Matching-based Coalition Formation for Multi-robot Task Assignment Under Partial Uncertainty

Brenden Latham, Vladimir Ufimtsev
East Central University
Ada, Oklahoma

Ayan Dutta
University of North Florida
Jacksonville, Florida

Abstract

In this paper, we study the multi-robot coalition formation problem for instantaneous task allocation, where a group of robots needs to be allocated to a set of tasks to execute optimally. One robot might not be enough to complete a given task, so forming teams to complete these tasks becomes necessary. In many real-world scenarios, the robots might have noisy localization. Due to this, cost calculations for robot-to-task assignments become uncertain. However, a small amount of resources might be available to accurately localize a subset of these robots. To this end, we propose a bipartite graph matching-based task allocation strategy (centralized and distributed versions) that gracefully handles the uncertainty arising from cost calculations using an interval-based technique while leveraging the fact that a small number of robots might be localized on demand using an external system such as drones. We have tested the proposed technique in simulation. Results show that our approach is moderately fast – scales up to 100 robots and 50 tasks in 0.85 sec. (distributed solution) while gracefully handling partial uncertainty.

Introduction

In today's era of automation, robots are deployed in real-world situations to complete complex tasks. In many such practical applications, including object transportation in warehouses, environmental monitoring, search and rescue, and precision agriculture, among others, multiple robots need to cooperate toward a global common goal. One of the first steps of such coordination is coalition structure formation – given m tasks and n robots, divide n robots into m tasks such that a given criterion, e.g., value is optimized (Service and Adams 2011; Gerkey and Matarić 2004). In this paper, we specifically optimize the cost metric – the overall cost to form m such teams needs to be minimized. We study instantaneous allocation, where the problem is solved when the allocation is complete once and no new robot can arrive or no new task can be taken into account (Zhang and Parker 2013).

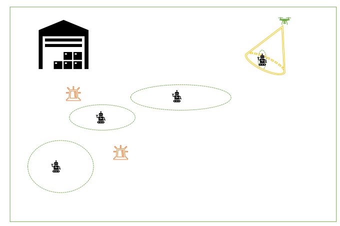


Figure 1: An illustration of the studied problem setup is presented. The probable ground robot locations are denoted with green shapes around the robots' ground-truth locations. The robot on the right is accurately localized by a drone. The tasks are represented with orange alarms. Now, the goal is to assign the robots to the tasks.

As the cost of forming such coalitions or teams depends on the distance that the robots need to travel from their current locations to the task locations, localization of the robots plays a significant role. However, in practical situations, such localization can be noisy. This might lead to uncertainty in the cost calculation, and as a result, the formed coalitions might not be of good quality (Yang and Chakraborty 2020). Although the multi-robot task allocation problem has received considerable attention and mostly game theoretic treatment in the past years, handling uncertainty remains a challenge. We model the solution as a coalition formation game where the distance from a robot to a task is estimated but it is uncertain due to localization noise, for example. In our problem setup, we assume that a small number of robots k $(< n)$ can be accurately localized using available limited resources. For example, k unmanned aerial vehicles with RF tag readers are available to be deployed to supplement k ground robots' on-board local and noisy localization technique (Buffi, Nepa, and Cioni 2017). In that case, the ground robots for task assignment can have RF tags on them that can be read by the drones to supplement them with accurate global localization (Buffi, Nepa, and Cioni 2017). The question then becomes how we choose k such robots that need to be prioritized for accurate localization.

To handle uncertainty in team costs, we take inspiration from a novel modification of the Hungarian algorithm proposed by Liu and Shell (Liu and Shell 2011). Note that these authors have studied the task assignment problem for single-task to single-robot assignment scenarios. Thus, they did not require the formation of coalitions. We, on the other hand, start by finding an initial set of m coalitions by not taking the uncertainty into account. Next, we modify these initial coalitions by integrating an interval-based mechanism. We also decide which k robots should be accurately localized, e.g., using an external system such as k drones. Fig. 1 shows an illustrative example.

