Decentralized Multi-robot Exploration under Low-bandwidth Communications

Jan Bayer[1*] and Jan Faigl[1]

[1]Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, Prague, 166 27, Czechia.

*Corresponding author(s). E-mail(s): bayerja1@fel.cvut.cz;
Contributing authors: faiglj@fel.cvut.cz;

Abstract

In this paper, we address the problem of coordinating multiple robots to explore large-scale underground areas covered with low-bandwidth communication. Based on the evaluation of existing coordination methods, we found that well-performing methods rely on exchanging significant amounts of data, including maps. Such extensive data exchange becomes infeasible using only low-bandwidth communication, which is suitable for underground environments. Therefore, we propose a coordination method that satisfies low-bandwidth constraints by sharing only the robot's positions. The proposed method employs a fully decentralized principle called *Cross-rank* that computes how to distribute robots uniformly at intersections and subsequently orders exploration waypoints based on the traveling salesman problem formulation. The proposed principle has been evaluated based on exploration time, traveled distance, and coverage in five large-scale simulated subterranean environments and a real-world deployment with three quadruped robots. The results suggest that the proposed approach provides a suitable tradeoff between the required communication bandwidth and the time needed for exploration.

Keywords: Decentralized coordination, Mobile robot exploration, Low-bandwidth communication, Traveling salesman problem

Nomenclature

$\mathbf{R}$	Set of all robots in a team.
n	No. of the robots in the team: $n = \lvert\mathbf{R}\rvert$.
r	The current robot making a decision.
$\mathbf{R}'$	Teammate robots of r; $\mathbf{R}' = \mathbf{R} \setminus \{r\}$.
$v_{r,\max}$	Maximum forward velocity of the robot r.
$C_{k,i}$	Path length between the kth robot and the ith exploration waypoint.
c_{trial}	No. of experiment repetitions (trials).
c_{env}	No. of different exploration environments.
t_{avg}	Average exploration time from the given set of trials.
d_{cov}	Maximum sensor coverage distance to consider an object as explored.
$l_{\text{avg,max}}$	Average longest exploration path.
l_{sum}	Sum of the distances traveled by all the robots in all trials.
cov_n	Average coverage for a team of n robots.
$S_{i,j,k}$	Area covered by the kth robot in the jth trial of the ith scenario.
S_i	Traversable area of the whole ith scenario.
$\mathcal{M}$	Local 3D grid map.
$\mathcal{M}_{\text{rad}}$	Radius of the local 3D grid map.
$\mathcal{M}_{\text{slope}}$	Terrain slope threshold for deciding traversability.
$\mathcal{M}_{\text{step}}$	Terrain step threshold for deciding traversability.
$b_{\text{positions}}$	Bandwidth required for the positions to be shared between the robots.

b_{maps} Bandwidth required for the maps to be shared between the robots.

$b_{\text{waypoints}}$ Bandwidth required for the waypoints to be shared between the robots.

$f_{\text{positions}}$ Frequency of positions broadcasting from each robot.

$\mathbf{W}$ Waypoints known to a particular robot. A waypoint location $\mathrm{w}_i \in \mathbf{W}$ is defined solely by the position $\mathbf{p}_i$, with no orientation component.

$\mathbf{p}_i$ 3D position $\mathbf{p}_i = (p_x, p_y, p_z)$ of the ith waypoint $\mathrm{w}_i \in \mathbf{W}$.

$G_{i,\text{MinPos}}$ Rank of the ith exploration waypoint calculated by the MinPos method.

$G_{i,\text{cross}}$ Rank of the ith exploration waypoint calculated by the proposed Cross-rank method.

$\mathbf{H}_k$ Ordered set of robots' $r'_k \in \mathbf{R}'$ positions received by the robot r.

$\mathbf{h}_{k,l}$ Particular position of r'_k received by the robot r. During the mission, robot r receives positions $\{\mathbf{h}_{k,0}, \mathbf{h}_{k,1}, \mathbf{h}_{k,2} \dots\}, \mathbf{h}_{k,l} \in \mathbf{H}_k$ from the kth robot as the kth robot updates its location when it explores the environment.

m_r Cross-rank parameter used for thresholding if a waypoint is considered to have already been visited by another robot.

$d_{\text{lseg}}(\mathbf{a}, \overline{\mathbf{b}})$ Function that measures a distance between a position $\mathbf{a}$ and line segment $\overline{\mathbf{b}}$.

$\overline{\mathrm{s}}(k, \mathbf{p}_i)$ Function that finds the closest line segment from the kth robot to the position p_i.

$\mathbf{W}_{\text{cr}}$ Waypoints with the lowest Cross-rank; $\mathbf{W}_{\text{cr}} \subset \mathbf{W}$.

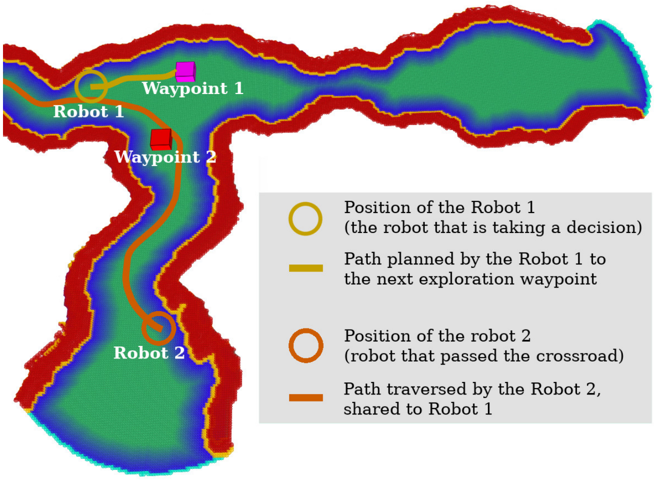

Waypoint 1
Robot 1
Waypoint 2
Robot 2
Position of the Robot 1
(the robot that is taking a decision)
Path planned by the Robot 1 to
the next exploration waypoint
Position of the robot 2
(robot that passed the crossroad)
Path traversed by the Robot 2,
shared to Robot 1

Fig. 1 An example of decision-making at the crossroads of a cave-like environment. Robot 1 is aware of only two possible exploration waypoints (Waypoint 1 and Waypoint 2) because the limited communication does not allow map sharing between the robots. Note that the map is shown only for reference. However, the traversed path (set of positions) has been shared from Robot 2 to Robot 1. Thus, based on the shared path, Robot 1 can make the desired decision to follow Waypoint 1 since Waypoint 2 is in the way that Robot 2 has already visited.

1 Introduction

The studied robotic exploration is to create a model of an unknown environment by a team of mobile robots. Several subproblems can be identified to address autonomous robotic exploration, such as localization, mapping, planning, navigation, determination of further exploration waypoints, selection of the next navigational waypoint, and, in the case of multi-robot exploration, also the teaming efficiency. Each of these parts represents a complex problem affecting the performance of the whole exploration, and all of them are actively studied and addressed by different approaches, such as localization and mapping for the exploration (Ebadi et al, 2020), while the exploration waypoint determination is addressed in (Williams et al, 2020), to name a few. The exploration performance can be improved by increasing the number of exploring robots. Then, robot coordination becomes crucial to efficiently allocating possible exploration waypoints for each robot. In the present work, we focus on robot coordination during exploration missions while considering communication constraints between the robots.

In the literature, we can find exploration approaches deployed in office-like environments (Smith and Hollinger, 2018) open outdoor areas (Huang et al, 2022), or in Subterranean Challenge (SubT) organized by Defense Advanced Research Projects Agency (DARPA) (Chung et al, 2023), where robots search various underground environments, including mine tunnels, caves, and urban underground. A significant limiter of coordination methods in underground environments is the quality of the communication, which we assume to be unreliable and low-bandwidth. Therefore, in the present study, we focus on decentralized exploration methods to account for unreliable communication (Azpúrua et al, 2023). We assume that unreliable communication causes occasional packet losses, and also, the communication between the robots might be temporarily unavailable. In the studied case, the robots are deployed underground, equipped with low-bandwidth 868/915 MHz communication modules for long-range communication. The communication modules were tested in a cave underground (Zoula et al, 2021), and they allow broadcasting about 100 B s^{-1} from each robot.

With such a limited communication bandwidth, the methods that require sharing dense maps or large amounts of data cannot be employed. Hence, we propose alternative methods that are based on the ranking of possible waypoints. The proposed ranking is called the **Cross-rank** and it is an integer value computed locally by the particular robot for each potential exploration waypoint to evaluate how many of the other robots in the team took each way at the crossroads. Thus, based on the

Cross-rank, robots aim to spread at each crossroad uniformly. It increases the likelihood of exploring areas that are either unexplored or unlikely to be visited by other robots. The idea of the proposed waypoints ranking is illustrated in an example depicted in Fig. 1. Because the Cross-rank is calculated in a decentralized way and only requires sharing position data, which is small enough to be broadcast redundantly, the method is inherently robust to temporary communication outages. Besides, it further avoids difficulties when merging maps from other robots when the mutual positions of the robots are not accurate enough.

The desired behavior can also be observed in existing exploration approaches, such as (Puig et al, 2011), where the authors rely on centralized decision-making and availability of the global map of the explored environment, which puts relatively high requirements on communication reliability and available bandwidth. Those communication properties are unavailable in the addressed scenarios, with unreliable communication capable of broadcasting just a hundred bytes per second. However, such a low-bandwidth, unreliable communication showed to be sufficient for sharing robots' positions (Bayer et al, 2023), which motivated us to study coordination methods that can exploit a broader range of communication technologies than methods that require map sharing. In addition, low bandwidth usage is also beneficial when other mission-specific data must be shared between the robots. We would like to highlight the following features of the proposed method.

- The proposed coordination principle, named the Cross-rank, improves exploration performance in comparison to independently exploring robots while using only low-bandwidth communication.
- Experiments with three quadruped walking robots demonstrate the method's ability to coordinate robots despite unreliable communication links with packet loss above 50 %.

Moreover, the contributions of the presented work are considered as follows.

- Benchmark of the 12 decentralized coordination methods, including MinPos and optimal task allocation using the Hungarian algorithm, in five large-scale underground setups, based on 600 exploration trials, where each robot traveled on average over 1.3 km in each experiment.
- Qualitative analysis of the reference state-of-the-art method *Multiple Traveling Salesman Problem* (MTSP) and qualitative analysis of the effects of greedy and TSP-based waypoint selection on the coordination efficiency.
- Empirical study of the localization drift impact on the exploration performance and coordination of the heterogeneous team.

The rest of the paper is organized as follows. An overview of the related multi-robot coordination approaches in exploration missions is summarized in the following section. The addressed problem and performance indicators used in the quality evaluation of the proposed and existing solutions are presented in Section 3. The low-level autonomy used by the coordination methods evaluated is described in Section 4. An overview of the evaluated methods from the literature and three proposed methods is presented in Section 5. The proposed decentralized coordination methods are detailed in Section 6. Evaluation results are summarized in Section 7 and further discussed in Section 8. The paper is concluded in Section 9.

2 Related Work

Autonomously exploring robots have been deployed in various outdoor (Huang et al, 2022) and indoor (Miller et al, 2020) scenarios using single and multi-robot approaches. Multi-robot teams can increase exploration efficiency by employing multiple units in parallel (Azpúrua et al, 2023). Besides, teams can also benefit from the complementary properties of the robots (Heppner et al, 2013). Nevertheless, in large-scale environments, communication plays a key role when sharing data necessary for the coordination of the robots. Existing multi-robot coordination approaches can be divided according to the demands on the bandwidth and reliability of the communication connectivity among the robots within the team. In a case where the bandwidth and connectivity are not an issue, the robots can share evidence grids to build a global map that is used to decide the next navigational waypoint for each robot, e.g., using the closest frontier (Yamauchi, 1998). Utility-based assessment of the next-to-visit location has been proposed to improve the exploration performance by evaluating the distance to the next waypoint location and the expected coverage of the not yet explored part of the environment (Burgard et al, 2000).

A segmentation method based on the Voronoi diagram capable of clustering potential exploration waypoints into sets corresponding to rooms and corridors is proposed by Wurm et al (2008). The Hungarian algorithm can then be used to assign robots to the clusters. The method is also used in (Xu et al, 2013) as centralized decision-making. The detection of important parts of the environment has been used in (Kim et al, 2023), where the authors propose to detect doors and rooms to improve the distribution of the robots in the environment. On the other hand, Puig et al (2011) propose to distribute robots in the environment by K-means clustering of the potential exploration targets, from which the next navigational waypoint is selected for each robot.

As for the environment models being explored, spatial and traversability models are jointly explored in (Prágr et al, 2022) using a formulation of the Generalized TSP to select the next navigational waypoint. Variants of the TSP are also used for solving inspection (Phung et al, 2017), single-robot surface mapping (Song and Jo, 2018), or coverage path planning (Xie et al, 2020). For the multi-robot case, Hussein et al (2014) formulated the problem as a variant of the MTSP for solving task allocation with multiple robots. The same idea is also described in (Alitappeh and Jeddisaravi, 2022), where the authors mention the usage of the MTSP for solving various task allocation problems, including coverage by multiple *Uncrewed Aerial Vehicles* (UAVs) (Ann et al, 2015).

The MTSP is further benchmarked in multi-robot exploration scenarios by Faigl and Kulich (2015) in a cluster-first, route-second manner, where potential exploration waypoints are clustered by K-means clustering first. Then, each robot selects the next exploration waypoint from the given cluster based on the solution of the TSP. Another benchmarked method uses a solution to the task-allocation problem using the Hungarian algorithm to assign robots directly to the potential exploration waypoints. Based on (Faigl and Kulich, 2015), the Hungarian algorithm and MTSP formulation of the multi-robot exploration problem lead to the overall best exploration performance among the evaluated methods. However, these methods are used in a centralized manner (Queralta et al, 2020) or require sharing dense environment models between the robots.

The coordination policy can also be learned through training, while having global knowledge about the environment to assess the quality of the actions taken. An example of such an approach is presented in (Westheider et al, 2023), where the authors propose training a coordination strategy for multiple UAVs to create a map of environments collectively. The authors used deep reinforcement learning with rewards based on the reduction of map entropy and demonstrated their approach in simulations using both synthetic and real-world data. The approach is for obstacle-free environments, thus not suitable for cluttered underground environments. A more suitable approach for environments with obstacles is proposed in (Tan et al, 2022), where the authors present a promising approach using a deep reinforcement learning model for decentralized multi-robot exploration of environments where robots may exchange maps and positions. The authors design a reward function to maximize the joint area explored while minimizing local interactions, distance traveled, and time to completion.

As for deciding which robot should follow which waypoint, we can distinguish between centralized and decentralized approaches. The centralized approaches rely on collecting maps from the robots to allocate tasks for each robot. Hence, it can be ineffective when communication is unreliable, such as when messages are not received within a few dozen seconds. Decentralized coordination methods might also use available information from other robots to estimate the behavior of the other robots as accurately as possible to make good decisions. If information from some robots is not precise or is delayed, the quality of the decisions decreases, but some decisions can still be made.

An example of the decentralized approach that considers the amount of data exchanged between the robots is described by Batinović et al (2020). The authors show a method that exchanges robots' positions and locations of the potential exploration waypoints between the robots. The method utilizes the Hungarian algorithm to assign potential exploration waypoints to the robots. However, the approach is tested only in a very small-scale environment with a few crossroads. Nevertheless, a considerable advantage of using the Hungarian algorithm on each robot is that it can utilize the last known states of the other robots to estimate their decisions without relying on explicitly exchanging messages. The need for explicit message exchange during the task allocation procedure increases the time required for a decision, especially when communication is unreliable. A representative approach of the explicit message exchange is a market-based approach, such as (Hollinger and Smith, 2018) with a single-bid local auction.

The MinPos method (Bautin et al, 2012) is another example of the decentralized decision-making method, where each robot makes decisions individually but considers the possible decisions of the other robots. MinPos is based on ranking possible further exploration waypoints to evaluate their importance with respect to (w.r.t.) each robot r. The robot then chooses the waypoint with the lowest rank. In (Bautin et al, 2012), the rank $G_{i,\mathrm{MinPos}}$ of the ith exploration waypoint for the robot r is computed as

$$G_{i,\mathrm{MinPos}} = \sum_{\forall R_k \in (\mathbf{R}\setminus\{r\}), C_{k,i}<C_{r,i}} 1, \tag{1}$$

where $\mathbf{R}$ is the set of all robots, and $C_{k,i}$ is the path length between the kth robot and the ith possible exploration waypoint. However, the computation of $C_{k,i}$ requires sharing maps and waypoints between the robots. Nevertheless, a more recent work (Wu and Luo, 2022) shows a satisfactory performance of the MinPos when used under unreliable communication.

