Reachability Games in Dynamic Epistemic Logic

Bastien Maubert
Università degli Studi di Napoli "Federico II", Italy

Sophie Pinchinat
Université de Rennes, France

François Schwarzentruber
ENS Rennes, France

Abstract

We define reachability games based on Dynamic Epistemic Logic (DEL), where the players' actions are finely described as DEL action models. We first consider the setting where an external controller with perfect information interacts with an environment and aims at reaching some epistemic goal state regarding the passive agents of the system. We study the problem of strategy existence for the controller, which generalises the classic epistemic planning problem, and we solve it for several types of actions such as public announcements and public actions. We then consider a yet richer setting where agents themselves are players, whose strategies must be based on their observations. We establish several (un)decidability results for the problem of existence of a distributed strategy, depending on the type of actions the players can use, and relate them to results from the literature on multiplayer games with imperfect information.

2012 ACM Subject Classification Theory of computation → Algorithmic game theory; Computing methodologies → Reasoning about belief and knowledge; Computing methodologies → Planning under uncertainty; Computing methodologies → Multi-agent planning; Computing methodologies → Multi-agent systems

Keywords and phrases Dynamic Epistemic Logic, game theory, imperfect information, planning

1 Introduction

Many applications fall within the scope of reachability games with imperfect information, such as video games [15] (Civilization, etc.), *Kriegspiel* (the epistemic variant of Chess) [25], Hanabi [3], or contingent and conformant planning [20].

Games with imperfect information are computationally hard, and even undecidable for multiple players [29]. One way to tame this complexity is to make assumptions on how the knowledge of the different players compare: if all players that cooperate can be ordered in a hierarchy where one knows more than the next, a situation called *hierarchical information*, then the existence of distributed strategies can be decided [28, 7]. Another natural approach is to consider fragments based on classes of action types, as done for instance in [32, 6, 11] where different kinds of public actions are considered. But the usual graph-based models of games with imperfect information, where the players' actions are modelled as labels on the edges, make it difficult to define subtle properties of actions.

	Public announcements	Public actions	Propositional actions	Full
Plan	NP-c	PSPACE-c	decidable	undecidable
Controller	PSPACE-c (Th. 12)	EXPTIME-c (Th. 13)	decidable (Th. 14)	undecidable
Distributed strategy	PSPACE-c (Th. 21)	EXPTIME-c (Th. 22)	undecidable (Th. 20) decidable case (Th. 23)	undecidable

Table 1 Known results (new in grey) for plan, controller and distributed strategy synthesis.

By contrast, Dynamic epistemic logic (DEL) [38] was designed to describe actions precisely: how they affect the world and how they are perceived. In particular, classic action types such as public/private announcements or public actions correspond to natural classes of DEL *action models*. Also, DEL extends epistemic logic and hence enables modelling *higher-order knowledge*, i.e., what an agent knows about what another agent knows etc, and the evolution of agents' knowledge over time.

A classification of the complexity with respect to action types was addressed in the literature of *epistemic planning*, a problem that asks for the existence of a *plan*, i.e., a finite sequence of DEL actions to reach a situation that satisfies some given objective expressed in epistemic logic. However this problem, which can be seen as solving one-player reachability games with epistemic objective, has never been considered in a strategic, adversarial context. This work bridges the gap between DEL and games by introducing adversarial aspects in DEL planning, thus moving from plan generation to strategy synthesis. We define two frameworks for DEL-based reachability games, where players start in a given epistemic situation and their possible moves are described by action models, and the objective is to reach a situation satisfing some epistemic formula.

In a first step we consider *open systems* [21], i.e., systems that interact with an environment. In our setting, two omniscient, external entities (that we call *controller* and *environment*) choose in turn which actions are performed. We call this setting *DEL controller synthesis*. Here, agents involved in the models and formulas are not active, they merely observe how the system evolves based on the actions chosen by the controller and the environment, and update their knowledge accordingly. DEL controller synthesis extends DEL planning, as the latter is a degenerate case of the former where the environment stays idle, and we therefore inherit undecidability for the general case. Nevertheless we show that, as for DEL planning, decidability is regained when actions do not increase uncertainty (so-called *non-expanding actions*) or when the preconditions of actions are propositional formulas. More precisely, we show Pspace-completeness when possible moves are public announcements, Exptime-completeness for the more general public actions, and membership in $(k+1)$-Exptime for propositional actions when the objectives are formulas of modal depth at most k.

We then generalise further this setting by turning agents into players. Unlike the omniscient controller of the former setting, agents have imperfect information about the current state of the game, and can only base their decisions on what they know. In the theory of games with imperfect information this is modelled by the notion of *uniform strategies*, also called *observation-based strategies* [1]. We study the problem of *distributed strategy synthesis*, where a group of players cooperate to enforce some objective against the remaining players. As for multi-player games with imperfect information the problem is undecidable, already for propositional actions and a coalition of two players. However we show that the two kinds of assumptions that make imperfect-information games decidable, namely public actions and hierarchical information, also yield decidable cases of multiplayer DEL games. Furthermore, in the case of public announcements and public actions, the complexity is not worse than for controller synthesis.

Related work

The complexity of DEL-based epistemic planning has been thoroughly investigated. It is undecidable already for actions with preconditions of modal depth one and propositional postconditions [9, 22]. For preconditions of modal depth one and no postconditions the problem has been open for years, but it is decidable when pre- and postconditions are propositional [39, 2, 16]. It is also known to be NP-complete for public announcements [10, 14], and PSPACE-complete for public actions [14].

The decidability for propositional actions has been extended in [2] by considering infinite trees of actions called *protocols* instead of finite plans, and specifications in branching-time epistemic temporal logic instead of reachability for epistemic formulas; this has been extended further in [16] by enriching the specification language with Chain Monadic Second-order Logic. Both results rely on the fact that when actions are propositional, the infinite structures generated by repeated application of action models form a class of *regular structures* [2, 26], i.e., relational structures that have a finite representation via automata. First-order logic is decidable on such structures [8], and chain-MSO is decidable on a subclass called *regular automatic trees* [16], but neither of these logics can express the existence of strategies in games. However we will show that the regular structures obtained from propositional DEL models can be seen as finite turn-based game arenas studied in games played on graphs. This allows us to transfer decidability results on games with epistemic temporal objectives to the DEL setting.

A notion of cooperative planning in DEL has been studied in [17], but without the adversarial aspect of games. Also, in [23], a game setting has been developed with the so-called Alternating-time Temporal Dynamic Epistemic Logic, but it does not consider uniform strategies and thus cannot express the existence of distributed strategies. Our controller synthesis problem can be expressed in this logic, but not in the fragment that they solve, which cannot express reachability. On the other hand, several decidability results for logics for strategic and epistemic reasoning have been established recently [5, 27], but they do not offer the fine modelling of actions possible in DEL. For instance they cannot easily model public announcements, which we show yield better complexity than those obtained in their settings.

Table 1 sums up previous results for epistemic planning, as well as the results established in this contribution.

2 Background in epistemic planning

Let us fix a countable set of *atomic propositions* AP.

2.1 The classic DEL setting

We recall models of epistemic logic [19].

▶ **Definition 1.** *An* epistemic model $\mathcal{M} = (W, (R_a)_{a\in Agt}, V)$ *is a tuple where*

- W *is a non-empty finite set of possible* worlds *(or situations),*
- $R_a \subseteq W \times W$ *is an* accessibility relation *for agent* a*, and*
- $V : W \to 2^{AP}$ *is a* valuation function*.*

We write wR_au instead of $(w, u) \in R_a$; its intended meaning is that when the actual world is w, agent a considers that u may be the actual world. The valuation function V provides the subset of atomic propositions that hold in a world. A pair $(\mathcal{M}, w)$ is called a *pointed epistemic model*, and we let $|\mathcal{M}|$ be the *size* of $\mathcal{M}$, defined as $|W| + \sum_{a\in Agt} |R_a| + \sum_{w\in W} |V(w)|$. We will only consider finite models, i.e., we assume that $V(w)$ is finite for all worlds.

The syntax of Epistemic Logic $\mathcal{L}_{\mathbf{EL}}$ is given by the following grammar:

$$\varphi \ ::= \ p \ \mid \ \neg\varphi \ \mid \ (\varphi \vee \varphi) \ \mid \ K_a\varphi$$

where p ranges over AP and a ranges over Agt.

