Autonomous Agents and Multi-Agent Systems (2020) 34:44
https://doi.org/10.1007/s10458-020-09468-5

Complexity of planning for connected agents

Tristan Charrier[1] · **Arthur Queffelec**[1] · **Ocan Sankur**[1] · **François Schwarzentruber**[1]

Published online: 16 June 2020

Abstract
We study a variant of the multi-agent path finding (MAPF) problem in which the group of agents are required to stay connected with a supervising base station throughout the execution. In addition, we consider the problem of covering an area with the same connectivity constraint. We show that both problems are PSPACE-complete on directed and undirected topological graphs while checking the existence of a bounded plan is NP-complete when the bound is given in unary (and PSPACE-hard when the encoding is in binary). Moreover, we identify a realistic class of topological graphs on which the decision problem falls in NLOGSPACE although the bounded versions remain NP-complete for unary encoding.

Keywords Artifical intelligence · Multi-agent systems · Planning · Computational complexity

Mathematics Subject Classification 68T20 · 03D15

1 Introduction

The *multi-agent path finding* (MAPF) problem asks for a plan to move a group of agents to a target configuration in a graph while avoiding collisions. It is an important problem in the design of groups of autonomous vehicles, and has been used in several applications such as Kiva (Amazon Robotics) warehouse systems [58], autonomous aircraft towing vehicles [36], characters in video games [49] and office robots [57].

A closely related problem is that of *coverage path planning* which consists in computing a plan that visits a set of given locations in a graph. A comprehensive survey is given

✉ Arthur Queffelec
Arthur.Queffelec@irisa.fr

Tristan Charrier
Tristan.Charrier@irisa.fr

Ocan Sankur
Ocan.Sankur@irisa.fr

François Schwarzentruber
Francois.Schwarzentruber@irisa.fr

[1] Univ Rennes, Inria, CNRS, IRISA, 263 Avenue Général Leclerc, 35000 Rennes, France

in [23]. Applications include underwater ship hull inspection [18], wildfire tracking with drones [40], to name a few.

An important application area is that of information gathering missions in which agents must visit a set of locations in an area and gather information using sensors (e.g. camera, smoke sensor, hygrometer etc.). Some applications, such as search and rescue missions, might require continuous connection between all agents and a base station, for instance, in order to stream video and allow human operators to take decisions [2]. In this case, the path planning and coverage path planning algorithms must take additional connectivity constraints into account in order to compute suitable plans. Several works have explored planning algorithms in this setting e.g. [39, 44].

Some works on path planning and coverage path planning either assume a given discrete graph model obtained by cell decomposition or visibility graph [34], or applies a sampling method to construct a graph on which combinatorial algorithms are applied (see also the Sect. 2). Graph-based algorithms such as those we study, and previous work such as A^* for multi-agent path finding [45, 52] and conflict-based search [48] are thus relevant in this setting. It was shown that multi-agent path planning is related to network flow problems, and that one can use algorithms for the latter to compute plans [63]. It is thus important to understand the computational complexity of these graph problems to understand the limits of the algorithmic solutions and as well as heuristics that can be applied to the problem at hand.

The variant of MAPF with connectivity constraints and related computational complexity results were studied in [26, 54]. In [54], complexity results are presented for connectivity constraints with and without collision constraints and the existence of a plan of arbitrary length in both cases was shown to be PSPACE-complete in undirected graphs. Interestingly, it means that it has the same complexity as classical planning [8]. The work in [7] considers the path coverage problem and provides experiments in which instances are described in Planning Domain Description Language and solved with the planner Functional STRIPS [21]. In other words, MAPF with connectivity constraints is more difficult than MAPF with collision constraints: deciding the existence of a collision-free plan of arbitrary length is in P, as stated in [64].

Now, concerning the optimization problems, there is a subtlety about the encoding (unary vs binary) of the length of plans. For MAPF with collision constraints, as the existence of a plan is equivalent to the existence of a plan of length $O(|V|^3)$ where $|V|$ is the number of nodes [64], the encoding of the length is not relevant. Thus, several papers on the topic do not specify the encoding [5, 35, 41, 61–63]. However, for MAPF with connectivity constraints, there is no such bound on the length of a plan. We show that the existence of a bounded plan when the bound is given in binary is PSPACE-complete; while when the bound is given unary, the problem is NP-complete, even on undirected graphs, as claimed by Hollinger and Singh in [26], although they do not explicitly specify the membership proof and are ambiguous about the encoding.[1]

We mainly consider a setting where collisions are ignored. In fact, we are interested in the computational complexity of maintaining connectivity in plans. As in [26], agents can be assumed to be equipped with a low-level collision avoidance system in some cases. Additionally, in drone applications with few agents, different altitudes can be used to avoid

[1] The bounded plan is clearly in NP when the bound ℓ is written in unary: simply guess a solution of length at most ℓ and check that it is correct. When the bound is given in binary, a solution may be in exponential size, and cannot be guessed in polynomial time.

collisions. Nevertheless, we do discuss the impact of taking collision constraints into account in our results as well.

In this work, we are interested in the computational complexity of two problems: **Reachability** is the variant of MAPF with connectivity constraints, without collision constraints; and **Coverage** is the variant of path coverage planning problem for multiple agents with connectivity constraints. As in [26, 54], a problem instance is a *topological graph*, which is a set of nodes given with *movement edges* along which agents can move, and *communication edges* which determine pairs of vertices at which communication is possible. Our results are as follows. We establish the computational complexity of determining the existence of plans, showing that **Reachability** is PSPACE-complete on directed graphs as well, and proving that **Coverage** is PSPACE-complete on both types of graphs. We study bounded versions of these problems where a bound on the length of the plan is given as part of the input; these are denoted by **bReachability** and **bCoverage**. As in [56], we advocate for lengths of plans written in unary although we also study the complexity of our problems when the encoding is binary. We show that both **bReachability** and **bCoverage** are NP-complete when the bound is given in unary, and we clearly state their PSPACE-complete when the bound is given in binary.

Given the prohibitively high complexity reported above, we are interested in searching for 'easier' classes of topological graphs which are realistic for the purpose of information gathering missions. One of our main contributions is the identification of a natural class of topological graphs, called *sight-moveable*, for which we give efficient algorithms. This class requires that whenever an agent can communicate with another node, then it can also move to that node while maintaining direct communication. This can be seen as a restriction on allowed topological graphs; however, if the graph at hand does not have this property, it may be possible to enforce it by removing some communication edges (not allowing the planning algorithm to rely on those edges). Section 6.3 describes a way to obtain sight-moveable graphs from a given topological graph. Thus, any plan found in the obtained sight-moveable graph can be applied on the original one.

The class of sight-moveable graphs offers good computational properties: both **Reachability** and **Coverage** belong to NLOGSPACE, meaning that they can be solved by a non-deterministic algorithm that only uses a logarithmic amount of memory. Practically, it means that algorithms for generating plans can be parallelized, because NLOGSPACE is included in the Nick's class (NC), [16], known to represent decision problems for which there is a parallel algorithm. However, the bounded versions remain NP-complete. We complete the investigation with the complexity analysis of the problem with complete communication graphs, where all pairs of nodes can communicate. In addition, we give complexity results on several variants of our problems.

Note that the PSPACE lower bound given in [54] concerned the reachability problem where agents start from arbitrary locations, whereas we prove that the PSPACE lower bound still holds for agents starting all from the base. Furthermore, this article extends the work of [11, 12], in which the complexity of **Reachability** (when agents start from the base) and **Coverage** on undirected topological graphs was left open. In this article, we solved them (Theorems 3 and 4). We also include detailed proofs and discuss relevant extensions that were not considered in previous work.

Overview We first discuss related work in Sect. 2. Section 3, we introduce the required notions for the rest of the paper. In Sect. 4, we describe the upper bounds of our problems on directed topological graphs and, in Sect. 5, we prove the lower bounds on undirected topological graphs to obtain completeness results. In Sect. 6, we study sight-moveable topological graphs. Section 7 contains the complexity analysis for complete-communication

topological graphs. We introduce relevant extensions of our problems, in Sect. 8. We present a conclusion and future works in Sect. 9.

2 Related work

Planning problems for multiple agents were considered in the robot motion planning setting, for instance the case of two disc-shaped robots moving in the presence of polygonal obstacles [47]. The planning problem for multiple agents is PSPACE-hard even when robot shapes are restricted to simple tiles [25]. PSPACE-hardness was shown in case of unit-square robots with polygonal obstacles as well when agents are *unlabelled/homogeneous*, or in our terminology, anonymous [51]. A line of algorithms that avoid this high complexity are those based on sampling methods such as [29, 32]. These randomly sample points and check whether these points can be connected respecting the constraints of the system. This allows one to quickly generate a tree or a graph between sampled points, and continue sampling points until the desired plan is found.

The coverage path planning problem has been studied in different settings; see the surveys [9, 14, 23]. This problem can be used to solve inspection (where a given set of points of interests must be observed), surveillance (points of interest must be continuously visited) problems, and can be used in applications such as lawn mowing and floor cleaning. Sampling-based algorithm were given in this setting as well. One can prove probabilistic completeness results in some cases, which means that if a solution exists, then the algorithm will eventually find it with probability 1; although it may not be able to detect that no solution exists [18]. Some of these algorithms work in two phases: the first phase consists in sampling points and checking the feasibility of the edges between them, and the second phase consists in solving a graph problem on this structure, and repeating phase 1 if no solution is found. Some works use approximations of metric *traveling salesman problem* (TSP) to find solutions in the constructed graph [17, 19]. Other works use A^* search algorithms with suitable heuristics to compute approximate solutions [22].

Several works consider the coverage path planning problem for multiple agents [9, 13, 23]. In [33], a centralized framework is given to compute plans for the persistent coverage problem without communication constraints. In [43], algorithms are given for the repeated coverage problem in unknown environments for a team of robots, including the case of the line-of-sight communication restriction. This uses the cell decomposition given in [15] for the coverage path planning. In [24, 65] a graph structure is obtained thanks to a cell decomposition, and variants of spanning trees are computed on this graph to ensure coverage with multiple robots. Graph problems related to this problem are studied in [59]. The generalization of traveling salesman problem was studied in [3] which is a relevant problem for coverage path planning in the multi-agent setting. Some works also consider maneuverability and camera angle constraints [1].

The continuous communication restriction appears in several works, e.g. [37, 39, 55, 60] which provide experiments to demonstrate the feasibility of the proposed algorithms. Some works consider the use of dynamic teams of robots which exchange their information on current solutions [38]. Communication and battery restrictions are considered together in [10] which gives an algorithm in which robots can adopt the roles for exploring, meeting, sacrificing themselves (continuing the mission in case of low battery), and serving as communication relay for other robots. Different communication restrictions have been

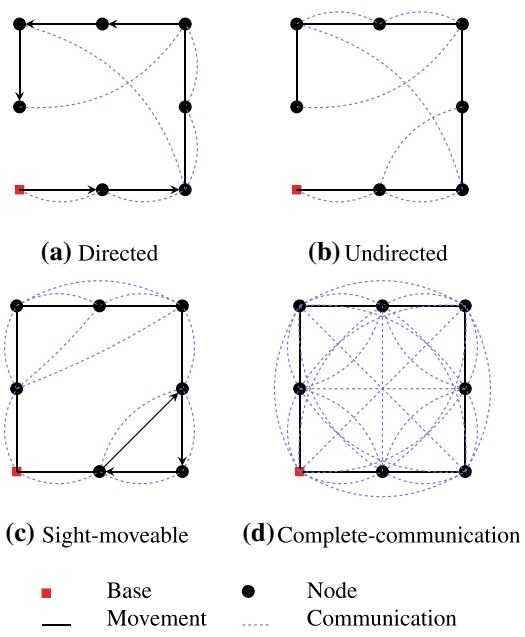


Fig. 1 Examples of topological graphs

considered as well [2] such as event-based communication where the discovery of new information triggers communication (see e.g [6]).

3 Preliminaries

In some path finding applications, one considers a discretization of the space which yields a graph of movements on which algorithms are run. For instance, *regular grids* which decompose the space in square, triangular or hexagonal cells, *irregular grids* with techniques such as *quadtrees* [20, 31] or *Voronoï diagrams* comprehensively discussed in the survey [4].

