

The Role of Communication in Coordination Protocols for Cooperative Robot Teams

Changyun Wei, Koen Hindriks and Catholijn M. Jonker
EEMCS, Delft University of Technology, Mekelweg 4, 2628 CD, Delft, The Netherlands

Keywords: Multi-Robot Coordination, Cooperative Teamwork, Performance.

Abstract: We investigate the role of communication in the coordination of cooperative robot teams and its impact on performance in search and retrieval tasks. We first discuss a baseline without communication and analyse various kinds of coordination strategies for exploration and exploitation. We then discuss how the robots construct a shared mental model by communicating *beliefs* and/or *goals* with one another, as well as the coordination protocols with regard to subtask allocation and destination selection. Moreover, we also study the influence of various factors on performance including the size of robot teams, the size of the environment that needs to be explored and ordering constraints on the team goal. We use the Blocks World for Teams as an abstract testbed for simulating such tasks, where the team goal of the robots is to search and retrieve a number of target blocks in an initially unknown environment. In our experiments we have studied two main variations: a variant where all blocks to be retrieved have the same color (no ordering constraints on the team goal) and a variant where blocks of various colors need to be retrieved in a particular order (with ordering constraints). Our findings show that communication increases performance but significantly more so for the second variant and that exchanging more messages does not always yield a better team performance.

1 INTRODUCTION

In many practical applications, robots are seldom stand-alone systems but need to cooperate and collaborate with one another. In this work, we focus on search and retrieval tasks, which have also been studied in the foraging robot domain (Cao et al., 1997; Campo and Dorigo, 2007; Krannich and Maehle, 2009). Foraging is a canonical task in studying multi-robot teamwork, in which the robots need to search targets of interest in the environment and then deliver them back to a home base. The use of multiple robots may yield significant performance gains compared to the performance of a single robot (Cao et al., 1997; Farinelli et al., 2004). But multiple robots may also lead to interference between teammates, which can decrease team performance. Therefore, it poses a challenge for a robot teams to develop effective coordination protocols for realising such performance gains.

We are in particular interested in the role of communication in coordination protocols and its impact on team performance. In previous work, e.g., (Balch and Arkin, 1994; Ulam and Balch, 2004), it has been reported that more complex communication strategies offer little benefit over more basic strategies.

The messages exchanged in (Balch and Arkin, 1994) among robots, however, are very simple, and they only studied a simple foraging task without ordering constraints on the targets to be collected. As no clear conclusion has been drawn on what kind of communication is most suitable for robot teams (Mohan and Ponnambalam, 2009), it is our aim to gain a better understanding of the impact of more advanced communication strategies where multiple robots can coordinate their behavior by exchanging their *beliefs* and/or *goals*. Here beliefs keep track of the current state of the environment, and goals keep track of the desired state of the environment of other robots. By communicating beliefs and/or goals, the robots can create a shared mental model to enhance team awareness. We are in particular interested in the multi-robot teams in which the robots can cooperate with each other without a central manager or any shared database, and they are fully autonomous and have their own decentralized decision making processes.

In this paper, we want to gain in particular a better understanding of the role of communication in the search and retrieval tasks *with and without ordering constraints* on the team goal. The first task without ordering constraints on the team goal is a sim-

ple foraging task that requires the robots to retrieve target blocks all of the same color, whereas the second task with ordering constraints on the team goal is a cooperative foraging task that requires the robots to retrieve blocks of various color in a particular order. In order to do so, we first analyse a baseline without communication and then proceed to analyse three different communication conditions where the robots exchange only beliefs, only goals, and both beliefs and goals. We use a simulation environment called the Blocks World for Teams (BW4T) introduced in (Johnson et al., 2009) as our testbed. In our experimental study, we analyse various performance measures such as time-to-complete, interference, duplication of effort, and number of messages and vary the size of the robot team and size of the environment.

The paper is organized as follows. Related work is discussed in Section 2. Section 3 introduces the search and retrieval tasks and the Blocks World for Teams simulator. In Section 4, we discuss the coordination protocols for the baseline without communication and for three communication cases. The experimental setup is presented in Section 5 and the results are discussed in Section 6. Section 7 concludes the paper and presents directions for future work.

2 RELATED WORK

Robot foraging tasks have been extensively studied and have in particular resulted in various bio-inspired, swarm-based approaches (Campo and Dorigo, 2007; Campo and Dorigo, 2007; Krannich and Maehle, 2009). In these approaches, typically, robots minimally interact with one another as in (Campo and Dorigo, 2007), and if they communicate explicitly, only basic information such as the locations of targets or their own locations are exchanged (Parker, 2008). Most of this work has studied the simple foraging task where the targets to be collected are not distinguished, so the team goal of the robots does not have ordering constraints. Another feature that distinguishes our setup from most related work on foraging tasks is that we use an office-like environment instead of the more usual open area with obstacles. Targets are randomly dispersed in the rooms of the environment, and the robots initially do not know which room has what kinds of targets. Our interest is to evaluate the contribution that explicit communication between robots can make on the time to complete foraging tasks, and to identify the role of communication in coordinating the more complicated foraging task in which the team goal has ordering constraints.

In order to enhance team awareness, we follow the

work of (Cannon-Bowers et al., 1993; Jonker et al., 2012), in which it is claimed that shared mental models can improve team performance, but it needs explicit communication among team members. Most of the current research, however, only has implicit communication for robot teams (Mohan and Ponnambalam, 2009). The work in (Balch and Arkin, 1994; Krannich and Maehle, 2009) studies communication in the simple foraging task without ordering constraints on the team goal and its impact on the completion time of the task. (Balch and Arkin, 1994) compares different communication conditions where robots do not communicate, communicate the main behavior that they are executing, and communicate their target locations. Roughly these conditions map with our no communication, communicating only beliefs, and communicating only goals, whereas we also study the case where both beliefs and goals are exchanged. A key task-related difference is that having multiple robots process the same targets speeds up completion of the task in (Balch and Arkin, 1994), whereas this is not so in our case. As a result, the use of communicated information is quite different as it makes sense to follow a robot or move directly to the same target location in (Balch and Arkin, 1994), whereas this is not true in our setting.

