Lightweight Parallel Multi-Agent Epistemic Planning

Martin Cooper[1] , **Andreas Herzig**[2] , **Frédéric Maris**[1] , **Elise Perrotin**[1] , **Julien Vianey**[1]
[1]IRIT, Université Toulouse 3 - Paul Sabatier
[2]IRIT, CNRS
cooper@irit.fr, herzig@irit.fr, maris@irit.fr, Elise.Perrotin@irit.fr, Julien.Vianey@irit.fr

Abstract

We study a simple version of multi-agent epistemic planning with common knowledge where the number of parallel steps has to be minimized. We prove that this extension of classical planning is in PSPACE. We propose an encoding in PDDL and present some experiments providing evidence that it allows us to solve practical problems. The types of problems we can encode include problems in which one agent can teach another agent how to perform a task and communication problems where some information must not be revealed to some agents.

1 Introduction

Epistemic planning is important in multiagent systems. None of these approaches investigated up to now studied the construction of parallel epistemic plans. Such plans are however particularly interesting when there is more than one agent. In this paper we investigate how multiple agents can act in parallel in order to achieve a common goal.

Simultaneous actions of several agents may conflict, and several notions of interference have been proposed to characterise this (Knoblock 1994; Dimopoulos, Nebel, and Koehler 1997). Most approaches to parallel classical planning choose the framework of independent parallel actions that was introduced in the planner GRAPHPLAN (Blum and Furst 1997) where two actions interfere if either their effects are contradictory or one effect contradicts the precondition of the other. Non-interfering actions can be arranged in any sequential order with exactly the same outcome. In a previous paper we used this framework in order to solve classical planning tasks with parallel plans (Herzig, Maris, and Vianey 2019). In that work we supposed that agents always have perfect knowledge of the current state and of the occurrence of actions. We here relax this hypothesis: an agent may fail to observe the truth value of a given propositional variable. Moreover, she may not know other agents' observational capabilities. The object of these capabilities can be a propositional variables, but also the other agents' visibility; in other words, we consider higher-order visibility information. This is the general setting of planning in Dynamic Epistemic Logic DEL (Bolander and Andersen 2011; Löwe, Pacuit, and Witzel 2011). DEL combines standard epistemic logic (the static component) with event models describing actions and their perception by the agents (the dynamic component). As DEL-based planning is undecidable (Bolander and Andersen 2011), restrictions of either the static or the dynamic component were explored. Most approaches focussed on the latter. It turned out that undecidability is already the case under severe restrictions: basically, DEL planning tasks are only decidable when all actions are public (Aucher and Bolander 2013; Bolander, Jensen, and Schwarzentruber 2015; Cong, Pinchinat, and Schwarzentruber 2018), which is not the case in many real world multiagent applications.

We here simplify the static component: we replace standard epistemic logic by a lightweight version, Epistemic Logic of Observation (EL-O), which is based on the notion of observability of a propositional variable by an agent. In EL-O it is supposed that agent i knows that p is true when p is true and i observes p. Symmetrically, i knows that p is false when p is false and i observes p. Thus when i observes p then i knows either that p is true or that p is false. The other way round, when i does not observe p then i does not know whether p: both p and $\neg p$ are possible for i. This extends to higher-order observability: i may observe whether j observes p, and so on. We showed that EL-O is suitable for sequential epistemic planning (Cooper et al. 2016). We here apply it to parallel epistemic planning: we provide a reduction of EL-O-based parallel planning to classical planning, which allows us to translate planning tasks into PDDL and use classical planners. We illustrate our approach with a parallel version of the epistemic gossip problem (Cooper et al. 2019) where n agents initially each know their secret but not the others'; agents can exchange all secrets they know during a phone call to another agent; and the goal is to achieve shared knowledge of all secrets, i.e., everybody knows every secret. A parallel solution is a sequence of sets of calls. This can be generalised to shared knowledge of depth k: it can be achieved in $\mathcal{O}(k\lceil\log_2 n\rceil)$ time steps, for $n \geq 2$ (Cooper et al. 2019, Proposition 6).

The paper is organised as follows. Section 2 overviews related work and Section 3 recalls EL-O. Section 4 defines epistemic planning tasks and Section 5 their parallel solutions. We then provide a polynomial translation into classical planning tasks, leading to complexity results (Section 6). We exploit it further to encode planning tasks into PDDL (Section 7) and to use classical planners for some experiments (Section 8). Section 9 concludes.

2 Related Work

We overview existing work on decidable epistemic planning in the DEL paradigm. We focus on sequential plans: as far as we know parallel plans have not been investigated yet.

The approach of (Muise et al. 2015) is most related to ours. It is based on a lightweight epistemic logic where the scope of the epistemic operator K_i is restricted to literals that are preceded by a sequence of epistemic operators and negations (Muise et al. 2015); in other words, no conjunctions or disjunctions can occur in the scope of K_i. Such restrictions however exclude formulas such as $K_i(K_j p \lor K_j \neg p)$ expressing that agent i knows that agent j knows whether the propositional variable p is true. This is a major drawback because such formulas are fundamental in communication and more generally in any forms of interaction: a situation where agent i does not know whether or not p is the case ($\neg K_i p \land \neg K_i \neg p$) but knows that j knows ($K_i(K_j p \lor K_j \neg p)$) may lead agent i to ask j about p. In our EL-O-based approach all these formulas can be expressed.

(Kominis and Geffner 2015) keep the language of standard epistemic logic (so their language is not restricted to epistemic literals) and restrict the dynamic component. It requires that the initial state is common knowledge and that all action occurrences are either public or semi-public. This makes it impossible to account for many natural everyday situations such as gossiping.

A series of papers by Liu et col. investigates epistemic planning with common knowledge based on the situation calculus paradigm (Liu and Liu 2018; Huang et al. 2017). They represent KD45 knowledge in a particular normal form that generalises Moss' characteristic formulas. Their actions have very general effects, such as a disjunction becoming common knowledge, which requires the integration of belief update and revision operations.

Le et al. study DEL-based planning with common knowledge under compact representations of the initial epistemic state and of event models (Le et al. 2018). For the former they use what they call S5-theories (although their epistemic logic is K, not S5); for the latter they use the action language $m\mathcal{A}$, which has statements of the kind "agent i observes action a". This differs from our modelling where agents observe propositional variables, which makes it difficult to compare the two approaches. We can however also model their 'coin in the box' example which they claim no approach can deal with. A limitation of $m\mathcal{A}$ is that only literals can be announced. Agents therefore cannot communicate higher-order knowledge, as required in the generalised gossip problem. We note that they mention the issue of interfering actions but do not develop this further because it is not clear how to compute effects under $m\mathcal{A}$.

3 EL-O: Epistemic Logic of Observation

We recall the Epistemic Logic of Observation, abbreviated EL-O. Its language is a fragment of that of the dynamic epistemic logic DEL-PAO (Herzig, Lorini, and Maffre 2015).

3.1 Atoms and Introspective Atoms

Let *Prop* be a countable set of propositional variables and let *Agt* be a finite set of agents. The set of *observability operators* is

$$Obs = \{S_i : i \in Agt\} \cup \{JS\},$$

where S_i stands for individual visibility of agent i and JS stands for joint visibility of all agents. The set of all sequences of visibility operators of length at most k is noted $Obs^{\leq k}$. Then $Obs^* = \bigcup_{k \geq 0} Obs^{\leq k}$ and $Obs^+ = \bigcup_{k \geq 1} Obs^{\leq k}$. Elements of Obs^* are noted σ, σ', etc.

Visibility atoms, or atoms for short, are finite sequences of visibility operators followed by a propositional variable. The set of all atoms is

$$Atm = \{\sigma\, p : \sigma \in Obs^*, p \in Prop\}.$$

We use α, α', β,... to denote atoms. For example, $S_1\, p$ reads "1 sees the value of p"; it means that 1 knows whether p is true or false. $JS\, S_2\, q$ reads "all agents jointly see whether agent 2 sees the value of q": there is joint attention in the group of all agents concerning 2's observation of q; agent 2 may or may not see the value of q, and in both cases this is jointly observed. $S_1\, S_2\, S_3\, p$ reads "1 sees whether 2 sees whether 3 sees p". Atoms with an empty sequence of observability operators are nothing but propositional variables.

