Distributed Task Assignment in Multi-Robot Systems based on Information Utility

Petra Mazdin[1], Michal Barciś[1], Hermann Hellwagner[1,2], and Bernhard Rinner[1,3]

Abstract— **Most multi-robot systems (MRS) require to coordinate the assignment of tasks to individual robots for efficient missions. Due to the dynamics, incomplete knowledge and changing requirements, the robots need to distribute their local state information within the MRS continuously during the mission. Since communication resources are limited and message transfers may be erroneous, the global state estimated by each robot may become inconsistent. This inconsistency may lead to degraded task assignment and mission performance. In this paper, we explore the effect and cost of communication and exploit information utility for online distributed task assignment. In particular, we model the usefulness of the transferred state information by its information utility and use it for controlling the distribution of local state information and for updating the global state. We compare our distributed, utility-based online task assignment with well-known centralized and auction-based methods and show how substantial reduction of communication effort still leads to successful mission completion. We demonstrate our approach in a wireless communication testbed using ROS2.**

I. Introduction

Multi-robot systems (MRS) are increasingly deployed in various applications including surveillance, first response, entertainment, and transportation [1]. Efficient coordination of robots, in particular the dynamic assignment of tasks to the robots based on their states and the mission requirements, could significantly increase the success rate and performance of the overall MRS. In general, three fundamental assignment strategies can be distinguished: (i) offline assignment, (ii) online centralized assignment, and (iii) online distributed assignment. The first strategy requires that all information about tasks and robot states are available prior to the mission. Such a requirement is unrealistic for many MRS applications. The second strategy assigns tasks during the mission using a central entity with all required information, while the third strategy assigns tasks by decentralized entities with potentially incomplete and inconsistent information. Both online task assignment strategies require communication for updating the state information.

In this paper, we explore the effect and cost of communication and exploit information utility for online distributed task assignment. Figure 1 depicts an example of our approach. In this coverage mission, robots move to spatially distributed locations, capture and process data, and send it to the base station for further processing. The information utility represents the temporal value of the usefulness of a particular message from the receiver's perspective and depends on the message type and the elapsed time. A highly relevant message type for task assignment is the (local) state information of the robots which needs to be regularly distributed in the MRS to maintain a consistent (global) state. However, communication resources are limited and data transfers are prone to failures in many real-world MRS applications, diminishing timeliness of data and global state consistency. We use information utility for controlling the transmission frequency of local state information and for updating the global state.

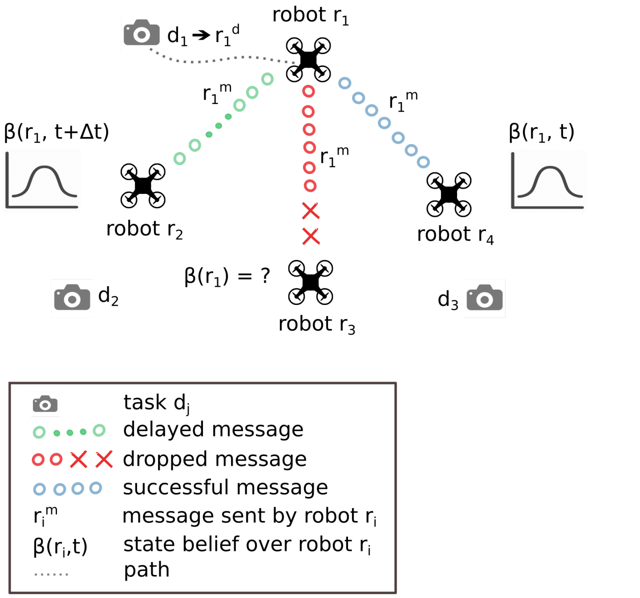


Fig. 1: Example of a coverage mission with information utility based task assignment. The robot r_i assigns its next coverage task d_j to execute based on the state belief β over all available robots and tasks. Robots communicate by sending messages which might get delayed or dropped. In this example, robot r_1 broadcasts its state via message r_1^m after selecting task d_1 for execution, while tasks d_2 and d_3 remain unassigned.

This article continues our previous work on coordination in MRS [2] but focuses on how to satisfy the mission requirements in the context of limited communication. We showed in [3] how large communication load can be when using specific coordination algorithms, and ideas to reduce it could significantly improve the mission quality. The contribution of this paper is fourfold. First, we propose a fully

[1] Karl Popper Kolleg on Networked Autonomous Aerial Vehicles, [2] Institute of Information Technology, [3] Institute of Networked and Embedded Systems, all University of Klagenfurt, Universitätsstraße 65-67, Klagenfurt, Austria firstname.surname@aau.at

distributed and dynamic task assignment strategy based on the information content and utility of messages. Second, we reduce communication load by estimating the data exchange frequency based on the relevance of exchanged messages. Furthermore, we compare our task assignment strategy with some prominent alternatives in this field, where our evaluation criteria are based on the mission quality in terms of task assignment success, time, and the communication load. Finally, we validate our algorithms via different experiments using ROS2 and a custom-built testbed.

