Privacy Preserving LAMA

Shlomi Maliah and Guy Shani and Roni Stern

Ben Gurion University of the Negev, Israel

Abstract

In collaborative privacy preserving planning (CPPP), multiple agents collaborate to achieve a goal while keeping certain facts about the world private. A prominent approach in the development of CPPP algorithms is to use components from single agent planners and adapt them to preserve privacy. In this short paper, we show how the components of LAMA, arguably one of the most successful single-agent planners, can be used in a privacy preserving manner. These components include alternating between a landmark heuristic and an FF heuristic, preferred operators and deferred heuristic evaluation. We integrate the components into the Greedy Privacy Preserving Planner, a state-of-the-art CPPP algorithm. The resulting algorithm performs better than other CPPP algorithms from the recent Competition of Distributed and Multiagent Planners.

1 Introduction

Collaborative privacy preserving planning (CPPP) is a recently introduced setting in which multiple agents cooperate to achieve joint goals while concealing certain facts. As a motivating scenario, consider an army organization outsourcing its food supply to external caterers. Caterers unloads packaged food in logistics center and army trucks deliver the packages to the various bases. The army and the caterer must plan together to deliver appropriate amounts of food to the army bases, but the army may not wish to discloser to the caterers the location of its bases or the number of soldiers in each base.

Brafman and Domshlak (2013) proposed an attractive framework for such planning problems called multi-agent STRIPS (MA-STRIPS), which has attracted much attention in recent years (Tozicka, Jakubuv, and Komenda, 2015; Torreno, Sapena, and Onaindia, 2015; Štolba and Komenda, 2014; Maliah, Shani, and Stern, 2014). The first Competition of Distributed and Multiagent Planners (CoDMAP), held last year, already featured many planners from 10 different groups Štolba, Komenda, and Kovacs (2015). Many successful CPPP algorithms borrow or adapt algorithmic components from the single-agent planning literature. For example, privacy preserving versions have been proposed for popular single-agent heuristics such as landmarks (Maliah, Shani, and Stern, 2014; Torreno, Sapena, and Onaindia, 2015; Štolba, Fišer, and Komenda, 2015), FF (Štolba and Komenda, 2014), and pattern databases (Maliah, Shani, and Stern, 2015). In this short paper we propose a CPPP algorithm that successfully adapts and uses key components of LAMA (Richter and Westphal, 2010), a renowned single-agent planner.

LAMA is perhaps one of the most successful single-agent planners. It uses a forward heuristic search algorithm employing both a landmark-based heuristic (Richter, Helmert, and Westphal, 2008) and the FF delete-relaxation heuristic (Hoffmann, 2001), and alternates between them (Röger and Helmert, 2010). In addition, LAMA introduced preferred operators and deferred heuristic evaluation (Richter and Helmert, 2009), which are both common components in state-of-the-art single-agent planning algorithms.

The main contribution of this work is in the integration of these components into the Greedy Privacy Preserving Planner (GPPP) (Maliah, Shani, and Stern, 2014), a state-of-the-art CPPP algorithm. Experiments with the CoDMAP benchmarks show that the resulting algorithm, which we call PP-LAMA, outperforms all previous CPPP algorithms. Some LAMA components were already adapted to preserve privacy by prior work. Štolba and Komenda (2014) and Maliah, Shani, and Stern (2014) proposed a privacy preserving versions of the FF and landmark heuristics, and Torreno, Sapena, and Onaindia (2015) proposed alternating multiple heuristics. We explain how the other components of PP-LAMA operate in a way that preserves privacy. In particular, we highlight the importance of deferred heuristic evaluation, which is especially useful for reducing the collaborative effort in computing privacy preserving heuristics. Moreover, we improve on the landmark detection algorithm used by Maliah, Shani, and Stern (2014), showing a simple modification that allows finding substantially more landmarks.

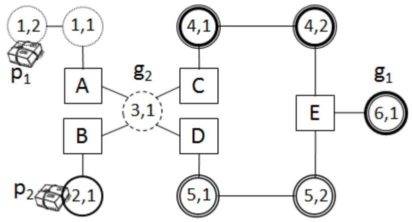

Figure 1: A logistics example.

