Proceedings of the Twenty-Seventh International Conference on Automated Planning and Scheduling (ICAPS 2017)

Multiagent Online Planning with Nested Beliefs and Dialogue

Filippos Kominis
Universitat Pompeu Fabra
08018 Barcelona, Spain
filippos.kominis@upf.edu

Hector Geffner
ICREA & Universitat Pompeu Fabra
08018 Barcelona, Spain
hector.geffner@upf.edu

Abstract

The problem of planning with partial observability in the presence of a single agent has been addressed as a contingent or POMDP problem. Since the task is computationally hard, on-line approaches have also been developed that just compute the action to do next rather than full policies. In this work, we address a similar problem but in a multiagent setting where agents share a common goal and plan with beliefs which are about the world and the possibly nested beliefs of other agents. For this, we extend the belief tracking formulation of Kominis and Geffner to the on-line setting where plans are supposed to work for the true hidden state as revealed by the observations, and develop an alternative translation into classical planning that is used within a plan-execute-observe-and-replan cycle. Planning is done from the perspective of the agents, and there is a single planning agent in each replanning episode that can change across episodes. We present empirical results and show that interesting agent dialogues arise in this setting where agents collaborate by requesting or volunteering information in a goal-directed manner.

Introduction

Single-agent planning with partial observability is a hard computational problem where even the size of the required policies is often exponential in the problem size (Rintanen 2004). For avoiding this bottleneck, *on-line* approaches have been developed that rather than computing full policies off-line, compute the next action to do given the observations gathered (Albore, Palacios, and Geffner 2009; Brafman and Shani 2012b; Bonet and Geffner 2014).

In this work, we address the problem of online planning in partially observable environments in the presence of multiple agents that share a common goal and plan with beliefs that can be about the world or about the possibly nested beliefs of other agents. This setting is addressed by *dynamic epistemic logics* (van Ditmarsch, van der Hoek, and Kooi 2007; van Ditmarsch and Kooi 2008; Van Benthem 2011), yet these logics are undecidable in general (Aucher and Bolander 2013; Charrier, Bastien, and Schwarzentruber 2016) with special decidable fragments associated with restrictions on action pre and postconditions (Löwe, Pacuit, and Witzel 2011; Bolander, Jensen, and Schwarzentruber 2015). In this paper, we build instead on the formulation developed recently by Kominis and Geffner that corresponds to a rich fragment of DEL, for which it provides a convenient modeling language, a simple semantics, and procedures akin to those used in the single-agent setting (Kominis and Geffner 2015). In this approach, the basic assumptions are that actions are public, physical actions are deterministic, and the set of possible initial states is common to all the agents. There is a clear tradeoff between expressivity, simplicity, and computational efficiency, and other approaches addressing planning in a multiagent setting make different tradeoffs (Baral et al. 2012; Brafman, Shani, and Zilberstein 2013; Muise et al. 2015; Engesser et al. 2015; Cooper et al. 2016).

For using Kominis' and Geffner's formulation in the on-line setting, three issues need to be addressed. First, beliefs must take into account the actual observations gathered by the agents. Second, plans must be computed by the agents themselves using their own private information. And third, plans do not have to achieve the goal for all possible initial states, but for the true hidden initial state only. We address these issues by adopting a suitable formulation of truth in the on-line setting that is used within a plan-execute-observe-and-replan cycle along with a translation into classical planning for selecting actions. The resulting on-line planning algorithm is guaranteed to reach the goal in a bounded number of calls to a classical planner provided that there are no dead-ends, even if different agents are chosen to plan in the different replanning episodes. We also show that interesting agent dialogues arise in this setting where agents request, provide, and volunteer information in a collaborative, goal-directed manner.

The rest of the paper is organized as follows. We review first the modeling language and the belief representation, and introduce the extensions required for the on-line setting. We then introduce the translation into classical planning, the replanning algorithm, and its formal properties. We finally present the examples and resulting agent dialogues, and the experimental results.

Motivation

The Active Muddy Child (Kominis and Geffner 2015) is a planning version of the the famous Muddy Children puzzle (Fagin et al. 1995). The problem is useful for illustrat-

ing the differences between the off-line and on-line settings that when planning with epistemic goals is more crucial than in the standard partially observable setting of single-agent planning. While in the original puzzle, the father announces that at least one child is muddy, and then asks the children repeatedly whether they know whether they are muddy or not until the muddy children all infer that they are muddy, in the Active version, one of the children is the one asking the questions to find out whether he is muddy or not. Moreover, he has to ask these questions to one child at a time, whose answer is heard by all the children. A *conformant plan* for the Active Muddy Child problem, with n children, is one where the active child asks the question to each one of the children in turn without leaving any one out, in any order. The plan achieves the goal regardless of the true initial state. The problem has indeed $2^n - 1$ possible initial states where different subsets of children are muddy, excluding the state where no child is muddy that is common knowledge.

In the *on-line version* of the Active Muddy Child problem, the child asking the questions to figure out whether he is muddy or not, does not have to ask each of the children in turn whether they are muddy or not. The "planning child", like the other children, senses the world and can perfectly see which children are muddy and which ones are not, except for himself. A more effective strategy in the on-line setting is to approach *only* the children that are seen to be muddy. Any plan where the "planning child" asks the question to each of the children that he sees muddy, will achieve the goal.

The difference between the off-line and on-line setting is not the presence of observations that the planning agent can use for selecting actions, but that plans must work for *all possible initial states* in the first case, and for just *the true hidden initial state* in the second one. Indeed, in the single-agent case, the solution form for contingent planning problems is a contingent tree. This tree makes use of observations but the planning method is *off-line*: the trees cover all the possibilities and hence all possible initial states. The solution form in *on-line* planning, on the other hand, is not a tree but an *action sequence*, as the solution must work for one state only: the true but hidden initial state. The planning agent in the on-line setting may find useful to consider many and even all possibilities before deciding what to do next, but this is just the criterion for choosing actions. The solution of the on-line problem is the sequence of actions that achieves the goal. The observations are relevant because they provide indirect information about the hidden state.

The distinction between off-line and on-line planning is often left implicit and without formalization in the single-agent, partial observable setting, because goals in the latter are objective and refer to the world. In the on-line setting of epistemic, multiagent planning, on the other hand, things are different and force us to make explicit and formal the conditions under which an epistemic goal is achieved from the internal perspective of the planning agent, and hence the role that the hidden true state plays in such conditions.

