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

Petra Mazdin\textsuperscript{1}, Michal Barcis\textsuperscript{2}, Hermann Hellwagner\textsuperscript{1,2}, and Bernhard Rinner\textsuperscript{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.

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

\textsuperscript{1}Karl Popper Kolleg on Networked Autonomous Aerial Vehicles, \textsuperscript{2}Institute of Information Technology, \textsuperscript{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 evalu-
ation 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 infor-
mation 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 communi-
cation 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 ex-
licit 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 $\beta = \{R, D\}$, a set of all available tasks
$D(t)$ and available robots $\hat{R}(t)$ at time $t$. Figure 2 depicts
the three entities and attributes. The robot’s status $r_i^a$ indicates
whether the robot is available to take on a new task (method
allocate($\beta$)) or is already executing the selected one by either
moving towards it or processing it (method execute_task($r_i^a$)).
The attribute $r_i^{pos}$ indicates the robot’s physical position
in the environment, $r_i^d$ the currently selected task, $r_i^m$ a
set of received messages, and $r_i^{\text{msg}}$ the message containing
information about the robot’s current state. Messages are sent
out by the broadcast_msg($r_i^{\text{msg}}$) method. The task’s attribute
$d_j^a$ indicates its current assignment state, $d_j^d$ represents its
processing time, and $d_j^{pos}$ the location. The content of the
message instance is stored in $m_k^i$, and the time the message
was generated is stored in $m_k^{\text{gen}}$. 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 $\kappa$. If the
utility at that moment exceeds $\kappa$, 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 $\kappa$ 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 $\beta$
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
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\(^4\):

\[
\begin{align*}
\min_{X, \kappa} & \quad e(t, X, \kappa), a(t, \kappa), o(t, \kappa) \\
\text{s.t.} & \quad e(t, X, \kappa) = \sum \sum \left( \frac{c_{ij}(t, \kappa)}{v} + d_{ij} \right) x_{ij}(t) \\
& \quad c_{ij}(t, \kappa) = \begin{cases} ||r_{ij}^\text{pos}(t) - d_{ij}^\text{pos}|| & i \in \hat{R}(t, \kappa) \land j \in \hat{D}(t, \kappa) \\ \infty & \text{else} \end{cases} \\
& \quad x_{ij}(t) \in \{0, 1\} \\
& \quad \kappa \in [0, 1]
\end{align*}
\]

where \( e(t, X, \kappa) \) is determined by the cost of moving \( r_i \) to \( d_j \) (\( c_{ij}(t, \kappa) \)) and the cost of processing it (\( d_{ij} \)). 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.

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

\[
U_k(m_k, t) = P(Q(t) | Q(m^\text{gen}_k)),
\]

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 \( Q(m^\text{gen}_k) \) at the time \( m^\text{gen}_k \) when message \( m_k \) was generated. If the utility \( U_k(m_k, t) \) exceeds a threshold \( \kappa \), 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:

\[
U_k(m_k, t) = 1 - \frac{F_{\mathcal{P}}(t) - F_{\mathcal{P}}(m^\text{gen}_k)}{1 - F_{\mathcal{P}}(m^\text{gen}_k)}.
\]

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 \( \mu \), standard deviation \( \sigma \), 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...
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 $\nu$. Whenever a new message is received, the robot updates its belief $\beta$. 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 $\beta$ 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 $\kappa$ 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 $\kappa$, computed in the method $\text{compute\_period}(U(r_i^{m}, t), \kappa)$, reduced by the average message delay $\tau(r_i^{m})$. 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 $r_i$

\begin{algorithm}
\begin{algorithmic}[1]
\Function{Distributed online task assignment for robot $r_i$}{\text{\,}}
\State $\nu$ \textbf{for} $r_i$
\If{$r_i$ \text{is available}}
\State $r_i^{d} \leftarrow \text{allocate}(\beta)$
\If{$r_i^{d} \neq \text{None}$}
\State $r_i^{t} \leftarrow \text{busy}$
\State $\text{start\_execute\_task}(r_i^{d})$
\State $\text{broadcast\_msg}(r_i^{m})$
\State $\text{period} \leftarrow \text{compute\_period}(U(r_i^{m}, t), \kappa) - \tau(r_i^{m})$
\Else
\If{$r_i^{d}$ \text{in conflict and $r_i$ \text{has lower priority}}}
\State $r_i^{d} \leftarrow \text{available}$
\Else
\If{$r_i^{d}$ \text{is completed}}
\State $r_i^{d} \leftarrow \text{available}$
\Else
\State $\text{continue\_execute\_task}(r_i^{d})$
\EndIf
\EndIf
\EndIf
\EndFunction
\end{algorithmic}
\end{algorithm}

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 $\lambda$ specifying the expected number of occurrences. The robot’s movement towards the assigned task is modeled as a linear movement with $v = \frac{1}{2\lambda}$. We distinguish between two scenarios for
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 $10 \times 10$ m area, with $q = 8$ robots, and with a time window of $30$ s during which tasks appear following the given $\lambda$. Robots are able to exchange messages at an update rate $\nu$ 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 $o(t, \kappa)$ 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 $\lambda$ 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 $\lambda$. When communication drops are introduced or more tasks need to be assigned (higher $\lambda$), 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 $\kappa$ controls the assignment time, the number of conflicts, and the communication effort. A higher $\kappa$ 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 $\kappa$, 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 $\kappa$ 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 $\kappa$ but also on the spatial distribution of the tasks and their appearance rate $\lambda$. In general, a higher $\lambda$ 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 $\kappa$ 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 $\kappa$ 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 $\lambda$. 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 $\lambda$. Therefore, the results for Scenario B show that increasing $\kappa$ 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 $\kappa$ 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