The proposed Cross-rank-based methods consist of two steps. In the first step, waypoint candidates are selected based on the Cross-rank to distribute robots at the crossroads. The second step is used when there are multiple candidates for the next exploration waypoint. MinPos inspired the design of the second steps for proposed methods CRSR^{m,p} and CRESR^{p} to use path lengths $C_{k,i}$ and their approximation, respectively. The last proposed method CRTSP^{p} has the second step based on solving the TSP instance to determine the order in which all possible waypoints can be visited, similar to the second step of the MTSP-based approach (Faigl and Kulich, 2015). Nevertheless, the first steps of the proposed methods are based on different ideas than in (Wu and Luo, 2022; Faigl and Kulich, 2015).

Horyna et al (2023) propose to coordinate the robots based on exchanging their positions using ultraviolet direction and ranging, which completely avoids radio communication. However, it requires the robots to be in the line of sight, thus making it unsuitable for a cluttered environment, such as underground. Burgard et al (2000) address a low-bandwidth communication by sharing maps represented by sets of polygons that are more memory efficient than grid maps. Schulz et al (2019) propose to use the occupancy *Normal Distributions Transform* (NDT) as a representation with low memory footprint, shareable via a network with the bandwidth of 37.5 kB s^{-1}. The authors report that at least 4.1 kB s^{-1} of the bandwidth has been used, which is more than the communication bandwidth available in our motivational scenario. Therefore, based on the literature review, we propose a novel method capable of multi-robot coordination based only on the robots' shared locations to fit the target communication bandwidth. Although methods including MinPos, MTSP, or task-allocation using the Hungarian algorithm do not fit bandwidth requirements, even when compression of the shared data is used, we use them in the presented comparison for reference.

3 Problem Statement

Since we are focused on the decentralized coordination of homogeneous multi-robot teams, an identical exploration framework runs on all robots. The architecture of the framework consists of two parts, as depicted in Fig. 2. Its central part is the "Coordination algorithm," which can be one of the existing methods evaluated or the proposed one. The second part is responsible for autonomous navigation, which consists of sensing, localization, an environment model, and navigating the robot toward the selected waypoint.

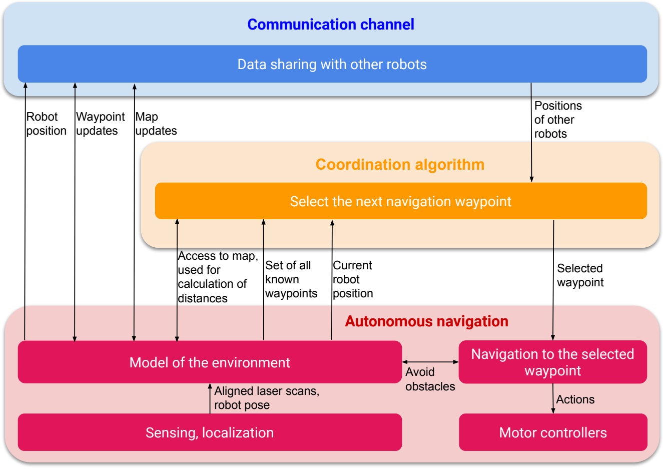

Communication channel
Data sharing with other robots
Robot position
Waypoint updates
Map updates
Positions of other robots
Coordination algorithm
Select the next navigation waypoint
Access to map, used for calculation of distances
Set of all known waypoints
Current robot position
Selected waypoint
Autonomous navigation
Model of the environment
Avoid obstacles
Navigation to the selected waypoint
Aligned laser scans, robot pose
Actions
Sensing, localization
Motor controllers

Fig. 2 Connections between the coordination algorithm and the rest of the exploration system.

Each coordination method can share a specific set of data with other robots. Although the shared data might be used in different ways depending on the coordination method used, the process of generating the data from sensory measurements is identical for all the evaluated methods. Three inputs from the navigation stack to a coordination method are (i) map updates, (ii) updates of the exploration waypoints, and (iii) robot position provided by the localization system.

The coordination algorithm outputs a single waypoint toward which the robot is navigated. Coordination algorithms that require calculating distances between the waypoints and robots do that through planners accessing both parts of the environment model: the global map and local map covering the robot's surroundings. The robot stops its exploration when no further exploration waypoints are available. The underlying navigation method, used for all the methods under the test, is detailed in Section 4.

The following assumptions are made to limit the scope of the paper and focus the presentation on the proposed approaches to the coordination mechanism.

1. The robots are localized within the same coordinate frame established at the beginning of the mission, where the individual localization systems are initialized to the same origin. *It allows sharing positions between the robots without the need for additional transformations between coordinate frames.* See (Bayer et al, 2023) or Section 4 for an overview of how such a joint coordinate frame can be established.
2. Corridors are considered sufficiently wide so that the robots cannot block each other while passing through a corridor. *It allows for simplifying navigation and path planning.*
3. The sensor equipment of the robots has similar characteristics, including sensory range, the density of the measurements, and field of view. *It unifies reasoning about further exploration waypoints detected from a similar distance for all the robots.*

The assumptions might be relaxed at the cost of more complex transformations, planning, and sensor models.

3.1 Performace Indicators

In the presented work, we compare the evaluated multi-robot coordination methods by the following performance indicators.

- **Exploration time** is the main performance indicator of the exploration system, also used in (Batinović et al, 2020) and (Qin et al, 2019). The exploration time t_j for the jth trial measures the time from the start of the mission triggered by the event when the union of the areas explored by the individual robots covers all reachable parts of the environment. Thus, none of the robots has to explore the whole environment by itself. For comparing the coordination methods using the same underlying navigation stack, the strategy yielding a lower time required to explore the environment is considered better. The overall performance of multi-robot exploration is indicated by the average time t_{avg} required to explore the environment and its standard deviation from all the performed trials c_{trial} for the particular given scenario and particular method.

$$t_{\text{avg}} = \frac{1}{c_{\text{trial}}} \sum_{j=1}^{c_{\text{trial}}} t_j. \tag{2}$$

Note that it is assumed that the robots have omni-sensing capabilities. Therefore, a place is considered explored by a robot if the distance between the place and the robot is shorter than the coverage distance d_{cov}. The particular value of d_{cov} might not necessarily be the longest sensor range. Its value can be motivated by tasks where robots, in addition to capturing 3D models of the environment, also collect other data from the environment, such as artifact recognition in the DARPA SubT Challenge. The robots can use cameras to detect artifacts, for which the distance required for the camera to detect an artifact is shorter than the range of the LiDAR sensors used for creating a spatial model or obstacle avoidance. Therefore, it can still be required to get closer to the area to ensure artifacts were not detected by the camera, since they were too far from the robot, despite the fact that a robot can already cover the specific area with its LiDAR from a distance. In the case that the only purpose of the exploration is the resulting spatial model, d_{cov} can be set to the used LiDAR sensing range, and the exploration can behave similarly to spatial frontier-based exploration.

- **Distance traveled by robots** is the indicator used similarly as in (Wang et al, 2019). More specifically, we compute the average length of the longest exploration path $l_{\text{avg,max}}$ based on (Faigl and Kulich, 2015):

$$l_{\text{avg,max}} = \frac{1}{c_{\text{trial}}} \sum_{j=1}^{c_{\text{trial}}} \max(l_{j,1}, l_{j,2}, \ldots, l_{j,n}), \tag{3}$$

where $l_{j,i}$ is the length of the traveled ith robot's path during the jth trial repetition, c_{trial} is the number of repetitions, e.g., for five trials used, $c_{\text{trial}} = 5$. Moreover, we also compute the sum of the distances traveled by all the robots in all performed trials l_{sum}, to provide an absolute value of the traveled distance in the exploration:

$$l_{\text{sum}} = \sum_{j=1}^{c_{\text{trial}}} \sum_{i=1}^{n} l_{j,i}. \tag{4}$$

- **The average coverage** is the average area covered by individual robots during the whole exploration mission. It evaluates the ability of the coordination methods to distribute robots within the environment. The average coverage for a team of n robots is calculated as

$$\text{cov}_n = \frac{1}{c_{\text{env}} c_{\text{trial}} n} \sum_{i=1}^{c_{\text{env}}} \sum_{j=1}^{c_{\text{trial}}} \sum_{k=1}^{n} \frac{S_{i,j,k}}{S_i}, \tag{5}$$

where c_{env} is the number of exploration environments ($c_{\text{env}} = 5$ is used), $S_{i,j,k}$ is the traversable area covered by the kth robot using its sensor with the range d_{cov}, and S_i is the total traversable area of the ith environment. When the count of the robots n is taken into account, cov_n should ideally be close to $1/n$. Nevertheless, an overlapped coverage is expected, especially in underground scenarios, where the robots start from the same area, and parts of the environment are reachable only by a single or limited number of passages.

- **Progress of the environment coverage** is an indicator also used in the literature, such as (Shrestha et al, 2019). By the progress of the coverage, the coordination methods can be evaluated qualitatively, showing how area coverage for each robot and the joined coverage are increasing until the whole environment is covered.
- **Estimated required bandwidth** is the indicator to compare the coordination methods regarding the amount of data broadcasted from one robot to the rest of the robotic team.

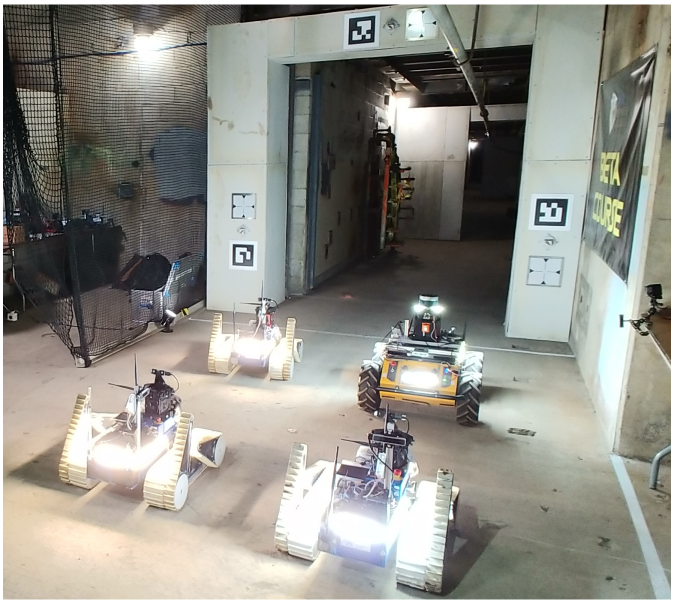


Fig. 3 The initialization process of the coordinate frames of the robots using a gate with various types of markers before the mission of the DARPA SubT Challenge.

4 Autonomous Navigation

Processing the sensory data and robot navigation toward the waypoint selected by one of the studied coordination methods is achieved by the navigation stack, identical for all the coordination methods, marked as “Autonomous navigation” in Fig. 2. The methods employed consist of localization and mapping, environment model building, path planning, and path following. Note that the detailed description of all the employed methods is out of the paper’s scope. Therefore, we highlight the main features and design choices. An interested reader can find further details in (Bayer et al, 2023).

Localization is solved by each robot independently; however, the important part of the system deployment is an initialization of the robots’ coordinate frames at the beginning of the mission. The initialization is crucial because it allows the robots’ positions, maps, and waypoints to be shared within the same global coordinate frame. That is the reason why such initialization is performed in the DARPA SubT Challenge (Bayer et al, 2023) using an entrance gate to the mission area; see a snapshot of the mission start in Fig. 3. After initialization of the robots’ coordinate frames, the robots use a localization system based on (Pomerleau et al, 2013) with various improvements, including deskewing (Deschênes et al, 2021) to localize themselves within the environment.

The localization system aligns scans from the robot’s range sensor. The aligned scans are used to build an environment model. Since the studied problem is motivated by large environment exploration, already addressed within the DARPA SubT Challenge (Chung et al, 2023), therein developed solutions might be used, such as (Bouman et al, 2020). Nevertheless, the model of the CERBERUS team (Dang et al, 2020) uses a precise local map for optimizing safe local plans and sparse global graphs that allow the robot to reposition to some previously visited locations, which is used by their exploration strategy (Kulkarni et al, 2022). Also, the CTU-CRAS-NORLAB team used a sparse global graph to represent already visited traversable areas and a dense local planning graph to avoid obstacles and optimize path cost (Bayer and Faigl, 2021). Hence, for the presented work, we opted for a model that combines (Dang et al, 2020) and (Bayer and Faigl, 2021).

The model is composed of two parts: a global sparse graph representing the topology of the traversable areas and a local 3D grid map $\mathcal{M}$ of the environment (Bayer et al, 2023). The local map represents the surroundings of the robot within the radius $\mathcal{M}_{\text{rad}}$ by occupancy probability and the shape of the terrain. For each cell of the 3D local map, the traversability based on the geometrical properties of the terrain, including step height and terrain slope, is computed. Then, a cell is considered traversable if both the slope of the terrain and the terrain step within the robot radius r_{rad} are below the thresholds $\mathcal{M}_{\text{slope}}$ and $\mathcal{M}_{\text{step}}$, respectively. Each traversable cell forms a vertex in the planning graph, and each of the two neighboring vertices defines an edge for the planning graph connecting the respective vertices. Such a local planning graph is merged with the global planning graph.

The global planning graph is used to estimate the distance between two locations, which is essential for the tested coordination methods when exploration waypoints are outside the robot's surroundings. The global graph is a sparse representation of an environment that is incrementally constructed from the local planning graph, where from each new local planning graph, obtained when the robot updates its local map, only vertices that are farther than the radius of the robot's circumference are added and connected to the global graph. If a map is shared with the robot from another robot, then the shared map is used to build a local graph connecting traversable vertices, which is then merged into the global graph in the same way as the robot's local planning graph. Note that the global planning graph is also used to navigate the robot when the navigation waypoint is located outside the local area.

New possible exploration waypoints are identified based on the measured information gain as follows. For a given cell, the information gain measures how much of the not yet observed spatial model can be covered if the robot reaches the given cell. For computing the coverage, it is assumed that the robot is equipped with an omnidirectional sensor with the range d_{cov}.

Having the determined waypoints, they are clustered as in (Faigl and Kulich, 2013). So, reachable cells with non-zero information gain are clustered, and the cluster representants are used as the next potential exploration waypoints. The navigation along the plan to the exploration waypoint selected by the particular coordination algorithm is executed by employing the model predictive control for path following. The parameters of the navigation system used for the reported results from the simulations and real-world deployment are summarized in Table 1.

Table 1 Main parameters of the navigation system

Parameter	Environment	
	Simulated	Real-world
Radius of the robot r_{rad} [cm]	40	28
Robot max speed $v_{r,\max}$ [m s^{-1}]	1.2	0.7
Environment coverage distance d_{cov} [m]	4	6
Local map size $\mathcal{M}_{\mathrm{rad}}$ [m]	13	8
Local map resolution $\mathcal{M}_{\mathrm{res}}$ [cm]	5	4.5
Terrain max traversable step $\mathcal{M}_{\mathrm{step}}$ [cm]	24	24
Terrain max traversable slope $\mathcal{M}_{\mathrm{slope}}$ [°]	20	40

5 Coordination Methods

Based on the literature review, we selected representative methods that have been implemented and compared with the proposed methods. All the coordination methods implement an interface for the decentralized coordination algorithm shown in Fig. 2. Thus, a method that coordinates the multi-robot team is running in parallel on all robots, and the only way a robot may affect the behavior of others is through shared data that can be map updates, shared waypoint set updates, or by sharing its position. The coordination method is triggered to select the next exploration waypoint when any of the following cases occur.

1. At the exploration mission start.
2. If the exploration waypoint is no longer valid since the area it represents is considered covered by the robot's sensors or other robots' sensors when the information about the waypoints and robots' positions is shared.
3. Robot reached the waypoint, or the waypoint became unreachable.

In the rest of the section, we briefly summarize the evaluated multi-robot coordination methods together with their data-sharing requirements.