Language

We consider planning problems $P = \langle A, F, I, O, N, S, G\rangle$ where A is the set of agent names or indexes, F is the set of relevant atoms or fluents, I represents the initial situation in the form of an objective formula over F, O is the set of *physical actions*, N is the set of *sensing actions*, S is the set of (passive) *sensors*, and G is the goal (Kominis and Geffner 2015). States represent truth-valuations over F, and the set of possible initial states b_I is made of the states that satisfy I. The physical actions a define a mapping f_a such that $f_a(s)$ represents the state that result from applying action a in the state s. Syntactically, such mappings are defined through a set of conditional effects of the form $C \rightarrow L$, where L is a literal and C is a formula over F. A sensing action in N is a set of expressions of the form **sense**$[i](\phi)$, where i is an agent, and ϕ is an objective or epistemic formula. A result of the action is that the truth value of ϕ is revealed to agent i. A (parallel) sensing action in N is a set of expressions of the form **sense**$[A_k](\phi)$, where the truth of ϕ is revealed to all the agents $j \in A_k$. Unlike sensing actions, sensors reveal information without having to act. We denote passive sensors like sensing actions but with the letter "p" in front; namely, as **psense**$[i](\phi)$ and **psense**$[A_k](\phi)$. Also, we write **sense**(ϕ) and **psense**(ϕ) when the sensing involves all the agents, i.e. $A_k = A$.

The goal G and the formulas ϕ above can be epistemic. The epistemic formulas ϕ include the atoms in F, and recursively, the formulas $K_i\phi$ for $i \in A$, and the boolean combinations of such formulas where K_i is the standard operator in logics of knowledge (Fagin et al. 1995).

Finally, physical actions a have a precondition formula $Pre(a)$ that can be objective or epistemic. We assume that each action has an 'owner" and that the action is applicable if the owner knows that the precondition is true (Engesser et al. 2015). The execution of physical and sensing actions however is public and so are the agent's sensors.

Beliefs

Beliefs are represented by a suitable collection of sets of states. The beliefs define a Kripke structure where arbitrary epistemic formulas can be evaluated.

External View

The beliefs of all the agents at time step t, denoted as $B(t)$, is represented by the beliefs $B(s,t)$ conditional on $s \in b_I$ being the true initial state, given as (Kominis and Geffner 2015):

$$B(s,t) = \langle v(s,t), r_1(s,t), r_2(s,t), ..., r_m(s,t)\rangle$$

where $v(s,t)$ is the state of the world that results from the initial state s after the action sequence $\pi(0), \ldots, \pi(t-1)$, and $r_i(s,t)$ is the set of possible *initial* states $s' \in b_I$ that agent i cannot distinguish at time t from the actual initial state s.

For $t = 0$, $v(s,t) = s$ and $r_i(s,t) = b_I$ for all agents i, while for $t > 0$, $B(t+1)$ is determined by $B(t)$ and the action $\pi(t)$ at time t.

If $\pi(t)$ is a sensing action or contains such actions, the current state given s does not change, i.e., $v(s,t+1) = v(s,t)$, but the set of possible initial states compatible with the hidden initial state s for agent i given by $r_i(s,t+1)$ becomes:

$$\{s' | s' \in r_i(s,t),\ B(t), s' \models \psi \text{ iff } B(t), s \models \psi, \forall \psi \in O_i(t)\}$$

where $O_i(t)$ represents the *observables* at time t and contains all the formulas ϕ such that the action **sense**$[A_k](\phi)$ is in $\pi(t)$ or **psense**$[A_k](\phi)$ is a passive sensor, in both cases with $i \in A_k$. The expression $B(t), s \models \phi$ denotes that ϕ is true in the belief $B(t)$ conditional on s being the true hidden state. The truth conditions for such expressions are spelled out below.

If $\pi(t)$ is a physical action a, the current state $v(s,t)$ associated with the hidden initial state s changes according to the transition function f_a associated with a as $v(s,t+1) = f_a(v(s,t))$, while the sets of initial states $r_i(s,t)$ change according to the displayed formula above, where the observables in $O_i(t)$ result from the passive sensors only. In addition, if the action a is "owned" by agent j, states $s \in r_i(s,t+1)$ where $B(t), s \models K_j Pre(a)$ does not hold are removed from $r_i(s,t+1)$, meaning that agents i learn that action a is then applicable.

From Beliefs to Kripke Structures

A Kripke structure is a tuple $\mathcal{K} = \langle W, R, V \rangle$, where W is a set of worlds, R is a set of binary accessibility relations R_i on W, one for each agent i, and V is a mapping from the worlds w in W into truth valuations $V(w)$. Due to the assumptions we make (actions are deterministic, known and public), accessibility relations are equivalence relations. The conditions under which an arbitrary formula ϕ is true in a world w of a Kripke structure $\mathcal{K} = \langle W, R, V \rangle$, written $\mathcal{K}, w \models \phi$, are defined inductively (Fagin et al. 1995):

- $\mathcal{K}, w \models p$ for an atom p, if p is true in $V(w)$,
- $\mathcal{K}, w \models \phi \vee \psi$ if $\mathcal{K}, w \models \phi$ or $\mathcal{K}, w \models \psi$,
- $\mathcal{K}, w \models (\phi \Rightarrow \psi)$ if $\mathcal{K}, w \models \phi$ implies $\mathcal{K}, w \models \psi$,
- $\mathcal{K}, w \models K_i \phi$ if $\mathcal{K}, w' \models \phi$ for all w' s.t. $R_i(w, w')$,
- $\mathcal{K}, w \models \neg\phi$ if $\mathcal{K}, w \not\models \phi$

The conditions under which a possible initial state s predicts the truth of a formula ϕ at time t, written $B(t), s \models \phi$, follow from replacing the belief $B(t)$ by the Kripke structure $\mathcal{K}(t) = \langle W^t, R^t, V^t \rangle$ defined by $B(t)$ where $W^t = \{s \mid s \in b_I\}$, $R_i^t = \{(s,s') \mid s' \in r_i(s,t)\ \}$, and $V^t(s) = v(s,t)$.

