Reachability and Coverage Planning for Connected Agents

Tristan Charrier[1] , **Arthur Queffelec**[1] , **Ocan Sankur**[2] and **François Schwarzentruber**[1]
Univ Rennes, CNRS, IRISA[1]
Univ Rennes, Inria, CNRS, IRISA[2]
firstname.lastname@irisa.fr

Abstract

Motivated by the increasing appeal of robots in information-gathering missions, we study multi-agent path planning problems in which the agents must remain interconnected. We model an area by a topological graph specifying the movement and the connectivity constraints of the agents. We study the theoretical complexity of the reachability and the coverage problems of a fleet of connected agents on various classes of topological graphs. We establish the complexity of these problems on known classes, and introduce a new class called *sight-moveable graphs* which admit efficient algorithms.

1 Introduction

A number of use cases of planning rose in information-gathering missions from the development of unmanned autonomous vehicles (UAVs). For instance, in search and rescue missions, a fleet of drones can cover a lot of ground in a short amount of time and report any finding to a mission supervisor to narrow the search for the rescue team. Other examples are the terrain analysis for smart farms and for areas in hazardous locations. For this kind of missions, the information gathered is used for decision making at a supervising station. Thus, the robots need to be constantly in communication with the station to report the information gathered during the mission. The use of multiple UAVs to cover an area not only reduces the time required to complete the mission but also enables reaching locations which may not be reachable with a single drone due to connection constraints.

The original multi-agent path finding problem asks for a plan to reach a configuration of agents in a graph [Ratner and Warmuth, 1986]. However, an important problem for search and rescue missions or terrain analysis is the *coverage* of an area. We thus study both the reachability and the coverage problems under a connection constraint over the agents which requires them to be connected to the base either directly or via another agent, who can relay its data. We establish the computational complexity of the connected coverage in its general case and for a practical subclass introduced recently [Tateo *et al.*, 2018] in which the UAVs can communicate with others located within one step, called the *neighbor-communicable* topological graphs. We show that the coverage is PSPACE-complete in the general case, and remains so for neighbor-communicable topological graphs. Thus, restricting to neighbor-communicable graphs does not make the problem feasible, and the relatively high complexity unfortunately remains. Note that this is in line with the PSPACE-completeness of the reachability problem recently reported in [Tateo *et al.*, 2018].

Our main result in this paper is the definition of a class of topological graphs which is well adapted and realistic for UAV missions, and for which the coverage and reachability problems admit efficient algorithms. Our subclass, called *sight-moveable* graphs, is defined assuming that the UAVs cannot communicate through obstacles and are restricted to line-of-sight communication. This class emerged from an ongoing case study for a drone assisted search and rescue project in which the authors take part[1]. For this class, we prove that both the reachability and coverage problems are in LOGSPACE. This drastically changes the status of this problem since by LOGSPACE $\subseteq$ NC (this is the class of problems solvable in polylogarithmic time in a parallel machine with a polynomial number of processors), one can build an efficient parallel algorithm [Cook, 1979]. The bounded versions are NP-complete. This means efficient SAT solvers can be used directly to compute bounded executions.

In this work, we consider anonymous agents. Furthermore, we consider the collisions to be handled by the agents themselves, hence are not considered along the results of this paper. We depicted a covering execution of a topological graph by 3 UAVs in Figure 1. In this example, the UAVs need to gather information at each node of the graph while staying connected to the base (red node) during the whole mission.

In Section 2, we present the typical notions used in Multi-Agent Path Planning (MAPP) and their extension for our case and the known results in connected planning. In Sections 3 to 6, we study the complexity of our problems from the general case to the most restrictive one. We describe the related work in Section 7. We conclude in Section 8.

This paper is the follow-up to the extended abstract presented at the 18th International Conference on Autonomous Agents and Multi Agent Systems (AAMAS 2019) [Charrier *et al.*, 2019]. An extended version[2] is also available.

[1] EIT Retina project
[2] https://arxiv.org/abs/1903.04300

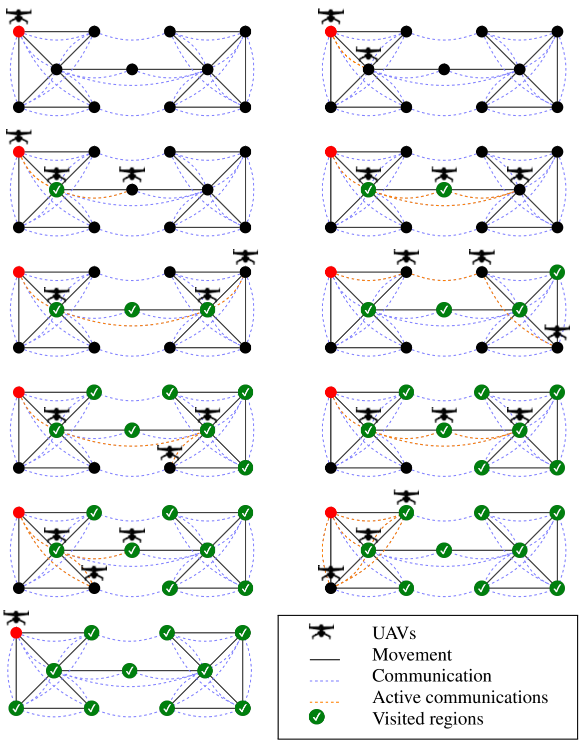
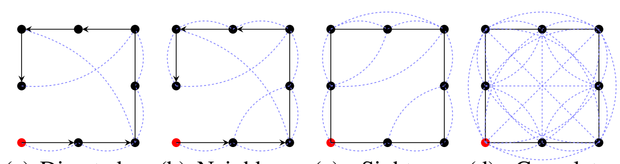