$K_a\varphi$ is read 'agent a knows that φ is true'. We define the usual abbreviations $(\varphi_1 \wedge \varphi_2)$ for $\neg(\neg\varphi_1 \vee \neg\varphi_2)$ and $\hat{K}_a\varphi$ for $\neg K_a \neg\varphi$, and use $\mathcal{L}_{\text{Prop}}$ for the fragment of $\mathcal{L}_{\mathbf{EL}}$ with propositional formulas only. The *modal depth* of a formula is its maximal number of nested knowledge operators; for instance, the formula $K_a K_b p \wedge \neg K_a q$ has modal depth 2. The *size* $|\varphi|$ of a formula φ is the number of symbols in it.

The semantics of $\mathcal{L}_{\mathbf{EL}}$ relies on *pointed epistemic models*.

▶ **Definition 2.** *We define $\mathcal{M}, w \models \varphi$, read as 'formula φ holds in the pointed epistemic model $(\mathcal{M}, w)$', by induction on φ, as follows:*

- $\mathcal{M}, w \models p$ *if* $p \in V(w)$*;*
- $\mathcal{M}, w \models \neg\varphi$ *if it is not the case that* $\mathcal{M}, w \models \varphi$*;*
- $\mathcal{M}, w \models (\varphi \vee \psi)$ *if* $\mathcal{M}, w \models \varphi$ *or* $\mathcal{M}, w \models \psi$*;*
- $\mathcal{M}, w \models K_a\varphi$ *whenever for all u such that $wR_a u$,* $\mathcal{M}, u \models \varphi$*.*

Dynamic Epistemic Logic (DEL) relies on *action models* (also called "event models"). These models specify how agents perceive the occurrence of an action as well as its effects on the world.

▶ **Definition 3.** *An* action model $\mathcal{A} = (A, (R_a^{\mathcal{A}})_{a \in Agt}, pre, post)$ *is a tuple where:*

- *A is a non-empty finite set of possible* actions*,*
- $R_a^{\mathcal{A}} \subseteq A \times A$ *is the* accessibility relation *for agent a,*
- $pre : A \to \mathcal{L}_{\mathbf{EL}}$ *provides the* precondition *for an action to be performed, and*
- $post : A \times AP \to \mathcal{L}_{Prop}$ *provides the* postcondition *(i.e., the effects) of an action.*

A *pointed action model* is a pair $(\mathcal{A}, \alpha)$ where α represents the actual action. We let $|\mathcal{A}|$ be the *size* of $\mathcal{A}$, defined as $|\mathcal{A}| := |A| + \sum_{a \in Agt} |R_a^{\mathcal{A}}| + \sum_{\alpha \in A} |\text{pre}(\alpha)| + \sum_{\alpha \in A, p \in AP} |\text{post}(\alpha, p)|$.

An action α is *executable* in a world w of an epistemic model $\mathcal{M}$ if $\mathcal{M}, w \models \text{pre}(\alpha)$, and in that case we define $V(w, \alpha) := \{p \in AP \mid \mathcal{M}, w \models \text{post}(\alpha, p)\}$, the set of atomic propositions that hold after occurrence of action α in world w. Since postconditions are always propositional, we can define similarly $V(v, \alpha)$ where $v \subseteq 2^{AP}$ is a valuation.

Types of actions

We identify noticeable types of actions. An action model $\mathcal{A}$ is *propositional* if all pre- and postconditions of actions in $\mathcal{A}$ belong to $\mathcal{L}_{\text{Prop}}$. A *public action* is a pointed action model $\mathcal{A}, \alpha$ such that for each agent a, $R_a^{\mathcal{A}}$ is the identity relation. A *public announcement* is a public action $\mathcal{A}, \alpha$ such that for all p, $\text{post}(\alpha, p) = p$.

We recall the product that models how to update an epistemic model when an action is executed [4].

▶ **Definition 4** (Product [4])**.** *Let $\mathcal{M} = (W, (R_a)_{a \in Agt}, V)$ be an epistemic model, and $\mathcal{A} = (A, (R_a^{\mathcal{A}})_{a \in Agt}, pre, post)$ be an action model. The* product *of $\mathcal{M}$ and $\mathcal{A}$ is defined as $\mathcal{M} \otimes \mathcal{A} = (W', (R_a)', V')$ where:*

- $W' = \{(w, \alpha) \in W \times A \mid \mathcal{M}, w \models pre(\alpha)\}$,
- $(w, \alpha) R'_a (w', \alpha')$ *if* $wR_a w'$ *and* $\alpha R_a^{\mathcal{A}} \alpha'$*, and*
- $V'((w, \alpha)) = V(w, \alpha)$*.*

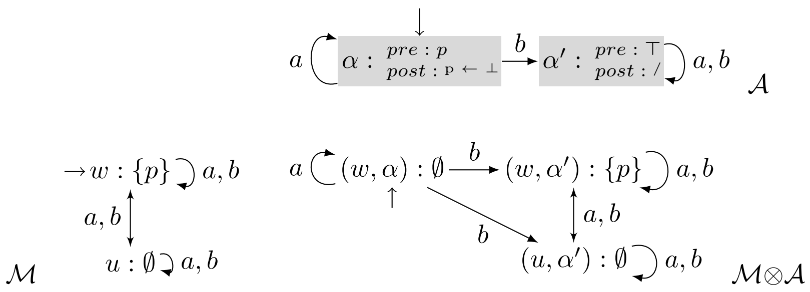

Figure 1 Example of DEL product. Symbol / indicates the trivial postcondition that leaves valuations unchanged.

▶ **Example 5.** Figure 1 shows the pointed model $\mathcal{M}, w$ that represents a situation in which p is true and both agents a and b do not know it. The pointed action model $\mathcal{A}, \alpha$ describes the action where agent a learns that p was true but that it is now set to false, while agent b does not learn anything (she sees action α' that has trivial pre- and postcondition). In the product epistemic model $\mathcal{M} \otimes \mathcal{A}, (w, \alpha)$, agent a now knows that p is false, while b still does not know the truth value of p, or whether agent a knows it.

An epistemic model (resp. an action model) is *S5* if all accessibility relations are equivalence relations. This property is important to model games with imperfect information, and we will assume it in Section 4.

2.2 Generated structure

Iteratively executing an action model from an initial epistemic model generates an infinite sequence of epistemic models, whose union yields an infinite epistemic structure where dynamics are represented by the possible sequences of actions, while information is captured by the accessibility relations.

▶ **Definition 6** (Generated structure)**.** *Given $\mathcal{M} = (W, \{R_a\}_{a\in Agt}, V)$ an epistemic model and $\mathcal{A} = (A, \{R_a^{\mathcal{A}}\}_{a\in Agt}, pre, post)$ an action model, we define the family of disjoint epistemic models $\{\mathcal{MA}^n\}_{n\geq 0}$ by letting $\mathcal{MA}^0 = \mathcal{M}$ and $\mathcal{MA}^{n+1} = \mathcal{MA}^n \otimes \mathcal{A}$. We finally define the infinite epistemic model $\mathcal{MA}^* = \bigcup_{n\in\mathbb{N}} \mathcal{MA}^n$.*

In the following we identify objects of the form $(\ldots((w, \alpha_1), \alpha_2), \ldots \alpha_n)$ with $(w, \alpha_1, \ldots, \alpha_n)$. Anticipating the game setting we later define, we call a *play* an infinite sequence $\pi = w\alpha_1\alpha_2\ldots$ such that all finite prefixes of π are in $\mathcal{MA}^*$. A *history* is a finite prefix h of a play. We let $\text{Plays}_{\mathcal{MA}^*}(w)$ and $\text{Hist}_{\mathcal{MA}^*}(w)$ be, respectively, the set of all plays and histories in $\mathcal{MA}^*$ that start with w. These definitions entail the following.

▶ **Lemma 7.** *For every world $(w, \alpha_1, \ldots, \alpha_n) \in \mathcal{M} \otimes \mathcal{A}^n$, and every formula $\varphi \in \mathcal{L}_{\mathbf{EL}}$,*

$$\mathcal{M} \otimes \mathcal{A}^n, (w, \alpha_1, \ldots, \alpha_n) \models \varphi \textit{ iff } \mathcal{MA}^*, w\alpha_1 \ldots \alpha_n \models \varphi.$$

This shows that the alternative definition of epistemic planning given in the next section is equivalent to the usual one.

2.3 Epistemic planning

The epistemic planning problem asks for the existence of an executable sequence of designated actions $\alpha_1, \ldots, \alpha_n$ in an action model $\mathcal{A}$, whose execution from $\mathcal{M}, w$ leads to a situation satisfying some objective expressed as an epistemic logic formula. Formally, we consider the following problem.