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.

We will, now, introduce the notions used throughout this work. We define formally the general topological graphs and some subclasses, the notion of execution and the properties considered, the decision problems we investigate and some known results on these problems.

3.1 Topological graphs

Our problems require graphs with two types of edges: *movement edges* along which agents can move, and *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 and examples are depicted in Fig. 1.

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

The node B is the supervision base station from which the agents start the mission and with which they are required to keep communication.

In some applications movement edges are reversible, that is, if an agent can travel from a node to another, it can also go back to the former through the same edge. Undirected graphs thus naturally arise in some applications. See Fig. 1b for an example.

Definition 2 (*Undirected topological graph*) A topological graph is said to be *undirected* if $\langle V, E_m \rangle$ is an undirected graph.

Let us now introduce our new class, called *sight-moveable* topological graphs, which is one of our main contributions. This class requires the movement edges to be reflexive. In addition, whenever an agent can communicate with another node, then it can also move to that node while maintaining the communication without using another agent as a relay. An example of a sight-moveable graph is given in Fig. 1c.

Definition 3 (*Sight-moveable topological graph*) A *sight-moveable* topological graph $G = \langle V, E_m, E_c \rangle$ is a directed topological graph such that

1. $E_m \subseteq E_c$,
2. for all $v \in V$, $(v, v) \in E_m$,
3. and whenever $(v, v') \in E_c$, there exists a sequence $\rho = \langle \rho_1, \ldots, \rho_n \rangle$ of nodes such that $v = \rho_1$, $v' = \rho_n$, $(v, \rho_i) \in E_c$ and $(\rho_i, \rho_{i+1}) \in E_m$ for all $i \in \{1, \ldots, n-1\}$.

Last, we define the *complete-communication* topological graphs which are simply sight-moveable topological graphs with undirected movement and complete communication topology. An example of such a graph is depicted in Fig. 1d, and the formal definition is the following.

Definition 4 (*Complete-communication topological graph*) A topological graph is said to be a *complete-communication* if it is a sight-moveable topological graph such that $\langle V, E_m \rangle$ is an undirected graph and $E_c = V \times V$.

Observe that a complete-communication graphs are reflexive, undirected, connected graphs with communication edges between each pair of nodes.

We say that a topological graph has a *planar (grid) movement graph* iff the graph $\langle V, E_m \rangle$ is a planar (resp. grid) graph.

3.2 Execution

An *execution* is a finite sequence of *configurations* describing the positions of the agents during the mission. We require that all agents should be connected to the base in all configurations. We will use multi-sets to denote nodes occupied by agents in configurations, since agents are *anonymous*. In other terms, if the goal is to reach a target configuration, it

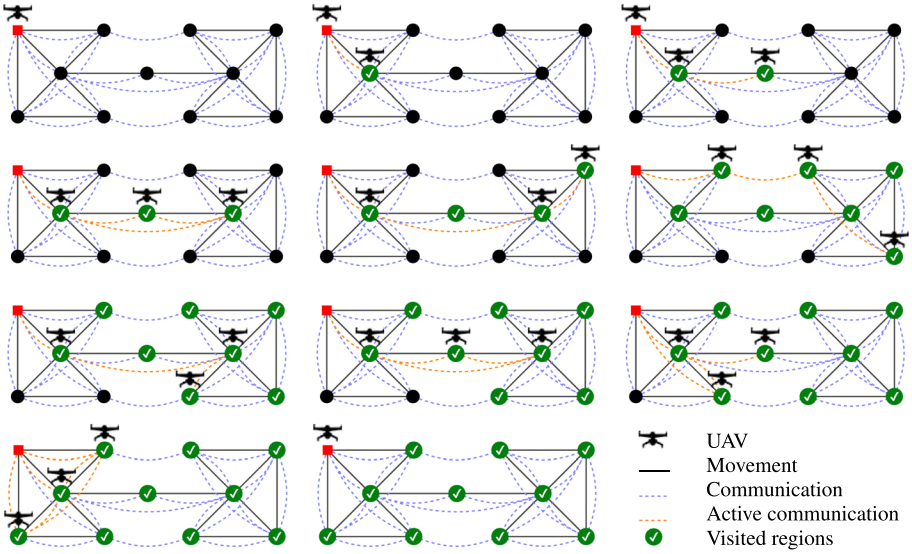


Fig. 2 Example of a covering execution of length 10 with 3 agents. The plan is depicted from left to right and from top to bottom

does not matter which agent occupies which node, as long as there is the right number of agents at each node. Other works use the term *unlabelled* or *homogeneous* (see e.g. [51]); but we use the terminology of MAPF.

The formal definition of a configuration is the following.

Definition 5 (*Configuration*) A *configuration* c of n agents in a topological graph G is a multi-set of elements of V of size n, denoted $c = \langle c_1, \ldots, c_n \rangle$.

A configuration is said to be *connected* iff the graph $\langle V_c, E_c \cap (V_c \times V_c) \rangle$ is connected with $V_c = \{B, c_1, \ldots, c_n\}$.

Given a topological graph $G = \langle V, E_m, E_c \rangle$, we write $c \rightarrow_G c'$ to say that agents in c perform one-step movements to occupy nodes in c'. Formally, we have $c \rightarrow_G c'$ if c (resp. c') can be written as $\langle c_1, \ldots, c_n \rangle$ (resp. $\langle c'_1, \ldots, c'_n \rangle$), and $(c_i, c'_i) \in E_m$ for all $1 \leq i \leq n$.

Definition 6 (*Execution*) An *execution* e of length ℓ with n agents in a topological graph G is a sequence of connected configuration $\langle c^1, \ldots, c^\ell \rangle$ such that $c^j \rightarrow_G c^{j+1}$ for all $1 \leq i < \ell$.

In our setting, the *makespan* of an execution is equal to its length.

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 $j \in \{1, \ldots, \ell\}$ such that v appears in c^j. An example of such an execution is depicted in Fig. 2 (from left to right).

3.3 Decision problems

We formally define the problems **Reachability** and **Coverage** and their bounded versions **bReachability** and **bCoverage**.

3.3.1 The unbounded case

Definition 7 (**Reachability**) Given a topological graph G, an integer n 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 8 (**Coverage**) Given a topological graph G and an integer n, decide if there exists a covering execution with n agents.

We also consider variants **Reachability-init** and **Coverage-init** denote the problems in which the agents start at a given configuration rather than at the base. In other words, the initial configuration is part of the input. In addition, **Coverage--init** requires the agents to return to the initial configuration rather than to the base.

In the above problems, the encoding of the integer n (unary or binary) does not matter. Indeed, in **Reachability**, **Reachability-init** and **Coverage-init**, the input already contains some configuration which is of size n; in **Coverage**, it is useless to have n greater than the number of nodes.

3.3.2 The bounded case

The bounded versions are inspired from the so-called polynomial-length planning problem [56] in which we ask for the existence of a plan of length bounded by a polynomial in the size of the planning task. This can be seen as the decision problem for the optimization problem that seeks to minimize the length of an execution, except that we assume that the bound is given in *unary*. In fact, the goal of planning algorithms is to compute plans, so given a bound ℓ on the length of the desired plan, the algorithm always allocates memory space of size $\Omega(\ell)$ to store the plan.

That is why we use unary encoding in the following definitions. Binary encoding of the length ℓ is discussed as well in Section 8.1.

Definition 9 (**bReachability**) Given a topological graph G, an integer n and a configuration c of size n and ℓ an integer 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 10 (**bCoverage**) Given a topological graph G, an integer n, an integer ℓ written in unary, decide if there exists a covering execution of length ℓ' such $\ell' \leq \ell$.

3.3.3 Restriction to subclasses of graphs

We consider the restriction of the above problems to the following subclasses of graphs: directed graphs (denoted by dir), undirected graphs (denoted by und), sight-moveable graphs (denoted by sm) and complete-communication graphs (denoted by cc). The variants of these problems to a given graph class will be denoted using a subscript, that is, **Reachability**$_\star$, **Coverage**$_\star$, **bReachability**$_\star$, **bCoverage**$_\star$, **Reachability-init**$_\star$, and **Coverage-init**$_\star$ denote the restriction of these problems to graphs of type $\star \in \{\text{und}, \text{dir}, \text{sm}, \text{cc}\}$.

<table>
<tr><th>Top. Graph/Problem</th><th>Reachability (Def. 7)</th><th>Coverage (Def. 8)</th><th>bReachability (Def. 9)</th><th>bCoverage (Def. 10)</th></tr>
<tr><td>Directed (Def. 1)</td><td>PSPACE-complete (Th. 3)</td><td rowspan="2">PSPACE-complete (Th. 4)</td><td rowspan="2">NP-complete [26]</td><td rowspan="4">NP-complete (Th. 8)</td></tr>
<tr><td>Undirected (Def. 2)</td><td>PSPACE-complete [54]</td></tr>
<tr><td>Sight-Moveable (Def. 3)</td><td rowspan="2">in LOGSPACE (Prop. 3)</td><td rowspan="2">in NLOGSPACE (Prop. 4)</td><td>NP-complete (Th. 6)</td></tr>
<tr><td>Complete-Comm. (Def. 4)</td><td>in NLOGSPACE (Prop. 5)</td></tr>
</table>

Fig. 3 Overview of the complexity results

3.4 Known results

The connected version of MAPF was introduced in [26], 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 non-anonymous agents in a bounded number of steps with collisions allowed is NP-hard:

Theorem 1 ([26]) **bReachability-init**$_{\text{und}}$ *is NP-hard.*

As stated before, the above paper actually states the NP-completeness of this problem but without specifying the encoding of the bound.

Tateo et al., in [54], establish the complexity of **Reachability-init**$_{\text{und}}$:

Theorem 2 ([54]) **Reachability-init**$_{\text{und}}$ *is PSPACE-complete.*

The setting of Tateo et al. is identical to ours with the exception that the starting configuration is part of the input in their case. We relate both problems showing PSPACE-hardness when agents all start at the base.

3.5 Overview of results

In the rest of the paper, we study upper and lower complexity bounds for the defined decision problems on different topological graphs. The following sections present our results, respectively, for the general case, the undirected graphs, sight-moveable graphs, and complete-communication graphs. An overview of these results is given in Fig. 3.

4 Directed topological graphs

In this section, we will consider the previous problems restricted to the class of directed topological graphs. These problems are thus denoted by **Reachability**$_{\text{dir}}$ and **bReachability**$_{\text{dir}}$ (resp. **Coverage**$_{\text{dir}}$ and **bCoverage**$_{\text{dir}}$), with dir denoting the class of directed topological graphs.

In this section, we show upper bounds for all our problems in the general case, that is, for directed topological graphs. Observe that this also provides upper bounds for other classes such as undirected graphs.

For the unbounded problems, we can design a straightforward non-deterministic algorithm running in polynomial space, that guesses an execution by keeping in memory the last configuration, and, for **Coverage**, the set of visited regions. In fact, the number of configurations is exponential, and a single configuration can be stored in polynomial space. Moreover, one can easily bound the length of executions by an exponential as well. We conclude with Savitch's Theorem (NPSPACE=PSPACE) [46].

Proposition 1 **Coverage** *and* **Reachability** *are in PSPACE.*

For the bounded versions of the problems, since the bound is encoded in unary, one can guess and check a path of bounded length in polynomial time. The result follows.

Proposition 2 **bCoverage** *and* **bReachability** *are in NP.*

5 Undirected topological graphs

In this section we prove the PSPACE lower bound of the problems **Reachability** and **Coverage** on undirected topological graphs.

We consider the result of Theorem 2 in the setting where all agents start at the base.

Theorem 3 $\textbf{Reachability}_{\text{und}}$ *is PSPACE-complete.*

Proof The upper bound comes from Proposition 1. The lower bound is by reduction from $\textbf{Reachability-init}_{\text{und}}$ (see Theorem 2). Let us denote by a tuple (G, B, n, s, t) the instances of $\textbf{Reachability-init}_{\text{und}}$ where G is the graph, B the base, n the number of agents, s the initial configuration and t the target configuration. Instances of $\textbf{Reachability}_{\text{und}}$ will be denoted by (G, B, n, t) as the initial configuration is fixed.

