Advances and Challenges in Privacy Preserving Planning

Guy Shani
Software and Information Systems Engineering, Ben Gurion University
shanigu@bgu.ac.il

Abstract

Collaborative privacy-preserving planning (CPPP) is a multi-agent planning task in which agents need to achieve a common set of goals without revealing certain private information. CPPP has gained attention in recent years as an important sub area of multi agent planning, presenting new challenges to the planning community. In this paper we describe recent advancements, and outline open problems and future directions in this field.

We begin with describing different models of privacy, such as weak and strong privacy, agent privacy, and cardinality preserving privacy. We then discuss different solution approaches, focusing on the two prominent methods — joint creation of a global coordination scheme first, followed by independent planning to extend the global scheme with private actions; and collaborative local planning where agents communicate information concerning their planning process. In both cases a heuristic is needed to guide the search process. We describe several adaptations of well known classical planning heuristic to CPPP, focusing on the difficulties in computing the heuristic without disclosing private information.

1 Introduction

Designing autonomous agents that act collaboratively is an important goal. A fundamental requirement of such collaboration is to plan for multiple agents acting to achieve a common set of goals. *Collaborative privacy-preserving planning (CPPP)* is a multi-agent planning task in which agents need to achieve a common set of goals without revealing certain private information [Brafman and Domshlak, 2008]. In particular, in CPPP an individual agent may have a set of private facts and actions that it does not share with the other agents. CPPP has important motivating examples, such as planning for organizations that outsource some of their tasks.

Figure 1 illustrates a logistics example of a CPPP problem in which the agents are trucks tasked with delivering packages. Trucks collaborate by loading and unloading packages in agreed logistics centers (marked by rectangles). Each truck has also a set of locations where only it can load and unload from (marked by circles). Whether a package is in a logistics center is of interest to multiple agents, and is thus public. All other information is private, such as the location of the individual trucks, and the location of the packages when they are not in a logistics center.

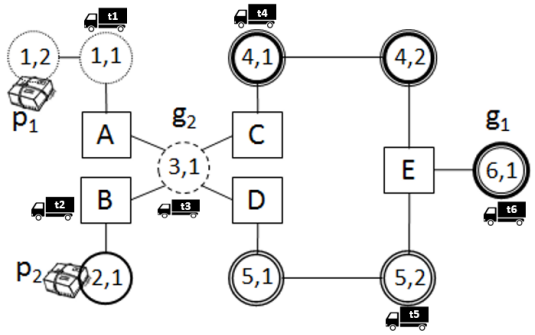

Figure 1: A logistics example, where trucks deliver packages between logistics centers, denoted by squares. Each agent i controls truck t_i and covers a set of cities. The cities are denoted by circles, each labeled by a pair i, j where i is the agent that covers this city and j is the local city index. Logistics centers can be entered by several agents, serving as collaboration sites.

There are two main approaches for solving CPPP [Torreño *et al.*, 2017]. First, the agents can jointly create a high level plan, composed only of public actions. Then the individual agents extend this high level scheme to a fully detailed multi-agent plan [Brafman and Domshlak, 2008; Brafman and Domshlak, 2013; Tozicka *et al.*, 2014; Jakubuv *et al.*, 2015]. Alternatively, each agent can run a distributed forward search to find a multi-agent plan directly, informing other agents of their progress [Nissim and Brafman, 2014]. In both cases, a heuristic may be needed to guide the search process [Maliah *et al.*, 2017; Štolba and Komenda, 2017]. Indeed, research has suggested adaptations of several classical planning heuristics into the CPPP, focusing mainly on avoiding the disclosure of private information through the heuristic computation.

In this paper we review research on CPPP, focusing on advancements, as well as on open challenges and interesting future directions. We discuss different models of privacy, examine important algorithmic advancements, and explain current

heuristic computations.

2 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 primitive propositions (facts).
- A_i is the set of actions agent i can perform.
- I is the start state.
- G is the goal condition.

