Dissertation Abstract: Distributed Privacy-preserving Multi-agent Planning

Andrea Bonisoli
Univrsità degli Studi di Brescia
Dipartimento di Ingegneria dell'Informazione
Via Branze, 38 - 25123 Brescia (IT)
andrea.bonisoli@unibs.it

Abstract

Planning is a well known and studied field of Artificial Intelligence. Multi-Agent Planning concerns the construction of plans for a group of autonomous agents that can interact. The aim of multi-agent planning is to automatically find a solution such that, if every agent executes successfully his plan, the environment changes to a goal state. The solution can be found either by centralized or distributed algorithms.

Introduction

The aim of planning, a well known field of Artificial Intelligence, is the automated synthesis of partially ordered sequences of actions, called plans, that can be executed in given settings by one or more agents. A plan is called a solution for a given problem if its execution from an initial known state achieves the problem goals.

Multi-agent planning can be seen as an extension of classical planning and in (De Weerdt and Clement 2009) is defined as "the problem of planning by and for a group of agents". This definition is intentionally general and therefore includes many different approaches. Multi-agent planning can be applied to a wide range of problems, from team of robots involved in space exploration or disaster recovery to logistics chains involving different companies. Whenever there are multiple actors that operate in the setting and they need to decide the best course of action, multi-agent planning can be used to find a solution. It is also worth noting that, although multi-agent planning is not a new research field, many important contributions in this topic are quite recent. One of the main motivations in multi-agent planning is that some or all agents have private knowledge that cannot be communicated to other agents during the planning process and the plan execution.

Problem Definition

Different authors use some slightly different definition of multi-agent planning, however the most common definition of this problem relies on the multi-agent language called `MA-STRIPS`, a minimal extension of the `STRIPS` planning language, which was first described in (Brafman and Domshlak 2008) and then adopted by several authors (Jonsson and Rovatsos 2011; Nissim and Brafman 2012; Brafman and Domshlak 2013; Štolba and Komenda 2013; Štolba, Fišer, and Komenda 2015a). Other definition of the problem are also possible (Torreño, Onaindia, and Sapena 2014; 2015; Bonisoli et al. 2014), nonetheless `MA-STRIPS` is a simple and effective language to represent the *cooperative* multi-agent planning.

Formally in a multi-agent planning task is given a set of k agents $\Phi = \{\varphi_i\}_{i=1}^k$ and a 4-tuple $\Pi = \langle P, \{A_i\}_{i=1}^k, I, G\rangle$ where:

- P is a finite set of atomic propositions, $I \subseteq P$ encodes the initial state and $G \subseteq P$ encodes the goal conditions, as in the classical planning;
- A_i, for $1 \leq i \leq k$, is the set of actions of the agent φ_i . Every action $a \in A = \bigcup A_i$ is given by its preconditions and effects; every agent has a different set of actions, i.e. $A_i \cap A_j = \emptyset$ if $i \neq j$.

A solution is a partially ordered sequence of actions such that each action in the plan is associated with a single agent. If there is only one agent in the problem, that is n = 1, this definition reduces exactly to a `STRIPS` problem. Therefore, one can see `MA-STRIPS` as a partition of the set of actions of a `STRIPS` problem and assignment of one agent to each partition set. This rather simple extension of the language is easy to understand, but it is quite limited: for example in `MA-STRIPS` it is not possible to define different goals for different agents. The model was also extended for the case of *self-interested* agents (Nissim and Brafman 2013). Another interesting proposal for a standard description language that allow for a more direct comparison between systems and approaches is MA-PDDL (Kovacs 2012), which is an extension of the PDDL language used by the international planning competitions. MA-PDDL is aimed at solving most of the limitations of other multi-agent planning languages. This language can be used to describe many different multi-agent systems and was used during the first Competition of Distributed and Multiagent Planners (Štolba, Komenda, and Kovacs 2015).

