Communication strategies for environment exploration using a frontier-guided Decentralised MCTS

Mathilde Jeannin, David Filliat, Eric Goubault, Sylvie Putot

Communication strategies for environment exploration using a frontier-guided Decentralised MCTS

Mathilde Jeannin[1], David Filliat[2], Eric Goubault[1] and Sylvie Putot[1]

***Abstract*— Multi-robot exploration of unknown environments is challenging, especially in scenarios with constrained communication. Decentralised Monte Carlo Tree Search (Dec-MCTS) is a promising online planning approach that enables robots to collaboratively explore an environment while handling communication failures. In this work, we propose a communication strategy for a frontier-guided version of the Dec-MCTS that is robust to communication loss and limited range while maintaining efficient exploration. Through extensive experimental evaluations, we demonstrate that limiting communication does not significantly impact performance and that discarding outdated information can negatively affect exploration efficiency. We also study how information propagation can positively or negatively impact the mission depending on the environment.**

I. Introduction

Robotics systems are increasingly being applied in civilian and military applications as hazardous or hard-to-reach environments can be dangerous or impractical for human intervention. Moreover, advances in computational power have enabled the coordination of multi-robot teams to autonomously perform exploration tasks such as reconnaissance [18] or search and rescue missions [1], in particular using collaborating UAVs. Indeed, these approaches reduce the reliance on direct human control, which is often constrained by limited communication in such environments. However, robust control methods are required to ensure these autonomous systems' effectiveness. They should be capable of coordinating exploration efficiently and maintaining performance in the event of robot failures or communication loss.

Numerous methods for autonomous and coordinated explorations of unknown environments have been developed. In scenarios where communication is constrained, distributed algorithms are commonly employed. These techniques encompass well-established techniques such as distributed consensus on objectives as the travelling salesman problem [19], Multi-Agent Reinforcement Learning (MARL) [11] or online planning methods as Dec-MCTS [4]. In these approaches, sharing information is necessary for a properly distributed solution because insufficient communication can lead to parallel individual missions with reduced efficiency. However, continuous communication is not feasible in scenarios such as reconnaissance missions, operations in confined areas, situations with limited computational resources, or instances of agent loss. In these conditions, distributed solutions must take into account the handling of communication defects.

[1] LIX, Ecole Polytechnique, Institut Polytechnique de Paris, Palaiseau, France mathilde.jeannin@polytechnique.edu

[2] U2IS, ENSTA, Institut Polytechnique de Paris, Palaiseau, France

We are specifically interested in the Dec-MCTS algorithm developed by Best et al. in [4]. It is based on Monte Carlo Tree Search (MCTS), a search algorithm using Markov Decision Processes (MDP) that analyses the most promising actions over random samples of the state space on long planning horizons [7]. Dec-MCTS is an online procedure where agents jointly choose sequences of actions by computing their own most promising sequences of actions while considering the others received through communication until a consensus is made. Best et al. in [5] showed that Dec-MCTS is robust to communication failure and additionally developed an algorithm that evaluates the benefit of sharing information to decide when to communicate and to which agent, with good performances.

In this paper, we propose a new version of Dec-MCTS, which includes a frontier-based rollout policy that shows better efficiency, particularly in indoor-like environments where dead-ends are disadvantageous. We also propose a communication strategy that maintains good performance with reduced range and connection probability: we consider data relaying between robots and keeping outdated information about others' sequences of actions. This approach allows for a more informed consensus over decision-making, and we observe that using outdated data is often still relevant, resulting in more robust explorations. Through an ablation study, we can determine the best parameters according to the context: we show that our communication strategy benefits the exploration of outdoor-like maps and that, unintuitively, too much knowledge slows indoor-like maps' explorations.

II. Related work

Exploring environments with a swarm of autonomous agents presents two key challenges: (i) planning and allocating tasks distributively and (ii) communicating. Various approaches have been proposed to address these problems.

Frontier-based methods initially introduced by Yamauchi [20] are widely used for exploring environments and can rely on efficient methods to detect and assign frontiers [13], [3]. Frontier-based control algorithms are simple and ensure full environment coverage but are not always efficient because they can be redundant, leading to unnecessary back-and-forth.