The worlds w in the structure $\mathcal{K}(t)$ are thus the possible initial states $s \in b_I$, while the worlds that are accessible from a world s to the agent i are the possible initial states s' that are in $r_i(s,t)$. Finally, the valuation associated to a world s in this structure is the state $v(s,t)$ that deterministically follows from the possible initial state s and the action sequence up to $t-1$. $B(t), s \models \phi$ is defined as true when $\mathcal{K}(t), s \models \phi$ is true.

Agent's View

While in the *off-line setting*, a formula ϕ is regarded as true at time t when $\mathcal{K}(t), s_0 \models \phi$ is true for *all* possible initial states $s_0 \in b_I$, i.e., all worlds in the structure, in the *on-line setting*, truth is defined in relation to the single *actual world*, which corresponds to a true but hidden initial state denoted as s_0^*:

Definition 1 (On-line Truth) *A formula ϕ is true at time t in the on-line setting, written $B(t) \models \phi$, iff $\mathcal{K}(t), s_0^* \models \phi$ where $s_0^* \in b_I$ is the hidden initial state.*

This is a simple but crucial definition. No similar explicit account for truth is required in contingent planning where the hidden initial state s_0^* plays an indirect role only. This is because the goal is an objective formula and it is sufficient then to keep track of the set of states that are possible at a given time point, the so-called *belief state* (Bonet and Geffner 2000), in order to determine if the goal holds or not.

Definition 1, however, can't be applied to arbitrary formulas, as the agents do not have access to the hidden state s_0^*. Yet, each agent i can use Definition 1 to evaluate formulas of the form $K_i\phi'$ provided that the set $S_i(t)$ of initial states that are possible to agent i by time t is tracked. This set depends on the actual observations gathered by agent i. Initially $S_i(0) = b_I$ and $S_i(t+1)$ is:

$$S_i(t+1) = \{s' | s' \in S_i(t), B(t), s' \models \psi, \forall \psi \in O_i^+(t)\}$$

where $O_i^+(t)$ stands for the set of observations available to agent i at time t; namely, the formulas ψ (observable) in $O_i(t)$ that have been observed to be *true* at t, and the *negation* of the formulas ψ (observable) in $O_i(t)$ that have been observed to be *false*. Provided with this set of possible initial states, the truth of formulas $K_i\phi$ according to Definition 1 can be evaluated as follows:

Theorem 1 *$B(t) \models K_i\phi$ iff $\mathcal{K}(t), s_0 \models \phi, \forall s_0 \in S_i(t)$.*

Indeed, for evaluating the formula $K_i\phi$ in s_0^*, the agent does not need to know the hidden state s_0^* but $r_i(s_0^*, t)$; i.e., the set of states that agent i cannot tell apart from s_0^* at time t. Yet this set is precisely $S_i(t)$.

As an illustration, if the problem P involves two agents 1 and 2, two fluents p and q, $I = \{p \equiv q\}$, and π is given by the action $\pi(0) =$ **sense**$[1](p)$ followed by $\pi(1) =$ **sense**$[2](q)$, we get a joint belief $B(t)$ for $t = 2$ that defines a Kripke structure $\mathcal{K}(t)$ where formulas such as $K_1 p \equiv K_2 q$ hold in all the states, and formulas such as $K_1 p$ and $K_2 q$ do not. Yet, if the true hidden state s_0^* is such that p and q are true in s_0^*, formulas such as $K_2 q$ and $K_2 K_1 p$ would be true in $B(t)$ according to Definition 1 for $t = 2$, and false for $t = 1$.

Planning

Planning in our setting involves the incremental computation and execution of a sequence of actions that makes the goal true. The algorithm shown in Figure 1 computes such sequences using a replanning method that is similar to those developed for single-agent on-line planning in partial observable settings (Brafman and Shani 2012b; Bonet and Geffner 2014). Initially, a selected planning agent i computes an action sequence π by calling a *classical planner* over a translation $K(P, B(t), S_i(t))$ that expresses a *relaxation* where agent i is allowed to make a guess about the

true hidden state s_0^*. This simplification does not make the hidden state known to the planning agent but determines the outcomes of all sensing actions which thus become deterministic. If the planning agent i is "lucky", the execution of the (normalized) action sequence π will not reveal to agent i that the choice is wrong. In such a case, the action sequence can be applied fully, achieving K_iG and hence the goal G. On the other hand, if the execution of π reveals to agent i at time $t' > t$ that s is not the true hidden initial state, then s is removed from $S_i(t')$, and the process repeats with the updated beliefs $B(t')$ and sets $S_i(t')$, possibly with a different planning agent. One agent is selected as the planning agent in each replanning episode. A fixed ordering among the agents is also assumed so that if for the selected planning agent i, the classical problem $K(P, B(t), S_i(t))$ has no solution, the selected planning agent becomes the next agent in the ordering. Notice that an action like **sense**$[j](K_i\phi)$ in a plan computed by agent k represents information sharing when $k = i$ and information request when $k = j$. Similarly, a physical action a planned by agent i and owned by agent j represents a request from i to j to do the action a.

Algorithm 1 Online planning and execution for problem P

```
1: Inputs: B(0), S(0), initial planning agent i
2: t ← 0
3: loop
4:   Generate classical problem K(P, B(t), S_i(t))
5:   Compute classical plan π from K(P, B(t), S_i(t))
6:   Normalize π by removing auxiliary actions
7:   Execute π incrementally updating B(t) and S_i(t)
       til first t' where K_iG true or inconsistency detected
8:   Agents j update S_j(t) til t = t' with own observation
9:   if K_iG achieved then
10:    exit
11:  else
12:    t ← t'
13:    Set new planning agent i
```

Properties

Before considering the translation in detail, we present the basic properties which can also be understood as the requirements that the translation must fulfill. The translation introduces auxiliary actions, such as assuming a hidden true state and simulating the passive sensors. For an action sequence π obtained from the translation, the *normalization* of π, denoted as $n(\pi)$, is the same sequence but with the auxiliary actions removed. The notion of *consistency* results from matching the observations assumed by the plan and the actual observations gathered. The former follow from the choice of the hidden state which is captured by an auxiliary action $assume(s)$ that must be unique and appear first in the plan.

