The 34th Canadian Conference on Artificial Intelligence

Canadian AI 2021
(Vancouver, May 25-28, 2021)

Planning for Connected Agents in a Partially Known Environment

Arthur Queffelec[†], Ocan Sankur[‡], François Schwarzentruber[†]
[†] Univ Rennes, CNRS, IRISA
[‡] Univ Rennes, Inria, CNRS, IRISA

Abstract

The Connected Multi-Agent Path Finding (CMAPF) problem asks for a plan to move a group of agents in a graph while respecting a connectivity constraint. We study a generalization of CMAPF in which the graph is not entirely known in advance, but is discovered by the agents during their mission. We present a framework introducing this notion and study the problem of searching for a strategy to reach a configuration in this setting. We prove the problem to be PSPACE-complete when requiring all agents to be connected at all times, and NEXPTIME-complete in the decentralized case, regardless of whether we consider a bound on the length of the execution.

Keywords: multi-agent path finding, multi-agent planning, connectivity, imperfect knowledge, decentralized planning

1. Introduction

The coordination of mobile agents is at the heart of many real world problems such as traffic control [1], robotics [2, 3], aviation [4] and more [5, 6]. Some of these problems have multiple aspects which make them complex: (1) Some systems are *multi-agent*, that is, the behaviors of agents influence others' and these influences must be taken into consideration when computing missions; this can be due for instance, to collisions [7], sensor interferences [8, 9] etc.; (2) Some missions must ensure *connectivity*, that is, ensure periodic or constant connection to a station/agent to share acquired information [10]; (3) The environment may be only *partially known*, and the agents may discover it during the mission [11, 12]. Several works have considered problems containing these three aspects. For instance, several algorithmic approaches have been investigated to solve the coordination of multi-robot exploration [13–15]. Our objective in this paper is to present a framework to study the theoretical complexity of planning problems with respect to these three aspects.

The theoretical complexity of some related problems have been studied in the literature. Multi-Agent Path Finding (MAPF) is an important framework introduced to study collision-free navigation of agents in warehouses (see [7, 16]). This problem was intensively studied and gave rise to a popular algorithm known as Conflict-Based Search (CBS) [17]. An extension of MAPF with connectivity constraints, called *Connected MAPF* (CMAPF), was studied as well [18]. The complexity of CMAPF and algorithmic solutions were studied in [19, 20]. However, CMAPF only addresses the multi-agent and connectivity aspects, and not the partial knowledge of the environment. The latter aspect is considered in the Canadian Traveler Problem (CTP), which is a well-known problem to study the navigation of an agent in a partially known graph [21]. While the initial framework was for a single agent, CTP has been extended to multiple agents in the settings of packet routing [22], multi-robot exploration [23] and more [24]. While a notion of communication was considered in [25, 26], it is limited to settings where all agents can receive information at all times or only designated agents can send information. In contrast, we are interested in studying the

[†‡] firstname.lastname@irisa.fr

setting where agents' ability to communicate depends on their positions in the graph, and in establishing theoretical complexity results of resulting problems.

In this paper, we study the theoretical complexity of generating plans for a group of agents to reach a given target configuration. More precisely, we analyze the impact of enforcing or ignoring: (A) connectivity; (B) collision; (C) a bound on the length of the execution. For (A), we consider either *fully-connected* strategies, requiring that the agents remain connected at all times during the mission, or a *decentralized* strategy, allowing agents to disconnect and reconnect. (B) In some applications, collisions can be handled by a local collision avoidance system [18], and one can thus abstract away and ignore collisions in graph-based planning algorithms. (C) By providing a bound on the execution length, we can study the complexity of the decision problem associated to the optimization problem. Our results are summarized in Table 1. Interestingly, The PSPACE algorithm for the connected problem with a bounded execution is subtle and relies on a variant of the Savitch's theorem [27] we present here. Additionally, the PSPACE-completeness holds even in the case in which agents can always communicate, thus the hardness of the problem already comes from the incomplete knowledge of the movement graph. For the decentralized case, we prove the NEXPTIME-hardness in the bounded and unbounded cases by two separate reductions from the True Dependency Quantified Boolean Formula problem (TDQBF) [28], thus showing that the problem becomes significantly harder in this case.

Let us compare our results with known complexity results. In the fully known environment, the CMAPF problem is PSPACE-complete in the connected and unbounded case [19], while it is NP-complete in the bounded case (with the bound given in unary) [18]. Thus, the partial knowledge of the environment does not render the problem harder in terms of complexity. In contrast, recall that in MAPF (without connectivity), one can check the existence of a solution in polynomial time [29], while the bounded problem is NP-hard [30], so the hardness is due to connectivity constraints. Some algorithms were presented for CMAPF in [19] that can scale up to about ten agents. Since both problems are similar and belong to PSPACE, one can hope that these approaches can be extended to the partial knowledge case. On the other hand, our results show that the complexity of the decentralized case is significantly higher. While some tools and algorithms are available for Decentralized POMDPs, which are in the same complexity class, the scalability is limited and the development of efficient algorithms for this case seems more challenging (see [31] for a survey).

Some proofs are omitted due to space constraints; they are provided in the long version [32].

	Decentralized	Connected
bounded	NEXP-complete (Th. 7)	PSPACE-complete (Th. 4, 5)
unbounded	NEXP-complete (Th. 6)	PSPACE-complete (Th. 3)

Table 1. Complexity Results.

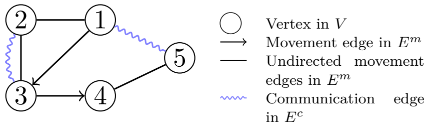

2
1
5
3
4
Vertex in V
Movement edge in E^m
Undirected movement edges in E^m
Communication edge in E^c

Figure 1. Example of a topological graph.