2 Background

We now briefly describe the privacy preserving planning setting, the heuristics used by LAMA, and the GPPP algorithm.

2.1 Privacy Preserving Planning

An MA-STRIPS problem (Brafman and Domshlak, 2013) is represented by a tuple $\langle P, \{A_i\}_{i=1}^{k}, I, G\rangle$ where k is the number of agents, P is a finite set of facts (can be true of false), A_i is the set of actions agent i can perform, I is the start state, and G is the goal condition. Each action $a = \langle \text{pre}(a), \text{eff}(a)\rangle$ is defined by its preconditions (pre(a)), and effects (eff(a)). Preconditions and effects are logical formulas over P. A state is a conjunction of facts in P (true or false). The goal G is also a conjunction of facts. A solution is a plan that achieves G, i.e., a sequence of actions transforming the initial state (I) to a state that satisfies G.

Privacy-preserving MA-STRIPS extends MA-STRIPS by defining sets of variables and actions as private, known only to a single agent. Formally, a privacy-preserving MA-STRIPS problem defines a set of facts as public, denoted public(P), and a set of actions as public, where $\text{public}_i(A) \subset A_i$ are the public actions of agent i. It is assumed that when a public action is executed, all agents are aware of the execution, and view the public effects of the action. A privacy-preserving MA-STRIPS problem also defines for each agent i a set of facts and actions as private, denoted by $\text{private}_i(P)$ and $\text{private}_i(A)$, respectively. The identity of the private facts and actions must not be revealed to the other agent during planning and during execution. The public and private facts of each agent are disjoint, i.e., $\text{private}_i(P) \cap \text{public}(P)\emptyset$, and the union of the public facts and all private facts form the entire set of facts in the underlying MA-STRIPS problem, i.e., $\bigcup_{i=1}^{k} \text{private}_i(P) \cup \text{public}(P) = P$.

For ease of exposition we assume that all goals are public. A solution to a privacy-preserving MA-STRIPS problem, is a sequence of public and private actions. We say that the sequence of public actions in such a solution is a high-level, or public, plan that must be extended to a full plan using private actions of various agents. A high-level plan is said to be valid if each agent can plan independently to achieve the private preconditions on the public actions it executes in the high level plan Maliah, Shani, and Stern (2014).

Figure 2.1 illustrates a simple logistic example in which the agents are trucks tasked with delivering packages. The set of facts P represents the location of the two packages and six trucks. Each truck has three actions: move, load, and unload, corresponding to moving the truck between locations, loading a package and unloading it. Each truck is owned by a different company and may have different areas of operation, and no company wants to share its location and coverage (which locations it can reach) with other companies. Thus, all the facts representing the location of trucks are private, while the facts representing whether a package is at a logistic center are public. Only the load/unload actions at the logistic centers are public, while the move actions are private for each agent, as well as loading and unloading packages at private locations.

2.2 Landmark Heuristics

Landmarks, as used in LAMA, are propositional formulas that must be hold at some point during the execution of every successful plan (Richter, Helmert, and Westphal, 2008). A landmark heuristic function evaluates a state by considering the number of landmarks that still needs to be achieved. A popular method for identifying landmarks searches backwards from the goal. At each phase, one landmark is selected for development. All actions that can satisfy this landmark are identified, and a new landmark is constructed from their preconditions. When all these actions share a single fact, that fact is identified as a new landmark. Otherwise, a disjunctive formula is created from some facts that appear in the actions preconditions. When developing a landmark, all actions that take facts in this landmark as preconditions are ignored, in order to avoid circular reasoning.