Each action $a = \langle pre(a), \mathit{eff}(a)\rangle$ is defined by its preconditions ($pre(a)$), and effects ($\mathit{eff}(a)$). Preconditions and effects are conjunctions of primitive propositions and literals, respectively. A state is a truth assignment over P. G is a conjunction of facts. $a(s)$ denotes the result of applying action a to state s. A *plan* $\pi = (a_1, \ldots, a_k)$ is a solution to a planning task iff $G \subseteq a_k(\ldots(a_1(I)\ldots)$.

Privacy-preserving MA-STRIPS extends MA-STRIPS by defining sets of variables and actions as private, known only to a single agent. $private_i(P)$ and $private_i(A_i)$ denote the variables and actions, respectively, that are private to agent i. $public(P)$ is the set of public facts in P. $public_i(A_i)$, the complement of $private_i(A_i)$ w.r.t. A_i, is the set of public actions of agent i. Some preconditions and effects of public actions may be private, and the action obtained by removing these private elements is called its *public projection*, and it is known to other agents. When a public action is executed, all agents are aware of the execution, and view the public effects of the action. The goals can be public, but can also be private to a single agent, posing an additional challenge to the planing process.

An agent is aware only of its *local view* of the problem, that is, its private actions and facts, its public actions, the public facts, and the public projection of the actions of all other agents. That is, for public actions of other agents, the agent's local view contains only the public preconditions and effects of these actions.

In the logistics example in Figure 1, the set of facts P represents the location of two packages and six trucks. Each truck has three actions: move, load, and unload, corresponding to moving between locations, loading a package and unloading it. Trucks can only drive along the edges in Figure 1. Agents are heterogeneous and their range is restricted, such that location i, j can only be reached using truck i. The rectangles are logistic centers visited by multiple trucks that load or unload packages.

Trucks are owned by different companies that do not want to share their locations and coverage (the locations that a truck 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, whereas the move actions are private for each agent, as well as loading and unloading at private locations.

One can consider an alternative to the public-private dichotomy, where facts are shared only among a subset of agents. For example, in Figure 1, whether package p_1 is at E is private to the subset of agents $\{4, 5, 6\}$. These facts are called *subset-private*, as they are private to a subset of agents, with public facts being a special case, where the subset contains all agents.

The above definitions can perhaps be simplified. Specifically, it is unclear whether one truly needs to define both public actions and public facts. It may well be that all actions can be private, some facts can be public, and agents need only be aware when the value of public fact changes. Such a model can be more compact, but it is unclear whether it has the same representational power as the current privacy preserving MA-STRIPS model.

It is also interesting to explore extensions of this basic problem definition. Most specifically, the above definitions assume that each agent operates in a classical setting. It is interesting to explore other possibilities, such as nondeterministic action effects, or partial observability. In both cases, the agent's local plans cannot be represented as a sequential plan, but rather as plan trees or graphs. In some cases the global public plan must also be represented as a plan graph. Under these settings, new algorithms must be developed or adapted from the relevant planning areas.

3 Privacy Models

An algorithm is privacy-preserving, if it provably does not "reveal private information". Much of the research on privacy-preserving planning considered revealing private information only if the information is explicitly communicated to another agent. For example, if an agent publishes during planning that it intends to bring a truck t to a private location loc, then clearly the agent has revealed the existence of this private location, as well as an ability to achieve the private fact *(at t loc)*, breaking the privacy constraint. However, if the agent only publishes that it can achieve a private fact with the obfuscated name p, then it is unclear what private information has been revealed. Thus, some privacy-preserving MA-STRIPS planners are built on *obfuscating* the private information they publish by applying some cryptographic tool [Luis and Borrajo, 2014; Borrajo and Fernandez, 2015].

Brafman [2015] shows that the above form of privacy is weak, in the sense that there is no well formed constraint on what other agents can *infer* from the information available to them. For example, if the public plan consists of agent i picking up a package and the pickup action requires a truck to be present at the location of the package, then all agents now know that i controls at least one truck. Brafman considered a stronger form of privacy, where a fact or a specific value of a fact is *strongly private* if other agents cannot deduce its existence from the available information.