We have implemented the proposed algorithms in simulation with up to 100 robots and 50 tasks in both centralized and asynchronous distributed settings. The main contributions of this paper are as follows.

- This approach outlines a novel way to quantify how critical a robot's accurate localization is to a multi-robot task assignment solution. These results are used to decide which robot to localize first with available resources in order to obtain a more optimal solution under uncertainty.
- Our presented algorithm has a realistic worst-case time complexity, is moderately fast and always outperforms an existing algorithm, namely OTMaM .

Related Work

Multi-robot task allocation is a very important problem to solve due to its practical relevance. In (Shehory and Kraus 1998), one of the first solutions to this problem was presented. Their solution provided a $(k+1)$-approximation where k is the highest number of members in a coalition. Similar to this, in our paper, we assume that the number of robots needed per coalition is bounded by k, which includes multiplicities. In (Service and Adams 2011), a bipartite matching-based coalition formation approach is proposed, which has a complexity of $O(n^{\frac{3}{2}}m)$ for the studied service model – which is similar to the setting of (Shehory and Kraus 1998) as well as ours. Along a similar path, Dutta and Asaithambi (Dutta and Asaithambi 2019) have also used a bipartite matching-based setting for task allocation. Greedy solutions are also popular for this problem. In (Zhang and Parker 2013), the authors propose two greedy heuristics for task allocation with heterogeneous robots. In (Czarnecki and Dutta 2021), the authors proposed a scalable task allocation solution by combining bipartite matching and a greedy heuristic. Distributed solutions possibly with heterogeneous robots can be found in (Dutta et al. 2021; Jang, Shin, and Tsourdos 2018; Mazdin and Rinner 2021). The authors in (Xue et al. 2019) have used the time window of the tasks and power consumption by the robots as the utility function. Recently, clustering similar robots to be assigned to tasks has been used as a solution approach for multi-robot task allocation (Dutta, Ufimtsev, and Asaithambi 2019; Martin et al. 2023). Simultaneous path planning and task allocation is also a relevant problem. The authors in (Yang and Chakraborty 2020) have considered uncertain robot-task assignment costs (e.g., due to unexpected obstacles on the path) for this. Unlike this, we take a graph-matching and interval-based approach to handle uncertainty. In (Liu and Shell 2011; Matarić, Sukhatme, and Østergaard 2003), the authors consider uncertainty albeit not for coalition formation. Our proposed approach in this paper combines ideas from (Dutta and Asaithambi 2019) and (Liu and Shell 2011) while presenting a centralized as well as a distributed solution to the studied problem.

Problem Definition and Notations

Robot and task model. Let $R = \{r_1, r_2, ..., r_n\}$ represent a set of n robots and every robot r_i is characterized by its location P_{r_i} and its unique ID. Each robot is able to localize using onboard sensing such as odometry, which is known to be noisy. They do not have more accurate localization sensors such as GPS.

Let $T = \{t_1, t_2, ..., t_m\}$ represent a set of m tasks where $n > m$. Similar to the robots, each task t_j is characterized by its location P_{t_j} as well as by the optimal number of robots required to execute it. This is denoted by the tuple $\langle P_{t_j}, O_j\rangle$. The value O_j for each task t_j is known beforehand and every robot has access to this information. We assume that $k < n$ of these robots can be accurately localized using an external system such as RF-tag reader drones.

A *coalition* $c \subseteq R$ is a subset of robots assigned to some task. Robots in the same coalition work in a coordinated manner to complete the assigned task. A *coalition structure* CS, is a non-overlapping partition of robots (a set of disjoint coalitions whose union is R). That is, $CS = \{c_1, c_2, ..., c_m\}, \forall c_i, c_j \in CS, i \neq j, c_i \cap c_j = \emptyset$, $\bigcup_{c_i \in CS} c_i = R$. Each coalition c_i is assigned to task t_i for $i = 1, 2, ..., m$. A coalition structure represents our solution to the problem. Let $\mathcal{C}$ denote the set of all such possible partitions (so $CS \in \mathcal{C}$). The size of $\mathcal{C}$ for n robots and m tasks is the Stirling set number of the second kind $S(n, m)$, i.e., it is the total number of non-overlapping partitions of an n set into exactly m nonempty clusters. The total number of partitions, if the number of clusters (tasks) is not taken into account, is the Bell number B_n, which grows exponentially.