Principles of introspection play an important role in epistemic logic: when agent i knows that p then i also *knows* that she knows that p; and when agent i does not know that p then i also *knows* that she does not know that p. In our visibility-based epistemic logic, introspection can be expressed as $S_i\, S_i\, \alpha$. Likewise, joint introspection is expressed as $JS\, JS\, \alpha$. The latter implies $\sigma\, JS\, \alpha$ for every non-empty σ because joint visibility implies any nesting of individual visibility. We therefore call an atom *introspective* if it contains two consecutive S_i, or a JS that is preceded by a non-empty sequence of observability operators. The set of all introspective atoms is

$$\begin{aligned} I\text{-}Atm = {} & \{\sigma\, S_i\, S_i\, \alpha : \sigma \in Obs^* \text{ and } \alpha \in Atm\} \cup \\ & \{\sigma\, JS\, \alpha : \sigma \in Obs^+ \text{ and } \alpha \in Atm\}. \end{aligned}$$

The complement of $I\text{-}Atm$ is the set of relevant atoms: $R\text{-}Atm = Atm \setminus I\text{-}Atm$.

3.2 Atomic Consequence

We define a relation of *atomic consequence* between visibility atoms as follows:

$$\alpha \Rightarrow \beta \quad \text{if} \quad \text{either } \alpha = \beta, \text{ or } \alpha = JS\, \alpha' \text{ and } \beta = \sigma\, \alpha' \text{ for some } \sigma \in Obs^+.$$

For example, $JS\, p \Rightarrow S_i\, p$ and $JS\, p \Rightarrow JS\, S_i\, p$. The relation $\Rightarrow$ is reflexive and transitive. When $\alpha \Rightarrow \beta$, we say that α is a *cause* of β and that β is a *consequence* of α. We will ensure that atomic consequences are valid implications. We note $\alpha^{\Leftarrow}$ the set of causes of α, and $\alpha^{\Rightarrow}$ the set of its consequences. Clearly, $(p)^{\Rightarrow} = (p)^{\Leftarrow} = \{p\}$ for $p \in Prop$. Moreover, $(S_i\, p)^{\Rightarrow} = \{S_i\, p\}$, $(S_i\, p)^{\Leftarrow} = \{S_i\, p, JS\, p\}$, $(JS\, p)^{\Rightarrow} = \{\sigma\, p : \sigma \in Obs^+\}$, and $(JS\, p)^{\Leftarrow} = \{JS\, p\}$. Observe that $\alpha^{\Leftarrow}$ is always finite while $\alpha^{\Rightarrow}$ is either infinite (namely when α starts by JS) or the singleton $\{\alpha\}$ (namely when α is a propositional variable or starts by some S_i).

$s \models \alpha$	if	$\alpha \in s^{\Rightarrow} \cup I\text{-}Atm$
$s \models \neg\varphi$	if	$s \not\models \varphi$
$s \models \varphi \wedge \psi$	if	$s \models \varphi$ and $s \models \psi$

Table 1: Interpretation of formulas

Proposition 1. *The following hold for every $A, B \subseteq Atm$:*

1. $(A \cup I\text{-}Atm)^{\Rightarrow} = A^{\Rightarrow} \cup I\text{-}Atm$;
2. $A^{\Rightarrow} \cap B^{\Leftarrow} = \emptyset$ *iff* $A^{\Rightarrow} \cap B = \emptyset$ *iff* $A \cap B^{\Leftarrow} = \emptyset$.

3.3 Language of EL-O

The language of EL-O is defined by the following grammar:

$$\varphi ::= \alpha \mid \neg\varphi \mid (\varphi \wedge \varphi)$$

where α ranges over Atm. The boolean operators $\top$, $\bot$, $\vee$, $\rightarrow$ and $\leftrightarrow$ are defined in the standard way. The set of EL-O *formulas* is noted $Fml_{\textsf{EL-O}}$. The set of *relevant formulas* is $R\text{-}Fml_{\textsf{EL-O}} = \{\varphi \in Fml_{\textsf{EL-O}} : Atm(\varphi) \subseteq R\text{-}Atm\}$. The set $Atm(\varphi)$ is the set of atoms occurring in φ. For example, $Atm(JS\, q \wedge S_2\, p) = \{JS\, q, S_2\, p\}$ and $Atm(S_1\, JS\, p) = \{S_1\, JS\, p\}$. Note that neither p nor $JS\, p$ are atoms of $S_1\, JS\, p$.

The *length* of a formula is defined recursively by:

$$\ell(\sigma\, p) = \ell(\sigma) + 1$$
$$\ell(\neg\varphi) = \ell(\varphi) + 1$$
$$\ell(\varphi \wedge \varphi') = \ell(\varphi) + \ell(\varphi') + 1$$

where $\ell(\sigma)$ is the length of the finite sequence σ. For example, $\ell(S_1\, S_2\, p) = 3$ and $\ell(S_2\, p \wedge \neg p \wedge JS\, q) = 8$. If $\beta \Rightarrow \alpha$ then the length of β is less than or equal to the length of α. Moreover, the set of causes of α has at most $\ell(\alpha)$ elements: $|\alpha^{\Leftarrow}| \leq \ell(\alpha)$. It follows that the sum of the lengths of all causes of α is at most quadratic in the length of α:

Proposition 2. *For every α, $\sum_{\beta\,:\,\beta\Rightarrow\alpha} \ell(\beta) \leq \big(\ell(\alpha)\big)^2$.*

Example 1. *In the generalised gossip problem, for $Agt = \{1, \ldots, n\}$ the set of secrets is $Prop = \{s_i : i \in Agt\}$. The goal is to obtain shared knowledge of depth k:*

$$Goal^{G_k} = \bigwedge_{i \in Agt} \quad \bigwedge_{\sigma\, \in Obs^{+}, \ell(\sigma) \leq k} \sigma\, s_i.$$

Hence $Goal^{G_1} = \bigwedge_{i \in Agt} \; \bigwedge_{j \in Agt} S_j\, s_i$.

3.4 Semantics of EL-O

A *state* is a subset of the set of atoms Atm. We denote states by s, s', etc. The set of all states is $STATES = 2^{Atm}$. The set of *relevant states* is $R\text{-}STATES = 2^{R\text{-}Atm}$.

A way of guaranteeing introspection was proposed in (Herzig, Lorini, and Maffre 2015) where formulas are interpreted exclusively in *introspectively closed states*: states that contain all introspective atoms and are closed under $\Rightarrow$, i.e., sets of atoms that equal $s \cup I\text{-}Atm^{\Rightarrow}$ for some state $s \subseteq Atm$. Such introspective states being always infinite, it is not clear how to define model checking, which requires finite states. Here we work with finite models and interpret formulas in such a way that introspection is simulated.

The truth conditions for EL-O formulas are in Table 1. The condition for atoms is the only non-standard one: α is true in state s if α is introspective or $\beta \Rightarrow \alpha$ for some $\beta \in s$. Our formulation in the table uses a generalisation of atomic consequence to sets of atoms $s \subseteq Atm$: $s^{\Rightarrow} = \bigcup_{\alpha \in s} \alpha^{\Rightarrow}$.

Example 2. *In the initial gossiping state (in which secrets may or may not be true) every agent only knows her own secret. Therefore $s_0^{G_1} = \{S_i\, s_i : i \in Agt\} \cup A_1$ where A_1 is some subset of $\{s_i : i \in Agt\}$. Then $s_0^{G_1} \models S_i\, s_i$ and $s_0^{G_1} \models \bigwedge_{j \neq i} \neg S_i\, s_j$ for every $i \in Agt$. Although $s_0^{G_1}$ does not contain $S_i\, S_i\, s_j$ we have $s_0^{G_1} \models S_i\, S_i\, s_j$.*

Given a set of states $St \subseteq STATES$, we say that a formula φ is *valid in St* if $s \models \varphi$ for every $s \in St$; when $s \models \varphi$ for some $s \in St$ then we say that φ is *satisfiable in St*. Clearly, an atom α is valid in the set of all states $STATES$ if and only if it is introspective. Moreover, atomic consequences are valid in $STATES$: if $\alpha \Rightarrow \beta$ then $\alpha \rightarrow \beta$ is valid in $STATES$.

Remark 1. When Agt is a singleton then $S_i\, p \wedge \neg JS\, p$ is satisfiable. While this anomaly could be taken care of by a modification of the semantics, we do not do so for the sake of readability and content ourselves with the observation that the JS operator is superfluous when there is only one agent.