II. Related Work

We briefly discuss two aspects of related work in MRS: communication awareness and task assignment.

A. Communication Awareness in MRS

Fowler et al. [4], [5] introduce the Intelligent Knowledge Distribution framework to answer the question "what information should be sent, to whom and when, with the limited resources available to each robot". In the work of Best et al. [6], [7] robots request information from other robots based on the belief uncertainty over the possible future action sequences of other robots. Williamson et al. [8], [9] present another approach where agents also apply a belief notion but with the difference of assessing the message content and deciding to communicate only if the deviation between the last communicated message belief and the current belief is large enough. Wu et al. [10] introduce a similar approach where agents communicate in order to maintain coordination whenever they detect an inconsistency in their shared belief. Sung et al. [11] address the problem of limited communication and limited time to share information among robots in multi-target tracking. Roth et al. [12] and Marcotte et al. [13] focus on the message content, in particular on "what to communicate". In addition, Amir et al. [14], [15] suggest a method to reduce information overload in a loosely-coupled teamwork. In our previous work [16] we have introduced a utility-based evaluation framework that assigns a *utility* value to each message. The utility depends on the message type and content, and it changes during the mission.

In this paper, we employ information distribution based on these utilities and show how they can support the mission by reducing the amount of exchanged information while still satisfying quality parameters. Our robots decide to communicate if the believed utility on the receiver's side is sufficiently large, and our receivers decide on the relevance of the message they received.

B. Task Assignment in MRS

There is a lot of work on task assignment in MRS, e.g., [17], [18]. However, task assignment becomes a dynamic decision problem when dynamic environments without explicit behavior models are considered. Some solutions to this problem comprise market-based approaches [19]–[22], game theoretical and machine learning approaches [23]–[26], optimization based approaches [27]–[29], and artificial potential fields [30]. Since these approaches rely on a (tight) coordination among robots, high quality communication is needed in order to successfully accomplish the mission. Otte et al. [22] show one example of an effect of lossy communication on different task assignment methods.

Our online task assignment is a fully distributed approach where robots, based on the current belief over the other robots' states and available tasks, assign the task that best satisfies the objective of the task assignment minimization problem at the decision time.

III. Problem Definition

Our approach deals with the dynamic assignment of tasks to robots. The problem is to assign every unassigned task to the available robots given the tasks' complexity, the robots' capabilities, the mission requirements, and the available information about the state of the robots and the environment. Once a task has been selected, the robot must move to the task's position and process the task upon arrival. In order to improve the assignment and hence the overall mission, each robot communicates its state.

We define three types of important entities with certain attributes: a set of q robots $R = \{r_1, \ldots, r_q\}$, a set of p tasks $D = \{d_1, \ldots, d_p\}$, and a set of o messages $M = \{m_1, \ldots, m_o\}$ sent by robots throughout the mission. With every message received, robots update their local state information belief $\beta = \{\hat{R}, \hat{D}\}$, a set of all available tasks $\hat{D}(t)$ and available robots $\hat{R}(t)$ at time t. Figure 2 depicts the three entities and attributes. The robot's status r_i^{s} indicates whether the robot is available to take on a new task (method *allocate(β)*) or is already executing the selected one by either moving towards it or processing it (method *execute_task(r_i^{d}))*. The attribute r_i^{pos} indicates the robot's physical position in the environment, r_i^{d} the currently selected task, r_i^{ψ} a set of received messages, and r_i^{m} the message containing information about the robot's current state. Messages are sent out by the *broadcast_msg(r_i^{m})* method. The task's attribute d_j^{s} indicates its current assignment state, d_j^{e} represents its processing time, and d_j^{pos} the location. The content of the message instance is stored in m_k^{c}, and the time the message was generated is stored in m_k^{gen}. We use one message type and hence define the same utility function for it. In order to define whether a message instance is useful at a specific time after its generation, we introduce the threshold κ. If the utility at that moment exceeds κ, the message is considered useful.

The task assignment problem can be described as an optimization problem. In particular, we address a dynamic assignment problem where we impose three criteria to be minimized for a specific utility threshold κ at time instance t: the aggregated task execution time $e(t, X, \kappa)$, the aggregated task assignment time $a(t, \kappa)$, and the total communication effort $o(t, \kappa)$. The first criterion $e(t, X, \kappa)$ aims at finding the optimal assignment matrix X (robot-task allocation) given β so that the total task execution time for all assigned tasks is minimized. The second criterion $a(t, \kappa)$ represents the sum of task assignment times for all tasks. The task assignment time is the time period from the moment a task becomes available