Let (G, B, n, s, t) be an instance of $\textbf{Reachability-init}_{\text{und}}$. We show how to map (G, B, n, s, t) to an instance of $\textbf{Reachability}_{\text{und}}$ in polynomial time. We construct the instance $(G', B', n+2, t')$ of $\textbf{Reachability}_{\text{und}}$ where G' is given in Fig. 4, B' is the base, and the final configuration t' is $\langle t_1, \ldots, t_n, t_{n+1}, t_{n+2}\rangle$.

Let us describe more precisely the construction of G'. We write $s = \langle s_1, \ldots, s_n\rangle$ and $t = \langle t_1, \ldots, t_n\rangle$. The graph G' contains the graph G: in particular, it contains B that is no longer the base; a new node B' is now the base. Let us describe the construction of G'.

Nodes We first create two layers of $n+2$ vertices. The first layer is composed of $v_1, \ldots, v_n, v_{n+1}, v_{n+2}$ and the second of $v'_1, \ldots, v'_n, v'_{n+1}, v'_{n+2}$. We also add two nodes t_{n+1} and t_{n+2}.

Movement We add movement edges between t_{n+1} and v'_{n+1}, t_{n+2} and v'_{n+2}, t_{n+1} and t_{n+1}, and t_{n+2} and t_{n+2}. Then, the role of t_{n+1} and t_{n+2} is to relay the communication from B' to nodes in G. We connect B' to the first layer, i.e. with a movement edge between B' and v_i, for all $1 \leq i \leq n+2$. The first layer has movement edges to the second layer, i.e. with a movement edge between v_i and v'_i, for all $1 \leq i \leq n+2$. The n first vertices of the

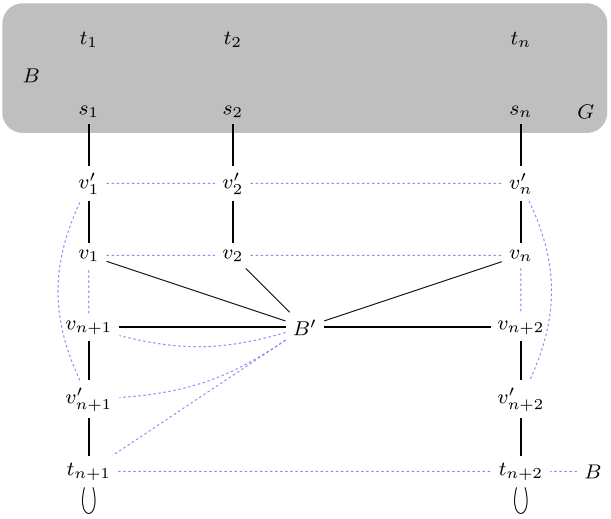


Fig. 4 Reduction of **Reachability-init**$_{\text{und}}$ into **Reachability**$_{\text{und}}$. The node t_{n+2} communicates with all nodes v such that v communicates with B

second layer has movement edges to the initial configuration s such that there is a movement edge from v'_i to s_i, for all $1 \leq i \leq n$.

Communication We add communication edges from B' to t_{n+1}, from t_{n+1} to t_{n+2} as well as from t_{n+2} to B and if there exists v such that B communicates v then we create a communication edge from t_{n+2} to v. We add a communication edge from B' to v_{n+1}, from v_i to v_{i+1}, for all $1 \leq i < n$, from v_{n+1} to v_1 and from v_n to v_{n+2}. We repeat this last procedure for the second layer as well.

We now give the formal definition of our reduction. In the sequel, we use the symbol $\sqcup$ to emphasize that the union is of *disjoint* sets. Formally, given (G, B, n, s, t) with $G = \langle V, E_m, E_c \rangle$ with base B, we define $(G', B', n+2, t')$ where $G' = \langle V', E'_m, E'_c \rangle$ with base B' where:

- $V' := V \sqcup \{B', v_1, \ldots, v_n, v'_1, \ldots, v'_n, v_{n+1}, v_{n+2}, v'_{n+1}, v'_{n+2}, t_{n+1}, t_{n+2}\}$
- E'_m is the symmetric closure of
  $E_m \cup \{(B', v_1), \ldots, (B', v_n), (B', v_{n+1}), (B', v_{n+2})\}$
  $\cup \{(v_1, v'_1), \ldots, (v_n, v'_n)\} \cup \{(v'_1, s_1), \ldots, (v'_n, s_n)\}$
  $\cup \{(v_{n+1}, v'_{n+1}), (v_{n+2}, v'_{n+2}), (v'_{n+1}, t_{n+1}), (v'_{n+2}, t_{n+2})\}$
  $\cup \{(t_{n+1}, t_{n+1}), (t_{n+2}, t_{n+2})\}$;
- E'_c is the symmetric closure of
  $E_c \cup \{(v'_1, v'_2), \ldots, (v'_{n-1}, v'_n)\} \cup \{(v_1, v_2), \ldots, (v_{n-1}, v_n)\}$
  $\cup \{(B', v_{n+1}), (B', v'_{n+1}), (B', t_{n+1})\}$
  $\cup \{(v'_1, v'_{n+1}), (v'_n, v'_{n+2}), (t_{n+2}, B)\}$
  $\cup \{(t_{n+2}, v) \mid (B, v) \in E_c\}$.

□

It is worth noting that all connected configurations in G are now connected via B', t_{n+1} and t_{n+2}. We now show the instance (G, B, n, s, t) of **Reachability-init**$_{\text{und}}$ is feasible if, and only if in the constructed instance $(G', B', n+2, t')$ of **Reachability**$_{\text{und}}$ is feasible.

($\Rightarrow$) Suppose that t is reachable from s in the instance (G, B, n, s, t). We construct an execution for $(G', B', n+2, t')$ as follows. All agents start from B'. The $n+2$ agents first reach $\langle v_1, \ldots, v_{n+2} \rangle$, then $\langle v'_1, \ldots, v'_{n+2} \rangle$, then $\langle s_1, \ldots, s_n, t_{n+1}, t_{n+2} \rangle$. After that, the agents at positions $s_1, \ldots, s_n$ reach positions $t_1, \ldots, t_n$ (following the same plan as in for (G, B, s, t)), while the two others remain in t_{n+1} and t_{n+2}.

($\Leftarrow$) Let us consider an execution e from $\langle B', \ldots, B' \rangle$ to t' for $(G', B', n+2, t')$. Let us extract an execution from s to t for (G, B, n, s, t). In order to do so, we prove the Facts 1 and 2.

Fact 1 The configuration $\langle s_1, \ldots, s_n, t_{n+1}, t_{n+2} \rangle$, up to a permutation, appears in the execution e.

Proof The configuration t' is reached at the end of the execution e and the $n+2$ agents started at B'. Thus, n agents must enter into G and be at one of the s_i at some point. Let us consider the moment of the execution e when an agent, denoted a, occupies a node s_i. For that, it must be at v'_i before going to s_i. Furthermore, for agent a to be connected at v'_i the nodes v'_{n+1} and $v'_1, \ldots, v'_{i-1}$ must be occupied. At the next step, at least the nodes t_{n+1} and t_{n+2} must be occupied for agent a to be connected at s_i. Hence, when a is at v'_i, the agent which will occupy t_{n+2} must be at v'_{n+2}, thus, forcing $v'_{i+1}, \ldots, v'_n$ to be occupied. No agent in this configuration can move to a node v_i of the first layer. Indeed, this agent would be disconnected, lacking a relay at v_{n+1}. Therefore, the next configuration must be $\langle s_1, \ldots, s_n, t_{n+1}, t_{n+2} \rangle$. □

Now, by Fact 1, we can consider the last time at which the agents are in the configuration $\langle s_1, \ldots, s_n, t_{n+1}, t_{n+2} \rangle$. We prove the Fact 2.

Fact 2 Between that last time and the end of the execution, there are always the same n agents in G, one agent on t_{n+1} and one on t_{n+2}.

Proof A similar reasoning to the last one can be used to show that if an agent is located at s_i before moving to v'_i then nodes from s_1 to s_n as well as t_{n+1} and t_{n+2} are occupied. Thus, since this is the last time the agents are in the configuration $\langle s_1, \ldots, s_n, t_{n+1}, t_{n+2} \rangle$, none of the n first agents can move out of G. Furthermore, t_{n+1} and t_{n+2} must stay occupied in order for the n agents in G to be connected to B'. □

From Facts 1 and 2, the positions of the first n agents in the portion of the execution between the last time in $\langle s_1, \ldots, s_n, t_{n+1}, t_{n+2} \rangle$ and t' are fully in G and gives an execution starting at s and finishing at t for (G, B, n, s, t). □

We now turn our attention to $\textbf{Coverage}_{\text{und}}$. We start by establishing the PSPACE-completeness of $\textbf{Coverage-init}_{\text{und}}$, and then show how to reduce this problem to $\textbf{Coverage}_{\text{und}}$.

Lemma 1 $\textbf{Coverage-init}_{\text{und}}$ *is PSPACE-complete*.

Proof The membership of $\textbf{Coverage-init}_{\text{und}}$ to PSPACE can be shown by using the same arguments as for the proof of Proposition 1. The proof of PSPACE-hardness is obtained by polynomial reduction from $\textbf{Reachability}_{\text{und}}$. We map a $\textbf{Reachability}_{\text{und}}$-instance (G, B, n, t) to the $\textbf{Coverage-init}_{\text{und}}$-instance $(G', B, 2n, s)$ where G' is depicted in Fig. 5

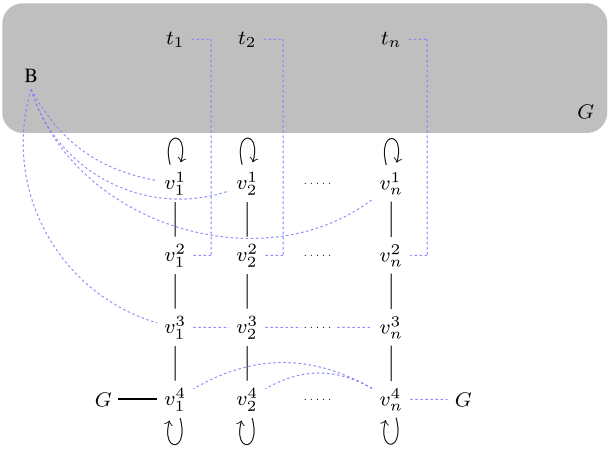


Fig. 5 Reduction of **Reachability**$_{\text{und}}$ to **Coverage-init**$_{\text{und}}$

and starting at the configuration $s = \langle B, \ldots, B, v_1^1, \ldots, v_n^1 \rangle$. The definition of s means that n agents start at the base B and n agents start in positions $v_1^1, \ldots, v_n^1$, that are in the reduction gadget. Let us describe the construction of G'.

Nodes	We make four layers of vertices from v_1^1, ..., v_n^1 to v_1^4, ..., v_n^4.
Movement	We create movement edges between v_i^j and v_i^{j+1}, with $1 \leq i \leq n$ and $1 \leq j \leq 3$. Each node of the first and last layers have self-loops. The node v_1^4 has a movement edge to all nodes of G.
Communication	The node v_n^4 has a communication edge to all nodes of G and all nodes of the fourth layer. We create a communication edge between v_i^2 to t_i, for all $1 \leq i \leq n$. We create a communication edge between v_i^3 and v_{i+1}^3, for all $1 \leq i < n$. Finally, we connect in communication the nodes of the first layer and the node v_1^3 to the base B.