▶ **Definition 8** (Plan existence problem).

- *Input: a pointed epistemic model $\mathcal{M}, w$, an action model $\mathcal{A}$ and an* objective *formula $\varphi \in \mathcal{L}_{\mathbf{EL}}$;*
- *Output: yes if there is a history h in $Hist_{\mathcal{M}\mathcal{A}^*}(w)$ such that $\mathcal{M}\mathcal{A}^*, h \models \varphi$.*

▶ Remark 9. Note that usual formulations of the plan existence problem consider a set of distinct pointed action models $(\mathcal{A}_1, \alpha_1), \ldots, (\mathcal{A}_n, \alpha_n)$ instead of one action model $\mathcal{A}$. Both formulations are equivalent, in the sense that they are interreducible in linear time.

Main known results on the plan existence problem are summarised in Table 1, while the relevant pointers to the literature are given in the related work paragraph of the introduction.

We recall some standard notions and notations that we will need in the rest of the paper. A *finite* (resp. *infinite*) *word* over some alphabet Σ is an element of Σ^* (resp. Σ^ω). The *length* of a finite word $w = w_0 w_1 \ldots w_n$ is $|w| := n + 1$, and $\text{last}(w) := w_n$ is its last letter. Given a finite (resp. infinite) word w and $0 \leq i < |w|$ (resp. $i \in \mathbb{N}$), we let w_i be the letter at position i in w, $w_{\leq i} := w_0 \ldots w_i$ is the prefix of w that ends at position i and $w_{\geq i} := w_i w_{i+1} \ldots$ is the suffix of w that starts at position i. We also use variables x that range over some finite domain. We will write $(x = d)$ for the fact "the value of x is d", and use $x := d$ for the effect of setting x to value d. This can all be encoded with atomic propositions.

3 Controller synthesis

We first generalise the plan existence problem to the setting where some environment may perturb the execution of the plan that should thus be robust against it.

Formally, we consider an initial epistemic model $\mathcal{M}$, as in Definition 1, with an initial world w_ι, and an action model $\mathcal{A} = (A, (R_a^{\mathcal{A}})_{a \in Agt}, \text{pre}, \text{post})$ whose set of actions A is partitioned into actions in A_{ctr} controlled by a *Controller* and actions in A_{env} controlled by the *Environment*.

Controller and Environment play in turn: in each round, Controller first chooses to execute an action in A_{ctr}, then Environment chooses to execute an action in A_{env}. Thus instead of seeking a history in $\mathcal{M}\mathcal{A}^*$ that reaches an objective formula, as in epistemic planning, one seeks a *strategy* for Controller: formally, it is a partial function $\sigma : \text{Hist}_{\mathcal{M}\mathcal{A}^*}(w_\iota) \rightharpoonup \mathsf{A}_{\text{ctr}}$ defined on histories of odd length (when it is the controller's turn). An *outcome* of a strategy σ is a play $\pi = w_\iota \alpha_1 \alpha_2 \ldots$ in which the controller *follows* σ, i.e., for all $i \in \mathbb{N}$, $\alpha_{2i+1} = \sigma(\pi_{\leq 2i}) \in \mathsf{A}_{\text{ctr}}$, while the other actions, of the form α_{2i+2}, are selected by the environment. A strategy σ for Controller is *winning* for an objective formula $\varphi \in \mathcal{L}_{\mathbf{EL}}$ if for every outcome π of σ, there exists $i \in \mathbb{N}$ s.t. $\mathcal{M}\mathcal{A}^*, \pi_{\leq i} \models \varphi$.

▶ **Definition 10** (The controller synthesis problem).

- *Input: a pointed epistemic model $\mathcal{M}, w_\iota$, action model $\mathcal{A}$ with $A = \mathsf{A}_{ctr} \uplus \mathsf{A}_{env}$, and an objective $\varphi \in \mathcal{L}_{\mathbf{EL}}$;*
- *Output: yes if there exists a winning strategy for Controller for objective φ; no otherwise.*

procedure controllerSynthesisPublicAnnouncements($\mathcal{M}, w_\iota$, $\mathcal{A}$, $A = \mathsf{A}_{\text{ctr}} \uplus \mathsf{A}_{\text{env}}$, φ)
  set $\mathcal{M}, w_\iota$ as the current pointed epistemic model;
  for $i := 0$ to the number of worlds in $\mathcal{M}$ **do**
    if the current pointed epistemic model satisfies φ **then accept** ;
    if i is even **then** existentially choose $\alpha \in \mathsf{A}_{\text{ctr}}$ that is executable in the current pointed epistemic model (fail if no such action exists);
    if i is odd **then** universally choose $\alpha \in \mathsf{A}_{\text{env}}$ that is executable in the current pointed epistemic model (fail if no such action exists);
    set $\mathcal{M} \otimes \mathcal{A}, (w, \alpha)$ as the current pointed epistemic model, where $\mathcal{M}, w$ was the previous current pointed epistemic model
  reject

Figure 2 Alternating algorithm for deciding in polynomial-time the controller synthesis problem when actions are public announcements.

▶ Remark 11. Formally, we define and study the problem of existence of a strategy. We take the liberty to call the problem *controller synthesis* because all the algorithms we provide can produce a winning strategy whenever there exists one. The same remark applies to the distributed strategy synthesis problem defined in the next section.

As the plan existence problem reduces to the controller synthesis problem, the undecidability of the former entails the one of the latter. We next establish that in all known subcases where the plan existence problem is decidable, so is the controller synthesis problem.

3.1 The case of non-expanding action models

We consider so-called *non-expanding* action models where actions do not expand epistemic models when executed, like public actions. For this type of actions, the search space is finite and thus the problem is decidable. We establish the precise computational complexity of the problem in these cases.

▶ **Theorem 12.** *When actions are public announcements, the controller synthesis problem is* PSPACE*-complete.*

Proof. Since applying public announcements to epistemic models only removes worlds, and does not change those that remain, the number of successive public announcements to consider can be bounded by the number of worlds in the initial epistemic model. We can thus solve the problem with an alternating algorithm that runs in polynomial time, guessing existentially actions of the controller and universally those of the environment. The algorithm is given in Figure 2.

We conclude by recalling that alternating polynomial time corresponds to deterministic polynomial space [13]. Note that checking epistemic formulas (preconditions and φ) in epistemic models, and thus also computing the update product, can be performed in polynomial time.

PSPACE-hardness comes from a polynomial reduction from TQBF (True Quantified Boolean Formulae) which is PSPACE-complete [35]. A QBF formula

$$\exists p_1 \forall p_2 \ldots \exists p_{2k-1} \forall p_{2k} \chi(p_1, \ldots, p_{2k})$$

is transformed in the following instance of the controller existence problem:

- $\mathcal{M}$ is the pointed Kripke model made up of a $\{q_i\}$-world and $\{p_i\}$-world for all $i \in \{1, \dots, 2k\}$ and an extra $\emptyset$-world w, which is the pointed world; the epistemic relation for agent a is universal;

$$\boxed{\begin{array}{cc} w_{2k} : \{p_{2k}\} & u_{2k} : \{q_{2k}\} \\ \vdots & \vdots \\ w_2 : \{p_2\} & u_2 : \{q_2\} \\ w_1 : \{p_1\} & u_1 : \{q_1\} \\ \multicolumn{2}{c}{\longrightarrow w : \emptyset} \end{array}}$$

- The possible announcements are

$$\varphi_{\neg p_i} = \bigwedge_{j=1}^{i-1} K_a \neg q_j \wedge \bigwedge_{j=i}^{2k} \hat{K}_a q_j \wedge \neg p_i \wedge \neg q_i \quad \text{and} \quad \varphi_{p_i} = \bigwedge_{j=1}^{i-1} K_a \neg q_j \wedge \bigwedge_{j=i}^{2k} \hat{K}_a q_j \wedge \neg q_i$$

  for $i \in \{1, \dots, 2k\}$.
  They belong to the controller when i is odd, and to the environment when i is even;
- The goal is $\bigwedge_{j=1}^{2k} K_a \neg q_j \wedge \chi(\hat{K}_a p_1, \dots, \hat{K}_a p_{2k})$.