Definition 2 (Consistency) *Let π be a prefix of a plan for $P' = K(P, B(t), S_i(t))$. The normalized sequence $n(\pi)$ is* consistent *with the observations iff a) for any formula ϕ rendered* observable *by $n(\pi)$ at time t' from active or passive sensing, $B(t'), s \models \phi$ iff ϕ is observed to be true at time t', and $assume(s)$ is the first action in π, and b) the physical actions a in $n(\pi)$ are all applicable in P (i.e., owners know the preconditions).*

The results below assume further that a physical action a owned by agent j that is *not* applicable in the plan computed by agent $i \neq j$ from the translation, is replaced by a communication; namely, the action **sense**$[i](K_j(Pre(a)))$. That is, agent i learns that the action is not applicable.

Theorem 2 (Soundness) *a) If π is plan for $K(P, B(t), S_i(t))$ that is consistent with the observations, the execution of $n(\pi)$ leads to the goal in P. b) Otherwise, if π' is the shortest prefix of π that is inconsistent and π includes the action $assume(s)$, after the execution of $n(\pi')$ in P, $s \notin S_i(t')$ where t' is the resulting time step.*

Theorem 3 (Completeness) *If $s = s_0^* \in S_i(t)$ is the true hidden state in P and there is an action sequence that achieves K_iG for an agent i, then there is a plan π for $K(P, B(t), S_i(t))$ that starts with the action $assume(s)$, and any such plan is consistent.*

These properties of the translation ensure that Algorithm 1 is a sound and complete replanning algorithm for P provided that no execution of P can reach a dead-end, i.e., a situation from which no action sequence can lead to K_iG for any agent i:

Theorem 4 (Goal Achievement) *If the executions in P cannot reach a dead-end, Algorithm 1 will solve P after a number of calls to the classical planner that is bounded by $|b_I| \times |A|^2$, where b_I is the set of initial states in P and A is the set of agents.*

In the worst case, a protocol may have to iterate over all the agents until finding an agent i that can find a plan in the translation for the goal K_iG. The execution of that plan ensures that the goal K_iG is reached or that at least one state s is removed from $S_i(t)$. The number of such removals is bounded by $|b_I| \times |A|$.

The Translation

The language for the translation $P' = K(P, B(t), S_i(t))$ in Algorithm 1 is STRIPS extended with *negation*, *conditional effects*, and *axioms*. The primitive fluents in P' are used to represent the states $v(s, t)$ and the collection of states $r_j(s, t)$ that define the beliefs $B(t)$. For encoding the states $v(s, t)$, P' contains atoms L/s that express that the objective literal L is true in the current state if s is the initial state, while for encoding the sets $r_j(s, t)$, P' contains fluents $D_j(s, s')$ that are true when $s' \notin r_j(s, t)$. P' also features atoms $T(s)$ for representing that s is the *assumed true initial state*, and atoms $D_i(s)$ for representing that $s \notin S_i(t)$. Formulas appearing in action preconditions, goals, and sensing expressions in P are assumed to be all literals or conjunctions of possibly epistemic literals L. A positive epistemic literal is an objective literal preceded by a sequence of epistemic operators possibly separated by negations, like $K_a\neg K_bK_cp$. The axioms in the translation are used to maintain the truth of epistemic literals. We denote the set of objective literals in P as $L_F(P)$, the set of

positive epistemic literals in P as $L_K(P)$, and the set of positive epistemic literals L that are suffixes of literals in $L_K(P)$ as $L_X(P)$. The literals ϕ/t in the translation are used to encode the truth of formulas ϕ in the assumed initial state; i.e., ϕ/t iff ϕ/s and $T(s)$. Such formulas ϕ are the ones appearing in sensing and preconditions. The actions in $K(P, B(t), S_i(t))$ comprise the physical actions in P, the auxiliary actions $assume(s)$ for guessing the initial state, the action $\mathcal{E}$ for capturing the effects of passive sensing, and the sensing actions **sense**$[A](\phi)$ in P. The action $assume(s)$ must appear first in any plan for some possible s, excluding all other $assume(s')$ actions from being applied.

Definition 3 *The* classical problem with axioms $K(P, B(t), S_\alpha(t)) = \langle F', I', O', G', X' \rangle$ *where* α *is the planning agent and* $P = \langle A, F, I, O, N, S, G \rangle$ *is such that:*

- $F' = \{L/s : L \in L_F(P), s \in b_I\} \cup \{T(s) : s \in b_I\} \cup \{D_i(s, s') : i \in A, s, s' \in b_I\} \cup \{D_\alpha(s) : s \in b_I\}$,
- $I' = \{L/s : L \in L_F(P), s \in b'(t), s \models L\} \cup \{D_\alpha(s) : s \in b_I, s \notin S_\alpha(t)\} \cup \{D_i(s, s') : s, s' \in b_I, s \notin r_i(s', t), i \in A\}$
- $G' = \wedge_{s \in b_I}(D_\alpha(s) \vee G/s)$
- *Axioms* X'*:*
  - $K_i L/s$ *iff* $\wedge_{s' \in b_I}[L/s' \vee D_i(s, s')]$, $K_i L \in L_X(P) \cup L_K(P)$
  - ϕ/t *iff* $\wedge_{s \in b_I}[\neg T(s) \vee \phi/s]$, ϕ *in sensing and preconditions*
- *Actions* O'*:*
  - ***auxiliary actions*** $assume(s)$*, for* $s \in b_I$*, with prec.* $\neg D_\alpha(s)$ *and effect* $T(s)$*,*
  - ***physical actions*** $a \in O$ *owned by* j *have prec.* $K_j(Pre(a))/t$ *and effects* $\neg K_j(Pre(a))/s \rightarrow D_i(s, s') \wedge D_\alpha(s)$ *for* $s, s' \in b_I$ *and* $C/s \rightarrow E/s$ *for each* $s \in b_I$ *and effect* $C \rightarrow E$ *of* a *in* P
  - ***sensing actions sense***$[B](\phi) \in N$ *with* $\alpha \notin B$ *mapped into same action without precs, and effects:*
    - $\phi/s \wedge \neg\phi/s' \rightarrow D_i(s, s'), D_i(s', s)$ *for* s, s' *in* b_I *and* $i \in B$*,*
  - ***sensing actions sense***$[B](\phi) \in N$ *with* $\alpha \in B$ *mapped into the same action, with effects*
    - $\phi/s \wedge \neg\phi/s' \rightarrow D_i(s, s'), D_i(s', s)$ *for* s, s' *in* b_I *and* $i \in B$*, and*
    - $\phi/t \wedge \neg\phi/s \rightarrow D_\alpha(s)$*,*
    - $\neg\phi/t \wedge \phi/s \rightarrow D_\alpha(s)$*, for* $s \in b_I$*,*
  - ***auxiliary action*** $\mathcal{E}$ *with effects*
    - $\phi/s \wedge \neg\phi/s' \rightarrow D_i(s, s'), D_i(s', s)$ *for each pair of states* s, s' *in* b_I*,* ***psense***$[B](\phi)$ *in* S*, and* $i \in B$*,*
    - $T(s) \wedge \phi/s \wedge \neg\phi/s' \rightarrow D_\alpha(s')$*, if* $\alpha \in B$*,* $s, s' \in b_I$*.*