Formally, given (G, B, n, t) with $G = \langle V, E_m, E_c \rangle$ with base B, we define $(G', B, 2n, s)$ where $G' = \langle V', E'_m, E'_c \rangle$ with base B where:

- $V' := V \sqcup \{v_1^1, \ldots, v_n^1, v_1^2, \ldots, v_n^2, v_1^3, \ldots, v_n^3, v_1^4, \ldots v_n^4\}$
- E'_m is the symmetric closure of
  $\cup E_m \cup \{(v, v_1^4) \mid v \in V\}$
  $\cup \{(v_i^1, v_i^1), (v_i^1, v_i^2), (v_i^2, v_i^3), (v_i^3, v_i^4), (v_i^4, v_i^4) \mid i \in \{1, \ldots, n\}\}$
- E'_c is the symmetric closure of
  $E_c \cup \{(B, v_1^1) \ldots (B, v_n^1)\} \cup \{(t_1, v_1^2), \ldots, (t_n, v_n^2)\}$
  $\cup \{(B, v_1^3), (v_1^3, v_2^3), \ldots, (v_{n-1}^3, v_n^3)\}$
  $\cup \{(v_1^4, v_n^4) \ldots (v_{n-1}^4, v_n^4)\} \cup \{(v_n^4, v) \mid v \in V\}$;

and $s = \langle B, \ldots, B, v_1^1, \ldots, v_n^1 \rangle$.

We now show the instance (G, B, n, t) of **Reachability**$_{\text{und}}$ is feasible iff the instance $(G', B, 2n, s)$ of **Coverage-init**$_{\text{und}}$ is feasible.

($\Rightarrow$) Suppose the configuration t is reachable in G. Let us construct an execution starting at s in G' that cover all nodes in G'. First, the first n agents reach configuration t, while

the other n agents stay at layer v_i^1 using the self-loops. After that, the first n agents stay at t while the others progress to the fourth layer v_i^4. Finally, while the agents at v_2^4 to v_n^4 don't move using the self loops, the agent at v_1^4 covers the whole graph since the presence of an agent at v_n^4 makes sure that all nodes of G are connected to other agents and to the base. Once finished, this agent can go back to v_1^4 and the n agents can return back to the layer 1 as initially. Finally, the n agents occupying configuration t can go back to B by following the same path in reverse. This constitutes a covering execution for **Coverage-init**$_{\text{und}}$.

($\Leftarrow$) Assume that G' can be covered, and let T denote the first time v_n^4 is visited. The agent at v_n^4 can only be at v_n^3 at time $T-1$. Thus, the nodes $v_1^3, \ldots, v_{n-1}^3$ must also be occupied due to connectivity constraint. Then, at time $T-2$, all n agents were on the second layer v_i^2. In fact, the layer 4 must be empty at this point by the choice of T and due to connectivity edges (and note also that there are no self-loops on the third layer). But since each node v_i^2 is only connected to t_i, at time $T-2$ the other n agent must be at configuration t. This concludes the proof. □

Theorem 4 **Coverage**$_{\text{und}}$ *is PSPACE-complete*.

Proof The upper bound comes from Proposition 1. We present the lower bound, which is by reduction from **Coverage-init**$_{\text{und}}$. We map a **Coverage-init**$_{\text{und}}$-instance (G, n, s) to the **Coverage**$_{\text{und}}$-instance $(G', n+2)$ of where G' is defined as in Fig. 4 (ignoring vertices $t_1, \ldots, t_n$). The formal description is given in the proof of Theorem 3.

We show that instance (G, n, s) of **Coverage-init**$_{\text{und}}$ is feasible iff instance $(G', n+2)$ of **Coverage**$_{\text{und}}$ is feasible. The proof is very similar to the proof of Theorem 3. The first n agents are used for the execution in G while the two others operate in the gadget.

($\Rightarrow$) If G can be covered starting from s then in G', the agents can first reach s as described in proof of Theorem 3, then follow the same plan to cover the graph, and execute the plan in reverse to come back to s and then back to B'. Meanwhile, the two others reach t_{n+1} and t_{n+2} loops there and come back to B'.

($\Leftarrow$) Assume there is a covering execution in G' from base B'. We already proved in Theorem 3 (Fact 1) that from configuration $\langle B', \ldots, B' \rangle$ the agents necessarily go through configuration $\langle s_1, \ldots, s_n, t_{n+1}, t_{n+2} \rangle$ to go in G and that the first n agents move in and out of G at the same step. Hence, the execution in G', minus the steps in the gadget, can be reproduced in G. □

We conclude this section by depicting, in Fig. 6, the reductions used and the proof scheme. The dotted arrows represent the unmentioned corollaries.

6 Sight-moveable topological graphs

The main challenge in deciding whether there exists a connected plan is to verify that the given number of agents can visit a node of the graph while staying connected. Indeed, to visit a location connected to, say, the base, an agent might have to rely on multiple other agents. Hence, if we can guarantee that whenever two locations are connected we can move an agent from one to the other without the need of an extra relay, the problem becomes "easy". This assumption underlies the definition of sight-moveable topological graphs. Interestingly, the unbounded decision problems **Reachability**$_{\text{sm}}$ and **Coverage**$_{\text{sm}}$

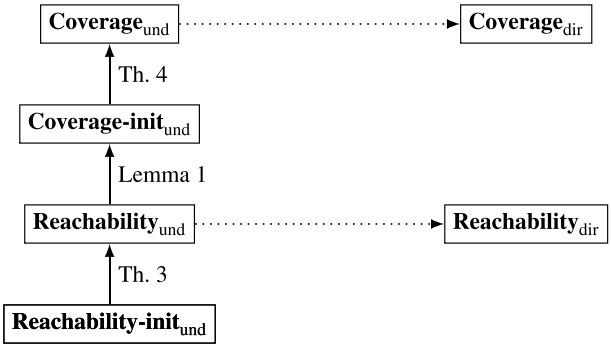


Fig. 6 Reductions of Sect. 5

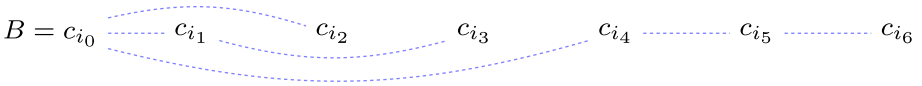


Fig. 7 Example of an ordering of nodes in $V' = \{c_1, \ldots, c_n, B\}$

are in LOGSPACE and NLOGSPACE, respectively. Unfortunately, the bounded version **bReachability**$_{\text{sm}}$ is NP-complete. At the end of the section, we discuss a relaxation method based on this class of graphs.

6.1 Upper bounds

Let us call USTCONN (resp. STCONN) the problem of determining whether two nodes of a given undirected (resp. directed) graph are connected. The algorithms presented in this section rely on the following complexity result:

Theorem 5 ([42]) *USTCONN is in LOGSPACE.*

Proposition 3 **Reachability**$_{\text{sm}}$ *is in LOGSPACE.*

Proof Let us define the problem UCONN as that of checking whether a given undirected graph is connected. By Theorem 5, this problem is in LOGSPACE since it suffices to check the connectivity between all pairs of nodes in LOGSPACE.

We are going to reduce **Reachability**$_{\text{sm}}$ to UCONN in logarithmic space.

Let $G = \langle V, E_m, E_c \rangle$ a sight-moveable topological graph and c a configuration. Let $V' = \{c_1, \ldots, c_n, B\}$. We show that the configuration c is reachable iff the restriction of $\mathfrak{G}' := (V, E_c)$ to the nodes in V' is a connected graph. It is clear that this condition is necessary since if $\mathfrak{G}'$ is not connected then the agents cannot occupy configuration c.

Conversely, assume that $\mathfrak{G}'$ is connected. Then, let us order the nodes $B, c_1, \ldots, c_n$ into $c_{i_0}, c_{i_1}, c_{i_2}, \ldots, c_{i_n}$ with $c_{i_0} = B$, such that for all $1 \leq j \leq n$, c_{i_j} is connected to some c_{i_k} with $0 \leq k < j$; such an order exists, and can be obtained by breadth-first search in a E_c-spanning tree of V' at root B, see Fig. 7 for an example.

We then construct an execution for reaching the configuration c as follows. The construction is by induction on $0 \leq j \leq n$: at step j, the first j agents occupy nodes $c_{i_1}, \ldots, c_{i_j}$.

The base case $j = 0$ is the empty execution. For $j \leq 1$, let $k < j$ such that $(c_{i_k}, c_{i_j}) \in E_c$. We send the j-th agent to c_{i_k} following the path that has been constructed earlier, by induction. We then move that agent to c_{i_j} following a path that stays connected with c_{i_j}, which exists by the sight-moveable property.

Thus, (G, n, c) is a positive $\textbf{Reachability}_{\text{sm}}$-instance iff $\mathfrak{G}'$ is a positive UCONN-instance. The reduction is in logarithmic space: we compute $\mathfrak{G}'$ by enumerating all E_c -edges (u, v) 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. [50], Ch. 8, Def. 8.21). □

Before giving the complexity of $\textbf{Coverage}_{\text{sm}}$, we need the following intermediary result. Let us call Bounded-USTCONN the following problem: given an undirected graph $\mathfrak{G}$, two nodes s, t, an integer n, decide whether there is a path of length at most n from s to t in G. Note that, whatever the encoding of n is, the problem Bounded-USTCONN can be decided in logarithmic space.

Lemma 2 *Bounded-USTCONN is in NLOGSPACE.*

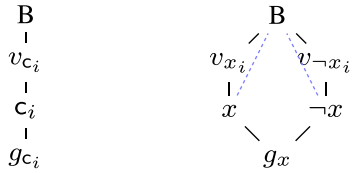
Proof We reduce Bounded-USTCONN to STCONN in logarithmic space as follows. From a Bounded-USTCONN instance $(\mathfrak{G}, s, t, n)$ we construct in logarithmic space a STCONN instance $(\mathfrak{G}', s', t')$:

1. The nodes of $\mathfrak{G}'$ are pairs (v, j) where v is a node of $\mathfrak{G}$ and $j \in \{0, 1, \ldots, m\}$ where m is the minimum of n and the number of nodes of $\mathfrak{G}$.
2. The graph $\mathfrak{G}'$ 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 $t' = (t, n)$.

The STCONN instance $(\mathfrak{G}', s', t')$ can be computed in log-space. Step 1 requires to store the current node v of $\mathfrak{G}'$ to be processed and an integer j written in binary, that is of size $\log m$. Importantly, as m is smaller than the number of nodes in $\mathfrak{G}'$, the representation of m is logarithmic in the size of the input. For this reason, note that the overall spatial complexity does not change even if the encoding of n is in binary. □

Proposition 4 $\textbf{Coverage}_{\text{sm}}$ *is in NLOGSPACE.*

Proof Let $G = \langle V, E_m, E_c \rangle$ be a sight-moveable topological graph and n an integer written in unary. We prove that for all vertices v, there is a path of length at most n from v to the base B in the communication graph iff (G, n) is a positive instance of $\textbf{Coverage}_{\text{sm}}$. One direction is obvious: if (G, n) is a positive instance, then all vertices must within a distance of at most n from the base in the communication graph; otherwise no connected configuration can visit that vertex. Assume that all vertices are within a distance of n from the base. For any vertex v, consider path $v_1, \ldots, v_k$ in the communication graph, with $v_1 = B$, $v_k = B$, and such that $(v_i, v_{i+1}) \in E_c$. We apply the construction of Proposition 3 to build an execution from configuration B^n to some configuration where k agents occupy $\{v_1, \ldots, v_k\}$, and others stay at B. We now extend this execution to "roll back" so that all agents return to the base. We first let the agent at v_k, go to v_{k-1}: this is possible by the sight-moveable property and since $(v_k, v_{k-1}) \in E_c$. Then the two agents together move to v_{k-2}, and so on, until all



(a) Clause gadget **(b)** Variable gadget

Fig. 8 Gadgets for reduction of 3SAT into **bReachability**$_{\text{sm}}$

come back to B. We can now combine these executions to cover all vertices, and let agents end in the base.

Thus, the algorithm consists in checking sequentially, for all v, that $((V, E_c), v, B, n)$ is a positive instance of Bounded-USTCONN. Hence, we obtain a non-deterministic algorithm in logarithmic space to decide **Coverage**$_{\text{sm}}$. □

6.2 Lower bounds

We now focus on the NP lower bound of **bReachability**$_{\text{sm}}$.

Theorem 6 **bReachability**$_{\text{sm}}$ *is NP-complete even for a fixed execution length* $\ell \geq 3$.