Figure 1: Example of a mission execution.



(a) Directed Graph (b) Neighbor-communicable (c) Sight-moveable (d) Complete-communication

Figure 2: Examples of topological graphs.

2 Preliminaries

In most applications of path planning, the space is discretized in order to generate a graph of movements on which algorithms are executed. For instance, *regular grids* which decompose the space in square, triangular or hexagonal cells, *irregular grids* with techniques such as *quadtrees* [Finkel and Bentley, 1974; Knoll, 2006] or *Voronoï diagrams* comprehensively discussed in the survey [Aurenhammer, 1991].

Our work is independent of the particular method used to obtain the discretization. We only work under the hypothesis that a feasible plan on the graph generated by the discretization is also feasible in the continuous space.

2.1 Topological Graph

Compared to the graphs used in MAPP, we also consider *communication edges* which specify whether agents at two different locations can communicate. We call graphs with this additional information *topological graphs*. The formal definition is the following.

Definition 1 (Topological graph). *A topological graph is a tuple $G = \langle V, \rightarrow, \cdots \rangle$, with V a finite set of nodes containing a distinguished element B, $\rightarrow \subseteq V \times V$ a set of movement edges and $\cdots \subseteq V \times V$ a set of undirected communication edges.*

The node B symbolizes the supervision base station from which the agents start the mission. A topological graph is undirected if $\langle V, \rightarrow \rangle$ is an undirected graph.

We will now consider three subclasses of interest.

In most situations, if an agent can move to a location in one step, it can also communicate with an agent at that location. This class has been discussed in [Tateo *et al.*, 2018]. We call topological graphs satisfying this requirement *neighbor-communicable*. An example is given in Figure 2b.

Definition 2 (Neighbor-Communicable topological graph). *A neighbor-communicable topological graph is a topological graph such that $v \rightarrow v'$ implies $v \cdots v'$.*

Another class of graphs is that of *sight-moveable* and is the main one for which we give efficient algorithms. First, this class requires the movement edges to be undirected and reflexive. Second, whenever an agent can communicate with another node, then it can also move to that node while maintaining the communication. This intuitively means that the communication is restricted to line-of-sight and is disallowed through obstacles. The formal definition follows, and an example is depicted in Figure 2c.

Definition 3 (Sight-Moveable topological graph). *A sight-moveable topological graph is an undirected neighbor-communicable topological graph in which for all $v \in V$, $v \rightarrow v$ and whenever $v \cdots v'$, there exists a sequence $\rho = \langle \rho_1, \ldots, \rho_n \rangle$ of nodes such that $v = \rho_1$, $v' = \rho_n$, $v \cdots \rho_i$ and $\rho_i \rightarrow \rho_{i+1}$ for all $i \in \{1, \ldots, n\}$.*

Last, we define the *complete-communication* topological graphs which are simply sight-moveable topological graphs with a complete communication topology. This subclass can model that the communication is not perturbed in the area. An example of such a graph is depicted in Figure 2d, and the formal definition is the following.

Definition 4 (Complete-Communication topological graph). *A complete-communication topological graph is a sight-moveable topological graph such that $\cdots = V \times V$.*

Observe that complete-communication graphs are reflexive, undirected, connected graphs with $\cdots = V \times V$.

2.2 Execution

An *execution*, in MAPP, is a finite sequence of *configurations* describing the placement of the agents during the mission. The formal definition of a configuration is the following.

Definition 5 (Configuration). *A configuration c of n agents in a topological graph G is an element of V^n denoted $c = \langle c_1, \ldots, c_n \rangle$ in which c_i is the location of agent i such that the graph $\langle V_a, \cdots \cap (V_a \times V_a) \rangle$ is connected with $V_a = \{B, c_1, \ldots, c_n\}$. We extend the notation $\rightarrow$ and denote $c \rightarrow c'$ when $c_i \rightarrow c'_i$ for all $1 \leq i \leq n$.*

MAPP asks to associate an agent to a specific goal. However, given that we are interested in covering an area with a fleet of agents, the anonymity is useful to get more efficient plans.

Anonymity. In the rest of this paper, agents are *anonymous*. In other words, a configuration c is equivalent to a configuration c' iff c is a permutation of c'.

