Intermittent Connectivity for Exploration in Communication-Constrained Multi-Agent Systems

Filip Klaesson, Petter Nilsson, Aaron D. Ames and Richard M. Murray

***Abstract*—Motivated by exploration of communication-constrained underground environments using robot teams, we study the problem of planning for intermittent connectivity in multi-agent systems. We propose a novel concept of information-consistency to handle situations where the plan is not initially known by all agents, and suggest an integer linear program for synthesizing information-consistent plans that also achieve auxiliary goals. Furthermore, inspired by network flow problems we propose a novel way to pose connectivity constraints that scales much better than previous methods. In the second part of the paper we apply these results in an exploration setting, and propose a clustering method that separates a large exploration problem into smaller problems that can be solved independently. We demonstrate how the resulting exploration algorithm is able to coordinate a team of ten agents to explore a large environment.**

I. Introduction

In many multi-agent applications such as mapping, frontier exploration, and search-and-rescue missions, communication between agents is critical to gain awareness of the situation. Significant effort has been devoted to study connectivity in continuous models [1], [2], [3]. Although such models are closer to the actual dynamics of robots, solution techniques become difficult to apply in complex geometries. This paper considers an alternative approach of planning on a discrete graph model that abstracts away low-level dynamics but, in our opinion, retains the core elements of the connectivity problem.

Various solutions have been proposed to maintain connectivity between agents in environments like tunnel and cave systems that suffer from severe communication constraints [4], [5], [6]. The fundamental difference between these methods is the type of connectivity constraint enforced between agents. The situational awareness of agents increases with the strictness of the constraint while a more relaxed constraint allows them to perform individual tasks to a higher degree.

The strictest connectivity constraint is the *continuous* connectivity constraint which demands that all agents are fully connected at all times [7]. This implies that two arbitrary agents in the network can share data with each other at any time. Although this results in global awareness, it can degrade the objective performance significantly. *Recurrent* connectivity [8] is a relaxation of continuous connectivity that allows agents to occasionally disconnect and instead enforces global connectivity only at a specific time instant (such as the final time), which allows more flexibility at the cost of awareness.

However, connectivity constraints like continuous and recurrent connectivity that occasionally enforce global connectivity do not perform well in exploration tasks if the environment is large relative to the number of agents. For instance, the exploration setting studied in [8] incorporates a fixed base station that needs to be updated of the exploration progress. In this case the recurrent connectivity constraint implies that agents should be connected to the base station even when they reach frontiers. If a frontier is further away than the reach of the longest fully connected configuration it is therefore not feasible to explore it, and even if the frontier is within reach recurrent connectivity may be inefficient since a large number of agents may be required to establish connectivity and therefore can not partake in exploration.

Motivated by exploration settings where the environment is large in relation to the number of robots we propose a novel *intermittent* connectivity constraint that formalizes data distribution behaviors. Intermittent connectivity does not ever impose simultaneous global connectivity; it only requires a way to pass data between certain agents within some time horizon. This allows agents to transport data and distribute it between each other in a collaborative manner. A stricter version of intermittent connectivity was formulated via linear temporal logic in [9], together with a way to generate trajectories. Our formulation is more permissive as it does not presuppose a division of agents into teams, and does not require all members of a team to meet repeatedly at the same location. Instead we allow for targeted information distribution which leaves more flexibility to perform actual exploration tasks. We also consider the case when the plan itself is part of the data that needs to be distributed, i.e., agents that have not yet received this data are not aware of the plan and can therefore not be expected to act in accordance with it. We handle this situation by designating a *master* agent with initial knowledge of the plan, and pose *master constraints* that ensure that the plan is consistent with information distribution constraints.

As previous work [8], [10], [11], [12], [13] we plan multi-agent trajectories via an Integer Linear Program (ILP). Connectivity problems have also been solved with the (mixed) ILP approach: [14] proposed a MILP that keeps static source states connected over time. Inspired by ideas from network flows we introduce new linear constraints that guarantee our relaxed intermittent connectivity criterion and show that this formulation has better scalability properties than those found in previous work; the number of constraints scales linearly with the graph size instead of combinatorially. However, since ILPs are NP-hard even the improved flow formulation encounters scalability issues for large graphs. To mitigate this issue we propose a clustering method tailored to exploration problems that decomposes a large problem into smaller problems that can be solved more efficiently.

The authors are with California Institute of Technology, Pasadena, CA.

II. Preliminaries and Problem Formulation

In this section we first present preliminaries in order to state the intermittent connectivity problem that we seek to solve.

A. Environment model

When constructing the environment model the possibility of communication between locations needs to be considered. Determining whether communication is possible between two locations could be done via models, data, or both. In order to not restrict the framework we do not specify a communication condition and instead use an abstract mobility-communication-network as a model of the environment.

Definition 1. A mobility-communication-network $\mathcal{N}$ is defined as the tuple $\mathcal{N} = (S, \rightarrow, \rightsquigarrow)$:

- S: finite set of states s,
- $\rightarrow \subset S \times S$: set of directed mobility edges (s, s'),
- $\rightsquigarrow \subset S \times S$: set of directed communication edges (s, s').

The states $s \in S$ correspond to locations in the environment and a mobility edge $(s, s') \in\rightarrow$ denotes a directed traversable path from s to s'. The existence of a communication edge $(s, s') \in\rightsquigarrow$ certifies that an agent located in s can send data to an agent located in s'. We assume that agents in the same state are able to share information with each other. It is also assumed that $\rightsquigarrow$ does not contain false positives.

For optimization purposes we introduce edge weights C and $\widetilde{C}$ that associate weights with the edges in $\rightarrow$ and $\rightsquigarrow$, respectively. Likewise, state rewards $\mathfrak{R}$ are used to associate a reward to reaching certain locations. We seek to find finite-time trajectories and denote the time horizon by T.

A key component to formalize the intermittent connectivity constraint is the time-extended version of the mobility-communication network, which is built by stacking copies of the network and associating each layer with a time step. The mobility edges in each layer are rewired across time steps, as illustrated in Fig 1. In the time-extended graph agents therefore traverse over time steps but communicate within time steps.

We introduce some simplifying notation. Given a mobility-communication-network $\mathcal{N} = (S, \rightarrow, \rightsquigarrow)$, denote the set of mobility and communication predecessors of state s as $C^-_{\rightarrow}(s) = \{s' \mid (s', s) \in\rightarrow\}$ and $C^-_{\rightsquigarrow}(v) = \{s' \mid (s', s) \in\rightsquigarrow\}$. Likewise, denote the set of mobility and communication successor of state s as $C^+_{\rightarrow}(s) = \{s' \mid (s, s') \in\rightarrow\}$ and $C^+_{\rightsquigarrow}(v) = \{s' \mid (s, s') \in\rightsquigarrow\}$. It's also useful to look at both mobility and communication predecessors and successors; denote the set of all predecessors by $C^-(s) = C^-_{\rightarrow}(s) \cup C^-_{\rightsquigarrow}(s)$ and the set of all successors by $C^+(s) = C^+_{\rightarrow}(s) \cup C^+_{\rightsquigarrow}(s)$.

B. Intermittent connectivity

An intermittent connection allows the agents to disconnect, move around, and establish connections to other agents in the network. Two agents 1 and 2 can therefore share data with each other without ever being directly connected if other agents set up a sequence of intermittent connections such that data is transferred from 1 to 2, and vice versa. A simple example is shown in Fig. 1 where the red and blue agents share information with each other using a sequence of intermittent connections via the green agent. We seek to formalize this type of intermittent connectivity and start by defining a *mobility path* which encodes an agent trajectory.