2.3 The Fast Forward (FF) Heuristic

The FF heuristic begins by computing the relaxed planning graph, where facts are organized into layers, and all edges are between consecutive layers (Hoffmann, 2001). The first layer contains all the facts that hold at the current search state. To construct the next layer the FF heuristic considers the actions that can be executed at the current state, i.e., the actions whose preconditions are satisfied by the facts in the current layer. The next layer in the relaxed planning graph consists all the facts in the previous layer and all the facts that are effects of these set of actions. Importantly, delete effects, i.e., effects that remove facts from the state, are ignored. A result of ignoring delete effects is that once an action has participated in the construction of a layer, there is no need to consider it in the construction of future layers. For each fact p, the FF heuristic maintains the action a_p that has achieved it for the first time. Then, it constructs an edge between the preconditions of a_p on the previous level and p. After the graph construction, FF computes a plan over the relaxed graph, starting from the goal and moving backwards: first considering the goal facts and adding all the actions that achieved them to the plan. Then considering the preconditions of these actions, and so forth, ignoring repeated facts.

2.4 GPPP

The GPPP algorithm (Maliah, Shani, and Stern, 2016) is a CPPP algorithm based on heuristic forward search. The first phase in GPPP is the high-level planning phase, in which the agents collaboratively perform a best-first search on a relaxed version of the CPPP problem in which only public actions are used. To maintain privacy, states in this search are represented by the set of public facts that hold for that state, and a set of private state identifiers, one for each agent. Every agent can map its private identifiers to a set of private facts. When a state s is expanded, each agent generates new states by applying the public actions it can execute in s. In addition to the effects of the executed public action, the generated state also includes all private facts that the corresponding agent can achieve by applying private actions and using delete relaxation (Hoffmann, 2001). The high-level planning phase ends when a high-level public plan has been found that enables achieving all goals. Then, all agents compute private plans to achieve the preconditions of the public actions in the high level public plan. If some agent cannot achieve the preconditions of one of its actions in the high-level plan then this second phase fails then the high level planning phase resumes.

Maliah, Shani, and Stern (2014) show that GPPP's efficiency depends on the heuristic function used to guide the high-level search and that effective heuristics can be computed by a collaborative effort of the agents. For example, collaboratively computing a landmark heuristic requires each agent to reports the number of landmarks satisfied in a state. Using such heuristic allowed GPPP to show impressive performance, but the need to collaborate for evaluating the heuristic of each expanded state slows down the search process considerably. This is especially problematic for the FF heuristic, which requires agents to collaboratively find a relaxed plan. Our PP-LAMA algorithm builds upon GPPP and is especially suited to address this limitation by borrowing from LAMA the preferred operators and deferred heuristic evaluation techniques (Section 3.3).

Algorithm 1: The PP-LAMA algorithm

PP-LAMA()
  Init all open lists to hold the initial state
  $c_{LM} \leftarrow 0$; $c_{FF} \leftarrow 0$; $c_{pLM} \leftarrow 0$; $c_{pFF} \leftarrow 0$;
  while some open list is not empty do
    active-list$\leftarrow$ ChooseList(c_{LM},c_{FF},c_{PLM},c_{pFF})
    $s \leftarrow$ best state in active-list
    Remove s from all open lists
    if $s \models G$ then
      $P_{pub} \leftarrow$ the public plan to achieve s
      $P_{full} \leftarrow$ private extensions for P_{pub}
      if P_{full} is valid then
        return P_{full}
    Find achievable private facts in s
    Compute all heuristic values for s
    if s was generated by a PO and h(s) is the lowest so far for either FF or LM then
      $c_{pLM} \leftarrow c_{pLM} + 1000$; $c_{pFF} \leftarrow c_{pFF} + 1000$
    else
      $c_{LM} \leftarrow c_{LM} + 1$; $c_{FF} \leftarrow c_{FF} + 1$
    foreach public action a_{pub} applicable in s do
      $s' \leftarrow apply(s, a_{pub})$
      if a_{pub} is a preferred operator for s then
        Add s' into all open lists w. $h(s)$
      else
        Add s' into FF and LM open lists w. $h(s)$

3 Privacy Preserving LAMA

Next, we present PP-LAMA, starting by describing its different components.