Moreover, an important notion in MAPP is the computation of collision-free plans. In the drone case, in which we are particularly interested, one can place drones at different heights to avoid collisions. Additionally, most drones are, nowadays, equipped with local collision avoidance systems.

Collisions. We do not deal with meet- or head-on-collisions of agents, *i.e.* we allow two agents to be located in a same node, and to move in opposite directions of an edge within a step.

An *execution* e of length ℓ with n agents in a graph G is a sequence of configuration $\langle c^1, \ldots, c^\ell \rangle$ such that for $c^i \rightarrow c^{i+1}$ for all $1 \leq i < \ell$.

A *covering execution* $e = \langle c^1, \ldots c^\ell \rangle$ of length ℓ with n agents in a graph G is an execution such that $c^1 = c^\ell = \langle B, \ldots, B \rangle$ and for all $v \in V$, there exists $i \in \{1, \ldots, \ell\}$ with v appearing in c^i.

2.3 Decision Problems

We define the MAPP problems, the *Reachability* problem along with its bounded version, *bReachability*, for the makespan optimization of the plan. In addition, we define the *Coverage* problem and the bounded coverage, *bCoverage*.

Definition 6 (Reachability)**.** *Given a topological graph G, $n \in \mathbb{N}$ written in unary and a configuration c of size n, decide if there is an execution $\langle c^1, \ldots, c^\ell \rangle$ in G such that $c^1 = \langle B, \ldots, B \rangle$ and $c^\ell = c$.*

Definition 7 (bReachability)**.** *Given a topological graph G, $n \in \mathbb{N}$ written in unary and a configuration c of size n and $\ell \in \mathbb{N}$ written in unary, decide if there is an execution $\langle c^1, \ldots, c^{\ell'} \rangle$ in G s.t. $\ell' \leq \ell$ and $c^{\ell'} = c$.*

Definition 8 (Coverage)**.** *Given a topological graph G and $n \in \mathbb{N}$ written in unary, decide if there exists a covering execution with n agents.*

Definition 9 (bCoverage)**.** *Given a topological graph G, $n, \ell \in \mathbb{N}$ written in unary, decide if there exists a covering execution of length ℓ' such $\ell' \leq \ell$.*

We study the restrictions of the above problems to classes of topological graphs. We denote $P_{\mathcal{C}}$ the problem P (one of the four above problems) restricted to a class $\mathcal{C}$ of topological graphs ($\mathcal{C}$ can either be dir for directed, nc for neighbor-communicable, und for undirected, sm for sight-moveable or cc for complete-communication topological graphs).

2.4 Known Results

The complexity of the decision problem associated to the minimization of the makespan with non-anonymous agents and collision, is known to be NP-hard [Ratner and Warmuth, 1986]. Throughout the study of MAPP, NP-hardness was shown to hold on planar graphs [Yu, 2016] and, later, on 2D grid graphs [Banfi *et al.*, 2017]. Variants of MAPP have been studied such as the package-exchange robot-routing problem [Ma *et al.*, 2016] where the robots are anonymous but not the package they exchange, is shown to be NP-hard. A class of grid graphs was shown to be solvable in polynomial time [Wang and Botea, 2009].

The connected version of MAPP was introduced in [Hollinger and Singh, 2012], in which a topological graph discretizes the space and it is proved that the existence of a plan for the reachability of a configuration of agents in a bounded amount of steps is NP-hard:

Theorem 10. *$bReachability$ restricted to undirected topological graphs is NP-hard [Hollinger and Singh, 2012].*

In [Tateo *et al.*, 2018], it is shown that deciding the existence of a feasible plan is PSPACE-complete:

Theorem 11. *$Reachability$ restricted to undirected topological graphs is PSPACE-complete [Tateo* et al.*, 2018].*

Authors prove this result for graphs with self-loops and a base [Tateo *et al.*, 2018] as in our setting (see Discussion following Theorem 1). The only difference with our setting is that the agents start at a specific configuration in [Tateo *et al.*, 2018]. Nevertheless, it can be shown that their complexity result holds for our problem by a simple but subtle construction given in the extended version.

In the rest of the paper, we study the upper bounds and the lower bounds complexity of the defined decision problems on the previously defined topological graphs. The following sections present our results, respectively, for the general case, the neighbor-communicable graphs, sight-moveable graphs, and complete-communication graphs.

3 Directed Topological Graphs

For the bounded versions, we can guess and check a path of bounded length in polynomial-time since the input is encoded in unary:

Proposition 12. *$bCoverage_{dir}$ and $bReachability_{dir}$ are in NP.*

For the unbounded problems, we can design a straightforward NPSPACE algorithm that guesses an execution by keeping in memory the last configuration, and, for $Coverage_{dir}$, the set of visited regions. We conclude with Savitch's Theorem (NPSPACE=PSPACE)[Savitch, 1970]:

Theorem 13. *$Coverage_{dir}$ and $Reachability_{dir}$ are PSPACE-complete.*

The lower bound of $Reachability_{dir}$ is given in Theorem 11. We now concentrate on $Coverage_{dir}$.