Proposition 3. *For every $\varphi \in Fml_{\textsf{EL-O}}$ there is a $\varphi' \in R\text{-}Fml_{\textsf{EL-O}}$ such that $\varphi \leftrightarrow \varphi'$ is valid in STATES. Moreover, for every $s \in STATES$, there is a $s' \in R\text{-}STATES$ such that $s \models \varphi$ iff $s' \models \varphi$ for every $\varphi \in Fml_{\textsf{EL-O}}$.*

Classical semantics for $Fml_{\textsf{EL-O}}$ is recovered by changing the truth condition for atoms: $s \models^{\textsf{CPC}} \alpha$ if $\alpha \in s$.

Proposition 4. *For $\varphi \in R\text{-}Fml_{\textsf{EL-O}}$ and $s \in R\text{-}STATES$, $s \models \varphi$ iff $s^{\Rightarrow} \cap Atm(\varphi) \models \varphi$ iff $s^{\Rightarrow} \cap Atm(\varphi) \models^{\textsf{CPC}} \varphi$.*

4 Action Descriptions and Simple Epistemic Planning tasks

We assume that actions are deterministic and have conditional effects that are described by add- and delete-lists. Such effects are crucial in epistemic planning: when an agent performs an action then the effects on another agent's epistemic state typically depend on whether that agent sees the variables that are modified by the action.

4.1 Action Descriptions

An *action description* is a pair $\mathsf{a} = \langle pre(\mathsf{a}), eff(\mathsf{a}) \rangle$ where $pre(\mathsf{a})$ is a relevant formula from $R\text{-}Fml_{\textsf{EL-O}}$ (the *precondition* of a) and

$$eff(\mathsf{a}) \subseteq R\text{-}Fml_{\textsf{EL-O}} \times 2^{R\text{-}Atm} \times 2^{R\text{-}Atm}$$

is the set of *conditional effects* of a, describing which atoms the action may add or remove from the current state under additional conditions. For a triple

$$ce = \langle cnd(ce), ceff^{+}(ce), ceff^{-}(ce) \rangle$$

in $eff(\mathsf{a})$, $cnd(ce)$ is the condition of ce, $ceff^{+}(ce)$ are the added atoms, and $ceff^{-}(ce)$ are the deleted atoms. We require effects to be *consistent*: we suppose that for every

$ce_1, ce_2 \in \mathit{eff}(\mathsf{a})$, if $\mathit{ceff}^+(ce_1) \cap (\mathit{ceff}^-(ce_2))^{\Leftarrow} \neq \emptyset$ then $pre(\mathsf{a}) \wedge cnd(ce_1) \wedge cnd(ce_2)$ is unsatisfiable in $\mathsf{EL\text{-}O}$. That is, we exclude actions with conditional effects $ce_1, ce_2 \in \mathit{eff}(\mathsf{a})$ and $\alpha_1 \in \mathit{ceff}^+(ce_1)$ and $\alpha_2 \in \mathit{ceff}^-(ce_2)$ such that $\alpha_1 \Rightarrow \alpha_2$. In other words, when $pre(\mathsf{a})$ and their triggering conditions $cnd(ce_1)$ and $cnd(ce_2)$ are jointly satisfiable then two conditional effects of a cannot conflict. This in particular forbids conditional effects $ce \in \mathit{eff}(\mathsf{a})$ with $\mathit{ceff}^+(ce) \cap \mathit{ceff}^-(ce) \neq \emptyset$ and $pre(\mathsf{a}) \wedge cnd(ce)$ satisfiable.

We disregard introspective atoms in the definition of actions because they are true at every state: adding or deleting them from a state does not change what is true in that state.

Example 3. *In the original gossip problem G_1 where the goal is to obtain shared knowledge of depth 1, $\mathsf{Call}^i_j = \langle pre(\mathsf{Call}^i_j), \mathit{eff}(\mathsf{Call}^i_j)\rangle$ with $pre(\mathsf{Call}^i_j) = \top$ and*

$$\begin{aligned}\mathit{eff}(\mathsf{Call}^i_j) = \{&\langle S_i\, s_1, \{S_j\, s_1\}, \emptyset\rangle, \langle S_j\, s_1, \{S_i\, s_1\}, \emptyset\rangle,\\ &\dots,\\ &\langle S_i\, s_n, \{S_j\, s_n\}, \emptyset\rangle, \langle S_j\, s_n, \{S_i\, s_n\}, \emptyset\rangle\}.\end{aligned}$$

That is, a call has two conditional effects per secret: if i sees a secret then that secret becomes visible to j, and vice versa.

Example 4. *In the generalised gossip problem G_k the precondition is $pre(\mathsf{Call}^i_j) = \top$ as before, and for every $0 \leq m < k$, $\sigma_m \in Obs^{\leq m}$ and $r \in Agt$ there is a conditional effect $ce \in \mathit{eff}(\mathsf{Call}^i_j)$ of the form:*

$$\begin{aligned}cnd(ce) &= S_i\, \sigma_m s_r \vee S_j\, \sigma_m s_r,\\ \mathit{ceff}^+(ce) &= \{\sigma\, S_i\, \sigma_m s_r \,:\, \sigma \in \{S_i, S_j\}^{\leq k-m-1}\} \cup\\ &\qquad \{\sigma\, S_j\, \sigma_m s_r \,:\, \sigma \in \{S_i, S_j\}^{\leq k-m-1}\}\\ &= \{\sigma\, \sigma_m s_r \,:\, \sigma \in \{S_i, S_j\}^{\leq k-m}\},\\ \mathit{ceff}^-(ce) &= \emptyset,\end{aligned}$$

where $\{S_i, S_j\}^{\leq k-m}$ denotes the set all sequences of observability operators S_i and S_j of length at most $k-m$. Hence a call achieves common knowledge of i and j up to level k of all secrets one of them knows. The set of all actions is $Act^{G_k} = \{\mathsf{Call}^i_j \,:\, i, j \in Agt, i \neq j\}$. All Call^i_j satisfy our consistency condition because they have no negative effects, which makes conflicts impossible.

The *length* of an action description is

$$\ell(\mathsf{a}) = \ell(pre(\mathsf{a})) + \sum_{ce \in \mathit{eff}(\mathsf{a})} \left(\begin{array}{c} \ell(cnd(ce)) \\ +\left(\sum_{\alpha \in \mathit{ceff}^+(ce)} \ell(\alpha)\right) \\ +\left(\sum_{\alpha \in \mathit{ceff}^-(ce)} \ell(\alpha)\right) \end{array} \right).$$

4.2 Simple Epistemic Planning Tasks

A *simple epistemic planning task* is a triple

$$\mathcal{P} = \langle Act, s_0, Goal\rangle$$

where Act is a finite set of consistent actions, $s_0 \in R\text{-}STATES = 2^{R\text{-}Atm}$ is a finite state (the initial state) and $Goal \in R\text{-}Fml_{\mathsf{EL\text{-}O}}$ is an $\mathsf{EL\text{-}O}$ formula without introspective atoms. (We again disregard introspective atoms as they have no effect on the truth of a formula.)

Example 5. *The planning task that corresponds to the original gossip problem is $G_1 = \langle Act^{G_1}, s_0^{G_1}, Goal^{G_1}\rangle$ with*

- *$Act^{G_1} = \{\mathsf{Call}^i_j \,:\, i, j \in Agt$ and $i \neq j\}$ (cf. Example 3),*
- *$s_0^{G_1} = \{S_i\, s_i \,:\, i \in Agt\} \cup A_1$ for $A_1 \subseteq \{s_i \,:\, i \in Agt\}$,*
- *$Goal^{G_1} = \bigwedge_{i \in Agt} \bigwedge_{j \in Agt} S_j\, s_i$.*

The set of atoms of a simple epistemic planning task is

$$Atm(\mathcal{P}) = \left(\bigcup_{\mathsf{a} \in Act} Atm(\mathsf{a}) \right) \cup s_0 \cup Atm(Goal)$$

and its length is $\ell(\mathcal{P}) = \ell(s_0) + \ell(Goal) + \sum_{\mathsf{a} \in Act} \ell(\mathsf{a})$.

Solutions to simple epistemic planning tasks can be either sequential plans or parallel plans. We focus on the latter in the rest of the paper.

5 Parallel Epistemic Planning with EL-O

A parallel plan is a sequence of *steps* each of which is a set of actions that are executed simultaneously. Actions in a step should not conflict: we start by determining the conditions of parallel executability of a set of actions in a state, following the $\forall$-step semantics and the notion of interference in a state of (Rintanen, Heljanko, and Niemelä 2006).