Given the partition of actions in `MA-STRIPS`, is is possible to distinguish between *private* and *public* actions and propositions. An atom is *private* for agent φ_i if it is required and affected only by the actions of φ_i. An action of φ_i is *pri-*

vate if all its preconditions and effects are private. All other action are classified as *public*. This definitions are quite important and widely used because are the basis for defining the privacy of the agents.

State of the Art

One of the first distributed algorithms to solve a `MA-STRIPS` problem is presented in (Brafman and Domshlak 2008). This approach is based on a distributed solver for constraint satisfaction problems: using the notions of public and private actions it is possible to define constraints for the agents and use them to agree on a solution and let each agent to autonomously plan their private part. The experimental results of this algorithm are presented in (Nissim, Brafman, and Domshlak 2010). Although the basic idea was highly innovative, the overall efficiency of the algorithm was insufficient and, therefore, was only able to solve problems of modest size and complexity. In the same work it is defined an upper limit to the complexity resolutive for the problems `MA-STRIPS` that depends exponentially on two parameters that quantify the level of coupling of the system.

To increase scalability than the number of agents involved in (Jonsson and Rovatsos 2011) is proposed an approach with iterative refinement of the plans: each agent involved may in turn change their plan to update it according to the actions planned by other agents. In this way a process is obtained which slowly converges towards a solution plan for the problem. Unfortunately, this convergence is not guaranteed in all cases, as it can not formally prove the optimality of the solutions produced by the algorithm. However it uses in it known technologies and planners available oof the shelf and can therefore take advantage of the latest innovations in the field of research into classical planning. It can also be used to solve problems with non-cooperative agents, but only if it is not required to keep private the plans of the agents.

In contrast in (Nissim and Brafman 2012; 2013) is presented the extension to the case of multi-agent of the known search algorithm `A*`, called `MAD-A*`. This new algorithm maintains the properties known from classic case if the heuristic function used is permissible and also ensures a level of privacy to agents involved which the authors define as *weak privacy*. Moreover, in (Nissim and Brafman 2014) is also described an additional feature of `MAD-A*` which allows him to prune the search tree of a larger number of nodes than classic algorithm. Finally in (Brafman 2015) describes a modification to the algorithm that allows it to further increase the level of privacy and exchanging fewer messages between agents, but unfortunately for this version are not present experimental results.

Instead in (Torreño, Onaindia, and Sapena 2012; 2013; 2014; 2015) is presented an algorithm based on forward planning with partial ordering and exchange of incomplete plans. In this case the problem of the proposed model is different than that described in `MA-STRIPS`, however, the two models are comparable and the most recent implementation of this algorithm is able to solve the problems described with `MA-PDDL`. It is noteworthy that the privacy of agents is kept obscuring a part of the partial plans that are exchanged during the planning of the agents.

Conversely in (Maliah, Shani, and Stern 2014) is presented the algorithm `GPPP` which uses explicitly two different phases to identify a solution plan. During the first phase the agents want to identify a joint coordination scheme through a high-level planning of a relaxed problem using only the public actions of the problem. In the second phase, each agent individually shall seek a local plan that can support public actions agreed in coordination shcema. It is evident that, since the high-level planning has been carried out on a relaxed problem, it is possible that during the second phase some of the agents are not able to find a viable solution. In this case the algorithm executes again the first phase to determine a different coordination scheme.

Finally in (Tozicka, Jakubuv, and Komenda 2014; 2015) describes a new and recent approach based on finite state machines. This algorithm called `PSM` uses finite automata to represent a set of plans and then use public screenings and intersection of automata to compute a solution. Since public projections do not contain private information, these are exchanged between agents that have generated a plan and if the intersection between the projections received from other agents and its plan is not empty, then the plan is a solution. The planner based on this algorithm has shown excellent results in the competition for distributed planners, in particular in the fully distributed track.