Lemma 14. *$Coverage_{dir}$ is PSPACE-hard.*

Proof. The proof is by reduction from $Reachability_{dir}$ in which the base node has a self-loop. As noted in the remark following Theorem 11, this problem remains PSPACE-hard. We map an instance (G, c) of $Reachability_{dir}$ to the instance G' of $Coverage_{dir}$ where G' is depicted in Fig. 3. Let k denote the number of agents in the instance (G, c). G' contains G as a subgraph, plus fresh nodes $v_1, \ldots, v_k$ and $s_1, \ldots, s_k$. An agent can move from any node of G to v_1 and back.

Node s_1 can communicate with the base B, and node v_k can communicate with all nodes of G'. Furthermore, we have the communication edges (s_i, s_{i+1}) and (v_i, v_{i+1}) for all $1 \leq$

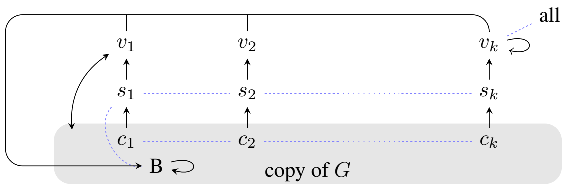


Figure 3: Topological graph G' constructed from the $Reachability_{dir}$-instance.

$i \leq k-1$. Now we prove that the k agents can progress to the configuration $(c_1, \ldots, c_k)$ in G if and only if there exists a covering execution in G'.

($\Rightarrow$) If the agents are in the configuration $(c_1, \ldots, c_k)$ then they can progress in one step to configuration $(s_1, \ldots, s_k)$. Then, they have no choice but progress to the configuration $(v_1, \ldots, v_k)$. Once in this configuration, the agent placed on the node v_k communicates with the base and with all other agents. This agent stays at v_k. Meanwhile the agent placed on the node v_1 will visit all unvisited nodes of G and come back to v_1 while keeping communication to the base through the agent placed on v_k. Meanwhile, agents placed on $v_2, \ldots, v_{k-1}$ come back to B. Finally, when all the nodes have been visited, both agents on v_1 and v_k come back to B.

($\Leftarrow$) If there exists a covering execution of the whole graph G', it means all nodes have been visited. In particular, node s_k has been visited and let us consider the first time t_{s_k} when s_k is visited. Time $t_{s_k} - 1$ denotes the time just before t_{s_k}.

Fact 15. At time $t_{s_k} - 1$, no node v_i and no node s_i were visited.

Proof. Suppose by contradiction that a node v_i was visited by some agent before t_{s_k}, then the only possibility such an agent to communicate to the base is that there is also an agent at v_k at time t_{s_k}. But then, it means that s_k was visited strictly before t_{s_k}, leading to a contradiction. Thus, no node v_i were visited at time t_{s_k} (thus at time $t_{s_k} - 1$).

As no node v_i are visited before t_{s_k}, no node s_i are visited before $t_{s_k} - 1$. □

Fact 16. At time $t_{s_k} - 1$, the configuration is $\langle c_1, \ldots, c_k \rangle$.

Proof. At time t_{s_k}, as the agent at s_k needs to communicate with the base, the only possibility is that the configuration is $\langle s_1, \ldots, s_k \rangle$. Thus, the only possibility is that configuration is $\langle c_1, \ldots, c_k \rangle$. □

Facts 15 implies that the prefix from time 0 to time $t_{s_k} - 1$ of the covering execution is an execution in G. Fact 16 implies that sub-execution reaches $\langle c_1, \ldots, c_k \rangle$. □

4 Neighbor-Communicable Topological Graphs

In this subsection, we show that our problems remain hard for neighbor-communicable graphs.

Theorem 17. *$Coverage_{nc}$ is PSPACE-complete.*

Proof. The upper bound is given by Theorem 13.

For the lower bound on $Coverage_{nc}$, the reduction given in Figure 3 is not adapted for neighbor communicable graphs. Indeed, all nodes may be visited although $c_1, \ldots, c_k$ is not reached: v_1 and v_k can be reached by two lines of agents connected to the base, making the coverage of the full graph possible. We nevertheless give a similar reduction by adapting the previous reduction.

The details are given in the extended version. □

5 Sight-Moveable Topological Graphs

In this subsection, we show that $Reachability_{sm}$ and $Coverage_{sm}$ are in LOGSPACE while the bounded version $bReachability_{sm}$ is NP-complete.

5.1 Upper Bounds

The results of this subsection rely on the problem of checking the connectivity of two nodes s and t in an undirected graph, namely USTCONN.

Theorem 18. *USTCONN is in LOGSPACE[Reingold, 2008].*

Proposition 19. *$Reachability_{sm}$ is in LOGSPACE.*

Proof. The idea of the proof is to reduce $Reachability_{sm}$ to UCONN, that is the problem of deciding whether an undirected graph is connected. From Theorem 18, we can reduce UCONN to USTCONN by simply looping over all pairs of nodes (s,t) and checking for a path from s to t. Therefore, UCONN is in LOGSPACE.