- **Closest** is the simplest method used in the comparison that is based on the single-robot approach (Yamauchi, 1997), where the robot is navigated to the closest exploration waypoint while no data are shared with other robots.
- **Closest**^{m,w} is based on (Yamauchi, 1998), where the robots still greedily select the closest waypoints. However, the robots share information about maps (denoted by the superscript symbol ^{m}) and exploration waypoint locations (denoted by the superscript symbol ^{w}).
- **TSP** is based on selecting the next exploration waypoint by utilizing a solution of the TSP, assuming that the robot is going to visit all possible exploration waypoints in the order given by the solution of the respective TSP instance. The method does not share any data with other robots.
- **TSP**^{m,w} is a variant of the TSP, but it shares robot positions, map updates, and waypoints.
- **ETSP** and **ETSP**^{m,w} are two evaluated methods that are similar to the TSP and TSP^{m,w} methods, respectively. However, they use the Euclidean distance between waypoints and between the waypoints and the robot position instead of the distance matrix based on the lengths of planned paths between the waypoints and between the waypoints and the robot position. Using

Euclidean distance saves computational resources when computing expected travel costs between the waypoints. Nevertheless, path planning using an available map is performed when a robot is navigated to a selected waypoint.

- **Hun**^{m,w,p} is our implementation of the Hungarian algorithm for task allocation based on (Batinović et al, 2020). The assignment optimizes the sum of the path lengths required for n robots to reach k different waypoints. Dummy waypoints or virtual robots are added to the instance of the task allocation problem when $n \neq k$. Similar to method Closest^{m,w}, the Hungarian algorithm uses a short decision horizon, not taking into account the steps required after reaching the selected waypoint. The main difference is that by employing Hun^{m,w,p}, the robot also takes into account the decisions of other robots. Hun^{m,w,p} requires sharing waypoint locations, map updates, and robot positions (denoted by the superscript symbol ^{p}).
- **MTSP**^{m,w,p} is based on (Faigl and Kulich, 2015) with K-means clustering to allocate a set of waypoints to each robot and then use the solution of the TSP to select the next exploration waypoint. Like the Hungarian algorithm, MTSP^{m,w,p} shares waypoints, map updates, and robot positions.
- **MinPos**^{m,w,p} (Bautin et al, 2012) is the last implemented method from the literature. Although the decision on the next exploration waypoint for each robot is decentralized using the computed rank of each possible waypoint, the method shares robot positions, waypoints, and map updates that are utilized in rank computation.
- **CRESR**^{p} is one of three variants of the proposed method that operates in two steps. In the first step, a subset of all possible exploration waypoints is obtained as waypoints to distribute the robots at crossroads. Then, the next exploration waypoint is selected from the subset to spread the robots as far from each other as possible using Euclidean distances between the robots.
- **CRSR**^{m,p} is proposed to demonstrate the effect of using Euclidean distance when spreading the robots. Thus, the CRSR^{m,p} variant of the proposed method uses "true" distances between the robots instead of Euclidean distances when spreading the robots. The distances are computed as path lengths found within the available maps. Hence, its disadvantage w.r.t. CRESR^{p} is the requirement on the increased bandwidth to share map updates ^{m} in addition to the robot positions ^{p}.
- **CRTSP**^{p} is a proposed alternative to show further properties of CRESR^{p} and the effects of ordering waypoints based on solving instances of the TSP. Thus, it uses the identical first step as CRESR^{p}, but then it uses a solution of the TSP to select the exploration waypoint from the subset of waypoints obtained by the first step, which is inspired by (Kulich et al, 2011).

Table 2 Data sharing requirements for the coordination method (exploration strategies), sorted from the most communication demanding strategy

Exploration	Type of shared data			Bandwidth
strategy	Maps	Waypoints	Positions	used [kB s^{-1}]
MTSP^{m,w,p}	✓	✓	✓	23.6
MinPos^{m,w,p}	✓	✓	✓	23.6
Hun^{m,w,p}	✓	✓	✓	23.6
Closest^{m,w}	✓	✓	–	23.5
TSP^{m,w}	✓	✓	–	23.5
ETSP^{m,w}	✓	✓	–	23.5
CRSR^{m,p}	✓	–	✓	23.5
CRTSP^{p}	–	–	✓	0.1
CRESR^{p}	–	–	✓	0.1
Closest	–	–	–	0.0
TSP	–	–	–	0.0
ETSP	–	–	–	0.0

5.1 Data Sharing Requirements

Data needed to be shared between the robots is summarized for each studied multi-robot exploration strategy in Table 2. The bandwidth required for each strategy is calculated as the sum of the data that are broadcast from each robot to share robot positions, waypoints, or maps, as the particular method needs. In particular, the position sharing is calculated as $b_{\text{positions}} = f_{\text{positions}}(12+1) = 5 \cdot 13 = 65\,\text{B}\,\text{s}^{-1}$, where $f_{\text{positions}}$ is the frequency of positions broadcasting from each robot and 12 B is the size of three floats representing the robot position in xyz coordinates, and 1 B is reserved for the robot identifier. The estimation of the bandwidth requirement for map sharing was measured during the simulated experiments as the average bandwidth $b_{\text{maps}} = 23.5\,\text{kB}\,\text{s}^{-1}$. Note that the relatively low required bandwidth to share maps is caused by compressing them and by sharing only map changes with the other robots. Since the sensory range of the robots is limited, and the robot speed is limited to $1.2\,\text{m}\,\text{s}^{-1}$, the rate of newly covered parts of the map is bounded, and thus, the amount of data for map updates is limited as well. The bandwidth requirements do not increase with the total size of the built map. Since the robots handle unknown space as untraversable, they broadcast to others only traversable parts of the environment or changes between traversable and untraversable areas, further decreasing the bandwidth requirements.

Waypoints are shared by broadcasting changes in waypoint sets by each robot, leading to an average required bandwidth of about $b_{\text{waypoints}} = 66\,\mathrm{B\,s^{-1}}$. The waypoints ^{w} are shared with the map updates ^{m} to enable the determination of how to reach each waypoint and thus determine the expected travel cost. All data shared between the robots broadcasts changes to the models or data with a fixed size. Thus, bandwidth requirements do not grow with the duration of the exploration mission. On the other hand, the bandwidth requirements are proportional to the velocities of the robots $v_{r,\max}$, which does not exceed $1.2\,\mathrm{m\,s^{-1}}$ in the present work. The effect is caused by the fact that when a robot moves faster, it needs to broadcast more map and waypoint changes per second. A similar principle holds for sharing the robots' positions since faster-moving robots should proportionally increase the frequency of position broadcasting so that the information received by the other robots would not be too sparse for making decisions.

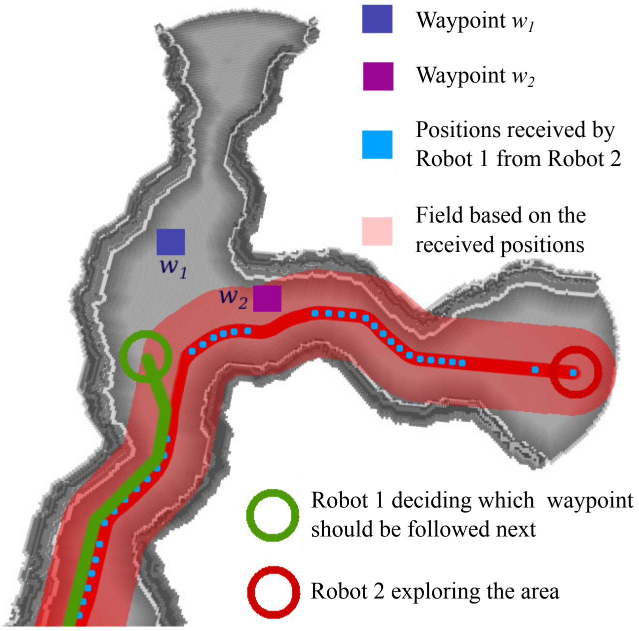
6 Proposed Coordination Methods

The three proposed methods are composed of two steps. In the first step, the developed Cross-rank is determined for all waypoints $\mathbf{W}$ that are known to the particular robot. The first step is identical to all three variants of the Cross-rank-based methods. However, in the second step, the next exploration waypoint is selected from the waypoints with the lowest Cross-rank based on the potential to spread the robots using true/approximated distances (Spread-rank) or a solution of the TSP instance. The steps are further detailed in the following parts of the section.

6.1 First Step: Cross-Rank Computation

The Cross-rank assigns non-negative integer $G_{i,\text{cross}}$ for each ith waypoint $\mathrm{w}_i \in \mathbf{W}$. The set of all n robots $\mathbf{R}$ is split into the actual robot r, for which the rank is being computed, and the set of the remaining $n-1$ robots $\mathbf{R}' = \mathbf{R} \setminus \{r\}$. The Cross-rank is designed to spread the robots at the crossroads uniformly by penalizing waypoints that are close to the locations visited by the other robots $\mathbf{R}'$.

Under unreliable communication, there can be areas with sparse information about other robots. Hence, the robot's traveled path is approximated by line segments connecting the received robot's positions; see Fig. 4. For the robot r, such a polyline is used to establish a corridor along the robot's path using a distance threshold m_r representing an area that is considered visited by r. For each robot $r'_k \in \mathbf{R}'$, there is an ordered set of received positions $\mathbf{H}_k$ and the Cross-rank $G_{i,\text{cross}}$ for the waypoint w_i is computed as



Waypoint w_1
Waypoint w_2
Positions received by Robot 1 from Robot 2
Field based on the received positions
w_1
w_2
Robot 1 deciding which waypoint should be followed next
Robot 2 exploring the area

Fig. 4 An illustrative example where Robot 1 decides which waypoint should be selected next. Based on the term $G_{i,\text{cross}}$, Robot 1 selects to follow waypoint w_1 because waypoint w_2 is closer to the estimated path of Robot 2 than m_r distance. The path of Robot 2 is estimated using the positions received by Robot 1. In the figure, the space closer than m_r to the estimated path of Robot 2 is visualized by the red field. Note that some messages containing the position of Robot 2 might be lost or delayed. Thus, the distance between positions received by Robot 1 from Robot 2 varies.

$$G_{i,\text{cross}} = \sum_{k=1}^{n-1} g_{\text{cross}}(\mathbf{p}_i, \mathbf{H}_k), \tag{6}$$

where $g_{\text{cross}}(\mathbf{p}_i, \mathbf{H}_k)$ is

$$g_{\text{cross}}(\mathbf{p}_i, \mathbf{H}_k) \begin{cases} 1 & \texttt{if } |\mathbf{H}_k| > 1 \text{ and} \\ & \quad m_r > d_{\text{lseg}}\left(\mathbf{p}_i,\ \overline{\mathbf{s}}(k, \mathbf{p}_i)\right) \\ 1 & \texttt{if } |\mathbf{H}_k| == 1 \text{ and} \\ & \quad m_r > \| \mathbf{p}_i - \mathbf{h}_{k,1} \| \\ & \qquad\qquad\qquad \mathbf{h}_{k,1} \in \mathbf{H}_k \\ 0 & \texttt{Otherwise} \end{cases} . \tag{7}$$

In (7), the function $d_{\text{lseg}}(\mathbf{a}, \overline{\mathbf{b}})$ measures distance between the position $\mathbf{a}$, and line segment $\overline{\mathbf{b}}$, $\mathbf{p}_i$ is the position of the ith waypoint, and the function $\overline{\mathbf{s}}(k, \mathbf{p}_i)$ is to

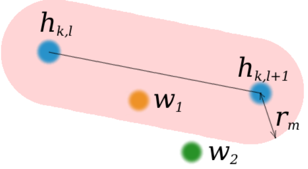


Fig. 5 Visualization of the Cross-rank computation for a line segment formed between two consecutive positions $\mathbf{h}_{k,l}$ and $\mathbf{h}_{k,l+1}$ received from the robot r'_k. The Cross-rank of the waypoint w_1 is increased since it is closer than r_m to the line segment representing the traveled path of another robot. On the other hand, the Cross-rank of the waypoint w_2 is not increased since w_2 is further than r_m from the line segment.

find the closest line segment from the kth robot to the position $\mathbf{p}_i$, such that

$$\bar{\mathbf{s}}(k, \mathbf{p}_i) = \underset{\overline{\mathbf{h}_{k,l}\mathbf{h}_{k,l+1}} \text{ for } \mathbf{h}_{k,l}\mathbf{h}_{k,l+1} \in \mathbf{H}_k}{\operatorname{argmin}} d_{\text{lseg}}(\mathbf{p}_i, \overline{\mathbf{h}_{k,l}\mathbf{h}_{k,l+1}}), \tag{8}$$

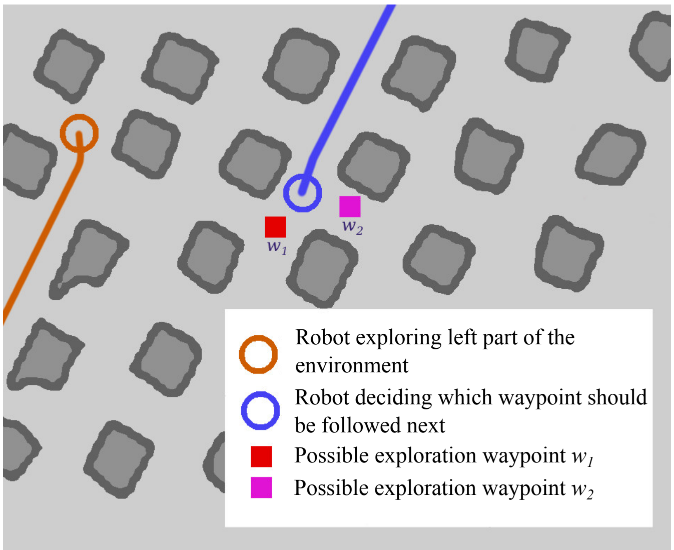
where $\overline{\mathbf{h}_{k,l}\mathbf{h}_{k,l+1}}$ denotes line segment connecting $\mathbf{h}_{k,l}$ and $\mathbf{h}_{k,l+1}$. The thresholding parameter m_r limits the distance from the line segments at which Cross-ranks of waypoints are increased; see Fig. 5. Suitable threshold m_r values have been analyzed empirically. The value of m_r should always be higher than half of the robot's largest diameter and always smaller than the sensory range. For all the robots in the team, the value of m_r is the same since we assume identical sensory systems. Furthermore, for subterranean environments, the suitable value of m_r should be set between half of the tunnel width and the distance between tunnel entrances at the crossroads if these parameters are known.

The proposed Cross-rank can be implemented in multiple ways depending on the frequency of the shared robots' positions. The Cross-rank uses $d_{\text{lseg}}(\mathbf{a}, \overline{\mathbf{b}})$ to compute the distance to a line segment since known locations of other robots can be far from each other because of unreliable communication, or fast motions of the robots. Suppose the positions are received frequently and reliably. In that case, the computation of the distance to a line segment can be simplified to the determination of the distance to the closest position of another robot.

The distance to the closest point calculation or determination of the closest segment requires finding the nearest neighbors, which can be implemented naively. However, in large exploration scenarios, robots might receive thousands or even tens of thousands of positions from other robots during the mission. Therefore, we recommend employing more sophisticated implementations, such as the FLANN library (Muja and Lowe, 2014).

6.2 Second Step: Waypoint Selection

Since the Cross-rank assesses all the possible exploration waypoints by a non-negative integer, the next exploration waypoint is selected using the proposed Spread-rank (methods CRESR^p and CRSR^{m,p}) or as the first waypoint of the solution of the TSP instance to visit all the waypoints. The Cross-rank assessed waypoints are combined into the set $\mathbf{W}_{\text{cr}}$ from which the next waypoint is selected as follows.



w_1
w_2
○ Robot exploring left part of the environment
○ Robot deciding which waypoint should be followed next
🟥 Possible exploration waypoint w_1
🟪 Possible exploration waypoint w_2

Fig. 6 A robot (in blue) decides where to explore next, while another robot explores the left part of the environment. Thus, based on the term $G_{i,\text{Spread}}$ only, the robot selects the waypoint w_2, which is farther from the area being explored by the other robot, marked by a brown color.