3.1 Privacy Preserving Landmark Detection

We use the landmark identification process of Maliah, Shani, and Stern (2014), where agents collaborate in identifying landmarks, and augment it with an improved detection of private landmarks. The process begins by adding each fact (or disjunction of facts) in the goal as a landmarks. Then, the agents agree on a landmark and develop it, potentially adding more landmarks. This process continues until there are no more landmarks to develop. Developing a landmark ϕ means checking which actions can achieve ϕ, and then considering the preconditions of these actions as additional landmarks. To identify which actions achieve ϕ, the agents first check which facts they can achieve without requiring any facts in ϕ. This is done effectively by ignoring the delete effects of agents' actions (i.e., using a delete-relaxation), applying them iteratively (starting from the initial state) and sharing the achieved public facts between the agents. Then, each agent considers which actions it can perform to achieve facts in ϕ given this set of achieved facts. The preconditions of these actions form a new landmark.

A landmark ϕ can be a disjunction of public and private facts: $\phi = \text{public}(\phi) \vee \phi_1, \ldots, \phi_k$, where public($\phi$) is the public facts in ϕ and ϕ_i are the private facts of agent $i \in [1, \ldots, k]$ in ϕ. Each agent develops its private facts in ϕ and all agents develop together the public facts. Formally, if $d(\phi)$ denoted developing a landmark ϕ then, $d(\phi) = d(\text{public}(\phi)) \vee d(\phi_1) \vee \ldots d(\phi_k)$. To preserve privacy, only the public facts are published to the other agents. For private facts, all agents agree on a unique ID for this landmark, and each agent maintains a mapping from this ID to its own set of private facts that participate in this landmark. Our landmark identification process improves on that of Maliah, Shani, and Stern (2014) in that it allows detecting and developing landmarks that contain private facts of multiple agents. Experimentally, this improvement allows finding more than twice as many landmarks in some domains from the CoDMAP centralized planning track Štolba, Komenda, and Kovacs (2015).

The detected landmarks are used to evaluate a state during planning. Each agent publishes which landmarks it identified as satisfied in the state. The number of landmarks that at least one agent can satisfy is used as the landmarks heuristic estimate for that state. We refer to this heuristic as LM.

3.2 Privacy Preserving FF

We use the privacy preserving FF heuristic computation suggested by Štolba, Fišer, and Komenda (2015). The method constructs the relaxed planning graph jointly, with each agent maintaining a part of the graph. In every level of the relaxed planning graph, each agent computes its own set of achievable facts. Then, the public facts in that level are published to the other agents which then inserts these facts into their local current layer. As in regular FF, when a public fact p is achieved for the first time we maintain the action a_p that achieved it, and draw an edge between the preconditions of a_p on the previous layer and p. If a public fact has been achieved for the first time by several agents at the same level, one agent is considered to be responsible for achieving it, and the fact is labeled by the public action that this agent has used to achieve the fact.

The construction of the relaxed plan is also done collaboratively, where each agent computes the part of the plan that contains its own actions, starting from the goal and moving backwards. If an action in the plan requires a precondition fact that was generated by another agent then that agent is notified to continue constructing the plan to achieve that precondition. When the plan construction phase has terminated, all agents report the number of actions in their portion of the relaxed plan. The sum of these counts is the FF heuristic estimate for that state.

3.3 Preferred Operators

A preferred operator (PO) is an action that is deemed to be valuable at a given state. These are computed differently for the FF heuristic and for the LM heuristic. For FF, the POs are actions that achieve at least one precondition of an action in the relaxed plan. For LM, the POs are actions that appear prior to the first landmark that the relaxed plan achieves.