Proof The upper bound comes from Proposition 2. The lower bound proof is by polynomial time reduction from **3-SAT** problem (see [28]). 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, k, c) of **bReachability**$_{\text{sm}}$ with $k = n + m$ agents.

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

Please recall that in a sight-moveable graph 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 Fig. 8b: nodes x_i, $\neg x_i$, staging nodes v_{x_i}, $v_{\neg x_i}$ and a *goal* node g_{x_i}. We add movement edges from B to v_{x_i}, from v_{x_i} to x and from x to g_{x_i} (resp. from B to $v_{\neg x_i}$, from $v_{\neg x_i}$ to $\neg x_i$ and from $\neg x_i$ to g_{x_i}). As for the communication, the node x_i (res. $\neg x_i$) communicates with the base.

For each clause c_j, we construct a gadget composed of 3 nodes depicted in Fig. 8a. We create a node c_j, a staging node v_{c_j} and a goal node g_{c_j}. We add movement edges from B to v_c, from v_{c_j} to c_j and from c_j to g_{c_j}. The communication between a clause c_j and a literal x_i or $\neg x_i$ is dictated by the existence of the literal in the clause. We do not use direct communication edges because the obtained topological graph should be sight-moveable. Instead, we use *fully connected paths of length 4*. Such a path between—let say—x_i and c_j consists three intermediate nodes p^1_{ij}, p^2_{ij}, p^3_{ij}, such that there is a path made up of movement edges from x_i to c_j, passing throw p^1_{ij}, p^2_{ij}, p^3_{ij}. Furthermore, we suppose that the nodes x_i, p^1_{ij}, p^2_{ij}, p^3_{ij} and c_j form a clique w.r.t. to the communication edges. Now, there is a fully connected path of length 4 between x_i and c_j if and only if $x_i \in \mathsf{c}_j$; and

there is a fully connected path of length 4 between $\neg x_i$ and c_j if and only if $\neg x_i \in \mathsf{c}_j$.

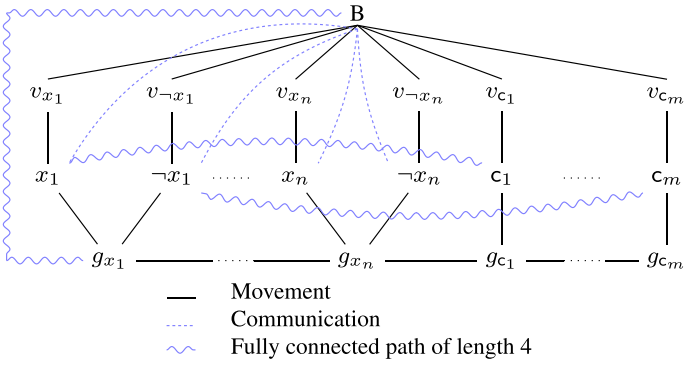


Fig. 9 Reduction of 3SAT into $\mathbf{bReachability}_{\mathrm{sm}}$. Communication edges implied by movement edges are not displayed. The variable x_1 is present in the clause c_1 and c_n

We add movement edges from g_{x_i} to $g_{x_{i+1}}$, and from g_{c_j} to $g_{\mathsf{c}_{i+1}}$ for all $1 \leq i < n$, as well as from g_{x_n} to g_{c_1}. Last, we add a fully connected path of length 4 from g_{x_1} to the base such that $(g_{x_1}, B) \in E_c$, 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 Fig. 9. The snake-like path from g_{x_1} to B is a fully connected path of length 4.

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

Formally, the topologic graph $G = \langle V, E_m, E_c \rangle$ is defined by:

- $V := \{B, v_{x_1}, v_{\neg x_1}, \ldots, v_{x_n}, v_{\neg x_n}, v_{\mathsf{c}_1}, \ldots, v_{\mathsf{c}_m}\}$
  $\cup \{g_{x_1}, \ldots, g_{x_n}, g_{\mathsf{c}_1}, \cdots, g_{\mathsf{c}_m}\}$
  $\cup \{p_B^1, p_B^2, p_B^3\} \cup \left\{ p_{ij}^1, p_{ij}^2, p_{ij}^3 \mid i \in \{1, \ldots, n\}, j \in \{1, \ldots, m\} \right\}$
- E_m is the symmetric closure of
  $\{(B, v_{x_1}), (B, v_{\neg x_1}), \ldots, (B, v_{x_n}), (B, v_{\neg x_n}), (B, v_{\mathsf{c}_1}), (B, v_{\mathsf{c}_1 x_m})\}$
  $\cup \{(v_{x_1}, x_1), (v_{\neg x_1}, \neg x_1), \ldots (v_{x_n}, x_n), (v_{\neg x_n}, \neg x_n)\}$
  $\cup \{(v_{\mathsf{c}_1}, \mathsf{c}_1), \ldots, (v_{\mathsf{c}_m}, \mathsf{c}_m)\}$
  $\cup \{(x_1, g_{x_1}), (\neg x_1, g_{x_1}), \ldots, (x_n, g_{x_n}), (\neg x_n, g_{x_n})\}$
  $\cup \{(\mathsf{c}_1, g_{\mathsf{c}_1}), \ldots, (\mathsf{c}_m, g_{\mathsf{c}_m})\}$
  $\cup \{(g_{x_1}, g_{x_2}), \ldots (g_{x_{n-1}}, g_{x_n}), (g_{x_n}, g_{\mathsf{c}_1}), (g_{\mathsf{c}_1}, g_{\mathsf{c}_2}), \ldots, (g_{\mathsf{c}_{m-1}}, g_{\mathsf{c}_m})\}$
  $\cup \{(B, p_B^1), (p_B^1, p_B^2), (p_B^2, p_B^3), (p_B^3, g_{x_1})\}$
  $\cup \left\{ (x_i, p_{ij}^1), (p_{ij}^1, p_{ij}^2), (p_{ij}^2, p_{ij}^3), (p_{ij}^3, \mathsf{c}_j) \mid x_i \in \mathsf{c}_j \right\}$
  $\cup \left\{ (\neg x_i, p_{ij}^1), (p_{ij}^1, p_{ij}^2), (p_{ij}^2, p_{ij}^3), (p_{ij}^3, \mathsf{c}_j) \mid \neg x_i \in \mathsf{c}_j \right\}$
- E_c is the symmetric closure of
  $E_m \cup \{(B, x_1), (B, \neg x_1), \ldots, (B, x_n), (B, \neg x_n)\}$
  $\cup \left\{ \begin{array}{l} (B, p_B^1), (p_B^1, p_B^2), (p_B^2, p_B^3), (p_B^3, g_{x_1}), \\ (B, p_B^2), (p_B^1, p_B^3), (p_B^2, g_{x_1}) \\ (B, p_B^3), (p_B^1, g_{x_1}), (B, g_{x_1}) \end{array} \right\}$

$$\cup\left\{\begin{array}{ll}(x_i,p^1_{ij}),(p^1_{ij},p^2_{ij}),(p^2_{ij},p^3_{ij}),(p^3_{ij},\mathsf{c}_j),\\(x_i,p^2_{ij}),(p^1_{ij},p^3_{ij}),(p^2_{ij},\mathsf{c}_j) & \mid x_i\in\mathsf{c}_j\\(x_i,p^3_{ij}),(p^1_{ij},\mathsf{c}_j),(x_i,\mathsf{c}_j)\end{array}\right\}$$
$$\cup\left\{\begin{array}{ll}(\neg x_i,p^1_{ij}),(p^1_{ij},p^2_{ij}),(p^2_{ij},p^3_{ij}),(p^3_{ij},\mathsf{c}_j),\\(\neg x_i,p^2_{ij}),(p^1_{ij},p^3_{ij}),(p^2_{ij},\mathsf{c}_j) & \mid \neg x_i\in\mathsf{c}_j\\(\neg x_i,p^3_{ij}),(p^1_{ij},\mathsf{c}_j),(\neg x_i,\mathsf{c}_j)\end{array}\right\};$$

where nodes $p^k_{\bullet}$ are the intermediate nodes in the fully connected paths. □

Fact 3 *G is a sight-moveable topological graph.*

Proof Concerning the communication between the base B and the nodes x_i (resp. $\neg x_i$), a path does exist under the communication of B to reach x_i (resp. $\neg x_i$), due to communication induced by the movement. For the other communication edges, they are part of full connected paths which guarantee the sight-moveable condition. This ends the proof of Fact 3.

□

Now let us prove that a **3-SAT** instance is satisfiable if there exists an execution of at most 3 steps in the graph G.

($\Rightarrow$) We show that if a **3-SAT** instance is satisfiable then there exists an execution of at most 3 steps in the graph G built from it. Let *val* be a truth assignment which satisfies the instance. Recall that there are $n+m$ agents. The first step of the execution consists in moving an agent in each v_{c_i}, and for each variable x_i, moving one agent to v_{x_i} if the $val(x_i)=1$ and to $v_{\neg x_i}$ otherwise. Note that all staging nodes communicate with B.

In the second step, all agents progress to their unique successors other than B. While all nodes x_i and $\neg x_i$ are connected to B, a node c_i is connected to B if and only if there is an agent in one of its literals. This is the case since *val* satisfies the formula. In the third step of the execution, agents go to states g_{x_i} and g_{c_i}. Here, the connection with the base is ensured since g_{x_1} is connected to it, and g_{x_2} is connected to g_{x_1}, g_{x_3} is connected to g_{x_2} and so on.

This execution is thus a solution of $\mathbf{bReachability}_{\mathrm{sm}}$ with bound $\ell=3$.

($\Leftarrow$) We show that if in the graph G there exists an execution of at most 3 steps constructed from a **3-SAT** instance, then the instance is satisfiable. Let us assume that we have an execution e of at most 3 steps with the last configuration being $\langle g_{x_1},\ldots,g_{x_n},g_{\mathsf{c}_1},\ldots,g_{\mathsf{c}_m}\rangle$.

The only shortest path from B to g_{c_i} is of length 3 and goes through v_{c_i}. For states g_{x_i}, the only shortest paths are also of length 3 and go through either v_{x_i} or $v_{\neg x_i}$. Thus, in order to reach the given target configuration, at the initial step, agents must cover the states v_{c_i} and either v_{x_i} or $v_{\neg x_i}$ for all i,j. At the second step, following the above mentioned shortest paths, agents will be at states c_i and either x_i or $\neg x_i$ depending on the staging nodes they were occupying. The last step is the target configuration. Since the agents are connected at the second, it follows that for each clause c_j, the state corresponding to some literal of c_j is occupied by an agent. Thus, the valuation on variables encoded by the choices of the agents satisfies the **3-SAT** instance. □

6.3 Relaxation

For unbounded reachability and coverage, it seems tempting to take advantage of efficient algorithms on sight-moveable topological graphs (Propositions 3 and 4). In this subsection, we propose a transformation of a topological graph into a sight-moveable sub-graph. This transformation leads to a *relaxation* method: a solution found in the obtained sight-moveable topological graph still holds in the original graph.

This transformation requires the original graph to already satisfy properties 1 and 2 of Definition 3. If not, (1) we remove movement edges $(v, v') \in E_m$ that are not in E_c, (2) we remove all nodes without a self-loop.

Algorithm 1 Transformation into a sight-moveable topological graph

Require: A topological graph $G = \langle V, E_m, E_c \rangle$ satisfying properties 1 and 2 of Definition 3
1: $C \leftarrow \emptyset$
2: **for all** $v \in V$ **do**
3: $\quad Q \leftarrow \{v\}$
4: $\quad$ **while** Q not empty **do**
5: $\quad\quad v' \leftarrow Q.pop()$
6: $\quad\quad$ **for all** $v'' \mid (v', v'') \in E_m$ **do**
7: $\quad\quad\quad$ **if** $(v, v'') \in E_c$ **then**
8: $\quad\quad\quad\quad C \leftarrow C \cup \{(v, v'')\}$
9: $\quad\quad\quad\quad Q \leftarrow Q \cup \{v''\}$
10: $E'_c := \{(v, v') \mid (v, v') \in C \text{ and } (v', v) \in C\}$
11: **return** $\langle V, E_m, E'_c \rangle$

The simple transformation, given in Algorithm 1, prunes the communication edges which do not respect the property 3 of sight-moveable topological graphs, described in Definition 3. We now show this transformation outputs a sight-moveable topological graph.