5.1 Semantics of a Single Action

We define the semantics of an action a in terms of a partial function τ_{a} on relevant states. The function τ_{a} is defined at s if $s \models pre(\mathsf{a})$. In that case we say that a is executable at s and stipulate:

$$\tau_{\mathsf{a}}(s) = \left(s \setminus \bigcup_{\substack{ce \in \mathit{eff}(\mathsf{a}),\\ s \models cnd(ce)}} (\mathit{ceff}^-(ce))^{\Leftarrow} \right) \cup \bigcup_{\substack{ce \in \mathit{eff}(\mathsf{a}),\\ s \models cnd(ce)}} (\mathit{ceff}^+(ce))^{\Rightarrow}.$$

That is, if the precondition of a is satisfied then a removes negative effects of all those conditional effects ce that 'fire', i.e., whose triggering conditions are satisfied, plus their causes; and it adds the positive effects of ce plus their consequences. As a's description is consistent it does not matter in which order we apply negative and positive effects.

5.2 Consistency of a Set of Actions at a State

Intuitively, in any parallel plan, no effect of an action should be destroyed by an effect of another action executed in parallel, and no precondition of an action should be destroyed by an effect of another action executed in parallel. Let us define these two consistency criteria formally.

Actions a_1 and a_2 *have no contradictory effects at s* if:

1. a_1 and a_2 are executable at s;
2. for every $ce_1 \in \mathit{eff}(\mathsf{a}_1)$ and $ce_2 \in \mathit{eff}(\mathsf{a}_2)$, if $s \models cnd(ce_1) \wedge cnd(ce_2)$ then $\mathit{ceff}^+(ce_1) \cap \mathit{ceff}^-(ce_2) = \emptyset$.

It follows that the description of the individual action a is consistent (see Section 4.1) iff a has non-contradictory effects with itself in every s such that $s \models pre(\mathsf{a})$.

In the sequel, "s and s' agree on φ" means that either $s \models \varphi$ and $s' \models \varphi$, or $s \not\models \varphi$ and $s' \not\models \varphi$. Then we say that two different actions a_1 and a_2 *have no cross-interaction at s* if the following hold:

1. s and $\tau_{\mathsf{a}_1}(s)$ agree on $pre(\mathsf{a}_2)$ and on the condition $cnd(ce_2)$ of every conditional effect $ce_2 \in \mathit{eff}(\mathsf{a}_2)$;
2. s and $\tau_{\mathsf{a}_2}(s)$ agree on $pre(\mathsf{a}_1)$ and on the condition $cnd(ce_1)$ of every conditional effect $ce_1 \in \mathit{eff}(\mathsf{a}_1)$.

Here are two examples from parallel classical planning: when there are no contradictory effects then one cannot pick up and drop the same object in parallel; when there is no cross interaction then two agents cannot pick up the same object in parallel. Gossiping calls have neither contradictory effects nor cross interaction at any state. The former is the case because there are no negative effects and the latter because the precondition of all calls is $\top$.

Putting things together, we say that a set of actions $\mathsf{A} = \{\mathsf{a}_1, \ldots, \mathsf{a}_m\}$ is *consistent in state* s if for every $\mathsf{a}_i, \mathsf{a}_j \in \mathsf{A}$ such that $\mathsf{a}_i \neq \mathsf{a}_j$,

1. a_i and a_j have no contradictory effects in s;
2. a_i and a_j have no cross interaction in s.

Example 6. *Any set of gossiping calls* Call^i_j *is consistent in any state. Therefore conference calls* $\{\mathsf{Call}^i_j, \mathsf{Call}^i_r\}$ *where* i *calls* j *and* r *at the same time are consistent, making the parallel gossiping task solvable in one step.*

One way to exclude conference calls is to replace Call^i_j *by* $\mathsf{Startcall}^i_j$ *plus a single* $\mathsf{Endcalls}$ *action as follows:*

$$
\begin{aligned}
pre(\mathsf{Startcall}^i_j) &= \mathit{free}_i \wedge \mathit{free}_j,\\
\mathit{eff}(\mathsf{Startcall}^i_j) &= \mathit{eff}(\mathsf{Call}^i_j) \cup \{\langle \top, \emptyset, \{\mathit{free}_i, \mathit{free}_j\}\rangle\},\\
pre(\mathsf{Endcalls}) &= \top,\\
\mathit{eff}(\mathsf{Endcalls}) &= \{\langle \top, \{\mathit{free}_i : i \in Agt\}, \emptyset\rangle\},
\end{aligned}
$$

and to add all free_i *to the initial state. Then there is no state where a set of actions with conference calls is consistent:* $\mathsf{Startcall}^i_j$ *and* $\mathsf{Startcall}^i_r$ *have cross interaction at any state satisfying* $\mathit{free}_i \wedge \mathit{free}_j \wedge \mathit{free}_r$*.*

While this solution is natural (agents cannot call two agents at a time because they are no longer available once they have begun a call), splitting calls into two separate actions artificially doubles the number of steps in an optimal solution. Another possibility that avoids the $\mathsf{Endcalls}$ *action is to replace all* Call^i_j *by* Tcall^i_j*, with:*

$$
\begin{aligned}
pre(\mathsf{Tcall}^i_j) &= \top,\\
\mathit{eff}(\mathsf{Tcall}^i_j) &= \mathit{eff}(\mathsf{Call}^i_j) \cup\\
&\quad \{\langle tg_i, \emptyset, \{tg_i\}\rangle\} \cup \{\langle \neg tg_i, \{tg_i\}, \emptyset\rangle\} \cup\\
&\quad \{\langle tg_j, \emptyset, \{tg_j\}\rangle\} \cup \{\langle \neg tg_j, \{tg_j\}, \emptyset\rangle\}.
\end{aligned}
$$

Here any two calls involving i *each toggles the value of* tg_i*, which makes that these calls have cross interaction at any state satisfying their preconditions.*

5.3 Semantics of a Consistent Set of Actions

A set of actions $\mathsf{A} = \{\mathsf{a}_1, \ldots, \mathsf{a}_m\}$ determines a partial function τ_{A} from $R\text{-}STATES^{\Rightarrow}$ to $R\text{-}STATES^{\Rightarrow}$, where $R\text{-}STATES^{\Rightarrow} = \{s^{\Rightarrow} : s \in R\text{-}STATES\}$. The function τ_{A} is defined at s if every $\mathsf{a}_i \in \mathsf{A}$ is executable at s and A is consistent in s. When τ_{A} is defined at s then:

$$
\begin{aligned}
\tau_{\mathsf{A}}(s) = \Bigg(s \setminus \Bigg(\bigcup_{\mathsf{a} \in \mathsf{A}, ce \in \mathit{eff}(\mathsf{a}), s \models cnd(ce)} \big(\mathit{ceff}^{-}(ce)\big)^{\Leftarrow} \Bigg)\Bigg)\\
\cup \Bigg(\bigcup_{\mathsf{a} \in \mathsf{A}, ce \in \mathit{eff}(\mathsf{a}), s \models cnd(ce)} \big(\mathit{ceff}^{+}(ce)\big)^{\Rightarrow} \Bigg).
\end{aligned}
$$

When a_1 and a_2 are consistent in s then they can be interleaved arbitrarily: we have $\tau_{\{\mathsf{a}_1, \mathsf{a}_2\}}(s) = \tau_{\mathsf{a}_2}(\tau_{\mathsf{a}_1}(s)) = \tau_{\mathsf{a}_1}(\tau_{\mathsf{a}_2}(s))$. More generally:

Proposition 5. *If* $\mathsf{a} \in \mathsf{A}$ *is consistent in* s *with any other action in* A *then* $\tau_{\mathsf{A}}(s) = \tau_{\mathsf{a}}(\tau_{\mathsf{A} \setminus \mathsf{a}}(s)) = \tau_{\mathsf{A} \setminus \mathsf{a}}(\tau_{\mathsf{a}}(s))$.

5.4 Solvability by Parallel Plans

A state s is *reachable by a parallel plan* from a state $s_0 \in R\text{-}STATES^{\Rightarrow}$ via a set of actions Act if there is a sequence $\langle \mathsf{A}_1, \ldots, \mathsf{A}_m \rangle$ of steps and a sequence of states $\langle s_0, \ldots, s_m \rangle$ with $m \geq 0$ such that $s = s_m$ and $\tau_{\mathsf{A}_k}(s_{k-1}) = s_k$ for every k such that $1 \leq k \leq m$.