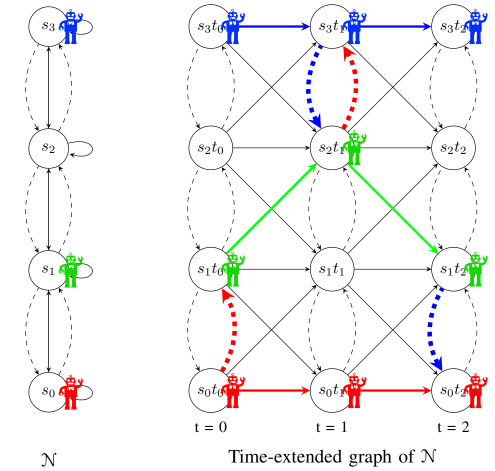


Fig. 1. Mobility-communication-network $\mathcal{N}$ and the time-extended graph of $\mathcal{N}$ for $T = 2$. Mobility edges are solid and communication edges are dashed. A solution for the problem of sharing data between the blue and red agents are shown in the time-extended graph. The colors of the communication edges indicate the origin agent of the data that is transmitted.

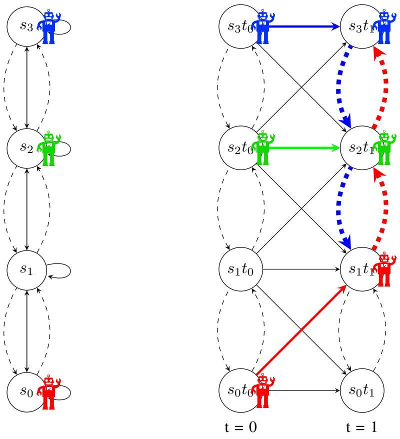
Definition 2. Given a mobility-communication-network $\mathcal{N} = (S, \rightarrow, \rightsquigarrow)$, a *mobility path* is a sequence $\pi : s_0 s_1 s_2 \dots s_T$ of states $s \in S$ such that $(s_t, s_{t+1}) \in\rightarrow$.

We use a superscript r to specify the mobility path for agent r by $\pi^r = s_0^r s_1^r s_2^r \dots s_T^r$. Information can be transferred spatially via two mechanisms: *(i)* agents can traverse mobility edges and carry data with them, and *(ii)* agents can transmit information to other agents over communication edges. These two types of information transfer guides us to the notion of an *information path*.

Definition 3. Given a mobility-communication network $\mathcal{N} = (S, \rightarrow, \rightsquigarrow)$, an *information path* is a sequence $\bar{\pi} : \mathfrak{s}_0^0\mathfrak{s}_0^1 \dots \mathfrak{s}_0^{n_0}\mathfrak{s}_1^0\mathfrak{s}_1^1 \dots \mathfrak{s}_1^{n_1} \dots \mathfrak{s}_T^{n_T}$ of states $\mathfrak{s}_t^i \in S$ such that $(\mathfrak{s}_t^{n_t}, \mathfrak{s}_{t+1}^0) \in\rightarrow$ and $(\mathfrak{s}_t^i, \mathfrak{s}_t^{i+1}) \in\rightsquigarrow$. In addition, an *agent-to-agent information path* $\bar{\pi}^{i,j}$ is an information path that begins in the initial state of agent i and ends in the final state of agent j, i.e., $\mathfrak{s}_0^0 = s_0^i$ and $\mathfrak{s}_T^{n_T} = s_T^j$.

Note that only one mobility transition is allowed for each time step t, while information can potentially travel over multiple communication edges within a time step via a multi-hop link. Fig. 2 shows an example where the red and blue agents share information with each other using multiple communication edges within a time step by sending information via the green agent. To associate information paths with agent mobility we introduce the notation of a *valid* information path.

Definition 4. Consider a network $\mathcal{N} = (S, \rightarrow, \rightsquigarrow)$ and a collection of mobility paths $\{\pi^r : s_0^r s_1^r s_2^r \dots s_T^r\}$. An information path $\bar{\pi} : \mathfrak{s}_0^0\mathfrak{s}_0^1 \dots \mathfrak{s}_0^{n_0}\mathfrak{s}_1^0\mathfrak{s}_1^1 \dots \mathfrak{s}_1^{n_1} \dots \mathfrak{s}_T^{n_T}$ is *valid* with respect to the mobility paths $\{\pi^r\}$ if the following conditions are



$\mathcal{N}$

Time-extended graph of $\mathcal{N}$

Fig. 2. Mobility-communication-network $\mathcal{N}$ and the time-extended graph of $\mathcal{N}$ for $T = 1$. A solution for the problem of sharing data between the blue and red agents are shown in the time-extended graph.

satisfied for all $t = 0, \ldots, T$:

$$\left\{\mathfrak{s}_t^i \mid i = 0, \ldots, n_t\right\} \subseteq \left\{s_t^r \mid r = 1, \ldots, R\right\}, \quad (1)$$

$$(\mathfrak{s}_t^{n_t}, \mathfrak{s}_{t+1}^0) \in \left\{\left(s_t^r, s_{t+1}^r\right) \mid r = 1, \ldots, R\right\}. \quad (2)$$

Equation (1) implies that there must be agents present at all locations that are used to transmit information over communication edges, and equation (2) specifies that if information moves along a mobility edge, then at least one agent must undertake that transition at the same time step.

An advantage of intermittent connectivity is that it allows for directed information transfer. In many applications we are only interested in transferring information between specific agents instead of sharing all information among all agents. A useful formulation used throughout this work is to enforce a subset of agents to transfer their information to all agents in another subset. We denote the set of agents initially having access to important information by sources $\texttt{src} \subset \{1, \ldots, R\}$ and the set of agents that are to receive information by sinks $\texttt{snk} \subset \{1, \ldots, R\}$. We require that all agents in $\texttt{snk}$ should receive information from all agents in $\texttt{src}$, i.e. that there should exist an agent-to-agent information path $\bar{\pi}^{i,j}$ from every agent $i \in \texttt{src}$ to each agent $j \in \texttt{snk}$. We are now ready to formulate the intermittent connectivity problem.

Problem 1 (Intermittent Connectivity)**.** Consider a network $\mathcal{N}$ with R agents, initial conditions $\{s_0^r\}$, and time horizon T. Find mobility paths $\{\pi^r\}$ such that for each pair $(i, j) \in \texttt{src} \times \texttt{snk}$ there exists an agent-to-agent information path $\bar{\pi}^{i,j}$ that is valid with respect to $\{\pi^r\}$ and such that

$$\mathfrak{R} - \sum_r \sum_{t=0}^{T-1} C(t, s_t^r, s_{t+1}^r) - \sum_{b \in \texttt{src}} \sum_{(t,s,s') \in E_b} \widetilde{C}(t, s, s') \quad (3)$$

is maximized, where $\mathfrak{R}$ is a reward function that gives rewards for sending agents to states:

$$\mathfrak{R} = \sum_{s \in S} \sum_{k=1}^{R} \begin{cases} \mathfrak{R}(s, k), & \text{if } \sum_{r=1}^{R} \mathbf{1}_{s_T^r = s} \geq k, \\ 0, & \text{otherwise}, \end{cases} \quad (4)$$

where $\mathbf{1}$ is the indicator function, and E_b contains communication edges used in information paths from source b:

$$E_b = \left\{(t, \mathfrak{s}_t^k, \mathfrak{s}_t^{k+1}) \in \rightsquigarrow \mid \mathfrak{s}_t^k, \mathfrak{s}_t^{k+1} \in \bar{\pi}^{b,j}, j \in \texttt{snk}\right\}. \quad (5)$$

The objective function (3) penalizes agent transitions and information transmission using the edge weights C and $\widetilde{C}$, while rewarding the final state of agents. In some applications it can be beneficial to send multiple agents which is why the state reward of reaching a state s in the final time step depend on the number of agents that reach s. Note that collecting reward $\mathfrak{R}(s, k)$ for sending k agents to s implies that all rewards $\mathfrak{R}(s, i)$ for $i < k$ are also received.