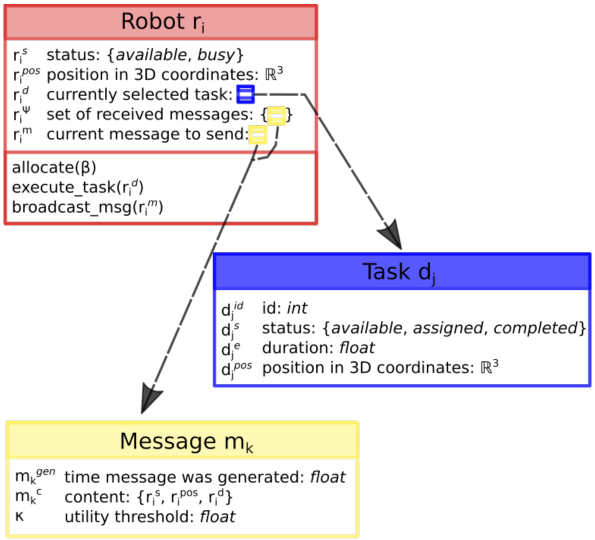

Robot r_i
r_i^s status: {*available*, *busy*}
r_i^{pos} position in 3D coordinates: $\mathbb{R}^3$
r_i^d currently selected task:
r_i^Ψ set of received messages: { }
r_i^m current message to send:
allocate(β)
execute_task(r_i^d)
broadcast_msg(r_i^m)
Task d_j
d_j^{id} id: *int*
d_j^s status: {*available*, *assigned*, *completed*}
d_j^e duration: *float*
d_j^{pos} position in 3D coordinates: $\mathbb{R}^3$
Message m_k
m_k^{gen} time message was generated: *float*
m_k^c content: {r_i^s, r_i^{pos}, r_i^d}
κ utility threshold: *float*

Fig. 2: Attributes of the entities: robot (top), task (center), and message (bottom). Robot's methods are shown as well.

until it is assigned. The third criterion $o(t,\kappa)$ represents the total communication effort and is expressed by the number of exchanged messages. We define dynamic task assignment in MRS as a multi-criteria optimization problem[4]:

$$\begin{aligned}
\min_{X,\kappa} \quad & (e(t,X,\kappa), a(t,\kappa), o(t,\kappa)) \\
\text{s.t.} \quad & e(t,X,\kappa) = \sum_{i\in R}\sum_{j\in D}\left(\frac{c_{ij}\left(t,\kappa\right)}{v} + d_j^{\mathrm{e}}\right) x_{ij}\left(t\right) \\
& c_{ij}(t,\kappa) = \begin{cases} ||r_i^{\mathrm{pos}}(t) - d_j^{\mathrm{pos}}|| & i \in \hat{R}(t,\kappa) \wedge j \in \hat{D}(t,\kappa) \\ \infty \quad \text{else} & \end{cases} \\
& x_{ij}(t) \in \{0,1\} \\
& \kappa \in [0,1]
\end{aligned} \tag{1}$$

where $e(t,X,\kappa)$ is determined by the cost of moving r_i to d_j ($\frac{c_{ij}(t,\kappa)}{v}$) and the cost of processing it (d_j^{e}). The average robot's speed is denoted by v and the assignment indicator $x_{ij}(t)$ is equal to 1 if the robot r_i is assigned to the task d_j, and 0 otherwise.

A. Information Utility

Motivated by our model for evaluation of information distribution schemes [16], we adopt the definition of information utility for deciding whether the message content is timely for any robot that successfully received the message. In our model, the utility depends on the message type and the time between generation and usage of the message. The greater this delay, the greater the uncertainty about the sender's state at the receiver.

We further define two important phases for robots when assigned to a task: *moving* and *processing*. The first phase represents moving to the task's location. The duration of this phase depends on the distance between the robot and the target position, the state of the environment including the position of other robots, path planning and the robot's motion capabilities. The robot transits to the processing phase when it has reached the task's location. The duration of this phase depends on the mission and typically changes due to variations in the environment and variations in low-level task executions. In our experiments, we model this duration by a random time value from a truncated normal distribution.

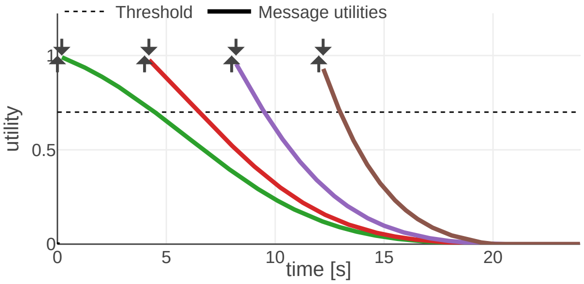


Fig. 3: Example graphs of expected information utility of four messages sent at time 0, 4, 8, and 12 seconds, and $\mathcal{P}$ distribution parameters $\mu = 6$ and $\sigma = 5$ for the interval $[0, 20]$. Arrows pointing up (down) represent the time a message was sent (received).