(Krannich and Maehle, 2009) studies the conditions where the robots can only exchange messages within certain communication ranges or in nest areas (i.e., the rooms in our case), whereas we do not study the constraints on the communication range; instead, we focus on the communication content. In this work, we assume that a robot can send messages to any of its teammates in the environment, and once a sender robot broadcasts a message, the receiver robots can receive the message without communication failures. By communicating beliefs and/or goals, decentralized robot teams can construct a shared mental model to keep track of the progress of their teamwork and other robots' intentions so as to execute significantly more complicated coordination protocols.

Several coordination strategies without explicit communication in foraging tasks have been studied in (Rosenfeld et al., 2008), which takes into account the avoidance of interference in scalable robot teams. Apart from the size of robot teams, the authors in (Rybski et al., 2008) consider the size of the environment. In our work, we also use scalable robot teams to perform foraging tasks in scalable environments in our experimental study. We consider a baseline in which the robots do not explicitly communicate with one another, but they can still apply various combinational strategies for exploration and exploitation in performing the foraging tasks. As robots

may easily interfere with each other without communication, these combinational coordination strategies in particular take account of the interference in multi-robot teams, and we carry out experiments to study which combinational strategy is the best one for the baseline case.

In this work, we assume that the robots only collide with each other when they want to occupy the same room at the same time in BW4T, and the robots can pass through each other in all the hallways. Once a robot has made a decision to move to a particular room, it can directly calculate the shortest path to that room. Thus, the multi-robot path planning problem is beyond the scope of this paper.

3 MULTI-ROBOT TEAMWORK

General multi-robot teamwork usually consists of multiple subtasks that need to be accomplished concurrently or in sequence. If a robot wants to achieve a specific subtask, it may first need to move to the right place where the subtask can be performed. An example of such teamwork is search and retrieval tasks, which are motivated by many piratical multi-robot applications such as large-scale search and rescue robots (Davids, 2002), deep-sea mining robots (Yuh, 2000), etc.

3.1 Search and Retrieval Tasks

Search and retrieval tasks have also been studied in the robot foraging domain, where the team goal of the robots is to search targets of interest in the environment and then deliver them to a home base. At the beginning of the entire task, the environment may be known, unknown or partially-known to the robots. Here the targets of interest correspond to the subtasks of general multi-robot teamwork, and if they can be delivered to the home base concurrently, then the team goal does not have ordering constraints; otherwise, all the needed targets must be collected in the right order.

In this work, the robots work in an initially unknown environment, so they do not know the locations of the targets at the beginning of the tasks. The robots have the map of the rough areas where the targets might be, but they have to explore these areas in order to find the exact dispersed targets. For instance, in the context of searching for and rescuing survivors in a village after an earthquake, even though the robots may have the map information of the village, they are hardly likely know the precise locations of the survivors when starting their work. Moreover, due to the limited carrying capability of robots, we

assume that a robot can only carry one target at one time in this work.

3.2 The Blocks World for Teams

We simulate the search and retrieval tasks using the Blocks World for Teams (BW4T¹) simulator, which is an extension of the classic single agent Blocks World problem. The BW4T has office-like environments consisting of *rooms* in which colored *blocks* are randomly distributed for each simulation (see Figure 1). One or more robots are supposed to search, locate, and retrieve the required blocks from rooms and return them to a so-called *drop-zone*.

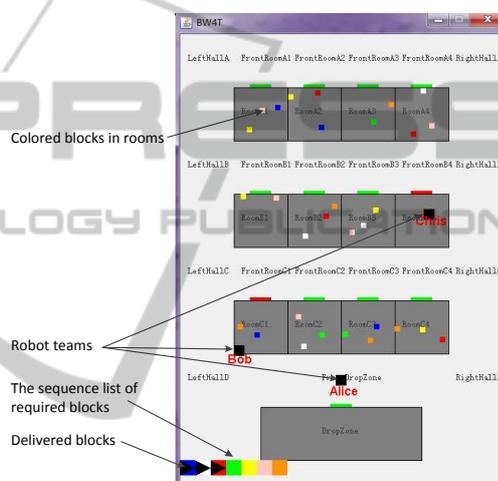


Figure 1: The Blocks World for Teams Simulator.

As indicated at the bottom of the simulator in Figure 1, required blocks need to be returned in a specific order. If all the required blocks have the same color, then the team goal of the task does not have ordering constraints. Access to rooms is limited in the BW4T, and at any time at most one robot can be present in a room or the drop-zone. Robots, moreover, can only carry one block at a time. The robots have the information about the locations of the rooms, but they do not initially know which blocks are present in which rooms. This knowledge is obtained for a particular room by a robot when it visits that room. Each robot, moreover, is informed of the complete required blocks and its teammates at the start of a simulation.

Robots in BW4T can be controlled by agents written in GOAL (Hindriks, 2013), the agent programming language that we have used for implementing and evaluating the team coordination strategies dis-

¹BW4T has been integrated into the agent environments in GOAL (Hindriks, 2013), which can be downloaded from <http://ii.tudelft.nl/trac/goal>.

cussed in this paper. GOAL also facilitates communication among the agents.

While interacting with the BW4T environment, each robot gets various percepts that allow it to keep track of the current environment state. Whenever a robot arrives at a place, in a room, or near a block, it will receive a corresponding percept, i.e., `at(PlaceID)`, `in(RoomID)` or `atBlock(BlockID)`. A robot also receives percepts about its state of movement (traveling, arrived, or collided), and, if so, which block the robot is holding. Blocks are identified by a unique ID and a robot in a room can perceive which blocks of what color are in that room by receiving percepts of the form `color(BlockID,ColorID)`. In BW4T, whenever the currently needed block is put down at the drop-zone, all the robots get a percept from the environment informing them about the color of the next needed block.