C. Information-Consistent Intermittent Connectivity

A potential issue with Problem 1 is that all agents may not be in communication range when a new plan is to be executed. If the optimal solution to Problem 1 is unique, all agents could solve the problem and arrive at the same solution, but this does not work if the solution is non-unique or if not all agents are computationally equipped to solve the problem. In this scenario a solution could be to solve a single instance of Problem 1 and then distribute the resulting plan to all agents via intermittent connectivity. However, there are two effects that need to be considered when planning in this fashion: agents that have not yet received the plan can not be expected to *(i)* move from their initial position and *(ii)* transmit information to other agents. These restrictions need to be accounted for in the planning step. We assume that there is a *master* agent that computes the plan and thus knows it at $t = 0$, but that other agents can not act until they have received information from the master agent. In the following, information paths originating from the master agent are denoted ${}^m\bar{\pi}$.

Definition 5. Consider a mobility-communication-network $\mathcal{N} = (S, \rightarrow, \rightsquigarrow)$ and an information path ${}^m\bar{\pi}$. A mobility path $\pi = s_0 s_1 s_2 \ldots s_T$ is *information-consistent* with respect to ${}^m\bar{\pi}$ if for all $t = 0, \ldots, T$ it holds that

$$s_0 \notin \left\{\mathfrak{s}_j^i \in^m \bar{\pi} \mid 0 \leq j \leq t,\ 0 \leq i \leq n_j\right\} \implies s_t = s_0. \quad (6)$$

Similarly, an information path $\bar{\pi} = \mathfrak{s}_0^0 \mathfrak{s}_0^1 \ldots \mathfrak{s}_0^{n_0} \ldots s_T^{n_T}$ is *information-consistent* with ${}^m\bar{\pi}$ if for all $t = 0, \ldots, T$,

$$\mathfrak{s}_t^l \notin \left\{\mathfrak{s}_j^i \in {}^m\bar{\pi} \mid 0 \leq j \leq t,\ 0 \leq i \leq n_j\right\} \implies n_t = l. \quad (7)$$

Equation (6) implies that an agent is not allowed to move until the information path ${}^m\bar{\pi}$ has reached the initial state s_0. Since an information path consists of one mobility edge in each time step, enforcing $n_t = l$ as in (7) denies the usage of any communication edge at time t and stage l. Thus an agent at state s_t^l is not allowed to transmit information until the information path ${}^m\bar{\pi}$ has reached that state. We are now ready to formulate the information-consistent version of Problem 1.

Problem 2 (Information-Consistent Intermittent Connectivity)**.** Consider a network $\mathcal{N}$ with R agents, initial conditions $\{s_0^r\}$, time horizon T, and a designated master $m \in \{1, \ldots, R\}$. Find a collection of mobility paths $\{\pi^r\}$ such that:

- for each pair of $\mathtt{src}$ i and $\mathtt{snk}$ j there exists an information path $\bar{\pi}^{i,j}$ that is valid with respect to $\{\pi^r\}$,
- for each agent r there exists an information path ${}^m\bar{\pi}_r$ starting in s_0^m that is valid with respect to $\{\pi^r\}$,
- all mobility paths are information-consistent with ${}^m\bar{\pi}_r$,
- all information paths are information-consistent with ${}^m\bar{\pi}_i$,

and such that

$$\mathfrak{R} - \sum_r \sum_{t=0}^{T-1} C(t, s_t^r, s_{t+1}^r) - \sum_{b\in\mathtt{src}\cup\{m\}} \sum_{(t,s,s')\in E_b} \widetilde{C}(t,s,s') \quad (8)$$

is maximized, where $\mathfrak{R}$ is given by (4) and E_b by (5).

Problem 2 is analogous to Problem 1 but with the additional requirement that for each agent r, there should exist a valid information path ${}^m\bar{\pi}_r$ that makes the mobility path π^r and all information paths $\bar{\pi}^{r,j}$ information-consistent.

III. Intermittent Connectivity via Flows

In combinatorial optimization, network flows are instrumental for solving many graph optimization problems. Examples include the maximal flow and minimum-cost flow problems [15]. Flow problems generally involve finding a feasible flow that optimizes some cost function. A flow is feasible if it starts in a source state, ends in sink states, and the flow over each edge is less than its capacity. We note that the intermittent connectivity problem has similarities with the multi-commodity flow problem with multiple flows, sources and sinks. Both Problem 1 and 2 involve finding multiple information paths with predefined initial states. However, there are two significant differences between a standard flow problem and the intermittent connectivity problems: in the intermittent connectivity problems, the terminal state for each information path is the location of an agent instead of a static state, and edge capacities depend on the movement of agents.

A. Intermittent Connectivity Problem

In the following we introduce variables and constraints that form an ILP that solves Problem 1.

1) Variables: We start by introducing optimization variables. Since there are multiple information flows we annotate flow variables with a *flow identifier* (or id).

Definition 6. Define variables z_{rst}, y_{sk}, x_{rijt}, f_{ijt}^b, $\bar{f}_{ijt}^b$ as

$$z_{rst} = \begin{cases} 1 \text{ if agent } r \text{ is at state } s \text{ at time } t, \\ 0 \text{ otherwise,} \end{cases} \quad (9)$$

$$y_{sk} = \begin{cases} 1 \text{ if there are } \geq k \text{ agents in } s \text{ at time } T, \\ 0 \text{ otherwise,} \end{cases} \quad (10)$$

$$x_{rss't} = \begin{cases} 1 \text{ if agent } r \text{ uses } (s,s') \in\rightarrow \text{ at time } t, \\ 0 \text{ otherwise,} \end{cases} \quad (11)$$

$$f_{ss't}^b = \text{flow over } (s,s') \in\rightarrow \text{ with id } b \text{ at time } t, \quad (12)$$

$$\bar{f}_{ss't}^b = \text{flow over } (s,s') \in\rightsquigarrow \text{ with id } b \text{ at time } t. \quad (13)$$

The variables z_{rst} and $x_{rss't}$ are binary variables that encode positions and transitions of agents at each time step while y_{sk} counts how many agents that occupy certain states at the final time step. To represent information transmitted over mobility and communication edges we use $f_{ss't}^b$ and $\bar{f}_{ss't}^b$.