The information utility is related to the probability that the (other) robot is in a particular state given the information we have received so far [16]. We thus estimate the utility as the conditional probability

$$\mathcal{U}_k(m_k,t) = P\left(Q\left(t\right)|Q\left(m_k^{\mathrm{gen}}\right)\right), \tag{2}$$

where $Q(t)$ is the status of a robot (i.e., whether it is busy with a task) at time t. The equation represents a conditional probability of the robot having a particular status, given the known status $Q(m_k^{\mathrm{gen}})$ at the time m_k^{gen} when message m_k was generated. If the utility $\mathcal{U}_k(m_k,t)$ exceeds a threshold κ, the robots assumes that particular status.

In order to compute the utility, we assume that $\mathcal{P}$ is a distribution of task execution duration and $F_\mathcal{P}$ is a cumulative distribution function (CDF) of that distribution (which returns the probability of a task being finished before the given moment). Using $F_\mathcal{P}$, Equation 2 can be rewritten as:

$$\mathcal{U}_k(m_k,t) = 1 - \frac{F_\mathcal{P}(t) - F_\mathcal{P}(m_k^{\mathrm{gen}})}{1 - F_\mathcal{P}(m_k^{\mathrm{gen}})}. \tag{3}$$

Figure 3 shows example graphs of the information utility plotted from the reception time of the messages that were sent every four seconds. We model $\mathcal{P}$ as the sum of the estimated moving duration and the processing duration, where the processing duration is represented by a truncated normal distribution with expected mean μ, standard deviation σ, and interval $[a,b]$ as parameters.

IV. Approach

We propose a distributed online task assignment method for robots. Our method is robust to environment dynamics as

[4] For sake of simplicity, Equation 1 does not distinguish between balanced or unbalanced task assignment. However, Equation 1 can be easily extended to the appropriate case.

well as to the changes within the MRS as no robot represents a central entity. Algorithm 1 sketches the method for each robot r_i executed at an update rate ν. Whenever a new message is received, the robot updates its belief β. Since message reception is happening asynchronously with respect to the main method, belief update and message reception are left out of the pseudocode.

If the robot is available, it applies the Hungarian algorithm [31] and finds the optimal assignment w.r.t. the belief β of r_i (line 2). If a task has been assigned and its execution has started (line 5), the robot broadcasts this information (message r_i^{m}). As explained in Section III-A, the belief depends on the utility value of the message r_i^{m} when it was received and the threshold κ set for it. During the execution of the tasks, i.e., during the moving and the processing phases, the robot broadcasts its state if the *period* has passed (line 17). In particular, the sender estimates the utility on the receiver's side and computes the *period* for broadcasting the message (lines 7 and 18). This period is the elapsed time after which the utility will be smaller than κ, computed in the method $compute\_period(\mathcal{U}(r_i^{\mathrm{m}}, t), \kappa)$, reduced by the average message delay $\overline{\tau}(r_i^{\psi})$. The average delay considers all received messages and time it took to get them delivered from the point in time each message was generated. The robot checks if the same task is also assigned to another robot. If this conflict is detected, the robot with the lower id releases its task assignment and becomes available (lines 9 and 10). Finally, when the task is completed the robot becomes available again (lines 12 and 13).

Algorithm 1 Distributed online task assignment for robot r_i

```
every $\nu$ do:
 1: if $r_i$ $is\ available$ then
 2:     $r_i^{\mathrm{d}} \leftarrow allocate(\beta)$
 3:     if $r_i^{\mathrm{d}} \neq$ None then
 4:         $r_i^{\mathrm{s}} \leftarrow busy$
 5:         start to $execute\_task(r_i^{\mathrm{d}})$
 6:         $broadcast\_msg(r_i^{\mathrm{m}})$
 7:         $period \leftarrow compute\_period(\mathcal{U}(r_i^{\mathrm{m}}, t), \kappa) - \overline{\tau}(r_i^{\psi})$
 8: else
 9:     if $r_i^{\mathrm{d}}$ in conflict and $r_i$ has lower priority then
10:         $r_i^{\mathrm{s}} \leftarrow available$
11:     else
12:         if $r_i^{\mathrm{d}}$ is $completed$ then
13:             $r_i^{\mathrm{s}} \leftarrow available$
14:         else
15:             continue to $execute\_task(r_i^{\mathrm{d}})$
16:             if $period$ has passed then
17:                 $broadcast\_msg(r_i^{\mathrm{m}})$
18:                 $period \leftarrow compute\_period(\mathcal{U}(r_i^{\mathrm{m}}, t), \kappa) - \overline{\tau}(r_i^{\psi})$
```

V. Evaluation

A. Reference Approaches

We compare the performance of our task assignment approach with two related approaches tested with the same experimental setup. The first reference approach is a distributed auction-based allocation called MURDOCH [19]. The basic idea of this approach is as follows: If a robot is an auctioneer and receives a task, it assigns it to the best fit among the available robots. Since MURDOCH is a variant of the Contract Net Protocol (CNP), it comprises several steps which directly impose a communication overhead: task announcement, bid submission, result acknowledgment, and reception and progress acknowledgment. By analyzing the computational complexity for each task, we conclude that bidders simply have to perform $O(1)$ operations as they are responding to a task they received, whereas auctioneers have $O(q)$ operations to perform for all q bidding robots.