4 COORDINATION PROTOCOLS

A coordination protocol for search and retrieval tasks consists of three main strategies: deployment, subtask allocation and destination selection strategies. The *deployment* strategy determines the starting positions of the robots. In our settings, all robots start in front of the drop-zone, so we will not further discuss this strategy in our coordination protocols. The *subtask allocation* strategy determines which target blocks the robots should aim for. As a decentralized robot has its own decision making process, it allocates itself a target based on its current mental states of the world. Once a robot has decided to retrieve a particular target, it needs to choose a room to move towards so that it can get such a target. The *destination selection* strategy determines which rooms the robots should move towards, consisting of *exploration* and *exploitation* sub-strategies that are used for exploring the environment and exploiting the knowledge obtained during the execution of the entire task.

We will first investigate a baseline without any communication and set it as the performance standard that we want to improve upon by adding various communication strategies. And then we are particularly interested in whether, and, if so, how much, performance gain can be realised by communicating only beliefs, only goals, and both beliefs and goals.

4.1 Baseline: No Communication

In the baseline, although the robots do not explicitly communicate with one another, they can still obtain some information about their teamwork because the

robots may interfere with each other in their shared workspace. Without communication, for the subtask allocation all the robots will aim for the currently needed block until it is delivered to the drop-zone. But for destination selection, the exploration and exploitation sub-strategies can ensure that they will not visit rooms more often than needed (as far as possible), and basically that knowledge is exploited whenever the opportunity arises (i.e., a robot is greedy and will start collecting a known block that is the closest one and has the needed color).

In the baseline, we have identified four dimensions of variation:

1. which room a robot initially selects to visit,
2. how a robot will use knowledge obtained about another robot through interference,
3. how it selects a (next) room to visit, and
4. what a robot will do when holding a block that is not needed now but is needed later.

4.1.1 Initial Room Selection

At the beginning of the task, a robot has to choose a room to explore since it does not have any information about the dispersed blocks in the environment. One possible option labeled (1a) is to choose a *random* room without considering any distance information. Assuming that k robots initially select a room to visit from n available rooms, the probability that each robot chooses a different room to visit is P , and the probability that collisions may occur in the team is $P_c = 1 - P$. Then we can know:

$$P = \begin{cases} \frac{n!}{(n-k)! \cdot n^k} & , k \leq n, \\ 0 & , k > n. \end{cases} \quad (1)$$

This gives, for instance, a probability of 9.38% that 4 robots select different initial rooms from 4 available rooms, which drops to only 1.81% for 8 robots performing in the environment with 10 rooms. Working as a team, robots are expected to have as few collisions as possible, and we use P_c to reduce the likelihood that robots may collide with each other. For example, suppose we want the likelihood of collisions for $k = 4$ robots to be less than 5%, then we need $n > 119$. This tells us that it is virtually impossible to avoid collisions without communication in large-scale robot teams as a very large number of rooms would be required then.

A second option labeled (1b) is to choose the *nearest* room, which means that the robot will take account of the distance from its current location to the room's location. In this case, almost all robots will choose the same initial room given that they all start

exploring from more or less the same location according to the deployment strategy in our settings.

4.1.2 Visited by Teammates

Another issue concerns how a robot should use the knowledge about a collision with its teammates. A collision occurs when a robot is trying to enter a room but fails to do so because the room is already occupied by one of its teammates. The first option labeled (2a) is to *ignore* this information. That simply means that the fact that the room is currently being visited by the teammate has no effect on the behavior of the robot.

The second option labeled (2b) is to take this information into account. The idea is to exploit the fact that the robot, even though it still does not know what blocks are in the room, believes that the team knows what is inside the room. Intuitively, there is no urgent need anymore to visit this room therefore. The robot thus will delay visiting this particular room and assign a higher priority to visiting other rooms. Only if there is nothing more useful to do, a robot then would consider visiting this room again.

4.1.3 Next Room Selection

If a robot does not find a block it needs in a room, it has to decide which room to explore next. The available options for this problem are very similar to those for the initial room selection but the situation may actually be quite different as the robots will have moved and most of the time will not locate at more or less the same position anymore. In addition, some rooms have already been visited, which means there are less options available to a robot to select from in this case.

One option labeled (3a) is to *randomly* choose a room from the rooms that have not yet been visited, and a second one labeled (3b) is to visit the room *nearest* to the robot's current position. It is not upfront clear which strategy will perform better. If the robots very systematically visit rooms, because they all start from the same location, this will most likely increase interference. The issue is somewhat similar to the initial room selection problem as it is not clear whether it is best to minimize distance traveled (i.e., choose the nearest room) or to minimize interference (i.e., choose a random room).

4.1.4 Holding a Block Needed Later

When the robots are required to collect blocks of various color with ordering constraints, this issue concerns what to do when a robot is holding a block that is not needed now but is needed later. For instance, robot Alice is delivering a red block to the

drop-zone because it believes that the current needed target should be a red one. If robot Bob completes the subtask of retrieving a red block before Alice moves to the drop-zone, and the remaining required targets still need a red block in the future, then Alice comes to this situation.

One option labeled (4a) is to *wait in front of the drop-zone*, and then enter the drop-zone and drop the block when it is needed. The waiting time depends on how long it will take before the block that the robot is holding will be needed. A second option labeled (4b) is to *drop the block in the nearest room*. Since the waiting time in the first option is uncertain, it might be better to store the block in a room where it can be picked up again later if needed and invest time now rather in retrieving blocks that are needed now.

In the baseline case, each dimension discussed above has two options, so we can at most have 16 combinational strategies, some of which can be eliminated for the experimental study (see Section 5). We will investigate the best combinational strategy of the baseline, and then we take it as the performance standard to compare with the communication cases.

4.2 Communicating Robot Teams

In decentralized robot teams, there are no central manager or any shared database, for example, in distributed robot teams, so the robots have to explicitly exchange messages to keep track of the progress of their teamwork. In the communication cases, we mainly focus on the communication content in terms of beliefs and goals, and the robots use those shared information enhance team awareness. Since the robots can be better informed about their teammates in comparison with the baseline case, they can have more sophisticated coordination protocols concerning subtask allocation and destination selection.