LAMA emphasizes POs in domains where they are useful as follows. When a state is generated by a PO, it is added to the regular open list and to an additional open list that contains only states generated by POs. We will refer to the latter open list as the preferred list. The search alternates between the regular open list and the preferred list, thus giving priority to states in the preferred list. Moreover, the priority of the preferred lists is boosted whenever a state generated by a PO has the best heuristic value so far when it is expanded. This boosting is embodied by having the next 1000 states to be expanded only from the preferred lists. If several heuristics are used then each heuristic has an open list and a preferred list. Thus, in LAMA as well as our PP-LAMA, we used four priority queues, two for FF and two for LM. In PP-LAMA, we followed this exact use of POs, except that only public actions are considered: a PO for FF is a public action that achieves a precondition of a public action in the relaxed plan, and similarly a PO for LM is any public action that is on the relaxed plan for achieving the first landmark.

The PP-LAMA algorithm is listed in Algorithm 1. The best-first search chooses which open list to use at each step according to their priorities (given by the counters c_{LM},c_{FF},c_{pLM}, and c_{pFF}), alternating between open lists with equal priority (line 5). As in LAMA, boosting the preferred lists is done whenever a state generated by a PO has a heuristic value that is the lowest seen so far (line 16). As in GPPP, when a public plan to the goal is found all agents try to extend it to a full plan 10.

Deferred State Evaluation. An important aspect of LAMA that is incorporated in PP-LAMA, and for the first time in a CPPP algorithm, is the deferred heuristic evaluation technique. When a state s' is inserted into the open lists, its associated heuristic value is that of its parent (lines 22 and 24). Computing the heuristic functions for s' is deferred until it is expanded (line 14). This deferred heuristic evaluation technique takes advantage of the prioritization obtain by the preferred lists, designed to reduce the number of costly heuristic computation. This is especially useful for improving GPPP, where the heuristic computation requires costly collaboration. Another process that is time consuming during state generation in GPPP is the computation of all achieveable private facts, which is done via delete relaxation. To resolve this, we also defer this process to the time when the state is expanded (line 13).

4 Empirical Evaluation

We evaluate the performance of PP-LAMA over all benchmarks, comparing PP-LAMA to all relevant algorithms from the centralized track of the CoDMAP competition Štolba, Komenda, and Kovacs (2015). All experiments were run on a 2.67 GHz machine with 32GB of memory. Table 1(left) shows how many problem instances each algorithm solved under a 30 minutes timeout (AKA "coverage"). In every domain the best performing algorithm is marked in bold. PP-LAMA solves 16 problems more than its best competitor and in all but two domains it is the best performing planner.

Next, we analyzed the impact of PP-LAMA's different components on planning performance. Table 1(right) presents coverage and runtime results (over problems solved by all variants) when using only the FF, only LM, FF with POs and deferred heuristics (denoted FF+PO+DH), both the FF and the land-

Domain	GPPP	MAPR-p	PMR	MAPlan/ FF+DTG	PSM-VRD	PP-LAMA
blocksworld	12	20	20	20	20	20
depot	11	0	0	13	17	18
driverlog	14	20	19	17	20	20
elevators	20	19	19	11	12	20
logistics	20	19	0	18	18	20
rovers	20	19	20	20	12	20
satellites	20	20	19	20	18	20
sokoban	9	0	6	18	18	12
taxi	20	0	19	20	0	20
wireless	3	2	7	4	0	4
woodworking	18	0	0	16	19	19
zenotravel	20	20	18	20	13	20
total	187	139	147	197	167	213

	FF		LM		FF+PO+DH		FF+LM		PP-LAMA	
Domain	C	T	C	T	C	T	C	T	C	T
blocksworld	18	49.5	12	35.5	18	19.2	20	49.4	20	12
depot	3	25	11	18.6	9	16.6	10	22.5	18	0.6
driverlog	14	155	14	203.4	20	2.8	17	26.8	20	2.7
elevators	20	148	20	25	20	4.4	20	149	20	4.2
logistics	20	7.8	20	2.5	20	0.8	20	7.9	20	0.8
rovers	14	217.2	20	156.7	20	0.8	19	222.9	20	0.8
satellites	16	266.2	20	89.5	20	2.5	17	260.1	20	2.3
sokoban	9	23.3	9	82.5	10	76	11	24.9	12	27
taxi	20	3.4	20	4.4	20	1.1	20	3.1	20	1
wireless	2	0.7	3	0.4	2	0.4	3	0.7	4	0.4
woodworking	5	2.3	18	0.5	16	1.2	15	1.3	19	0.4
zenotravel	20	165.8	20	65.5	20	4.1	20	164.8	20	2.1
total	161		187		195		192		213	