A simple epistemic planning task $\langle Act, s_0, Goal \rangle$ is *solvable by a parallel plan* if there is at least one state s that is reachable by a parallel plan from $s_0^{\Rightarrow}$ via Act such that $s \models Goal$; otherwise it is unsolvable by a parallel plan. Solvability by a sequential plan is the special case where the parallel plan is a sequence of singletons.

Example 7. *Let* G_1' *be modification of* G_1 *that is obtained by replacing the actions* Call^i_j *by* Tcall^i_j *of Example 6. Then* G_1' *can be solved in* $\lceil \log_2 n \rceil$ *steps of parallel calls if the number of agents* n *is even, and in* $\lceil \log_2 n \rceil + 1$ *steps if* n *is odd (Bavelas 1950; Landau 1954; Knödel 1975; Cooper et al. 2019). For instance, for* $n = 4$ *the parallel plan* $\langle \{\mathsf{Tcall}^1_2, \mathsf{Tcall}^3_4\}, \{\mathsf{Tcall}^1_3, \mathsf{Tcall}^2_4\} \rangle$ *is a solution of* G_1' *with* 2 *steps.*

6 Translation into Classical Planning and Complexity

We now translate simple epistemic planning into classical planning. There, solvability by a parallel plan and by a sequential plan are equivalent under $\forall$-Step semantics, and both are PSPACE-complete (Bylander 1994). Our translation is polynomial, so the solvability of simple epistemic planning tasks is in PSPACE. It also gives us an encoding into PDDL, which allows us to use classical planners in sections 7 and 8. For a bounded horizon planning task we can translate into DL-PPA model checking and use the PSPACE membership result of (Herzig, Maris, and Vianey 2019).

6.1 Two Versions of Classical Planning

There are two possibilities to define classical planning in our context. Version 1 amounts to epistemic planning restricted to the fragment of the language of $\mathsf{EL\text{-}O}$ without S_i and JS: none of them can occur in classical action descriptions and planning tasks. It immediately follows that $\mathsf{EL\text{-}O}$-based planning is PSPACE hard because classical planning is so.

Version 2 of classical planning keeps the language of EL-O but changes the semantics by weakening the consistency condition for action descriptions: a version 2 classical planning task is a triple $\mathcal{P} = \langle Act, s_0, Goal\rangle$ such that for every $ce_1, ce_2 \in \mathit{eff}(\mathsf{a})$, if $\mathit{ceff}^+(ce_1) \cap \mathit{ceff}^-(ce_2) \neq \emptyset$ then $pre(\mathsf{a}) \wedge cnd(ce_1) \wedge cnd(ce_2)$ is unsatisfiable in Classical Propositional Calculus CPC. We then define the partial functions $\tau_{\mathsf{a}}^{\mathsf{CPC}}$ as follows. First, $\tau_{\mathsf{a}}^{\mathsf{CPC}}$ is defined if $s \models^{\mathsf{CPC}} pre(\mathsf{a})$; second, the resulting state is obtained without closing under atomic causes and consequences:

$$\tau_{\mathsf{a}}^{\mathsf{CPC}}(s) = \left(s \setminus \left(\bigcup_{ce \in \mathit{eff}(\mathsf{a}), s \models^{\mathsf{CPC}} cnd(ce)} \mathit{ceff}^-(ce)\right)\right) \cup \left(\bigcup_{ce \in \mathit{eff}(\mathsf{a}), s \models^{\mathsf{CPC}} cnd(ce)} \mathit{ceff}^+(ce)\right).$$

From there we modify the definitions of consistency of a set of actions and of $\tau_{\mathsf{A}}^{\mathsf{CPC}}$ in a similar manner, removing all atomic causes and consequences of sets and requiring conditions to be satisfied classically. A classical planning task is classically solvable if a goal state is reachable from the initial state via a set of actions Act, with the difference that reachability is now defined in terms of the function $\tau_{\mathsf{A}}^{\mathsf{CPC}}$.

6.2 Expansion of Planning Tasks

Let $\mathcal{P} = \langle Act, s_0, Goal\rangle$ be a simple epistemic planning task. Its *expansion* is obtained by closing the initial state and the action descriptions under the atomic causes and consequences that are relevant for $\mathcal{P}$:

$$\begin{aligned}\mathsf{Exp}(\mathcal{P}) = \langle \{\langle pre(\mathsf{a}), \mathsf{Exp}_{\mathcal{P}}(\mathit{eff}(\mathsf{a}))\rangle \ :\ \langle pre(\mathsf{a}), \mathit{eff}(\mathsf{a})\rangle \in Act\},\\ s_0^{\Rightarrow} \cap Atm(\mathcal{P}),\\ Goal\rangle,\end{aligned}$$

where the expansion of a conditional effect is defined as:

$$\begin{aligned}\mathsf{Exp}_{\mathcal{P}}(\mathit{eff}(\mathsf{a})) = \big\{\big\langle & cnd(ce),\\ & \big(\mathit{ceff}^+(ce)\big)^{\Rightarrow} \cap Atm(\mathcal{P}),\\ & \big(\mathit{ceff}^-(ce)\big)^{\Leftarrow} \cap Atm(\mathcal{P})\big\rangle \ :\ ce \in \mathit{eff}(\mathsf{a})\big\}.\end{aligned}$$

Proposition 6. *Let $\mathcal{P} = \langle Act, s_0, Goal\rangle$ be a simple epistemic planning task. Then $\mathcal{P}$ is solvable if and only if its expansion $\mathsf{Exp}(\mathcal{P})$ is classically solvable.*

Proof. For the sake of brevity we only sketch the proof. Let

$$R\text{-}STATES|_{\mathcal{P}} = \{s \cap Atm(\mathcal{P}) \ :\ s \in R\text{-}STATES^{\Rightarrow}\}.$$

We define a semantics of actions *relative to* $\mathcal{P}$ in the following manner: if $s \in R\text{-}STATES|_{\mathcal{P}}$ and $\mathsf{A} = \{\mathsf{a}_1, \ldots, \mathsf{a}_m\} \subseteq Act$ is a consistent set of actions then $\tau_{\mathsf{A}}^{\mathcal{P}}(s)$ is defined iff for all $\mathsf{a} \in \mathsf{A}$ we have $s \models pre(\mathsf{a})$, and in that case

$$\tau_{\mathsf{A}}^{\mathcal{P}}(s) = \left(s \setminus \left(\bigcup_{\substack{\mathsf{a} \in \mathsf{A}, ce \in \mathit{eff}(\mathsf{a}),\\ s \models cnd(ce)}} \left(\big(\mathit{ceff}^-(ce)\big)^{\Leftarrow} \cap Atm(\mathcal{P})\right)\right)\right) \cup \left(\bigcup_{\substack{\mathsf{a} \in \mathsf{A}, ce \in \mathit{eff}(\mathsf{a}),\\ s \models cnd(ce)}} \left(\big(\mathit{ceff}^+(ce)\big)^{\Rightarrow} \cap Atm(\mathcal{P})\right)\right).$$

It is easily shown that if $s \in R\text{-}STATES|_{\mathcal{P}}$ and A is consistent in s then $\tau_{\mathsf{A}}^{\mathcal{P}}(s) \in R\text{-}STATES|_{\mathcal{P}}$.

By Proposition 4, for any state $s \in R\text{-}STATES^{\Rightarrow}$ and set of actions $\mathsf{A} \subseteq Act$, $\tau_{\mathsf{A}}(s)$ is defined iff $\tau_{\mathsf{A}}^{\mathcal{P}}\big(s \cap Atm(\mathcal{P})\big)$ is defined, and in that case $\tau_{\mathsf{A}}^{\mathcal{P}}\big(s \cap Atm(\mathcal{P})\big) = \tau_{\mathsf{A}}(s) \cap Atm(\mathcal{P})$. We can then extend this result to any sequence of steps, i.e., to every parallel plan: for any state $s \in R\text{-}STATES^{\Rightarrow}$, there exists a state s' reachable from s via $\langle \mathsf{A}_1, \ldots, \mathsf{A}_m\rangle$ iff there exists a state s'' that is $\mathcal{P}$-reachable from $s \cap Atm(\mathcal{P})$ via this same sequence of sets of actions, where $\mathcal{P}$-reachability is defined in the natural way following the semantics of actions relative to $\mathcal{P}$, and in that case $s'' = s' \cap Atm(\mathcal{P})$. In particular s' and s'' agree on $Goal$, and therefore $\mathcal{P}$ is solvable iff the planning task $\langle Act, s_0^{\Rightarrow} \cap Atm(\mathcal{P}), Goal\rangle$ is $\mathcal{P}$-solvable on $R\text{-}STATES|_{\mathcal{P}}$, where $\mathcal{P}$-solvability is once again defined in the natural manner following the semantics of actions relative to $\mathcal{P}$.