4.2.1 Constructing Shared Mental Models

By communicating beliefs with one another, robots can be informed about *what other robots have observed in the environment and where they are*. Messages about beliefs are differentiated by the indicative operator ":" from those about goals, whose type is indicated by the imperative operator "!" in GOAL agent programming language. In this work, the robots exchange the following messages in respect of beliefs with associated meaning listed:

- :block(BlockID, ColorID, RoomID) means block BlockID of color ColorID is in room RoomID,

- `:holding(BlockID)` means the message sender robot is holding block `BlockID`,
- `:in(RoomID)` means the message sender robot is in room `RoomID`, and
- `:at('DropZone')` means the message sender robot is at the drop-zone.

Each of the messages listed above can also be negated to represent. For example, when a robot leaves a room, it will inform the other robots that it is not in the room using the negated message `:not(in(RoomID))`. Upon receiving a negated message, a robot will remove the corresponding belief from its belief base. A robot sends the first message to its teammates when entering room `RoomID` and getting the percept of `color(BlockID,ColorID)`. Note that only this message does not implicitly refer to the sending robot, and therefore except for the first message type, a robot who receives a message will also associate the name of the sender with the message. For instance, if robot Chris receives a message `in('RoomC1')` from robot Bob, Chris will insert `in('Bob','RoomC1')` into its belief base (see Figure 2).

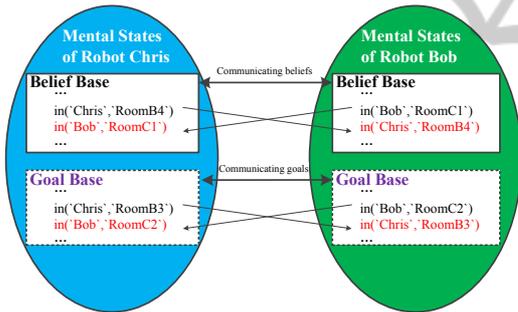


Figure 2: Constructing a shared mental model via communicating beliefs and/or goals.

By communicating goals with one another, robots can be informed about *what other robots are planning to do*. Robots exchange the following messages in respect of goals with associated meaning listed:

- `!holding(BlockID)` means the message sender robot wants to hold block `BlockID`,
- `!in(RoomID)` means the message sender robot wants to be in room `RoomID`, and
- `!at('DropZone')` means the message sender robot wants to be at the drop-zone.

Negated versions of goal messages indicate that previously communicated goals are no longer pursued and have been dropped by the sender. All goal messages received are associated with the sender who sent the message and stored or removed as expected (e.g., see Figure 2).

As we assume that the robots are cooperative, and they communicate with each other truthfully, the received messages can be used to update their own beliefs and goals. Algorithm 1 shows the main decision making process of an individual robot: how the robot updates its own mental states and at the same time shares them with its teammates for constructing a shared mental model. In its decision making process, the robot first handles new environmental percepts (line 2-5), and then uses received messages to update its own mental states (line 6-8). Based on the updated mental states, the robot can decide whether some dated goals should be dropped (line 9-12) and whether, and, if so, what, new goals can be adopted to execute (line 13-16). When the robot has obtained new percepts about its environment and itself or has adopted or dropped a goal, it will also inform its teammates (line 4, 11 and 15).

Algorithm 1: Main decision making process of an individual robot.

```

1: while entire task has not been finished do
2:   if new percepts then
3:     update own belief base, and
4:     send message “:percepts”.
5:   end if
6:   if receive new messages then
7:     update own belief base and goal base.
8:   end if
9:   if some goals are dated and useless then
10:    drop them from own goal base, and
11:    send messages “!goals”.
12:  end if
13:  if new goals are applicable then
14:    adopt them in own goal base, and
15:    send messages “!goals”.
16:  end if
17:  execute actions.
18: end while
    
```

As shown in Figure 2, when decentralized robot teams construct such a shared mental model via communicating belief and goal messages discussed above, even though they do not have any shared database or centralised manager, they could fully know each other like what they can know in a centralised robot team. Therefore, such a shared mental model can enable the robots to have more sophisticated coordination protocols.

4.2.2 Subtask Allocation and Destination Selection Strategies

By just communicating belief messages, there is quite a bit of potential to avoid interference since a robot

will inform its teammates when entering a room. More interestingly, it is even possible to avoid duplication of effort because a robot can also obtain what block should be picked up next from the information about the blocks that are being delivered by teammates. Robot use the shared beliefs to coordinate their activities for subtask allocation (see item 3 and 4) and destination selection (see item 1 and 2) as follows:

1. A robot will not visit a room for exploration purposes anymore if it has already been explored by a team member;
2. A robot will not adopt a goal to go to a room (or the drop-zone) that is currently occupied by another robot;
3. A robot will not adopt a goal to hold a block if another robot is already holding that block (which may occur when another robot beats the first robot to it);
4. A robot will infer which of the blocks that are required are already being delivered from the information about the blocks that its teammates are holding and will use this to adopt a goal to collect the next block that is not yet picked up and is still to be delivered.

By just communicating goal messages, the robots can also coordinate the activities to avoid, as in the case of sharing only beliefs, interference and duplication of effort. Whereas it is clear that a robot should not want to hold a block that is being held by another robot, it is not clear per se that a robot should not want to hold a block if another already wants to hold it. The focus of our work reported here, however, is not on negotiating options between robots. Instead, we have used a “first-come first-serve” policy here, and the shared goals can be used for subtask allocation (see item 2 and 3) and destination selection (see item 1) as follows:

1. A robot will not adopt a goal to go to a room (or the drop-zone) that another robot already wants to be in,
2. A robot will not adopt a goal to hold a block that another robot already wants to hold, and
3. A robot will infer which of the blocks that are required will be delivered from the information about goals to hold a block from its teammates and will use this to adopt a goal to collect the next block that is not yet planned to be picked up and is still to be delivered.

It should be noted that the difference between item 3 listed above for the use of goal messages and that of item 4 listed above for the use of belief messages is rather subtle. Whereas the information about goals is

an indication of what is planned to happen in the near future, the information about beliefs represents what is going on right now. It will turn out that the potential additional gain that can be achieved from the third rule for goals above, because the information is available before the actual fact takes place, is rather limited. Still, we have found that because the information about what is planned to happen in the future precedes the information about what actually is happening, it is possible to almost completely remove interferences between robots.