It is also important to note the studies regarding the heuristic evaluation functions because of their importance during the research phase: a more accurate heuristic could bring significant performance gains for algorithms that use it. For this reason, in (Štolba, Fišer, and Komenda 2015b) shows a comparison between the performance of different heuristics multi-agent. Many of those presented are in fact an adaptation of heuristics known for classical planning adapted to the case multi-agent. Most of these heuristics is an adaptation of the very heuristic notes taken by the famous classic planner `Fast-forward` described in (Hoffmann and Nebel 2001). It is for example the case of (Štolba and Komenda 2013; 2014) which distributes between agents graphs of relaxed schedule used by `fast-forward`. The authors Torreño, Onaindia, and Sapena on the contrary using an approach based on the latest graphs domain transition, although in their last work (Torreño, Onaindia, and Sapena 2015) show that a hybrid approach between the two different heuristics may result on average in more accurate heuristic. In (Maliah, Shani, and Stern 2014) is instead presented a multi-agent heuristi based on landmark that can also to maintain the privacy of the agents, while in (Štolba, Fišer, and Komenda 2015a) describes another heuristic based on lankmark but admissible. In (Maliah, Shani, and Stern 2015) a new heuristic based on pattern database is proposed and shoed to be more effectiv on some domains. It should however be noted that the distributed heuristic algorithms can be substantially different from the classic ones that inspired them and in general could also get different results with respect to their central counterparts.

Privacy-preserving Multi-agent Planning

One fundamental problem of `MA-STRIPS` is that it cannot express the privacy of the agents beyond the definitions of public and private facts and actions and does not fully guarantee the privacy of the involved agents when at least one public proposition is confidential (i.e., it should be kept hidden from some agent). For example a proposition shared by two agents should be public for every other involved agent.

Therefore in this section, we propose a more general model that preserves the privacy of the involved agents. The model of multi-agent planning that is most similar to the one we propose here is the model adopted by MAP-POP (Torreño, Onaindia, and Sapena 2012). This model was first presented in (Bonisoli et al. 2014).

A *privacy-preserving multi-agent planning problem* for a set of agents $\Sigma = \{\alpha_i\}_{i=1}^n$ is a tuple $\langle \{A_i\}_{i=1}^n, \{F_i\}_{i=1}^n, \{I_i\}_{i=1}^n, \{G_i\}_{i=i}^n, \{M_i\}_{i=1}^n \rangle$ where:

- A_i is the set of actions agent α_i is capable of executing, and such that for every pair of agents α_i and α_j, $A_i \cap A_j = \emptyset$;
- F_i is the set of relevant facts for agent α_i;
- $I_i \subseteq F_i$ is the portion of the initial state relevant for α_i;
- $G_i \subseteq F_i$ is the set of goals for agent α_i;
- $M_i \subseteq F_i \times \Sigma$ is the set of messages agent α_i can send to the other agents.

Facts and actions are literals and pair $\langle \mathit{Pre},\mathit{Eff} \rangle$, respectively, where *Pre* is a set of positive literals and *Eff* is a set of positive or negative literals. Let $X+/X-$ denote the positive/negative literals in set X, respectively. Let $\mathcal{G}$ be the graph induced by $\{M_i\}_{i=1}^n$, where nodes represent agents, and edges represent possible information exchanges between agents; i.e., an edge from node α_i to node α_j labelled p represents the agent α_i's capability of sending p to agent α_j. In order to have well-defined sets $\{M_i\}_{i=1}^n$, $\forall \alpha_i, \alpha_j \in \Sigma$, $\forall p$ s.t. $p \in F_i$ and $p \in F_j$, there should be a path in $\mathcal{G}$ from the node representing α_i to the node representing α_j formed by edges labelled p, if $p \in I_i$, or $\exists a \in A_i \cdot p \in \mathit{Eff}{+}(a)$, or $\exists a \in A_i \cdot p \in \mathit{Eff}{-}(a)$.

