Planning and Optimization for Multi-Robot Planetary Cave Exploration under Intermittent Connectivity Constraints

Filip Klaesson, Petter Nilsson, Tiago Vaquero,
Scott Tepsuporn, Aaron D. Ames and Richard M. Murray

Abstract

Exploring subsurface structures with autonomous robots is of growing interest in the context of planetary caves studies. Communication between robots in these environments is severely degraded which complicates coordination and information distribution. In this paper we focus on planning for mobility and communication in a cave exploration scenario where the situational awareness of a static base station is critical. We propose a notion of information-consistency where a plan itself is part of the information to be shared between robots, and propose a method for generating information-consistent plans. We discuss in detail how the resulting plan can be robustly implemented with minimal communication through local mission executives that run on individual robots. We investigate the performance of the planning algorithm, and integrate the local mission executives in a high-fidelity simulation environment.

Introduction

Planetary caves have been of increasing interest from the planetary science and robotics communities for their environmental and structural potential to host human habitats (Boston et al. 2003). Before humans can settle in Moon or Mars caves, such unknown subterranean structures will need to be well studied and mapped, potentially by teams of autonomous robots (Husain et al. 2013). However, robotic exploration in such underground environments brings several technical challenges to enable access and detailed investigation of their interiors. In this paper, we focus on the challenges associated with multi-robot cave exploration where the communication ability is poor and ranges are limited.

Multi-robot exploration not only provides means to extend data gathering and mapping operations into large, kilometer-long, hard-to-reach areas of planetary caves, but it also provides system redundancy and the capability of distributed deployment and spatiotemporal sensing. However, due to limited and intermittent communication in caves (between robots and to Earth), traditional operations used in current planetary surface robots would be infeasible. In this case, autonomy is a key enabler of such subsurface missions.

Existing multi-robot systems typically reside in the academic domain and are highly tuned to specific applications on Earth. The majority of these systems operate in controlled or moderate environments, compared to long rock-strewn subsurface voids. In this work we focus on communication-constrained cave exploration using a heterogeneous group of robots while keeping a base station updated about the exploration progress. Our main contributions are the construction of a novel execution plan designed for communication-constrained systems, and an integrated software framework for planning, optimization and execution in the context of multi-vehicle coordination for caves. The proposed planning system and framework are developed in the context of the Defense Advanced Research Projects Agency (DARPA) Subterranean Challenge, which provides unique opportunities for this line of work.

The rest of this paper is organized as follows. First we introduce the DARPA Subterranean Challenge and existing efforts on planetary and earth cave exploration. We then present the proposed approach for planning and optimization of robot exploration, communication and data sharing tasks. We also describe how the approach is integrated for plan execution and task coordination. We then present our experimental results and, finally, we conclude by discussing the results and future efforts.

Background

DARPA Subterranean Challenge. DARPA has traditionally developed technological challenges to push the development of ground breaking technologies in robotics and autonomy. Between 2018-2021, the agency has been running the DARPA Subterranean (SubT) Challenge[1] which aims to motivate scientists, engineers and developers to create new approaches to rapidly map, navigate, and search subsurface environments. DARPA SubT is one of the largest efforts in developing technologies for autonomous exploration of subterranean environments like tunnels, caves, and mines. It provides a unique opportunity to innovate on both robotics and automated task planning and coordination. While the main goal is disaster response, the developed technologies are of utmost importance and utility to the planetary science community.

In DARPA SubT, teams compete in three preliminary Circuit events and a Final event. The first three circuits are:

Copyright © 2020, California Institute of Technology. U.S. Government sponsorship acknowledged.

[1]DARPA SubT. https://www.subtchallenge.com/

large human-made tunnel systems, underground urban environments, and naturally occurring cave networks. The Final event in 2021 is designed to incorporate diverse challenges from all the three prior environments. Each of these events explore the challenges of operating robotic vehicles in their respective environments including communication, access and navigation in extreme terrains, autonomous mapping, and precise localization. The goal is to map and explore the subsurface structure in a time-critical scenario while identifying and localizing specific objects placed in environment, e.g. survivor, backpack, cellphone. Teams have to develop novel robotics platforms and autonomous behaviours to be deployed during the events. The robots are controlled from a base station at the entrance of the environment, where an operator can interact with the system.

Robotic Cave Exploration. Exploring subsurface structures with robots is a hard problem due to the unknown environment, difficult terrain for robots to traverse, anticipated communication challenges, and potentially limited power and lifetime. A significant amount of research on robotic cave exploration has focused on mobility. On that topic, research suggests that a successful mission design for a cave scenario might utilize multiple robots to provide redundancy and carry a mix of instruments with different capabilities. For example, Dubowsky, Plante, and Boston (2006) have investigated the use of a large numbers of small, cheap robots to be deployed over a large area. They suggest these smaller robots compared to rovers may move better in a cave environment and might be more robust to robot loss/failure. The same concepts of using small robots is also explored in (Kesner 2007; Kalita et al. 2018) where hopping robots are proposed.

Multi-robot exploration approaches have been increasingly investigated in the context of cave exploration, especially in time-critical missions. The multi-robot setting requires complex synergy between vehicle coordination, planning, information sharing and data routing in highly dynamic, unpredictable and potentially hazardous operating environments. In terrestrial applications, although several multi-robot coordination techniques have been developed for mapping and exploration of unknown environments (Yamauchi 1998; Sheng et al. 2006; Koes, Nourbakhsh, and Sycara 2005; Koes, Sycara, and Nourbakhsh 2006), efficient ways to explore such environments is still an open area of research. Most of existing work target environments that are usually quite structured and communication constraints that are not realistic or non-existent.