Reinforcement learning (RL)-based techniques for multi-agent cooperation and communication have been studied for a long time (e.g., [17]). They are now widely employed thanks to advances in computational power. They usually rely on a Markov Decision Processes (MDP) model for which an optimal policy is computed. Recent surveys and works

provide a good understanding of multi-agent RL in various scenarios [11], including the handling of communication [21] and the analysis of teammates intentions [16]. Although RL methods can show good results, they require much computational power for offline training and are difficult to generalise.

MDP-based approaches can also be used in online contexts. They can be extended to Partially Observable Markov Decision Process (POMDP), which adds consideration of uncertainties [14] and more specifically in multi-agents contexts [8]. In particular, Monte Carlo Tree Search (MCTS), described in a survey by Browne et al. in [7], is a robust and adaptive technique capable of handling large state and action spaces. MCTS can be extended with existing techniques such as POMDP in [15]. Another interesting variant is the decentralised MCTS (Dec-MCTS) described by Best et al. in [4]. In this work, each agent computes its own tree by taking into account the most likely action sequences of the other agents. They show promising results in decision-making using intermittent communication compared to a centralised MCTS. The MCTS structure is particularly interesting because its heuristic and the agents' objective can be easily adapted to suit the context. Dec-MCTS is even more promising as it reduces the instability due to the loss of communication and robots.

In survey [2], which condenses studies about communication in multi-agent systems, the authors describe different ways to handle restricted communication depending on the needed level of connectivity. The robots' behaviour can be modified to stay in line of sight or gather regularly. Such a solution is described in [12], where the robots decide on the date and place of gathering to share their data at regular intervals. Best et al. also extended their work on Dec-MCTS to determine when to communicate and the impact of message loss [5]. They developed an algorithm that predicts the benefit of communicating, demonstrating that only the closest neighbours are essential. However, it adds complexity to the already resource-consuming Dec-MCTS and repeated communication is required at each step to create a consensus over the planning. Nonetheless, they showed that message loss does not significantly impact performance.

In this work, we propose a modified version of the Dec-MCTS algorithm to include a frontier search method as a rollout policy instead of a random walk. We also propose a computationally simple communication strategy that is robust to communication loss and limited range while relaying the data between the agents to improve the consensual decision-making without gathering.

III. Problem Statement

We now formalise the problem of exploration of an unknown environment by a multi-agent system using the Dec-MCTS control algorithm.

A. Environment modeling

The environment is a two-dimensional enclosure of known size with unknown static obstacles. It is represented by a gridmap $G = \{g_{i,j}\}, i \in [1,M], j \in [1,N]$. The cell value is -2 if it is unknown, -1 if there is an obstacle, and 0 if it is free. The team of robots is composed of N_R robots. One robot can move from its position p^r to one of the 8 cells around with corresponding actions a. It has a limited vision range v for mapping its surroundings in its own gridmap G^r. It can communicate with a communication range of c. The stored information by the robot r about the other robots is noted $\_^{-r}$.

B. MCTS

MCTS [7] is a heuristic search algorithm for decision-making processes based on MDP. One call of MCTS consists of creating a tree where nodes represent the states, and edges represent the actions. This process iterates the following steps a given number of times: selection of a node, expansion of the tree, simulation, and backpropagation. In the selection phase, the MCTS algorithm uses a tree policy, typically the Upper Confidence Bound (UCB), to navigate through the tree until it reaches a state with an unvisited child. Once identified, the expansion phase creates a new state by choosing an action. The simulation phase then follows, where the algorithm simulates the outcome of following a given policy from the newly added state, estimating its value. This phase is also called rollout. The rollout policy is usually a random walk. Finally, this value is backpropagated through the tree, updating the value of the nodes along the path. This iterative process searches the space of states to find the best action sequence.