In the above translation we omit the auxiliary literals used for specifying ordering of actions that force an action $assume(s)$ as the first action, and the action $\mathcal{E}$ after each other action. Also, while not covered in the above description, parallel sensing actions are also accommodated.

The translation is quadratic in the number of possible initial states $|b_I|$, and hence exponential in the number of atoms in the worst case. The same is true however for sound and complete translations in the single-agent setting (Brafman and Shani 2012a).

Protocols

We consider four *protocols*, each either identifying the next planning agent or forcing information sharing.

In *fixed agent,* the initial planning agent remains so throughout the execution until reaching the goal.

In *last-agent,* when the shortest inconsistent plan ends with a sensing action **sense**$[B](K_j\phi)$ or a physical action owned by an agent j different than the planning agent, the control is given to agent j.

Third is the *volunteering* protocol. When the shortest inconsistent plan ends with a sensing action involving agent j (e.g., **sense**$[i](K_j L)$) and i is the planning agent, j "volunteers" information to i. This is achieved by selecting and applying the most informative sensing action of the form **sense**$[i](K_j L')$. The *most informative* sensing action is the one that removes the largest number of states from the set of states R that i may consider possible, according to j. Formally, $R = \{s \,|\, s \in r_i(s', t)$ and $s' \in S_j(t)\}$ is the set of states i may consider possible, from the perspective of j. Then, for all possible sensing actions **sense**$[i](K_j L')$, we define $R(K_j L') = \{s \,|\, s \in R,\ B(t), s \models K_j L'$ iff $B(t) \models K_j L'\}$, the set of states in R which agree with the truth value of $K_j L'$. The action with the smallest $|R(K_j L')|$ is chosen as the most informative. Ties are broken randomly and no sensing action is applied if there is no $|R(K_j L')| < |R|$.

The last protocol is the *vol-mutex* protocol. Similarly to the *volunteering* protocol, when the shortest inconsistent plan ends with a sensing action involving agent j (eg **sense**$[i](K_j L)$), and i is the planning agent, j "volunteers" information to i. The difference is that instead of j volunteering the most *informative* information, he will volunteer the information most *relevant* to L. We define this relevance using sets of mutually exclusive (mutex) literals that are precomputed in low polynomial time: two literals L and L' are relevant if they are mutex. If the plan ended with a sensing action **sense**$[i](K_j L)$, where i expected $K_j L$ to be true but he actually sensed that it is false, and there exists a literal L' relevant to L such that j knows L', then the sensing (communicative) action **sense**$[i](K_j L')$ is applied. If for all L' relevant (mutex) to L we have that $B(t) \not\models K_j L'$, then the actions **sense**$[i](K_j L')$ are done in parallel, thus communicating j-ignorance about such literals.

The difference between the *volunteering* and the *vol-mutex* protocol is a subtle one. We can see that in the *volunteering* protocol the agent shares the knowledge which will have *possibly* the biggest impact, yet it is possible that the information is irrelevant to the asking agent. Imagine a problem where two balls are placed in a grid. Ball 1 has 20 possible positions while ball 2 only four, the four corners of the grid. Imagine agent j knows the positions of both balls, and i, who is the planning agent, has as goal to learn the position of ball 2 only. Agent i may execute a plan where ball 2 is assumed to be in a specific corner of the grid, asking

then j to confirm. If the ball is in a different corner in the true hidden state, j will reply negatively. The *volunteering* protocol specifies that j will then announce the position of ball 1, as this removes the largest number of states. The *volmutex* protocol, on the other hand, will make agent j share the position of the ball 2, that is more relevant to the question even if it doesn't convey as much information as measured by the number of states that agent i would no longer view as possible.

From plans to dialogues

It is useful to display the trace left by the executions of plans in this on-line, multiagent, epistemic setting, as dialogues. For this, we follow some conventions:

- *Acting:* a physical action a with owner i and preconditions $Pre(a)$ is translated into "i: I apply a".
- *Requesting:* a physical action a with owner j and preconditions $Pre(a)$ is translated into "i: j, apply action a", to which a response will follow: "j: I applied a" or "j: I cannot apply a", depending on whether $K_j Pre(a)$. If the action has no preconditions, no response will follow since it is known that the action can be applied.
- *Providing:* a sensing action **sense**$[D](K_i L)$ is translated into "i: I tell all agents in D whether I know L", and if $D = A - \{i\}$ then into "i: I do know L", or "i: I do not know L", depending on the hidden true state.
- *Asking:* a sensing action **sense**$[D](K_j L)$, where $i \notin D$, is translated into "i: j, tell all agents in D whether you know L", to which a response will follow "j: I did tell all agents in D". If $i \in D$, the response depends on the view of the plan we have: if we present it from the point of view of the planning agent, the response will be "j: I do (not) know L", otherwise it would be "j: I did tell all agents in D". If $D = A - \{j\}$, then the the question would be "j: do you know L?", while the response will be "j: Yes, I do know L" or "j: I do not know L".

A more natural mapping is possible, given the names of the actions. For example, if we have **sense**$[i](K_j L)$, where L represents the fact that j sees the red ball, and i is the planning agent, then the action can be translated as "i: j, tell me that you see the red ball".