In the context of planetary exploration, only a handful of works have investigated multi-robot coordination, task allocation and communication strategies (Nilsson et al. 2018; Vaquero, Troesch, and Chien 2018). Vaquero, Troesch, and Chien (2018) proposed the *Dynamic Zonal Relay with Sneakernet Relay Algorithm*. In this algorithm, rovers spread out into a linear cave, each one following its neighbor, while collecting and sending data towards a base station at the entrance. They incrementally extend further into the cave as data is transferred out. The routing decision is based purely on the existence of a connection with a rover that is closer to the base station. At some point, the rovers spread out enough that the communication connection between rovers is lost, and they need to drive back and forth to transfer data out of the cave. The preliminary experiments performed with this algorithm in simulated linear caves indicated that transferring data is a major source of energy usage during the mission and that more data could potentially be sent out if data transfer was managed and routed more intelligently.

In (Husain et al. 2013), the proposed multi-robot coordination framework also targets planetary caves and consists of an autonomous frontier and capability-based task generator, a distributed market-based strategy for coordinating and allocating tasks to the different robots, and a communication system for sharing information between the robots. The work covers insights on systems integration and coordination for semi-realist cave scenarios; however, intermittent communication is not considered as a constraint. In our work, intermittent communication plays a major role on robot planning and coordination, as well as information and data sharing.

Problem Statement

In this paper we consider scenarios where a team of mobile robots $R = \{r_1, \ldots, r_{|R|}\}$ is deployed to explore and map an unknown cave environment. At the entrance of the cave there is a stationary base station that is responsible for coordinating the movement of the mobile robots by updating and merging data collected inside the cave. A benefit of delegating these potentially computationally intensive processing tasks to the base station is that the mobile robots can rely on hardware that consumes less power. However, inter-robot communication in a cave is generally only possible between robots that are close, which makes distribution of information from and to the base station a core element of the problem.

Exploration involves sending robots to *frontiers*, which are locations at the boundary between free and unknown space, and from there letting them venture further into unknown space. As long as there are frontiers remaining, further exploration is possible. As exploration takes place, robots need to update the base station with the progress, and may also need to regroup to newly found frontiers. This necessitates a communication mechanism for robots to coordinate their movements as a team.

By the nature of the problem, planning is only possible in explored free space. However, as the free space is expanded it becomes increasingly important to plan efficient behaviors for getting robots to frontiers, and for carrying new information to the base station. Assume that free space is represented by an expanding graph structure called a *mobility-communication network* $\mathcal{N} = (S, \rightarrow, \rightsquigarrow)$ that consists of a finite set of locations $s \in S$, a set of *mobility edges* $\rightarrow \subset S \times S$, and a set of *communication edges* $\rightsquigarrow \subset S \times S$. A robot located at s can move to a location s' provided that $(s, s') \in \rightarrow$, and if two robots r_1 and r_2 are in locations s_1 and s_2 such that $(s_1, s_2) \in \rightsquigarrow$, then r_1 can share information with r_2. The network $\mathcal{N}$ should be thought of as a topological map of the known free space, constructed so that a local

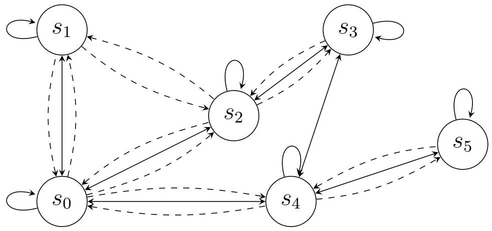

Figure 1: Mobility-communication network $\mathcal{N}$. Mobility edges are solid and communication edges are dashed.

planner can steer robots between locations connected by mobility edges. As the known space is expanded, new locations are added to $\mathcal{N}$, similarly to how a probabilistic roadmap is expanded. Figure 1 shows a mobility-communication network where mobility edges are indicated with solid arrows and communication edges with dashed arrows.

The objective in this article is to devise a multi-robot planner that synthesizes a plan of robot behaviors/actions (e.g., move between locations, transfer data) on a mobility-communication network that: i) sends robots to frontiers so that they can perform exploration, and ii) periodically updates the base station in s_{base} about the progress.

Since the planner runs at the base station, and the environment is communication-constrained, distributing a newly computed plan to the mobile robots is nontrivial and must be accounted for in the plan. To conserve energy, and to optimize the time that can be spent exploring, it is desirable to promote data-muling behaviors where robots collaborate to transport information to and from the base station.

Approach

To account for communication limitations it is necessary to not only plan for robot motion, but also plan for how information is shared between robots. For this reason we let a *plan* $\Pi = (\Pi_t, \Pi_c)$ consist of two components: a *trajectory collection* $\Pi_t = \{s_0^r s_1^r \dots s_T^r\}$ for $r \in R$, such that $(s_t^r, s_{t+1}^r) \in \rightarrow$, and a *communication collection* $\Pi_c = \{(t, r, r', b)\}$. A trajectory is simply a sequence of locations that can be followed by moving along mobility edges. The communication collection consists of tuples (t, r, r', b), which indicate that at time t robot $r \in R$ sends the information labeled b to robot r'. The information label b refers to an agent that possessed some information at time $t = 0$ that is to be shared with other agents.