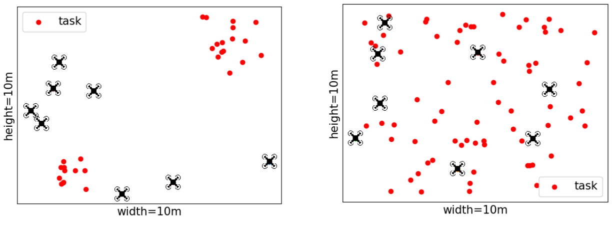
The second reference approach is the centralized Hungarian method [31] where robots communicate their states and actions to the central entity which then searches for the optimal allocation. This approach represents task assignment with consistent global knowledge and serves as a reference for measuring the effect of the knowledge gap between centralized and distributed assignment. Since we address the dynamic task assignment problem where tasks become available at different points in time, we assume $n \leq p$ available tasks that need to be assigned to q robots at time t. Then the centralized Hungarian method has a computational complexity $O(nq + q^2 \log q)$ if $n > q$, otherwise $O(nq + n^2 \log n)$. The communication complexity equals $O(nq)$.

Even though our approach has the same computation complexity as the centralized Hungarian algorithm, it has a significantly reduced communication overhead which equals $O(q)$, since robots only need to broadcast their state. On top of that, our approach is more resilient to communication losses and does not have a single point of failure.

B. Experiment Setup

We demonstrate the applicability of our task assignment approach with a multi-robot coverage mission from our previous work [2] where we simulate the robots' movement, data capturing and processing. Since we focus on assessing the effect of information utility in MRS, we have implemented the simulation with *ROS 2 Dashing Diademata* using *eProsima Fast RTPS* data distribution service (DDS). The experiments were performed using a custom-built testbed composed of eight *Raspberry Pi 3 Model B+* computers communicating with each other using an 802.11b network in ad-hoc mode. Even though our platform supports multi-hop connections using the *Babel* routing protocol, the robots operate in close proximity in our experiments, so they are always able to communicate directly with each other. The internal clocks of all agents are synchronized using Chrony and maintain the MRS clock discrepancy offset below 0.1 ms. Such a configuration allows us to utilize an actual network that mimics the communication properties of real MRS. At the same time it emulates the robots' physical properties in software, which allows us to reliably reproduce mission progress and focus on analyzing communication performance.

In our experiments tasks appear in an online fashion following a Poisson distribution with parameter λ specifying the expected number of occurrences. The robot's movement towards the assigned task is modeled as a linear movement with $v = 1\frac{m}{s}$. We distinguish between two scenarios for



(a) (b)

Fig. 4: Two scenarios for tasks' and initial robots' spatial distribution: Scenario A (4a) and Scenario B (4b).

the experiment setup as shown in Figure 4: Scenario A with $\lambda = 1$ (with 27 tasks in total) and a clustered spatial distribution, which resembles specific points of interest in a coverage mission, and Scenario B with $\lambda = 2$ (with 66 tasks in total) and a uniform spatial distribution. Both scenarios are placed on a $10\ m \times 10\ m$ area, with $q = 8$ robots, and with a time window of 30 s during which tasks appear following the given λ. Robots are able to exchange messages at an update rate ν of 0.1 s. Once a robot reaches the assigned task's location, it waits there for the time necessary to successfully process the task. This processing time depends on, e.g., dynamics or details in the captured images, and we model it with a truncated normal distribution with $\mu = 4,\ \sigma = 3$ in $[0, 8]$. Based on the sum of the moving and the processing time, we define the status message utility as described in Section III-A.

C. Results

In our experiments we compare the different assignment methods with a special focus on the quality of the underlying data distribution, i.e., by setting the message drop rate to 0, 0.2 or 0.4 for MURDOCH and our approach. We assume perfect communication with no drop rate for the centralized Hungarian approach. The drop rate represents the probability that a message from the sender will not be properly received by the receiver. For our approach we also vary the utility threshold ($\kappa \in \{0.2, 0.5, 0.8\}$) to assess changes in the belief of the system state. We run 50 simulations for each experiment and present the mean value of each result.

The achieved results are shown in Figure 5 where the left column depicts Scenario A and the right column Scenario B. We evaluate each approach in terms of (i) the overall mission time (Figure 5a and Figure 5b) which is related to the aggregated task execution time $e(t, X, \kappa)$ in Equation 1, (ii) the average time per task from the moment it appeared until it was assigned (Figure 5c and Figure 5d) representing the aggregated task assignment time $a(t, \kappa)$ in Equation 1, (iii) the total number of exchanged messages (Figure 5e and Figure 5f) controlled by the total communication effort $o(t, \kappa)$ in Equation 1, and (iv) the number of conflicts (tasks assigned to multiple robots) unresolved by the time robots reached the task (Figure 5g and Figure 5h).