As the robots are decentralized and have their own decision processes, even though they use first-come first-serve policy to compete for subtasks and destinations, it may happen that they make decisions in a synchronous manner. For instance, both robot Alice and Bob may adopt a goal to explore the same room at the same time, which indeed does not violate the first protocol of shared goals when they make such a decision but may actually lead to an interference situation. In order to prevent such inefficiency, in our coordination protocols the robots can also drop goals that have already been adopted so that they can stop corresponding actions that are being executed. For example, a robot can drop a goal to enter a room if it finds that another robot also wants to enter that room. Apart from this reason, some dated goals should also be removed from the goal base. For example, a robot should drop a goal of retrieving a block if it does not have the currently needed color any more. As can be seen in Algorithm 1, a robot will check dated and useless goals and then drop them (see see line 9-12) in its decision making process.

When robots communicate both beliefs and goals, the coordination protocol combines the rules listed above for belief and goal messages. For example, a robot will not adopt a goal of going to a room if the room is occupied now or another robot already wants to enter it. Similarly, a robot will not adopt a goal to hold a block if the block is already being held by another robot or another robot already wants to hold it. In the case of communicating both beliefs and goals, as the robots should have more complete information about each other, in comparison with the cases of communicating only beliefs and only goals, so they are expected to achieve more additional gains with regard to interference and duplication of effort. But it should be noted that since all the robots have much knowledge to avoid interference, they may need to frequently change their selected destinations or to choose farther rooms, which may result in an increase in the walking time. In our experiments, we will investigate how much performance gain can be realised in these three communication cases.

5 EXPERIMENTAL DESIGN

5.1 Data Collection

All the experiments are performed in the BW4T simulator in GOAL. We have collected data on a number of different items in our experiments. The main performance measure, i.e., *time-to-complete*, has been measured for all runs. In order to gain more detailed insight into the effort needed to finish the task, we have collected data on *duplicated effort* that gives some insight into both the effectiveness of the strategy as well as in the complexity of the tasks. Duplication can be obtained by keeping track of the number of blocks that are dropped by the robots without contributing to the team goal.

Each time when two robots collide, i.e., one robot tries to enter a room occupied by another, is also logged. The *total number of interference* provides an indication of the level of coordination within the team. Finally, to obtain a measure of the cost involved in communication in multi-robot teams, the *number of exchanged messages* is also counted. A distinction is made between messages about the beliefs and messages about goals.

5.2 Experimental Setup

There are many variations in setup that one would like to run in order to gain as much insight as possible into the impact of various factors on the performance of a team. Since we want to understand the relative speed up of teams compared to a single robot to measure the effectiveness of various of our coordination protocols, we need to run simulations with a single robot. For multiple robots, we have used team sizes of 5 and 10. We also consider the factor of robots' environmental size, and we use maps of 12 rooms and 24 rooms.

In our experiments, the robots are required to retrieve 10 blocks from their environments where there are total 30 blocks that are randomly distributed for each simulation, but there are two different tasks. Recall that the first task does not have ordering constraints on the team goal and the robots retrieve block of the same color, which is relatively simple and similar to the tasks that many researchers have addressed in the robot foraging domain. Comparatively, the second task is more complicated and is a cooperative foraging task that has ordering constraints on the team goal, requiring the robots to retrieve blocks of various colors. In order to so so, in each run, BW4T simulator randomly generates a sequence of blocks of various colors, so the team goal is to collected blocks of random colors in the right order.

We list the baseline instances in Table 1 based on the combinational strategies used by the robots. A single robot's behavior is relatively simple because it does not need to consider interference with other robots, and duplicated effort will never occur. Although there are 4 dimensions in baseline strategies, a single robot only needs to consider the first and the third dimension with regards to initial room selection and next room selection, respectively. Accordingly, the single robot case has 4 setups in each environmental condition. For instance, we use $S(\text{iii})$ in Section 6 to indicate the strategy combining (1b, 3a).

When multiple robots participate in the tasks, there are more combinations based on the strategies that the robots may use. As baseline has four dimensions, each of which has two options, we can get at most 16 combinational strategies. Although we cannot directly figure out which combinational strategy is the best one without experiments, we can still eliminate several choices that are apparently inferior to the other ones.

One issue occurs when the strategy combines (2a) and (3b). In case a robot wants to go to room A but one of its teammates arrives earlier, the robot will then reconsider but select the same room again because it will select the nearest room based on the strategy. This behavior will result in very inefficient performance, so we can eliminate those choices combining (2a) and (3b). Another issue arises when a choice combines (1b) and (3b), which will make robots cluster together as they more or less start from the same location, and then they always try to visit the nearest rooms. A cluster of robots will cause inefficiency and interference, so we can further eliminate those choices combining (1b) and (3b). As a result, for the second task, we have 10 setups as shown in Table 1 and, for example, we use $M_R(\text{iii})$ in Section 6 to indicate the strategy combining (1a,2b,3a,4a).

When the robots perform the first task, as all the required blocks have the same color, the fourth dimension does not make sense because any holding block can contribute to the team goal until the task is finished. Therefore, we can eliminate this dimension and finally have 5 setups left for this task, and we use $M_S(\text{iii})$ in Section 6 to indicate the strategy combining (1a,2b,3b). For the communication cases, we have three setups, communicating only beliefs, only goals, and both beliefs and goals, in each environmental condition. Each setup has been run for 50 times to reduce variance and filter out random effects in our experiments.

Table 1: Baseline instances based on coordination protocols.