Examples and Experimental results

We present the dialogue traces for three problems, using various protocols. We obtained the results using the online replanning algorithm shown, and the FD planner as the classical planner (Helmert 2006), over a Linux machine at 2.93GHz with 4GB of RAM. In our implementation, each planning phase is a different call to FD, with the corresponding PDDL files. We present experimental results as tuples $\langle S, T, R \rangle$ next to each problem and protocol used. In these tuples, S stands for the average search time , T is the average total time, and R is the average number of replans. Search (total) time is the average search (total) time for each planning phase, while the average number of replans is taken by running the experiments over each possible initial state as the true initial state. An asterisk '*' next to an action indicates that a replanning phase occurred after the action, and we report when a change of planning agent or a volunteering occurred. Due to space, we collapse actions when the execution is clear. For example, a "j, move right twice. Do you see l?" indicates two consecutive physical actions and a sensing action, all relating to j.

Meeting Problem

We have two agents (a, b) and a ring-shaped grid of size six $(p_1, .., p_6)$. Within the grid there are three landmarks (l, q, r), each one positioned in either p_2, p_4 or p_6, and no two landmarks can be in the same position. The agents do not know the actual position of the landmarks. It is commonly known that a is initially positioned in either p_1 or p_2, while b in one of p_2, p_4 and p_6. An agent can see a landmark only if they are in the same position. The goal is for agent a to know that both agents are in p_1.

Each agent has a physical action "move-clockwise" and "move-anticlockwise", three sensors for seeing a landmark, and three actions for communicating if he is in the same position with one of the landmarks. We introduce auxiliary derived atoms $i@L$ with definition $\bigvee_{x \in \{2,4,6\}} i@p_x \wedge L@p_x$, where i the agent, L one of the landmarks and $i@p_x$ that i is in position p_x. Agents can sense their respective auxiliary derived atoms.

We have in total 4 physical actions: "move-clockwise(i)" with conditional effects $i@p_6 \rightarrow \neg i@p_6 \wedge i@p_1$ and $i@p_x \rightarrow \neg i@p_x \wedge i@p_{x+1}$ for $x \in \{1..5\}$, and "move-anticlockwise(i)" with conditional effects $i@p_1 \rightarrow \neg i@p_1 \wedge i@p_6$ and $i@p_x \rightarrow \neg i@p_x \wedge i@p_{x-1}$ for $x \in \{2..6\}$ and $i \in \{a, b\}$. There are 6 sensors **psense**$[i](i@L)$, for $i \in \{a, b\}$ and 6 sensing actions, **sense**$[a](K_b b@L)$, **sense**$[b](K_a a@L$, for $L \in \{l, r, q\}$, representing what the agent sees in the position he is at and what he communicates. The number of possible initial states are 36: 6 possible states due to the initial unknown positioning of landmarks, 2 possible states due to the uncertainty of a's position, and 3 possible states concerning b's positioning $(6 * 2 * 3)$. Goal $G = K_a a@p_1 \wedge K_a b@p_1$.

The following executions assume a hidden true state where a is positioned at p_1, b is positioned at p_4, and the position of the landmarks is: $r@p_2$, $q@p_4$ and $l@p_6$.

Fixed-agent protocol. **Experiments:** $\langle 0.3s, 1.9s, 2.1 \rangle$.

1. **A**: B, do you see l?
2. **B**: No, I do not see l.*
3. **A**: B, do you not see q?
4. **B**: No, I do see q.*
5. **A**: B, move clockwise twice. Do you not see l?
6. **B**: No, I do see l.*
7. **A**: I move anticlockwise. I move clockwise.
8. **A**: B, move clockwise.

In order for a to achieve the goal he needs to learn the position of b in terms of landmarks *and* the position of the landmarks on the grid. After the first two questions, a knows b sees q. He then moves b to a different location and a learns that b sees l. Up to this point, a knows that he is in p_1 since he sees no landmark, has learned that l is clockwise next to q and that b is now at the same position with l.

Then, a moves to p_2, and by seeing r learns the actual positions of the landmarks, and, subsequently, the position of b.

Last-agent protocol. **Experiments:** $\langle 0.3s, 1.9s, 3.3\rangle$.

1. **A**: B, do you see l?
2. **B**: No, I do not see l.*
3. **B**: A, do you see q?
4. **A**: No, I do not see q.*
5. **A**: B, do you not see q?
6. **B**: No, I do see q.*
7. **B**: A, do you see r?
8. **A**: No, I do not see r.*
9. **A**: B, move clockwise twice. Do you not see l?
10. **B**: No, I do see l.*
11. **B**: A, do you see l?
12. **A**: No, I do not see l.*
13. **A**: I move anticlockwise. I move clockwise. B, move clockwise.

In the above execution, we see that both agents, when they are the *planning agent*, try first to reduce their uncertainty. We have a constant exchange of information, up to the point where a happens to become the planning agent while he knows the hidden true state. If a was at p_1, his last response would have made his position known to b, as well as the fact that he knows b's position as well, allowing b to achieve the goal $K_bK_a a@p_1 \wedge K_bK_a b@p_1$.

Vol-mutex protocol. **Experiments:** $\langle 0.3s, 1.9s, 1.7\rangle$.

1. **A**: B, do you see l?
2. **B**: No, I do not see l.
3. **B**: I do see q.* (*volunteering*)
4. **A**: B, move clockwise twice. Do you not see l?
5. **B**: No, I do see l.*
6. **A**: I move anticlockwise. I move clockwise. B, move clockwise.

Literals $b@q$, $b@r$ and $b@l$ are mutexes: since b can be in only one position, he can see only one landmark. When he is asked about l and he responds negatively, he volunteers the information of what he actually sees, saving a from asking another question.

A general strategy for solving the problem would be for a to move to a position with a landmark, asking b if he sees the same landmark, and if he does not, move him to another position and ask him again the same question. Such a policy is good since it takes into account the issue that replans may be needed. Though such a plan is possible to be found by our approach, the fact that a state is assumed as true in every planning phase leads to optimistic plans, in terms of that assumption. Imagine a assuming a state where he is right next to landmark l and b is in the same position as l. From a's point of view, the plan where he asks first b if he sees l and then a moves left and sees l himself, is the same as first moving to l and then asking b. Yet, the second plan is better considering the possibility of replanning since he at least knows the position of one landmark, while in the first he only learns where b isn't.

Situated dialogue

In this problem, we have a table of size 6x6 (with the $(0, 0)$ coordinates on the top left), six objects (Q, W, E, R, T, Y) placed on it in different positions, and two agents a and b. Each agent can see only part of the table: a can see the entire table *except* of *five* positions which are hidden to him, and it is known that object Q is placed in one of these positions.