By "deducing the existence" of a private fact, we mean that regardless of the reasoning power of the agents, they cannot infer the existence of a strongly private fact from the available information. The information available to an agent is (1) its local view of the problem, (2) the messages between the agents during planning, and (3) the sequence of public actions (of all agents) in the resulting plan. A multi-agent planning algorithm is said to be *strongly privacy preserving* if the only

information that agents can deduce, following the execution of the algorithm, is information that is implied by the public projection, their local view, and the public part of the solution [Brafman, 2015].

While appealing, achieving such a strong form of privacy may be difficult. In fact, only two algorithms proven so far have this strict form of privacy: Secure MAFS [Brafman, 2015] and two special variants of PSM [Tozicka *et al.*, 2017]. Even these algorithms preserve this form of privacy only under restrictive conditions. For example, secure MAFS was shown to preserves strong privacy in a short list of specific domains (logistics, satellites, rovers), having unit action cost and when the heuristic function ignores the private state.

Thus, using more informed heuristics that we later discuss, may violate the strong privacy of Secure MAFS. The current planners that preserve strong privacy are either inefficient or incomplete [Tozicka *et al.*, 2017; Stolba *et al.*, 2016b]. Inefficiency in this context is that for any MA-STRIPS problem, all public solutions to all local views must be computed before a solution is returned. Tozicka et al. [2017] suggested to consider cases where an algorithm maintains strong privacy for a class of problems.

Researchers have offered other definitions of privacy aside from the two extreme cases of weak and strong privacy. Maliah et al. [2016a] suggest that an agent should only be aware of *neighboring* agents that modify a subset-private fact that the agent uses as precondition or effect. An algorithm preserves *agent privacy* if no agent can infer the existence of another agent with which it does not share subset-private facts. For example, in our running example, agent 6 should be unaware of the existence of agents $1, 2, 3$.

Alternatively, Maliah et al. [2016c] suggest that agents may be aware of the types of objects that other agents manipulate, but not of their cardinality. In the logistics example above, even though all agents may be aware that trucks carry packages between cities, they should not be aware of the number of trucks, or the number of cities that trucks travel between. An algorithms preserves *cardinality privacy* if no agent can infer the number of private objects controlled by another agent. In our running example, no agent should know whether 5 covers 2 or 3 cities.

Finally, instead of designing binary privacy criteria, one can consider more refined privacy metrics, that quantify the amount of leaked information [Štolba *et al.*, 2018]. For example, in distributed search [Maheswaran *et al.*, 2006] one often quantifies information loss using the entropy over the possible state space. In that case, it may be possible for an application designer to sacrifice some privacy in favor of efficiency, such as the ability to scale up to larger problems.

4 Algorithmic Approaches

There are two major approaches to planning in CPPP [Torreño *et al.*, 2017]. The first approach begins by computing a public plan, which is known as a coordination scheme [Nissim *et al.*, 2010; Brafman and Domshlak, 2013; Torreno *et al.*, 2014]. Then, the agents independently extend the public plan into a complete plan by adding private actions. In this extension each agent attempts to achieve the preconditions of its own public actions in the public plan.

For example, in the GPPP planner [Maliah *et al.*, 2017], agents search for a coordination scheme jointly over a relaxed planning problem. The agents decide together on a public state to expand. Then, each agent reports to other agents the set of public states that it can reach from the expanding state. To reduce the computational burden, the agents report the states they can achieve in a delete relaxation of the problem. Thus, it may well be that when finding a plan, this plan cannot be extended to a complete valid plan, and the agents must backtrack.

In the DPP planner [Maliah *et al.*, 2016c], on the other hand, the agents compute together a single agent projection of the CPPP problem that captures the dependencies between public actions. That is, which public actions facilitate the execution of other public actions of that agent. In our running example, such a dependency exists between loading a package at C and unloading it at E. These dependencies are computed using limited regression from the precondition of a public action to the effects of other public actions. Given this projection, one can compute a high level plan using a standard classical planner. The projection is incomplete, and it is hence possible that the generated public plan cannot be extended to a complete plan, in which case DPP fails.

An alternative approach to computing a high level scheme is to compute a complete plan directly. This can be done by each agent running a distributed forward search algorithm over its own action space, informing other agents of advancements in the search process.