The expansion of $\mathcal{P}$ then 'spells out' the definition of the functions $\tau_{\mathsf{A}}^{\mathcal{P}}$ for $\mathsf{A} \subseteq Act$. Moreover, Proposition 4 tells us that for any $\varphi \in Atm(\mathcal{P})$, if $s \in R\text{-}STATES|_{\mathcal{P}}$ then $s \models \varphi$ iff $s \models^{\mathsf{CPC}} \varphi$. This gives us equivalence with classical planning. □

6.3 Complexity

By Proposition 2, the length of the expansion of epistemic planning tasks $\mathcal{P}$ is polynomial in the length of $\mathcal{P}$: $\ell(\mathsf{Exp}(\mathcal{P})) \leq (\ell(\mathcal{P}))^2$. Then PSPACE membership follows from Proposition 6. Hardness is the case because version 1 classical planning is a particular case of simple epistemic planning, as we have observed in Section 6.1. As for the lower bound, it follows because classical planning tasks (version 1) are a special case of epistemic planning tasks.

Proposition 7. *The problem of deciding solvability of a simple epistemic planning task and its bounded horizon version are both* PSPACE *complete.*

7 Encoding into PDDL

In order to be able to use classical planners we encode simple epistemic planning tasks into the Planning Domain Definition Language PDDL (McDermott et al. 1998). Fortunately, almost all planners from the 2018 International Planning Competition (IPC 2018)[1] handle conditional effects and negative preconditions, and most of them handle disjunctive preconditions.

7.1 Encoding of Formulas

When encoding a planning task into PDDL, some PDDL requirement flags have to be set depending on the form of conditions $cnd(ce)$ of conditional effects ce of actions as well as on the form of the formula $Goal$:

- the default flag `:strips` for conjunctions;
- the flag `:negative-preconditions` for negations;
- the flag `:disjunctive-preconditions` for disjunctions (if used to simplify writing) and negations of conjunctions.

[1] https://ipc2018-classical.bitbucket.io/

action	preconditions	conditional effects
$\mathsf{DoTask}_{i,t,k}$	$free_i \land S_i\, skill_k \land needs_{t,k}$	$\{\langle\top, \{done_t\}, \{free_i\}\rangle\}$
$\mathsf{Teach}_{i,j,k}$	$S_i\, skill_k \land (free_i \lor teaching_{i,k})$	$\{\langle\top, \{S_j\, skill_k, teaching_{i,k}\}, \{free_i, free_j\}\rangle\}$
$\mathsf{EndStep}$	$\top$	$\{\langle\top, \{free_i : i \in Agt\},$ $\{teaching_{i,k} : i \in Agt, skill_k \in Skills\}\rangle\}$
$\mathsf{DoTask}_{i,t,\ell}$	$free_i \land mdone_\ell \land S_i\, mdone_\ell \land \neg mdone_{\ell+1}$	$\{\langle\top, \{tdone_t\}, \{free_i\}\rangle\}$
$\mathsf{DoMeeting}_\ell$	$\bigwedge_{i \in Agt} free_i$	$\{\langle\top, \{mdone_\ell, JS\, mdone_\ell\}, \{free_i : i \in Agt\}\rangle\}$
$\mathsf{EndStep}$	$\top$	$\{\langle\top, \{free_i : i \in Agt\}, \emptyset\rangle\}$

Table 2: Action descriptions for the management task (top) and the meeting task (bottom)

For a formula φ without introspective atoms, we define a recursive function $f(\varphi)$ which returns the encoding of φ into PDDL:

$$f(S_{i_1} \dots S_{i_m} p) = \begin{cases} \texttt{(p)} & \text{if } m = 0 \\ \texttt{(\textbf{S-m} i1 ... im p)} & \text{otherwise} \end{cases}$$

$$f(JS\, S_{i_1} \dots S_{i_m} p) = \begin{cases} \texttt{(\textbf{JS} p)} & \text{if } m = 0 \\ \texttt{(\textbf{JS-m} i1 ... im p)} & \text{otherwise} \end{cases}$$

$$f(\neg\varphi) = \texttt{(not } f(\varphi)\texttt{)}$$

$$f(\varphi_1 \land \varphi_2) = \texttt{(and } f(\varphi_1)\ f(\varphi_2)\texttt{)}$$

with $p \in Prop$, $m \geq 0$, and $i_1, \dots, i_m \in Agt$. In words, a visibility atom $\alpha = S_{i_1} \dots S_{i_m} p$ is encoded by a special fluent with $m{+}1$ parameters. If $m = 0$, then the propositional variable p is encoded as a fluent without parameters. A visibility atom $\alpha = JS\, S_{i_1} \dots S_{i_m} p$ is encoded by a special fluent with $m{+}1$ parameters. If $m = 0$ then a special fluent is encoded with the propositional variable p as unique parameter.

The formula $Goal$ and the preconditions of every action are $\mathsf{EL\text{-}O}$ formulas and are encoded as $f(Goal)$ etc. The initial state s_0 is encoded as a set of fluents, encoding each $\alpha \in s_0$ as $f(\alpha)$.

7.2 Encoding of Actions

For every action a and every conditional effect $ce \in \mathit{eff}(\mathsf{a})$ with $\big(\mathit{ceff}^+(ce)\big)^{\Rightarrow} \cap Atm(\mathcal{P}) = \{\alpha_1, \dots, \alpha_m\}$ and $\big(\mathit{ceff}^-(ce)\big)^{\Leftarrow} \cap Atm(\mathcal{P}) = \{\beta_1, \dots, \beta_\ell\}$ we add the conditional effect:

```
(when f(cnd(ce))
   (and f(α1) ... f(αm)
        (not f(β1)) ... (not f(βℓ))))
```

Example 8 (Example 3, ctd.)**.** *The action* Call_2^1 *is encoded as:*

```
(:action call-1-2
  :effect (and
    (when (or (S-1 1 s1) (S-1 2 s1))
      (and (S-1 1 s1) (S-1 2 s1)))
    ...
    (when (or (S-1 1 sn) (S-1 2 sn))
      (and (S-1 1 sn) (S-1 2 sn)))))
```

This is the direct encoding of a call into PDDL. It can be generalised to any i and j by:

```
(:action call
  :parameters (?i ?j)
  :effect (and (forall (?s)
    (and
      (when (or (S-1 ?i ?s) (S-1 ?j ?s))
        (and (S-1 ?i ?s) (S-1 ?j ?s)))))))
```

8 Experimental Results

To experiment with simple epistemic planning tasks we considered some benchmark tasks using planners from the international planning competition IPC 2018. The experiments were done using three classical planners of the optimal track of IPC 2018: Planning-PDBs, Complementary1 and Complementary2.

We are in a multi-agent setting where agents execute actions simultaneously in steps. We designed our problems so that it is beneficial (in terms of minimising the number of steps) that agents cooperate and perform tasks in parallel. We want to find the shortest plan in which the agents cooperate in this way. For this, we used cost-optimal planners from the classical tracks of the competition. Given a cost function defined for all actions of the planning task, these planners return sequential plans minimizing total cost. To simulate parallel steps and ensure independence of actions in one step, we use in each experiment an $\mathsf{EndStep}$ action. That is, we adapt the action descriptions such as to ensure that all actions executed between two $\mathsf{EndStep}$ actions are applicable in parallel, and an $\mathsf{EndStep}$ action must occur in order to simulate the following parallel step. Moreover, we give zero cost to all actions but $\mathsf{EndStep}$, therefore effectively counting the number of steps and guaranteeing that our experiments return optimal parallel plans.

The experiments all gave similar results for the problems described below, so we choose to show the results for Planning-PDBs. The results were obtained on a GNU/Linux machine running on a 3.6 to 4.4 GHz CPU with 32 GB of RAM and a 30 minutes time limit (wall clock time). All the execution times given below are CPU times.

8.1 Parallel Gossip Task

We use the gossip modelling of Example 6 with the actions $\mathsf{Startcall}_j^i$ and $\mathsf{Endcalls}$, which here becomes our $\mathsf{EndStep}$.