A couple of requirements must be satisfied for a communication collection Π_c to be *valid* with respect to the trajectories Π_t. Firstly, a piece of communication (t, r, r', b) can only happen if the robots are in the same location, i.e. $s_t^r = s_t^{r'}$, or if there is a communication link between the robot locations, i.e. $(s_t^r, s_t^{r'}) \in \rightsquigarrow$. Secondly, robot r must possess the information at time t which is true if either $b = r$, i.e. r had the information at $t = 0$, or if r has received information b at some time $t' \leq t$. Finally, for a fixed b and t there may be no communication cycles in the induced directed graph, which ensures that information does not appear out of nowhere. To promote efficient behaviors we let the *cost* of a plan $\Pi = (\Pi_t, \Pi_c)$ be

$$C(\Pi) = \sum_{r \in R} \sum_{t=0}^{T-1} \bar{C}_t(s_t^r, s_{t+1}^r) + \sum_{(t,r,r',b) \in \Pi_t} \tilde{C}_t(s_t^r, s_t^{r'}), \tag{1}$$

where $\bar{C}_t(s, s')$ is the *mobility cost* of moving from s to s' over a mobility edge, and $\tilde{C}_t(s, s')$ is the *communication cost* of sending information from s to s' over a communication edge.

As the free space is expanded, old frontiers disappear but may be replaced with new ones that are found deeper into the cave, which necessitates frequent re-planning to account for the changing circumstances. Restricting all robots to be within communication range when a new plan is computed would limit efficiency or potentially be infeasible, for instance in large, kilometer-long caves. Instead, the plan itself can be regarded as part of the information (i.e. a data product) that is to be shared among robots, but this needs to be accounted for in planning. In particular, the plan can not assume any actions by robots that are not yet aware of the plan; those robots must remain idle until the new plan is communicated to them. We call this property of a plan *information-consistency*. We let the piece of information consisting of the plan itself be denoted m for *master*.

Definition 1. Let a subset of robots $R_m \subset R$ be *master robots* that have knowledge of information m at time $t = 0$. A plan $\Pi = (\Pi_t, \Pi_c)$ is *information-consistent* if:

- A robot without information m does not move;
- A robot without information m does not send information

We divide the problem described in the previous section into three phases: a *pre-exploration phase* that sends robots to frontiers, an *exploration phase*, and a *post-exploration phase* where robots coordinate to send data found in exploration back to the base station. As noted above, only the pre- and post-exploration phases are amenable to planning. Since the robots are not necessarily in communication range at the end of the exploration phase, both pre- and post-exploration must be planned before the execution of the overall plan starts. Later on we suggest a behavior for the exploration phase, but for now we focus on planning for pre- and post-exploration.

Problem 1 (Pre-exploration planning)**.** Consider a network $\mathcal{N}$, initial robot positions s_0^r for $r \in R$, frontier locations $S_f \subset S$, and a time horizon T. Devise an information-consistent multi-robot plan $\Pi = (\Pi_t, \Pi_c)$, using the base station as master, such that

$$-C(\Pi) + \sum_{s \in S_f} \begin{cases} \mathfrak{R}(s, k), & \text{if } \sum_{r=1}^{R} \mathbf{1}_s(s_T^r) = k, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

is maximized.

Here the function $\mathfrak{R}(s, k)$ denotes the reward associated with sending k robots to a frontier $s \in S_f$. Typically it is better to have more robots at a frontier, but the marginal reward for additional robots is decreasing, i.e. $\mathfrak{R}(k + 1) <$

$2\mathfrak{R}(k) - \mathfrak{R}(k-1)$, since the frontier may turn out to be small or narrow enough to be explored by a single robot.

After executing the pre-exploration plan some robots arrive at frontiers, and can begin exploring. Due to the decentralized nature of the exploration phase, we require robots to return to the frontier from where they started exploring, since newly explored locations are not known at the time of planning. From those frontiers we then execute a behavior that coordinates transmission of the newly found information (i.e., locations found in exploration) to the base station.

Problem 2 (Post-exploration planning)**.** Consider a network $\mathcal{N}$, initial robot positions s_0^r for $r \in R$, a subset of robots $R_f \subset R$ with new information, a subset of master robots $R_m \subset R$, and a time horizon T. Devise an information-consistent multi-robot plan Π, using R_m as master robots, such that the base station receives information b for all $b \in R_f$, and such that

$$-C(\Pi) \tag{3}$$

is maximized.

In this second problem we allow for multiple master agents: all agents that received master information m in the pre-exploration plan can potentially share information m with robots that remained idle in pre-exploration but that are useful in post-exploration.

In what follows we outline a method to solve these two problems via integer linear optimization. Subsequently, we describe a mission executive for individual robots that handle mobility and communication to achieve collective execution of a plan $\Pi = (\Pi_t, \Pi_c)$. The protocol is robust to delays which is important for decentralized and asynchronous plan execution.

Planning and Optimization

In order to plan for efficient information distribution we use *intermittent connectivity*. This is a flexible notion of information sharing that allows for directed information transfer. Flexibility is especially important in large-scale networks such as caves (Wyatt et al. 2018; Vaquero et al. 2019), where strict communication constraints such as continuous and recurrent connectivity (Rooker and Birk 2007; Banfi et al. 2018) can prohibit robots from performing the main exploration objective. In the following we summarize the main elements of intermittent connectivity from (Klaesson et al. 2019).

Intermittent Connectivity and Integer Linear Programs. The intermittent connectivity problem is to maximize an objective whilst satisfying intermittent connectivity constraints. Given a mobility-communication network, consider two sets of robots $\texttt{src} \subset R$ and $\texttt{snk} \subset R$. The intermittent connectivity constraint associated with the pair $(\texttt{src}, \texttt{snk})$ is satisfied by a plan Π if each robot in $\texttt{snk}$ receives information from each robot in $\texttt{src}$. In its most basic form, the intermittent connectivity problem is as follows:

Inputs: Network $\mathcal{N}$, master robots R_m, initial positions s_0^r, connectivity constraint $(\texttt{src}, \texttt{snk})$, optimization objective, time horizon T.