In the model $\mathcal{M}$, worlds w_i are used to encode assignments of truth values to atoms p_i: removing world w_i means setting p_i to true, while keeping it means setting p_i to false. Worlds u_i, bearing atoms q_i, are used to enforce that the value of each atom p_i is set exactly once. In announcements φ_{p_i} and $\varphi_{\neg p_i}$, conjunct $\bigwedge_{j=1}^{i-1} K_a \neg q_j \wedge \bigwedge_{j=i}^{2k} \hat{K}_a q_j$ implies that worlds $u_1, \dots, u_{i-1}$ have already been removed, while worlds $u_i, \dots, u_{2k}$ are still in the model. Thus announcements φ_{p_i} and $\varphi_{\neg p_i}$ are possible in round i, and only there.

Now observe that announcement $\varphi_{\neg p_i}$, because of conjunct $\neg p_i \wedge \neg q_i$, removes both world w_i and world u_i, thus setting p_i to true. Announcement φ_{p_i} instead removes only world u_i, thus setting p_i to true.

In the goal formula, $\bigwedge_{j=1}^{2k} K_a \neg q_j$ means that all the variables $p_1, \dots, p_{2k}$ have been assigned. The clause $\chi(\hat{K}_a p_1, \dots, \hat{K}_a p_{2k})$ is the formula $\chi(p_1, \dots, p_{2k})$ in which we replaced p_i by $\hat{K}_a p_i$, which holds if and only if world w_i has not been removed by announcements, i.e., if and only if announcement φ_{p_i} was chosen at round i.

The fact that the announcements that assign values to $p_1, p_3, \dots$ are assigned to the controller and that the announcements that assign values to $p_2, p_4, \dots$ are played by the environment reflects the alternation of quantifiers in the formula $\exists p_1 \forall p_2 \dots \exists p_{2k-1} \forall p_{2k} \chi(p_1, \dots, p_{2k})$. ◀

▶ **Theorem 13.** *The controller synthesis problem for public actions is* Exptime*-complete.*

Proof. As for public announcements, applying a public action in a model does not add worlds. However it may change facts in worlds, so that sequences of actions of linear length may not suffice. Nonetheless, linear space is enough to store the current pointed epistemic model, and we can turn the alternating algorithm from the proof of Theorem 12 into one that runs in polynomial space. The new algorithm is given in Figure 3, in which we do not bound the length of the sequence of actions. Note that the algorithm may not terminate, but it is

procedure controllerSynthesisPublicActions($\mathcal{M}, w_\iota$, $\mathcal{A}$, $A = \mathsf{A}_{\text{ctr}} \uplus \mathsf{A}_{\text{env}}$, φ)

    set $\mathcal{M}, w_\iota$ as the current pointed epistemic model;
    $i := 0$;
    while the current pointed epistemic model does not satisfy φ **do**
        if i is even **then** existentially choose $\alpha \in \mathsf{A}_{\text{ctr}}$ that is executable in the current pointed epistemic model (fail if no such action);
        if i is odd **then** universally choose $\alpha \in \mathsf{A}_{\text{env}}$ that is executable in the current pointed epistemic model (fail if no such action);
        set $\mathcal{M} \otimes \mathcal{A}, (w, \alpha)$ as the current pointed epistemic model, where $\mathcal{M}, w$ was the previous current pointed epistemic model;
        $i := 1 - i$;
    accept

Figure 3 Alternating algorithm for deciding in polynomial-space the controller synthesis problem when actions are public actions.

folklore that we can add a counter to ensure termination while staying in polynomial-space; we do not present these tedious technicalities here.

The EXPTIME-membership of the problem follows from the fact that alternating polynomial space corresponds to exponential time [13].

EXPTIME-hardness is obtained by reduction from the conditional planning problem, a variant of classical planning with full observability and non-deterministic actions, where the plan should lead to a situation satisfying the goal no matter how nondeterminism is resolved. However the plan can depend on how nondeterminism is resolved, hence the name "conditional plan" [24, 34].

Stated in our terms, conditional planning essentially corresponds to a particular case of controller synthesis which is purely boolean (no epistemic content), but where actions chosen by the controller have nondeterministic effects, and the environment resolves nondeterminism. Since everything is purely boolean, the initial situation is a one-world epistemic model, i.e., a valuation over a finite set of atoms AP, the goal is a boolean formula over AP, and each action is a one-state action model with nondeterministic postcondition. A conditional plan is then a winning strategy for the controller. Thus, to finish the reduction, we only have to show how to simulate nondeterministic actions in our setting.

In [24, 34], a nondeterministic action is modelled as a tuple $\langle \varphi, \overrightarrow{\text{post}} \rangle$ where φ is a Boolean precondition, and $\overrightarrow{\text{post}}$ is a finite set $\{\text{post}_1, \ldots, \text{post}_n\}$, where $\text{post}_i : AP \to \mathcal{L}_{\text{Prop}}$ is a postcondition. The idea is that in each round the controller chooses an action among those whose precondition is true, and the environment resolves the non-determinism by choosing which postcondition of $\overrightarrow{\text{post}}$ to apply to the current valuation. For each nondeterministic action $\langle \varphi, \{\text{post}_1, \ldots, \text{post}_n\} \rangle$ of the conditional planning instance, we create one action model for the controller that stores in a finite-domain variable *action* which action has been played, and n actions for the environment that correspond to the different possible poscondtions. The action for the controller is defined as follows:

$pre : \varphi$
$post : action := \langle \varphi, \overrightarrow{\text{post}} \rangle$

while the actions for the environment are, for each $i \in \{1, \ldots, n\}$,

$pre : action = \langle \varphi, \overrightarrow{\text{post}} \rangle$
$post : \text{post}_i$

This finishes the proof, and also shows how the controller synthesis problem subsumes conditional planning. We now present an alternative proof that reduces from a more basic decision problem called G_4, introduced by Chandra and Stockmeyer [36]. This is essentially an adaptation of the proof from [24] for the EXPTIME-hardness of conditional planning.

The input to the G_4 problem is a 13-DNF formula over $2k$ atomic propositions $p_1, \ldots, p_k$, $q_1, \ldots, q_k$ and an initial valuation. Atoms $p_1, \ldots, p_k$ are controlled by the controller (the existential player) while $q_1, \ldots, q_k$ are controlled by the environment (the universal player). Now, the following game is played: each player, when it is her turn to play, flips the assignment of one of the variables she controls, and turns alternate. The game stops when the 13-DNF formula becomes true, and the winner is the player that made the last move. An instance of the G_4 problem is positive if the controller has a winning strategy.

We construct the following instance of our controller synthesis problem. The initial epistemic model is made up of one world, whose valuation is the initial valuation of G_4. Actions of the controller are:

$$\begin{array}{lcl} pre : \top & & pre : \top \\ post : p_1 := \neg p_1 & \cdots & post : p_k := \neg p_k \end{array}$$

Actions of the environment are:

$$\begin{array}{lcl} pre : \top & & pre : \top \\ post : q_1 := \neg q_1 & \cdots & post : q_k := \neg q_k \end{array}$$

The goal is the 13-DNF formula. ◀

Theorem 13 also generalises to other non-expanding actions models such as the so-called *separable action models* [14], where the preconditions of any two actions in the same connected component are logically inconsistent.

3.2 The case of propositional action models

To solve our controller synthesis problem we rely on the approach followed in [26] to solve the plan existence problem for propositional actions. This approach has two main ingredients: (I1) when $\mathcal{A}$ is propositional, the generated structure $\mathcal{MA}^*$ can be represented finitely, and (I2) one can decide the existence of a winning strategy in a certain class of two-player games with epistemic objectives.

▶ **Theorem 14.** *When action models are propositional, the controller synthesis problem is decidable, and in* $(k+1)$-EXPTIME *if the objective's modal depth is bounded by* k*.*

We devote the rest of this section to prove Theorem 14, which requires to introduce particular game arenas.

▶ **Definition 15.** *A* two-player epistemic game arena *is a structure* $\mathcal{G} = (W, w_\iota, \Delta, (R_a)_{a\in Agt}, V)$ *where* $(W, (R_a)_{a\in Agt}, V)$ *is an epistemic model,* $W = W_0 \uplus W_1$ *is partitioned into the positions of players 0 and 1,* w_ι *is an* initial world *and* $\Delta \subseteq W \times W$ *is a* transition relation*.*

A *play* in a game arena $\mathcal{G}$ is an infinite sequence of worlds $\pi = w_0 w_1 w_2 \ldots$ such that for all $i \in \mathbb{N}$, $w_i \Delta w_{i+1}$, and a *history* is a finite nonempty prefix of a play. We let $\mathrm{Plays}_{\mathcal{G}}$ and $\mathrm{Hist}_{\mathcal{G}}$ be the sets of plays and histories in $\mathcal{G}$, respectively. Accessibility relations $(R_a)_{a\in Agt}$ are extended to histories to interpret epistemic formulas: two histories $h = w_0 \ldots w_n$ and $h' = w'_0 \ldots w'_m$ are related by R_a whenever $n = m$ (same length) and $w_i R_a w'_i$ for every $i \leq n$.