A plan for a multi-agent planning problem is a set $\{\pi_i\}_{i=1}^n$ of n single-agent plans. Each single agent plan is a sequence of happenings. Each happening of agent α_i consists of a (possibly empty) set of actions of α_i, and a (possibly empty) set of exogenous events. Exogenous events are facts that become true/false because of the execution of actions of other agents; in this sense, these events cannot be controlled by agent α_i. Formally, $\pi_i = \langle h_i^1, \ldots, h_i^l \rangle$, $h_i^j = \langle A_i^j, E_i^j \rangle$, $A_i^j \subseteq A_i$, $E_i^j \subseteq \bigcup_k F_k$, for $i = 1 \ldots n$, $j = 1 \ldots l$, $k \in \{1, \ldots, i-1, i+1 \ldots, n\}$.

The execution of plan π_i generates a state trajectory, $\langle s_i^0, s_i^1, \ldots, s_i^l \rangle$, where $s_i^0 = I_i$, and a sequence of messages, $\langle m_i^1, \ldots, m_i^l \rangle$, each of which is a set of literals. At planning step j agent α_i sends literal $p/\neg p$ if either α_i executes an action that makes p true/false or α_i receives the message that lets the agent know p becoming true/false. In this latter case, α_i forwards the received message $p/\neg p$ to the agents it is connected to. For every planning step, the forwarding is repeated $n-1$ times so that, if sets $\{M_i\}_{i=1}^n$ are well-defined, every agent α_k such that $p \in F_k$ is advised that p becomes true or false (the length of the shortest path between any pair of nodes in the graph induced by $\{M_i\}_{i=1}^n$ is at most $n-1$). At planning step j agent α_i sends literal $p/\neg p$ if either α_i executes an action that makes p true/false or α_i receives the message that lets the agent know p becoming true/false. In this latter case, α_i forwards the received message $p/\neg p$ to the agents it is connected to. For every planning step, the forwarding is repeated $n-1$ times so that, if sets $\{M_i\}_{i=1}^n$ are well-defined, every agent α_k such that $p \in F_k$ is advised that p becomes true or false (the length of the shortest path between any pair of nodes in the graph induced by $\{M_i\}_{i=1}^n$ is at most $n-1$).

Formally, state s_i^j and message m_i^j are defined as follows, for $j = 1 \ldots l$ and $k = 1 \ldots i-1, i+1 \ldots n$.

$$s_i^j = s_i^{j-1} \cup \bigcup_{a \in A_i^j} \mathit{Eff}{+}(a) \cup E{+}_i^j \setminus \bigcup_{a \in A_i^j} \mathit{Eff}{-}(a) \setminus E{-}_i^j;$$

$$m_i^j = \bigcup_k \mathit{sm}{+}_{i \to k}^j(n-1) \cup \bigcup_k \mathit{sm}{-}_{i \to k}^j(n-1), \text{ with}$$

$$\mathit{sm}{+}_{i \to k}^j(t) = \Big\{ \langle p, \alpha_k \rangle \mid \langle p, \alpha_k \rangle \in M_i,\ p \in \bigcup_{a \in A_i^j} \mathit{Eff}{+}(a) \cup \mathit{rm}{+}_i^j(t-1) \Big\},$$

$$\mathit{sm}{-}_{i \to k}^j(t) = \Big\{ \langle \neg p, \alpha_k \rangle \mid \langle p, \alpha_k \rangle \in M_i,\ p \in \bigcup_{a \in A_i^j} \mathit{Eff}{-}(a) \cup \mathit{rm}{-}_i^j(t-1) \Big\},$$

$$\mathit{rm}{+}_i^j(t) = \Big\{ p \mid \langle p, \alpha_i \rangle \in \bigcup_k \mathit{sm}{+}_{k \to i}^j(t) \Big\},$$

$$\mathit{rm}{-}_i^j(t) = \Big\{ p \mid \langle \neg p, \alpha_i \rangle \in \bigcup_k \mathit{sm}{-}_{k \to i}^j(t) \Big\},$$

$$\mathit{rm}{+}_i^j(0) = \mathit{rm}{-}_i^j(0) = \emptyset.$$

Intuitively, for planning step j, $\mathit{sm}{+}_{i \to k}^j(t)/\mathit{sm}{-}_{i \to k}^j(t)$ is the set of positive/negative literals that at the t-th forwarding step ($t = 1 \ldots n-1$) agent α_i sends to agent α_k; $\mathit{rm}{+}_i^j(t)/\mathit{rm}{-}_i^j(t)$ is the set of positive/negative literals that at the t-th forwarding step agent α_i receives. Note that propositional planning assumes that at every planning step the execution of actions is instantaneous, and hence the information exchanges also happens instantaneously.