Output: Information-consistent plan $\Pi = (\Pi_t, \Pi_c)$ that satisfies the intermittent connectivity constraint.

It is clear that this problem encompasses both Problem 1 (pre-exploration) and Problem 2 (post-exploration).

Intermittent connectivity constraints can be written as linear constraints on variables that model flows of information in the network. Flow capacities can furthermore be constrained as a function of robot positions, and the cost function can be written as a linear expression, which allows posing a single integer linear program (ILP) that optimizes the objective subject to intermittent connectivity. Information consistency can be accounted for in this framework via additional linear constraints that prevent robots from moving or communicating before the information flow associated with the master robot(s) has arrived to their initial locations.

In addition there are multiple extensions relevant in the exploration setting. For instance, a set of robots $R_{static} \subset R$ can be constrained to remain static during the execution, naturally including the base station as well as non-mobile communication relays that are deployed in the cave. Furthermore, collision avoidance constraints can be included for a subset of robots $R_{col} \subset R$ and a subset of locations $S_{col} \subset S$, so that robots in R_{col} cannot simultaneously occupy certain locations or traverse the same mobility edge. Moreover, it is possible to account for heterogeneous capabilities among the robots; we can for example restrict exploration to a subset of robots $R_{col} \subset R$ that are equipped with the necessary sensing instruments. We refer to (Klaesson et al. 2019) for a detailed discussion on intermittent connectivity problems and how they can be solved as ILP.

Clustering for Large-Scale Networks. To cope with large-scale intermittent connectivity problems, a clustering method was proposed in (Klaesson et al. 2019). The clustering method divides the network into smaller clusters as illustrated in Figure 2 and poses local intermittent connectivity problems that can be patched together to form a feasible, but in general sub-optimal, solution to the original problem. We develop two improvements to the clustering approach.

First of all, the starting time of pre-exploration problems and the end time of post-exploration problems can be aligned to maximize the amount of time spent in the exploration phase in each cluster. We patch the cluster executions together with a common initial and final time as indicated by Figure 3. The cluster with the longest joint pre-exploration and post-exploration execution times thus have the shortest amount of exploration time, and in order to allow this cluster to contribute to the network expansion we set this exploration time to a *minimum exploration time* and adjust the exploration times for other clusters accordingly.

Secondly, in order to solve the coordination problem on larger graphs we extend the clustering method by disregarding locations that are unlikely to be used in the solutions. A location is marked as `idle` if *i)* all its mobility neighbours further away from the base station are `idle`, *ii)* it is not a frontier, and *iii)* does not contain a robot, leading to a recursive algorithm for identifying `idle` locations. Similarly, if a cluster does not contain any frontiers and does not have any non-idle child clusters, all locations in the cluster are considered `idle`. This is handled by planning evacuation behaviors that reward robots in idle clusters to move towards

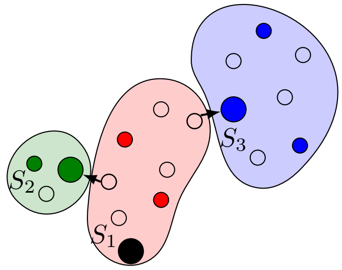

Figure 2: Illustration of clustering for exploration. Locations are depicted with circles, and occupied locations are filled. The master agent is marked with a black circle. The master cluster S_1 has two child clusters S_2 and S_3.

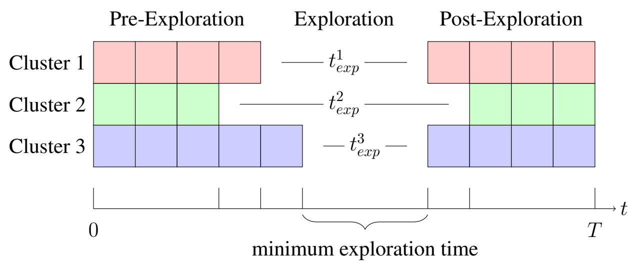

Figure 3: Patching pre-exploration and post-exploration plans with varying exploration times in each cluster.

active locations where they are more likely to contribute to the exploration effort. This prevents robots from being stuck in undesirable locations and not utilized efficiently.

Execution and Coordination

Above we discussed how information-consistent plans $\Pi = (\Pi_t, \Pi_c)$ that solve Problem 1 and 2 can be synthesized. However, for a plan to be useful it must be converted into behaviors for the individual robots that lead to a collective effort that is in accordance with the plan. In this section we propose a plan execution behaviour for individual robots that leads to collective plan execution, and that exhibits robustness to delays and asynchrony.

For this work, we used the TRACE mission executive (de la Croix et al. 2017). This executive is part of the NASA Jet Propulsion Laboratory's CARACaS (Control Architecture for Robotic Agent Command and Sensing) autonomomy architecture (Wolf et al. 2017) where it is responsible for the execution and monitoring of processes modelled in Business Process Modelling Notation (BPMN) (OMG 2011). BPMN is a standard user-oriented graphical language specifically designed for modelling processes, including multi-threaded flowchart-like processes. In addition to graphical elements (which are similar to state machines and UML activity diagrams), the language provides execution semantics which is heavily utilized in this work. The table in Figure 4 describes the BPMN elements and the corresponding tasks used in this work to implement the plan execution behavior.