The first algorithm in this family is MAFS [Nissim and Brafman, 2014] — a distributed algorithm in which every agent runs a best-first forward search to reach the goal. Each agent maintains an open list of states, and in every iteration each agent chooses a state in the open list to expand, generating all its children and adding them to the open list (avoiding duplicates). Whenever an agent expands a state that was generated by applying a public action, it also broadcasts this state to all other agents. An agent that receives a state adds it to the open list. For example, in Figure 1, when agent 1 unloads p_1 at A, it broadcasts this state to all other agents. Agent 3 can now use this state to load p_1 and transport it to logistic centers B, C, and D. To preserve privacy, the private part of a state is obfuscated when broadcasting it, e.g., by replacing the private facts with some index, such that only the broadcasting agent knows how to map this index to the corresponding private facts. Once the goal is reached, the agent achieving the goal informs all others, and the search process stops.

MAFS can be extended in several ways. The MADLA planner [Štolba and Komenda, 2017] augments MAFS with two different open lists, one ordered by a local heuristic, while the other ordered by a global heuristic. Maliah et al. [2016b] compute macros — sequences of private actions bounded by public actions — to expedite the local search process of the agent. For example, in Figure 1, once agent 5 has found the sequence of actions allowing it to transfer package p_1 from D to E, it can save this sequence as a macro, allowing agent 5 to apply this macro in all future explored states where the package is at D, expediting the search process.

The Forward-Backward planner [Maliah *et al.*, 2016a] at-

tempts to send newly generated states only to agents that can apply an action using the newly achieved facts. For example, when unloading p_1 at E, agent 4 can report this only to agents 5 and 6. For completeness, though, states must also be sent to other neighbors, but never to agents that do not share subset private facts. This allows MAFS to achieve agent privacy.

The efficiency of CPPP planners can be evaluated through basic operations, such as the amount of expansions, or wall clock time, but it is also important to evaluate the amount of required communication throughout the planning process, and prefer algorithms that send less information.

When evaluating plan quality, we can compute the plan's makespan, assuming maximal parallel execution. For example, in Figure 1, agents 1 and 2 can transfer packages p_1 and p_2 concurrently. When computing makespan, agents execute their private plans concurrently, and public action induce necessary synchronization points. As CPPP is collaborative, one can also compute the total amount of resources consumed during plan execution as another measure of plan quality. Other properties, such as fairness in the distribution of effort among agents, may also apply in some problems.

5 Computing Heuristics

Heuristic search is the main technique for both the joint public plan scheme, as well as the distributed individual plans. Hence, the computation of a useful heuristic is clearly an important consideration in CPPP research. As in classical planning, the heuristic functions must be both informative and easy to compute, but in CPPP their computation must also preserve the same level of privacy as the planning process.

The adaptation of many classical planning heuristics to CPPP has already been studied. The fast-forward (FF) heuristic computes a planning delete relaxation graph. This can be done in CPPP as well, where the agents jointly construct a planning graph [Torreno *et al.*, 2015; Štolba and Komenda, 2017]. At each iteration agents develop an internal relaxed planning graph, starting from the public facts at the last layer. Then, the agent report the newly achieved public facts, which become the next layer of the global graph. This construction requires much collaboration and many messages for each heuristic computation.

Other classical heuristics use a preprocessing phase to reduce the effort in computing heuristic values during planning. For example, landmarks — facts that must be achieved in every possible solution — are computed prior to planning using a regression process. Then, one can estimate the heuristic value of a state based on the amount of landmarks that need to be achieved [Richter *et al.*, 2008]. In CPPP, landmarks can also be computed in a joint preprocessing phase [Maliah *et al.*, 2017]. In GPPP, agents decide on a landmark to develop together. A landmark may be public, or private to an agent, in which case other agents are only aware of its identifier.