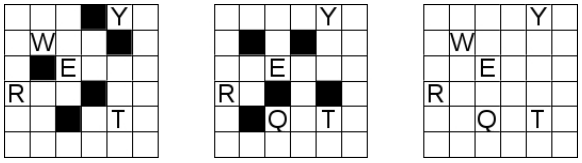

Figure 1: Situated dialog example: on the left we see what agent a knows, in the middle what b knows and on the right the hidden, true state. Covered positions on the table indicate the positions that the respective agent cannot see.

Similarly, there are five, *different that a's*, positions which are hidden for agent b, and it is known that object W is in one of them. In other words, it is known that each agent can see 5 out of 6 objects and the the positions of four of them (E, R, T, Y) are known to both, leading to 25 possible initial states (5 for the position of W and 5 for the position of Q).

Objects E, R, T and Y can be moved in four directions by b, and agents can communicate only spatial relationships: they cannot communicate the position of the objects W and Q but they can communicate whether that object is on left/right/over/under another object. The goal is for a to know the position of Q and for agent b to know the position of W.

We have four physical actions for each of the four objects, "move-object-X-right/left/up/down", each with conditionals effect $X@p_{x,y} \rightarrow \neg X@p_{x,y} \wedge X@p_{x',y'}$, for $X \in \{E, R, T, Y\}$, and for all positions $p_{x,y}$ to a new position $p_{x',y'}$, depending on which direction the object is move.

Agent a can communicate whether object W is on the left/right/over/under of one of the E, R, T, and Y, and agent b similarly, for object Q. This means that we have in total 32 sensing actions: 16 sensing actions of a communicating **sense**$[b](K_aWrZ)$, and 16 sensing actions for b: **sense**$[b](K_bQrZ)$, both with $r \in \{right, left, over, under\}$ and $Z \in \{E, R, T, Y\}$. Literals XrZ are derived literals with definitions indicating whether the spatial relationship between X and Z holds. As an example, $WoverE$ is derived by a DNF formula with terms $W@p_{x,y} \wedge E@p_{x,y+1}$, where x, y are the five possible initial positions of W (since W cannot be moved there is no reason to define the derived literal over all possible positions of the grid).

The goal is written as $G = (K_aQ@p_{0,3} \vee K_aQ@p_{1,4} \vee K_aQ@p_{3,3} \vee K_aQ@p_{2,1} \vee K_aQ@p_{4,2}) \wedge (K_bW@p_{1,1} \vee K_bW@p_{1,3} \vee K_bW@p_{3,2} \vee K_bW@p_{3,4} \vee K_bW@p_{4,1})$.

The true hidden initial state for the executions below is shown in Figure 1.

Fixed agent protocol. **Experiments:** $\langle 0.92s, 2.86s, 2.1\rangle$.

A: B, do you know Q is left of E?
B: No, I do not know.*
A: B, do you know Q is left of Y?
B: No, I do not know.*
A: B, do you know Q is under of Y?
B: No, I do not know.*
A: B, move E down. Do you know Q is under E?
B: Yes, I do know.
A: B, move R right. Move R up. I do know W over R.

Initially, agent a tries to find out the position of Q, learning that Q does *not* have a spatial relationship with any object. Object E is then moved next to the remaining possible positions of Q, creating the necessary relationships. When a learns the position of Q, object R is moved to allow a to communicate his knowledge of the position of W.

Volunteering protocol. **Experiments:** $\langle 1.1s, 2.9s, 1.4\rangle$.

A: B, do you know Q is left of E?
B: No, I do not know Q is left of E.
B: I do not know Q under Y.* (*volunteering*)
A: B, do you know Q is left of Y?
B: No, I do not know Q is left of Y.*
A: B, move E down. B, do you know Q is under E?
B: Yes, I do know Q is under E.
A: B, move R right. B, move R up. I do know W is over R.

After the first question of a, to which b replies negatively, b volunteers that he *also* does not know that Q is under Y. If Q *had* a spatial relationship with another object, b would choose to volunteer that relationship, after which a would know the hidden true state and with one planning phase achieve the goal. Since there is no such relationship, volunteering that Q does *not* have a spatial relationship with an object removes 5 states from the set of possible initial states.

The Lights problem

In this problem there are four lights (l_1, l_2, l_3, l_4) and three agents (a, b, c). Initially it is known that at least one of the lights is on. No agent can see the lights themselves, but agent b can sense whether at least one of the lights l_1 and l_2 are on $((l_1 \vee l_2))$. Similarly, agent c can sense $(l_3 \vee l_4)$, while a cannot sense anything about the physical world. Additional to these two passive sensors, there are ten sensing actions: **sense**$[b](K_c(l_3 \vee l_4)$, **sense**$[a](K_b(l_1)$, **sense**$[a](K_b(l_2)$, **sense**$[a](K_bK_cL)$, **sense**$[a](K_b\neg K_cL)$, **sense**$[a](K_bK_c\neg L)$, and **sense**$[a](K_b\neg K_c\neg L)$, with $L \in \{l_3, l_4\}$. Simply, c can communicate his knowledge about what he senses only to b, and b can communicate to a his knowledge about the lights he can sense and his knowledge about the knowledge of c concerning l_3 and l_4. Lastly, there are four physical actions "toggle(L)", for $L \in \{l_1, l_2, l_3, l_4\}$, that toggle light L: turn it on if it was off, and off if it was on, whose owner is a. The goal is for a to know that all lights are on: $K_al_1 \wedge K_al_2 \wedge K_al_3 \wedge K_al_4$.

The true hidden initial state for the execution is the one where only l_2 and l_4 are on. In the execution we show the actual response of b since a is the planning agent.

Fixed agent protocol. **Experiments:** $\langle 0.8s, 1.5s, 3.2\rangle$.