Figure 1 shows the difference between parallel and sequential gossiping. We can see that with only two agents (blue and longest curve) the execution times are very similar but when, for a fixed depth, the number of agents increases then the execution time increases, too. Thus, we have less results for parallel gossiping than for sequential gossiping.

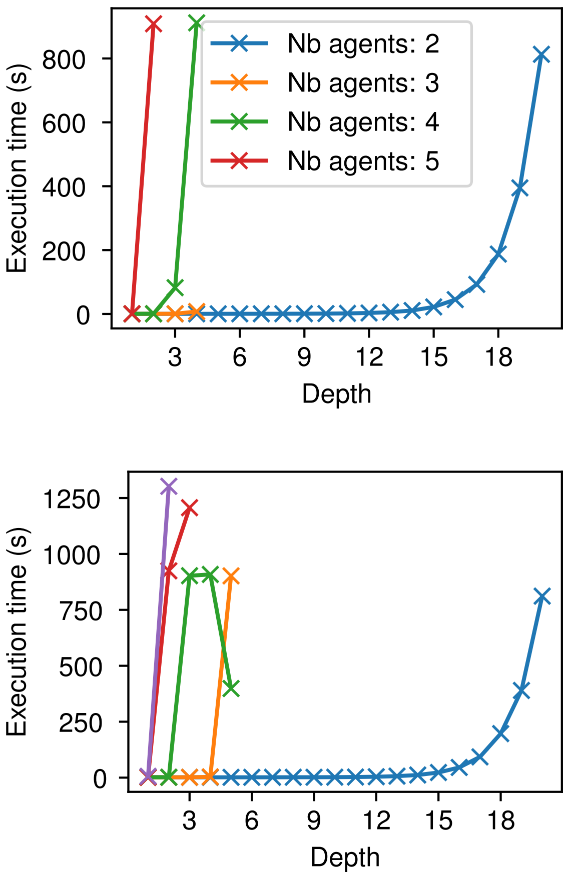

Figure 1: Parallel gossip: time to find an optimal parallel plan vs. epistemic depth (top) and time to find an optimal sequential plan vs. epistemic depth (bottom)

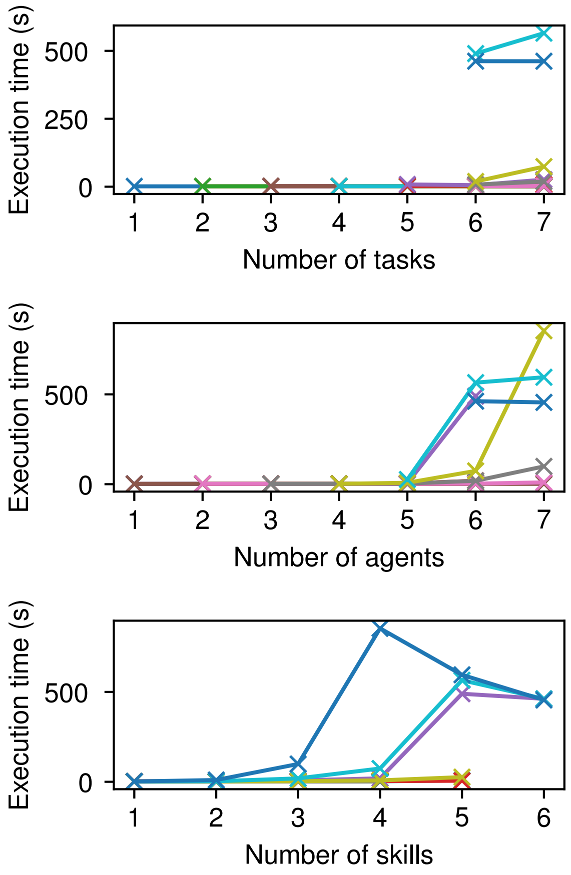

Figure 2: Management: time to find an optimal parallel plan vs. number of tasks, agents or skills (for different fixed values of the two other parameters)

The largest parallel gossiping task solved under 1800s was for 5 agents and an epistemic depth of 2. This task has 125 atoms, 26 actions and 1920 conditional effects. For sequential gossiping the largest solved planning task was for 8 agents and an epistemic depth of 1. This task has 56 atoms, 64 actions and 1512 conditional effects.

8.2 Management Task

For the next benchmark planning task, we consider that a set of tasks has to be performed by the set of agents and that the execution of any task by an agent requires the agent to have a corresponding skill. In the beginning, the skills are split between the agents so that a particular agent may lack some of the skills required to perform particular tasks. This can limit the ability of the agents to perform tasks in parallel. However, there are also actions which allow agents to teach some of their skills to other agents.

Initially all agents are free and the state is $s_0 = \{free_i : i \in Agt\} \cup S$ for some subset S of $\{S_i\, skill_k : i \in Agt, k \in Skills\} \cup \{needs_{t,k} : t \in Tasks, k \in Skills\}$. The goal is to perform all tasks: $Goal = \bigwedge_{t \in Tasks} done_t$.

The action descriptions are listed in the upper half of Table 2. For each pair of agents i,j and skill $skill_k$, the action $\mathsf{Teach}_{i,j,k}$ can be executed when i knows the skill she is teaching and is either free or is teaching the same skill to another agent. Its effects are that j knows the skill, that i and j are both no longer free, and that i is still available to teach $skill_k$ to other agents. The action $\mathsf{DoTask}_{i,t,k}$ requires that i is free and knows skill $skill_k$ and that $skill_k$ is a necessary and sufficient condition to accomplish t. Its effect is that the agent is not free and that the task is done. Finally, the $\mathsf{EndStep}$ action frees all agents (regardless of whether they were teaching, learning or performing a task).

Suppose there are n agents, n tasks and one skill $skill_k$ that is only known by agent i. Then an optimal parallel plan has two steps: first i teaches $skill_k$ to the other agents and then each agent j executes task t_j in parallel. In contrast, the optimal sequential plan is that i executes all the tasks herself, which is an n-step plan that cannot be parallelized.

Figure 2 shows the time needed by Planning-PDBs to find a plan, in seconds. It compares the effect of the number of tasks, the number of agents and the number of skills. The first plot shows the effect of the number of tasks with different fixed values of the number of agents and skills. We can see that this variable has almost no effect on the difficulty to find a plan. The second plot does the same with the number of agents with different fixed values for the number of tasks and skills, while in the third plot the number of skills is the only variable which varies. These latter two plots show that the number of agents and the number of skills have an effect on the complexity. However, we do not have enough values to say more about this relationship.

The number of results were limited by the increasing complexity of the planning task. The most complex planning task for which we found an optimal parallel plan involved 7 tasks, 7 agents and 6 skills: this instance has 92 atoms and 638 actions. The most complex planning task tested and which timed out, had 7 tasks, 7 agents and 7 skills: this instance has 105 atoms and 736 actions.

Remark 2. One may replace $\mathsf{Teach}_{i,j,k}$ by the action $\mathsf{ReqSkill}_{j,i,k}$ of j asking i to teach her skill $skill_k$. Then we have to add to the precondition that j does not have the skill but knows that i does:

$$pre(\mathsf{ReqSkill}_{j,i,k}) = pre(\mathsf{Teach}_{i,j,k}) \land \neg S_j\ skill_k \land S_j\ S_i\ skill_k.$$

We note that such a precondition cannot be expressed with epistemic literals of the approach of (Muise et al. 2015).

8.3 Meetings Task

As an example of the use of common knowledge, consider a planning task which involves cooperation between different agents and which can be divided into m different stages with tasks to be performed at each step by each agent. Agents are only authorized to start stage $\ell{+}1$ if *all* tasks of stage ℓ have been completed and all agents have common knowledge of this. The only way this can be achieved is by having a plenary meeting at the end of each stage during which each agent announces that their stage-ℓ task has been completed (action $\mathsf{DoMeeting}_\ell$).

Initially all agents are free and no meeting or task has been completed: $s_0 = \{free_i : i \in Agt\}$. The goal is for all tasks and meetings to be completed: $Goal = \bigwedge_{t \in Tasks} tdone_t \land \bigwedge_{\ell \in Meetings} mdone_\ell$.