Spread-rank $G_{i,\text{Spread}}$ is designed to get the rth robot as far from the other robots as possible based on the most recent positions of the robots. The ranking is inspired by the MinPos (Bautin et al, 2012) that counts how many of the robots $\mathbf{R}'$ have a shorter path to the ith waypoint than the actual robot r. The proposed ranking further takes into account ratios of distances between the waypoints and the robots. The idea of the Spread-rank $G_{i,\text{Spread}}$ is visualized in Fig. 6 and it is computed as

$$G_{i,\text{Spread}} = \sum_{k=1}^{n-1} \frac{C_{r,i}}{C_{k,i}}, \tag{9}$$

where $C_{r,i}$ is the distance between the position of the robot that is making the decision and the ith waypoint that is being ranked, $C_{k,i}$ is the distance between the kth robot from $\mathbf{R}'$ and the ith waypoint.

The calculation of $C_{k,i}$ requires sharing maps between the robots and Spread-rank using $C_{k,i}$ is implemented in the proposed method CRSR^{m,p}. However, under low bandwidth communication, detailed maps for the computation of shortest paths might not be available. Therefore, we propose CRESR^{p} where the path lengths $C_{r,i}$ and $C_{k,i}$ are approximated by Euclidean distances, which does not require sharing maps between the robots.

The resulting next exploration waypoint is selected as the waypoint from the set $\mathbf{W}_{\mathrm{cr}}$ with the lowest rank $G_{i,\mathrm{Spread}}$. If multiple waypoints have the lowest Spread-rank, the resulting waypoint is selected based on the path length from the waypoint to the current position of the robot.

Finally, we propose to evaluate waypoint selection based on solving an instance of the TSP (Kulich et al, 2011) to visit waypoints $\mathbf{W}_{\mathrm{cr}}$. The method is denoted CRTSP^{p}, and since it uses the TSP on $\mathbf{W}_{\mathrm{cr}}$ waypoints instead of $G_{i,\mathrm{Spread}}$ with path lengths to other robots, it does not require sharing maps and waypoints between the robots.

6.3 Computational Complexity

The computational complexity of determining the Cross-rank and Spread-rank is proportional to all the received positions from the robots $\mathbf{R}'$ stored in the sets $\mathbf{H}_k$, which grow over the mission. Therefore, the growth is decreased by using a spatial filter, which outputs a new robot position only if the robot has moved far enough from the previous location. Furthermore, the computational requirements are impacted by computing paths between the robots' positions and waypoints.

When compared with the most similar ranking-based approach MinPos^{m,w,p}(1), its complexity is proportional to the number of robots n and the determination of the path lengths from the robots to the waypoints. Thus, also in the MinPos^{m,w,p}, the actual computational requirements depend on the particular course of the mission. Since we avoid sharing maps using Euclidean distances in the proposed CRESR^{p}, it further reduces the computational burden. In all evaluated scenarios reported in the following section, the MinPos^{m,w,p} is significantly more demanding than the proposed Cross-rank combined with the Spread-rank using Euclidean distances of the CRESR^{p}.

7 Results

The performance of the nine selected existing methods and three proposed decentralized multi-robot coordination methods has been empirically evaluated in simulation environments that are reported in Section 7.1. Besides, the proposed Cross-rank-based coordination has been experimentally deployed with a real multi-robot system using three quadruped robots. The experimental results are presented in Section 7.2. A discussion of the reported results is presented in Section 8. The parameterization of the employed autonomous navigation stack is listed in Table 1. The proposed Cross-rank methods are parameterized with $m_r = 1.5\,\mathrm{m}$ and $m_r = 5.0\,\mathrm{m}$ for the simulated and real-world deployments, respectively.

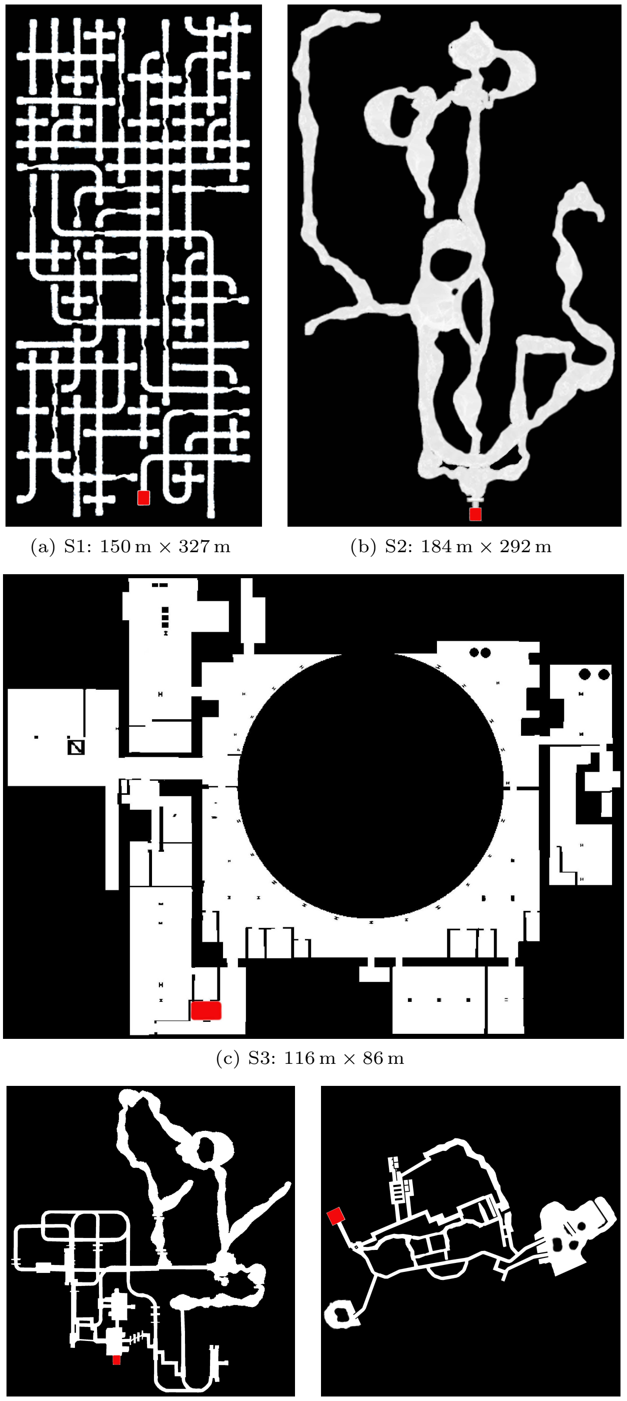
7.1 Results in Simulated Environments

All 12 coordination methods, as listed in Section 5, are qualitatively evaluated in five testing environments depicted in Fig. 7. The focus is on teams with 3 and 5 robots selected w.r.t. the size of the environments based on the practical deployment experience (Bayer et al, 2023). Besides, we further include scalability analyses using 7 and 9 robots in Section 7.1.4, the impact of the localization error on the exploration performance in Section 7.1.5, and using heterogeneous robots in Section 7.1.6.

The five different environments, each with the robots' starting location, define evaluation scenarios designed to force the coordination methods for a significant number of decisions, supporting the relevance of the results. Robots need to travel hundreds of kilometers to explore all relatively large environments multiple times. Therefore, the simulation setup is set to simulate the robot's motion, its collisions with the environment, and its primary LiDAR sensor. The robot motion is simulated using a kinematic unicycle model (Luca et al, 2005), where applied forward and angular velocities are affected by 5 % simulation random error drawn from a normal distribution.

The reflections of the LiDARs from the environment are simulated by raycasting LiDAR rays to 3D models of the environment. The 3D models of the environments are built based on the outlines shown in Fig. 7. The sensory noise of the LiDAR is based on the parameters of the Ouster OS0 LiDARs (Ouster OS0, accessed Jan 11, 2024) and experiments with a real device. An example of the simulated LiDAR data produced based on a 3D scene is visualized in Fig. 8.

The selected environments are based on the scenarios used in the virtual circuits of the DARPA SubT Challenge and two systems circuits, namely in the Urban and Final



(a) S1: 150 m × 327 m

(b) S2: 184 m × 292 m

(c) S3: 116 m × 86 m

(d) S4: 117 m × 126 m

(e) S5: 161 m × 105 m

Fig. 7 The environments used for the simulation trials and their sizes. The red areas indicate the initial positions of the robots.

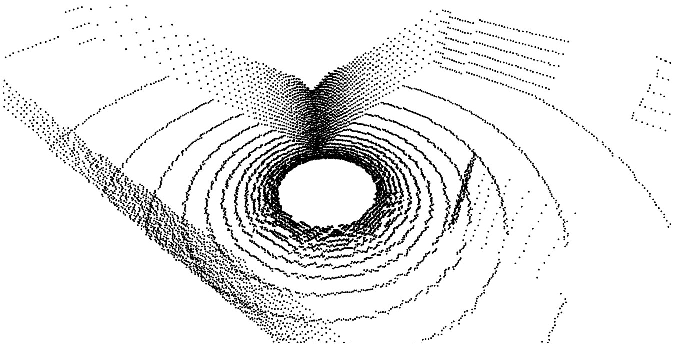


Fig. 8 Scan from a simulated model of the omnidirectional Ouster OS0-32 LiDAR with 32 layers (beams) and noise in the range measurements.

circuits, here denoted as S3 and S5 scenarios, respectively. For each environment, two scenarios are evaluated with 3 and 5 deployed robots. In each scenario, the robots' starting area is defined, shown as the red area in Fig. 7. The robots are deployed from the starting area with 15 s delay.

Note that the robots' poses are provided directly by the simulator instead of the localization system from the navigation stack described in Section 4. It saves computational requirements and also ensures that evaluation is focused on coordination and not on the performance of the localization system. However, the simulation still includes a certain level of randomness induced by simulated sensory noise and imprecision in executing control actions. Therefore, each scenario is repeated five times, and $12 \cdot 5 \cdot 5 \cdot 2 = 600$ evaluation trials were run in total.

By the described simplifications, a simulation real-time factor of 100 % is achieved while still having sufficient computational resources to run all components of the considered autonomous navigation systems. Nevertheless, running all the simulation trials took approximately two weeks with computational resources consisting of four Intel i7-10710U computers, each with 32 GB RAM running simulations scenarios with three robots, and AMD Ryzen R5 3600 with 32 GB RAM running simulations with five robots.

The methods are evaluated using the performance indicators described in Section 3.1. The evaluation results w.r.t. the individual performance indicators are presented in Sections 7.1.1, 7.1.2, and 7.1.3, which are followed by scalability analysis, localization error impact study, and influence of heterogeneous robots.

7.1.1 Evaluation Results: Exploration Time

Evaluation results on the average required time t_{avg} to explore the environments are depicted in Table 3. In the presentation of the results, we consider the methods ordering according to the *Time-performance* score denoted **TP score**. The score is calculated as the number of times any other methods have outperformed the given method for all 12 methods, 5 environments, 2 sizes of robotics teams, and 5 repetitions. For example, the score 9 indicates that in 9 cases, other methods performed with shorter t_{avg} in the respective scenarios. Thus, a lower TP score means better overall performance.

Table 3 Average exploration time and TP score

Exploration strategy	**Time required to explore scenario t_{avg} [s]**						**TP Score**
	3 robots			**5 robots**			
	S1	**S2**	**S3**	**S1**	**S2**	**S3**	[-]
MTSP^{m,w,p}	2577	1265	**537**	**1732**	872	**381**	9
MinPos^{m,w,p}	**2542**	**1201**	572	1771	873	408	10
Hun^{m,w,p}	2566	1288	566	1790	**788**	421	14
Closest^{m,w}	2715	1278	634	1965	930	465	31
TSP^{m,w}	3195	1299	581	2210	1033	463	48
CRSR^{m,p}	3158	1466	670	2379	1178	517	52
CRTSP^{p}	3569	1584	640	2398	983	520	61
CRESR^{p}	3142	1554	648	2726	1184	589	65
ETSP^{m,w}	3180	1451	668	2610	1204	468	70
Closest	5069	2283	1060	3812	1791	722	96
TSP	6085	2859	713	5118	2504	596	98
ETSP	7632	2485	868	6220	2044	697	106

TP score is computed from all the scenarios with 3 and 5 robots in all S1–S5 environments. Detailed results for all scenarios with standard deviations are in Table 8.

From the results, we can see a significant performance gap (according to TP score) between the three best-performing methods: MTSP^{m,w,p}, MinPos^{m,w,p}, and Hun^{m,w,p} and the rest of the methods. It is because the methods use shared maps and compute optimal task allocation or a given decision moment using a heuristic MTSP solution, ranking, and the optimal Hungarian algorithm, respectively. Relatively good performance can be observed for Closest^{m,w}, which outperforms even more sophisticated methods TSP^{m,w} and ETSP^{m,w}. The largest cluster of methods with TP score between 48 and 70 contains TSP^{m,w}, CRSR^{m,p}, CRTSP^{p}, CRESR^{p}, and ETSP^{m,w}. From these five methods, ETSP^{m,w} provides the highest TP score, so it performs the worst. The proposed methods CRTSP^{p} and CRESR^{p} are based on the Cross-rank with the lowest amount of data used for the coordination. Nevertheless, the three worst-performing methods do not share any information for coordination, which justifies that at least some information exchange is desirable.

7.1.2 Evaluation Results: Traveled Distance

The distance traveled by the robots is summarized in Table 4. The results reflect the time performance from Table 3, showing that the methods that explore the environments fastest also lead to the lowest traveled distances. It is mainly caused by using the same navigation stack for all exploration methods, yielding a similar average velocity of the robots.

The results suggest that the three best-performing methods require all types of sharable data: robot positions, potential exploration waypoints, and map updates. Table 4 includes the total distance traveled, which further indicates how efficiently the robots in the team are used to explore the environment. The total traveled distance

Table 4 The distance traveled in the exploration scenarios with n robots in the team

Exploration strategy	**Average maximum path length $l_{\text{avg,sum}}$ [km]**										**Total distance traveled**		
	$n=3$					$n=5$					l_{sum} [km]		
	S1	**S2**	**S3**	**S4**	**S5**	**S1**	**S2**	**S3**	**S4**	**S5**	$n=3$	$n=5$	**Total**
MTSP^{m,w,p}	2.82	**1.30**	**0.56**	0.55	0.55	**1.89**	0.96	**0.39**	**0.39**	**0.38**	85	**96**	**181**
MinPos^{m,w,p}	**2.78**	1.32	0.59	**0.49**	**0.51**	1.93	0.95	0.43	0.40	0.42	**84**	99	183
Hun^{m,w,p}	2.80	1.42	0.58	0.56	0.56	1.95	**0.86**	0.43	0.42	0.39	87	98	185
Closest^{m,w}	2.98	1.41	0.65	0.63	0.59	2.15	1.02	0.48	0.47	0.40	92	109	201
TSP^{m,w}	3.50	1.34	0.60	0.76	0.63	2.41	1.16	0.48	0.51	0.53	101	123	224
CRSR^{m,p}	3.47	1.60	0.68	0.72	0.57	2.61	1.29	0.53	0.55	0.50	104	132	236
CRTSP^{p}	3.94	1.75	0.66	0.79	0.64	2.64	1.09	0.54	0.57	0.47	115	128	243
CRESR^{p}	3.48	1.71	0.66	0.74	0.68	3.02	1.31	0.61	0.56	0.52	107	145	252
ETSP^{m,w}	3.53	1.64	0.69	0.82	0.71	2.93	1.37	0.50	0.59	0.55	109	142	252
Closest	5.54	2.50	1.09	1.00	1.07	4.18	1.95	0.74	0.71	0.83	165	205	370
TSP	6.65	2.99	0.74	0.99	0.98	5.63	2.73	0.62	0.79	0.86	183	259	442
ETSP	8.50	2.80	0.90	1.31	1.45	6.93	2.32	0.72	1.42	1.24	221	309	530

Methods are ordered according to the total distance traveled depicted in the last column, which, however, corresponds to the order according to the TP score listed in Table 3.

for scenarios with three robots by all methods is about 1455 km, and 1844 km for five robots, which is almost 3300 km in total.

7.1.3 Evaluation Results: Coverage Progress

The average coverage of the robots for each method is summarized in Table 5. The last table's column shows the difference between the average coverage using three and five robots, indicating how well the method can exploit two more robots, where a higher value suggests improved utilization of the robots. Although the highest difference is for Closest, its overall performance suggests that robot coverage is highly overlapping compared with the best-performing methods, with about fifty percentage points of coverage per robot. More robots also significantly improve the performance of the proposed CRTSP^p. It is also worth noting that the three best-performing methods based on the TP score still benefit from more robots; however, the overall coverage is similar for them. Visualization of the taken robots' decisions, areas covered, and coverage overlap is depicted in Fig. 18 and Fig. 19.