C. Dec-MCTS

Algorithm 1 Original Dec-MCTS

$T^r \leftarrow$ initialise tree
while Computational budget not met, at iteration n **do**
  $x_n^r \leftarrow SelectSetOfSequences(T^r)$
  for τ_n iterations **do**
    $T^r \leftarrow GrowTree(T^r, x_n^{-r}, q_n^{-r})$
    $q_n^r \leftarrow UpdateDistribution(x_n^r, q_n^r, x_n^{-r}, q_n^{-r})$
    $CommunicationTransmit(x_n^r, q_n^r)$
    $(x_n^{-r}, q_n^{-r}) \leftarrow CommunicationReceive$
  end for
end while
return $x^r \leftarrow argmax_{x^r}[q_n^r]$

1) Original Dec-MCTS: The original Dec-MCTS algorithm performs distributed decision-making using MCTS. Each robot maintains its own tree to evaluate its own actions, but it also maintains a probability distribution over the joint action space. To do so, each robot communicates a compressed form of its tree representing its own best sequences of actions and uses the sequences it received from the other robots to optimise its tree. It is described in Algorithm 1 where T^r is the robot's r tree, x^r is its best sequences of actions from T^r and q^r their probability distribution.

2) Frontier-guided Dec-MCTS description: We now describe the detailed implementation of our new approach of Dec-MCTS through frontier-guided rollouts in an exploration problem and our proposed strategy to improve communication. Our framework is described in Algorithm 2.

Algorithm 2 Frontier-guided and communication-oriented Dec-MCTS

$p_0^r, p_0^{-r} \leftarrow initialise\_positions()$
$G_0^r \leftarrow initialise\_gridmap()$
$s_0^r \leftarrow \{p_0^r, G_0^r, p_0^{-r}, x^{-r}, q^{-r}\}$
while $\exists (i,j), g_t^r(i,j) = -2$ **do**
  for 10 iterations **do**
    if available communication **then**
      for each robot r' in communication range **do**
        $G_t^r, p^{-r}, x^{-r}, q^{-r} \leftarrow receive\_communication(r')$
        $send\_communication(r')$
      end for
    end if
    $s_t^r \leftarrow \{p_t^r, G_t^r, p^{-r}, x^{-r}, q^{-r}\}$
    $a_t^r, x_t^r, q_t^r \leftarrow MCTS(s_t^r)$
    $q_t^r \leftarrow optimization(x_t^r, q_t^r, x^{-r}, q^{-r})$
  end for
  $p_{t+1}^r \leftarrow move\_agent(a_t^r)$
  $G_{t+1}^r \leftarrow update\_map()$
end while

The MDP (S, A, T, R) describing the problem for one robot is the following:

- state s_t^r in S at time t is represented by $s_t^r = (p_t^r, G_t^r, p^{-r}, x^{-r}, q^{-r})$ with $p_t^r \in \mathbb{N}^2$ the position of robot r, G_t^r its knowledge of the occupancy gridmap, $p^{-r} = \{p_{t_c}^{r'}, r' \neq r\}$ the last known positions of the other robots communicated at time t_c, $x^{-r} = \{x_{t_c}^{r'}, r' \neq r\}$ is the last known best sequences of actions of the other robots, t_c depending on each robot r' and $q^{-r} = \{q_{t_c}^{r'}, r' \neq r\}$ the probability distributions associated with x^{-r}.
- action a_t in A at time t corresponds to one of the eight possible directions.
- the transition function $T : S \times A \to S$ computes the next simulated state s_{t+1}^r using the action a_t^r but also the most likely actions of the other robots using x^{-r}. If the state s_t^r is the tree's root, the transition function T chooses for every other robot the sequence of actions they will follow from the child s_{t+1}^r. They are selected using their probability distributions q^{-r}. For example, for a robot r', $x_{i,t_c}^{r'} = \{a_{i,t_c}^{r'}, \ldots, a_{i,t_c+k-1}^{r'}\}$ with $x_{i,t_c}^{r'} \in x_{t_c}^{r'}$ and $x_{t_c}^{r'} \in x^{-r}$, one of the sequences that have been communicated by r' at time t_c is selected with a probability $q^{r'}(x_{i,t_c}^{r'})$. When the sequence has ended, meaning the robot r has applied the whole r' sequence, it will simulate r' motion with random actions.
  We also consider the case where $t_c + k$ is in the past at the beginning of the simulation, which implies that the sequence of actions is outdated for robot r'. It means that robot r' has already applied a sequence of more than k actions in the real exploration and is possibly far from its last known position $p_{t_c}^{r'}$. Hence, we tested the impact of discarding such outdated information and replaced them with random actions from $p_{t_c}^{r'}$ in the ablation study in section V.
- the reward function $R : S \times A \times S$ computes the reward associated with a sequence $(s_t^r, a_t^r, s_{t+1}^r)$. The reward R is proportional to the number of cells discovered by the robot r when it moves.