Then, each agent can use regression to discover facts that are required to achieve this landmark, which become new landmarks, and are published to all agents. This process continues until no new landmarks are discovered. For example, in Figure 1, package p_1 must arrive at g_1, which is a private location for agent 6. This is a private landmark for agent 6. Developing this landmark agent 6 reports a newly found public landmark, that p_1 must arrive at logistic center E. Now, developing this landmark, agents 4 and 5 discover a disjoint landmark, that the package must be either at $(5, 2)$ or at $(4, 2)$. As these are both private landmarks, they report just the indexes of the discovered landmark facts, and other agents know only of the existence of this landmark.

Pattern databases (PDBs) are a second popular classical planning heuristic that trades off preprocessing time to planning time [Edelkamp, 2001; Pommerening *et al.*, 2013]. The PDB contains a heuristic estimation for a state, based on precomputed solutions to a set of relaxed problems. PDBs can also be computed for CPPP, where agents report costs of producing public facts [Maliah *et al.*, 2015]. For example, in Figure 1, the PDB may contain the cost of transferring each package by agent 5 between logistic centers D and E. The relaxed problems in CPPP can be the private space of each agent, and the PDB consists of the cost of achieving one public fact given another public fact.

Instead of computing heuristics directly over the original problem, one can create a projection of CPPP to a classical single agent problem, and compute heuristics over the projection. On such simple projection is the local view of the agent, consisting of its own private actions and facts, and the public projections of actions of all other agents. However, this projection is extremely limited, ignoring essential information. For example, the public action of agent 5 for unloading p_1 at E has no public preconditions. In the view of other agents it seems that agent 5 can unload p_1 at E without any previous actions, making it useless to plan to bring p_1 to logistic center D. This results in a heuristic estimation which is a gross under estimation. A projection that maintains the dependency between bringing p_1 to D before it can be brought to E will be much more informative [Maliah *et al.*, 2016c; Tožička *et al.*, 2018]. Using such stronger projections one can compute much more useful heuristic estimations.

Finally, most of the heuristics above may not be admissible, and are hence inadequate for optimal planning. Some admissible heuristics such as LM-Cut and potential heuristics were adapted to CPPP [Štolba *et al.*, 2015; Stolba *et al.*, 2016a]. Still, optimal planning for CPPP has not yet received sufficient attention from the community.

6 Conclusions

In this paper we have reviewed privacy preserving multi agent planning (CPPP) — a problem setting that has gained much attention from the planning community in recent years. We discussed four different interesting topics in this area — the structure of planning problems in this setting, the definition of privacy, planning approaches and algorithms, and the computation of heuristic estimations.

For each topic we outlined the current trends, and discussed questions that are yet under investigated, with a significant potential for future investigation.

We believe that CPPP provides both a realistic problem setting, and a challenging ground for new development, and as such, hope to see this area continuing to bloom with new ideas and contributions.

Acknowledgments

Most of the work described in this paper is based on the research of my student, Shlomi Maliah, with the collaboration of Roni Stern and Ronen Brafman. Perhaps the most influential contribution to this field is the pioneering work of Raz Nissim who designed the MAFS algorithm. I would also like to especially note the contribution of Antonin Komenda, Michal Štolba, and Daniel Kovacs, who ran the CoDMAP competition — a critical milestone in the CPPP research.

This work is partially supported by ISF Grant 933/13, by the Cyber Security Research Center at Ben-Gurion University of the Negev, and by the Helmsley Charitable Trust through the Agricultural, Biological and Cognitive Robotics Center of Ben-Gurion University of the Negev.

References

[Borrajo and Fernandez, 2015] Daniel Borrajo and Susana Fernandez. MAPR and CMAP. In *CoDMAP-15*, 2015.

[Brafman and Domshlak, 2008] Ronen I Brafman and Carmel Domshlak. From one to many: Planning for loosely coupled multi-agent systems. In *ICAPS*, pages 28–35, 2008.

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

[Brafman, 2015] Ronen I. Brafman. A privacy preserving algorithm for multi-agent planning and search. In *IJCAI*, pages 1530–1536, 2015.

[Edelkamp, 2001] Stefan Edelkamp. Planning with pattern databases. In *ECP*, pages 13–34, 2001.