Value Function. Initially, all robots are placed in an environment such as an agricultural field. Upon assignment to some task, a robot moves to the task location in order to complete it. Each robot spends some amount of energy while moving to a task. To account for this, we use a cost function, defined as $cost(r_i, t_j) = d(P_{r_i}, P_{t_j})$, where $d(P_{r_i}, P_{t_j})$ denotes the travel cost between the robot and task's locations. The value of a robot-task pair is $val(r_i, t_j) = MAX\_COST + 1 - cost(r_i, t_j)$. We define the value of a coalition to be the sum of the values incurred by all its member robots, i.e.,

$$val(c_i) = \sum_{r_j \in c_i} val(r_j, t_i). \quad (1)$$

Then the value of a coalition structure is the sum of the values of all the coalitions in it, i.e.,

$$val(CS) = \sum_{c_i \in CS} val(c_i). \quad (2)$$

Problem Objective. Given a set of n robots and m tasks with each task t_i requiring O_i robots to finish it, the objective is to find the coalition structure CS^* containing exactly m coalitions for m tasks where

$$CS^* = \arg\max_{CS \in \mathcal{C}} val(CS), \\ \text{s.t., } \forall i \in \{1, 2, ..., m\}, |c_i| = O_i. \quad (3)$$

Note that maximizing the value is equivalent to minimizing the cost of a coalition structure in this setting.

Algorithm 1: OTMaM Algorithm

Input: G_0: A bipartite graph
Output: A: An allocation.
$\mathcal{D} \leftarrow \emptyset$;
$S_v, S_u \leftarrow$ Adjacent nodes of $v \in V$ and $u \in U$ respectively
$A(v) \leftarrow \phi, \forall v \in V$
for *each* $v \in V$ **do**
  $u \leftarrow$ The best match of v
  if u *and* v *are mutually best* **then**
    $\mathcal{D} = \mathcal{D} \cup \{v, u\}$
    $A(v) = u$
while $\mathcal{D} \neq \emptyset$ **do**
  $u \leftarrow$ A node in U from an edge in $\mathcal{D}$
  $\mathcal{D} = \mathcal{D} \setminus \{v', u\}$ where $A(v') = u$
  for *each* $v \in S_u$ *where* $A(v) = \phi$ **do**
    if v *and* u' *are mutually best and* $|c_{u'}| < O_{u'}$ **then**
      $\mathcal{D} = \mathcal{D} \cup \{v, u'\}$
      $A(v) = u'$
return A

Probabilistic One-to-Many Matching

One-to-Many Matching. The foundation of our proposed approach is the One-to-many bipartite matching (OTMaM algorithm)-based task allocation technique (Dutta and Asaithambi 2019). Note that it does not consider the uncertainty in robot locations, and consequently, it cannot handle the uncertainty.

Given that the OTMaM is our foundation algorithm, we briefly explain its working procedure here (Algorithm 1). First, a bipartite graph $(G_0(\{V, U\}, E, W))$ is created with two sets of nodes – robots (R) and tasks (T) respectively. Edges (E) between them indicate their potential allocations with the edge weights (W) representing the values of pair-wise allocations $(val(r_i, t_j))$. The first phase of OTMaM algorithm is a one-to-one bipartite matching followed from (Manne and Bisseling 2007), where the dominating edges $(\mathcal{D})$ are kept in the matching. An edge $e(u, v) \in E$ is a dominating edge if for both u and v, edge $e(u, v)$ carries the maximum weight among all the edges originating from them. These dominating edges indicate that the robot-task pairs are mutually most advantageous to be allocated. Next, the remaining edges are processed until all the tasks have the required number of robots. The solution of this algorithm is a set of edges $S_0 \subseteq E$ representing the allocated task-robot pairs. It has been shown that OTMaM algorithm provides a low worst-case time complexity while guaranteeing a performance bound.