2. Preliminaries

Connected MAPF. We consider a setting where agents move on a graph, with the *vertices/nodes* being their possible locations and the *movement edges* defining how they can move. In addition, *communication edges* define how the agents can communicate. The graphs we consider thus have two types of edges and are called *topological graphs.*

Definition 1. *A* topological graph *is a tuple $G = \langle V, E^m, E^c \rangle$, with V a finite non-empty set of vertices, $E^m \subseteq V \times V$ a set of movement edges and $E^c \subseteq V \times V$ a set of undirected communication edges.*

A movement edge (u,v) is said to be undirected if $(u,v),(v,u) \in E^m$. Figure 1 gives an example of a topological graph. For instance, an agent can go from 1 to 3 in one step. Two agents at vertices 1 and 5 can communicate, but two agents at 1 and 4 cannot.

We suppose that each vertex has a self-loop movement edge and a self-loop communication edge. This respectively represents the ability of an agent to stay at their location and to communicate with a nearby agent (i.e. at the same location/vertex).

Definition 2. *A* configuration *is a tuple $c = \langle c_1, \ldots, c_n \rangle$ where c_i is the vertex of agent i.*

We write $c \to c'$ when $(c_i, c'_i) \in E^m$ for all $1 \leq i \leq n$. This means that each agent i can move from their vertex c_i in c to their vertex c'_i in c' in one step.

Definition 3 (Execution)**.** *An execution π is a sequence of configurations $\langle \pi[0], \pi[1], \ldots, \pi[\ell] \rangle$ such that $\pi[t] \to \pi[t+1]$ for all $t \in \{0, \ldots, \ell - 1\}$.*

We denote the length of π by $|\pi|$, and write $\pi[0..t-1]$ to denote the sub-execution $\langle \pi[0], \pi[1], \ldots, \pi[t-1] \rangle$ of π. Moreover, $\mathsf{last}(\pi)$ denotes the last configuration of π.

We say that a configuration c is *connected* iff the subgraph of the vertices occupied by the agents form a connected graph for relation E^c, i.e. the graph $\langle V_a, E^c \cap (V_a \times V_a) \rangle$ is connected with $V_a = \{c_1, \ldots, c_n\}$. An execution is said to be *connected* iff all its configurations are connected. The Connected Multi-Agent Path Finding problem consists in finding a connected execution for the agents from a given initial configuration to a target configuration. We summarize below the known complexity results for the CMAPF problem.

Theorem 1 ([18])**.** *The problem of deciding whether, in a given instance (G, c^s, c^t, k) where k is unary, there is a* connected *execution from c^s to c^t of length at most k is NP-complete.*

Theorem 2 ([19])**.** *The problem of deciding whether for a given instance (G, c^s, c^t, ∞), there exists a connected execution from c^s to c^t is PSPACE-complete.*

We do not consider collision constraints which would forbid agents from sharing the same vertex. For CMAPF with perfect knowledge, executions are not harder to compute with or without collision constraints [19, 20]. The same holds in our case, and we discuss, in the Section 6, how to incorporate these constraints in our setting.

Dependency Quantified Boolean Formula. A Dependency QBF (DQBF) is a formula in which dependencies of existential variables over universal variables are explicitly specified. A DQBF is of the form $\forall y_1, \ldots, y_n \exists x_1(O_{x_1}) \ldots \exists x_n(O_{x_n})\ \psi$, where each O_{x_i} is, the *dependency set*, a subset of universally quantified variables, and ψ is a Boolean formula in CNF over $x_1, \ldots, x_n, y_1, \ldots, y_n$. It is worth noting that a QBF can be seen as a DQBF with $O_{x_1} \subseteq O_{x_2} \subseteq ... \subseteq\ O_{x_n}$.

The *True DQBF* (TDQBF) is the problem of deciding whether a given DQBF holds. Formally, a DQBF φ holds iff there exists a collection of Skolem functions $A = (A_{x_i} : \{0,1\}^{O_{x_i}} \to \{0,1\})_{i=1..n}$ such that replacing each existential variable x_i by a Boolean formula representing A_{x_i}, turns ψ into a tautology. TDQBF is NEXPTIME-complete [28], and will be used to prove NEXPTIME lower bounds in Section 5.

3. Our framework

3.1. Modeling Imperfect Information

To formalize CMAPF in the imperfect information setting, let us show how to represent the initial knowledge of the agents, and how the information they have evolves during the execution. Agents initially know the exact set of vertices, but only have a lower and an upper approximation of the actual graph: they know that some (communication or movement) edges are *certain* (they must be present), while some are *uncertain* (they may be absent).

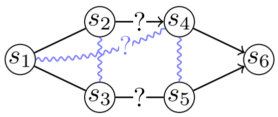


Figure 2. An initial knowledge (G_1, G_2).

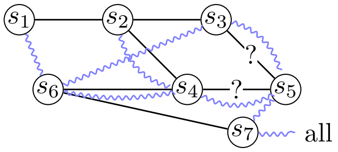
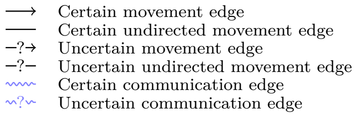


Figure 3. An initial knowledge of a topological graph. Vertex s_7 has communication edges with all other vertices.


Certain movement edge
Certain undirected movement edge
Uncertain movement edge
Uncertain undirected movement edge
Certain communication edge
Uncertain communication edge

Definition 4 (Initial Knowledge)**.** *The initial knowledge is modeled by a pair of topological graphs (G_1, G_2), with the graph $G_1 = \langle V, E_1^m, E_1^c \rangle$ a lower bound, and $G_2 = \langle V, E_2^m, E_2^c \rangle$ an upper bound on the knowledge about the actual graph with $E_1^m \subseteq E_2^m$ and $E_1^c \subseteq E_2^c$.*

The agents initially know G_1 and G_2 while the actual graph $G = \langle V, E^m, E^c \rangle$ is initially unknown to them. They only know that $E_1^m \subseteq E^m \subseteq E_2^m$ and $E_1^c \subseteq E^c \subseteq E_2^c$, written as $G_1 \subseteq G \subseteq G_2$. The perfect information case is captured by $G_1 = G_2 (= G)$.