Table 5 The average percentage of the environment covered by a robot of the team with n robots

Exploration strategy	**Coverage** – cov$_n$ [%] $n=3$	$n=5$	**Diff.** cov$_3$ − cov$_5$
MTSP^{m,w,p}	50	**38**	12
MinPos^{m,w,p}	**48**	40	9
Hun^{m,w,p}	51	39	13
Closest^{m,w}	54	43	11
TSP^{m,w}	56	47	9
CRSR^{m,p}	52	42	10
CRTSP^p	56	41	15
CRESR^p	50	41	9
ETSP^{m,w}	58	49	9
Closest	82	64	**18**
TSP	83	74	10
ETSP	83	75	8

The average coverage aggregated from all the scenarios and trials. Detailed results are depicted in Table 9.

The coverage progress of the individual robots in the exploration of the S1 environment using CRESR^p is depicted in Fig. 9. The total coverage curve between 250 s and 600 s is slightly flat because Robot 2 and Robot 3 are exploring the same area already explored by Robot 1. The coverage progress of MTSP^{m,w,p} in the same environment is depicted in Fig. 10. From the plots, we can observe that the total coverage is increasing more consistently for MTSP^{m,w,p} with map update sharing. For both methods CRESR^p and MTSP^{m,w,p}, a detailed analysis of their behavior is provided in Appendix A, more specifically in Fig. 17 and Fig. 16.

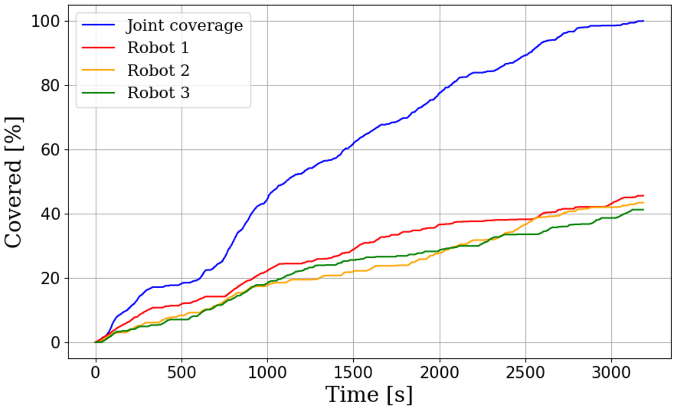


Fig. 9 Coverage progress during the exploration of S1 using the proposed CRESR^p method with three robots.

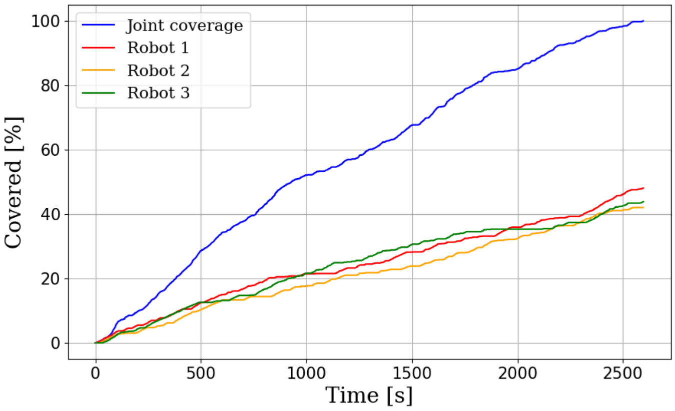
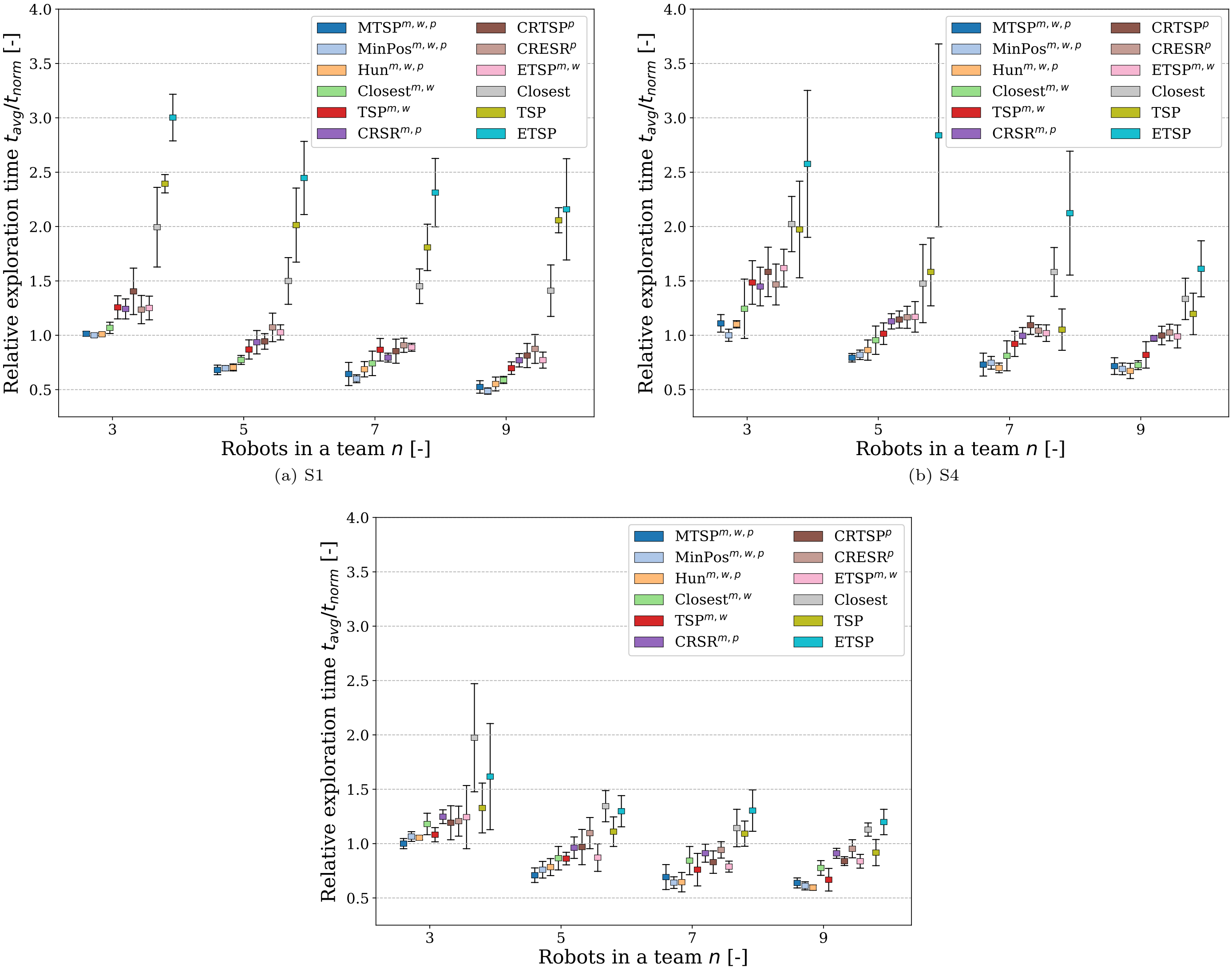


Fig. 10 Coverage progress during the exploration of S1 using the MTSP^{m,w,p} coordination method with three robots.

7.1.4 Scalability Analysis

In addition to 3 and 5 robots, we analyze the performance of the exploration using 7 and 9 robots. The environments S1, S3, and S4, representing different environmental types, have been considered in the simulations conducted similarly to the previous ones. All the robots are identical; they start from the same location with 15 s delays. Each simulation was repeated five times, resulting in $12 \cdot 5 \cdot 3 \cdot 2 = 360$ simulated exploration missions. The performance is measured by the average exploration time that is normalized using the shortest exploration time t_{norm} for three robots ($n = 3$) to show improvements using more robots. The increased number of robots increases computational demands, and therefore, the computational



(a) S1

(b) S4

(c) S3

Fig. 11 Normalized average exploration times in S1, S3, and S4 environments using a team of n robots for $n \in \{3, 5, 7, 9\}$. The exploration times are normalized by the shortest exploration time with $n = 3$, which is achieved by MinPos^{m,w,p}, MTSP^{m,w,p}, and MinPos^{m,w,p} for S1, S3, and S4, respectively. The box denotes the average value, and the error bars denote the standard deviation.

environment consists of the Intel i9-14900K with 128 GB RAM to achieve the 100 % real-time factor for up to nine robots.

The normalized average exploration times are summarized in Fig. 11. The plots suggest that the first five methods benefit from the increased team size and can exploit more robots; however, differences between 7 and 9 robots are marginal. The proposed methods and ETSP^{m,w} generally benefit from increasing the team size.

Noticeable differences can be observed when considering particular environments. In the relatively large S1, one of the three best-performing methods MTSP^{m,w,p}, MinPos^{m,w,p}, Hun^{m,w,p} is always a better choice than Closest^{m,w}, see Fig. 11a. On the other hand, in S4, which is less than half the size of S1, the three best-performing methods and Closest^{m,w} perform similarly when the number of robots is sufficiently increased, see Fig. 11b. For S3, the plot in Fig. 11c shows that Closest^{m,w} is outperformed by TSP^{m,w}. It is primarily due to relatively large open areas in S3, where exploring waypoints ordered based on solving TSP^{m,w} yields a more efficient exploration strategy than using the greedy method Closest^{m,w}. A similar phenomenon can also be observed in the proposed methods. Specifically, in S3, CRTSP^p with the employed TSP performs better than CRSR^{m,p} and CRESR^p.

The three worst-performing methods, Closest, TSP, and ETSP, exhibit a trend of decreasing average exploration time with increasing team size; however, they are, in general, the worst choice for any studied n.

7.1.5 Localization Error Impact Study

The proposed coordination methods have been further studied to determine the impact of the localization drift that can cause inconsistencies in the shared data between the robots. The localization drift is simulated as an (x, y, z) position error, calculated as a percentage of the distance traveled from the initial position of the robot. The position error is added to a direction that is randomly generated at the beginning of the mission. Hence, the localization error is increasing over the mission duration, and each position of the robot shared with the other robots is affected by the position error, simulating localization drift that affects other robots.

We include the Closest and TSP methods in the study because they do not share any data; hence, there is no observable impact. Since the methods continue to explore the environment, if reachable exploration waypoints are determined, they avoid incomplete exploration when maps are incorrectly merged because these methods do not use any shared maps of the other robots. However, the proposed CRTSP^p and CRESR^p methods rely on the shared positions of the robots, and therefore, an impact on the exploration time is expected. The methods that use shared map data have to deal with map inconsistencies, which is a challenging problem itself, and it would indeed yield degraded performance. Therefore, they are not included in the evaluation.

The localization error impact is studied for the localization drifts 0.5 %, 1.0 %, 1.5 %, 2.0 %, and 2.5 %; each for three exploration trials in the S4 environment with 5 robots. The normalized average exploration times are depicted in Fig. 12. Closest and TSP are independent of localization drift since the methods do not exchange data. For CRTSP^p and CRESR^p, increasing exploration time can be observed for the drifts above 0.5 %, which cause accumulated error above the parameter m_r. Up to the drift 1.5 %, the proposed methods perform better than the reference ones. Note that, for such a drift, the absolute localization error is about 10 m at the end of the exploration missions, which is significantly more than m_r, as robots traveled around 560 m on average. Thus, 1.5 % drift prolongs the exploration mission to about 672 m. A similar issue with localization error is also expected for the methods using the shared maps, which would yield inconsistencies in the merged maps. Hence, methods that share less data should be preferred for missions with a high localization error. However, when the error is below the Cross-rank parameter m_r used for accounting waypoints near the crossroads, the proposed CRTSP^p and CRESR^p methods outperform localization drift-independent Closest and TSP. Further study on the robustness of coordination methods to localization errors is out of the paper's scope.

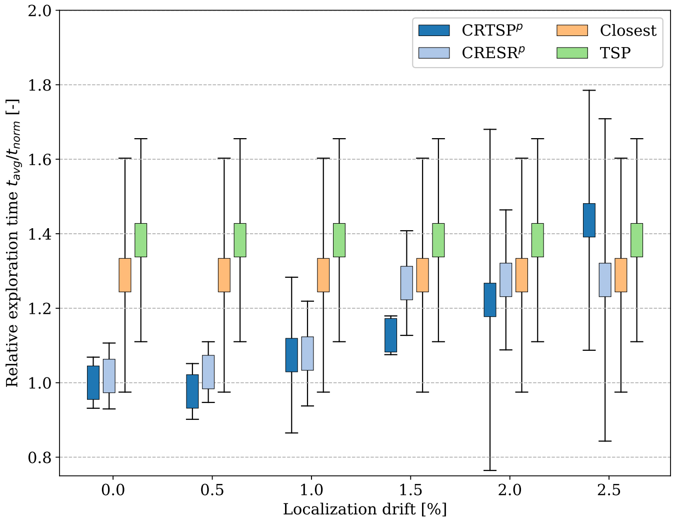


Fig. 12 Normalized average exploration times of the proposed methods under the simulated localization drift in the environment S4 using $n = 5$ robots. The normalization is by the shortest exploration time without the localization drift that is achieved by the proposed CRTSP^p. The boxes represent the average values, and the error bars denote the standard deviation among the performed trials.

7.1.6 Heterogeneous Robots

The final evaluation in the simulated exploration missions is on the influence of the heterogeneous robots in the team using robots with different maximum speeds. The evaluation scenarios are adjusted to have three robots with the original maximum speed set to 1.2 m s^{-1} and two additional slower robots with 0.6 m s^{-1}. Hence, we can expect a shorter exploration time with five heterogeneous robots than with three robots, yet still slower than five fast robots. The average exploration times for the S3 and S4 environments are depicted in Table 6, where the time for the heterogeneous team is denoted as “Heter.” and results for $n = 3$ and $n = 5$ are from Table 8. The reported average times are computed from five trials.

The results indicate the expected slight decrease in the exploration time for most of the methods. However,

Table 6 Average exploration time with heterogeneous robots

Exploration strategy	Time required to explore scenario t_{avg}					
	S3			**S4**		
	$n = 3$	Heter.	$n = 5$	$n = 3$	Heter.	$n = 5$
MTSP^{m,w,p}	537	513	381	537	461	381
MinPos^{m,w,p}	572	485	408	484	442	408
Hun^{m,w,p}	566	466	421	533	446	421
Closest^{m,w}	634	541	465	602	454	465
TSP^{m,w}	581	513	463	719	479	463
CRSR^{m,p}	670	618	517	701	627	517
CRTSP^{p}	640	628	520	766	646	520
CRESR^{p}	648	671	589	710	653	589
ETSP^{m,w}	668	562	468	783	620	468
Closest	1060	758	722	979	906	722
TSP	713	679	596	955	858	596
ETSP	868	972	697	1247	1169	697

CRESR^p performs worse with heterogeneous robots in S3 than with three fast robots. It is primarily caused by the fact that when two robots, one fast and one slow, enter two corridors from a crossroad, the Cross-rank is calculated as if the robots have the same speed. As a result, the slow robots unnecessarily discourage the fast robots from exploring parts of the environment that would require more robots. The limitation highlights opportunities for future improvements to the proposed Cross-rank, such as weighting by the average speed of the robots.

7.2 Real-world Deployment

The proposed CRESR^p Cross-rank coordination method has been experimentally verified in an outdoor environment with three quadruped robots. The method has low communication requirements and can operate even under a relatively narrow bandwidth limited to broadcasting $100\,\mathrm{B\,s^{-1}}$. The experimental environment is visualized in Fig. 13, showing a complete 3D point cloud map created from the locally collected maps by the robots after the mission ended.

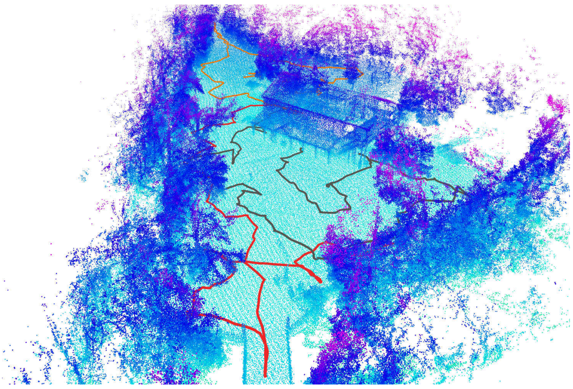


Fig. 13 3D point cloud map of the explored environment created from the local maps of the robots after the mission ended. The area is approximately 100 m × 85 m.