1. **A**: B, tell *me*, do you know that C does not know that l_4 is off?
2. **B**: No, I do not know.*
3. **A**: C, tell B whether you know $l_3 \vee l_4$.
4. **C**: I told B.
5. **A**: B, tell *me*, do you know that C knows that l_4 is off?
6. **B**: No, I do not know that.*
7. **A**: I toggle the second light. B, tell *me*, do you know l_1 is on?
8. **B**: No, I do not know it.*
9. **A**: I toggle the first, the second and the third light. C, tell B whether you know $l_3 \vee l_4$.
10. **C**: I told B.
11. **A**: B, tell *me*, do you not know that C knows that l_4 is on?
12. **B**: Yes, I do.
13. **A**: I toggle the fourth light. C, tell B whether you know $l_3 \vee l_4$.
14. **C**: I told B.
15. **A**: B, tell *me*, do you know that C knows that l_3 is on?
16. **B**: Yes, I do know that.
17. **A**: I toggle the fourth light.

Agent b's first response allows a to derive that $l_1 \vee l_2$ is true. Otherwise, b would know $l_3 \vee l_4$ is true (at least one of l_k must be initially on) and since c can sense $l_3 \vee l_4$, b would also know that c could not know l_4 was off. After c tells b what he sensed (4), b knows that c knows either both l_3 and l_4 to be off, or that at least one is on. Since b does not know that c knows l_4 is off, a is able to derive that $l_3 \vee l_4$ is true. Toggling l_2 at step 7, while $l_1 \vee l_2$ is true, creates a situation where either both are off or l_1 is definitely on. The response of b allows a to derive both are off, and turning them on at step 9. Similarly, for achieving $l_3 \wedge l_4$.

Related Work

In recent years, there has been a growing interest in multiagent epistemic planning with a number of works placing emphasis on different aspects of the problem. Some place the focus on expressivity and modeling (Baral et al. 2012; Cooper et al. 2016), others in distributed computation and coordination (Engesser et al. 2015), while the most closely related approaches focus on computational issues and the use of classical planners (Brenner 2010; Brafman, Shani, and Zilberstein 2013). The works most relevant to ours are (Muise et al. 2015) and (Cooper et al. 2016). A key difference to our approach is they can only represent beliefs about literals, not about arbitrary formulas. This is how they manage to reason about nested beliefs without using explicit or implicit Kripke structures. The complexity of planning in dynamic epistemic logic and restricted versions of it are studied in (Aucher and Bolander 2013; Charrier, Bastien, and Schwarzentruber 2016; Löwe, Pacuit, and Witzel 2011; Bolander, Jensen, and Schwarzentruber 2015).

Conclusion

We have extended the belief tracking formulation of Kominis and Geffner to the on-line setting where plans are supposed to work for the true hidden state as revealed by the observations, and have developed an alternative translation into classical planning for selecting actions within a replanning architecture. Planning is done from the perspective of the agents themselves that have beliefs about the world and nested beliefs about each other. As in the single-agent setting, the replanning approach ensures that goals are reached in a bounded number of episodes provided that dead-ends are not reached. We have shown that interesting agent dialogues can arise in the proposed setting where agents collaborate by requesting or volunteering information in a goal-directed manner. The account, however, is restricted to public actions only, and even with this restriction, the computational approach is not yet scalable, as only problems with tens of possible initial states can be handled in this way. One way for scaling up further is by adapting the techniques that have been used to improve scalability in the single-agent setting.

Acknowledgements

We thank François Schwarzentruber for useful comments. The work is partially supported by grant TIN2015-67959-P, MEC, Spain.

References

Albore, A.; Palacios, H.; and Geffner, H. 2009. A translation-based approach to contingent planning. In *Proc. IJCAI-09*, 1623–1628.

Aucher, G., and Bolander, T. 2013. Undecidability in epistemic planning. In *Proc. IJCAI*, 27–33.

Baral, C.; Gelfond, G.; Pontelli, E.; and Son, T. C. 2012. An action language for reasoning about beliefs in multi-agent domains. In *Proc. of the 14th International Workshop on Non-Monotonic Reasoning*.

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

Bonet, B., and Geffner, H. 2000. Planning with incomplete information as heuristic search in belief space. In *Proc. of AIPS-2000*, 52–61.

Bonet, B., and Geffner, H. 2014. Flexible and scalable partially observable planning with linear translations. In *Proc. AAAI*, 2235–2241.

Brafman, R. I., and Shani, G. 2012a. A multi-path compilation approach to contingent planning. In *Proc. AAAI*.

Brafman, R. I., and Shani, G. 2012b. Replanning in domains with partial information and sensing actions. *Journal of Artificial Intelligence Research* 45(1):565–600.

Brafman, R. I.; Shani, G.; and Zilberstein, S. 2013. Qualitative planning under partial observability in multi-agent domains. In *Proc. AAAI*.

Brenner, M. 2010. Creating dynamic story plots with continual multiagent planning. In *Proc. AAAI*.

Charrier, T.; Bastien, M.; and Schwarzentruber, F. 2016. On the impact of modal depth in epistemic planning. In *Proc. IJCAI*, 1030–1036.

Cooper, M.; Herzig, A.; Maffre, F.; Maris, F.; and Regnier, P. 2016. A simple account of multiagent epistemic planning. In *Proc. ECAI*.

Engesser, T.; Bolander, T.; Mattmüller, R.; and Nebel, B. 2015. Cooperative epistemic multi-agent planning with implicit coordination. In *Proc. Workshop on Distributed and Multi-Agent Planning (DMAP-15)*, 68–76.

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

Helmert, M. 2006. The Fast Downward planning system. *Journal of Artificial Intelligence Research* 26:191–246.

Kominis, F., and Geffner, H. 2015. Beliefs in multiagent planning: From one agent to many. In *Proc. ICAPS*, 147–155.

Löwe, B.; Pacuit, E.; and Witzel, A. 2011. DEL planning and some tractable cases. In *Logic, Rationality, and Interaction*. Springer. 179–192.

Muise, C.; Belle, V.; Felli, P.; McIlraith, S.; Miller, T.; Pearce, A. R.; and Sonenberg, L. 2015. Planning over multi-agent epistemic states: A classical planning approach. In *Proc. AAAI*.

Rintanen, J. 2004. Complexity of planning with partial observability. In *Proc. ICAPS-2004*, 345–354.

Van Benthem, J. 2011. *Logical dynamics of information and interaction*. Cambridge University Press.

van Ditmarsch, H., and Kooi, B. 2008. Semantic results for ontic and epistemic change. *Logic and the Foundations of Game and Decision Theory (LOFT 7)* 87–117.

van Ditmarsch, H.; van der Hoek, W.; and Kooi, B. 2007. *Dynamic Epistemic Logic*. Springer.