2) Dynamics constraints: For space reasons we omit the sets of "for all" quantifiers in the following. That is, instead of $\forall r \in \{1,\ldots,R\}$, $\forall s \in S$, and $\forall t \in \{0,\ldots,T\}$ we simply write $\forall r$, $\forall s$, and $\forall t$. We can encode the dynamics of the agents in the mobility-communication-network $\mathcal{N}$ using the variables (9) and (11) [8]. The possible transitions at each time can be written as follows.

$$z_{rs(t+1)} = \sum_{s'\in\mathbf{C}_{\rightarrow}^{-}(s)} x_{rs'st}, \quad \forall r,s,t, \quad (14)$$

$$z_{rst} = \sum_{s'\in\mathbf{C}_{\rightarrow}^{+}(s)} x_{rss't}, \quad \forall r,s,t. \quad (15)$$

Equation (14) means that an agent transiting to a state s must do so via an incoming mobility edge $(s',s) \in\rightarrow$ and equation (15) specifies that an agent at state s must use an outgoing mobility edge $(s,s') \in\rightarrow$.

3) Flow constraints: In order to establish information paths from each $\mathtt{src}$ to each $\mathtt{snk}$ we look at the net inflow of information with identifier b to a state. To simplify notation, let F_{st}^b denote the net inflow to state s at time t of flow b:

$$\begin{aligned} F_{st}^b = &\sum_{s'\in C_{\rightarrow}^{-}(s)} f_{s's(t-1)}^b + \sum_{s'\in C_{\rightsquigarrow}^{-}(s)} \bar{f}_{s'st}^b \\ &- \sum_{s'\in C_{\rightarrow}^{+}(s)} f_{ss't}^b - \sum_{s'\in C_{\rightsquigarrow}^{+}(s)} \bar{f}_{ss't}^b. \end{aligned} \quad (16)$$

We can construct a one-to-many flow from one agent $i \in \mathtt{src}$ to all agents $j \in \mathtt{snk}$ and thus incorporate all information paths $\bigcup_{j\in\mathtt{snk}} \bar{\pi}^{i,j}$ into a single flow. In this case the identifier b for the flow corresponds to the source agent. The resulting constraint can be written as follows.

$$F_{st}^b = \begin{cases} -|\mathtt{snk}| z_{bs0}, & \text{if } t = 0, \\ \sum_{r\in\mathtt{snk}} z_{rsT}, & \text{if } t = T, \\ 0, & \text{otherwise,} \end{cases} \quad \forall b \in \mathtt{src}, s, t. \quad (17)$$

We note that the state including the source agent initially outputs $|\mathtt{snk}|$ units of flow and that the net inflow to any state is zero for $0 < t < T$. This ensures that information can not vanish or be created and that all information initially comes from the source agents. The condition for the final time T implies that a state with k agents from $\mathtt{snk}$ at time T receives k units of this flow. Therefore each agent in $\mathtt{snk}$ is guaranteed to receive one unit of the flow with identifier b. As this constraint should hold for all information paths we enforce this constraint for all source agents $b \in \mathtt{src}$. Note that in constraint (17) the terminal state of each agent is free and is determined by z_{rst}, as opposed to classical flow problems where it is predefined.

4) Bridge constraints: As previously mentioned, in classical flow problems the capacity of each edge is static and predefined. However, in a mobility-communication network with dynamic agents, the capacity of each edge is dynamic and a function of the position of the agents. For communication edges, information can be transmitted over an edge $(s, s') \in \rightsquigarrow$ at time t only if there is an agent in state s and another agent in state s' at time t. This can be written as the big-M constraint (18) where $N \geq R$. For a mobility edge, information can only be transferred over the edge if an agent traverses it. Therefore, the flow over a mobility edge $(s, s') \in \rightarrow$ at time t must be zero unless at least one agent utilizes that edge, which can be written as the big-M constraint (19).

$$\bar{f}^b_{ss't} \leq N \min\left(\sum_r z_{rst}, \sum_r z_{rs't}\right), \quad \forall b, s, s', t, \tag{18}$$

$$f^b_{ss't} \leq N \sum_r x_{rss't}, \quad \forall b, s, s', t. \tag{19}$$

5) Cost function: To implement the objective function in Problems 1 and 2, we can penalize state transitions and information transmission costs C and $\bar{C}$ using the variables (9)-(13). The cost of using the mobility (resp. communication) edge (s, s') at time t can be written as $x_{rss't}C(t, s, s')$ (resp. $\bar{f}^b_{ss't}\bar{C}(t, s, s')$). We can formalize the reward of reaching state s with k agents in the final time as $\mathfrak{R}(s, k)y_{sk}$. We now apply the cost and reward for all time steps, states, edges, and agents and get the following objective function that is linear in the variables introduced above:

$$c_{\text{obj}} = \sum_{s \in S}\left(\sum_{k=1}^{R} y_{sk}\mathfrak{R}(s, k) - g_1(s) - g_2(s)\right), \tag{20}$$

$$g_1(s) = \sum_{\substack{r \in \{1,\dots,R\} \\ s' \in C^-_{\rightarrow}(s) \\ t \in \{0,\dots,T-1\}}} x_{rs'st}C(t, s', s), \tag{21}$$

$$g_2(s) = \sum_{\substack{s' \in C^-_{\rightarrow}(s) \\ t \in \{1,\dots,T\} \\ b \in \texttt{src}}} \bar{f}^b_{s'st}\bar{C}(t, s', s). \tag{22}$$

Note that for the transition and communication costs, only the incoming edges C^- are used to prevent double-counting of costs. Collecting a reward is restricted by y_{sk} to agents final state, therefore the reward $\mathfrak{R}$ in Problem 1 and 2 is

$$\mathfrak{R} = \sum_{s \in S}\sum_{k=1}^{R} y_{sk}\mathfrak{R}(s, k).$$

It follows by inspection that

$$\sum_r \sum_{t=0}^{T-1} C(t, s^r_t, s^r_{t+1}) = \sum_{s \in S} g_1(s),$$
$$\sum_{b \in \texttt{src} \cup \{m\}} \sum_{(s,s') \in E_b} \widetilde{C}(t, s', s) = \sum_{s \in S} g_2(s).$$

Therefore (20) is equivalent to the objective function (3) in the intermittent connectivity problem. If we change the index set to $b \in \texttt{src} \cup \{m\}$ in (22), then (20) is equivalent to the information-consistent objective function (8).

As the terminal state reward $\mathfrak{R}(s, k)$ can only be collected if there are k agents in state s at time T we need to constrain y_{sk}. Enforcing the following constraint for all k:s associated with rewards is equivalent with Definition 10.

$$y_{sk} \leq \frac{\sum_r z_{rsT}}{k} \quad \forall s. \tag{23}$$

We are now ready to pose a solution to Problem 1.

Proposition 1. *Consider the ILP:*

$$\begin{aligned} \max \quad & (20), \\ \text{subject to} \quad & (14), (15), (17) - (19), (23). \end{aligned} \tag{24}$$

A solution to (24) is also a solution to Problem 1.