Team Size	Single Robot		Multiple Robots					
Team Goal	Either		Blocks of Same Color		Blocks of Random Color			
Instances	i	(1a,3a)	i	(1a,2a,3a)	i	(1a,2a,3a,4a)	vi	(1a,2b,3b,4b)
	ii	(1a,3b)	ii	(1a,2b,3a)	ii	(1a,2a,3a,4b)	vii	(1b,2a,3a,4a)
	iii	(1b,3a)	iii	(1a,2b,3b)	iii	(1a,2b,3a,4a)	viii	(1b,2a,3a,4b)
	iv	(1b,3b)	iv	(1b,2a,3a)	iv	(1a,2b,3a,4b)	ix	(1b,2b,3a,4a)
			v	(1b,2b,3a)	v	(1a,2b,3b,4a)	x	(1b,2b,3a,4b)

6 RESULTS

6.1 Baseline Performance

Figure 3 and 4 show the performance of the various strategies for the baseline on the horizontal axis and the four different conditions related to team size and room numbers on the vertical axis. This results show that the relation between team size and environment size has an important effect on the team performance that also relies on which combinational coordination strategy is used.

Statistically, the performance of the combinational strategies does not significantly differ from any of the others. Even so, from Figure 3 we can see that the strategy $M_S(\text{iii})$ on average performs better than any of the other ones and has minimal variance which is why we choose this strategy as our baseline to compare with the communication cases. This strategy combines options (1a) which initially selects a room randomly, option (2b) which uses information from collisions with other robots to avoid duplication of effort, and option (3b) which selects the nearest room to go to next. Interestingly, this strategy does not minimize interference. This is because if the robots do not communicate with one another, using the option to go to the nearest rooms for the next room selection, i.e., option (3b), increases the likelihood of selecting the same room at the same time and causes interference.

Similarly, from the data shown in Figure 4, it follows that strategy $M_R(\text{v})$ on average takes the minimum time-to-complete the second task and again its variance is also less than those of the other strategies. This strategy combines options (1a), (2b), (3b) and (4a). It thus is an extension of the best strategy $M_S(\text{iii})$ for the first task in Figure 3 with option (4a) which means that robots wait in front of the drop-zone if they hold a block that is needed later. We can conclude that even though sometimes option (4a) makes the robots idle away and stop in front of the drop-zone with a block that is needed later, it is more economical than the idea of storing the block in the nearest room so that it can be picked up again later.

6.2 Speedup with Multiple Robots

Balch (Balch and Arkin, 1994) introduced a speedup measurement, which is used to investigate to what extent multiple robots in comparison with a single robot can speed up a task. We have plotted the performance of the best strategies for single and multiple robot cases to analyse the speedup of various team sizes in Figure 4(d). The results show that doubling a robot team does not double its performance. For example, 5 robots take 27.26 seconds on average to complete the second task in 12 rooms, but 10 robots on average need 26.88 seconds. We therefore conclude that speedup obtained by using more robots is sub-linear, which is consistent with the results reported in (Balch and Arkin, 1994).

In order to better understand the relation between speedup and the strategies we have proposed, we can inspect Figure 4(a) again. We can see that speedup depends on the strategy that is used. One particular fact that stands out is that the time-to-complete for the odd numbered strategies in Figure 4(a) is similar for 5 and 10 robots that are exploring 12 rooms (we find no statistically significant difference). We can conclude that adding more robots does not necessarily increase team performance because more robots may also bring about more interference among the robots. We can also see in Figure 4(b) that it is quite clear that the average numbers of interference in 5 and 10 robots exploring 12 rooms are significantly different. Thus, more interference makes the robots take more time to complete the task, but it also depends on the strategies. It follows that using more robots only increases the performance of a team if the right team strategy is used. It is more difficult to explain the fact that both interference and duplication of effort are lower for the odd-numbered strategies than for the even-numbered ones. It turns out that the difference relates to option 4 where robots in all odd-numbered strategies wait in front of the drop-zone if a block is needed later. In a smaller sized environment, adding more robots in the second task means also adding more waiting time which in this particular case cancels out any speedup that one might have expected.

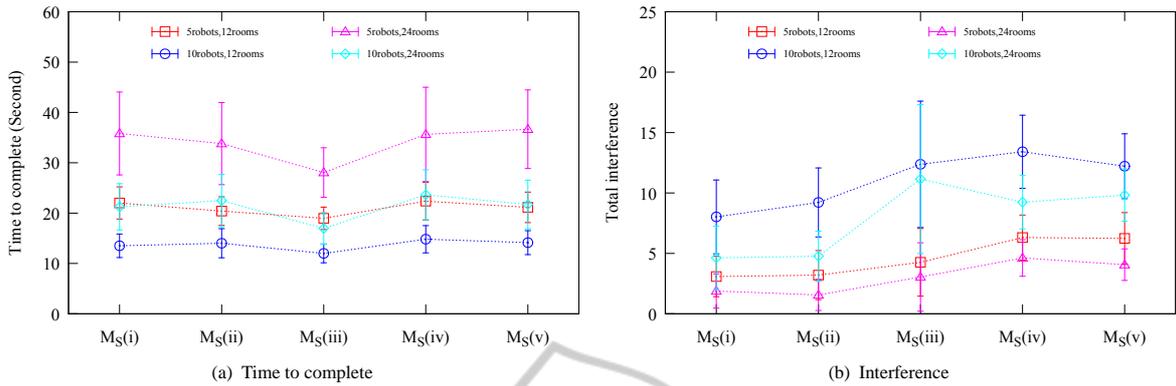


Figure 3: Baseline performance for the first task (i.e., team goal without ordering constraints).