A *strategy* for player 0 is a partial function $\sigma : \mathrm{Hist}_{\mathcal{G}} \rightharpoonup W$ such that for every h with $\mathrm{last}(h) \in W_0$: $\mathrm{last}(h) \Delta \sigma(h)$. A play $\pi = w_\iota w_1 w_2 \ldots$ is an *outcome* of σ if for every $i \in \mathbb{N}$

with $\pi_i \in W_1$, we have $\pi_{i+1} = \sigma(\pi_{\leq i})$. Strategy σ is *winning* for an epistemic objective $\varphi \in \mathcal{L}_{\mathbf{EL}}$, if for every outcome π of σ there is some $i \in \mathbb{N}$ with $\pi_{\leq i} \models \varphi$.

▶ **Theorem 16** ([12]). *The existence of a winning strategy for player 0 in an epistemic game $\mathcal{G}$ for an epistemic objective φ of modal depth k can be decided in time k-exponential in $|\mathcal{G}|$ and $|\varphi|$.*

We show that the controller synthesis problem for propositional action models reduces to solving an epistemic game:

▶ **Proposition 17.** *Given an instance $((\mathcal{M}, w_\iota), \mathcal{A}, \varphi)$ of the controller synthesis problem where $\mathcal{A}$ is propositional, one can construct a game arena $\mathcal{G}$ such that Controller wins in $((\mathcal{M}, w_\iota), \mathcal{A}, \varphi)$ iff Player 0 wins in $\mathcal{G}$ for objective φ and $|\mathcal{G}| \leq |\mathcal{M}| + |\mathcal{A}| \times 2^{m+1}$, where m is the number of atomic propositions involved in $\mathcal{M}, \mathcal{A}$ and φ.*

Proof. Let $\mathcal{M} = (W, (R_a)_{a\in Agt}, V)$ and $\mathcal{A} = (A, \{R_a^{\mathcal{A}}\}_{a\in Agt}, \mathrm{pre}, \mathrm{post})$, and let AP_u be the atomic propositions involved. The worlds of the game arena $\mathcal{G}$ that we build are either worlds $w \in \mathcal{M}$ or tuples (α, v, i) where $\alpha \in \mathcal{A}$ represents the last action performed, $v \in 2^{AP_u}$ is the current valutation, and $i \in \{0, 1\}$ indicates whose turn it is to play: 0 for Controller and 1 for Environment. In an initial world w, Controller can choose an action $\alpha \in \mathsf{A}_{\mathrm{ctr}}$ such that $w \models \mathrm{pre}(\alpha)$ and move to $(\alpha, V(w, \alpha), 1)$; in a world of the form (α, v, i), if $i = 1$ (resp., $i = 0$), Environment (resp, Controller) chooses an action $\alpha' \in \mathsf{A}_{\mathrm{env}}$ (resp., $\alpha' \in \mathsf{A}_{\mathrm{ctr}}$) such that $v \models \mathrm{pre}(\alpha')$, and moves to $(\alpha', V(w, \alpha), 1 - i)$.

Formally, letting $A_0 = \mathsf{A}_{\mathrm{ctr}}$ and $A_1 = \mathsf{A}_{\mathrm{env}}$, we define the epistemic game arena $\mathcal{G} = (W', w_\iota, \Delta, (R'_a)_{a\in Agt}, V')$ as follows: Let $W_0 = W \cup A \times 2^{AP} \times \{0\}$, $W_1 = A \times 2^{AP} \times \{1\}$, and $W' = W_0 \cup W_1$. For transitions, for $w \in W$, $\alpha, \alpha' \in A$ and $v, v' \in 2^{AP}$, we let $w\Delta(\alpha, v, 1)$ if $\alpha \in \mathsf{A}_{\mathrm{ctr}}, w \models \mathrm{pre}(\alpha)$ and $v = V(w, \alpha)$, and $(\alpha, v, i)\Delta(\alpha', v', 1 - i)$ if $\alpha' \in A_i, v \models \mathrm{pre}(\alpha')$ and $v' = V(v, \alpha)$. We also let $wR'_a w'$ if $wR'_a w'$, and $(\alpha, v, i)R_{\prime}(\alpha', v', i)$ if $\alpha R_a^{\mathcal{A}} \alpha'$, and finally $V'(w) = V(w)$ and $V'(\alpha, v, i) = v$.

The structure given by the set of histories $\mathrm{Hist}_{\mathcal{G}}$ and relations R'_a extended to these histories is isomorphic to $\mathcal{M}\mathcal{A}^*$, and histories of odd (resp. even) length in $\mathcal{M}\mathcal{A}^*$ correspond to histories that end in W_0 (resp. W_1) in $\mathcal{G}$ (provided they start in w_ι). It follows that there is a winning strategy for Controller in the original problem if and only if there is winning strategy for Player 0 in $\mathcal{G}$ with objective φ. ◀

Note that, as stated in Proposition 17, the resulting game arena is indeed polynomial in the size of the epistemic and action models, but it is exponential in the number of atomic propositions involved in the problem. This is because states of the game arena contain all possible valuations. Theorem 14 now follows from Theorem 16 and Proposition 17.

With the controller synthesis problem we enriched epistemic planning with an adversarial environment. Still, as in epistemic planning, the agents are mere observers. We now make a step further and make the agents players of the game.

4 Distributed strategy synthesis

In this section agents are no more passive, but instead they are players who choose themselves the actions that occur. The set Agt of agents is split into two teams $Agt_\exists$ and $Agt_\forall$ that play against each other, and we may say *players* instead of agents.

4.1 Setting up the game

Unlike the external controller from the previous section, our players now have imperfect information. The fundamental feature of games with imperfect information is that when a player cannot distinguish between two different situations, a strategy for this player should prescribe the same action in both situations. All the additional complexity in solving games with imperfect information compared to the perfect information setting arises from this constraint. Such strategies are often called *uniform* or *observation-based* (see for instance [33, 37, 1]). Since games with imperfect information consider S5 models, i.e., where accessibility relations are equivalence relations, and it is unclear what uniform strategies mean in non-S5 models[1], we also assume from now on that all epistemic and action models are S5. We stress this assumption by writing $\sim_a$ (resp. $\sim_a^{\mathcal{A}}$) instead of R_a (resp. $R_a^{\mathcal{A}}$).

We start from an initial pointed epistemic model $\mathcal{M}, w$, and an action model $\mathcal{A}$ whose set of actions is partitioned into subsets $(A_a)_{a \in Agt}$ of actions for each player. The game we describe is turn-based. We use the variable turn ranging over Agt to represent whose turn it is to play. We require that for each $a \in A_a$, $\mathrm{pre}(\alpha)$ implies $(\mathsf{turn}{=}a)$, and that postconditions for variable turn do not depend on the current world, but instead the next value of turn is completely determined by the action only.

Moreover, in order to obtain a proper imperfect-information game, we demand the following hypotheses:

Hypotheses on $\mathcal{M}$ and $\mathcal{A}$

(H1) **The starting player is known:** there is a player a such that for all $u \in W$, $\mathcal{M}, u \models (\mathsf{turn}{=}a)$;

(H2) **The turn stays known:** for all actions α, α' and agent a, if $\alpha R_a \alpha'$, then α and α' assign the same value to turn.

(H3) **Players know their available actions:** if agent a can execute α after history h, then she can also execute it after every history h' with $h \sim_a h'$.

All these hypotheses can be either enforced syntactically or checked in the different decidable cases we consider in the rest of this work (see the long version for detail).

We now define formally the notion of uniform strategies.

▶ **Definition 18** (Uniform strategy). *A strategy σ for player a is* uniform *if for every pair of histories h, h' where it is player a's turn, $h \sim_a h'$ entails $\sigma(h) = \sigma(h')$.*

In the rest of this section, a strategy of a player in $Agt_\exists$ is implicitly uniform. When one selects a strategy for each player in $Agt_\exists$, the result is called a *distributed strategy*, and an *outcome* of a distributed strategy is a play in which all players in $Agt_\exists$ follow their prescribed strategy. A distributed strategy is *winning* for an objective formula φ if all its outcomes eventually satisfy φ.