Proof. By the argument in Section III-A5, equation (20) is equivalent to the cost function in Problem 1 when (23) holds. By [8] constraints (14) and (15) result in dynamically feasible mobility paths. Under (17), for each agent $r \in \texttt{src}$ there exists a flow over the time-extended graph that starts at s^r_0 and that has sinks at the locations at time T of the $\texttt{snk}$ agents. The flow decomposition theorem [15, Theorem 3.5] implies that this flow can be decomposed into information paths, and these information paths are valid since (18) and (19) prohibit flow over edges that violate the validity requirement. □

Remark 1. *We can pose an alternative constraint to (17) that also guarantees existence of the desired information paths. Instead of constructing a one-to-many flow, we can use a many-to-one flow from all agents $i \in \texttt{src}$ to a single agent $j \in \texttt{snk}$. The interpretation of the identifier b for this flow then changes to the single sink agent $j \in \texttt{snk}$. This formulation would yield a different cost associated with the communication edges in the objective in Problems 1 and 2 as the sum would be over sinks instead of sources. This many-to-one flow is formalized by the following constraint:*

$$F^b_{st} = \begin{cases} -\sum\limits_{r \in \texttt{src}} z_{rs0}, & \textit{if } t = 0, \\ |\texttt{src}|z_{bsT}, & \textit{if } t = T, \\ 0, & \textit{otherwise}, \end{cases} \quad \forall b \in \texttt{snk}, s, t. \tag{25}$$

Each one of the constraints (17) and (25) ensure the existence of an information path from each source $r \in \texttt{src}$ to all sinks $r \in \texttt{snk}$. Constraint (17) results in a total of $|\texttt{src}||S|(T+1)$ constraints and constraint (25) in $|\texttt{snk}||S|(T+1)$. It may be advantageous to select the one with the lowest number.

B. Information-Consistent Intermittent Connectivity Problem

We now seek to extend the result in Proposition 1 to a solution for Problem 2. The difference between Problem 1 and 2 is the existence of yet another information path ${}^m\bar{\pi}_r$ for each agent r that the mobility path π^r and all information paths $\bar{\pi}^{r,j}$ are information-consistent with. In the following we will refer to the information path ${}^m\bar{\pi}_r$ as the master information path for r as it is initialized by the master agent. In the same manner as in Section III-A, we associate the union of the master information paths $\bigcup_r {}^m\bar{\pi}_r$ with a flow in the network, naturally called the *master flow*. Denote the net inflow of the master flow into state s at time t by F^m_{st}, defined as in (16).

1) Master constraints: The main difference between a regular flow F^b and the master flow F^m is that the amount of flow F^b and its terminal state are predefined in terms of agent positions, leading to equality constraints in (17) and (25). For the master flow, there is no predefined amount of flow nor fixed terminal state. Therefore, we cover the worst case where master flow is supplied to every state in the network, leading to the initialization of at most $|S|$ units of master flow in the state containing the master agent. As there are no hard constraints on the terminal state for the master flow, we use inequality constraint to allow any terminal state. Let $s_0(r)$ denote the initial state of agent r and let m denote the master agent; the constraint on the master flow can be formalized as follows:

$$F^m_{st} \geq \begin{cases} -|S|, & \text{if } s = s_0(m), t = 0, \\ 0, & \text{otherwise,} \end{cases} \qquad \forall s, t. \tag{26}$$

In order to make sure that mobility path π^r and all information paths $\{\bar\pi^{r,j}\}$ are information consistent with some master information path ${}^m\bar\pi_r$ we need to enforce that agent r is *(i)*, static, and *(ii)*, does not transmit information to other agents until its initial position intersects with ${}^m\bar\pi_r$. This can be ensured with the following constrains that should hold for all r such that $s_0(r) \neq s_0(m)$:

$$z_{rs_0(r)t} \geq 1 - \sum_{\tau=0}^{t-1} F^m_{s_0(r)\tau}, \quad \forall r, t, \tag{27}$$

$$N \sum_{\tau=0}^{t} F^m_{s_0(r)\tau} \geq \sum_j \bar f^b_{s_0(r)jt}, \quad \forall b, r, t. \tag{28}$$

These constraints require a net outflow of the master flow to have occurred, as captured by the sum of net flow over time, before mobility (eq. (27)) or communication (eq. (28)) is allowed. Together with flow balancing (26) these constraints imply information consistency as defined in Section II-C.

Proposition 2. *Consider the ILP:*

$$\begin{aligned} \max \quad & (20) \\ \text{subject to} \quad & (14), (15), (17) - (19), (23), (26) - (28) \end{aligned} \tag{29}$$

A solution to (29) *is also a solution to Problem* 2.

Proof. Problem 2 is a further constrained version of Problem 1, thus Proposition 1 applies for the constraints that are common between the two. As motivated above, constraints (26)-(28) in addition ascertain the information consistency requirements in Problem 2. □

C. Extensions

In order to demonstrate the flexibility of the optimization framework, this section demonstrates multiple extension relevant to multi-agent networks.

1) Agent Heterogeneity: In a multi-agent exploration setting, agents might be equipped with different types of sensors so that only a subset of the agents are capable of exploring and mapping unknown parts of the environment, while other agents perform tasks in the known environment. Therefore it can be useful to only allow a subset of agents to collect rewards by reaching a terminal state. This is can be solved by only summing over agents capable of exploring in constraint (23). Agents may also have different mobility capabilities (e.g., ground and aerial agents), which can be accommodated by imposing additional constraints that specify that certain mobility edges can only be used by certain agents.

In systems where it is useful to have static agents, such as exploration with a static base station, stationarity must be considered in the planning. This is incorporated by not allowing an agent to leave its initial position using the constraint

$$z_{rst} = \begin{cases} 1, & \text{if } s = s_0(r), \\ 0, & \text{otherwise,} \end{cases} \qquad \forall t, r \in R_{static}, \tag{30}$$

where R_{static} is the set of static agents.

2) Collision Avoidance: In multi-agent networks, we can guarantee that two agents i and j can not be in the same position or use facing edges by enforcing the following constraints:

$$x_{iss't} + x_{js'st} \leq 1 \ \forall t, (s, s') \in \rightarrow, (i, j) \in \binom{\{1, \dots, R\}}{2}, \tag{31}$$

$$z_{ist} + z_{jst} \leq 1 \ \forall t, s, (i, j) \in \binom{\{1, \dots, R\}}{2}, \tag{32}$$

where $\binom{\{1,\dots,R\}}{2}$ is the set of all possible 2-combinations of agents in the network. In a heterogeneous group of agents, this constraint might only be necessary for a certain subsets of agents and vertices in the network. As an example, an application may require that ground agents avoid other ground agents, that aerial agents avoid other aerial agents, but allow ground and aerial agents to be in the vicinity of each other.

3) Awareness-Based Reward: In applications such as exploration or mapping it might not be enough to only have agents in beneficial positions unless they also have information about its duties, such as exploration a new area or acting as a relay for other agents. It is therefore natural to require that agents have knowledge about the execution plan in order to collect rewards. This is not an issue in Problem 1 as all agents plan deterministically, but for Problem 2 we have to restrict the ability to receive the terminal state reward $\mathfrak{R}(v, k)$ to agents with knowledge about the solution. We can accomplish this with the following constraint:

$$y_{sk} \leq \sum_{\tau=0}^{T} F^m_{s\tau} \quad \forall k, s \neq s_0(m). \tag{33}$$

Constraint (33) together with (27) ensures that an agent that collects a reward has been provided with master information.

4) Multiple Master Agents: In some settings the master plan may be available to multiple agents at the initial time step. This would for example be the case if a subset of agents solve Problem 2 deterministically and find the same unique solution, or in a tunnel exploration scenario where planning is done outside the tunnel but the tunnel has multiple openings where agents can enter. It is straightforward to extend Problem 2 and its solution to the case with multiple masters by substituting

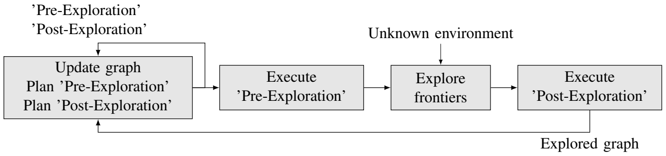


Fig. 3. Block diagram illustrating the use the pre-exploration and post-exploration plans. The final agent positions in the post-exploration problem become the initial positions in the subsequent pre-exploration problem, as indicated by the self-loop in the planning block.

the condition $s_0(r) \neq s_0(m)$ for $s_0(r) \notin \{s_r(0) | r \in m\}$ in constraints (26)-(28) and (33). This corresponds to initializing $|S|$ units of master flow in the initial state of each master agent. Even with this extension only a single master flow is used since the master information is the same irrespective of master agent.