Table 1: (left) coverage results over the CoDMAP instances. (right) Impact of the various PP-LAMA components.

mark heuristics using alternating lists without deferred heuristics (FF+LM), and the complete PP-LAMA. The computation of the POs for LM requires the FF relaxed planning graph, and was hence not evaluated separately. Without POs, FF produces the poorest results, significantly worse than LM. This is because landmarks are faster to compute, as they are identified only once, and then the agents only need to evaluate which landmarks are satisfied in the current state, while FF needs to compute a relaxed plan in every state. The combination of both landmarks and FF is somewhat better than each heuristic alone, but the largest gain comes from applying PO and deferred heuristic evaluation. PP-LAMA (FF+LM+PO+DH) is very fast even though it uses the FF heuristic, because the heuristics are computed only for states that are removed from the open list, and not for all generated states.

5 Conclusion

We propose an adaptation of the renowned LAMA classical planner to CPPP. The resulting algorithm, called PP-LAMA, is built on GPPP and includes privacy preserving versions of the FF heuristic, landmark heuristic, preferred operators and deferred heuristic evaluation. PP-LAMA outperforms all relevant planners from the CoDMAP competition. In the future, we would extend our approach with additional heuristics, as was done for the classical LAMA.

Acknowledgments: We thank the reviewers for their useful comments. This work was supported by ISF Grant 933/13, by the Helmsley Charitable Trust through the Agricultural, Biological and Cognitive Robotics Center of Ben-Gurion University of the Negev.

We consider a multi-level jury problem in which experts We consider a multi-level jury problem in which experts

References

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.

Hoffmann, J. 2001. FF: The fast-forward planning system. AI magazine 22(3):57.

Maliah, S.; Shani, G.; and Stern, R. 2014. Privacy preserving landmark detection. In the European Conference on Artificial Intelligence (ECAI), 597–602.

Maliah, S.; Shani, G.; and Stern, R. 2015. Privacy preserving pattern databases. In ICAPS workshop on Distributed and Multi-Agent Planning (DMAP).

Maliah, S.; Shani, G.; and Stern, R. 2016. Collaborative privacy preserving multi-agent planning. Autonomous Agents and Multi-Agent Systems 1–38.

Richter, S., and Helmert, M. 2009. Preferred operators and deferred evaluation in satisficing planning. In ICAPS, 273–280.

Richter, S., and Westphal, M. 2010. The LAMA planner: Guiding cost-based anytime planning with landmarks. Journal of Artificial Intelligence Research (JAIR) 39(1):127–177.

Richter, S.; Helmert, M.; and Westphal, M. 2008. Landmarks revisited. In AAAI, volume 8, 975–982.

Röger, G., and Helmert, M. 2010. The more, the merrier: Combining heuristic estimators for satisficing planning. In ICAPS, 246–249.

Štolba, M., and Komenda, A. 2014. Relaxation heuristics for multiagent planning. In International Conference on Automated Planning and Scheduling (ICAPS).

Štolba, M.; Fišer, D.; and Komenda, A. 2015. Admissible landmark heuristic for multi-agent planning. In International Conference on Automated Planning and Scheduling (ICAPS).

Štolba, M.; Komenda, A.; and Kovacs, D. L. 2015. Competition of distributed and multiagent planners (codmap). The International Planning Competition (WIPC-15) 24.

Torreno, A.; Sapena, O.; and Onaindia, E. 2015. Global heuristics for distributed cooperative multiagent planning. In International Conference on Automated Planning and Scheduling (ICAPS).

Tozicka, J.; Jakubuv, J.; and Komenda, A. 2015. On internally dependent public actions in multiagent planning. In ICAPS workshop on Distributed and Multi-Agent Planning (DMAP).