A movement (resp. communication) edge is said to be *certain* (i.e. sure to be present) if it is in E_1^m (resp. E_1^c); it is said *uncertain* (i.e. can be absent) if it is in $E_2^m \setminus E_1^m$ (resp. $E_2^c \setminus E_1^c$). We assume that the communication edges of the actual graph are undirected, so for all $(u, v) \in E_2^c \setminus E_1^c$, either $(u, v), (v, u) \in E^c$, or $(u, v), (v, u) \notin E^c$.

We say that an edge (u, v) is an *uncertain undirected movement edge* when $(u, v), (v, u) \in E_2^m \setminus E_1^m$. The environment can leave both edges $(u, v), (v, u)$, remove both edges $(u, v), (v, u)$, but also remove (u, v) and leave (v, u), or remove (v, u) and leave (u, v). That is, the movements edges of the actual graph are not necessarily undirected.

Example 1. *Figure 2 depicts an initial knowledge (G_1, G_2). The area is divided in two zones connected by two bridges, represented by the edges (s_2, s_4) and (s_3, s_5), with an uncertainty on which bridge is open and on the communication between s_1 and s_4.*

A strategy σ_i for agent i tells where to go next after a given execution π. Formally:

Definition 5. *A* strategy *σ_i for agent i maps any execution π to a vertex such that $(v, \sigma_i(\pi)) \in E^m$ where v is the vertex at which agent i is in the last configuration of π.*

A *joint strategy* σ is a tuple $\langle \sigma_1, \ldots, \sigma_n \rangle$ where σ_i is a strategy for agent i. The *outcome* of a joint strategy σ starting from configuration c^s is the execution π defined by induction as follows: $\pi[0]$ is c^s, and for $t \geq 1$, $\pi[t]$ is the configuration in which agent i is at vertex $\sigma_i(\pi[0..t-1])$.

In the context of imperfect information, the behaviors of the agents only depend on their observations, as in imperfect information games as in [33]. The strategies, as defined above, do not necessarily take observations of the agents into account. We will now formalize observations and *uniform* strategies, that is, those respecting the observations of the agents.

In our setting, at any time, an agent observes all movement edges adjacent (both in- and out-coming) to the vertex v it occupies. Moreover, they observe the presence or absence of a communication edge between v and v' if v' is occupied by another agent with which there is a direct or indirect communication (via other agents). Intuitively, during an execution, at each step, each agent updates their knowledge about the graph with these observations they receive. Moreover, they share all their knowledge with all agents with which they are connected at each step.

The observation of adjacent movement edges has been a recurrent practice in theoretical works [21, 34] as well as robotics [35, 36], and our formalism is inspired from these works.

The knowledge of an agent at any time corresponds intuitively to a pair of graphs as in Definition 4. For agent i and execution π, let us denote by $k_i(\pi)^G$ the knowledge of agent i

about the graph after observing the execution π in actual graph G. Given such knowledge $K = k_i(\pi)^G$, the agent can deduce an under-approximation and an over-approximation of the actual graph; let us denote these by $\underline{G}^K$ and $\overline{G}^K$ respectively. In particular, if K is the knowledge the agents have initially, then $\underline{G}^K = G_1$ and $\overline{G}^K = G_2$. We present a representation of the knowledge using predicates in the appendix where a detailed formalization of $k_i(\pi)^G$ can be found.

In the rest of the paper, we assume all considered strategies to be *uniform*, that is, they comply with the knowledge of the agents: the strategies prescribe the same move to all executions that are indistinguishable with the agent's observations. Formally, if $|\pi| = |\pi'|$ and $k_i(\pi)^G = k_i(\pi')^G$, then $\sigma_i(\pi) = \sigma_i(\pi')$.

3.2. **Decision Problems**

We consider the decision problem of reaching a configuration c^t from a configuration c^s in less than k steps, using uniform strategies. For two configurations c^s, c^t, let us call a topological graph G (c^s, c^t)-*admissible* if there is an execution from c^s to c^t, which is not necessarily connected.

Definition 6. *We say that an instance (G_1, G_2, c^s, c^t, k) is positive if there exists a joint strategy σ such that in all (c^s, c^t)-admissible graphs G satisfying $G_1 \subseteq G \subseteq G_2$, the outcome of σ starting in c^s ends in c^t in less than k steps.*

Observe that the above problem requires that a strategy ensures the reachability of the target configuration only for graphs that are (c^s, c^t)-admissible and compatible with the initial knowledge. In fact, intuitively, we would like the strategy to work under all possible graphs G with $G_1 \subseteq G \subseteq G_2$. However, requiring a strategy to ensure reachability in a non-admissible graph does not make sense, since even a strategy with full information would fail. We thus require the strategies to make their best efforts, that is, to ensure the objective unless it is physically impossible.

Example 2. *Consider the example of Figure 2. If both bridges (i.e. movement edges (s_2, s_4) and (s_3, s_5)) are absent in the actual graph, the graph is not (s_1, s_6)-admissible and there cannot be a strategy ensuring reachability. The admissible graphs contain either (s_2, s_4), or (s_3, s_5), or possibly both. Note that, this instance is negative for $k < 6$ (that is, it does not admit a solution). Indeed, consider a strategy that moves the agent, for instance, to s_2. In the graph where (s_2, s_4) is absent, the agent would need to come back to s_1 and take the alternative path, which requires an execution of total length 6; and the situation is symmetric if the first move is towards s_3. The instance is nonetheless positive for $k \geq 6$ with the described strategy. However, if the edges (s_2, s_1) and (s_3, s_1) were not present, then the instance would be negative. In fact, once the agent moves to s_2 or s_3, they get stuck if the graph only contains the other bridge.*

We now define the *connected* version of Definition 6. For two configurations c^s, c^t, we say that a topological graph G is (c^s, c^t)-*c-admissible* if there is a connected execution from c^s to c^t. We will often omit the pair of configurations which will be clear from the context, and write simply admissible or c-admissible.