4.2 The distributed strategy synthesis problem

We study the existence of a distributed strategy for players in $Agt_\exists$ that ensures to reach an epistemic goal property.

[1] Actually the usual definition seems to make sense for KD45, i.e. models whose relations are serial, transitive, and Euclidean. But this would be highly non-standard, and since all the literature on games with imperfect information considers S5 models.

▶ **Definition 19** (Distributed strategy existence problem)**.**

- *Input: a pointed epistemic model* $\mathcal{M}, w$ *and an action model* $\mathcal{A}$ *partitioned into* $(A_a)_{a \in Agt}$ *that satisfy hypotheses (H1)-(H3), and an objective formula* $\varphi \in \mathcal{L}_{\mathbf{EL}}$*;*
- *Output: yes if there exists a winning distributed strategy for players in* $Agt_\exists$*; no otherwise.*

Unlike the controller synthesis problem which we proved decidable for propositional actions, synthesising distributed strategies is undecidable for propositional actions, already for a team of two players.

4.3 Undecidability for two existential players

The following Theorem 20 is a reformulation in our setting of the classical undecidability result from Reif and Peterson [29]. However, we decide to promote an existing elegant reformulation of that very same result, called TEAM DFA GAME [15, Def. 1, p. 14:7], that can be reduced to our distributed strategy synthesis problem.

▶ **Theorem 20.** *The distributed strategy synthesis problem is undecidable, already for a propositional action model and two existential players against one universal player.*

Proof. The proof is given by reduction from the problem TEAM DFA GAME [15, Def. 1, p. 14:7], shown to be undecidable.

We consider a two-versus-one (players a and b versus player $\forall$) team game played on a deterministic finite automaton (DFA) $\mathcal{A}$ whose alphabet is $\{0, 1\}$, whose set of states is Q, initial state is q_0, transition function δ. Special subsets of states $F_\exists$ and $F_\forall$ are given. The game starts with $\mathcal{A}$ being in state q_0. Each round is divided in six steps:

1. if the current state q is in $F_\exists$ then team $\{a, b\}$ wins; if the current state q is in $F_\forall$ then team $\{\forall\}$ wins;
2. Player $\forall$ inputs two bits β, β' into $\mathcal{A}$;
3. Player a learns β;
4. Player a inputs one bit m into $\mathcal{A}$;
5. Player b learns β';
6. Player b inputs one bit m' into $\mathcal{A}$.

At each step, player $\forall$ has perfect information. TEAM DFA GAME is the decision problem: given an DFA $\mathcal{A}$, subsets of states $F_\exists$, $F_\forall$, does the team $\{a, b\}$ have a winning strategy?

The rest of the proof consists in representing the initial situation, the game rules and the goal of a TEAM DFA GAME instance as a distributed strategy existence problem instance.

Definition of the reduction. Let $(\mathcal{A}, F_\exists, F_\forall)$ be an instance of TEAM DFA GAME. Teams are $Agt_\exists = \{a, b\}$ and $Agt_\forall = \{\forall\}$. We introduce a finite-domain variable q that ranges over the set of states of $\mathcal{A}$. The variable q can be represented by a finite set of atomic propositions: for example, for an automaton with 8 states from $\{0,\ldots,7\}$, three atomic propositions, $bit_1(q)$, $bit_2(q)$ and $bit_3(q)$ so that say $(q = 5)$ is the Boolean formula $bit_1(q) \wedge \neg bit_2(q) \wedge bit_3(q)$. We prefer to keep with a finite-domain variable q to avoid technicalities.

We also introduce a finite-domain variable stp that ranges over $\{1, 2, 3, 4, 5, 6\}$. The Boolean variable lost is true if the team $Agt_\exists$ has lost. We define $\mathcal{M}, w$ to be the single-world S5 epistemic model in which $(\mathsf{turn}{=}\forall)$, $(q{=}q_0)$, $(\mathsf{stp}{=}1)$, $\neg\mathsf{lost}$. The actions in $A_\forall$ form an a- and b-indistinguishably equivalence class and are of the form:

- pre: $(\mathsf{turn}{=}\forall) \wedge (\mathsf{stp}{=}1) \wedge q{\in}F_\forall$; post: $\mathsf{lost}{:=}\top, \mathsf{stp}{:=}2$
- pre: $(\mathsf{turn}{=}\forall) \wedge (\mathsf{stp}{=}1) \wedge q{\notin}F_\forall$; post: $\mathsf{stp}{:=}2$
- pre: $(\mathsf{turn}{=}\forall){\wedge}(\mathsf{stp}{=}2)$; post: $\beta{:=}0, \beta'{:=}0, q{:=}\delta(q,00), \mathsf{turn}{:=}a, \mathsf{stp}{:=}3$
- pre: $(\mathsf{turn}{=}\forall){\wedge}(\mathsf{stp}{=}2)$; post: $\beta{:=}1, \beta'{:=}0, q{:=}\delta(q,10), \mathsf{turn}{:=}a, \mathsf{stp}{:=}3$
- pre: $(\mathsf{turn}{=}\forall){\wedge}(\mathsf{stp}{=}2)$; post: $\beta{:=}0, \beta'{:=}1, q{:=}\delta(q,01), \mathsf{turn}{:=}a, \mathsf{stp}{:=}3$
- pre: $(\mathsf{turn}{=}\forall){\wedge}(\mathsf{stp}{=}2)$; post: $\beta{:=}1, \beta'{:=}1, q{:=}\delta(q,11), \mathsf{turn}{:=}a, \mathsf{stp}{:=}3$

A_a is a b-indistinguishably equivalence class and contains:

- pre: $(\mathsf{turn}{=}a) \wedge (\mathsf{stp}{=}3) \wedge \beta$; post: $\mathsf{stp} := 4$
- pre: $(\mathsf{turn}{=}a) \wedge (\mathsf{stp}{=}3) \wedge \neg\beta$; post: $\mathsf{stp} := 4$
- pre: $(\mathsf{turn}{=}a) \wedge (\mathsf{stp}{=}4)$; post: $m{:=}\bot; \mathsf{stp} := 5; q := \delta(q,0); \mathsf{turn} := b$
- pre: $(\mathsf{turn}{=}a) \wedge (\mathsf{stp}{=}4)$; post: $m{:=}\top; \mathsf{stp}{:=}5; q{:=}\delta(q,1); \mathsf{turn} := b$

A_b is an a-indistinguishably equivalence class and contains:

- pre: $(\mathsf{turn}{=}b) \wedge (\mathsf{stp}{=}5) \wedge \beta'$; post: $\mathsf{stp}{:=}6$
- pre: $(\mathsf{turn}{=}b) \wedge (\mathsf{stp}{=}5) \wedge \neg\beta'$; post: $\mathsf{stp}{:=}6$
- pre: $(\mathsf{turn}{=}b) \wedge (\mathsf{stp}{=}6)$; post: $m'{:=}\bot; \mathsf{stp}{:=}1; q{:=}\delta(q,0); \mathsf{turn}{:=}\forall$
- pre: $(\mathsf{turn}{=}b) \wedge (\mathsf{stp}{=}4)$; post: $m'{:=}\top; \mathsf{stp}{:=}1; q{:=}\delta(q,1); \mathsf{turn}{:=}\forall$

The assignments $q := \delta(q,0)$ and $q := \delta(q,1)$ are shortcuts for some "if statements" on states, e.g. 'if $q = 5$, then $q := 2$'. For instance, assuming that we have eight states $\{s_0, \dots, s_7\}$ which are thus representable with three bits, the assignment $q := \delta(q,0)$ is simulated by the following set of propositional assignments: $\{bit_i(q) := \psi_i\}_{i=1..3}$, where ψ_i is the Boolean formula $\bigvee_{k\in 0..7 \text{ s.t. the i-th bit of } \delta(s_k,0) \text{ is } 1}(q = s_k)$.

The goal formula φ is $\neg\mathsf{lost} \wedge (\mathsf{stp}{=}1) \wedge (q{\in}F_\exists)$. ◀

We now turn to decidable cases: games with imperfect information and epistemic objectives are known to be decidable either when actions are public [6], or when information is hierarchical [27]. We establish similar results in our setting.

4.4 The case of non-expanding action models

Theorems 12 and 13 of Section 3 generalise to the distributed strategy synthesis problem. First, we inherit the lower bounds by letting $Agt_\exists = \{\text{Controller}\}$ and $Agt_\forall = \{\text{Environment}\}$, and by making them alternate turns. Second, the upper bounds are obtained by adapting the alternating algorithms for the upper bounds of Theorems 12 and 13. We need to ensure that existential choices of actions of an agent $a \in Agt_\exists$ lead to a *uniform* strategy. To do that, every time agent a picks an action α, we perform an extra universal choice over $\sim_a$-indistinguishable worlds, and continue executing the algorithm from these worlds.