D. Discussion

In both scenarios, the centralized Hungarian approach outperforms the other approaches in terms of the total mission time (Figure 5a and Figure 5b) and the average assignment time (Figure 5c and Figure 5d). Apart from the fact that the Hungarian approach assumes "perfect" communication and complete global state information, communication load (Figure 5e and Figure 5f) is much larger than in our approach, in particular with higher λ in Scenario B.

For $drop\ rate = 0$ MURDOCH outperforms our approach wrt. the total mission time (Figure 5a) and the average assignment time (Figure 5c) for Scenario A. The reason for this performance is that each auctioneer has an accurate knowledge about the robots' states to assign the tasks due to "perfect" communication and the waiting time for all acknowledgments is lower than in Scenario B due to a lower λ. When communication drops are introduced or more tasks needs to be assigned (higher λ), the performance of MURDOCH deteriorates. The large communication effort of MURDOCH is clearly visible in both Figure 5e and Figure 5f. Note that the bars of MURDOCH's communication effort are not plotted true-to-scale in Figure 5f to ease the comparison with the other methods. The true numbers of messages are printed next to the bars. Another drawback is a very high number of conflicts with other robots (Figure 5g and Figure 5h). As expected, there are no conflicts in all approaches when there are no communication drops.

In our approach, the threshold κ controls the assignment time, the number of conflicts, and the communication effort. A higher κ means that a robot has to broadcast messages more frequently in order to keep other robots informed about its state. If a robot is executing a task but the state information from the last received message has low utility, the other robots might incorrectly assume that the sender completed the task. This loss of information can possibly cause deficiencies in subsequent task allocation. On the other hand, if this robot completed the task and the utility of state information sent while processing the task is still larger than κ, a conflict might occur due to the mistaken assumption that the sender is still pursuing the task. Such conflicts can only be resolved by frequent state information updates in the moving phase of the task execution.

In both scenarios, choosing a higher κ results in a lower assignment time, as seen in Figure 5c and Figure 5d. On the other hand, the number of unresolved conflicts depends not only on κ but also on the spatial distribution of the tasks and their appearance rate λ. In general, a higher λ reduces the chances of conflicts simply because there are more tasks available. If the tasks have a wider spatial distribution, they are more distant from each other on average and robots have more time to resolve the conflict (if any) by sending the state updates while moving. A high utility threshold κ leads to an improved belief state, i.e., a higher likelihood that the predicted state matches the ground truth state, and hence less conflicts. With that said, one can observe that κ has rather limited influence on the number of unresolved conflicts in