Definition 7. *We say that an instance (G_1, G_2, c^s, c^t, k) is c-positive if there exists a joint strategy σ such that in all (c^s, c^t)-c-admissible graphs G satisfying $G_1 \subseteq G \subseteq G_2$, the outcome of σ starting in c^s is connected and ends in c^t in less than k steps.*

In the connected case, agents cannot visit a disconnected configuration. Hence, the considered strategies only visit configurations that are certainly connected. Observe that agents can make observations about the presence or absence of communication edges while being connected and use this information later.

Example 3. *Let us illustrate the above property on the example of Figure 3. Assume there are two agents, the starting and goal configurations are $c^s = \langle s_1, s_6 \rangle$ and $c^t = \langle s_5, s_7 \rangle$, and the only uncertainty is about the movement edges (s_3, s_5) and (s_4, s_5). Here, Agent 2 could immediately move to her target s_7; however, she could also cooperate with Agent 1 and lower the total completion time. Indeed, from their start configuration $\langle s_1, s_6 \rangle$, the agents first move to $\langle s_2, s_4 \rangle$ where Agent 2 observes whether (s_4, s_5) is present. Assume the edge is present. Then, they follow the sequence $\langle s_4, s_6 \rangle \cdot \langle s_5, s_7 \rangle$; and otherwise $\langle s_3, s_6 \rangle \cdot \langle s_5, s_7 \rangle$. Thus, in order to minimize the length of the execution, the agents do not always take their shortest paths but might help other agents by obtaining information about the graph.*

Consider now the same example in which the communication edge (s_3, s_6) is uncertain. If this edge is absent then Agent 2 cannot help Agent 1 achieve the target faster since if the former moves to s_4 and (s_4, s_5) is absent, then, in order to maintain connectivity, the next configurations should be $\langle s_4, s_6 \rangle \cdot \langle s_2, s_7 \rangle \cdot \langle s_3, s_7 \rangle \cdot \langle s_5, s_7 \rangle$. An execution of the same size is obtained when Agent 2 moves to s_7 in the first step.

For both Definitions 6 and 7 above, let us call a joint strategy a *witness* if it witnesses the fact that the given instance is positive, and respectively, c-positive.

We instantiate the Connected MAPF problem in four different settings. The four following decision problems are defined depending on whether we consider the connectivity requirement and whether the bound is finite. Note that the bounded problems are the decision problems associated to the optimization problems.

Bounded Decentralized Reachability. Is given (G_1, G_2, c^s, c^t, k), with $k < \infty$, positive?

Bounded Connected Reachability. Is given (G_1, G_2, c^s, c^t, k), with $k < \infty$, c-positive?

Unbounded Decentralized Reachability. Is given $(G_1, G_2, c^s, c^t, \infty)$ positive?

Unbounded Connected Reachability. Given $(G_1, G_2, c^s, c^t, \infty)$ c-positive?

As we will see, the encoding of the integer k does not change the overall complexity. Lower bounds are all obtained directly for the unary encoding; the lower bounds for the binary encoding follows. Concerning the upper bounds, we explain for each case how to design an algorithm with k encoded in binary.

4. Connected Reachability

We first address the case where agents must be connected at each step of the execution. In this case, agents share their knowledge at all times and thus the group of agents can be considered as a single agent playing against the environment.

4.1. Unbounded Case

We first focus on the existence of an unbounded connected strategy. Interestingly, we show that verifying the existence of a connected strategy in a partially known environment is not harder than in a perfectly known environment.

Theorem 3. *The unbounded connected reachability problem is PSPACE-complete.*

4.2. Bounded Case

We now study the existence of a bounded connected strategy. We show that this problem is PSPACE-complete even when the communication graph is complete. Please find in supplementary material detailed proofs of the Lemmas 1, 2 and Theorem 5.

Theorem 4. *The bounded connected reachability problem is in PSPACE when the bound is given in binary.*

Proof. Let us first prove the upper bound when k is given in unary. As APTIME = PSPACE [37], we give an alternating algorithm that runs in polynomial time, as follows. At each step, the existential player chooses the next connected configuration to move the agents; and the universal player chooses the information about the newly discovered edges. After k steps the algorithm accepts if the target configuration is reached, or the revealed edges mean that the graph is not c-admissible. The number of steps is bounded by k, which is polynomial, thus the algorithm runs in polynomial time.

There is one subtlety to prove the correctness. The alternating algorithm actually corresponds to a slight variant of our setting which can be seen as a game. In our setting, the environment chooses a graph G with $G_1 \subseteq G \subseteq G_2$ at the beginning, and the agents discover the graph G as they move. In contrast, in the alternating algorithm, the universal player reveals the graph step by step; therefore the environment might adapt the graph to the moves of the existential player.

Lemma 1. *The alternating algorithm is correct.*

When k is binary, the previous algorithm does not run in polynomial time. However, observe that the number of alternations can be bounded by a polynomial because there is only a polynomial number of steps in which the universal players reveal *new* information to the coalition of agents. In fact, the universal player is only useful when some agent is at a vertex that has not been seen before, and this can only happen a linear number of times. Furthermore, the previous algorithm runs in polynomial space.

When k is binary, our problem is in $\mathsf{STA}(\mathsf{poly}(n), *, \mathsf{poly}(n))$ where $\mathsf{STA}(s(n), t(n), a(n))$ is the set of problems decided in space $O(s(n))$, time $O(t(n))$ with $O(a(n)))$ alternations. Our problem is in PSPACE thanks to the generalization of Savitch's theorem we prove:

Lemma 2. $\mathsf{STA}(\mathsf{poly}(n), *, \mathsf{poly}(n)) \subseteq \mathsf{PSPACE}$.

Intuitively, this lemma is proved by guessing the computations between each universal choice by a PSPACE oracle, which yields an overall PSPACE algorithm. □

Theorem 5. *The bounded connected reachability problem with complete connectivity is PSPACE-hard.*

Our reduction actually builds an undirected movement graph. Thus, PSPACE-hardness holds already for undirected movement graphs. Note that in our current setting, pairs of uncertain edges of the form (u, v) and (v, u) are treated separately, but the lower bound proof still holds when they are seen as one.