Given a BPMN model, the executive will dispatch tasks based on the specified task flow and events. Herein, as opposed to modeling the information-consistent plan directly as a sequence of task, we follow a more abstract approach in which the BPMN represents the planning process as a task and the plan execution as loop processes over the list of actions in the plan. We can then execute any arbitrary plan that contains the target set of actions considered in this work (i.e. explore, transit and communicate). Figure 4 shows the BPMN diagram used by each robot's executive. The BPMN service tasks are as follows, with `OK` begin normal task termination, and `EBE` being triggering of an Error Boundary Event (c.f. Figure 4):

- **Leader?**: If r is responsible for planning, return `OK`, else return `EBE`. In this work, the base station is the lead robot.
- **COPS**: Solve the pre- and post-exploration problems by calling the planner.
- **Wait to receive plan**: Remain idle until another robot sends plan to r.
- **Explore?**: If r is in the exploration phase, return `OK`, otherwise return `EBE`.
- **Communication Action**: Perform communication action as prescribed in the plan, following the communication protocol described below. Return `OK` when completed.
- **Transition Action**: Move to location s^r_{t+1} on $\mathcal{N}$ and increase time step, $t = t + 1$.
- **Finished Plan?**: If r has completed its portion of the plan return `OK`, otherwise return `EBE`.
- **Exploration Action**: Perform frontier exploration as described below.
- **Transit to Return Node**: Return to location from where exploration started.
- **Exploration Timer** and **Action Timer**: The timer tasks simply return `OK` after a certain amount of time has elapsed, which is used to limit the exploration behavior to the allocated exploration time (Exploration Timer), and to synchronize behaviors across robots to prevent collisions (Action Timer).

In addition to the aforementioned tasks, there is a separate *information storage task* (omitted in the diagram) that runs in the background to keep track of the information that robot r is in possession of.

Running a mission executive at each robot r that implements the BPMN in Figure 4 results in collective execution of a plan. This approach is robust to delays due to the mechanism of not going into **Transition Action** that transfers the robot from s^r_t to s^r_{t+1} until all communication associated with time step t is completed. However, it is inevitable that delays propagate throughout the system. To reduce delays the network $\mathcal{N}$ is built in such a way that all mobility transitions take approximately the same time. In what follows we further describe the communication and exploration actions in some detail, and provide some of the required aspects for integrating the executive into the software ecosystem developed for the DARPA SubT competition.

Symbol	Action
	Start Event: Starts a process.
	End Event: Ends a process.
	Terminate End Event: Terminates all processes.
	Error Boundary Event: Triggered if an task error occurs.
	Signal Throw Task: Throws a specific signal.
	Signal Catch Event: Catches a specific signal.
	Exclusive Gateway: Proceeds with first input-process.
	Parallel Gateway: Breaks one process into multiple processes, and merge multiple processes into one process by waiting for all input-processes.
	Service Task: Performs a task when triggered by a process.

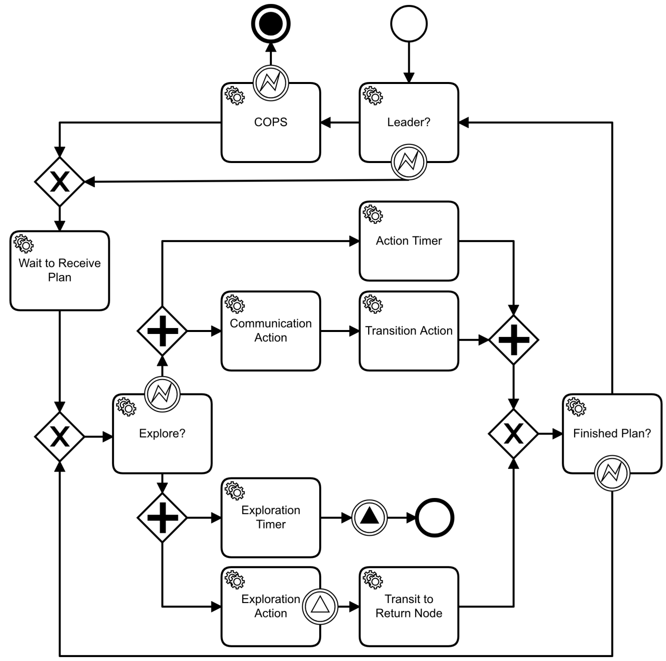

Figure 4: standard BPMN semantics and the BPMN autonomy scheme executed by each robot.

Communication Action: To limit unnecessary communication we distinguish between *internal* communication among robots within a location and *external* communication across communication edges.

To minimize utilization of low-bandwidth communication links it is desirable to design the **Communication Action** to avoid unnecessary long-distance communication. When there are multiple robots present at the same location we therefore select one as the *communication leader*. The role of the leader is to collect all information possessed by robots at the location, share that information across communication edges to other communication leaders in accordance with the plan, and finally distribute any new information that was received from other locations to the robots at the same location. After these steps all robots at a given location have the same information. We call these three communication phases: pre internal communication (`PreIntCom`), external communication (`ExtCom`), and post internal communication (`PostIntCom`).

We denote the communication leader of location s at time t by $l_t^s \in R$, and the set of all communication leaders at time t by L_t. For a time step t and robot r we can then extract the communication that occurs in each of the three phases described above:

- $\texttt{PreIntCom}(t, r) = \{(t, r_1, r_2, b) : r \in \{r_1, r_2\} \wedge s_t^{r_1} = s_t^{r_2} \wedge r_2 = l_t^s | s = s_t^{r_1} = s_t^{r_2}\}$
- $\texttt{ExtCom}(t, r) = \{(t, r_1, r_2, b) : r \in \{r_1, r_2\} \wedge s_t^{r_1} \neq s_t^{r_2} \wedge r \in L_t\}$
- $\texttt{PostIntCom}(t, r) = \{(t, r_1, r_2, b) : r \in \{r_1, r_2\} \wedge s_t^{r_1} = s_t^{r_2} \wedge r_1 = l_t^s | s = s_t^{r_1} = s_t^{r_2}\}$

Given this categorization of communication involving robot r at time t, the **Communication Action** behavior is implemented by completing the three types of communications in sequence. In particular, at time t robot r continuously listens for incoming information messages and sends a confirmation if a message is received. At first robot r broadcasts outgoing $\texttt{PreIntCom}(t, r)$ messages, and does so until all outgoing $\texttt{PreIntCom}(t, r)$ messages have been accepted by other robots, and all incoming $\texttt{PreIntCom}(t, r)$ messages have been received. Then the same procedure is executed for $\texttt{ExtCom}(t, r)$, and finally for $\texttt{PostIntCom}(t, r)$. Note that the set $\texttt{ExtCom}(t, r)$ is non-empty only for local communication leaders.