▶ **Theorem 21.** *For public announcements, the distributed strategy synthesis problem is PSPACE-complete.*

Proof. The alternating algorithm that runs in polynomial time is similar to the one given in the proof of Theorem 12, except that we also add universal choices of $R_a^{\mathcal{A}}$-successors for player a in $Agt_\exists$. More precisely, the steps of the algorithm works as follows. When it is the turn of a player a in $Agt_\forall$ to play, simply universally guess an executable action in A_a. When it is the turn of a player a in $Agt_\exists$ to play, then perform the following steps:

- first existentially guess an executable action α in A_a;

- second, universally guess a R_a-successor of the pointed world in the current epistemic model and make it as the new pointed world;
- compute the new epistemic model by executing the α.

As for Theorem 12, the length of such a sequence is bounded by the number of worlds in the initial epistemic model $\mathcal{M}$. The obtained algorithm is given in Figure 4.

```
procedure DistrStratSynthesisPublicAnnouncements(M, w_ι, A, (A_a)_{a∈Agt}, φ)
    set M, w_ι as the current pointed epistemic model;
    for i := 0 to the number of worlds in M do
        if the current pointed epistemic model satisfies φ then accept ;
        let a be the agent such that (turn=a) is true in the current pointed epistemic
        model;
        if a ∈ Agt_∃ then
            existentially choose α ∈ A_a that is executable in the current pointed
            epistemic model (fail if no such action exists);
            universally choose an R_a-successor of the pointed world in the current
            epistemic model and make it as the new pointed world;
        else if a ∈ Agt_∀ then
            universally choose α ∈ A_a that is executable in the current pointed
            epistemic model (fail if no such action exists);
        set M ⊗ A, (w, α) as the current pointed epistemic model, where M, w was
        the previous current pointed epistemic model;
    reject
```

Figure 4 Alternating algorithm for deciding in polynomial-time the distributed strategy synthesis problem when actions are public announcements.

Hardness follows from Theorem 12 (simply consider the controller as the single agent in $Agt_\exists$, the environment as the single agent in $Agt_\forall$, and make them alternate). ◀

▶ **Theorem 22.** *For public actions, the distributed strategy synthesis problem is EXPTIME-complete.*

Proof. The alternating algorithm that runs in polynomial space is similar to the one given in the proof of Theorem 12. The algorithm is in polynomial space for the same reason said in the proof of Theorem 13. The obtained algorithm is given in Figure 5.

Hardness follows from Theorem 13. ◀

We now turn to a decidable case for propositional actions.

4.5 Propositional actions+hierarchical information

We consider propositional action models, which may make the size of epistemic models grow unboundedly, but where the information of the different players is hierarchical, making it easier to synchronise the existential players' strategies.

According to Theorem 20, the distributed strategy synthesis problem is undecidable for propositional actions and a two-player team $Agt_\exists = \{a, b\}$ against team $Agt_\forall = \{\forall\}$. Observe that in the proof of Theorem 20, the information of players a and b is incomparable: in each round a only learns the first bit produced by $\forall$'s move, while b only learns the second

procedure DistrStratSynthesisPublicActions($\mathcal{M}, w_\iota$, $\mathcal{A}$, $(A_a)_{a \in Agt}$, φ)
    set $\mathcal{M}, w_\iota$ as the current pointed epistemic model;
    while the current pointed epistemic model does not satisfy φ **do**
        let a be the agent such that $(\mathsf{turn}{=}a)$ is true in the current pointed epistemic model;
        if $a \in Agt_\exists$ **then**
            existentially choose $\alpha \in A_a$ that is executable in the current pointed epistemic model (fail if no such action exists);
            universally choose a R_a-successor of the pointed world in the current epistemic model and make it as the new pointed world;
        else if $a \in Agt_\forall$ **then**
            universally choose $\alpha \in A_a$ that is executable in the current pointed epistemic model (fail if no such action exists);
        set $\mathcal{M} \otimes \mathcal{A}, (w, \alpha)$ as the current pointed epistemic model, where $\mathcal{M}, w$ was the previous current pointed epistemic model;
    accept

Figure 5 Alternating algorithm for deciding in polynomial-space the distributed strategy synthesis problem when actions are public actions.

bit. This cannot be simulated in games with so-called *hierarchical information*, a classic restriction to regain decidability in multi-player games of imperfect information [28, 30].

We say that an input of the distributed strategy synthesis problem $((\mathcal{M}, w_\iota), \mathcal{A}, \varphi)$ presents *hierarchical information* if the set of $Agt_\exists$ can be totally ordered ($a_1 < \ldots < a_n$) so that $\sim_{a_i} \subseteq \sim_{a_{i+1}}$ and $\sim^{\mathcal{A}}_{a_i} \subseteq \sim^{\mathcal{A}}_{a_{i+1}}$, for each $1 \leq i < n$.

▶ **Theorem 23.** *Distributed strategy synthesis with propositional actions and hierarchical information is decidable.*

We end the section by sketching the proof of Theorem 23. We start by introducing a multi-player variant of the epistemic game arenas from Definition 15.

▶ **Definition 24.** *A* multi-player epistemic game arena $\mathcal{G} = (W, w_\iota, \Delta, (\sim_a)_{a \in Agt}, V)$ *is a structure such that*

- $(W, (\sim_a)_{a \in Agt}, V)$ *is an epistemic model,*
- $W = \uplus_{a \in Agt} W_a$ *is partitioned into positions of each agent,*
- w_ι *is an* initial world *and*
- $\Delta \subseteq W \times W$ *is a* transition relation.

Accessibility relations $\sim_a$ are extended to histories, strategies of agent a are required to be uniform with respect to $\sim_a$, and the notions of outcomes, distributed strategies and winning distributed strategies are defined as before.

Theorem 23 is established by reducing the distributed strategy synthesis problem to a similar problem in multi-player epistemic games, known to be decidable:

▶ **Theorem 25** ([31, 27]). *Existence of winning distributed strategies in multi-player epistemic games with hierarchical information and epistemic temporal objectives is decidable.*

The reduction is very similar to the one in the proof of Proposition 17. The main difference is that we use variable turn instead of bit $i \in \{0, 1\}$ to define the positions of the different agents. The imperfect information of players is defined based on the accessibility relations in $\mathcal{M}$ and $\mathcal{A}$. More precisely, we can show that:

▶ **Proposition 26.** *Given an instance $((\mathcal{M}, w_\iota), \mathcal{A}, \varphi)$ of the distributed strategy synthesis problem where $\mathcal{A}$ is propositional, one can construct a multi-player epistemic game arena $\mathcal{G}$ such that the distributed strategy synthesis problem is equivalent to the existence of a winning distributed strategy for $Agt_\exists$ to enforce φ in $\mathcal{G}$ and $|\mathcal{G}| \leq |\mathcal{M}| + |\mathcal{A}| \times 2^m$, where m is the number of atomic propositions involved.*

Proof. Let $\mathcal{M} = (W, (\sim_a)_{a\in Agt}, V)$ be an S5 epistemic model, $\mathcal{A} = (A, \{\sim_a^{\mathcal{A}}\}_{a\in Agt}, \mathrm{pre}, \mathrm{post})$ an S5 propositional event model with $A = \uplus_{a\in Agt} A_a$, and $w_\iota \in W$ an initial world. Let AP_u be the atomic propositions involved.

We define the multiplayer epistemic game arena $\mathcal{G} = (W', w_\iota, \Delta, (\sim'_a)_{a\in Agt}, V')$ as follows. First, let $W' = W \cup A \times 2^{AP}$, and for each $a \in Agt$ let $W'_a = \{w \mid w \models (\mathsf{turn}{=}a)\} \cup \{(\alpha, v) \mid v \models (\mathsf{turn}{=}a)\}$. Next, for transitions, we let $(w, (\alpha, v)) \in \Delta$ if $w \models \mathrm{pre}(\alpha)$ and $v = V(w, \alpha)$, and $((\alpha, v), (\alpha', v')) \in \Delta$ if $v \models \mathrm{pre}(\alpha')$ and $v' = V(v, \alpha)$. Next, for each $a \in Agt$, we let $\sim'_a = \sim_a \cup \{((\alpha, v), (\alpha', v')) \mid (\alpha, \alpha') \in R_a^{\mathcal{A}}\}$, and finally $V'(w) = V(w)$ and $V'(\alpha, v) = v$.