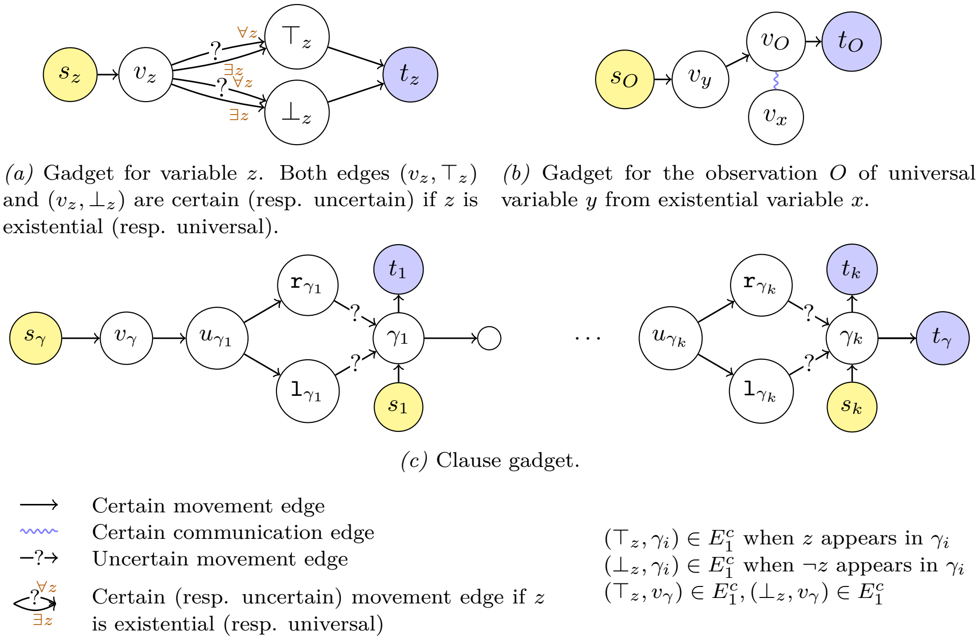
5. Decentralized Reachability

We now tackle the case where agents are allowed to be disconnected; at each configuration, they share their knowledge with all agents to which they are connected. This case is harder because agents no longer follow a centralized strategy and they must cooperate to exchange information at the right moment to reach their targets.

5.1. Unbounded Case

Theorem 6. *The unbounded decentralized reachability problem is NEXPTIME-complete.*

Proof Sketch. For the upper bound, an NEXPTIME algorithm consists in guessing uniform strategies for all agents and checking whether the joint strategy is a witness. Such a strategy has exponential size since it is a function of the sets of knowledge of the agents and the current vertex. One can enumerate all graphs G between G_1 and G_2, and execute the joint strategy on G to check that it ensures the reachability of the target. Moreover, the executions



(a) Gadget for variable z. Both edges $(v_z, \top_z)$ and $(v_z, \bot_z)$ are certain (resp. uncertain) if z is existential (resp. universal).

(b) Gadget for the observation O of universal variable y from existential variable x.

(c) Clause gadget.

Figure 4. Gadgets in the reduction from DQBF to unbounded decentralized reachability.

to be checked have at most exponential length. In fact, executions can be seen as paths in a meta-graph where vertices are configurations augmented with the sets of knowledge of the agents. This meta-graph is of exponential size, so it is sufficient to consider executions of exponential length. The overall non-deterministic algorithm is thus in exponential time.

The lower bound is shown by reduction from TDQBF. Given a DQBF $\varphi = \forall y_1, \ldots, y_n \exists x_1(O_{x_1}) \ldots \exists x_n(O_{x_n})\ \psi$, we build an instance (G_1, G_2, c^s, c^t, k) of unbounded decentralized reachability. We denote by $\gamma_1, \ldots, \gamma_m$ the clauses in ψ.

The graph G_1 and G_2 as follows. For each variable z, we create the gadget depicted in Figure 4a.

We create the observation gadget for all existential variables x and for all (universal) variables $y \in O_x$, depicted in Figure 4b. For convenience, we write O for the pair (x, y) corresponding to observation of y by x.

Finally, we create the clause gadget, depicted in Figure 4c. A vertex γ_i certainly communicates with $\top_z$ iff $z \in \gamma_i$, and with $\bot_z$ iff $\neg z \in \gamma_i$. Moreover, the vertex v_γ communicates with all $\top_z$ and $\bot_z$ for all variables z.

We define the initial and target configurations as $c^s = \langle s_\gamma, s_{\gamma_1}, \ldots, s_{\gamma_m}, s_{x_1}, \ldots, s_{x_n}, s_{y_1}, \ldots, s_{y_n}, s_{O_1}, \ldots, s_{O_k} \rangle$, and $c^t = \langle t_\gamma, t_{\gamma_1}, \ldots, t_{\gamma_m}, t_{x_1}, \ldots, t_{x_n}, t_{y_1}, \ldots, t_{y_n}, t_{O_1}, \ldots, t_{O_k} \rangle$.

We give some intuitions about the reduction; the full proof is in the supplementary material. Assuming the DQBF φ holds, one can build a witness strategy as follows. The agents starting at a universal s_z are forced to follow a path not deleted by the environment, which determines the value of z. The agents starting at s_O can observe the valuation of their respective universal variables (by visiting v_y), and then inform their respective existential variables thanks to the communication edge between v_O and v_x. Thus, agents starting at an existential s_z are aware of the values of all variables in O_z and choose an appropriate value so as to satisfy φ. The agents starting at s_i idle at γ_i at step 3. At this point, the agent that starts s_γ also idles at v_γ, and all agents that start s_z are at $\top_z$ or $\bot_z$. Thanks to the communication edges, and because φ is true, each γ_i communicates with at least some $\top_z$

or $\perp_z$, thus also with the agent at v_γ. Therefore, the latter agent knows about all available edges in the clause gadget and can successfully go to t_γ.