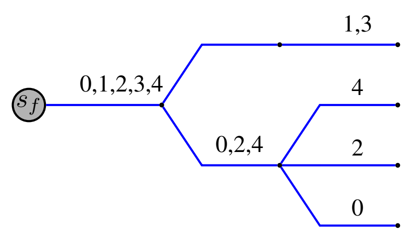

Figure 5: Illustration of robots distributed in different frontier forks via Algorithm 1. Five robots $r_{k_0}, \ldots, r_{k_4}$ are assigned to the frontier s_f. At the first fork r_{k_0}, r_{k_2} and r_{k_4} go right, and the other two go left. The indices are updated, and when the next choice appears the three robots in the right fork all select different paths to explore.

Exploration Action: The robot's Exploration Action is simple: identify a region between known and unknown space and move to it, and repeat this until there is no more unknown space. However, when there are multiple robots exploring the same area it is desirable that they avoid overlaps. For example, if a fork is encountered robots should split up to cover both the left and right parts of the fork. We suggest a communication-free behavior that results in robots spreading out for exploration.

Consider a single frontier location s_f and assume that the

Algorithm 1: Exploration Action

Data: Robot frontier index i
while *True* **do**
  $c \leftarrow$ number of choices;
  $\texttt{choice} \leftarrow i \mod c$;
  $i \leftarrow i/c$;
  Move towards $\texttt{choice}$ for time ΔT ;
end

pre-exploration plan sends n robots $r_{k_0}, r_{k_1}, \ldots, r_{k_{n-1}}$ to that location (potentially arriving at different times). Each robot r_{k_i} then follows the behavior in Algorithm 1 which only requires knowledge of the index i pertaining to the robot. As long as there is only one way to proceed ($c = 1$), the index i does not change, but if there are multiple choices ($c > 1$) the robots split evenly among the choices, and the index i is updated to reflect the number of robots that have taken the same path. As long as all robots identify the same possible paths, this algorithm works without communication even if robots start their exploration behaviors at different times. The algorithm is illustrated in Figure 5.

Integration into SubT Framework. This work is only concerned with planning on the abstract graph structure we call mobility-communication network. To implement the method in practice additional capabilities are required, in particular local planning and controls to traverse between locations, mapping and localization, as well as incremental construction of the abstract graph structure. These are all challenging tasks that are beyond the scope of this paper, but we comment briefly on the major integration requirement of the proposed planning and execution system.

To maintain awareness in the system it is necessary to periodically synchronize the information collected by the different robots. In the framework presented in this paper it is natural to do this at the base station, which is the first "agent" that is guaranteed to periodically receive all information. Thus, when a cycle of pre-exploration, exploration, and post-exploration has been completed, the base station is tasked with merging the different extensions into a single network, and potentially also calculate loop closures to maintain map coherency. When this is completed a new cycle is planned for the updated environment model.

Experimental Results

In this section we analyze the planning approach presented in this paper. As our approach aims to increase the amount of collaboration and reduce the amount of transitions, while also being time efficient, we consider how these properties are affected by the number of robots in the network. The structure of the map is illustrated by Figure 6. With a distance between two locations of 50 meters, the maximum depths of a fork is approximately 1 kilometer.

Setup

Consider a mobility-communication network with 100 locations. Initially we only have knowledge about a single location where the base station is located. We add $|R| - 1$

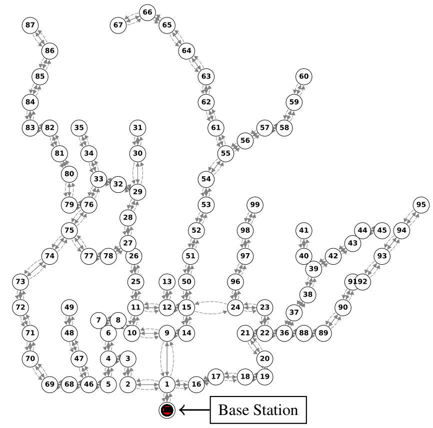

Figure 6: Tunnel-like network structure of 100 locations including a stationary base station.

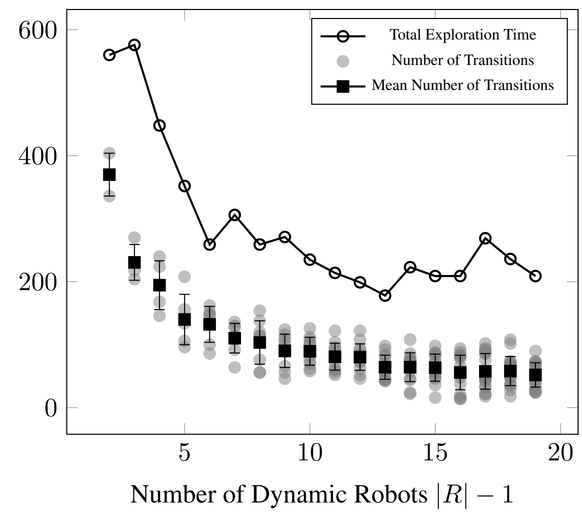

Figure 7: Y-axis represents: *Number of Transitions* for each robot, the *Mean Number of Transitions* and standard deviation, and the *total exploration time* as a function of number of robots in the network. The robots perform exploration of a network of 100 locations.

robots initially places in the same location as the base station, and initiate exploration of the network via cycles of pre-exploration, exploration, and post-exploration. We allow all robots to explore and do not enforce any collision avoidance constraints, i.e. $R_{static} = R_m = \{\text{base station}\}$, $R_{exp} = R$ and $R_{inc} = S_{inc} = \emptyset$.