IV. Exploration and Clustering for Scalability

We now present a way to leverage the framework developed in Section III to explore an unknown environment using a team of mobile agents. We study a case where there is a static base station and a heterogeneous group of agents that should explore a communication-constrained tunnel-like environment. Although the flow approach is capable of solving problems of moderate size, large networks are still challenging since solving ILPs is NP-hard. To mitigate this issue we propose a clustering method that separates a large exploration problem into smaller problems that can be solved independently.

A. Exploration via Intermittent Connectivity

The main goal in an exploration scenario is to get agents to *frontiers*, which are states on the boundary between known and unknown space. However, it is also important to share information about the progress between agents and potentially with an external operator. For this reason we require that a static *base station* is periodically updated with the progress.

Since we do not assume continuous connectivity it is necessary to synthesize a plan that gets agents to frontiers, and also a plan that distributes the new information to the base station when frontier exploration is finished. We call these two planning problems *pre-exploration* and *post-exploration*. Fig. 3 shows how the two problems are solved and executed in a repeating manner to explore an unknown environment.

In pre-exploration we use Proposition 2 with a single designated master that computes a global plan. The master could be the base station, or it could be a mobile agent that has sufficient computational capabilities for solving the planning problem. Since the only information that needs to be shared is the plan itself, there are no sources or sinks in this problem instance, i.e., $\texttt{src} = \texttt{snk} = \emptyset$. To promote exploration we assign high rewards to frontiers states. In addition, we include a heuristic reward based on the betweenness-centrality measure of a state [16], which implies that agents get a higher reward by finishing in a state that is more central in the graph. This is useful for efficient data distribution in the subsequent post-exploration step. Executing the resulting plan sends agents to frontier states from which a suitable exploration behavior can be executed.

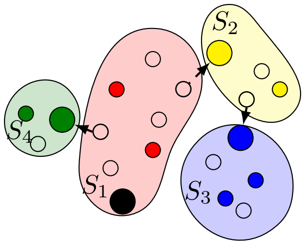


Fig. 4. Illustration of clustering for exploration. States are depicted with circles, and occupied states are filled. The master agent is marked with a black circle. The master cluster S_1 has two child clusters S_2 and S_4, and S_3 is the child of S_2. Each child cluster has a designated submaster agent (shown as a larger circle) whose state is connected to a state in the parent cluster via a communication edge.

For the post-exploration problem we again solve an instance of Problem 2, this time with information sources $\texttt{src}$ set to the agents that performed exploration (and thus have new information), and the information sinks $\texttt{src}$ set to the base station and the master agent. It is possible that agents that were idle during pre-exploration are useful in post-exploration. For this purpose, in the post-exploration problem each agent that received the pre-exploration plan is considered a master agent in the post-exploration instance of Problem 2 (leveraging the multi-master extension discussed in Section III-C4).

When the master is a static base station it is necessary that it can communicate to a mobile agent for a plan to be distributable. To ensure this property we add an additional constraint to the post-exploration problem that requires at least one dynamic agent to be in communication range with the static master. If we denote the set of dynamic agents by R_{dyn} and the set of states connected to the master, possibly via other static relay agents, by ${}^m S_{\text{static}}$, we add the following constraint to the post-exploration problem to guarantee feasible information distribution in the subsequent pre-exploration problem.

$$\sum_{r \in R_{\text{dyn}}} \sum_{s \in {}^m S_{\text{static}}} z_{rsT} \geq 1. \tag{34}$$

B. Clustering for Improved Scalability

We next exploit the inherent structure of the exploration setting to improve scalability. We remark that agents that are far away from each other in the network are unlikely to directly interact, and therefore aim to divide the network into multiple sub-networks, or *clusters*, and constrain the movement of each agent to a single cluster. We exploit the hierarchical structure of the problem to design an information sharing scheme that passes information between clusters.

Definition 7. A *clustering* of a mobility-communication network $\mathcal{N} = (S, \to, \rightsquigarrow)$ with agent positions $\{s_0^r\}$ and master m is a partition of S into connected (via mobility edges) subsets $S_1, \ldots, S_k$ such that the following holds:

- $s_0^m \in S_1$ and we call S_1 the *master cluster*,
- Each cluster $S_2, \ldots, S_k$ has a designated *parent cluster*,
- Each cluster S_i except the master cluster has a *submaster*, which is an agent r s.t. $s_0^r \in S_i$ s.t. there is a communication edge from a state in the parent cluster to s_0^r.

An illustration of the clustering concept in Definition 7 is shown in Fig. 4. To arrive at a clustering that satisfies these requirements we start by grouping agents together via *spectral clustering* [17],[18], which takes a similarity matrix A and desired number of clusters K as inputs and categorizes the agents into clusters by performing K-means clustering based on the spectrum of A. The similarity matrix A is constructed by defining the similarity $A_{i,j}$ of two agents i and j as the inverse of the shortest weighted distance between i and j using only mobility edges and the transition costs as weights.

Once agents have been grouped together into *agent clusters*, the next step is to find corresponding subgraphs, or *state clusters*. Naturally, if agent r belongs to agent cluster k, then $s_0(r)$ should belong to the k:th state cluster. Algorithm 1 starts by assigning these initial states to clusters, and then iteratively expands the state clusters in a way that guarantees that the conditions in Definition 7 are satisfied. It starts with only the master cluster being *active*, and at each iteration finds a free (i.e., not assigned to a cluster) state s that is at the minimal distance to any agent in an active cluster S, and assigns s to S. If s has a communication edge to an agent r in an inactive cluster S', then S' is activated as a child of S and r becomes the submaster of S'. In case the algorithm results in a non-connected state cluster, the corresponding agent cluster is split and the algorithm restarted.

C. Exploration Planning on a Clustering

We next discuss how to plan for exploration as in Fig. 3 by solving intermittent connectivity problems locally in each cluster. For the pre-exploration plan we solve an instance of Problem 2 for each cluster in a reverse breath-first manner: we first solve for clusters with no children (leaf clusters), proceed by solving for clusters that only have leaf clusters as children, and so on, until the master cluster is reached.

Consider the problem instance for a cluster S_k. The submaster of S_k is considered the master agent in the problem instance, and submasters of child clusters of S_k are included as static agents so that information can be transferred to from S_k to its child clusters. The reward for passing information to a submaster is set to the optimal value of the optimization problem in the corresponding child cluster, so that it becomes beneficial to pass the plan to high-value child clusters. If a cluster S_k does not receive plan information from its parent, agents in S_k become inactive and not part of the overall plan.

For the post-exploration plan we require that every cluster that received the pre-exploration plan should send information back to the submaster. Both information from frontiers, as well as information from child clusters, should be passed to the parent cluster. To achieve this we solve an instance of Problem 2 where each agent that received the pre-exploration plan is considered a master, `src` contains all agents that reached frontiers and all submasters of child clusters, and `src` contains the cluster submaster. This forces every agent that may have new information—either from a frontier or from a child cluster—to transmit that information to the cluster submaster. Since this is repeated in every cluster new information from frontiers eventually reaches the global master.