Now we describe the logarithmic space reduction of $Reachability_{sm}$ to UCONN. Let $G = \langle V, \rightarrow, \dashrightarrow \rangle$ a sight-moveable topological graph and c a configuration. Let $V' = \{c_1, \ldots, c_n, B\}$. The configuration c is reachable iff the restriction of $\mathfrak{G}' := (V, \dashrightarrow)$ to the nodes in V' is $\dashrightarrow$-connected. Indeed, if it is, then c is reachable: each agent follows some $\rightarrow$-path from B to c_i contained in a $\dashrightarrow$-path from B to c_i. In other words, (G, c) is a positive $Reachability_{sm}$-instance iff $\mathfrak{G}'$ is a positive UCONN-instance. The reduction is in logarithmic space: we compute $\mathfrak{G}'$ by enumerating all (u, v) $\dashrightarrow$-edges in G, and we output (u, v) when $u, v \in V'$. We recall that we only take into account the working memory for computing $\mathfrak{G}'$; the output – $\mathfrak{G}'$ itself – is not taken into account in the used space (see e.g. [Sipser, 1997], Ch. 8, Def. 8.21). □

Proposition 20. *$Coverage_{sm}$ is in LOGSPACE.*

Proof. First we prove that the bounded version of the connectivity in undirected graphs is also in LOGSPACE.

Lemma 21. *Bounded-USTCONN, that is the problem, giving an undirected graph $\mathfrak{G}$, two nodes s, t, an integer n written in binary, of deciding whether there is a path of length at most n from s to t in G is in LOGSPACE.*

Proof. We reduce Bounded-USTCONN to USTCONN in logarithmic space as follows. From a Bounded-USTCONN instance $(\mathfrak{G}, s, t, n)$ we construct in logarithmic space a USTCONN instance $(\mathfrak{G}', s', t')$: 1. The nodes of $\mathfrak{G}'$

are pairs (v, j) where v is a node of $\mathfrak{G}$ and j is an integer in $\{0, n\}$ but smaller than the number of nodes in $\mathfrak{G}'$; 2. $\mathfrak{G}'$ contains an edge between (v, j) and $(v', j+1)$ when there is an edge between v and v' in $\mathfrak{G}$ or when $v = v'$; 3. $s' = (s, 0)$ and $t' = (t', n)$. □

Now let $G = \langle V, \rightarrow, \cdots\rangle$ be a sight-moveable topological graph and n an integer written in binary. There is a path from any node v to the base B with at most n communication edges iff (G, n) is a positive instance of $Coverage$. Thus, we test sequentially, for all v, that $((V, \cdots), v, B, n)$ is a positive instance of Bounded-USTCONN . Hence, we obtain an algorithm in logarithmic space to decide $Coverage$.

□

5.2 Lower Bounds

We now focus on the NP lower bound of $bReachability_{sm}$.

Proposition 22. *$bReachability_{sm}$ is NP-hard for a fixed execution length $\ell \geq 3$.*

Proof. The proof is by polynomial time reduction from 3-SAT problem (see [Karp, 1972]). Given a 3-SAT instance, set of clauses $c_1, \ldots, c_m$ with variables $x_1, \ldots, x_n$, we describe the construction of an instance (G, c) of $bReachability_{sm}$ with $k = n + m$ agents.

The topological graph $G = \langle V, \rightarrow, \cdots\rangle$ is constructed as follows. We start by placing the base B from which the agents start their mission.

Please recall that a sight-moveable graph is also a neighbor-communicable graph so all movements edges are also communication edges in the construction below even if not explicitly stated.

For each variable x, we construct a gadget composed of 5 nodes connected to the base depicted in Figure 5a: nodes x, $\neg x$, staging nodes n_x, $n_{\neg x}$ and a *goal* node g_x. We add movement edges from B to n_x, from n_x to x and from x to g_x (resp. from B to $n_{\neg x}$, from $n_{\neg x}$ to $\neg x$ and from $\neg x$ to g_x). As for the communication, the node x (res. $\neg x$) communicates with the base.

For each clause c, we construct a gadget composed of 3 nodes depicted in Figure 5b. We create a node c, a staging node n_{c} and a goal node g_{c}. We add movement edges from B to n_c, from n_{c} to c and from c to g_{c}. The communication between a clause c and a literal x or $\neg x$ is dictated by the existence of the literal in the clause: $\mathsf{c}_i \cdots x_j$ if and only if $x_j \in \mathsf{c}_i$; and $\mathsf{c}_i \cdots \neg x_j$ if and only if $\neg x_j \in \mathsf{c}_i$.