3) Frontier-guided rollouts: We modified the rollout policy to a standard frontier exploration. As explained in Section III-B, it usually consists of a random walk from the newly added node, which is the case in the original Dec-MCTS [4]. However, the robot may be too far from unknown cells in an exploration context requiring the rollout random walk to reach those cells so the backpropagated reward is non-zero. However, the distance travelled by a random walk is of the order of $\sqrt{n}$, which means that if the robot is too far away from unknown areas, it may never find the optimal action leading towards them. That is why we decided to replace the random walk of the rollout policy with a standard frontier-based method. Specifically, at the beginning of the rollout phase, the robot computes the frontiers in the simulated gridmap. It randomly chooses a frontier cell and uses an A^* algorithm to find a path toward this cell. When it reaches it, it starts again until it either reaches the given depth of the tree or finds a final state. This policy allows the robot to always reach unknown cells during the rollouts.

The rollouts are interrupted periodically to allow the robots to communicate the best sequences of actions computed and include the received sequences from others (see parameters in section IV-A).

4) Communication: In the original Dec-MCTS algorithm, the robots repeatedly communicate and expand their search trees during one occurrence of the algorithm. The authors showed the algorithm's robustness to communication loss and introduced a method to decide when and with whom to communicate, narrowing it to neighbours. However, this communication strategy has been added to Dec-MCTS, which is already very resource-consuming. In this work, we introduce another simpler approach. It consists of data relaying among robots to ensure that critical information transits from one robot to another and is robust to the limitation of communication range and frequency while exploring an unknown environment.

Robot r communicates with neighbouring robots with a probability $\mathbb{P}_c$ within a radius c. We assume communication without data corruption. A robot r shares its position p_t^r and occupancy gridmap G_t^r. Receiving robot r' merges their gridmaps :

$$\forall (i,j) \quad g_t^{r'}(i,j) \Leftarrow g_t^r(i,j) \text{ if } g_t^{r'}(i,j) = -2 \wedge g_t^r(i,j) \neq -2$$

Robot r also shares its most probable action sequences x_t^r and their associated probability distribution q_t^r.

Finally, robot r shares with r' its stored information about the other robots: p^{-r}, x^{-r}, q^{-r}. During the communication

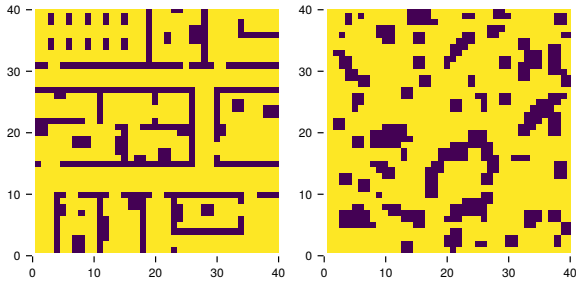

Fig. 1: Indoor-like (left) and outdoor-like (right) maps. Yellow cells are empty cells, and dark blue cells are obstacles.

phase, the robot r receives similar information from its neighbours. If that information is more recent than its own saved data, r' stores it. We call this approach data relaying (DR), and it is part of the communication strategy we propose in this work.

5) Sequences of actions, probability distribution and optimisation: We use the MCTS tree to sample the most promising sequences of actions x^r that will be transmitted. We also compute the associated probability distribution q^r. It is then optimised using the last known sequences of actions x^{-r} and probability distributions q^{-r} received from the other robots to accentuate the sequences of actions that give the best reward. The optimisation of the probability distribution q^r is a decentralised gradient descent method over the space of product distributions $\{\prod_{r\in\{1,\dots,R\}} q^r(x^r), \forall x^r\}$ such as presented in [4].

IV. Experimental evaluation

In this section, we evaluate the performances of our algorithm, Frontier-guided MCTS, compared to the original Dec-MCTS from [4]. Code is available on GitHub at https://github.com/MathildeJeannin/MultiRobotExplorationAlgos.