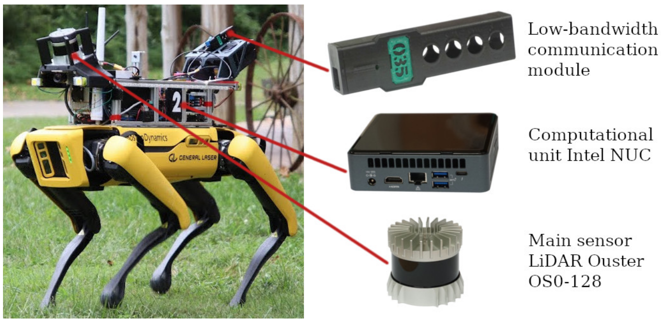

Low-bandwidth communication module
Computational unit Intel NUC
Main sensor LiDAR Ouster OS0-128

Fig. 14 One of the deployed quadruped robots used in the real-world experiment.

All the robots are identical quadruped platforms, Spot by Boston Dynamics, equipped with a custom payload as depicted in Fig. 14. The main part of the payload is the computational unit with the Intel NUC10i7FNK with the Intel i7-10710U processor and 32 GB RAM dedicated to robot localization and running autonomous navigation and multi-robot coordination. The sensor used is the Ouster OS0-128 LiDAR with 128 lines and 1024 measurements per line running at 10 Hz. Communication between the robots is provided by the communication modules working at frequency 868/915 MHz, which represents a low-bandwidth communication channel (Bayer et al, 2023).

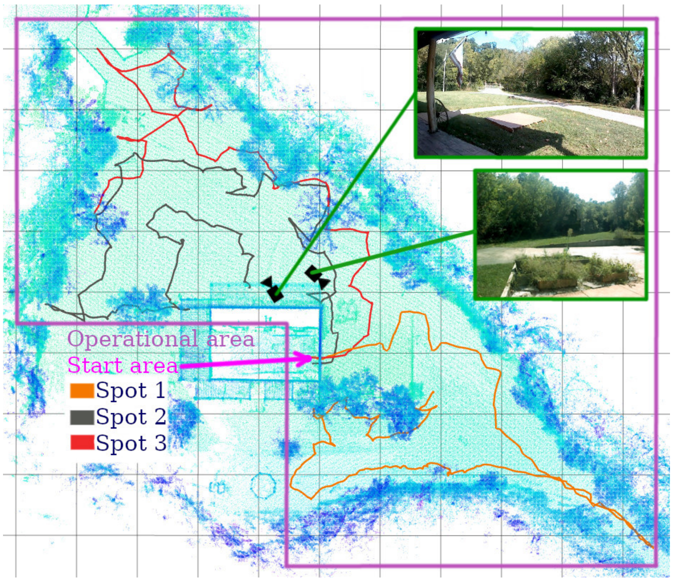


Fig. 15 Map of the environment explored by three quadruped robots. The size of the shown squared grid cell corresponds to 10 m.

The testing environment with the indicated initial location of the robots is shown in Fig. 15. The robots start from the house garage marked as the **start area**. The nearby forest creates most of the outline of the explorable environment. The rest of the outline is defined by the border of the operational area marked in Fig. 15 to keep the robots from the nearby road and small parking lot near the house.

Table 7 Traveled distances and pack loss ratio in the real-world exploration

Robot	**Distance Traveled** [m]	**Packet Loss Ratio**[†] [%] Spot 1	 Spot 2	 Spot 3
Spot 1	223.4	-	82	69
Spot 2	225.3	83	-	91
Spot 3	175.7	47	78	-

[†] The loss ratio is determined from the packets sent from the particular robot. The exchanged data fit a single packet.

The whole experiment took 575 s, during which each robot broadcasted its positions at the frequency of 1 Hz. The distance traveled by the robots is listed in Table 7 together with the relatively low reliability of the delivered messages, which highlights the benefits of decentralized coordination methods.

During the mission, Spot 1 started the exploration, and after it avoided a small wall, it continued to explore the right part of the environment from the start. Spot 2 was sent approx. 30 s after Spot 1 and decided to explore the opposite part of the environment than the first robot. Finally, approx. 45 s after Spot 2, Spot 3 started to explore the remaining parts of the environment. The paths traversed by the robots are visualized in Fig. 15. The performed real-world experiment supports the feasibility of the proposed method in coordinating real robots in an outdoor scenario, where the robots suffer from imprecise measurements and localization, imperfect execution of the actions, and unreliable communication capable of receiving only about 25 % of the transmitting position data for coordination.

8 Discussion

Based on the reported results, the proposed coordination methods using the Cross-rank successfully led the robots to the least visited areas at the crossroads. The Cross-rank performs best in combination with the Spread-rank that uses the path lengths in CRSR^{m,p}, albeit it does not fulfill the bandwidth requirements. The performance slightly decreases with approximate path lengths using Euclidean distances of CRESR^{p}. However, the method fulfills the bandwidth requirements and provides slightly better results for exploration scenarios with three robots than CRTSP^{p}. The proposed CRTSP^{p} performs better than CRESR^{p} in scenarios with five robots. Both proposed methods CRESR^{p} and CRTSP^{p} show to be a suitable choice when low-bandwidth communication is the only suitable option. In real-world experimental deployment, the real system suffers from limited communication with the average packet loss of about 75 %; see Table 7. Despite the imperfect communication, the proposed Cross-rank method successfully handles the robot distribution in the operational area as depicted in Fig. 15.

Moreover, when analyzing the impact of the localization drift, the results in Fig. 12 indicate that the proposed CRTSP^{p} and CRESR^{p} perform better in the environment S4 than methods that do not share any information. The improved performance is noticeable even for the localization drift 1.5 %. For the drift above that value, the shared data becomes too noisy, damaging the coordination.

The overall best-performing methods are MTSP^{m,w,p}, MinPos^{m,w,p}, and Hun^{m,w,p}, which share maps, waypoints, and positions between the robots. Sharing maps is too communication-demanding w.r.t. the motivational scenario involving low-bandwidth communication only. The real bandwidth requirements of these methods can further increase in more open areas. Based on the presented evaluation results, increased performance is at the cost of increased bandwidth requirements. The exception is the ETSP^{m,w} method with the required map updates, which is outperformed by the proposed CRESR^{p} and CRTSP^{p}, both with lower bandwidth requirements.

When considering computational requirements, the best performance-to-computational-demand ratio has Closest^{m,w} with performance just behind the best three performing methods and has the second-lowest computational requirements. Note that, when considering TSP-based methods, decreasing computational requirements by replacing the computation of real distances between the waypoints with Euclidean distances results in a significant decrease in performance for the environments S1, S2, S4, and S5, where Euclidean distance poorly approximates path lengths between the locations. On the other hand, when the space being explored contains primarily sparse obstacles or open areas, like the environment S3, such an approximation is less damaging to exploration performance, especially for robot teams larger than three.

Exploration using TSP-based Strategies

The presented results suggest high-performance variations in TSP-based strategies, which, in general, assume that path lengths between the waypoints are known and that the robot is going to visit the waypoints in the order of the TSP solution. The performance decreases when violating the assumptions. In decentralized exploration, the assumption of visiting the waypoints in the determined order may not hold because robots uncover new parts of the environment, leading to new waypoints, or another robot might cover the waypoint of the first robot's route. The violation of the assumptions can be decreased by clustering the waypoints, such as in MTSP^{m,w,p}, to keep the robots from visiting other robots' waypoints. Another way is to limit the planning horizon by limiting the length of the routes, e.g., by splitting waypoints between more robots or planning the TSP route only for a subset of possible exploration waypoints.

The effect of decreased performance can be observed for TSP^{m,w} that is outperformed by Closest^{m,w}. If we further utilize Euclidean distances between the waypoints instead of path lengths, performance further decreases, which can be observed when comparing TSP^{m,w} and ETSP^{m,w}. Similarly, without any information exchange, the Closest method performs better than TSP and ETSP. Therefore, data sharing helps to improve the exploration performance; however, when communication is limited, the proposed Cross-rank method is a more suitable choice than methods that do not share anything, including the Closest method.

Limitations of the proposed method

Since the coordination is based on the shared positions between the robots, the Cross-rank's bandwidth requirements depend on the speed at which the robots move. In simulations, the robots move at speeds up to $1.2\,\mathrm{m\,s^{-1}}$ and share positions at $f_{\text{positions}} = 5\,\mathrm{Hz}$, which also accounts for unreliable communication; thus, it can be lowered, possibly to 1 Hz, as during the real-world experiment. However, if the robots move more than five times faster, or the frequency of sharing positions is less than 1 Hz, the performance of the Cross-rank would start degrading, because the distance between the shared positions would start becoming too large to cover sharp turns. The limitation can be overcome by increasing the frequency of sharing positions between the robots proportionally to the increased speed of the robots; thus, more bandwidth would be needed.

The proposed method is also limited w.r.t. the used localization system. Based on assumptions in Section 3, the localization system must provide positions of all robots w.r.t. a common coordinate frame. Moreover, based on the reported empirical results, when the localization drift exceeds 1.5 %, the performance of the proposed method decreases below the level of independently exploring robots.

When the assumptions outlined in Section 3 are not fulfilled, the coordination performance may also decrease. For example, when robots are capable of moving at different speeds (the team of robots is not homogeneous), the performance of method CRESR^{p} decreases, as shown in Section 7.1.6.

9 Conclusion

The presented evaluation results of multi-robot exploration strategies in relatively large-scale environments show that the performance of the methods is strongly related to the communication bandwidth used. The three best-performing methods MTSP^{m,w,p}, MinPos^{m,w,p}, and Hun^{m,w,p} share all considered data types: map updates, exploration waypoints, and robots' positions. Besides, addressing multi-robot exploration by the solution of the TSP or Euclidean TSP instances independently for each robot demonstrates that it is less effective than greedy exploration, in both cases, with and without the map updates shared between the robots. The proposed Cross-rank method performed worse than the best-performing methods that require the sharing of waypoints and map updates. However, it performs best when the bandwidth is limited to only sharing the robots' positions. Finally, the real-world experimental deployment further supports the practical feasibility of the proposed method in outdoor exploration with three robots connected using only a low-bandwidth and relatively unreliable communication system. The performance of the method could be improved by sharing maps between the robots if the communication bandwidth allows that. Nevertheless, the benefit of the proposed method is in communication-restricted setups or when a high number of robots limits achievable communication bandwidth.

In future work, we would like to extend the calculation of the Cross-rank to account for different robot speeds to improve performance when coordinating heterogeneous teams of robots. Moreover, we would like to increase the resistance of the proposed coordination method to the localization drift by matching the shapes of the robot's trajectories.

References

Alitappeh RJ, Jeddisaravi K (2022) Multi-robot exploration in task allocation problem. Applied Intelligence

52(2):2189–2211. https://doi.org/10.1007/s10489-021-02483-3

Ann S, Kim Y, Ahn J (2015) Area allocation algorithm for multiple uavs area coverage based on clustering and graph method. IFAC-PapersOnLine 48(9):204–209. https://doi.org/10.1016/j.ifacol.2015.08.084

Azpúrua H, Saboia M, Freitas GM, Clark L, Agha-mohammadi Aa, Pessin G, Campos MF, Macharet DG (2023) A survey on the autonomous exploration of confined subterranean spaces: Perspectives from real-word and industrial robotic deployments. Robotics and Autonomous Systems 160:104304. https://doi.org/10.1016/j.robot.2022.104304

Batinović A, Oršulić J, Petrović T, Bogdan S (2020) Decentralized strategy for cooperative multi-robot exploration and mapping. IFAC-PapersOnLine 53(2):9682–9687. https://doi.org/10.1016/j.ifacol.2020.12.2618

Bautin A, Simonin O, Charpillet F (2012) Minpos : A novel frontier allocation algorithm for multi-robot exploration. In: International Conference on Intelligent Robotics and Applications (ICIRA), pp 496–508, https://doi.org/10.1007/978-3-642-33515-0_49

Bayer J, Faigl J (2021) Decentralized topological mapping for multi-robot autonomous exploration under low-bandwidth communication. In: 2021 European Conference on Mobile Robots (ECMR), IEEE, pp 1–7, https://doi.org/10.1109/ECMR50962.2021.9568824

Bayer J, Čížek P, Faigl J (2023) Autonomous multi-robot exploration with ground vehicles in darpa subterranean challenge finals. Field Robotics 3:266–300. https://doi.org/10.55417/fr.2023008

Bouman A, Ginting MF, Alatur N, Palieri M, Fan DD, Touma T, Pailevanian T, Kim SK, Otsu K, Burdick J, a. Agha-Mohammadi A (2020) Autonomous spot: Long-range autonomous exploration of extreme environments with legged locomotion. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 2518–2525, https://doi.org/10.1109/IROS45743.2020.9341361

Burgard W, Moors M, Fox D, Simmons R, Thrun S (2000) Collaborative multi-robot exploration. In: IEEE International Conference on Robotics and Automation (ICRA), pp 476–481, https://doi.org/10.1109/ROBOT.2000.844100

Chung TH, Orekhov V, Maio A (2023) Into the robotic depths: Analysis and insights from the darpa subterranean challenge. Annual Review of Control, Robotics, and Autonomous Systems 6:477–502. https://doi.org/10.1146/annurev-control-062722-100728

Dang T, Tranzatto M, Khattak S, Mascarich F, Alexis K, Hutter M (2020) Graph-based subterranean exploration path planning using aerial and legged robots. Journal of Field Robotics 37(8):1363–1388. https://doi.org/10.1002/rob.21993

Deschênes SP, Baril D, Kubelka V, Giguère P, Pomerleau F (2021) Lidar scan registration robust to extreme motions. In: Conference on Robots and Vision (CRV), pp 17–24, https://doi.org/10.1109/CRV52889.2021.00014

Ebadi K, Chang Y, Palieri M, Stephens A, Hatteland A, Heiden E, Thakur A, Funabiki N, Morrell B, Wood S, Carlone L, Agha-mohammadi Aa (2020) Lamp: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments. In: IEEE International Conference on Robotics and Automation (ICRA), pp 80–86, https://doi.org/10.1109/ICRA40945.2020.9197082

Faigl J, Kulich M (2013) On determination of goal candidates in frontier-based multi-robot exploration. In: European Conference on Mobile Robots (ECMR), pp 210–215, https://doi.org/10.1109/ECMR.2013.6698844

Faigl J, Kulich M (2015) On benchmarking of frontier-based multi-robot exploration strategies. In: European Conference on Mobile Robots (ECMR), pp 1–8, https://doi.org/10.1109/ECMR.2015.7324183

Heppner G, Roennau A, Dillman R (2013) Enhancing sensor capabilities of walking robots through cooperative exploration with aerial robots. Journal of Automation Mobile Robotics and Intelligent Systems 7

Hollinger G, Smith A (2018) Distributed inference-based multi-robot exploration. Autonomous Robots 42:1651–1668. https://doi.org/10.1007/s10514-018-9708-7

Horyna J, Baca T, Walter V, Albani D, Hert D, Ferrante E, Saska M (2023) Decentralized swarms of unmanned aerial vehicles for search and rescue operations without explicit communication. Autonomous Robots 47(1):77–93. https://doi.org/10.1007/s10514-022-10066-5

Huang P, Lin L, Xu K, Huang H (2022) Autonomous outdoor scanning via online topological and geometric path optimization. IEEE Transactions on Intelligent Transportation Systems 23(4):3682–3695. https://doi.org/10.1109/TITS.2020.3039557

Hussein A, Adel M, Bakr M, Shehata OM, Khamis A (2014) Multi-robot task allocation for search and rescue missions. In: Journal of Physics: Conference Series, IOP Publishing, p 052006, https://doi.org/10.1088/1742-6596/570/5/052006

Kim S, Corah M, Keller J, Best G, Scherer S (2023) Multi-robot multi-room exploration with geometric cue extraction and circular decomposition. IEEE Robotics and Automation Letters https://doi.org/10.1109/LRA.2023.3342553

Kulich M, Faigl J, Přeučil L (2011) On distance utility in the exploration task. In: IEEE International Conference on Robotics and Automation (ICRA), pp 4455–4460, https://doi.org/10.1109/ICRA.2011.5980221