Algorithm 1: Agent clustering to state clustering.

```
Result:
  • clusters: map from id to set of vertices
Input:
  • $\mathcal{N}$: mobility-communication network
  • agent_clusters: map from cluster id to agent cluster
  • activate($\mathcal{N}, s$): id:s of clusters that $s$ can activate
  • closest_free_state($\mathcal{N}$, clusters): state that is at minimal
    distance to any agent in a cluster id$\in$ clusters
active_clusters $=$ {master_cluster};
clusters[id] $= \{s_0(r) : r \in$ agent_clusters[id]$\}$;
free_states $= S \setminus (\bigcup_k$ clusters[k]);
while free_states not empty do
    $s$, id $=$ closest_free_state($\mathcal{N}$, active_clusters)
    clusters[id].add($s$)
    free_states.remove($s$)
    for new_id $\in$ activate($\mathcal{N}, s$) $\setminus$ active_clusters do
        active_clusters.add(new_id);
```

V. Results

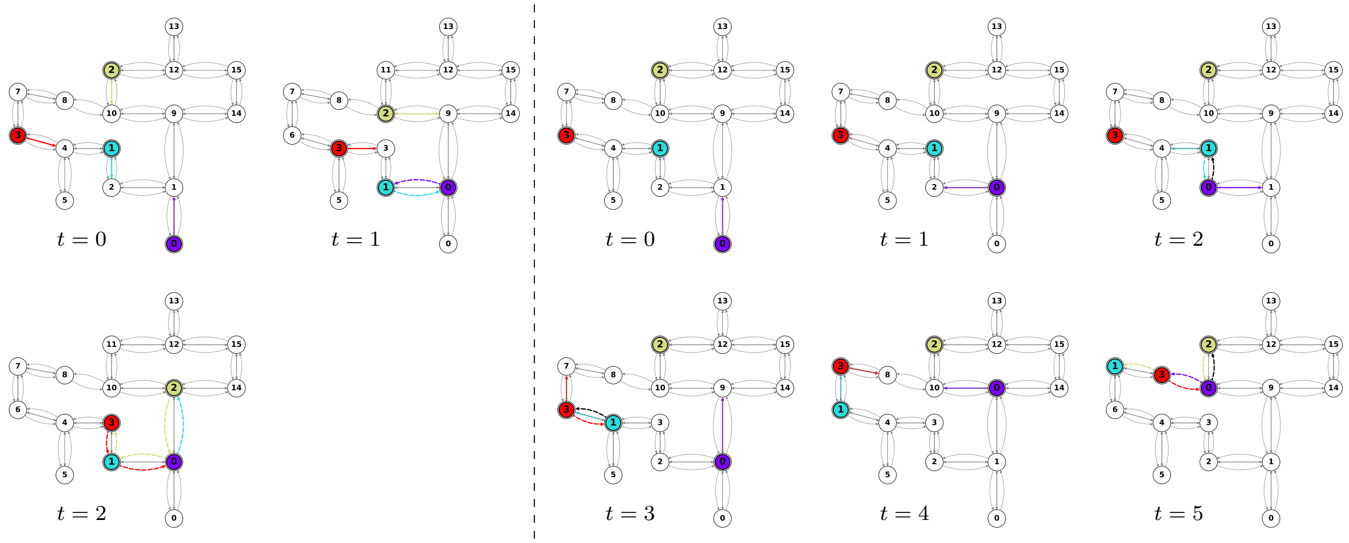
We next show how the results in this paper can be used to plan for exploration. We start with a numerical example that exhibits the difference between intermittent connectivity, and information-consistent intermittent connecivity, and then show numerical results of the improved scalability of our flow-based constraints for information sharing. Finally, we show how the clustering method can be used to plan for exploration in very large environments. All the results in this section have been computed with our Python toolbox COPS[1], using Gurobi [19] as the underlying ILP solver. The computations are performed on a laptop with an Intel Core i7-8850H processor.

A. Information-Consistency in Intermittent Connectivity

We first demonstrate solutions to the intermittent connectivity problems synthesized by solving the ILPs in Proposition 1 and 2. Fig. 5a shows a solution of Problem 1 where all four agents are required to share information with each other, i.e. $\texttt{src} = \texttt{snk} = 0, \ldots, 3$. As can be seen from the figures, all agents are moving and sharing information as expected and are therefore the problem only requires $T = 2$. However, since all four agents move in the first time step this plan is contingent upon all agents knowing the plan at time $t = 0$.

We next solve Problem 2 to demonstrate what information-consistency implies for the same problem. The purple agent 0 is the master agent and again, we require all 4 agents to share information with each other. The resulting plan is illustrated in Fig. 5b. Naturally, the additional information-consistency requirement implies that more time steps are needed to perform the information sharing task. We see that

[1] COPS is available at https://github.com/FilipKlaesson/cops.



(a) (b)

Fig. 5. Simulation of intermittent connectivity with (a) and without (b) master constraints. All agents are required to share information with each other.

agents do not perform any tasks before master information (black communication edges) is received, which fulfills the requirement of information consistency.

B. Scalability

To demonstrate the improved scalability of flow information sharing constraints compared to previous work we compute optimal solutions for a linear graph of length N. In these experiments there are three agents positioned in 0, $\lceil \frac{N}{2} \rceil$ and $N-1$, with the constraint that all agents are required to share information with each other. We compare the performance with two other methods. In [8], a configuration with recurrent connectivity constraint is planned by looking at the powerset 2^S of the vertices in the network and requiring that if an agent is in a set $A \in 2^S$, then the solution has to use at least one communication edge into A. We extended the powerset idea to plan for intermittent connectivity over time by including dynamics and enforce constraints as in [8] on the time-extended graph. The powerset of the time extended graph scales poorly: there is a copy of the original graph for each time step and hence $2^{(T+1)|S|}$ different subsets.

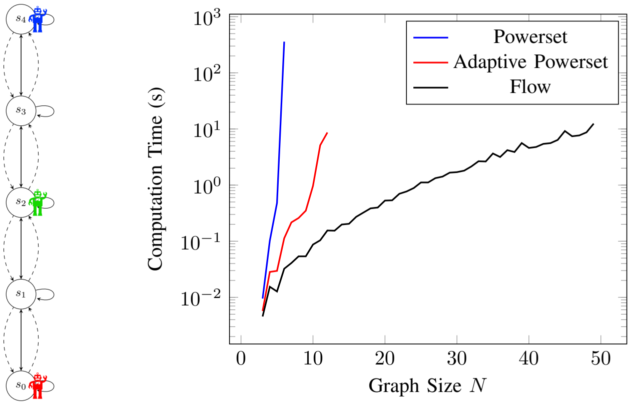


Fig. 6. Computation time of the intermittent connectivity problem as a function of graph size N with agents in state 0, $\left\lceil \frac{N}{2} \right\rceil$ and $N-1$, and all agents are required to share information with each other.

We also consider an adaptive version of the powerset method that first solves the optimization problem without any communication constraints, and, in case a communication constraint is violated, adds the corresponding subset constraint to the problem and solves it again. Thus, constraints corresponding to additional subsets are added until a feasible solution is found or the problem becomes infeasible.