□

5.2. Bounded Case

In the bounded case, the problem is NEXPTIME-complete independently of the encoding of the bound. Moreover, the hardness holds even for undirected graphs.

Theorem 7. *The bounded decentralized reachability problem is NEXPTIME-complete. NEXPTIME-hardness holds for undirected graphs.*

6. Discussion

Additional Results. We present results obtained by a simple observation/modification.

Unbounded Reachability and Undirected Graphs. Both the unbounded connected and unbounded decentralized reachability become *trivial* on undirected graphs. This is because we only require reachability for (c-)admissible graphs. In the decentralized case, each agent can run a DFS independently, and eventually reach their targets in at most $2|V|$ steps, and a similar search can be done by the set of agents in the connected case.

Base Station. Several works consider a designated *base* vertex to which all agents must stay connected during the execution [19, 20, 38]. This concept is only relevant in the connected case. Our results also hold with this additional constraint. In fact, the lower bound of Theorem 3 follows from [19], which proves the bound also with a base. In Theorem 5, we can add the base vertex as an isolated vertex so that the reduction is still valid.

Collisions. We did not require the paths to be collision-free in the results presented in this paper. However, this property is already ensured by our proofs or can be obtained by simple modifications. The lower bound proof of Theorem 3 relies on Theorem 2 from [19] which holds with collision constraints as well, so this is also true for our case. The proof of Theorem 5 does not generate collision-free paths as the groups of occurrence agents start and finish at the same location and follow almost the same path. This proof can be adapted to prevent collisions by delaying each occurrence agent by 3 steps behind one another. This can be achieved easily by extending the movement paths and shifting the starting location and target location of an agent up by 3 vertices behind the previous agent. The proof of lower bound of Theorem 6 features a construction ensuring a collision-free strategy. Indeed, the observations agents only need to take turns to visit the universal variables. Thus, the result holds with collision constraints as well. The algorithms of Theorem 3, 5, and 7 can be adapted by restricting all considered configurations to collision-free ones; while c-admissibility of a graph with collision constraints can be checked using Theorem 2.

Graph Classes. The MAPF and CMAPF problems have been studied for different classes of graphs (planar, grid, ...). The proof of lower bound in Theorem 3 relies on the proof of unbounded reachability done in [20], thus the result of PSPACE-hardness on planar graphs also carries over to our problem. Planar QBF is known to be PSPACE-complete [39], and the construction of Theorem 5 is such that when applied to a planar QBF, the resulting graph is planar. Our PSPACE-completeness result thus holds on planar graphs.

Related Work. Different definitions of robust plans [40–42] have been studied. A *k-robust plan* guarantees the reachability of the target in the events of at most k delays. A *p-robust plan* executes without a conflict with probability at least p. Our framework does not consider delayed agents but focus on synchronous executions with imperfect knowledge of the area.

The problem of MAPF with a dynamic environment has multiple formulations. The Adversarial Cooperative Path-Finding [43] considers that the obstacles are agents which reason to prevent the cooperation to reach its goal. [44] considered the problem where the dynamics of the environment is predictable. Additionally, when obstacles have unknown dynamics, one can estimate their movements and plan to minimize the probability of a collision [45], or predict their movements [46] and plan online the movement of the agents [47]. In our setting, the environment is static, thus, all observations are fixed.

MAPF with Uncertainty (MAPFU) asks for a plan which guarantees that mishaps, localization and sensing errors do not impact the proper execution of the plan. This problem can be solved by temporal logic [48], POMDPs [49], replanning [50–52], interaction regions [1, 53], and belief space planning [54–56]. Nebel et al. [57] studied the MAPF problem with an uncertainty on the destination of the agents and lack of communication. The asynchronous movement of the agents, studied in those papers, cannot be expressed in our framework as we require the agent to follow some universal clock to execute their plan.

Perspectives. We proposed a setting for CMAPF in the imperfect case and studied the theoretical complexity of the reachability problem. The first natural question is to find classes of graphs (e.g. grid graphs) on which the reachability problem is easier to solve, as it was done for MAPF in [58, 59], and CMAPF in [20]. Another possible direction is to study the coverage of all vertices [20]. An alternative way to handle non-admissible graphs is to require that agents return to their starting configuration if the graph is discovered not to be admissible. We believe that such variants should be as hard as reachability. Furthermore, there are several possible generalizations that could be considered by introducing dynamic environments (instead of static), faulty sensing of agents, robustness, uncertainty, etc.

References

[1] K. Dresner and P. Stone. “A Multiagent Approach to Autonomous Intersection Management”. In: *JAIR* 31.1 (2008), pp. 591656. DOI: 10.1613/jair.2502.

[2] E. Erdem, D. G. Kisa, U. Oztok, and P. Schüller. “A General Formal Framework for Pathfinding Problems with Multiple Agents”. In: *Proc. of AAAI*. 2013, pp. 290296.

[3] J. Yu and S. LaValle. “Planning Optimal Paths for Multiple Robots on Graphs”. In: 2012, pp. 3612–3617. DOI: 10.1109/ICRA.2013.6631084.

[4] L. Pallottino, V. G. Scordio, A. Bicchi, and E. Frazzoli. “Decentralized Cooperative Policy for Conflict Resolution in Multivehicle Systems”. In: *IEEE Trans, on Rob.* 23.6 (2007), pp. 11701183. DOI: 10.1109/TRO.2007.909810.

[5] D. Silver. “Cooperative Pathfinding”. In: *Proc. of AIIDE*. 2005, pp. 117122.

[6] J. Yu and S. M. LaValle. “Optimal Multirobot Path Planning on Graphs: Complete Algorithms and Effective Heuristics”. In: *IEEE Trans. on Rob.* 32.5 (2016), pp. 11631177. DOI: 10.1109/TRO.2016.2593448.

[7] H. Ma and S. Koenig. “AI Buzzwords Explained: Multi-Agent Path Finding (MAPF)”. In: *AI Matters* 3 (2017). DOI: 10.1145/3137574.3137579.