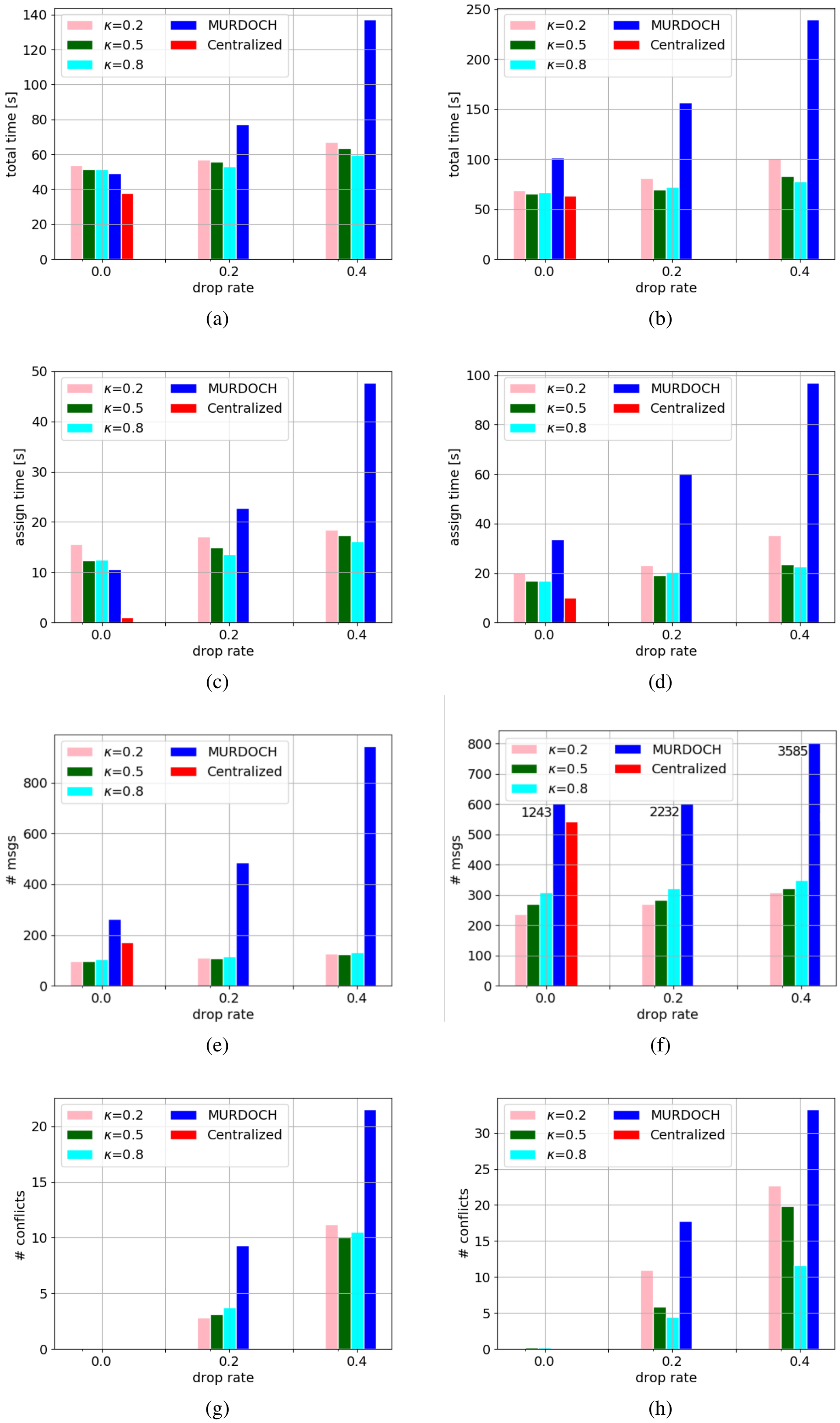


Fig. 5: Results for Scenario A (left) and Scenario B (right): total mission time (a) (b), average assignment time (c) (d), number of exchanged messages (e) (f), number of conflicts (g) (h).

Scenario A (see Figure 5g) due to two reasons. First, it is more likely that conflicts will occur at all in this mission due to a lower λ. Second, tasks are located very close to each other due to the clustered spatial distribution and therefore the robots have little time to resolve the potential conflict by sending state updates. On the other hand, tasks in Scenario B (see Figure 5h) are located with larger distances due to the uniform spatial distribution and there is a lower possibility that conflicts may happen at all due to higher λ. Therefore, the results for Scenario B show that increasing κ results in a lower number of conflicts.

VI. Conclusions

In this paper we have presented a distributed task assignment approach which tries to assign tasks among robots as soon and with as little communication effort as possible by using information utilities. We compared our distributed task assignment with two related approaches and demonstrated the decreased communication requirements in terms of transmitted messages. Furthermore, our approach keeps the effect of communication failures on the mission results low. By tuning the utility threshold κ we can give priority to either time or to communication load.

As future work we plan to (i) introduce utility to position messages when considering robot movement, (ii) adapt the approach to different multi-robot applications, e.g., where tasks require multiple robots for their execution, and (iii) test our approach on real drones performing coverage missions.

Acknowledgment

This work is supported by the Karl Popper Kolleg on Networked Autonomous Aerial Vehicles (KPK NAV) at the University of Klagenfurt, Austria.

References

[1] J. Scherer, S. Yahyanejad, S. Hayat, E. Yanmaz, T. Andre, A. Khan, V. Vukadinovic, C. Bettstetter, H. Hellwagner, and B. Rinner, "An Autonomous Multi-UAV System for Search and Rescue," in *Proc. Intern. Conf. Mobile Systems, Applications, and Services*, 2015, pp. 1–6.

[2] P. Mazdin and B. Rinner, "Coordination of Mobile Agents for Simultaneous Coverage," in *Proc. International Conference on Principles and Practice of Multi-Agent Systems*. Springer, 2019, pp. 170–185.

[3] ——, "Efficient and QoS-Aware Drone Coordination for Simultaneous Environment Coverage," in *Proc. IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2019, pp. 333–338.

[4] M. Fowler, P. Tokekar, T. C. Clancy, and R. K. Williams, "Constrained-Action POMDPs for Multi-Agent Intelligent Knowledge Distribution," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1–8.

[5] M. C. Fowler, T. C. Clancy, and R. K. Williams, "Intelligent Knowledge Distribution: Constrained-Action POMDPs for Resource-Aware Multi-Agent Communication," *arXiv*, pp. arXiv–1903, 2019.

[6] G. Best, M. Forrai, R. R. Mettu, and R. Fitch, "Planning-Aware Communication for Decentralised Multi-Robot Coordination," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1050–1057.

[7] G. Best, "Planning Algorithms for Multi-Robot Active Perception," Ph.D. dissertation, University of Sydney, 2019.

[8] S. Williamson, E. Gerding, and N. Jennings, "A Principled Information Valuation for Communications During Multi-Agent Coordination," in *Proc AAMAS Workshop on Multi-Agent Sequential Decision Making in Uncertain Domains*, 2008, pp. 137–151.

