2006.876939

24. Cao Y., Yu W., Ren W., and Chen G., “An Overview of Recent Progress in the Study of Distributed Multi-Agent Coordination,” *IEEE Trans. Ind. INFORMATICS*, vol. 9, no. 1, p. 427, 2013, https://doi.org/10.1109/TII.2012.2219061

25. Bänziger T., Kunz A., and Wegener K., “Optimizing human–robot task allocation using a simulation tool based on standardized work descriptions,” *J. Intell. Manuf.*, vol. 2018 317, vol. 31, no. 7, pp. 1635–1648, Apr. 2018, https://doi.org/10.1007/S10845-018-1411-1

26. Sujit P. B., Georget J. M., and Beard R., “Multiple UAV task allocation using particle swarm optimization,” *AIAA Guid. Navig. Control Conf. Exhib.*, 2008, https://doi.org/10.2514/6.2008–6837

27. Nam C. and Shell D. A., “Assignment Algorithms for Modeling Resource Contention in Multirobot Task Allocation,” *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 889–900, Jul. 2015, https://doi.org/10.1109/TASE.2015.2415514

28. C. Wang, L. Cao, C. C.-I. T. on Systems, undefined Man, and undefined 2015, “Formalization and verification of group behavior interactions,” *ieeexplore.ieee.org*. Accessed: Jul. 16, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7047907/?casa_token=PjScLmzyDg8AAAAA:H3wIshL2TExFkkl+GwKFy807GHE_j-m—ZvEWDjcj_WUWgLYyooopplPH9kIcE99OZD._ke9Blb.

29. Dias M. B. and Stentz A., “A Free Market Architecture for Distributed Control of a Multirobot System,” Jan. 2000, https://doi.org/10.1109/R1.6550274.V1

30. Gerkey B. P. and Mataric M. J. M., “Sold!: Auction Methods for Multirobot Coordination,” *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, 2002, https://doi.org/10.1109/TRA.2002.803462
31. Smith R. G., “The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver,” *IEEE Trans. Comput.*, vol. C-29, no. 12, pp. 1104–1113, 1980, https://doi.org/10.1109/TC.1980.1675516

32. Schoenig A. and Pagnucco M., “Evaluating sequential single-item auctions for dynamic task allocation,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, vol. 6464 LNAI, pp. 506–515, https://doi.org/10.1007/978-3-642-17432-2_51

33. Farinelli A., Boscolo N., Zanotto E., and Pagello E., “Advanced approaches for multi-robot coordination in logistic scenarios,” *Rob. Auton. Syst.*, vol. 90, pp. 34–44, Apr. 2017, https://doi.org/10.1016/J.ROBOT.2016.08.010

34. S. K. C. T. M. L. V. M. D. Kempe1, “The Power of Sequential Single-Item Auctions for Agent Coordination,” *Proceedings of the AAAI on artificial Intelligence*, 2006. https://www.aaai.org/Library/AAAI/2006/aaai06-266.php (accessed Jun. 03, 2021).

35. Maskin E. S. and Riley J. G., “Auction Theory with private values,” *Am. Econ. Rev.*, vol. 75, no. 2, pp. 150–155, May 1985, https://doi.org/10.2307/1805587

36. Lagoudakis M. G. et al., “Auction-Based Multi-Robot Routing,” *Robic Sci. Syst.*, vol. 1, 2005.

37. “Traveling Salesman Problem: Theory and Applications—Google Books.” https://books.google.com.pk/books?hl=en&lr=&id=gKWdDwAAQBAJ&oi=fnd&pg=PA1&ots=a9ay-39cG1&sig=HMXdDlqVukB1f26V_dIf_2BsngGg&redir_esc=y#v=onepage&q&f=false (accessed Jul. 06, 2021).

38. Lee D. H., “Resource-based task allocation for multi-robot systems,” *Rob. Auton. Syst.*, vol. 103, pp. 151–161, 2018, https://doi.org/10.1016/j.robot.2018.02.016

39. Dai W., Lu H., Xiao J., Zeng Z., and Zheng Z., “Multi-Robot Dynamic Task Allocation for Exploration and Destruction,” https://doi.org/10.1007/s10846-019-01081-3

40. De Ryck M., Pissort D., Holvoet T., and Demeester E., “Decentral task allocation for industrial AGV-systems with resource constraints,” *J. Manuf. Syst.*, vol. 59, pp. 310–319, Apr. 2021, https://doi.org/10.1016/j.jmsy.2021.03.008