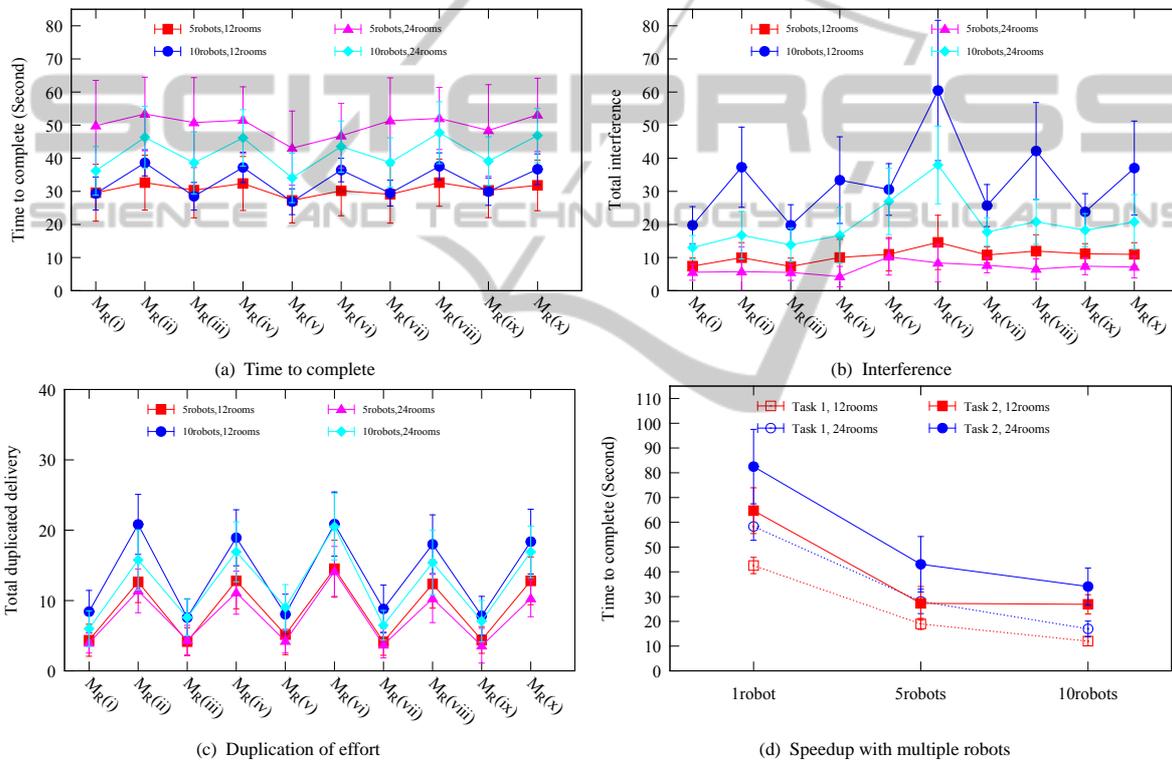


Figure 4: Baseline performance for the second task (i.e., team goal with ordering constraints) & Speedup.

6.3 Communication Performance

Figure 5 shows the results that we obtained for the performance measures for the communication conditions we study here. First, we can see in Figure 5(a) that communication is much more useful in the second task than the first one. When 5 robots operate in the 12 rooms environment, communicating beliefs yields a 34.15% gain compared to the no communication case for the first task whereas it yields a 44.5% gain in the second task. It is also clear from Figure 5(a) that communication yields more predictable

performance as the variance in each of the communication conditions is significantly less than that without communication. Recall that the second task requires robots to retrieve blocks in a particular order, and thus we can conclude that when a multi-robot task consists of multiple subtask that need to be achieved with ordering constraints, communication will be particularly helpful to enhance team performance.

Second, though communicating beliefs is more costly than communicating goals in terms of messages, the resulting performance of time-to-complete is significantly better when communicating only be-

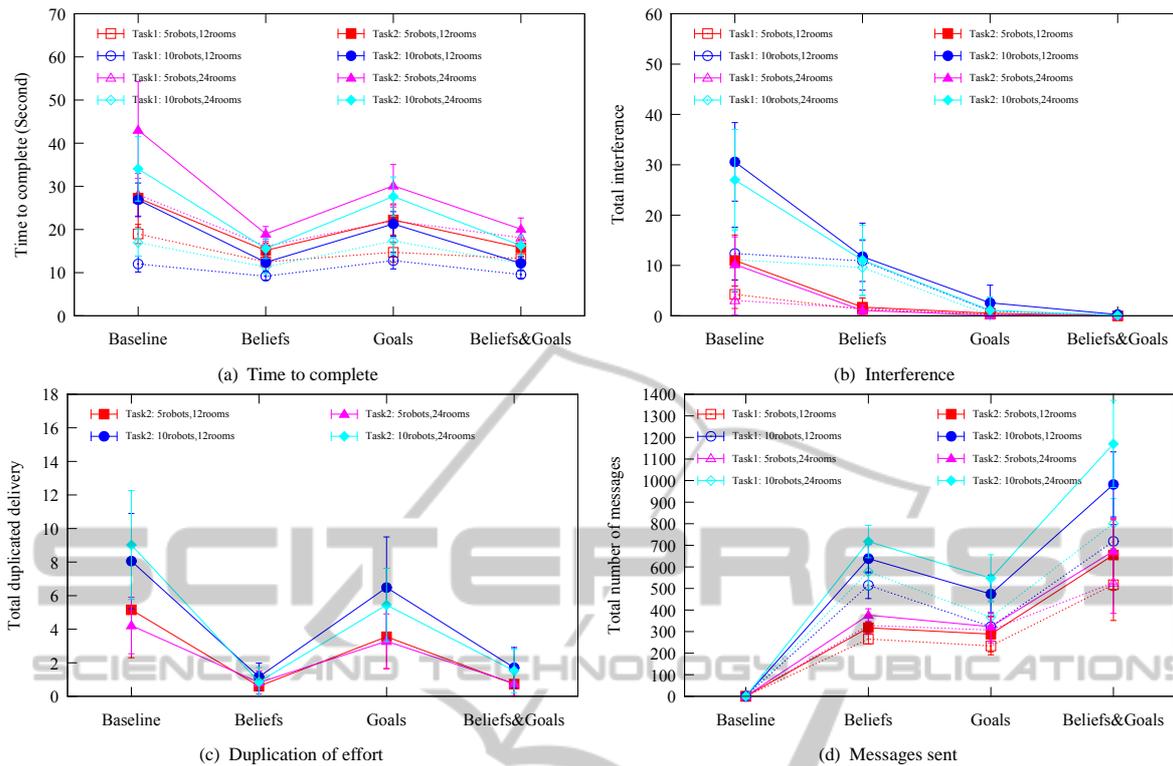


Figure 5: Performance measures for different communication conditions.

liefs compared to the performance when only goals are communicated. For example, Figure 5(a) and 5(d) show that when 5 robots only communicate beliefs, the team takes 15.13 seconds and sends 318 messages on average to complete the second task in the 12 rooms environment while only communicating goals takes the same team 22.23 seconds and 288 messages. This is because communicating beliefs can inform robots about what blocks have been found by teammates, and the environment can become known for all the robots sooner than the case of communicating goals, which can save the exploration time.