A. Experimental Setup

Both algorithms were implemented using the Julia programming language. In particular, we use the Agents.jl, MCTS.jl and POMDPs.jl packages [9], [10]. These algorithms are tested on three maps: indoor-like and outdoor-like (Figure 1), and random maps of dimension 40×40 cells. The random maps are generated with 8 obstacles of size between 4 and 7 cells in length and height. Each algorithm has been tested 52 times on each map with a team of 5 robots and a vision range of $v=3$ cells. They always start exploring at the bottom left of the maps and stop after 500 steps if it is not finished. We consider that the exploration ends when at least one robot knows the whole map. We iterate 300 times per MCTS call and stop every 30 rollouts to receive and communicate the best sequences of actions that have been computed so far. The depth of the tree is 100, and the discount has been empirically set to 0.85.

We compare the performances, i.e., the number of steps needed for at least one robot to discover the whole map, of both algorithms in different scenarios:

- Full communication: implemented as a communication radius of $c=100$ so they can cover the whole map without loss ($\mathbb{P}_c=1$).
- Limited communication: robots can communicate with a radius $c=10$ and they have a probability $\mathbb{P}_c=0.2$ of successfully communicating.
- Limited communication with DR: the communication is limited with the same parameters as before, but they can relay information about position and sequences of actions about others. This is our version of the control algorithm, which we will analyse in the ablation study in the next section.

B. Results

Figure 2 shows the comparison results between the performances of original Dec-MCTS and Frontier-guided MCTS depending on the map and the set of parameters.

Firstly, we can look closer at the difference in results between outdoor and random maps and indoor map. Outdoor and random maps' results evolve similarly, indicating that Dec-MCTS and Frontier-guided Dec-MCTS behaviours are linked to the distribution of obstacles. The obstacles are point-like, unlike the indoor map's continuous walls and corridors.

The original Dec-MCTS performances have better medians when exploring outdoor and random maps, particularly with perfect communication. However, it is less robust than Frontier-guided Dec-MCTS when the communication is limited (with or without DR) as its IQR and boxplots' whiskers are wider. Additionnal F-tests of equality of variances confirm those results. However, unequal variances T-tests show that their means are similar on the outdoor map but are different on random maps. As an example, on the outdoor map with limited communication, there is a significant difference between Original Dec-MCTS ($M=194, SD=88$) and Frontier-guided Dec-MCTS ($M=164, SD=31$) in term of variance $F=8.08, p<10^{-11}$, and in term of mean $t(63)=2.33, p=0.023$.

The results are quite different for indoor map exploration. In this scenario, the original Dec-MCTS is very inefficient as its random walk rollout policy struggles to find remote unseen cells, making the end of the exploration very difficult. It is also surprising that Frontier-guided Dec-MCTS performances are degraded with perfect communication. This is due to hesitating behaviour between remote unknown areas offering greater but discounted rewards and closer but single cells offering less discounted but smaller rewards.

Hence, the Frontier-guided Dec-MCTS seems better suited for an indoor-like environment and is always more robust to weaker communication. The data relaying strengthens this robustness in outdoor-like environments.

V. Influence of communication in Frontier-guided Dec-MCTS

We now provide an ablation study of the parameters of our communication strategy (see Section III-C.4) within our

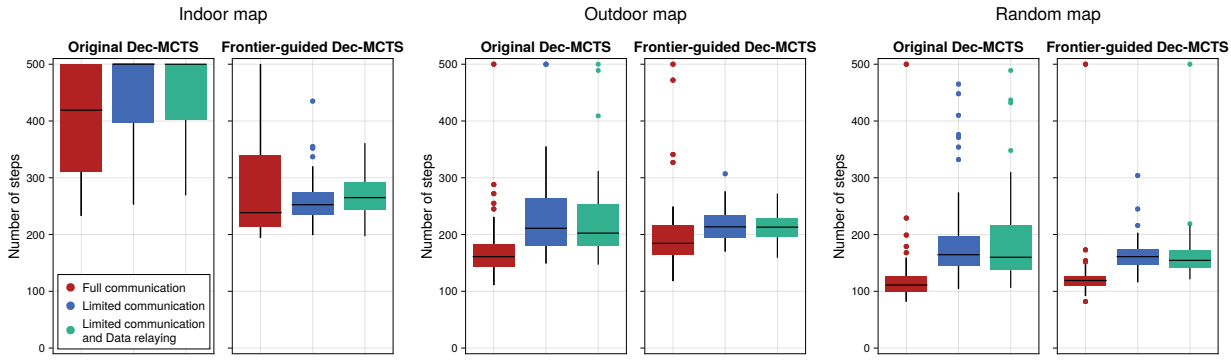

Fig. 2: Statistics illustrating the number of steps needed to explore the three maps depending on the algorithm and parameter. "Limited communication and Data relaying" corresponds to our version analysed in the ablation study. Each box represents the interquartile range (IQR) with a midline marking the median. The whiskers span 1.5× IQR.