We add movement edges from g_{x_i} to $g_{x_{i+1}}$, and from g_{c_i} to $g_{\mathsf{c}_{i+1}}$ for all $1 \leq i < n$, as well as we from g_{x_n} to g_{c_1}. Last, we add a fully connected path containing 3 fresh nodes from g_{x_1} to the base such that $g_{x_1} \cdots B$, in the sense that all nodes of this path have communication edges between them. This translation is polynomial in the number of clauses and variables. The construction is depicted in Figure 4. The snake-like path from g_{x_1} to B is the fully connected path.

From a 3-SAT instance, one can construct the graph G and ask for an execution of length 3 to reach the configuration $\langle g_{x_1}, \ldots, g_{x_n}, g_{\mathsf{c}_1}, \ldots, g_{\mathsf{c}_m}\rangle$.

The rest of the proof is given in the extended version. □

From Propositions 12 and 22, we have:

Theorem 23. *$bReachability_{sm}$ is NP-complete.*

6 Complete-Communication Topological Graphs

The following result relies on the fact that the communication is complete.

Proposition 24. *$bReachability_{cc}$ is in LOGSPACE.*

Proof. From Lemma 21, one can construct an algorithm in LOGSPACE for $bReachability_{cc}$. Indeed, given a configuration c and $\ell \in \mathbb{N}$, the straightforward iteration on the locations c_i followed by the verification of a path of at most ℓ (given in unary) steps from B to c_i yields a sound and complete algorithm for $bReachability_{cc}$. □

Our NP lower bound proof of the $bCoverage_{cc}$ problem is by reduction from the grid Hamiltonian cycle (G-HC) problem which is the Hamiltonian cycle problem restricted to grid graphs and is NP-complete [Itai *et al.*, 1982].

Theorem 25. *$bCoverage_{cc}$ is NP-complete.*

The upper bound follows from Proposition 12. The NP-hardness proof is given in the extended version.

7 Related Work

The coverage planning is an interesting approach to path planning. Indeed, a covering plan can be used for fields such as floor cleaning, lawn mowing, etc. A survey of this field appears in [Choset, 2001]. This multi-agent extension has the ability to reduce the length of the overall mission and also reach parts of the area a single agent would not able to. This problem was studied in [Rekleitis *et al.*, 1997] for two agents. As shown in the survey by Chen et al. [Chen *et al.*, 2014], many coverage problems have been addressed by using analytic techniques. For instance, in [Yanmaz, 2012] and [Teacy *et al.*, 2010], they consider UAVs that should cover an area while staying connected to the base, but only empirically study some path planning algorithms without proving formally their soundness and completeness.

We advocate formal methods that give formal guarantees and have already been applied to generate plans for robots and UAVs. Model checking has been applied to robot planning (see [Lacerda *et al.*, 2014]) and to UAVs [Webster *et al.*, 2011]. Humphrey [Humphrey, 2013] shows how to use LTL (linear-temporal logic) model checking for capturing response and fairness properties in cooperation (for instance, if a task is requested then it is eventually performed).

Bodin et al. [Bodin *et al.*, 2018] treat a similar problem except that the UAVs cover the graph without returning to the base. Without the return-to-the-base constraint, we claim that all our hardness results still hold, except for $bCoverage_{cc}$. They provide an implementation by describing the problem in Planning Domain Description Language and then run the planner Functional Strips [Francès *et al.*, 2017].

Murano et al. [Murano *et al.*, 2015] advocate for a graph-theoretic representations of states, that is, by assigning locations to agents as in Definition 5. In [Aminof *et al.*, 2016;

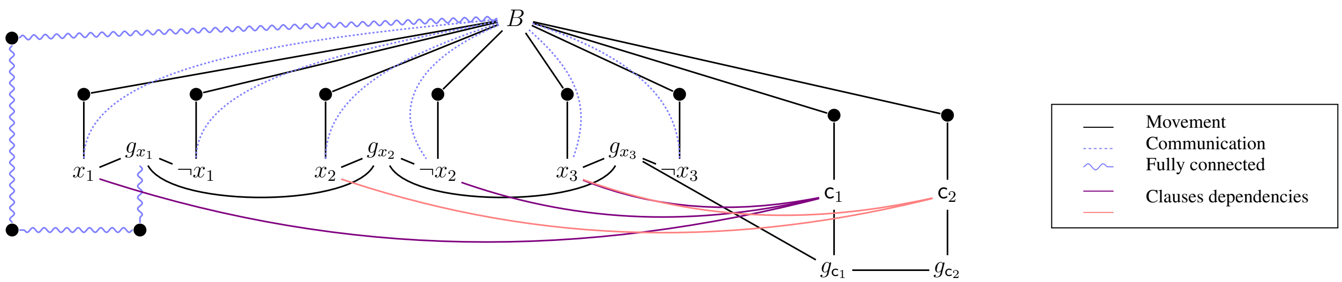
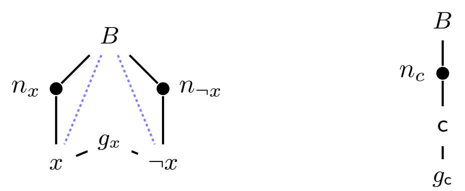


Figure 4: Sight-Moveable topological graph computed from the formula $(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_3)$.