Our Proposed Algorithm

First, we discuss the centralized solution, which is computed by a leader robot. Let S_0 be the initial coalition structure found by Algorithm 1 on G_0. Under non-noisy cost settings, following (Dutta and Asaithambi 2019), we can state that S_0 is a near-optimal solution. For each edge $e_k \in E$ and $e_k \in S_0$ (meaning e_k is part of the solution), let G_k be a bipartite graph of the robots and tasks that do not include e_k. We run algorithm 1 on G_k and obtain a resultant solution S_k where $d_k = val(S_0) - val(S_k)$ is the maximum error allowed in e_k before S_0 is no longer the solution determined by the OTMaM algorithm. If $e_k \notin S_0$, we hide all adjacent edges in G_k instead of the edge itself.

We then can construct an interval I_k for each e_k in which the actual weight of e_k must lie in order to maintain S_0 as the solution determined when the algorithm is run on the system.

$$I_k = \begin{cases} [w_k - d_k, \infty) \ , \ e_k \in S_0 \\ (-\infty, w_k + d_k] \ , \ e_k \notin S_0 \end{cases} \quad (4)$$

Edges with large allowable errors, and therefore, large tolerance intervals, can have large fluctuations in their weight before S_0 can no longer be trusted as a solution. Similarly, edges with small allowable errors, and therefore, small tolerance intervals, cause the solution to be more sensitive to fluctuations in that edge's weight. We can then use the size of an edge's tolerance interval as a way to quantify the importance of the edge's actual weight to the original solution (Liu and Shell 2011). This approach, however, does not take into account the probability of each value in the tolerance interval. For this, we can use the probability density function for that edge. We then have a useful quantification of the importance of a robot's accurate localization and the corresponding edge's true weight to the solution taking into account its most probable weights. In other words, the criticality of an edge is equal to the probability of the true weight of the edge being a value that changes the solution given by our algorithm.

Now we have that the criticality of an edge's actual weight calculated from the true localization of a robot to the determined solution is

$$Y_k = \begin{cases} 1 - \int_{w_k - d_k}^{\infty} f(x) \ , \ e_k \in S_0 \\ 1 - \int_{-\infty}^{w_k + d_k} f(x) \ , \ e_k \notin S_0. \end{cases} \tag{5}$$

where f is a probability density function for the true weight of e_k. We, therefore, have a near-optimal solution for the system using estimated edge weights, and a value for each edge representing how critical the actual edge weight is to this solution. We can then for each node of robot r_i, sum the criticality of its outgoing edges to get a value L_i to determine how critical the exact location of robot r_i is to the solution. Next, we can sort the robots by their L values, and use available resources to localize the most critical k robots first. The pseudocode is presented in Algorithm 2.

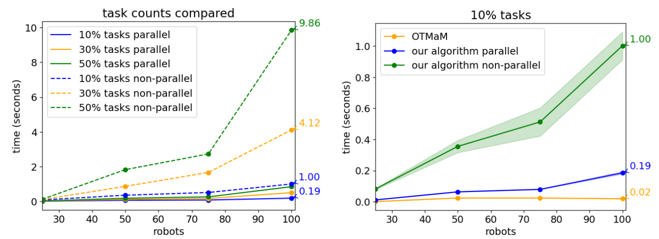
Distributed Version

In the distributed version of our algorithm, after the initial solution has been determined by some arbitrarily selected leader robot, it broadcasts the edge weight matrix and initial assignments to all other robots. Next, each robot computes its own local criticality by running steps 3-11 of Algorithm 2 on its adjacent edges. In this way, each robot computes its L value and reports back to the lead robot, which then sorts the robots by their L values.