Theorem 7 *Algorithm 1 constructs a sight-moveable topological graph.*

For simplicity, we say that a node v is sight-moveable to a node v' iff there exists a sequence $\rho = \langle \rho_1, \dots, \rho_n \rangle$ of nodes such that $v = \rho_1$, $v' = \rho_n$, $(v, \rho_i) \in E_c$ and $(\rho_i, \rho_{i+1}) \in E_m$ for all $i \in \{1, \dots, n-1\}$. Algorithm 1 consists in running a breadth-first search from each vertex, and marking all reachable vertices v' to which v is sight-moveable. The communication edges are kept if the sight-moveable property was shown to hold in both directions.

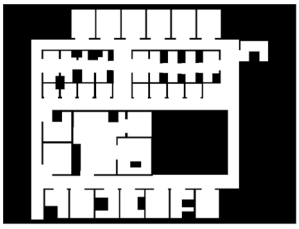
Proof First, given a node v, we show that the queue Q contains only nodes that v is sight-moveable to.

Invariant Given v, let I_v: for all $v' \in Q$, v is sight-moveable to v'. □

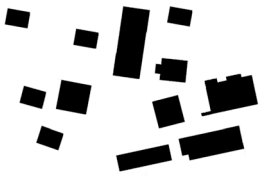
Proof Before entering the while loop, at Line 3, the queue Q contains only v. Hence, the Invariant I_v is initially satisfied. Let us suppose that Invariant I_v holds after an arbitrary number of loop iterations. If Q is empty, Invariant I_v holds. Otherwise, a node v' is popped out of Q. If a successor of v' communicates with v then we add the successors of v' to Q. Invariant I_v holds since v is sight-moveable to v', by induction, and $(v, v'') \in C$. □

We add a pair (v, v') to C iff v' was in Q and $(v, v') \in E_c$. Hence, given Invariant I_v, $(v, v') \in C$ iff v is sight-moveable to v'. Finally, E'_c is the set of pairs (v, v') such that

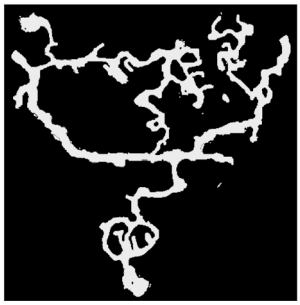
Fig. 10 Maps



(a) Office



(b) Open



(c) Coast

Map	Original Graph			Relaxation
	$\|V\|$	$\|E_m\|$	$\|E_c\|$	$\|E_c\|$
Office	1669	5618	277059	227155
Open	2421	9114	295155	294503
Coast	5184	20448	580644	580644

Fig. 11 Characteristics of the discretized graphs and their relaxations

$(v, v') \in C$ and $(v', v) \in C$, that is v is sight-moveable to v' and v' is sight-moveable to v. Therefore, the graph returned by Algorithm 1 is sight-moveable. □

This relaxation offers the opportunity to verify efficiently if a topological graph contains a solution. It is worth noting that the original topological graph might admit a solution, while no solution is found by the relaxation.

We illustrate the usage of the relaxation on the following maps, depicted in Fig. 10, previously used by Tateo et al., in [54], for the study of their algorithms. We use a similar discretization of the maps, i.e. the nodes of the graph are obtained by cells of 11×11 pixels for Office and 13×13 for Open, and we assume the communication range to be of 100 pixels. In addition, we use the same discretization with cells of 15×15 pixels for Coast, a map from the Benchmarks for Grid-Based Pathfinding [53], with an identical communication range.

We show, in Fig. 11, the size of each component of the graphs obtained after discretization and the impact of the relaxation. The relaxation of Office removes up to 18% of the communication edges. We can observe that most of the communication trough the walls, allowed by the communication range, will be removed in the relaxation since the

sight-moveable property cannot be ensured. In particular the large room in the center loses a lot of communication edges with its surrounding. However, the relaxation of Open removes only 0.2% of the communication due to its small number of obstacles. Finally, the relaxation does not remove any communication edges from the discretization made of Coast.

Intuitively, the relaxation removes communication links between two vertices which cannot be connected by maintaining communication. In fact, if a node communicates through an obstacle, then the sight-moveable property requires it to have a bypass path that remains connected. This is why, in the Office map, in particular at the borders of the central room, most communication links through walls are removed by the algorithm. But if the size of the obstacles is larger than the communication range, then there will be mostly no communication through obstacles, and it is unlikely that the relaxation will remove any communication edges: this can be observed on the Coast map where no edge is removed by the algorithm.

Thus, in open maps such as cities, forests etc., we expect that our relaxation removes a small number of edges and preserves the feasibility of our problems. It might however remove more communication edges, rendering the problems infeasible in indoor applications (inside of buildings, mazes, etc.). We leave the empirical evaluation of this relaxation for future work.

7 Complete-communication topological graphs

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

Proposition 5 $\mathbf{bReachability}_{\mathrm{cc}}$ *is in NLOGSPACE.*

Proof We refer to Lemma 2. 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 $\mathbf{bReachability}_{\mathrm{cc}}$. □

Our NP lower bound proof of the $\mathbf{bCoverage}_{\mathrm{cc}}$ problem is by reduction from the grid Hamiltonian cycle ($\mathbf{G-HC}$) problem which is the Hamiltonian cycle problem restricted to grid graphs and is NP-complete [27]. We use this particular version of the Hamiltonian cycle problem for simplicity of the proof. Futhermore, we obtain a lower bound on the $\mathbf{bCoverage}_{\mathrm{cc}}$ problem with a grid movement graph.

Theorem 8 *Even restricted to grid graphs,* $\mathbf{bCoverage}_{\mathrm{cc}}$ *is NP-complete for a fixed number of agents* $n \geq 1$.

Proof The upper bound follows from Proposition 2.

We give a polynomial-time reduction from the $\mathbf{G-HC}$ problem. Consider an instance of $\mathbf{G-HC}$ denoted $G = \langle V, E \rangle$.

Consider the sight-moveable topological graph $G' = \langle V, E_m, E_c \rangle$ with undirected movement $E_m = E$ and $E_c = V \times V$ and associate a single agent and the bound $|V|$ to the

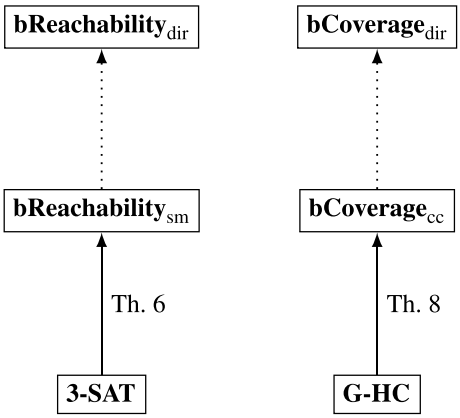


Fig. 12 Reductions of Sects. 6 and 7

bCoverage$_{cc}$ instance. We call a simple cycle containing all vertices a *tour*. We prove that there exists a tour t in G iff there exists a covering execution of length $|V|$ in G'.

($\Rightarrow$) Any tour of G is a valid execution satisfying **bCoverage**$_{cc}$ since the communication edges form a complete graph, and the bound is $|V|$.

($\Leftarrow$) Let us suppose that we have an execution of length $|V|$ which covers the graph G'. The execution starts and ends at B and visits all nodes in $|V|$ steps. Hence, the execution visits all nodes only once and is a cycle in the graph. □

In Fig. 12, we depicted the results obtained in this section and in Sect. 7.

8 Variants

In this section, we introduce several variants and study their impact on the complexity.

8.1 Bounded reachability and coverage with binary bounds

Our membership NP proofs for the bounded versions of the reachability and coverage problems rely on the unary encoding of the bound given in input. It is relevant to investigate the impact of providing that bound in binary on the complexity of the problems.

We show that both problems **bReachability**$_{dir}$ and **bReachability**$_{und}$ with the bound ℓ written in binary are both PSPACE-complete. The membership to PSPACE can be shown as in Proposition 1: since a binary counter can be added to count up to ℓ. For **bReachability**$_{und}$, the PSPACE-hardness follows from a reduction from **Reachability**$_{und}$, whose PSPACE-hardness is established in Theorem 3. Indeed, any instance of **Reachability**$_{und}$ can be reduced to bounded reachability with bound $\ell = |V|^n$. In fact, if **Reachability**$_{und}$ has a solution, then there is a plan of length at most $|V|^n$. Indeed, in the worst case, the agents must go through all configurations possible in the graph. This number can be computed in polynomial time and represented in binary. The same argument is used to show the PSPACE-hardness of **bReachability**$_{dir}$ by reduction from **Reachability**$_{dir}$ whose PSPACE-hardness was proved in [54] (see Theorem 2).

Theorem 9 *The variants of* $\textbf{bReachability}_{\text{dir}}$ *and* $\textbf{bReachability}_{\text{und}}$ *with* ℓ *written in binary are PSPACE-complete.*

A similar argument can be used to show the PSPACE-hardness of $\textbf{bCoverage}_{\text{und}}$ and $\textbf{bCoverage}_{\text{dir}}$. In fact, if there is a covering execution, then there must be one of length at most $|V|^n \times (|V| + 1)$. Indeed, the agents must traverse at most the $|V|^n$ possible configurations to reach each node of the graph and go back to the base, which means an upper bound of $|V|^n \times (|V| + 1)$ for the whole execution. This bound can be written in binary in polynomial space, so PSPACE-completeness also holds for these problems.

Theorem 10 *The variants of* $\textbf{bCoverage}_{\text{dir}}$ *and* $\textbf{bCoverage}_{\text{und}}$ *with* ℓ *written in binary are PSPACE-complete.*

8.2 Weighted movement graph

An interesting extension is obtained by assigning costs to edges of the movement graph, and considering the bounded reachability problem with respect to the total cost of the execution.

Consider a *weighted* topological graph $G = \langle V, E_m, E_c, cost \rangle$, where for each edge $cost(e)$ is the cost of e, a positive integer.

One could consider several ways of aggregating the weights along executions. We consider the case where the weights correspond to travel times between vertices, and the goal is to minimize total travel time, which also implies minimizing battery usage in drone applications. We consider a synchronous setting where all agents synchronize at each configuration in the plan; therefore, the travel time between two configurations is the maximum of the weights of the edges along which agents travel. In other terms, we assume that agents wait for each other at each step of the execution.

Formally, for an execution $c_0, c_1, \ldots, c_k$, the total cost is defined as follows:

$$\sum_{i=0}^{k-1} \max_{j \in \{1,2,\ldots,n\}} cost(c_i[j], c_{i+1}[j]), \tag{1}$$

where $c_i[j]$ denotes the vertex occupied by agent j at configuration c_i. Assuming binary encoding of all weights, we are interested in checking the existence of an execution of bounded cost, and that of a covering execution of bounded cost.

When weights are encoded in unary, both reachability and coverage problems are NP-hard since when all costs are equal to 1, then the problems are identical to **bReachability** and **bCoverage**, respectively. Since all weights are natural numbers, the length of the execution is not more than the cost bound given in input. Thus, similarly to Proposition 2, one can guess an execution and bound its length with a polynomial number of guesses.

Theorem 11 *The weighted variants of* **bReachability**, **bCoverage** *on directed and undirected topological graphs with unary encoding are NP-complete.*

When the weights are encoded in binary, then the problems are PSPACE-hard as shown in Sect. 8.1; and the PSPACE algorithms can be extended to establish PSPACE-completeness.

Theorem 12 *The weighted variants of* **bReachability**, **bCoverage** *on directed and undirected topological graphs with binary encoding are PSPACE-complete*.

8.3 Bounded disconnection

Another extension consists in allowing the agents to be disconnected for a bounded number of steps along the execution. Such a feature can be desirable in order to allow the agents to reach difficult locations where communication cannot be guaranteed. By bounding the disconnection time, the connection with the base is only disturbed temporarily.

This extension is also PSPACE-complete. In fact, our membership results can easily be extended both for reachability and coverage problems. For PSPACE-hardness, observe that when the allowed bound is 0, the problems become identical to our setting, so all hardness proofs carry over.