[9] S. A. Williamson, E. H. Gerding, and N. R. Jennings, "Reward Shaping for Valuing Communications During Multi-Agent Coordination," in *Proc. 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 2009, pp. 641–648.

[10] F. Wu, S. Zilberstein, and X. Chen, "Online planning for multi-agent systems with bounded communication," *Artificial Intelligence*, vol. 175, no. 2, pp. 487–511, 2011.

[11] Y. Sung, A. K. Budhiraja, R. K. Williams, and P. Tokekar, "Distributed Assignment with Limited Communication for Multi-Robot Multi-Target Tracking," *arXiv preprint arXiv:1812.11172*, 2018.

[12] M. Roth, R. Simmons, and M. Veloso, "What to Communicate? Execution-time Decision in Multi-agent POMDPs," in *Distributed Autonomous Robotic Systems 7*. Springer, 2006, pp. 177–186.

[13] R. J. Marcotte, X. Wang, D. Mehta, and E. Olson, "Optimizing multi-robot communication under bandwidth constraints," *Autonomous Robots*, vol. 44, p. 43–55, 2020.

[14] O. Amir, B. J. Grosz, and K. Z. Gajos, "Mutual Influence Potential Networks: Enabling Information Sharing in Loosely-Coupled Extended-Duration Teamwork," in *Proc. International Joint Conference on Artificial Intelligence (IJCAI'16)*, 2016, pp. 796–803.

[15] O. Amir, B. J. Grosz, K. Z. Gajos, and L. Gultchin, "Personalized change awareness: Reducing information overload in loosely-coupled teamwork," *Artificial Intelligence*, 2019.

[16] M. Barciś, A. Barciś, and H. Hellwagner, "Information Distribution in Multi-Robot Systems: Utility-Based Evaluation Model," *Sensors*, vol. 20, no. 3, pp. 1–28, 2020.

[17] M. Badreldin, A. Hussein, and A. Khamis, "A Comparative Study between Optimization and Market-Based Approaches to Multi-Robot Task Allocation," *Advances in Artificial Intelligence*, vol. 2013, pp. 1–12, 2013.

[18] A. Khamis, A. Hussein, and A. Elmogy, "Multi-robot Task Allocation: A Review of the State-of-the-Art," in *Cooperative Robots and Sensor Networks*. Springer, 2015, pp. 31–51.

[19] B. P. Gerkey and M. J. Mataric, "Sold!: Auction Methods for Multirobot Coordination," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 758–768, 2002.

[20] S. Koenig, P. Keskinocak, and C. Tovey, "Progress on Agent Coordination with Cooperative Auctions," in *Proc. Twenty-fourth AAAI Conference on Artificial Intelligence*, 2010, pp. 1713–1717.

[21] M. McIntire, E. Nunes, and M. Gini, "Iterated Multi-Robot Auctions for Precedence-Constrained Task Scheduling," in *Proc. International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2016, pp. 1078–1086.

[22] M. Otte, M. Kuhlman, and D. Sofge, "Multi-Robot Task Allocation with Auctions in Harsh Communication Environments," in *Proc. Inter. Symp. on Multi-Robot and Multi-Agent Systems*, 2017, pp. 32–39.

[23] A. Krause, A. Singh, and C. Guestrin, "Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies," *Journal of Machine Learning Research*, vol. 9, no. Feb, pp. 235–284, 2008.

[24] A. X. Jiang, A. D. Procaccia, Y. Qian, N. Shah, and M. Tambe, "Defender (Mis)coordination in Security Games," in *Proc. 23rd Intern. Joint Conference on Artificial Intelligence*, 2013, pp. 220–226.

[25] M. Tambe, *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge university press, 2011.

[26] C. Zhang and V. Lesser, "Coordinated Multi-Agent Reinforcement Learning in Networked Distributed POMDPs," in *Proc. Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011, pp. 764–770.

[27] S. Berman, Á. Halász, M. A. Hsieh, and V. Kumar, "Optimized Stochastic Policies for Task Allocation in Swarms of Robots," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 927–937, 2009.

[28] J. Capitan, M. T. Spaan, L. Merino, and A. Ollero, "Decentralized Multi-Robot Cooperation with Auctioned POMDPs," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 650–671, 2013.

[29] B. Fonooni, A. Jevtić, T. Hellström, and L.-E. Janlert, "Applying Ant Colony Optimization algorithms for high-level behavior learning and reproduction from demonstrations," *Robotics and Autonomous Systems*, vol. 65, pp. 24–39, 2015.

[30] S. Poduri and G. S. Sukhatme, "Constrained Coverage for Mobile Sensor Networks," in *Proc. IEEE International Conference on Robotics and Automation (ICRA04)*, 2004, pp. 165–171.

[31] H. W. Kuhn, "The Hungarian Method for the Assignment Problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.