Algorithm 2: Probabilistic One-to-many Bipartite Matching

```
1  Run Algorithm 1 on G_0 to determine the initial
     matching S_0
2  for e_k ∈ G_0 do
3  |  let G_k = G_0
4  |  if e_k ∈ S_0 then
5  |  |  Hide e_k in G_k
6  |  else
7  |  |  Hide all edges adjacent to r_i where r_i is
   |  |    on e_k
8  |  end
9  |  Run Algorithm 1 on G_k and get a new
   |    matching S_k
10 |  d_k = val(S_0) − val(S_k)
11 |  Calculate Y_k from Eq. 5
12 end
13 Calculate L_i from Y_k and sort robots by
     non-increasing L
14 Begin localizing robots in sorted order
15 Once all resources are used, run Algorithm 1 on
     G_0 with updated weights
```

Complexity Analysis. In (Dutta and Asaithambi 2019), it is shown that the worst-case time complexity of Algorithm 1 is $O(|E|)$. Algorithm 2 proceeds by first running Algorithm 1, then for each edge in the graph, Algorithm 1 is called once, resulting in $|E|$ calls to Algorithm 1 in total. This results in a worst-case time complexity of $O(|E|^2)$. All of the other control structures within the for loop in Algorithm 2 run in time that is less than $O(|E|)$ so the run time in each loop iteration is dominated by the call to Algorithm 1. In the distributed version of Algorithm 2, each robot calls Algorithm 1 one time per task, giving us a worst-case time complexity of $O(nm^2)$. This reduction in complexity by a factor of the number of agents n is substantial in that there is always a significantly greater number of agents than tasks in a given system.



(a) (b)

Figure 2: Time comparison among a) different task counts with our approach, and b) against OTMaM .

Experiments and Results

Settings. We tested with 10 to 100 robots and the count is varied between 10 and 50% of n. We assumed the robots and tasks to be uniformly distributed across an environment of size 100×100. The uncertainty is modeled as a Gaussian noise where the current robot locations are sampled from $\mathcal{N}(l, \sigma)$. l is the ground-truth location and σ is the standard deviation. Note that the optimal task requirements in terms of the number of robots were randomly generated such that the total number of required robots across all tasks was equal to n. We constructed a matrix of our edge weights and compared our solution's performance to the optimal solution. This helped us to report how close-to-optimal our found solutions are compared to OTMaM . We used the Message Passing Interface (MPI) as the distributed processing framework for the distributed version. For testing, we utilized the Titan supercomputer at the Advanced Research Computing and Analytics Center at Oral Roberts University. Titan is a supercomputer with 394 compute nodes and 4240 cores. Each robot is assigned a unique computing core similar to a real-world multi-robot setting, and the communication happens via one-to-one message passing asynchronously. We have assumed to have complete communication, i.e., each robot can communicate with every robot in R. Each test is run 10 times and the average and standard deviation results are presented here.

Note that when no robot is correctly localized, that would be equivalent to running OTMaM algorithm with the estimated edge weights, i.e., all the robots with positional uncertainty. On the other hand, when 100% robots are localized, that is equivalent to executing OTMaM algorithm without any uncertainty present in the environment. In the optimality plots, the blue and red lines indicate the results with these two baselines respectively whereas the black lines indicate our results. Next, our algorithm's performance metrics compared to OTMaM algorithm are presented.