(a) Variable gadget.

(b) Clause gadget.

Figure 5: Gadgets in proof of Proposition 22.

<table>
<tr><th></th><th>Reachability</th><th>Coverage</th><th>bReachability</th><th>bCoverage</th></tr>
<tr><td>dir</td><td rowspan="3">PSPACE-c
[Tateo et al.]</td><td>PSPACE-c</td><td rowspan="3">NP-c
[Hollinger]</td><td rowspan="5">NP-c</td></tr>
<tr><td>nc</td><td>PSPACE-c</td></tr>
<tr><td>und</td><td>Open</td></tr>
<tr><td>sm</td><td rowspan="2">in L</td><td rowspan="2">in L</td><td>NP-c</td></tr>
<tr><td>cm</td><td>in L</td></tr>
</table>

Figure 6: Complexity results (results in the paper are in gray).

Rubin, 2015], a general formalism is given to specify LTL and monadic second-order logic properties, which are expressive enough to describe the connectivity constraint. They provide an algorithm for parametrized verification in the sense that they check a temporal property in a class of graphs. This is relevant for partially-known environments. The algorithm described is non-elementary (*i.e.* the running time cannot bounded by any tower of exponentials) and therefore not usable in practice. We believe that this is an important problem and our paper identifies an efficient and relevant fragment.

The multiple traveling salesman problem (mTSP) is a generalization of the traveling salesman problem (TSP) in which multiple salesmen are located at a depot [Anbuudayasankar *et al.*, 2016]. mTSP asks for the coverage of all cities so as to minimize the total plan cost by visiting each city exactly once. An overview of TSP and its extensions are presented [Matai *et al.*, 2010]. The $Coverage$ problem is related to mTSP, since we use results on Hamiltonian cycle to prove the NP-hardness of $bCoverage_{cc}$. However, we wish to minimize the length of the execution and not the cost of the execution. Those problems are equivalent on unit graphs, but it is not trivial to use general results on mTSP in order to solve $Coverage$. Furthermore, to the best of our knowledge, connected versions of mTSP and VRP have not been studied.

8 Conclusion

Sight-moveable topological graphs we introduced only constrain the communication graph. One can be interested to constrain the movement graph to be planar or a 2D grid given the common usage of grid modelling of the environment. Given the intractability of MAPP on planar graphs [Yu, 2016] and on general 2D grid graphs [Banfi *et al.*, 2017], it is likely that this problem is intractable as well. Furthermore, in [Tateo *et al.*, 2018], the decision is proved to stay PSPACE-complete on planar graphs and grids as well. However, one can study this problem on solid grid graphs, given that the Hamiltonian cycle is tractable on such graphs [Umans and Lenhart, 1997].

Note that our NP lower bounds hold without the anonymity of the agents. Indeed, the $bCoverage$ case is straightforward and for $bReachability$ case, each agent can be associated to a clause or variable, so the reduction would still hold.

We do not know if $Coverage$ remains PSPACE-hard when the $\rightarrow$-relation is symmetric (see Figure 6). We think this open issue is important since symmetric $\rightarrow$-relations (if UAVs can go from v to v', they can also come back from v' to v) are relevant for practical applications. We plan to study the *parametrized complexity* [Downey and Fellows, 1999] of our problems - parameters could be for instance the treewidth of the topological graph or the number of UAVs.

Acknowledgements

This work was partially supported by UAV Retina Funded by EIT Digital. Special thanks to François Bodin for initiating the idea of this work. We thank Eva Soulier for the provided work during her internship.

References

[Aminof *et al.*, 2016] B. Aminof, A. Murano, S. Rubin, and F. Zuleger. Automatic verification of multi-agent systems in parameterised grid-environments. In *International Conference on Autonomous Agents & Multiagent Systems*, 2016.

[Anbuudayasankar *et al.*, 2016] S. P. Anbuudayasankar, K. Ganesh, and S. Mohapatra. *Models for practical routing problems in logistics*. Springer, 2016.

[Aurenhammer, 1991] F. Aurenhammer. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, September 1991.

[Banfi *et al.*, 2017] J. Banfi, N. Basilico, and F. Amigoni. Intractability of time-optimal multirobot path planning on 2d grid graphs with holes. *IEEE Robotics and Automation Letters*, Oct 2017.

[Bodin *et al.*, 2018] F. Bodin, T. Charrier, A. Queffelec, and F. Schwarzentruber. Generating plans for cooperative connected uavs (demo). In *27th IJCAI and the 23rd ECAI*, 2018.

[Charrier *et al.*, 2019] T. Charrier, A. Queffelec, O. Sankur, and F. Schwarzentruber. Reachability and coverage planning for connected agents. In *Proceedings of AAMAS, Montreal, QC, Canada*, pages 1874–1876, 2019.