[8] D. Goldberg and M. J. Matarić. “Interference as a Tool for Designing and Evaluating Multi-Robot Controllers”. In: *Proc. of AAAI*. 1997, pp. 637–642.

[9] M. Schneider-Fontán and M. Mataric. “Territorial multi-robot task division”. In: *IEEE Trans. on Rob. and Autom.* 14 (1998), pp. 815–822.

[10] F. Amigoni, J. Banfi, and N. Basilico. “Multirobot Exploration of Communication-Restricted Environments: A Survey”. In: *IEEE Intelli. Sys.* 32.6 (2017), pp. 48–57. DOI: 10.1109/MIS.2017.4531226.

[11] N. Rao, S. Kareti, W. Shi, and S. Iyengar. “Robot navigation in unknown terrains: Introductory survey of non-heuristic algorithms”. In: 1993.

[12] S. Thrun. “Robotic Mapping: A Survey”. In: *Exploring Artificial Intelligence in the New Millennium*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003, pp. 135.

[13] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider. “Coordinated multi-robot exploration”. In: *IEEE Transactions on Robotics* 21.3 (2005), pp. 376–386. DOI: 10.1109/TRO.2004.839232.

[14] K. M. Wurm, C. Stachniss, and W. Burgard. “Coordinated multi-robot exploration using a segmentation of the environment”. In: *IROS*. IEEE. 2008, pp. 1160–1165.

[15] L. Matignon, L. Jeanpierre, and A.-I. Mouaddib. “Coordinated multi-robot exploration under communication constraints using decentralized markov decision processes”. In: *Proc. of AAAI*. Vol. 26. 1. 2012.

[16] A. Felner, R. Stern, S. E. Shimony, E. Boyarski, M. Goldenberg, G. Sharon, N. Sturtevant, G. Wagner, and P. Surynek. “Search-based optimal solvers for the multi-agent pathfinding problem: Summary and challenges”. In: *SoCS*. 2017, pp. 28–37.

[17] G. Sharon, R. Stern, A. Felner, and N. Sturtevant. “Conflict-based search for optimal multi-agent pathfinding”. In: *Artificial Intelligence* 219 (Feb. 2015), pp. 40–66. DOI: 10.1016/j.artint.2014.11.006.

[18] G. A. Hollinger and S. Singh. “Multirobot Coordination With Periodic Connectivity: Theory and Experiments”. In: *IEEE Trans. on Rob.* 28.4 (2012), pp. 967–973. DOI: 10.1109/TRO.2012.2190178.

[19] D. Tateo, J. Banfi, A. Riva, F. Amigoni, and A. Bonarini. “Multiagent Connected Path Planning: PSPACE-Completeness and How to Deal With It”. In: *Thirty-Second Conference on Artificial Intelligence,* 2018.

[20] T. Charrier, A. Queffelec, O. Sankur, and F. Schwarzentruber. “Complexity of planning for connected agents”. In: *JAAMAS* 34.2 (2020), p. 44. DOI: 10.1007/s10458-020-09468-5.

[21] C. H. Papadimitriou and M. Yannakakis. “Shortest paths without a map”. In: *Theoretical Computer Science* 84.1 (1991), pp. 127–150. DOI: 10.1016/0304-3975(91)90263-2.

[22] A. Itai and H. Shachnai. “Adaptive source routing in high-speed networks”. In: *ISTCS*. 1993, pp. 212–221. DOI: 10.1109/ISTCS.1993.253468.

[23] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun. “Collaborative multi-robot exploration”. In: *Proc. of ICRA*. Vol. 1. 2000, pp. 476–481. DOI: 10.1109/ROBOT.2000.844100.

[24] L. V. Lita, J. Schulte, and S. Thrun. “A System for Multi-Agent Coordination in Uncertain Environments”. In: *Proc. of AGENTS*. 2001, pp. 21–22. DOI: 10.1145/375735.375806.

[25] H. Zhang and Y. Xu. “The k-Canadian Travelers Problem with Communication”. In: *FAW-AAIM*. 2011, pp. 17–28. DOI: 10.1007/s10878-012-9503-x.

[26] D. Shiri and F. S. Salman. “On the Online Multi-Agent O—D k-Canadian Traveler Problem”. In: *J. of Comb. Opt.* 34.2 (2017), pp. 453–461. DOI: 10.1007/s10878-016-0079-8.

[27] W. J. Savitch. “Relationships Between Nondeterministic and Deterministic Tape Complexities”. In: *J. Comput. Syst. Sci.* (1970). DOI: 10.1016/S0022-0000(70)80006-X.

[28] G. Peterson, J. Reif, and S. Azhar. “Lower Bounds for Multiplayer Noncooperative Games of Incomplete Information”. In: *Comput & Math. Appl.* 41.7-8 (2001), pp. 957–992. DOI: 10.1016/S0898-1221(00)00333-3.

[29] J. Yu and S. M. LaValle. “Multi-agent Path Planning and Network Flow”. In: *Algorithmic Foundations of Robotics X*. 2013, pp. 157–173.

[30] P. Surynek. “An Optimization Variant of Multi-Robot Path Planning is Intractable”. In: *Proc. of AAAI*. 2010, pp. 1261–1263.

[31] C. Amato, G. Chowdhary, A. Geramifard, N. K. Üre, and M. J. Kochenderfer. “Decentralized Control of Partially Observable Markov Decision Processes”. In: *CDC*. 2013, pp. 2398–2405. DOI: 10.1109/CDC.2013.6760239.

[32] A. Queffelec, O. Sankur, and F. Schwarzentruber. “Planning for Connected Agents in a Partially Known Environment”. In: *The 34th Canadian Conference on Artificial Intelligence*. Vancouver, Canada, May 2021. URL: https://hal.archives-ouvertes.fr/hal-03205744.

[33] R. Berthon, B. Maubert, A. Murano, S. Rubin, and M. Y. Vardi. “Strategy logic with imperfect information”. In: *LICS*. 2017, pp. 1–12. DOI: 10.1109/LICS.2017.8005136. eprint: 1805.12592.