Kulkarni M, Dharmadhikari M, Tranzatto M, Zimmermann S, Reijgwart V, De Petris P, Nguyen H, Khedekar N, Papachristos C, Ott L, et al (2022) Autonomous teamed exploration of subterranean environments using legged and aerial robots. In: IEEE International Conference on Robotics and Automation (ICRA), pp 3306–3313, https://doi.org/10.1109/ICRA46639.2022.9812401

Luca AD, Oriolo G, Samson C (2005) Feedback control of a nonholonomic car-like robot. In: Laumond JP (ed) Robot Motion Planning and Control. Springer, p 171–253, https://doi.org/10.1007/BFb0036073

Miller ID, Cladera F, Cowley A, Shivakumar SS, Lee ES, Jarin-Lipschitz L, Bhat A, Rodrigues N, Zhou A, Cohen A, et al (2020) Mine tunnel exploration using multiple quadrupedal robots. IEEE Robotics and Automation Letters 5(2):2840–2847. https://doi.org/10.1109/LRA.2020.2972872

Muja M, Lowe DG (2014) Scalable nearest neighbor algorithms for high dimensional data. IEEE Transactions on Pattern Analysis and Machine Intelligence 36(11):2227–2240. https://doi.org/10.1109/TPAMI.2014.2321376

Ouster OS0 (accessed Jan 11, 2024) OS0: Ultra-Wide View High-Resolution Imaging Lidar. https://data.ouster.io/downloads/datasheets/datasheet-rev7-v3p0-os0.pdf

Phung MD, Quach CH, Dinh TH, Ha Q (2017) Enhanced discrete particle swarm optimization path planning for uav vision-based surface inspection. Automation in Construction 81:25–33. https://doi.org/10.1016/j.autcon.2017.04.013

Pomerleau F, Colas F, Siegwart R, Magnenat S (2013) Comparing ICP Variants on Real-World Data Sets. Autonomous Robots 34(3):133–148. https://doi.org/10.1007/s10514-013-9327-2

Prágr M, Bayer J, Faigl J (2022) Autonomous robotic exploration with simultaneous environment and traversability models learning. Frontiers in Robotics and AI 9:910113. https://doi.org/10.3389/frobt.2022.910113

Puig D, Garcia M, Wu L (2011) A new global optimization strategy for coordinated multi-robot exploration: Development and comparative evaluation. Robotics and Autonomous Systems 59(9):635–653. https://doi.org/10.1016/j.robot.2011.05.004

Qin H, Meng Z, Meng W, Chen X, Sun H, Lin F, Ang MH (2019) Autonomous exploration and mapping system using heterogeneous uavs and ugvs in gps-denied environments. IEEE Transactions on Vehicular Technology 68(2):1339–1350. https://doi.org/10.1109/TVT.2018.2890416

Queralta JP, Taipalmaa J, Pullinen BC, Sarker VK, Gia TN, Tenhunen H, Gabbouj M, Raitoharju J, Westerlund T (2020) Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision. IEEE Access 8:191617–191643. https://doi.org/10.1109/ACCESS.2020.3030190

Schulz C, Hanten R, Reisenauer M, Zell A (2019) Simultaneous collaborative mapping based on low-bandwidth communication. In: IEEE International Conference on Robotic Computing (IRC), pp 413–414, https://doi.org/10.1109/IRC.2019.00076

Shrestha R, Tian FP, Feng W, Tan P, Vaughan R (2019) Learned map prediction for enhanced mobile robot exploration. In: IEEE International Conference on Robotics and Automation (ICRA), pp 1197–1204, https://doi.org/10.1109/ICRA.2019.8793769

Smith AJ, Hollinger GA (2018) Distributed inference-based multi-robot exploration. Autonomous Robots 42(8):1651–1668. https://doi.org/10.1007/s10514-018-9708-7

Song S, Jo S (2018) Surface-based exploration for autonomous 3d modeling. In: IEEE International Conference on Robotics and Automation (ICRA), pp 4319–4326, https://doi.org/10.1109/ICRA.2018.8460862

Tan AH, Bejarano FP, Zhu Y, Ren R, Nejat G (2022) Deep reinforcement learning for decentralized multi-robot exploration with macro actions. IEEE Robotics and Automation Letters 8(1):272–279. https://doi.org/10.1109/LRA.2022.3224667

Wang C, Zhu D, Li T, Meng MQH, De Silva CW (2019) Efficient autonomous robotic exploration with semantic road map in indoor environments. IEEE Robotics and Automation Letters 4(3):2989–2996. https://doi.org/10.1109/LRA.2019.2923368

Westheider J, Rückin J, Popović M (2023) Multi-uav adaptive path planning using deep reinforcement learning. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 649–656, https://doi.org/10.1109/IROS55552.2023.10342516

Williams J, Jiang S, O'Brien M, Wagner G, Hernandez E, Cox M, Pitt A, Arkin R, Hudson N (2020) Online 3d frontier-based ugv and uav exploration using direct point cloud visibility. In: IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), pp 263–270, https://doi.org/10.1109/MFI49285.2020.9235268

Wu L, Luo H (2022) A multi-robot exploration approach for larger maps with unreliable communications. In: IEEE International Conference on Unmanned Systems (ICUS), pp 1083–1088, https://doi.org/10.1109/ICUS55513.2022.9986603

Wurm KM, Stachniss C, Burgard W (2008) Coordinated multi-robot exploration using a segmentation of the environment. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 1160–1165, https://doi.org/10.1109/IROS.2008.4650734

Xie J, Carrillo LRG, Jin L (2020) Path planning for uav to cover multiple separated convex polygonal regions. IEEE Access 8:51770–51785. https://doi.org/10.1109/ACCESS.2020.2980203

Xu Z, Fitch R, Sukkarieh S (2013) Decentralised coordination of mobile robots for target tracking with learnt utility models. In: IEEE International Conference on Robotics and Automation (ICRA), pp 2014–2020, https://doi.org/10.1109/ICRA.2013.6630846

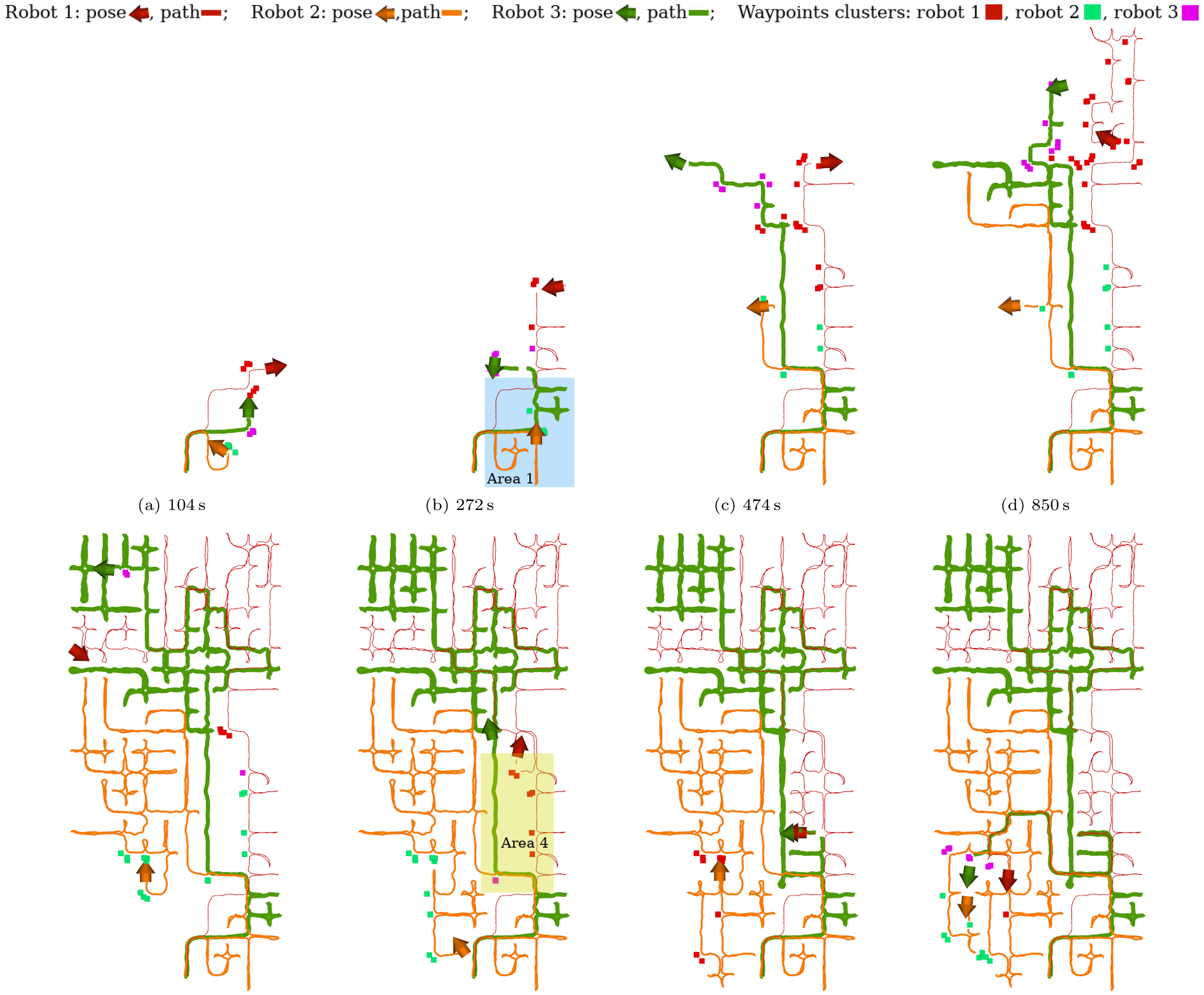
Yamauchi B (1997) A frontier-based approach for autonomous exploration. In: IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), pp 146–151, https://doi.org/10.1109/CIRA.1997.613851

Yamauchi B (1998) Frontier-based exploration using multiple robots. In: Second International Conference on Autonomous Agents. Association for Computing Machinery, p 47–53, https://doi.org/10.1145/280765.280773

Zoula M, Prágr M, Faigl J (2021) On building communication maps in subterranean environments. In: 2020 Modelling and Simulation for Autonomous Systems (MESAS), pp 15–28, https://doi.org/bib:10.1007/978-3-030-70740-8_2

A Appendix

Detailed results on the coordination methods CRESR^p and MTSP^{m,w,p} are depicted in Fig. 17 and Fig. 16, respectively. Besides, full results of the exploration time and coverage are presented in Table 8 and Table 9, respectively, for completeness. Moreover, robots' paths are visualized in Fig. 18 and Fig. 19 for exploration of the environment S1 with three and five robots, respectively.


Robot 1: pose ⬅, path ▬; Robot 2: pose ⬅,path ▬; Robot 3: pose ⬅, path ▬; Waypoints clusters: robot 1 🟥, robot 2 🟩, robot 3 🟪
Area 1
Area 4

(a) 104 s

(b) 272 s

(c) 474 s

(d) 850 s

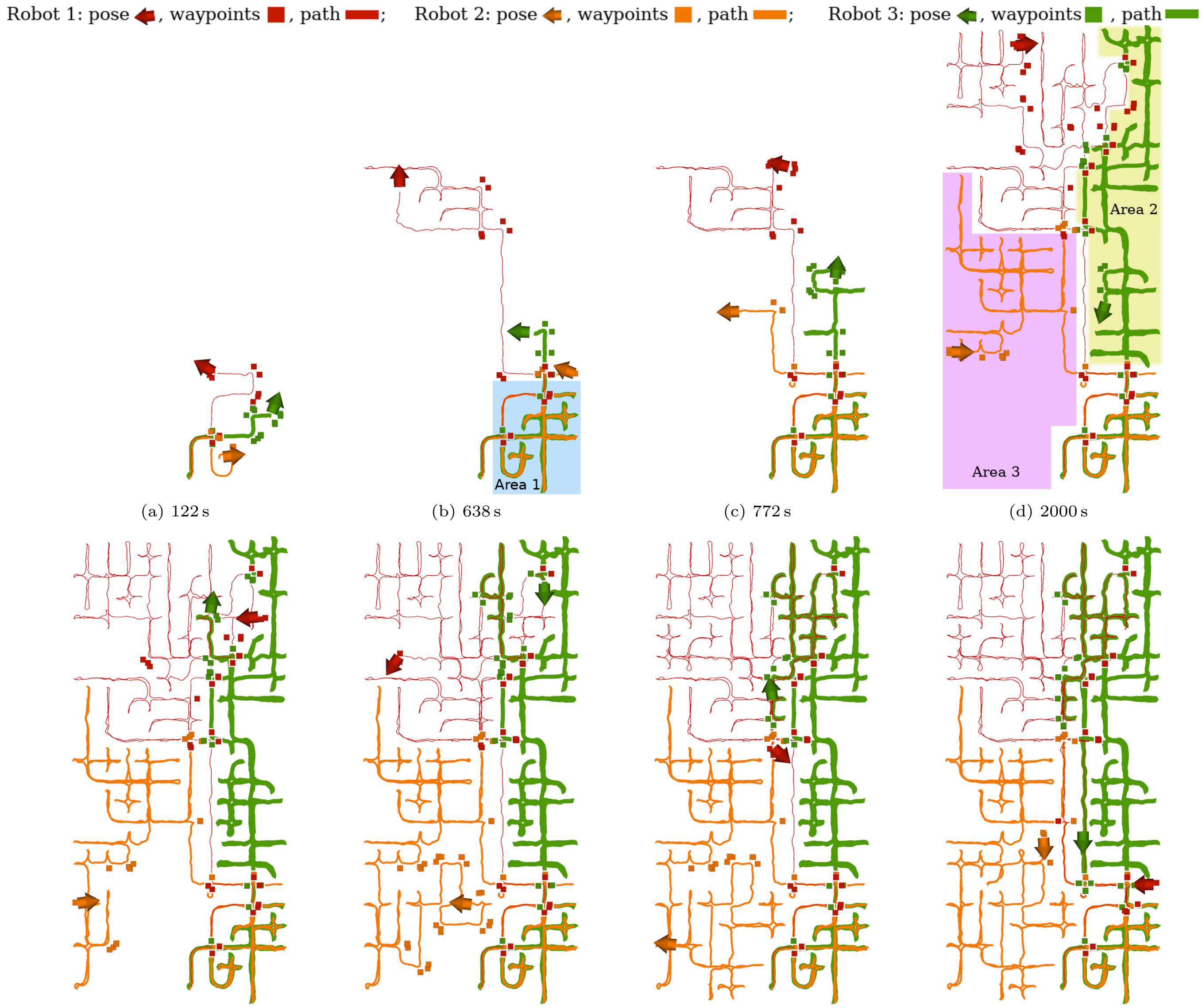
(e) 1824 s

(f) 2037 s

(g) 2240 s

(h) 2444 s

Fig. 16 Evolution of the exploration using MTSP^{m,w,p} in the environment S1 with three robots in the first evaluation trial. The figures show the poses and paths of each robot and shared waypoints assigned by clustering from the perspective of Robot 1 (red robot) to all three robots. At the first crossroad, MTSP^{m,w,p} spreads the robots to different corridors; see Fig. 16a. From Fig. 16b, we can see that the red and green robots left Area 1 before it was fully explored, and the rest of the exploration of the area was left to the orange robot, which saved a significant amount of time in comparison to, e.g., CRESR^p method. Then, the clustering splits the robots into mostly disjoint areas; see Fig. 16c and Fig. 16d. At 1824 s from the mission start, the red and green robots are finishing the upper part of the environment, and they are returning to explore the rest of Area 4, see Fig. 16f. When the red and green robots finished exploring Area 4, see Fig. 16g, they moved together to the last unexplored area, where the orange robot was already exploring, which can be observed from Fig. 16c. In Fig. 16h, we can further see how all the robots are exploring the remaining parts of the last area. Here, we can see an advantage of sharing waypoints and maps between the robots since robots can then move very efficiently to the remaining parts of the environment. The coverage by each robot and the joined coverage increase during the trial are shown in Fig. 10.