We say that the single-agent plan π_i is *consistent* if the following conditions hold for $j = 1 \ldots l$ and $t = 1 \ldots n-1$:

(1) $E{+}_i^j = \bigcup_t \mathit{rm}{+}_i^j(t)$, $E{-}_i^j = \bigcup_t \mathit{rm}{-}_i^j(t)$;

(2) $\forall a, b \in A_i^j \cdot \mathit{Pre}(a) \cap \mathit{Eff}{-}(b) = \mathit{Pre}(b) \cap \mathit{Eff}{-}(a) = \emptyset$;

(3) $\forall a, b \in A_i^j \cdot \mathit{Eff}{+}(a) \cap \mathit{Eff}{-}(b) = \mathit{Eff}{+}(b) \cap \mathit{Eff}{-}(a) = \emptyset$;

(4) $\forall a \in A_i^j, \forall e \in E-_i^j \cdot Pre(a) \cap e = \emptyset = Eff+(a) \cap e = \emptyset$.

Basically, (1) asserts that at planning step j all the exogenous events for agent α_i are the positive/negative literals α_i receives during the information exchange, i.e., (1) guarantees that these events are generated by some other agent; (2) and (3) assert that at planning step j agent α_i executes no pair of mutually exclusive actions; finally, (4) asserts that at planning step j agent α_i executes no action that is mutex with some action executed by other agents.

Let $\langle s_i^0, s_i^1, \ldots, s_i^l \rangle$ be the state trajectory generated by single-agent plan π_i. Plan π_i is executable if $Pre(a) \subseteq s_i^{j-1}$, $\forall a \in A_i^j, j = 1 \ldots l$. Plan π_i is valid for agent α_i if it is executable, consistent, and achieves the goals of agent α_i, i.e., $G_i \subseteq s_i^l$. A multi-agent plan $\{\pi_i\}_{i=1}^n$ is a solution of the multi-agent privacy-preserving planning task if single-agent plan π_i is valid for agent α_i, for $i = 1 \ldots n$.

The main difference with existing models to multi-agent planning, like (Torreño, Onaindia, and Sapena 2012), is related to sets $\{M_i\}_{i=1}^n$ and the purpose for which agents use them. Essentially, M_i determines the messages agent α_i can generate during the execution of its plan, that can be sent to other agents without loss of privacy.

Conclusion

Multi-agent planning is an open field of research as many new contributions in recent years have showed and there are still some open issues and challenges to address. First of all, many theoretical properties of some settings of multi-agent planning are not well known. For example it is still unknown the actual complexity of different settings or what make them so difficult. Also, while the theoretical properties of multi-agent systems are well studied in the multi-agent system community, the relation to planning is not a well studied topic and further research work may improve the understanding of the multi-agent planning problem.

Furthermore, while privacy issues are strong reasons for using distributed algorithms, the definition of privacy in multi-agent planning is debated, e.g., what agents should kept private information (state variables, actions, goals) and what minimal information they should exchange in order to be able to construct a joint plan remain an open question. While the distinction between public and private fluents and actions is a first step towards the definition of privacy, it is too weak for many settings not involving cooperative agents. It is unknown whether partial observability can cope with the privacy issues.

References

Bonisoli, A.; Gerevini, A. E.; Saetti, A.; and Serina, I. 2014. A privacy-preserving model for the multi-agent propositional planning problem. *Distributed and Multi-Agent Planning* 25–29.

Brafman, R. I., and Domshlak, C. 2008. From one to many: Planning for loosely coupled multi-agent systems. In *ICAPS*, 28–35.

Brafman, R. I., and Domshlak, C. 2013. On the complexity of planning for agent teams and its implications for single agent planning. *Artificial Intelligence* 198:52–71.