[34] A. Bar-Noy and B. Schieber. “The Canadian Traveller Problem.” In: *SODA*. Vol. 91. 1991, pp. 261–270.

[35] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. “Robotic exploration as graph construction”. In: *IEEE Transactions on Robotics and Automation* 7.6 (1991), pp. 859–865. DOI: 10.1109/70.105395.

[36] S. Koenig, C. Tovey, and W. Halliburton. “Greedy Mapping of Terrain”. In: *Proc. of ICRA*. Vol. 4. 2001, 3594–3599 vol.4. DOI: 10.1109/ROBOT.2001.933175.

[37] A. K. Chandra, D. Kozen, and L. J. Stockmeyer. “Alternation”. In: *J. of ACM* 28.1 (1981), pp. 114–133. DOI: 10.1145/322234.322243.

[38] T. Charrier, A. Queffelec, O. Sankur, and F. Schwarzentruber. “Reachability and Coverage Planning for Connected Agents”. In: *Proc. of IJCAI*. 2019, pp. 144–150. DOI: 10.24963/ijcai.2019/21.

[39] D. Lichtenstein. “Planar Formulae and Their Uses”. In: *SIAM Journal on Computing (SICOMP)* 11 (1982), pp. 329–343. DOI: 10.1137/0211025.

[40] H. Ma, T. K. S. Kumar, and S. Koenig. “Multi-Agent Path Finding with Delay Probabilities”. In: *Proc. of AAAI*. 2017, pp. 3605–3612.

[41] D. Atzmon, A. Felner, R. Stern, G. Wagner, R. Barták, and N.-F. Zhou. “k-Robust Multi-Agent Path Finding”. In: *International Symposium on Combinatorial Search (SoCS)*. 2017, pp. 157–158.

[42] D. Atzmon, R. Stern, A. Felner, N. R. Sturtevant, and S. Koenig. “Probabilistic Robust Multi-Agent Path Finding”. In: *Proc. of ICAPS*. 2020, pp. 29–37.

[43] M. Ivanová and P. Surynek. “Adversarial Cooperative Path-Finding: Complexity and Algorithms”. In: *ICTAI*. 2014, pp. 75–82. DOI: 10.1109/ICTAI.2014.22.

[44] A. Murano, G. Perelli, and S. Rubin. “Multi-agent Path Planning in Known Dynamic Environments”. In: *PRIMA*. 2015. DOI: 10.1007/978-3-319-25524-8_14.

[45] J. Miura and Y. Shirai. “Probabilistic Uncertainty Modeling of Obstacle Motion for Robot Motion Planning”. In: *Journal of Robotics and Mechatronics* 14 (2002), pp. 349–356.

[46] N. C. Griswold and J. Eem. “Control for mobile robots in the presence of moving objects”. In: *IEEE Trans. on Rob. and Autom.* 6.2 (1990), pp. 263–268. DOI: 10.1109/70.54744.

[47] Yun Seok Nam, Bum Hee Lee, and Nak Yong Ko. “A View-Time Based Potential Field Method for Moving Obstacle Avoidance”. In: *Proc. of SICE*. 1995, pp. 1463–1468. DOI: 10.1109/SICE.1995.526730.

[48] A. Ulusoy, S. L. Smith, X. C. Ding, and C. Belta. “Robust multi-robot optimal path planning with temporal logic constraints”. In: *Proc. of ICRA*. 2012, pp. 4693–4698. DOI: 10.1109/ICRA.2012.6224792.

[49] S. A. Miller, Z. A. Harris, and E. K. P. Chong. “Coordinated Guidance of Autonomous UAVs via Nominal Belief-State Optimization”. In: *ACC*. 2009, pp. 2811–2818. DOI: 10.1109/ACC.2009.5159963.

[50] A. Stentz. “Optimal and Efficient Path Planning for Unknown and Dynamic Environments”. In: *IJRA* 10 (Feb. 1993).

[51] D. Ferguson, N. Kalra, and A. Stentz. “Replanning with RRTs”. In: *Proc. of ICRA*. 2006, pp. 1243–1248. DOI: 10.1109/ROBOT.2006.1641879.

[52] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun. “Anytime Dynamic A*: An Anytime, Replanning Algorithm”. In: *Proc. of ICAPS*. 2005, pp. 262–271.

[53] C. Ferrari, E. Pagello, J. Ota, and T. Arai. “Multirobot motion coordination in space and time”. In: *Robotics and Autonomous Systems* 25.3-4 (1998), pp. 219–229. DOI: 10.1016/S0921-8890(98)00051-7.

[54] A. Bry and N. Roy. “Rapidly-exploring Random Belief Trees for motion planning under uncertainty”. In: *Proc. of ICRA* (2011), pp. 723–730.

[55] J. P. Gonzalez and A. Stentz. “Planning with uncertainty in position an optimal and efficient planner”. In: *IROS*. 2005, pp. 2435–2442. DOI: 10.1109/IROS.2005.1545048.

[56] S. Prentice and N. Roy. “The Belief Roadmap: Efficient Planning in Belief Space by Factoring the Covariance”. In: *IJRR* 28 (Oct. 2009), pp. 1448–1465. DOI: 10.1177/0278364909341659.

[57] B. Nebel, T. Bolander, T. Engesser, and R. Mattmüller. “Implicitly Coordinated Multi-Agent Path Finding under Destination Uncertainty: Success Guarantees and Computational Complexity”. In: *JAIR* 64.1 (2019), pp. 497527. ISSN: 1076-9757. DOI: 10.1613/jair.1.11376.

[58] K.-H. C. Wang and A. Botea. “Tractable Multi-Agent Path Planning on Grid Maps”. In: *Proc. of IJCAI*. 2009, pp. 1870–1875.

[59] J. Banfi, N. Basilico, and F. Amigoni. “Intractability of Time-Optimal Multirobot Path Planning on 2D Grid Graphs with Holes”. In: *RA-L* 2.4 (2017), pp. 1941–1947. DOI: 10.1109/LRA.2017.2715406.