Computational times for solving problems on a linear graph are presented in Fig. 6 for the three methods. The adaptive powerset method outperforms the original powerset method, however both are unable to solve the problem for moderately sized graphs which limits their practical applicability. The flow method scales better and is practical for graphs of moderate size, which is expected since the number of constraints is much lower in the flow formulation: the number scales linearly with the number of edges in the network, instead of combinatorially with graph size. The relationship between the flow approach and the powerset approach is similar to that between the max-flow and min-cut problems. Although the two are dual [15], flows are parameterized by a linear number of variables, whereas there are combinatorially many cuts.

C. Exploration via Clustering

The clustering method presented in Section IV-B allows us to solve much larger problems than what is possible with a single intermittent connectivity problem. In addition, further scalability improvements are possible via heuristics that disregard states that are unlikely to be used (dead states), and evacuating agents in clusters that have no child cluster or frontiers (dead clusters). As an example we consider exploration via the strategy in Fig. 3 of a network with 100 states, using 10 agents including a static base station as master, with initially just one state known. Exploration

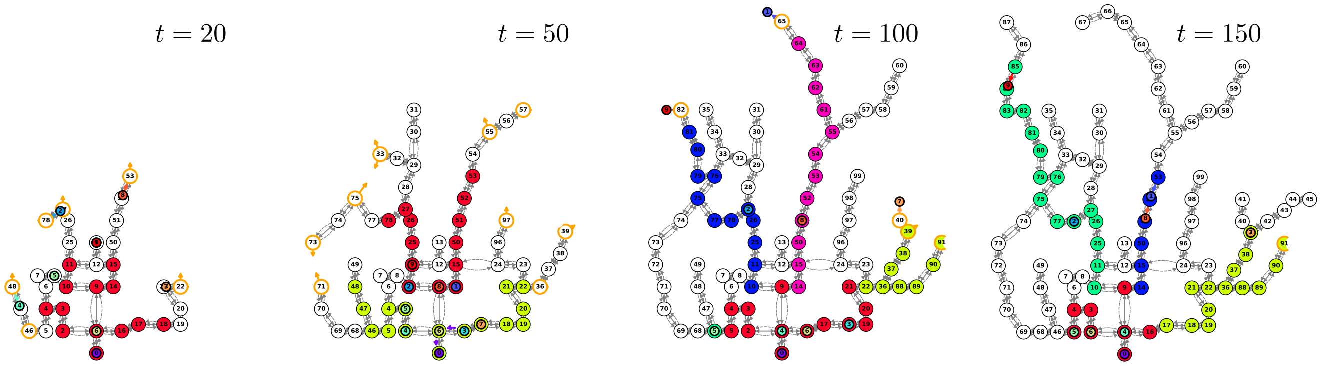


Fig. 7. Simulation of intermittent connectivity constraint in an exploration setting using 10 agents in a network with 100 states. The color of the states indicates the cluster they belong to. Agent 0 in state 0 is a static base station acting as master and is systematically updated about the progress between each exploration cycle.

of the whole environment required computing and executing nine pre-exploration and post-exploration plans, resulting in a total of 48 solved intermittent connectivity problems. The computation time in any cluster takes no more than 1.82 seconds which is negligible in networks of this size where real-world execution of a single plan can take several minutes.

The time evolution of the exploration can be seen in Fig. 7 that shows how the known set of states is expanded[2]. The color of a state indicates the cluster the state belong to. States that are not used in the optimization problem (dead states) are shown as white. Agents are well distributed in the network and simultaneously explore different parts of the network.

VI. Conclusion

In this article we studied intermittent connectivity planning in multi-robot systems. Motivated by information constraints in applications, we proposed information-consistent intermittent connectivity to handle situations where the plan itself is part of the information that should be distributed. As a solution to this problem we presented an Integer Linear Program based on ideas from the literature on network flows, and showed that it scales better than established methods. In addition, we proposed novel "master constraints" that ensure that the resulting plan is information-consistent.

Motivated by utilizing intermittent connectivity for exploration, we presented an exploration strategy based on repeated solutions of information-consistent intermittent connectivity problems. To further improve scalability of this strategy we proposed a clustering method that decomposes a large intermittent connectivity problem for exploration into smaller problems that can be solved independently, and demonstrated in simulation how the resulting algorithm is capable of efficiently exploring a 100-node network using a team of 10 robots.

Current work includes real-world implementation of the exploration algorithm for use in the DARPA Subterranean Challenge by NASA Jet Propulsion Laboratory.

References

[1] M. M. Zavlanos and G. J. Pappas, "Distributed connectivity control of mobile networks," *Proceedings of the IEEE Conference on Decision and Control*, vol. 24, no. 6, pp. 3591–3596, 2007.

[2] M. C. De Gennaro and A. Jadbabaie, "Decentralized Control of Connectivity for Multi-Agent Systems," *Proc. IEEE CDC*, pp. 3628–3633, 2008.

[3] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[4] E. A. Jensen, L. Lowmanstone, and M. Gini, *Communication-Restricted Exploration for Search Teams*. Springer, 2018, pp. 17–30.

[5] G. Hollinger and S. Singh, "Multirobot coordination with periodic connectivity: Theory and experiments," *IEEE. Tran Robotics*, vol. 28, pp. 967–973, 2012.

[6] X. Liu, T. Xi, E. Ngai, and W. Wang, "Path planning for aerial sensor networks with connectivity constraints," in *Proc IEEE ICC*, May 2017, pp. 1–6.

[7] M. N. Rooker and A. Birk, "Multi-robot exploration under the constraints of wireless networking," *Control Engineering Practice*, vol. 15, no. 4, pp. 435 – 445, 2007.

[8] J. Banfi, A. QuattriniLi, I. Rekleitis, F. Amigoni, and N. Basilico, "Strategies for coordinated multirobot exploration with recurrent connectivity constraints," *Autonomous Robots*, vol. 42, no. 4, pp. 875–894, 2018.

[9] Y. Kantaros and M. M. Zavlanos, "Distributed Intermittent Connectivity Control of Mobile Robot Networks," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3109–3121, 2017.

[10] F. Rossi, S. Bandyopadhyay, M. Wolf, and M. Pavone, "Review of Multi-Agent Algorithms for Collective Behavior: a Structural Taxonomy," *IFAC-PapersOnLine*, vol. 51, no. 12, pp. 112–117, 2018.

[11] I. Kara and T. Bektas, "Integer linear programming formulations of multiple salesman problems and its variations," *European Journal of Operational Research*, vol. 174, pp. 1449–1458, 2006.

[12] M. G. Earl and R. D'Andrea, "Modeling and control of a multi-agent system using mixed integer linear programming," *Proc. IEEE CDC*, pp. 107–111, 2002.

[13] Y. E. Sahin, P. Nilsson, and N. Ozay, "Multirobot coordination with counting temporal logics," 2018, arXiv:1810.13087 [cs.RO].

[14] N. Chatzipanagiotis and M. M. Zavlanos, "Distributed scheduling of network connectivity using mobile access point robots," *IEEE Tran. Robotics*, vol. 32, no. 6, pp. 1333–1346, 2016.

[15] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network flows*. Prentice Hall, 1993.

[16] U. Brandes, "A faster algorithm for betweenness centrality," *The Journal of Mathematical Sociology*, vol. 25, no. 2, pp. 163–177, 2001.

[17] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

[18] G. Bonaccorso, *Machine Learning Algorithms*. Packt Publishing, 2017.

[19] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2019. [Online]. Available: http://www.gurobi.com

[2] Full simulation available at https://youtube.com/watch?v=VQjEOFJNRjk.