8.4 Collisions

Collision constraints consist in disallowing several agents to share the same location at a given time, and provide an interesting extension of our setting. In our case, we allow several agents to be at the base but not at other nodes.

The PSPACE-completeness results hold by using the same reductions used for the proof of Theorems 3 and 4 from the collision-free variant problem of Tateo et al. [54]. Indeed, the reductions gadgets used are collision-free, and thus we obtain:

Theorem 13 *The variants of* $\textbf{Reachability}_{\text{und}}$, $\textbf{Reachability}_{\text{dir}}$, $\textbf{Coverage}_{\text{und}}$ *and* $\textbf{Coverage}_{\text{dir}}$ *with collision constraints are PSPACE-complete*.

We can show that when the topological graph is sight-moveable, then the complexity of deciding the reachability and coverage without allowing collisions is unchanged. We use the same membership proof of Propositions 3 and 4. Indeed, a positive instance of $\textbf{Reachability}_{\text{sm}}$ (resp. $\textbf{Coverage}_{\text{sm}}$) is a positive instance of $\textbf{Reachability}_{\text{sm}}$ (resp. $\textbf{Coverage}_{\text{sm}}$) with collision constraints. In order to obtain a collision-free execution, it suffices to modify as follows: we consider an ordering, as depicted in Fig. 7, and consider each branch one by one. For each branch, we dispatch first the agent with the furthest target from the base, followed by the second furthest agent and so on. This yields a collision-free execution for $\textbf{Reachability}_{\text{sm}}$. $\textbf{Coverage}_{\text{sm}}$ can be solved by a repeated application of $\textbf{Reachability}_{\text{sm}}$. Observe that $\textbf{Reachability}_{\text{cc}}$ and $\textbf{Coverage}_{\text{cc}}$ can use the same algorithms.

Theorem 14 *The variants of* $\textbf{Reachability}_{\text{sm}}$, $\textbf{Reachability}_{\text{cc}}$ (*resp*. $\textbf{Coverage}_{\text{sm}}$ *and* $\textbf{Coverage}_{\text{cc}}$) *with collision constraints are in LOGSPACE* (*resp*. *NLOGSPACE*).

Regarding the bounded versions, the membership of Proposition 2 hold without allowing collisions. Indeed, checking if the path contains a collision can be done in polynomial-time. The lower bound can be showed by reduction from the MAPF problem to

bReachability$_{\text{cc}}$. Additionally, the proof of Theorem 8 holds without allowing collisions given that the proof holds for a single agent.

Theorem 15 *The variants of* **bReachability**$_\star$ *and* **bCoverage**$_\star$ *without collisions are NP-complete for all* $\star \in \{\text{und}, \text{dir}, \text{sm}, \text{cc}\}$.

8.5 Planar movement graphs

One could wonder whether the complexity of these problems change when restricted to topological graphs whose movement graphs are planar. This is an interesting question since in targeted applications the graphs are indeed planar; and some problems are known to become easier on planar graphs (e.g. the shortest path computation [30]). In this section, we show that our reductions proving the PSPACE-hardness of **Reachability**$_{\text{und}}$, **Coverage-init**$_{\text{und}}$, **Coverage**$_{\text{und}}$ and **Reachability**$_{\text{sm}}$ use planar graphs, which means that these hardness results hold even when the input is restricted to planar graphs.

Actually, the proof of **Reachability-init**$_{\text{und}}$, in [54], uses a planar movement graph. They even prove a stronger result: the reachability problem is PSPACE-hard even for graphs that are disjoint unions of *paths*.[2]

Let us consider an instance $(G_{Reach_{init}}, B, s, t)$ of **Reachability-init**$_{\text{und}}$ where $G_{Reach_{init}}$ is a topological graph that is a disjoint union of paths (depicted in the middle of Fig. 13), B is its base, $s = (s_1, \ldots, s_n)$ is initial configuration for n agents, depicted at the bottom of $G_{Reach_{init}}$ in Fig. 13 and $t = (t_1, \ldots, t_n)$ is the target configuration (the target nodes $t_1, \ldots, t_n$ are in $G_{Reach_{init}}$ as defined in [54] but are not displayed in the figure).

Then as shown in Fig. 13, the gadgets we proposed in this article can all be arranged to obtain planar graphs as well.

1. The gadget given in Fig. 4, used in the proof of Theorem 3 is a planar graph; in fact, it can be arranged, to obtain G_{Reach} (in Fig. 13) whose base is B' and is designed for $n + 2$ agents. This proves that **Reachability**$_{\text{und}}$ is PSPACE-hard on planar graphs.
2. We consider now the reduction of Fig. 5 used in proof of Lemma 1, which contains G_{Reach}. This reduction, named $G_{\text{Cover}_{\text{init}}}$ in Fig. 13 is also planar. The base of this graph is still B', and it is designed for $n + 2 + n + 2 = 2n + 4$ agents. In this reduction, the first $n + 2$ agents start in B' while the others start in $v^1_1, \ldots, v^1_{n+2}$. Indeed, the latter gadget is disconnected from the rest of the graph, except that the node v^4_1 has a movement edge to all nodes in the graph. One can observe that a movement edge can be created from v^4_1 to all nodes in the graph G_{Reach}, while keeping the construction planar, thanks to the particular structure of G_{Reach}. This shows that **Coverage-init**$_{\text{und}}$ is PSPACE-hard on planar graphs.
3. Finally, we can add the gadget of Fig. 4 used in proof of Theorem 4, shown in the top part of Fig. 13. We thus obtain G_{Cover} designed for $2n + 4 + 2 = 2n + 6$ agents starting

[2] Their reduction is from a problem called *the reconfiguration problem on constraint graphs*. The topological graph computed from a constraint graph contains single nodes (( )), paths of length 3 (which we denote by) and a single path graph of length $|E|$ (denoted by), with E the set of edges in the constraint graph.

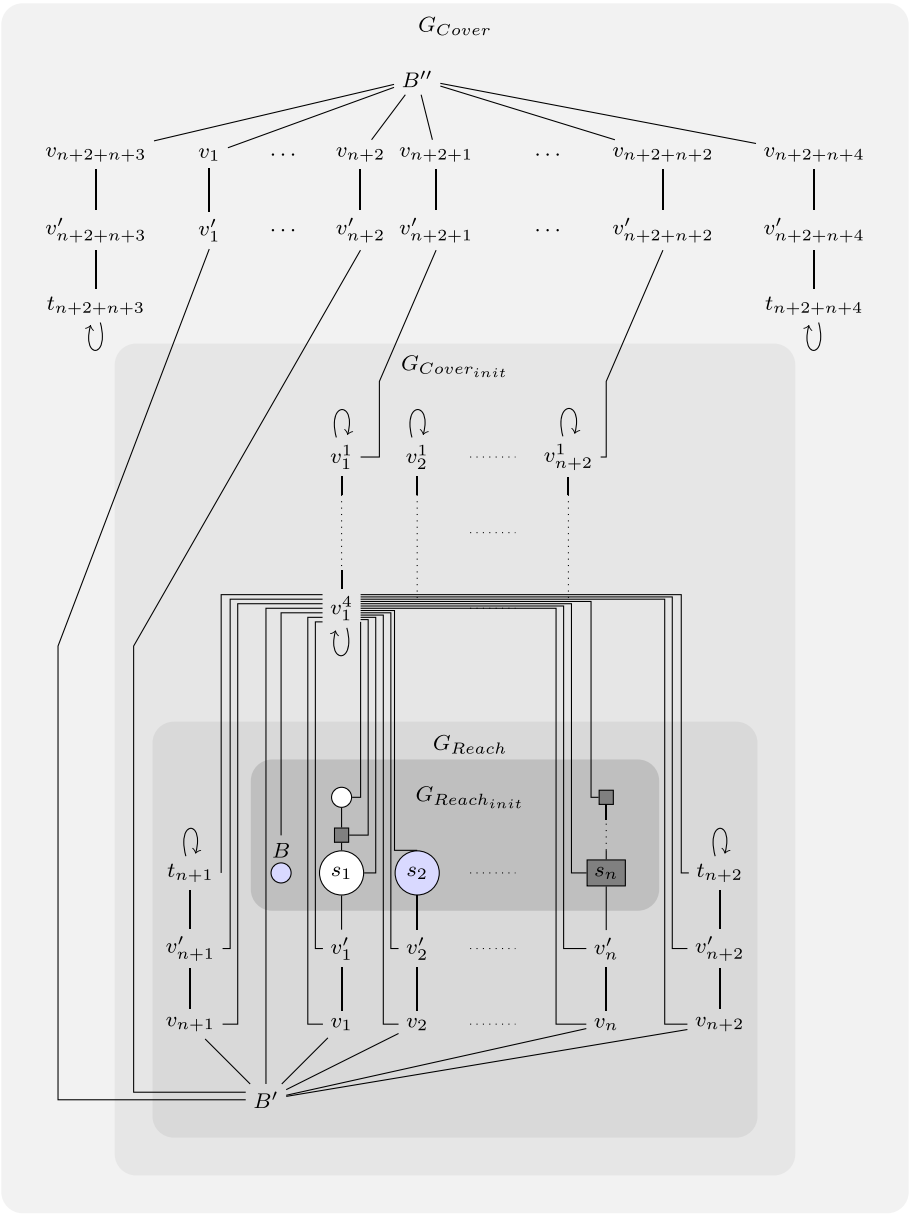


Fig. 13 Summary of all the planar constructions

at the base B'', which is the reduction of Theorem 4. The obtained graph is also planar. This shows that **Coverage**$_{\text{und}}$ is PSPACE-hard on planar graphs.

To sum up:

Theorem 16 **Reachability**$_{\text{und}}$, **Coverage-init**$_{\text{und}}$ *and* **Coverage**$_{\text{und}}$ *are PSPACE-hard on planar graphs*.

9 Conclusion

The main result of this paper is the introduction of sight-moveable topological graphs. Indeed, being able to decide in NLOGSPACE whether the reachability or the coverage can be done is an important improvement over the previous results. However, this class does not yield an improvement for the optimization of the execution.

We have studied numerous variants of the defined problems in the hope to give a complete overview of this new extension of the MAPF problem. Unfortunately, we still lack results for some of these variants. Furthermore, it is also unclear whether the combination of these variants can yield higher complexity. Additionally, there are still many variants of MAPF that have not been extended to these problems.

In realistic situation, the topological graph may not be fully known in advance. Indeed, the discretization may not represent faithfully the area in which the agents evolve. Thus, we attend to extend this work to incomplete knowledge in which the agents only have a over-approximation of the actual topological graph. While the agents move around the area they can observe the graph and update their knowledge.

In addition, if we consider the variants with bounded disconnection, introduced in Sect. 8.3, with incomplete knowledge we fall in the setting of imperfect information. In this setting, an agent can fail to reconnect during the mission and the cooperation needs to consider the possible cause of this disconnection to finish the plan.

Finally, not only can the area be partially known by the agents but some mishaps can happen during the mission. Indeed, an external actor might disable an agent and, in this setting, the group of agent should have to find the agent or finish the mission without it. In the weighted graph extension (Sect. 8.2), mishaps can also be due to strong wind which modifies the cost of the movement.

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. We also thank Sophie Pinchinat for useful comments.

References

1. Ahmadzadeh, A., Keller, J., Pappas, G., Jadbabaie, A., & Kumar, V. (2008). An optimization-based approach to time-critical cooperative surveillance and coverage with UAVs. In O. Khatib, V. Kumar, & D. Rus (Eds.), *Experimental Robotics: The 10th International Symposium on Experimental Robotics*, pp. 491–500). Berlin, Heidelberg: Springer.
2. Amigoni, F., Banfi, J., & Basilico, N. (2017). Multirobot exploration of communication-restricted environments: A survey. *IEEE Intelligent Systems*, *32*(6), 48–57.
3. Anbuudayasankar, S. P., Ganesh, K., & Mohapatra, S. (2016). *Models for practical routing problems in logistics*. New York: Springer.
4. Aurenhammer, F. (1991). Voronoi diagrams—A survey of a fundamental geometric data structure. *ACM Computing Surveys*, *23*(3), 345–405.
5. Banfi, J., Basilico, N., & Amigoni, F. (2017). Intractability of time-optimal multirobot path planning on 2D grid graphs with holes. *IEEE Robotics and Automation Letters*, *2*(4), 1941–1947.