Performance

First we investigate how the total number of transitions and the total exploration time depends on the number of robots in the network. Figure 7 shows the number of transitions for each dynamic robot in gray, the mean number of transitions together with the standard deviation in black, as a function of number of dynamic robots. Both the number of transitions and the exploration time decreases with an increased number of robots. However, for ≥ 14 dynamic robots, the total exploration time and the mean number of transitions

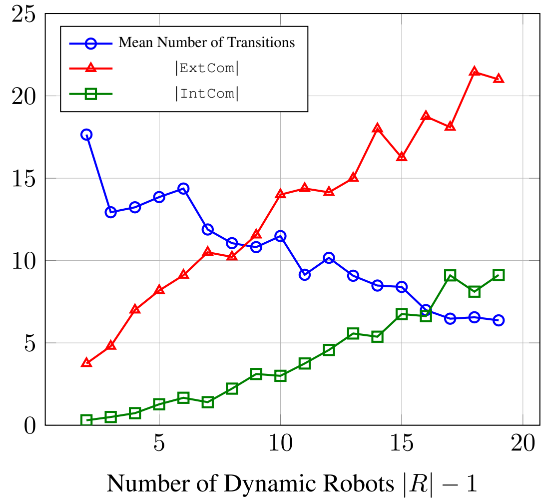

Figure 8: Y-axis represents: *Mean Number of Transitions* per plan, $|\texttt{ExtCom}|$ as the number of external communications per plan, and $|\texttt{IntCom}|$ as the number of external communications per plan, as a function of the number of dynamic robots. The robots perform exploration of a network of 100 locations.

no longer decreases. The increased variance for ≥ 14 robots indicates that several robots are not used. Therefore, it is not beneficial energy-wise to add these additional robots in this particular environment.

A main component of this work is for robots to utilize the communication edges in a collaborative manner in order to decrease the number of transitions for each robot in order to be able to explore larger graphs. Figure 8 shows the mean number of transitions and mean number of both internal and external communication per plan, where $\texttt{IntCom} = \texttt{PreIntCom} \bigcup \texttt{PostIntCom}$. The average number of transitions per robot and iteration decreases when adding more robots. Figure 8 also indicates that the decrease in number of transitions is associated with an increase in communication, especially external communication using communication edges.

SubT Software Integration

The planning approach presented in this paper is implemented as a component in ROS Melodic (Quigley et al. 2009) and simulated using the Gazebo Simulator (Koenig and Howard 2004). The plans are computed using the COPS toolbox for information-consistent planning in communication-constrained multi-agents networks using intermittent connectivity (Klaesson and Nilsson 2019). Figure 9 shows two Husky robots (Clearpath Robotics, Inc. 2019) and a base station exploring a cave structure approximately 3 meters wide. The simulation includes constructing multiple plans, exploring frontiers and updating the base station in multiple iterations. Figure 9 shows the graph structure being transmitted to the base station through a relay configuration of the two robots. The testings with Gazebo helped us to validate the integration of the planning system together with mobility, localization, mapping and sensing components.

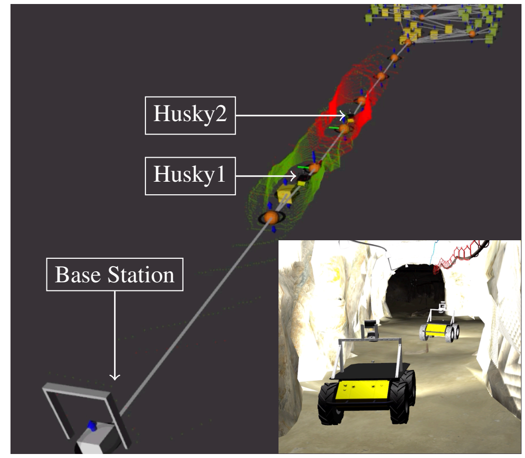

Figure 9: Base station is updated with information about the exploration through a relay configuration of two Huskies.

Conclusion

This paper considers the problem of constructing multi-robot plans for exploration in severely communication-constrained networks. Motivated by the DARPA SubT Challenge, we focus on exploring an unknown environment while maintaining high situational awareness in a base station. Due to limited communication ability, we propose information-consistent plans to ensure feasible information distribution. The plan is divided into three phases: the pre-exploration and post-exploration phases that ensure periodically situational awareness in the base station, and the exploration phase where robots explore unknown space.

We present a novel implementation of the plan execution by constructing behaviors robust to delays. The plan is modeled by a process scheme in Business Process Modelling Notation and we utilize the mission executive TRACE for executing processes. A communication protocol that aims to minimize the amount of long-distance communication is presented. For this purpose, the communication action is divided into three parts: pre-internal, external and post-internal communication. The performance analysis of the planning approach shows that adding robots to the network results in a decrease of the number of robot transitions, an increase of the number of communications, and a decrease of the total exploration time. Therefore, we conclude that planning using intermittent connectivity constraints is suitable for large-scale multi-robot exploration in communication-constrained environments such as caves and tunnels.