Ablation study parameters	c	$\mathbb{P}_c$	Data relaying	Discarding outdated information
Our version	10	0.2	True	False
Total communication coverage	**100**	0.2	True	False
No communication loss	10	**1.0**	True	False
Without data relaying	10	0.2	**False**	False
With discarding outdated information	10	0.2	True	**True**

TABLE I: Set of parameters used for the ablation study.

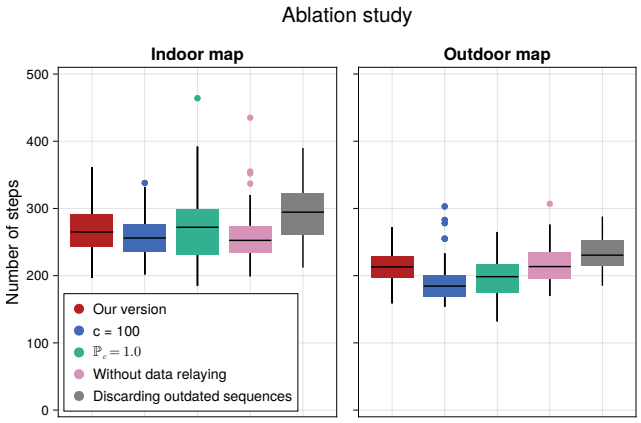

Fig. 3: Statistics illustrating the results of the ablation study for both indoor (left) and outdoor (right) maps. Each box represents the interquartile range (IQR) with a midline marking the median. The whiskers span 1.5× IQR.

Dec-MCTS framework. We aim to demonstrate each parameter's contribution to the control algorithm's performances on exploration tasks depending on the environment. The set of parameters used for the ablation study can be found in Table I.

In the following sections, we analyse the impact of the communication range, probability of communication, data relaying, and discarding of outdated sequences of actions in the tree. The results of the ablation study can be found in Figure 3.

A. *Communication range*

As expected, Figure 3 shows that a communication range c covering the entire map gives better results. However, the increased performance is not as high as expected. This can be explained for two reasons. Firstly, results showed that, on the outdoor map, robots tend to be more spread out when having a greater communication rang, which means faster covering. However, in rare cases, they leave some areas unseen, requiring them to return and delaying the exploration's end. For the indoor map, we observed that the average distance between a robot and its second closest neighbour stays under 10 for $c = 10$. This communication range is, therefore, sufficient for robots to relay their knowledge of the map during exploration. Indeed, this map has a natural direction of exploration through the principal corridor, creating a line of sight between robots. This explains why the performance gain for the full communication range is not as great as expected in the indoor case.

Secondly, as knowledge of the map spreads quickly, robots tend to hesitate between farther unknown areas, which give a greater but discounted rewards, and closer, single cells which give smaller but less discounted rewards. This leads to a very hesitating behaviour explaining that slower explorations can occur.

Thus, in a realistic scenario where the communication can be disturbed by obstacles or need to be subtle, our framework keeps acceptable performances and robustness.

B. *Probability of communication*

In this part of the ablation study, we aimed to show that the loss of messages does not strongly impact consensus decision-making. This is the case for the outdoor map (Figure 3, right). However, for the indoor map (Figure 3, left), we show that full communication during the MCTS phase can be surprisingly counter-productive. Figure 4 displays some statistics on the exploration of the environment. The

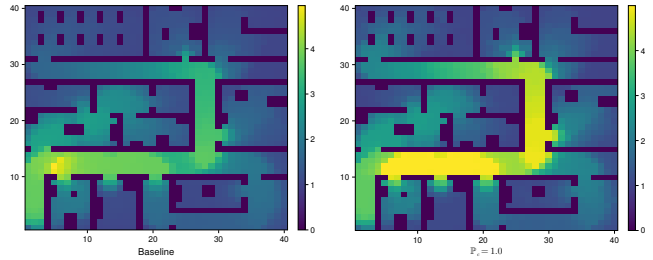

Fig. 4: Statistics over the explorations of the indoor map showing the average number of robots that has seen each cell. Left map shows our version, and right map shows the results when $\mathbb{P}_c = 1.0$.