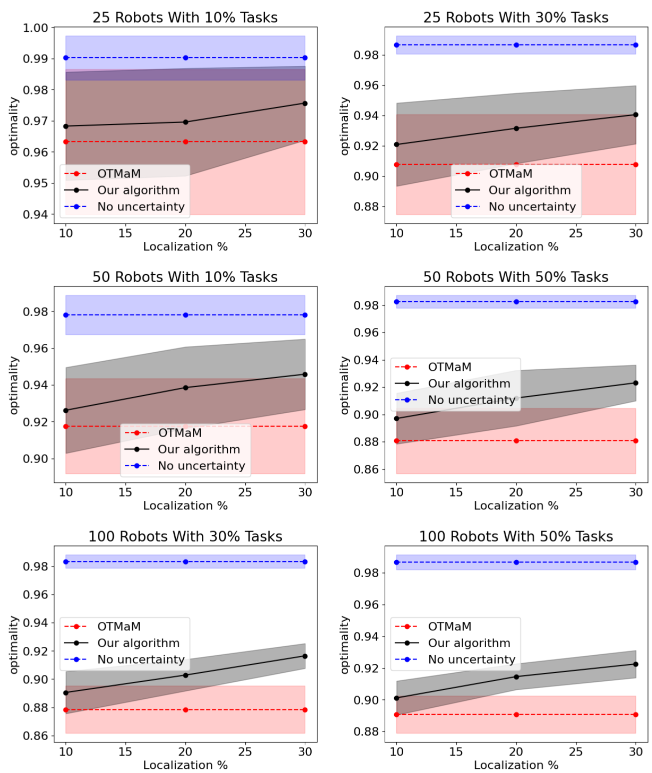


Figure 3: Optimality comparison among our algorithm, OTMaM , and the OTMaM with no uncertainty for various task and robot settings. σ is set to 30.

Results. First, we are interested in the run time of our proposed algorithm. Fig. 2. (a) shows the time results for various task and robot counts. As can be seen, with more tasks, the time requirements were higher. As the number of edges in G_0 was higher with more tasks, more processing of intervals and criticality calculations were required. Therefore, we find that as the task count increased, the execution time of the algorithm also increased. We also investigated the time difference between the executions of Algorithm 1 and our proposed solution. The time results with 10% tasks are presented in Fig. 2. (b). Given that Algorithm 1 is called multiple times to get our final solution, there is a considerable gap in terms of run time between OTMaM and our solution. As expected, the run time is reduced significantly by our distributed version – the maximum time is less than 1 sec. for $n = 100$ and $m = 50$.

Next, we turn our attention to the optimality results.

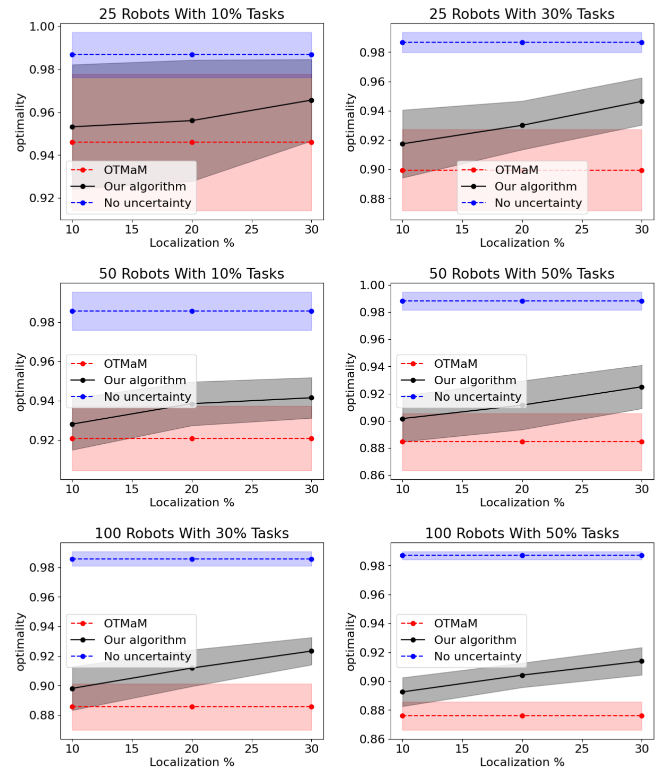


Figure 4: Optimality comparison among our algorithm, OTMaM , and the OTMaM with no uncertainty for various task and robot settings. σ is set to 50.

We have varied the value of σ between $\{30, 50, 70\}$, which controlled the uncertainty in the robot locations. On the other hand, as mentioned previously, we assumed to have limited resources to localize some of the robots perfectly. To emulate this, we varied the number of localized robots between $\{10, 20, 30\}$% of n that are most critical to the final value of the coalition structure (as discussed in Section 16). The results are presented in Figs. 3 – 5. Overall, our proposed technique performed 2.2% better in terms of achieving closer to optimal coalition structure than the baseline OTMaM algorithm in all the experiments. The statistics across all the test cases are provided in Table 1. In general, as there are many robots in a relatively small test area, we have noticed that the difference in optimality in our presented algorithm and OTMaM is limited whereas in a relatively sparse environment, such as shown in the working example, our algorithm almost always yields the optimal solution.