Robot 1: pose 🟥, waypoints 🟥, path ▬;
Robot 2: pose 🟧, waypoints 🟧, path ▬;
Robot 3: pose 🟩, waypoints 🟩, path ▬
Area 1
Area 2
Area 3

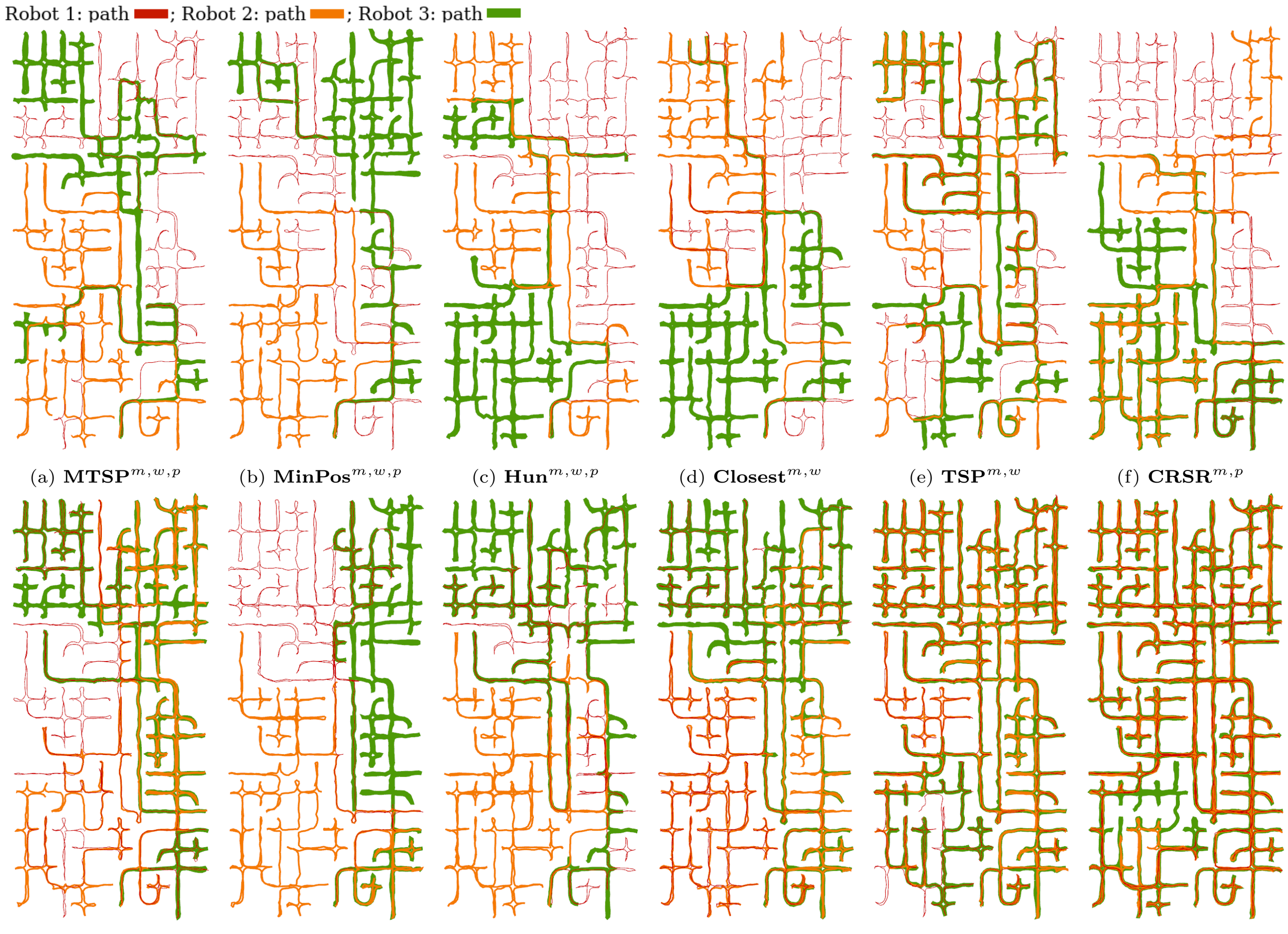
(a) 122 s (b) 638 s (c) 772 s (d) 2000 s

(e) 2278 s (f) 2584 s (g) 2969 s (h) 3155 s

Fig. 17 Evolution of the exploration using CRESR^p in the environment S1 with three robots in the first evaluation trial. The figures show the poses, paths, and exploration waypoints detected by each robot. At the first crossroad, depicted in Fig. 17a, the robots select different corridors. Next, the red robot finds its way out of Area 1, marked in Fig. 17b, and starts exploring the rest of the environment. Because the red robot explored the only way out of Area 1 before the Cross-rank of the waypoints corresponding to the way out was the lowest, the orange and green robots explored most of Area 1 before they left it. The observed behavior of the Cross-rank is induced by the fact that the robots do not share exploration waypoints. Next, when the green and orange robots leave Area 1, they start exploring Area 2 and Area 3, which are bounded by the corridors that are already explored by the red robot, see Fig. 17c, and Fig. 17d. The green robot finishes exploring the area bounded by the corridors explored by the red robot approximately 2278 s after the mission starts; see Fig. 17e. Then, the green robot starts exploring parts of the environment for waypoints with a high value of the Cross-rank, which have already been explored by the red robot. About 2585 s after the mission started, the red and orange robots are still exploring unique areas of the environment; see Fig. 17f. Next, the red robot finishes exploring waypoints with a value of the Cross-rank equal to 0 and starts exploring waypoints with high ranks, which leads to exploring already visited corridors, which is shown in Fig. 17g. The whole environment is explored when the orange robot explores its last two waypoints with the zero value of the Cross-rank, which are the two closest waypoints to the robot's location, shown in Fig. 17h. The coverage by each robot and the joined coverage increasing during the trial are shown in Fig. 9.

Table 8 Average time required to explore environments S1–S5 with three and five robots. The methods are sorted based on the TP score calculated from all the scenarios.

Exploration strategy	**Average time required to explore the scenario t_{avg} [s]**										**TP score**
	3 robots					**5 robots**					
	S1	**S2**	**S3**	**S4**	**S5**	**S1**	**S2**	**S3**	**S4**	**S5**	[-]
MTSP^{m,w,p}	2577±55	1265±75	**537±25**	537±39	541±68	**1732±112**	872±121	**381±36**	**384±19**	**372±13**	9
MinPos^{m,w,p}	**2542±47**	**1201±32**	572±24	**484±27**	**493±36**	1771±60	873±22	408±41	397±21	410±14	10
Hun^{m,w,p}	2566±30	1288±84	566±11	533±16	534±46	1790±81	**788±73**	421±42	418±45	377±23	14
Closest^{m,w}	2715±132	1278±73	634±53	602±132	552±73	1965±105	930±89	465±58	462±63	388±33	31
TSP^{m,w}	3195±270	1299±105	581±35	719±97	591±83	2210±224	1033±88	463±31	491±48	508±49	48
CRSR^{m,p}	3158±234	1466±189	670±34	701±86	541±55	2379±273	1178±108	517±53	546±34	476±35	52
CRTSP^{p}	3569±543	1584±207	640±84	766±110	604±68	2398±182	983±100	520±87	554±38	443±51	61
CRESR^{p}	3142±330	1554±136	648±74	710±91	640±55	2726±333	1184±93	589±77	564±49	490±88	65
ETSP^{m,w}	3180±277	1451±143	668±156	783±84	679±141	2610±176	1204±283	468±68	566±68	515±71	70
Closest	5069±931	2283±379	1060±267	979±123	1026±129	3812±546	1791±112	722±77	714±174	803±116	96
TSP	6085±214	2859±114	713±123	955±215	921±152	5118±866	2504±226	596±73	766±151	805±129	98
ETSP	7632±545	2485±415	868±262	1247±327	1367±138	6220±855	2044±219	697±77	1374±407	1174±82	106



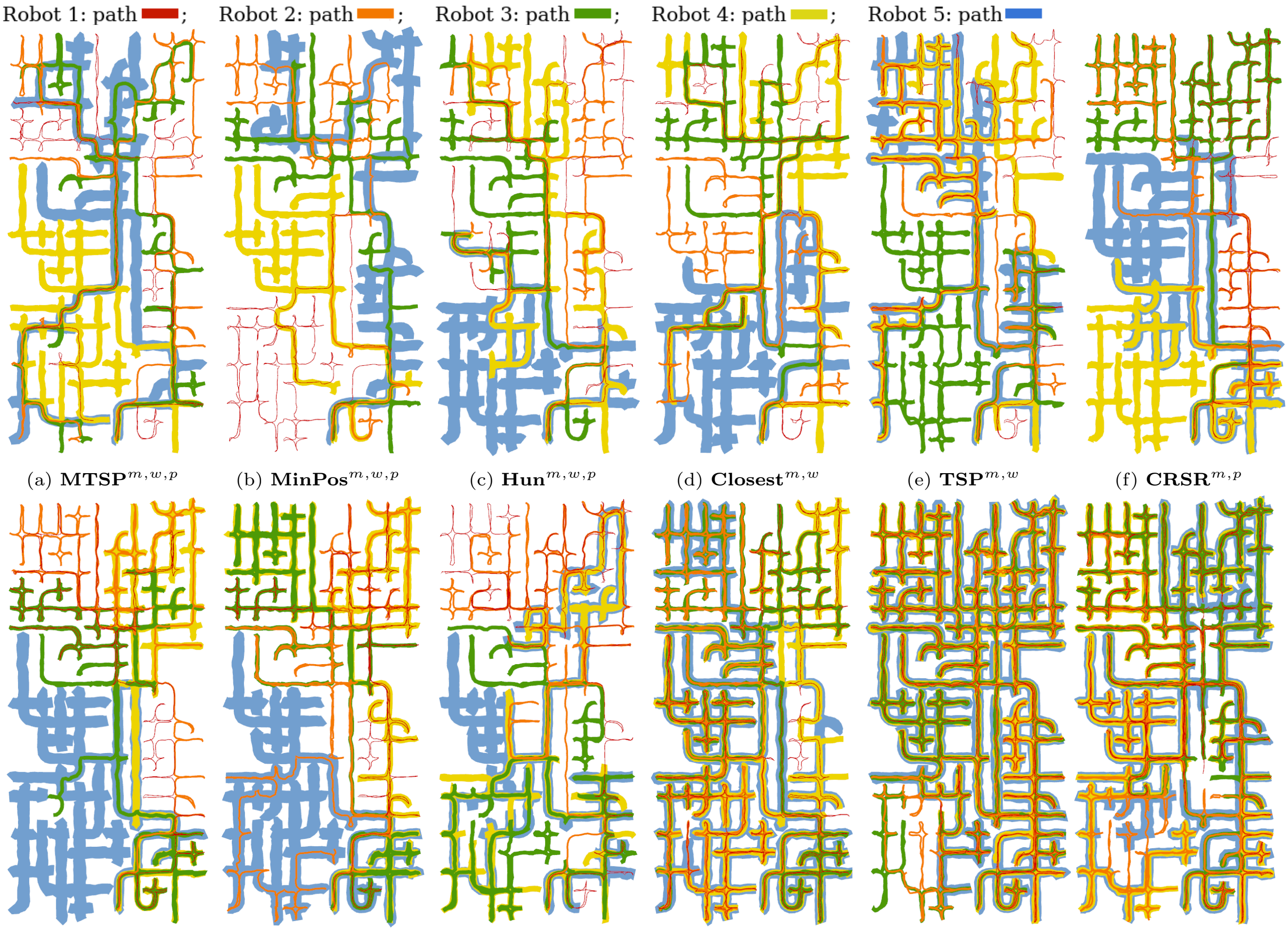
(a) **MTSP^{m,w,p}** (b) **MinPos^{m,w,p}** (c) **Hun^{m,w,p}** (d) **Closest^{m,w}** (e) **TSP^{m,w}** (f) **CRSR^{m,p}**

(g) **CRTSP^{p}** (h) **CRESR^{p}** (i) **ETSP^{m,w}** (j) **Closest** (k) **TSP** (l) **ETSP**

Fig. 18 Paths of the robots taken from the exploration with three robots in the environment S1 (first trial out of five).

Table 9 Evaluation of the exploration performance based on the average percentage of the environment covered by the individual robot

	Average coverage c_{avg} [%]											
Exploration			**3 robots**						**5 robots**			
strategy	**S1**	**S2**	**S3**	**S4**	**S5**	**All**	**S1**	**S2**	**S3**	**S4**	**S5**	**All**
MTSP^{m,w,p}	45±2	48±4	51±1	56±5	51±6	50	33±3	38±4	40±5	42±3	39±1	38
MinPos^{m,w,p}	42±2	46±1	53±3	51±5	49±4	48	33±2	37±1	41±3	43±4	44±3	40
Hun^{m,w,p}	43±1	52±3	52±1	55±4	54±6	51	34±2	34±4	42±5	44±5	40±2	39
Closest^{m,w}	46±4	52±3	57±6	60±9	57±9	54	37±3	42±4	47±5	49±6	42±4	43
TSP^{m,w}	57±6	49±5	54±5	68±9	53±8	56	40±5	46±2	46±4	54±5	50±3	47
CRSR^{m,p}	48±4	50±6	53±2	59±5	49±5	51	37±3	39±3	44±3	47±3	41±2	42
CRTSP^{p}	55±8	56±6	55±6	62±5	51±7	56	38±2	36±3	45±6	46±3	40±3	41
CRESR^{p}	45±4	51±4	50±3	52±6	53±4	50	38±4	39±3	44±2	43±2	40±3	41
ETSP^{m,w}	49±3	56±5	59±13	69±10	56±6	58	44±3	53±10	44±5	52±8	49±9	49
Closest	78±12	75±11	79±17	90±6	88±7	82	60±7	62±4	57±3	68±14	73±11	64
TSP	93±2	94±2	63±11	84±14	84±10	83	78±12	87±5	53±5	75±13	75±7	74
ETSP	93±4	83±11	70±16	73±10	95±1	83	80±8	70±9	58±5	81±11	87±6	75



(a) **MTSP**^{m,w,p} (b) **MinPos**^{m,w,p} (c) **Hun**^{m,w,p} (d) **Closest**^{m,w} (e) **TSP**^{m,w} (f) **CRSR**^{m,p}

(g) **CRTSP**^{p} (h) **CRESR**^{p} (i) **ETSP**^{m,w} (j) **Closest** (k) **TSP** (l) **ETSP**

Fig. 19 Paths of the robots taken from the exploration with five robots in the environment S1 (first trial out of five).

Acknowledgements. The presented work has been supported by the Czech Science Foundation (GAČR) under research project No. 22-05762S, the support for carrying out the extensive evaluation and the text finalization by the project of the Ministry of Interior of the Czech Republic No. VK01030216 is acknowledged.

Jan Bayer is a Ph.D. student at the Czech Technical University in Prague (CTU), Czech Republic. In 2019, he received his master's (Ing.) degree in cybernetics and robotics at the Faculty of Electrical Engineering (FEE), CTU. His master thesis won the first place in the IT Spy competition, which is awarded by the Czech and Slovak ACM chapters. Since then, he has worked under the supervision of Jan Faigl, who heads the Computational Robotics Laboratory, Artificial Intelligence Center, FEE, CTU. He has participated in DARPA SubT Challenge as a member of the CTU-CRAS-NORLAB team. His research is focused on mobile robot exploration, with a particular interest in multi-robot scenarios.

Jan Faigl is a Professor of Computer Science at the Faculty of Electrical Engineering (FEE), Czech Technical University in Prague (CTU). He received his Ph.D. in Artificial Intelligence and Biocybernetics in 2010 and his Ing. degree in Cybernetics in 2003, both from CTU. In 2013/2014, he was a Fulbright Visiting Scholar at the University of Southern California. He was awarded the Antonín Svoboda Award by the Czech Society for Cybernetics and Informatics in 2011. He served as the guest editor of the special issue on "Online decision making in Multi-Robot Coordination" in Autonomous Robots. He also served as the associate editor of the IEEE Transactions on Automation Science and Engineering (T-ASE) from 2019 to 2022. Since 2013, he has been leading the Computational Robotics Laboratory (CRL; https://comrob.fel.cvut.cz). He is co-founder of the Center for Robotics and Autonomous Systems (CRAS; https://robotics.fel.cvut.cz. He participated in the DARPA Subterranean (SubT) Challenge as co-leader of the CTU-CRAS-NORLAB team. In 2021, he received the Amazon Research Award. His current research interests include autonomous field navigation, robotic information gathering, and human-machine teaming.