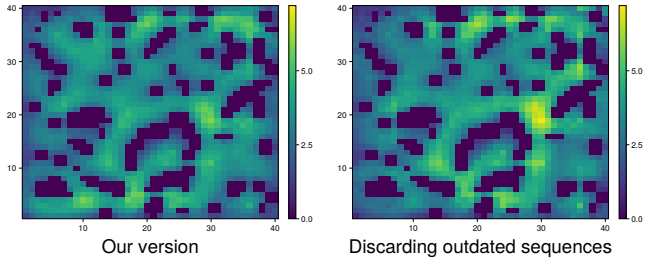

Fig. 5: Statistics over the explorations of the outdoor map showing the average number of times a cell has been seen. Left map shows our version and right map shows the results when outdated sequences are discarded.

maps show that, with a better decision-making consensus ($\mathbb{P}_c = 1.0$), all five robots reach the central corridor. However, with each communication during the MCTS phase, a robot r adds more possible sequences of actions for other robots r' creating bigger tree along and making exploiting more difficult. These abrupt changes of optimal actions due to the action sequences updates are called breakpoints in Best et al.'s work [4]. They assume that this number is upper bounded when $t \to \infty$. Hence, the number of iterations is insufficient, leading to suboptimal actions and hesitating behaviour. This is especially true where a strong decision must be made, for instance, at crossroads or room entrances (Figure 4, right).

Hence, if the loss of communication can slow the exploration in an open environment and impact the decision-making, the impact is very small. The impact can even be profitable in a closed map (indoor).

C. Data Relaying

The impact of DR is also counter-intuitive at first sight. It would seem that more knowledge about the positions and likely sequences of actions of farther robots would help to find a better decision. In the context of the outdoor map, Figures 2 and 3 show that DR makes the performances more robust as there are fewer outliers in this case. However, it is damaging in indoor map exploration. Suppose a robot knows about the positions and future sequences of actions of farther robots. In that case, it is less likely to move towards them and help them as its own MCTS simulation will probably compute that those robots have already explored their area.

For that reason, DR does not improve performance, as knowledge about remote robots is unnecessary. However, it adds robustness in outdoor-like environments.

D. Discarding outdated sequences of actions

The question here is: how do we simulate the actions $a^{r'}$ when the sequences $x_{t_c}^{r'}$ are too old? We can either simulate random actions from its last known position $p_{t_c}^{r'}$ or simulate random actions after we applied the last known sequences $x_{t_c}^{r'}$ from $p_{t_c}^{r'}$. The first behaviour is what we call discarding outdated sequences of actions. As expected, even if the last information robot r has about robot r' is old, it is still relevant for the Dec-MCTS. Hence, the ablation study shows that discarding outdated sequences in indoor and outdoor maps leads to degraded performances. Figure 5 shows that discarding sequences of actions leads a higher number of robots to visit the same areas of the map, slowing the exploration of the outdoor map. In the indoor map, too, discarding outdated information pushes the robots to have a more redundant exploration.

Hence, we show that it is better to simulate with outdated information than with random actions. Even if the data is old, it still carries pertinent hints about the other robots' whereabouts.

VI. Conclusion

This work presents an extension of Dec-MCTS using a frontier-based rollout policy and a communication strategy using DR and keeping outdated information. The frontier-guided Dec-MCTS ensures far more efficient explorations in an indoor-like environment and is as efficient in outdoor-like maps. It fails less than the original Dec-MCTS and is more robust to communication failures and restrictions, especially when combined with our communication strategy.

However, we show that the results are closely linked to the environment. DR is an approach that is more efficient in an environment where the obstacles are disseminated because the information can be used to find a better trajectory. However, indoor-like maps are more difficult to explore because the robots must take the same path regardless of the information. Hence, in this case, we show that sharing too much information and knowledge damages the exploration. However, keeping outdated sequences of actions adds robustness and effectiveness in both cases because it hints about what has already been visited.