The action descriptions are listed in the lower half of Table 2. Each stage-ℓ task requires that the agent executing it knows that meeting ℓ has taken place. To avoid having stage-ℓ tasks done at stage-ℓ' for $\ell' > \ell$, the task also requires the meeting $\ell{+}1$ not to have taken place yet. The action $\mathsf{DoMeeting}_\ell$ of holding a meeting is executable if all the agents are free. Its effect is that all agents jointly see that the meeting has been held and that the agents are no longer free. Finally, the action $\mathsf{EndStep}$ frees the agents and ends the step.

For example, with 2 agents and 3 meetings (2 stages), the following is a solution plan:

$$\begin{aligned}&\langle\{\mathsf{DoMeeting}_1\}, \{\mathsf{EndStep}\},\\ &\{\mathsf{DoTask}_{1,1,1}, \mathsf{DoTask}_{2,2,1}\}, \{\mathsf{EndStep}\},\\ &\{\mathsf{DoMeeting}_2\}, \{\mathsf{EndStep}\},\\ &\{\mathsf{DoTask}_{1,3,2}, \mathsf{DoTask}_{2,4,2}\}, \{\mathsf{EndStep}\},\\ &\{\mathsf{DoMeeting}_3\}\rangle.\end{aligned}$$

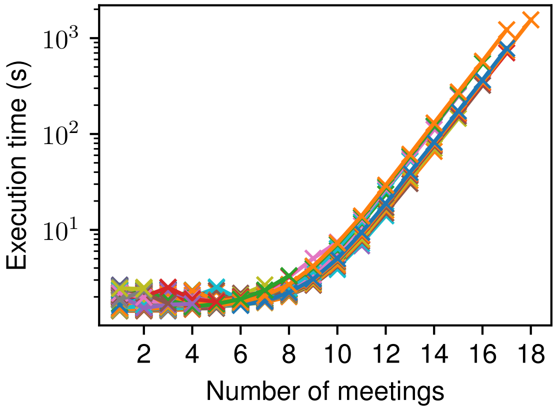

Figure 3: Meetings: time to find an optimal parallel plan vs. number of meetings (for different fixed numbers of tasks and agents)

Figure 3 shows the time needed by Planning-PDBs to find a plan, in seconds, relative to the number of meetings, where each plotted curve is for a fixed number of tasks and agents. We can see that the other two parameters (number of tasks and number of agents) have little to no effect on the difficulty of finding an optimal plan. In contrast, the number of meetings seems to have an exponential effect when it is greater than 10.

The largest planning task which was solved within the time limit of 1800s involved 18 tasks, 17 meetings and two agents: this instance has 46 atoms and 16400 actions. The most complex planning tasks that we tested and that timed out had 18 tasks, 18 meetings and one agent: the planners were not able to translate this problem.

9 Conclusion

We have defined simple epistemic planning tasks and their solvability by a parallel plan and characterised its complexity via a polynomial translation into classical planning. This allows us to solve epistemic bounded horizon planning tasks by translating them to classical bounded horizon planning, which is known to be PSPACE-complete.

Our ‘knowing whether’-based language can express more than the ‘knowing-that’-based language of (Muise et al. 2015) (see Remark 2). Contrarily to (Kominis and Geffner 2015), we are not restricted to common knowledge of the initial state and public or semi-public actions, as illustrated by the gossiping task where actions are private. Moreover, none of the latter two approaches deals with common knowledge or concurrent actions. Experiments demonstrated the possibility of solving some interesting practical problems.

In future work we plan to use SAT-based planners such as Rintanen et al.’s (Rintanen, Heljanko, and Niemelä 2006), which output parallel plans and therefore do not require the EndStep action.

Acknowledgements

We would like to thank the reviewers and the meta-reviewer of KR for their constructive comments.

References

Aucher, G., and Bolander, T. 2013. Undecidability in epistemic planning. In Rossi, F., ed., *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013)*, 27–33. AAAI Press.

Bavelas, A. 1950. Communication patterns in task-oriented groups. *The Journal of the Acoustical Society of America* 22(6):725–730.

Blum, A., and Furst, M. L. 1997. Fast planning through planning graph analysis. *Artif. Intell.* 90(1-2):281–300.

Bolander, T., and Andersen, M. B. 2011. Epistemic planning for single and multi-agent systems. *Journal of Applied Non-Classical Logics* 21(1):9–34.

Bolander, T.; Jensen, M. H.; and Schwarzentruber, F. 2015. Complexity results in epistemic planning. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, 2791–2797.

Bylander, T. 1994. The computational complexity of propositional STRIPS planning. *Artificial Intelligence* 69:165–204.

Cong, S. L.; Pinchinat, S.; and Schwarzentruber, F. 2018. Small undecidable problems in epistemic planning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, 4780–4786.

Cooper, M. C.; Herzig, A.; Maffre, F.; Maris, F.; and Régnier, P. 2016. A simple account of multi-agent epistemic planning. In *ECAI 2016 - 22nd European Conference on Artificial Intelligence*, 193–201.

Cooper, M. C.; Herzig, A.; Maffre, F.; Maris, F.; and Régnier, P. 2019. The epistemic gossip problem. *Discrete Mathematics* 342(3):654–663.

Dimopoulos, Y.; Nebel, B.; and Koehler, J. 1997. Encoding planning problems in nonmonotonic logic programs. In *Recent Advances in AI Planning, 4th European Conference on Planning, ECP'97, Toulouse, France, September 24-26, 1997, Proceedings*, 169–181.

Herzig, A.; Lorini, E.; and Maffre, F. 2015. A poor man's epistemic logic based on propositional assignment and higher-order observation. In van der Hoek, W.; Holliday, W. H.; and Wang, W.-f., eds., *Proceedings of the 5th International Conference on Logic, Rationality and Interaction (LORI 2015)*, 156–168. Springer Verlag.

Herzig, A.; Maris, F.; and Vianey, J. 2019. Dynamic logic of parallel propositional assignments and its applications to planning. In Kraus, S., ed., *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 5576–5582. ijcai.org.

Huang, X.; Fang, B.; Wan, H.; and Liu, Y. 2017. A general multi-agent epistemic planner based on higher-order belief change. In Sierra, C., ed., *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, 1093–1101. ijcai.org.

Knoblock, C. A. 1994. Generating parallel execution plans with a partial-order planner. In *Proceedings of the Second International Conference on Artificial Intelligence Planning Systems*, 98–103.

Knödel, W. 1975. New gossips and telephones. *Discrete Mathematics* 13(1):95.

Kominis, F., and Geffner, H. 2015. Beliefs in multiagent planning: From one agent to many. In *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling, ICAPS 2015, Jerusalem, Israel, June 7-11, 2015*, 147–155.

Landau, H. G. 1954. The distribution of completion times for random communication in a task-oriented group. *The Bulletin of Mathematical Biophysics* 16(3):187–201.

Le, T.; Fabiano, F.; Son, T. C.; and Pontelli, E. 2018. EFP and PG-EFP: epistemic forward search planners in multi-agent domains. In de Weerdt, M.; Koenig, S.; Röger, G.; and Spaan, M. T. J., eds., *Proceedings of the Twenty-Eighth International Conference on Automated Planning and Scheduling, ICAPS 2018, Delft, The Netherlands, June 24-29, 2018*, 161–170. AAAI Press.

Liu, Q., and Liu, Y. 2018. Multi-agent epistemic planning with common knowledge. In Lang, J., ed., *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, 1912–1920. ijcai.org.

Löwe, B.; Pacuit, E.; and Witzel, A. 2011. DEL planning and some tractable cases. In van Ditmarsch, H.; Lang, J.; and Ju, S., eds., *Logic, Rationality, and Interaction - Third International Workshop, LORI 2011, Guangzhou, China, October 10-13, 2011. Proceedings*, volume 6953 of *Lecture Notes in Computer Science*, 179–192. Springer.

McDermott, D.; Ghallab, M.; Howe, A.; Knoblock, C.; Ram, A.; Veloso, M.; Weld, D.; and Wilkins, D. 1998. PDDL, The Planning Domain Definition Language. *Technical Report CVC TR98003/DCS TR1165, Yale Center for Computational Vision and Control.*

Muise, C.; Belle, V.; Felli, P.; McIlraith, S. A.; Miller, T.; Pearce, A. R.; and Sonenberg, L. 2015. Planning over multi-agent epistemic states: A classical planning approach. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI 2015)*, 3327–3334. AAAI Press.

Rintanen, J.; Heljanko, K.; and Niemelä, I. 2006. Planning as satisfiability: parallel plans and algorithms for plan search. *Artif. Intell.* 170(12-13):1031–1080.