One can see that the structure given by the set of histories $\mathrm{Hist}_{\mathcal{G}}$ and relations $\sim'_a$ extended to these histories is isomorphic to $\mathcal{M}\mathcal{A}^*$, and histories that satisfy $(\mathsf{turn}{=}a)$ in $\mathcal{M}\mathcal{A}^*$ correspond to histories that also satisfy $(\mathsf{turn}{=}a)$ in $\mathcal{G}$ (provided they start in w_ι), and we are done. ◀

5 Perspectives

We have incrementally extended the framework of epistemic planning to a game setting where players' actions are described by action models from DEL. We have established fine-grained results depending on the type of action models.

Works on classical planning that consider game features exist, and they can easily be located in the landscape of decision problems we have considered. Typically, our controller (resp. distributed strategy) synthesis problem subsumes *conditional planning* with full (resp. partial) observability [34]. Also, *conformant planning* (partial information where the plan is a sequence of actions) corresponds to a particular case of our distributed strategy synthesis problem where $Agt_\exists = \{\exists\}$, $Agt_\forall = \{\forall\}$ are singletons, and $\exists$ is blind, i.e., all actions in $A_\forall$ are indistinguishable for her. Blindness and uniformity assumption make that the strategies of $\exists$ can be seen as sequential plans.

Moreover, the decision problems we have considered go well beyond classical planning by addressing, e.g. distributed planning with cooperative or/and adversarial features. We are thus confronted in a DEL setting to issues usually met in game theory, as witnessed by the undecidability of the distributed strategy synthesis problem for rather simple action models (Theorem 20). However, the DEL perspective we propose offers a language to specify actions (preconditions, postconditions, and epistemic relations between actions) that may help identifying yet unknown decidable cases.

We note that our results should transfer to Game Description Language, equivalent to DEL [18].

One interesting extension of the unifying setting of DEL games would be to consider concurrent games, where players execute actions concurrently, but this will require to first generalise the product operation of DEL. Another direction would be to consider richer objectives such as ones expressed in epistemic temporal logic, which can express not only reachability but also safety or liveness objectives for instance.

Our approach contributes to putting closer the field of multi-agent planning in AI with the field of multi-player games in formal methods. The setting of DEL games may be beneficial to both, allowing the transfer of powerful automata and game techniques from formal methods to epistemic planning, and bringing in multiplayer games new insights from the fine modelling offered by DEL.

References

1 K.R. Apt and E. Grädel. *Lectures in game theory for computer scientists*. Cambridge UP, 2011.

2 G. Aucher, B. Maubert, and S. Pinchinat. Automata techniques for epistemic protocol synthesis. In *SR'14*, pages 97–103, 2014. pathdoi:10.4204/EPTCS.146.13.

3 J.F. Baffier, M.K. Chiu, Y. Diez, M. Korman, V. Mitsou, A. Van Renssen, M. Roeloffzen, and Y. Uno. Hanabi is np-hard, even for cheaters who look at their cards. *arXiv preprint arXiv:1603.01911*, 2016.

4 A. Baltag, L. S Moss, and S. Solecki. The logic of public announcements, common knowledge, and private suspicions. In *TARK'98*, pages 43–56, 1998.

5 F. Belardinelli, A. Lomuscio, A. Murano, and S. Rubin. Verification of broadcasting multi-agent systems against an epistemic strategy logic. In *IJCAI'17*, pages 91–97, 2017.

6 F. Belardinelli, A. Lomuscio, A. Murano, and S. Rubin. Verification of multi-agent systems with imperfect information and public actions. In *AAMAS'17*, 2017.

7 D. Berwanger, A. B. Mathew, and M. van den Bogaard. Hierarchical information and the synthesis of distributed strategies. *Acta Inf.*, 55(8), 2018.

8 A. Blumensath and E. Grädel. Automatic structures. In *LICS'00*, pages 51–62, 2000.

9 T. Bolander and M.B. Andersen. Epistemic planning for single and multi-agent systems. *J. Appl. Non-Class. Log.*, 21(1):9–34, 2011.

10 T. Bolander, M.H. Jensen, and F. Schwarzentruber. Complexity results in epistemic planning. In *IJCAI*, pages 2791–2797, 2015.

11 P. Bouyer. Games on graphs with a public signal monitoring. In *FOSSACS'18*. Springer, 2018.

12 L. Bozzelli, B. Maubert, and S. Pinchinat. Uniform strategies, rational relations and jumping automata. *Inf. Comput.*, 242:80–107, 2015.

13 A.K. Chandra and L.J. Stockmeyer. Alternation. In *FOCS'76*, 1976.

14 T. Charrier. *Theoretical complexity of reasoning in dynamic epistemic logic and symbolic approach*. PhD thesis, Université de Rennes 1, 2018.

15 M. J. Coulombe and J. Lynch. Cooperating in video games? impossible! undecidability of team multiplayer games. In *FUN'18*, pages 14:1–14:16, 2018.

16 G. Douéneau-Tabot, S. Pinchinat, and F. Schwarzentruber. Chain-monadic second order logic over regular automatic trees and epistemic planning synthesis. In *AiML'18*, 2018.

17 T. Engesser, T. Bolander, R. Mattmüller, and B. Nebel. Cooperative epistemic multi-agent planning for implicit coordination. In *M4M*, pages 75–90, 2017.

18 T. Engesser, R. Mattmüller, B. Nebel, and M. Thielscher. Game description language and dynamic epistemic logic compared. In *IJCAI'18*, pages 1795–1802, 2018.

19 R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

20 Hector Geffner and Blai Bonet. A concise introduction to models and methods for automated planning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 8(1):1–141, 2013.

21 D. Harel and A. Pnueli. On the development of reactive systems. In *Logics and models of concurrent systems*, pages 477–498. Springer, 1985.

22 S. Lê Cong, S. Pinchinat, and F. Schwarzentruber. Small undecidable problems in epistemic planning. In *IJCAI'18*, 2018.

23 T. De Lima. Alternating-time temporal dynamic epistemic logic. *J. Log. Comput.*, 24(6):1145–1178, 2014.

24 M.L. Littman, J. Goldsmith, and M. Mundhenk. The computational complexity of probabilistic planning. *J. Artif. Intell. Res.*, 9:1–36, 1998.

25 A. Matros. Lloyd shapley and chess with imperfect information. *Games and Economic Behavior*, 108:600–613, 2018.

26 B. Maubert. *Logical foundations of games with imperfect information : uniform strategies*. PhD thesis, University of Rennes 1, France, 2014.

27 B. Maubert and A. Murano. Reasoning about knowledge and strategies under hierarchical information. In *KR'18*, pages 530–540, 2018.

28 G. Peterson, J. Reif, and S. Azhar. Decision algorithms for multiplayer noncooperative games of incomplete information. *Computers & Mathematics with Applications*, 43(1):179–206, 2002.

29 G.L. Peterson and J.H. Reif. Multiple-person alternation. In *FOCS'79*, 1979.

30 A. Pnueli and R. Rosner. Distributed reactive systems are hard to synthesize. In *FOCS'90*, pages 746–757, 1990.

31 Bernd Puchala. Asynchronous omega-regular games with partial information. In *MFCS'10*, pages 592–603, 2010.

32 R. Ramanujam and S. Simon. A communication based model for games of imperfect information. In *CONCUR'10*, pages 509–523, 2010.

33 John H. Reif. The complexity of two-player games of incomplete information. *J. Comput. Syst. Sci.*, 29(2):274–301, 1984.

34 J. Rintanen. Complexity of planning with partial observability. In *ICAPS'04*, pages 345–354, 2004.

35 M. Sipser. *Introduction to the Theory of Computation*, volume 2. Thomson Course Technology Boston, 2006.

36 Larry J. Stockmeyer and Ashok K. Chandra. Provably difficult combinatorial games. *SIAM J. Comput.*, 8(2):151–174, 1979. URL: https://doi.org/10.1137/0208013, path-doi:10.1137/0208013.

37 J. Van Benthem. Games in dynamic-epistemic logic. *Bulletin of Economic Research*, 53(4):219–248, 2001.

38 H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*. Springer, Dordecht, 2008.

39 Q. Yu, X. Wen, and Y. Liu. Multi-agent epistemic explanatory diagnosis via reasoning about actions. In *IJCAI'13*, 2013.