These results motivate future work to understand why too much knowledge and data sharing can be ineffective and disadvantageous. In this case, it could be interesting to determine more closely which strategy and parameters are needed according to the environment. Moreover, another Decentralised MCTS approach has been proposed by Bone et al. in [6]. They also explore unknown environments and use frontier centroids to add promising actions. It would be interesting to compare both methods.

References

[1] Ebtehal Turki Alotaibi, Shahad Saleh Alqefari and Anis Koubaa. "LSAR: Multi-UAV Collaboration for Search and Rescue Missions". In: *IEEE Access* (2019).

[2] Francesco Amigoni, Jacopo Banfi and Nicola Basilico. "Multirobot Exploration of Communication-Restricted Environments: A Survey". In: *IEEE Intelligent Systems* 32.6 (Nov. 2017), pp. 48–57.

[3] Antoine Bautin, Olivier Simonin and François Charpillet. "MinPos : A Novel Frontier Allocation Algorithm for Multi-robot Exploration". *Intelligent Robotics and Applications*. Vol. 7507. 2012, pp. 496–508.

[4] Graeme Best et al. "Dec-MCTS: Decentralized planning for multi-robot active perception". In: *The International Journal of Robotics Research* (Mar. 2019).

[5] Graeme Best et al. "Planning-Aware Communication for Decentralised Multi-Robot Coordination". *2018 IEEE International Conference on Robotics and Automation (ICRA)*. May 2018.

[6] Sean Bone et al. "Decentralised Multi-Robot Exploration Using Monte Carlo Tree Search". *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Oct. 2023, pp. 7354–7361.

[7] Cameron B. Browne et al. "A Survey of Monte Carlo Tree Search Methods". In: *IEEE Transactions on Computational Intelligence and AI in Games* (Mar. 2012).

[8] Jesus Capitan et al. "Decentralized multi-robot cooperation with auctioned POMDPs". In: *The International Journal of Robotics Research* (May 2013), pp. 650–671.

[9] George Datseris, Ali R. Vahdati and Timothy C. DuBois. "Agents.jl: a performant and feature-full agent-based modeling software of minimal code complexity". In: *SIMULATION* (Jan. 2022).

[10] Maxim Egorov et al. "POMDPs.jl: A Framework for Sequential Decision Making under Uncertainty". In: *Journal of Machine Learning Research* 18.26 (2017), pp. 1–5.

[11] Iou-Jen Liu et al. "Cooperative Exploration for Multi-Agent Deep Reinforcement Learning". *Proceedings of the 38th International Conference on Machine Learning*. July 2021.

[12] Keisuke Okumura, Yasumasa Tamura and Xavier Défago. "Amoeba Exploration: Coordinated Exploration with Distributed Robots". *2018 9th International Conference on Awareness Science and Technology (iCAST)*. Sept. 2018.

[13] Phillip Quin et al. "Approaches for Efficiently Detecting Frontier Cells in Robotics Exploration". In: *Frontiers in Robotics and AI* (Feb. 2021).

[14] S. Ross et al. "Online Planning Algorithms for POMDPs". In: *Journal of Artificial Intelligence Research* 32 (July 2008), pp. 663–704.

[15] David Silver and Joel Veness. "Monte-Carlo Planning in Large POMDPs". *Advances in Neural Information Processing Systems*. Vol. 23. Curran Associates, Inc., 2010.

[16] Aaron Hao Tan et al. "Deep Reinforcement Learning for Decentralized Multi-Robot Exploration With Macro Actions". In: *IEEE Robotics and Automation Letters* 8.1 (Jan. 2023), pp. 272–279.

[17] Ming Tan. "Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents". *Machine Learning Proceedings*. 1993.

[18] Yubing Wang et al. "Reconnaissance Mission Conducted by UAV Swarms Based on Distributed PSO Path Planning Algorithms". In: *IEEE Access* (2019).

[19] Xiao-Feng Xie and Jiming Liu. "Multiagent Optimization System for Solving the Traveling Salesman Problem (TSP)". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39 (Apr. 2009), pp. 489–502.

[20] Brian Yamauchi. "Frontier-based exploration using multiple robots". *Proceedings of the second international conference on Autonomous agents*. 1998, pp. 47–53.

[21] Lulu Zheng et al. "Episodic Multi-agent Reinforcement Learning with Curiosity-driven Exploration". *Advances in Neural Information Processing Systems*. 2021.