Algorithm	Max	Avg	Min
Our algorithm	**98.8**	**92.4**	**86.9**
OTMaM	97.9	90.2	84.3

Table 1: Statistics of optimality percentages across all test cases (better numbers are bolded).

When we look deeper into the numbers, we observe the following. When changing the uncertainty level σ, the optimality levels of each localization percentage remain similar. For instance, at $\sigma = 30$, we have that OTMaM gives us an average optimality of 90.29%, while $\sigma = 50$ gives 89.77% and $\sigma = 70$ gives 90.57%. The optimality of similar localization amounts also stays very close across different uncertainty levels, however as uncertainty increases there is a slight increase in the amount of optimality gained per number of robots localized using our algorithm. When $\sigma = 30$, we have that there is an average 1% increase in optimality for every 10% of robots localized. When $\sigma = 50$, we have 1.14% increase per 10% robots, and with $\sigma = 70$, we get 1.35% increase per 10% more localized robots. The rate of increase did not have a significant change across different robot counts for any uncertainty level. In every configuration, we noticed an average increase in optimality values as the number of localized robots grew. We find that even in uncertain environments with agent location approximations and a margin of error, OTMaM performs well, achieving an average of 90.2% and a minimum of 84.3% of the optimal. Therefore, it can be considered a proficient heuristic algorithm in such scenarios, while our approach strategically reduces the gap between the resulting CS value and the optimal value leveraging available resources. Statistical analysis indicates that fully localizing the top critical agents leads to a proportional reduction in the gap between the solution value and the true optimal across various configurations. See Table 2 for numerical details.

robots localized	mean gap reduction	p-value
10%	$> 10\%$	$< 8.7e^{-9}$
20%	$> 20\%$	$< .00011$
30%	$> 30\%$	$< .0025$

Table 2: Analysis of reduction in the gap from optimal.

Conclusion and Future Work

In this paper, we propose an instantaneous multi-robot task allocation strategy for a scenario where a subset of the robots can be localized with almost no positioning error (e.g., via an unmanned aerial vehicle) whereas the rest of the robots might have uncertainties in positioning. Our proposed algorithm combines the strengths of an existing task allocation technique that does not handle uncertainty and an interval-based solution for one-to-one robot-to-task mapping. By doing so while leveraging the available resources to localize a partial set of robots, we show that our proposed algorithm achieves closer to optimal solutions under uncertainty compared to an existing baseline algorithm. Furthermore, our proposed solution is easily scalable to a hundred robots and fifty tasks while incurring a moderate time cost. In the future, we are interested in investigating how uncertainty in communication, i.e., missing communication packets, affects the final solution.