Third, the communication of goals yields a significantly higher decrease in interference compared to communicating beliefs. For instance, communicating goals eliminated 91.56% of the interference present in the no communication condition compared to only 61.6% when communicating beliefs for 10 robots that perform the second task in 12 rooms. This is because communicating goals can inform a robot about what its teammates want to go, so it can choose a different room as its destination. Comparatively, a robot only inform its teammates when entering a room in the case of communicating beliefs, which cannot effectively prevent other teammates clustering together in front of this room. On the other hand, the communication of goals does not significantly decrease

duplication of effort as shown in Figure 5(c). There is a simple explanation of this fact: even though a robot knows which blocks its teammates want to handle in this case, it does not know what color these blocks have if it did not observe the block itself before. This lack of information about the color of the blocks makes it impossible to avoid duplication of effort in that case.

A somewhat surprising observation is that though communicating only beliefs or only goals would never negatively impact performance, communicating both of them does not always yield a better performance than just communicating beliefs. T-tests show that there is no significant difference between communicating only beliefs and communicating both beliefs and goals with regards to time-to-complete. The reason is that when the robots share both beliefs and goals, they are even better informed about what their teammates are doing, which allows them to reduce interference even more. We can see in Figure 5(b) that communicating both beliefs and goals can ensure that the robots do not collide with each other anymore. However, this more careful behavior though not often but still sometimes results in a robot choosing rooms that are farther away on average in order to avoid collisions with teammates, which may increase the time to complete the entire task.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we presented various coordination protocols for cooperative multi-robot teams performing search and retrieval tasks, and we compared the performance of a baseline case without communication with the cases with various communication strategies. We performed extensive experiments using the BW4T simulator to investigate how various factors, but most importantly how the content of communication, impacts the performance of robot teams in such tasks with or without ordering constraints on the team goal. A key insight from our work is that communication is able to improve performance more in the task with ordering constraints on the team goal than the one without ordering constraints. At the same time, however, we also found that communicating more does not always yield better team performance in multi-robot teams because more robots will increase the likelihood of interference that depends on what coordination strategy the robots have used. This suggests that we need to further improve our understanding of factors that influence team performance in order to be able to design appropriate coordination protocols.

In future work, we aim to study the impact of various other aspects on coordination and team performance. We are planning to do a follow-up study in which communication range is limited and it is possible that messages get lost. We also want to study resource consumption issues where robots, for example, need to recharge their batteries. The fact that the BW4T environment abstracts from various more practical issues allows one to focus on aspects that we believe they are most relevant and related to team coordination. At the same time we believe that it is both interesting and necessary to increase the realism of this environment in order to be able to take account of various aspects that real robots have to deal with. One particular example that we are currently implementing is to allow for collisions of robots in the environment, i.e., the robots cannot pass through each other in hallways in our case. Thus, cooperative foraging tasks also involve multi-robot path planning problems, in which, given respective destinations, multiple robots have to move to their destinations while avoiding stationary obstacles as well as teammates. Finally, we are working on an implementation of our coordination protocols on real robots, which will allow us to compare the results found by means of simulation with those we obtain by having a team of real robots to complete the tasks.

REFERENCES

- Balch, T. and Arkin, R. C. (1994). Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1(1):27–52.
- Campo, A. and Dorigo, M. (2007). Efficient multi-foraging in swarm robotics. In *Advances in Artificial Life*, pages 696–705. Springer.
- Cannon-Bowers, J., Salas, E., and Converse, S. (1993). Shared mental models in expert team decision making. *Individual and group decision making*, pages 221–245.
- Cao, Y. U., Fukunaga, A. S., and B., K. A. (1997). Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4:1–23.
- Davids, A. (2002). Urban search and rescue robots: from tragedy to technology. *Intelligent Systems, IEEE*, 17(2):81–83.
- Farinelli, A., Iocchi, L., and Nardi, D. (2004). Multi-robot systems: a classification focused on coordination. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(5):2015–2028.
- Hindriks, K. (2013). The goal agent programming language. <http://ii.tudelft.nl/trac/goal>.
- Johnson, M., Jonker, C., van Riemsdijk, B., Feltovich, P. J., and Bradshaw, J. M. (2009). Joint activity testbed: Blocks world for teams (bw4t). In *Engineering Societies in the Agents World X*, pages 254–256.
- Jonker, C. M., van de Riemsdijk, B., van de Kieft, I. C., and Gini, M. (2012). Towards measuring sharedness of team mental models by compositional means. In *Proceedings of 25th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE)*, pages 242–251.
- Krannich, S. and Maehle, E. (2009). Analysis of spatially limited local communication for multi-robot foraging. In *Progress in Robotics*, pages 322–331. Springer.
- Mohan, Y. and Ponnambalam, S. (2009). An extensive review of research in swarm robotics. In *World Congress on Nature & Biologically Inspired Computing*, pages 140–145. IEEE.
- Parker, L. E. (2008). Distributed intelligence: Overview of the field and its application in multi-robot systems. *Journal of Physical Agents*, 2(1):5–14.
- Rosenfeld, A., Kaminka, G. A., Kraus, S., and Shehory, O. (2008). A study of mechanisms for improving robotic group performance. *Artificial Intelligence*, 172(6):633–655.
- Rybski, P. E., Larson, A., Veeraraghavan, H., Anderson, M., and Gini, M. (2008). Performance evaluation of a multi-robot search & retrieval system: Experiences with mindart. *Journal of Intelligent and Robotic Systems*, 52(3-4):363–387.
- Ulam, P. and Balch, T. (2004). Using optimal foraging models to evaluate learned robotic foraging behavior. *Adaptive Behavior*, 12(3-4):213–222.
- Yuh, J. (2000). Design and control of autonomous underwater robots: A survey. *Autonomous Robots*, 8(1):7–24.