Several areas of future work have been identified while conducting this study. First and foremost, if a robot collapses the plan becomes infeasible and the execution would end. This can be handled by a lower-level local planner or by adding redundancy to the plan. Another concern is if the transition time between locations vary significantly, it would be favorable to consider the transition time when planning.

References

Banfi, J.; Quattrini Li, A.; Rekleitis, I.; Amigoni, F.; and Basilico, N. 2018. Strategies for coordinated multirobot exploration with recurrent connectivity constraints. *Autonomous Robots* 42(4):875–894.

Boston, P.; Frederick, R.; Welch, S.; Werker, J.; Meyer, T.; Sprungman, B.; Hildreth-Werker, V.; Thompson, L.; and Murphy, D. 2003. Human utilization of subsurface extraterrestrial environments. *Gravitational and Space Biology Bulletin* 26(2).

Clearpath Robotics, Inc. 2019. Husky unmanned ground vehicle. https://clearpathrobotics.com/.

de la Croix, J.-P.; Lim, G.; Vander Hook, J.; Rahmani, A.; Droge, G.; Xydes, A.; and Scrapper Jr, C. 2017. Mission modeling, planning, and execution module for teams of unmanned vehicles. In *Unmanned Systems Technology XIX*, volume 10195, 101950J. International Society for Optics and Photonics.

Dubowsky, S.; Plante, J.-S.; and Boston, P. 2006. Low cost micro exploration robots for search and rescue in rough terrain. In *IEEE International Workshop on Safety, Security and Rescue Robotics, Gaithersburg, MD, USA*.

Husain, A.; Jones, H.; Kannan, B.; Wong, U.; Pimentel, T.; Tang, S.; Daftry, S.; Huber, S.; and Whittaker, W. L. 2013. Mapping planetary caves with an autonomous, heterogeneous robot team. In *IEEE Aerospace Conference*, 1–13.

Kalita, H.; Morad, S.; Ravindran, A.; and Thangavelautham, J. 2018. Path planning and navigation inside off-world lava tubes and caves. In *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 1311–1318.

Kesner, S. 2007. Mobility feasibility study of fuel cell powered hopping robots for space exploration. Master's thesis, Massachusetts Institute of Technology (MIT).

Klaesson, F., and Nilsson, P. 2019. COPS: Connectivity planning software for communication constrained networks. https://github.com/FilipKlaesson/cops.

Klaesson, F.; Nilsson, P.; Ames, A. D.; and Murray, R. M. 2019. Intermittent connectivity for exploration in communication-constrained multi-agent systems. arXiv:https://arxiv.org/abs/1911.08626 [cs.RO].

Koenig, N., and Howard, A. 2004. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *In IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2149–2154.

Koes, M.; Nourbakhsh, I.; and Sycara, K. 2005. Heterogeneous multirobot coordination with spatial and temporal constraints. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 3*, AAAI'05, 1292–1297. AAAI Press.

Koes, M.; Sycara, K.; and Nourbakhsh, I. 2006. A constraint optimization framework for fractured robot teams. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Nilsson, P.; Haesaert, S.; Thakker, R.; Otsu, K.; Vasile, C.-I.; akbar Agha-Mohammadi, A.; Ames, A. D.; and Murray, R. M. 2018. Toward specification-guided active mars exploration for cooperative robot teams. In *Proceedings of Robotics: Science and Systems Conference*.

OMG. 2011. Business Process Model and Notation (BPMN), Version 2.0.

Quigley, M.; Conley, K.; Gerkey, B. P.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; and Ng, A. Y. 2009. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*.

Rooker, M. N., and Birk, A. 2007. Multi-robot exploration under the constraints of wireless networking. *Control Engineering Practice* 15(4):435 – 445.

Sheng, W.; Yang, Q.; Tan, J.; and Xi, N. 2006. Distributed multi-robot coordination in area exploration. *Robot. Auton. Syst.* 54(12):945–955.

Vaquero, T.; Troesch, M.; Net, M. S.; Gao, J.; and Chien, S. 2019. Energy-aware data routing for disruption tolerant networks in planetary cave exploration. In *11th International Workshop on Planning and Scheduling for Space (IWPSS)*, 186–193. Also appears at the 29th International Conference on Automated Planning and Scheduling (ICAPS 2019) Workshop on Planning and Robotics (PlanRob).

Vaquero, T.; Troesch, M.; and Chien, S. 2018. An approach for autonomous multi-rover collaboration for mars cave exploration: Preliminary results. In *International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS 2018)*. Also appears at the ICAPS PlanRob 2018.

Wolf, M. T.; Rahmani, A.; de la Croix, J.-P.; Woodward, G.; Hook, J. V.; Brown, D.; Schaffer, S.; Lim, C.; Bailey, P.; Tepsuporn, S.; Pomerantz, M.; Nguyen, V.; Sorice, C.; and Sandoval, M. 2017. CARACaS multi-agent maritime autonomy for unmanned surface vehicles in the Swarm II harbor patrol demonstration. In Karlsen, R. E.; Gage, D. W.; Shoemaker, C. M.; and Nguyen, H. G., eds., *Unmanned Systems Technology XIX*, volume 10195, 218 – 228. International Society for Optics and Photonics.

Wyatt, E. J.; Belov, K.; Castillo-Rogez, J.; Chien, S.; Fraeman, A.; Gao, J.; Herzig, S.; Lazio, T. J. W.; Troesch, M.; and Vaquero, T. 2018. Autonomous networking for robotic deep space exploration. In *International Symposium on Artificial Intelligence, Robotics, and Automation for Space (ISAIRAS 2018)*.

Yamauchi, B. 1998. Frontier-based exploration using multiple robots. In *Proceedings of the Second International Conference on Autonomous Agents*, AGENTS '98, 47–53. New York, NY, USA: ACM.