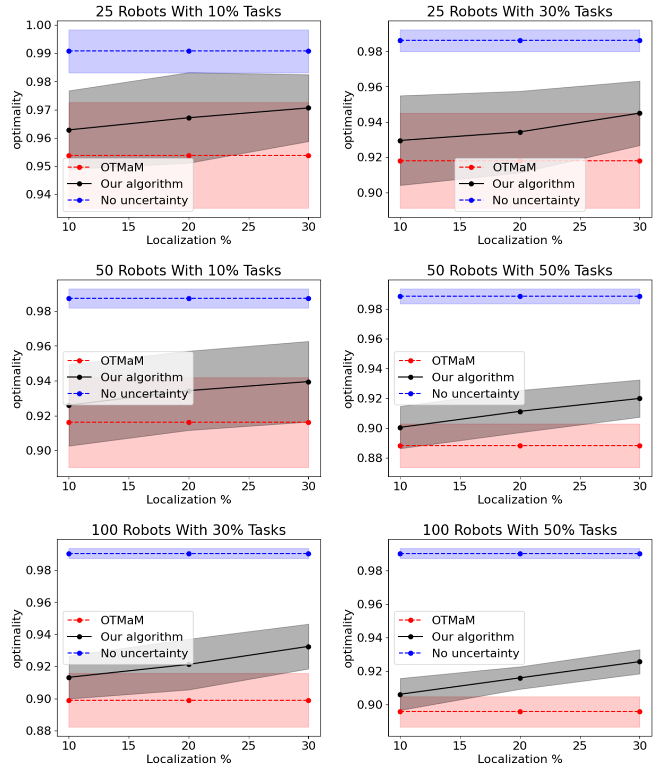


Figure 5: Optimality comparison among our algorithm, OTMaM , and the OTMaM with no uncertainty for various task and robot settings. σ is set to 70.

Acknowledgments

V. Ufimtsev and B. Latham are supported by NASA under the National Space Grant and Fellowship Pipeline Program through Oklahoma NASA Space Grant Consortium / EPSCoR.

References

Buffi, A.; Nepa, P.; and Cioni, R. 2017. Sarfid on drone: Drone-based uhf-rfid tag localization. In *2017 IEEE International Conference on RFID Technology & Application (RFID-TA)*, 40–44. IEEE.

Czarnecki, E., and Dutta, A. 2021. Scalable hedonic coalition formation for task allocation with heterogeneous robots. *Intelligent Service Robotics* 14(3):501–517.

Dutta, A., and Asaithambi, A. 2019. One-to-many bipartite matching based coalition formation for multi-robot task allocation. In *2019 International Conference on Robotics and Automation (ICRA)*, 2181–2187. IEEE.

Dutta, A.; Ufimtsev, V.; Said, T.; Jang, I.; and Eggen, R. 2021. Distributed hedonic coalition formation for multi-robot task allocation. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, 639–644. IEEE.

Dutta, A.; Ufimtsev, V.; and Asaithambi, A. 2019. Correlation clustering based coalition formation for multi-robot task allocation. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, 906–913.

Gerkey, B. P., and Matarić, M. J. 2004. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research* 23(9):939–954.

Jang, I.; Shin, H.-S.; and Tsourdos, A. 2018. Anonymous hedonic game for task allocation in a large-scale multiple agent system. *IEEE Transactions on Robotics* 34(6):1534–1548.

Liu, L., and Shell, D. A. 2011. Assessing optimal assignment under uncertainty: An interval-based algorithm. *The International Journal of Robotics Research* 30(7):936–953.

Manne, F., and Bisseling, R. H. 2007. A parallel approximation algorithm for the weighted maximum matching problem. In *International Conference on Parallel Processing and Applied Mathematics*, 708–717. Springer.

Martin, J. G.; Muros, F. J.; Maestre, J. M.; and Camacho, E. F. 2023. Multi-robot task allocation clustering based on game theory. *Robotics and Autonomous Systems* 161:104314.

Matarić, M. J.; Sukhatme, G. S.; and Østergaard, E. H. 2003. Multi-robot task allocation in uncertain environments. *Autonomous Robots* 14:255–263.

Mazdin, P., and Rinner, B. 2021. Distributed and communication-aware coalition formation and task assignment in multi-robot systems. *IEEE Access* 9:35088–35100.

Service, T. C., and Adams, J. A. 2011. Coalition formation for task allocation: Theory and algorithms. *Autonomous Agents and Multi-Agent Systems* 22(2):225–248.

Shehory, O., and Kraus, S. 1998. Methods for task allocation via agent coalition formation. *Artificial Intelligence* 101(1):165–200.

Xue, F.; Tang, H.; Su, Q.; and Li, T. 2019. Task allocation of intelligent warehouse picking system based on multi-robot coalition. *KSII Transactions on Internet & Information Systems* 13(7).

Yang, F., and Chakraborty, N. 2020. Chance constrained simultaneous path planning and task assignment for multiple robots with stochastic path costs. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 6661–6667. IEEE.

Zhang, Y., and Parker, L. E. 2013. Considering inter-task resource constraints in task allocation. *Autonomous Agents and Multi-Agent Systems* 26(3):389–419.