Brafman, R. I. 2015. A privacy preserving algorithm for multi-agent planning and search. *Joint Conference on Artificial Intelligence (IJCAI 2015)* 1530–1536.

De Weerdt, M., and Clement, B. 2009. Introduction to planning in multiagent systems. *Multiagent and Grid Systems* 5(4).

Hoffmann, J., and Nebel, B. 2001. The ff planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research* 253–302.

Jonsson, A., and Rovatsos, M. 2011. Scaling up multiagent planning: A best-response approach. In *Proceedings of the 21st International Conference on Automated Planning and Scheduling*.

Kovacs, D. L. 2012. A multi-agent extension of pddl3. *WS-IPC 2012* 19.

Maliah, S.; Shani, G.; and Stern, R. 2014. Privacy preserving landmark detection. In *Proceedings of ECAI*, volume 14.

Maliah, S.; Shani, G.; and Stern, R. 2015. Privacy preserving pattern databases. *Distributed and Multi-Agent Planning (DMAP-15)* 9.

Nissim, R., and Brafman, R. I. 2012. Multi-agent a* for parallel and distributed systems. In *Proceedings of the Workshop on Heuristics and Search for Domain-Independent Planning*, 42–51. ICAPS.

Nissim, R., and Brafman, R. I. 2013. Cost-optimal planning by self-interested agents. In *AAAI*.

Nissim, R., and Brafman, R. 2014. Distributed heuristic forward search for multi-agent planning. *Journal of Artificial Intelligence Research* 51(1):293–332.

Nissim, R.; Brafman, R. I.; and Domshlak, C. 2010. A general, fully distributed multi-agent planning algorithm. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, 1323–1330. International Foundation for Autonomous Agents and Multiagent Systems.

Štolba, M., and Komenda, A. 2013. Fast-forward heuristic for multiagent planning. In *Proceedings of Distributed and Multi-agent Planning (DMAP) Workshop of 23rd International Conference on Automated Planning and Scheduling (ICAPS'13)*, volume 120.

Štolba, M., and Komenda, A. 2014. Relaxation heuristics for multiagent planning. In *24th International Conference on Automated Planning and Scheduling (ICAPS)*, 298–306.

Štolba, M.; Fišer, D.; and Komenda, A. 2015a. Admissible landmark heuristic for multi-agent planning. In *Twenty-Fifth International Conference on Automated Planning and Scheduling*.

Štolba, M.; Fišer, D.; and Komenda, A. 2015b. Comparison of rpg-based ff and dtg-based ff disrtibuted heuristics. *Proceedings of Distributed and Multi-Agent Planning Workshop(DMAP-15)* 77.

Štolba, M.; Komenda, A.; and Kovacs, D. L. 2015. Competition of distributed and multiagent planners, http://agents.fel.cvut.cz/codmap/.

Torreño, A.; Onaindia, E.; and Sapena, Ó. 2012. An approach to multi-agent planning with incomplete information. *Proceedings of ECAI*.

Torreño, A.; Onaindia, E.; and Sapena, Ó. 2013. Fmap: a heuristic approach to cooperative multi-agent planning. In *Proceedings of DMAP Workshop of ICAPS*, volume 13, 84–92.

Torreño, A.; Onaindia, E.; and Sapena, Ó. 2014. Fmap: Distributed cooperative multi-agent planning. *Applied Intelligence* 41(2):606–626.

Torreño, A.; Onaindia, E.; and Sapena, Ó. 2015. Global heuristics for distributed cooperative multi-agent planning. In *Twenty-Fifth International Conference on Automated Planning and Scheduling*.

Tozicka, J.; Jakubuv, J.; and Komenda, A. 2014. Generating multi-agent plans by distributed intersection of finite state machines. In *Proceedings of ECAI*.

Tozicka, J.; Jakubuv, J.; and Komenda, A. 2015. On internally dependent public actions in multiagent planning. *Distributed and Multi-Agent Planning (DMAP-15)* 18.