6. Banfi, J., Li, A.Q., Basilico, N., Rekleitis, I., & Amigoni, F. (2016, May). Asynchronous multirobot exploration under recurrent connectivity constraints. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5491–5498.
7. Bodin, F., Charrier, T., Queffelec, A., & Schwarzentruber, F. (2018). Generating plans for cooperative connected uavs. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13–19, 2018, Stockholm, Sweden*, pp. 5811–5813.
8. Bylander, T. (1994). The computational complexity of propositional STRIPS planning. *Artificial Intelligence*, *69*(1–2), 165–204.
9. Cabreira, T. M., Brisolara, L. B., & Ferreira, P. R, Jr. (2019). Survey on coverage path planning with unmanned aerial vehicles. *Drones*, *3*(1), 4.
10. Cesare, K., Skeele, R., Yoo, Soo-Hyun, Zhang, Yawei, & Hollinger, G. (2015, May). Multi-UAV exploration with limited communication and battery. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2230–2235.
11. Charrier, T., Queffelec, A., Sankur, O., & Schwarzentruber, F. (2019). Reachability and coverage planning for connected agents. In *Proceedings of AAMAS, Montreal, QC, Canada, May 13-17, 2019*, pp. 1874–1876.
12. Charrier, T., Queffelec, A., Sankur, O., & Schwarzentruber, F. (2019). Reachability and coverage planning for connected agents. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10–16, 2019*, pp. 144–150.
13. Chen, Y., Zhang, H., & Xu M. (2014, July). The coverage problem in UAV network: A survey. In *Proceedings of the Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*.
14. Choset, H. (2001). Coverage for robotics—A survey of recent results. *Annals of Mathematics and Artificial Intelligence*, *31*(1–4), 113–126.
15. Choset, H., & Pignon, P. (1998). Coverage path planning: The boustrophedon cellular decomposition. In A. Zelinsky, editor, *Field and Service Robotics*, pp. 203–209. London: Springer.
16. Cook, S. A. (1985). A taxonomy of problems with fast parallel algorithms. *Information and Control*, *64*(1–3), 2–21.
17. Danner, T., & Kavraki, L.E. (2000, April). Randomized planning for short inspection paths. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2, pp. 971–976.
18. Englot, B., & Hover, F. (2012). Sampling-based coverage path planning for inspection of complex structures. In *International Conference on Automated Planning and Scheduling*.
19. Englot, B., & Hover, F. (2017). Planning complex inspection tasks using redundant roadmaps. In H. I. Christensen & O. Khatib (Eds.), *Robotics Research: The 15th International Symposium ISRR* (pp. 327–343). Cham: Springer International Publishing.
20. Finkel, R. A., & Bentley, J. L. (1974). Quad trees: A data structure for retrieval on composite keys. *Acta Informatica*, *4*, 1–9.
21. Francès, G., Ramírez, M., Lipovetzky, N., & Geffner, H. (2017). Purely declarative action descriptions are overrated: Classical planning with simulators. In *Twenty-Sixth International Joint Conference on Artificial Intelligence*.
22. Fu, M., Kuntz, A., Salzman, O., & Alterovitz, R. (June 2019). Toward asymptotically-optimal inspection planning via efficient near-optimal graph search. In *Proceedings of Robotics: Science and Systems*, FreiburgimBreisgau, Germany.
23. Galceran, E., & Carreras, M. (2013). A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, *61*(12), 1258–1276.
24. Hazon, N., & Kaminka, G.A. (2005, April). Redundancy, efficiency and robustness in multi-robot coverage. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 735–741.
25. Hearn, R. A., & Demaine, E. D. (2005). Pspace-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *Theoretical Computer Science*, *343*(1), 72–96.
26. Hollinger, G. A., & Singh, S. (2012). Multirobot coordination with periodic connectivity: Theory and experiments. *IEEE Transactions of Robotics*, *28*(4), 967–973.
27. Itai, A., Papadimitriou, C. H., & Szwarcfiter, J. L. (1982). Hamilton paths in grid graphs. *SIAM Journal on Computing*, *11*(4), 676–686.
28. Karp, R.M. (1972). Reducibility among combinatorial problems. In *Proceedings of a symposium on the Complexity of Computer Computations,*.

29. Kavraki, L.E., Kolountzakis, M.N., & Latombe, J. (1996, April). Analysis of probabilistic roadmaps for path planning. In *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3020–3025.
30. Klein, P. N., Mozes, S., & Weimann, O. (2010). Shortest paths in directed planar graphs with negative lengths: A linear-space o(n log2 n)-time algorithm. *ACM Transactions on Algorithms*, *6*(2), 30:1–30:18.
31. Knoll, A. (2006). A survey of octree volume rendering methods. In *GI, the Gesellschaft für Informatik*, p. 87.
32. Kuffner, J.J., & LaValle, S.M. (2000, April). Rrt-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2, pp. 995–1001.
33. Kusnur, T., Mukherjee, S., Saxena, D.M., Fukami, T., Koyama, T., Salzman, O., & Likhachev, M. (2019). A planning framework for persistent, multi-UAV coverage with global deconfliction. *CoRR*, arXiv:abs/1908.09236.
34. LaValle, S. M. (2006). *Planning algorithms*. Cambridge: Cambridge University Press.
35. Ma, H., Tovey, C.A., Sharon, G., Kumar, T.K.S., & Koenig, S. (2016). Multi-agent path finding with payload transfers and the package-exchange robot-routing problem. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12–17, 2016, Phoenix, Arizona, USA*, pp. 3166–3173.
36. Morris, R., Pasareanu, C.S., Luckow, K.S., Malik, W., Ma, H., Kumar, T.K.S., & Koenig, S. (2016). Planning, scheduling and monitoring for airport surface operations. In *Proceedings of the Planning for Hybrid Systems, Papers from the 2016 AAAI Workshop, Phoenix, AZ, USA, February 13, 2016*.
37. Nestmeyer, T., Giordano, P. R., Bülthoff, H. H., & Franchi, A. (2017). Decentralized simultaneous multi-target exploration using a connected network of multiple robots. *Auton. Robots*, *41*(4), 989–1011.
38. Otte, M., & Correll, N. (2014). Any-com multi-robot path-planning with dynamic teams: Multi-robot coordination under communication constraints. In O. Khatib, V. Kumar, & G. Sukhatme (Eds.), *Proceedings of the Experimental Robotics: The 12th International Symposium on Experimental Robotics*, pp. 743–757, Berlin, Heidelberg: Springer.
39. Pandey, R., Singh, A.K., & Krishna, K.M. (2012, August). Multi-robot exploration with communication requirement to a moving base station. In *Proceedings of the 2012 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 823–828.
40. Pham, H.X., La, H.M., Feil-Seifer, D., & Deans, M.C. (2018). A distributed control framework of multiple unmanned aerial vehicles for dynamic wildfire tracking. *Proceedings of the IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–12.
41. Ratner, D., & Warmuth, M.K. (1986). Finding a shortest solution for the N $\times $ N extension of the 15-puzzle is intractable. In *Proceedings of the 5th National Conference on Artificial Intelligence. Philadelphia, PA, USA, August 11–15, 1986. Volume 1: Science*, pp. 168–172.
42. Reingold, O. (2008). Undirected connectivity in log-space. *Journal of ACM*, *55*(4), 17:1–17:24.
43. Rekleitis, I., New, A. P., Rankin, E. S., & Choset, H. (2008). Efficient boustrophedon multi-robot coverage: An algorithmic approach. *Annals of Mathematics and Artificial Intelligence*, *52*(2), 109–142.
44. Rooker, M. N., & Birk, A. (2007). Multi-robot exploration under the constraints of wireless networking. *Control Engineering Practice*, *15*(4), 435–445.
45. Ryan, M. R. K. (2008). Exploiting subgraph structure in multirobot path planning. *Journal of Artificial Intelligence Research*, *31*, 497–542.
46. Savitch, W. J. (1970). Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*,. https://doi.org/10.1016/S0022-0000(70)80006-X.
47. Schwartz, J. T., & Sharir, M. (1983). On the piano movers' problem: III. Coordinating the motion of several independent bodies: The special case of circular bodies moving amidst polygonal barriers. *The International Journal of Robotics Research*, *2*(3), 46–75.
48. Sharon, G., Stern, R., Felner, A., & Sturtevant, N. (2012). Conflict-based search for optimal multi-agent path finding. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, pp. 563–569. AAAI Press.
49. Silver, D. (2005). Cooperative pathfinding. In *Proceedings of the First AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, AIIDE'05, pp. 117–122. AAAI Press.
50. Sipser, M. (1997). *Introduction to the theory of computation*. Boston: PWS Publishing Company.
51. Solovey, K., & Halperin, D. (2016). On the hardness of unlabeled multi-robot motion planning. *The International Journal of Robotics Research*, *35*(14), 1750–1759.
52. Standley, T. (2010). Finding optimal solutions to cooperative pathfinding problems. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, pp. 173–178. AAAI Press.

53. Sturtevant, N. (2012). Benchmarks for grid-based pathfinding. *Transactions on Computational Intelligence and AI in Games*, *4*(2), 144–148.
54. Tateo, D., Banfi, J., Riva, A., Amigoni, F., & Bonarini, A. (2018). Multiagent connected path planning: Pspace-completeness and how to deal with it. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
55. Teacy, W.L., Nie, J., McClean, S., & Parr, G. (2010). Maintaining connectivity in UAV swarm sensing. In *2010 IEEE Globecom Workshops*, pp. 1771–1776. IEEE.
56. Turner, H. (2002). Polynomial-length planning spans the polynomial hierarchy. In *Proceedings of the Logics in Artificial Intelligence, European Conference, JELIA 2002, Cosenza, Italy, September, 23–26*.
57. Veloso, M., Biswas, J., Coltin, B., & Rosenthal, S. (2015). Cobots: Robust symbiotic autonomous mobile service robots. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pp. 4423–4429. AAAI Press.
58. Wurman, P.R., D'Andrea, R., & Mountz, M. (2007). Coordinating hundreds of cooperative, autonomous vehicles in warehouses. In *Proceedings of the 19th National Conference on Innovative Applications of Artificial Intelligence—Volume 2*, IAAI'07, pp. 1752–1759. AAAI Press.
59. Xu, L. (August 2011). *Graph Planning for Environmental Coverage*. Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA.
60. Yanmaz, E. (2012). Connectivity versus area coverage in unmanned aerial vehicle networks. In *Proceedings of IEEE International Conference on Communications, ICC 2012*.
61. Yu, J. (2016). Intractability of optimal multirobot path planning on planar graphs. *IEEE Robotics and Automation Letters*, *1*(1), 33–40.
62. Yu, J., & LaValle, S. (2012). Planning optimal paths for multiple robots on graphs. *Proceedings of the IEEE International Conference on Robotics and Automation*.
63. Yu, J., & LaValle, S.M. (2013). Multi-agent path planning and network flow. In E. Frazzoli, T. Lozano-Perez, N. Roy, & D. Rus (Eds.), *Proceedings of the Algorithmic Foundations of Robotics X*, pp. 157–173, Berlin, Heidelberg: Springer.
64. Yu, J., & Rus, D. (2014). Pebble motion on graphs with rotations: Efficient feasibility tests and planning algorithms. In H. L. Akin, N. M. Amato, V. Isler, & A. F. van der Stappen (Eds.), *Proceedings of the Algorithmic Foundations of Robotics XI—Selected Contributions of the Eleventh International Workshop on the Algorithmic Foundations of Robotics, WAFR 2014, 3–5 August 2014, Boğaziçi University, Istanbul, Turkey*, volume 107 of *Springer Tracts in Advanced Robotics*, pp. 729–746. New York: Springer.
65. Zheng, X., Sonal, J., Koenig, S., & Kempe, D. (2005, August). Multi-robot forest coverage. In *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3852–3857.