[Chen *et al.*, 2014] Y. Chen, H. Zhang, and M. Xu. The coverage problem in uav network: A survey. In *Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, July 2014.

[Choset, 2001] H. Choset. Coverage for robotics – a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, Oct 2001.

[Cook, 1979] S. A. Cook. Deterministic cfl's are accepted simultaneously in polynomial time and log squared space. In *Eleventh Annual ACM Symposium on Theory of Computing*, STOC '79. ACM, 1979.

[Downey and Fellows, 1999] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999.

[Finkel and Bentley, 1974] R. A. Finkel and J. L. Bentley. Quad trees a data structure for retrieval on composite keys. *Acta Inf.*, March 1974.

[Francès *et al.*, 2017] G. Francès, M. Ramírez, N. Lipovetzky, and H. Geffner. Purely declarative action descriptions are overrated: Classical planning with simulators. In *Twenty-Sixth International Joint Conference on Artificial Intelligence,*, 2017.

[Hollinger and Singh, 2012] G. A. Hollinger and S. Singh. Multirobot coordination with periodic connectivity: Theory and experiments. *IEEE Transactions on Robotics*, Aug 2012.

[Humphrey, 2013] L. R. Humphrey. *Model Checking for Verification in UAV Cooperative Control Applications*. Springer Berlin Heidelberg, 2013.

[Itai *et al.*, 1982] A. Itai, C. Papadimitriou, and J. Szwarcfiter. Hamilton paths in grid graphs. *SIAM Journal on Computing*, 1982.

[Karp, 1972] R. M. Karp. Reducibility among combinatorial problems. In *Proceedings of a symposium on the Complexity of Computer Computations,*, 1972.

[Knoll, 2006] Aaron Knoll. A survey of octree volume rendering methods. In *GI, the Gesellschaft für Informatik*, page 87, 2006.

[Lacerda *et al.*, 2014] B. Lacerda, D. Parker, and N. Hawes. Optimal and dynamic planning for markov decision processes with co-safe LTL specifications. In *IROS,*, 2014.

[Ma *et al.*, 2016] H. Ma, C. Tovey, G. Sharon, T. K. S. Kumar, and S. Koenig. Multi-agent path finding with payload transfers and the package-exchange robot-routing problem. In *Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16. AAAI Press, 2016.

[Matai *et al.*, 2010] R. Matai, S. Singh, and M. L. Mittal. Traveling salesman problem: an overview of applications, formulations, and solution approaches. In *Traveling Salesman Problem, Theory and Applications*. InTech, 2010.

[Murano *et al.*, 2015] A. Murano, G. Perelli, and S. Rubin. Multi-agent path planning in known dynamic environments. In *PRIMA 2015: Principles and Practice of Multi-Agent Systems - 18th International Conference,*, 2015.

[Ratner and Warmuth, 1986] D. Ratner and M. Warmuth. Finding a shortest solution for the nxn extension of the 15-puzzle is intractable. In *Fifth AAAI National Conference on Artificial Intelligence*, AAAI'86. AAAI Press, 1986.

[Reingold, 2008] O. Reingold. Undirected connectivity in log-space. *J. ACM*, September 2008.

[Rekleitis *et al.*, 1997] I. M. Rekleitis, G. Dudek, and E. E. Milios. Multi-robot exploration of an unknown environment, efficiently reducing the odometry error. In *IJCAI*. Morgan Kaufmann Publishers Inc., 1997.

[Rubin, 2015] S. Rubin. Parameterised verification of autonomous mobile-agents in static but unknown environments. In *AAMAS*, 2015.

[Savitch, 1970] Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *J. Comput. Syst. Sci.*, 1970.

[Sipser, 1997] M. Sipser. *Introduction to the theory of computation*. PWS Publishing Company, 1997.

[Tateo *et al.*, 2018] 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 AAAI Conference on Artificial Intelligence,*, 2018.

[Teacy *et al.*, 2010] W. T. L. Teacy, J. Nie, S. McClean, and G. Parr. Maintaining connectivity in uav swarm sensing. In *GLOBECOM Workshops, 2010 IEEE*. IEEE, 2010.

[Umans and Lenhart, 1997] C. Umans and W. Lenhart. Hamiltonian cycles in solid grid graphs. In *FOCS*, 1997.

[Wang and Botea, 2009] K. C. Wang and A. Botea. Tractable multi-agent path planning on grid maps. In *IJCAI*. Morgan Kaufmann Publishers Inc., 2009.

[Webster *et al.*, 2011] M. Webster, M. Fisher, N. Cameron, and M. Jump. Formal methods for the certification of autonomous unmanned aircraft systems. *Computer Safety, Reliability, and Security*, 2011.

[Yanmaz, 2012] E. Yanmaz. Connectivity versus area coverage in unmanned aerial vehicle networks. In *Proceedings of IEEE International Conference on Communications, ICC 2012,*, 2012.

[Yu, 2016] J. Yu. Intractability of optimal multirobot path planning on planar graphs. *IEEE Robotics and Automation Letters*, Jan 2016.