[Jakubuv *et al.*, 2015] Jan Jakubuv, Jan Tozicka, and Antonın Komenda. Multiagent planning by plan set intersection and plan verification. *ICAART*, 15, 2015.

[Luis and Borrajo, 2014] Nerea Luis and Daniel Borrajo. Plan merging by reuse for multi-agent planning. In *DMAP–ICAPS*, 2014.

[Maheswaran *et al.*, 2006] Rajiv T Maheswaran, Jonathan P Pearce, Emma Bowring, Pradeep Varakantham, and Milind Tambe. Privacy loss in distributed constraint reasoning: A quantitative framework for analysis and its applications. *JAAMAS*, 13(1):27–60, 2006.

[Maliah *et al.*, 2015] Shlomi Maliah, Guy Shani, and Roni Stern. Privacy preserving pattern databases. In *DMAP–ICAPS*, volume 15, pages 9–17, 2015.

[Maliah *et al.*, 2016a] Shlomi Maliah, Ronen I Brafman, and Guy Shani. Increased privacy with reduced communication and computation in multi-agent planning. In *DMAP–ICAPS*, volume 16, pages 106–112, 2016.

[Maliah *et al.*, 2016b] Shlomi Maliah, Guy Shani, and Ronen I Brafman. Online macro generation for privacy preserving planning. In *ICAPS*, pages 216–220, 2016.

[Maliah *et al.*, 2016c] Shlomi Maliah, Guy Shani, and Roni Stern. Stronger privacy preserving projections for multi-agent planning. In *ICAPS*, pages 221–229, 2016.

[Maliah *et al.*, 2017] Shlomi Maliah, Guy Shani, and Roni Stern. Collaborative privacy preserving multi-agent planning - planners and heuristics. *JAAMAS*, 31(3):493–530, 2017.

[Nissim and Brafman, 2014] Raz Nissim and Ronen I. Brafman. Distributed heuristic forward search for multi-agent planning. *JAIR*, 51:293–332, 2014.

[Nissim *et al.*, 2010] Raz Nissim, Ronen I Brafman, and Carmel Domshlak. A general, fully distributed multi-agent planning algorithm. In *AAMAS*, pages 1323–1330, 2010.

[Pommerening *et al.*, 2013] Florian Pommerening, Gabriele Röger, and Malte Helmert. Getting the most out of pattern databases for classical planning. In *IJCAI*, pages 2357–2364, 2013.

[Richter *et al.*, 2008] Silvia Richter, Malte Helmert, and Matthias Westphal. Landmarks revisited. In *AAAI*, volume 8, pages 975–982, 2008.

[Štolba and Komenda, 2017] Michal Štolba and Antonín Komenda. The madla planner: Multi-agent planning by combination of distributed and local heuristic search. *Artificial Intelligence*, 252:175–210, 2017.

[Štolba *et al.*, 2015] Michal Štolba, Daniel Fišer, and Antonín Komenda. Admissible landmark heuristic for multi-agent planning. In *ICAPS*, 2015.

[Stolba *et al.*, 2016a] Michal Stolba, Daniel Fiser, and Antonín Komenda. Potential heuristics for multi-agent planning. In *ICAPS*, pages 308–316, 2016.

[Stolba *et al.*, 2016b] Michal Stolba, Jan Tozicka, and Antonín Komenda. Secure multi-agent planning algorithms. In *ECAI*, pages 1714–1715, 2016.

[Štolba *et al.*, 2018] Michal Štolba, Jan Tožička, and Antonín Komenda. Quantifying privacy leakage in multi-agent planning. *TOIT*, 18(3):28, 2018.

[Torreño *et al.*, 2017] Alejandro Torreño, Eva Onaindia, Antonín Komenda, and Michal Štolba. Cooperative multi-agent planning: A survey. *ACM Comput. Surv.*, 50(6), 2017.

[Torreno *et al.*, 2014] Alejandro Torreno, Eva Onaindia, and Oscar Sapena. FMAP: Distributed cooperative multi-agent planning. *Applied Intelligence*, 41(2):606–626, 2014.

[Torreno *et al.*, 2015] Alejandro Torreno, Oscar Sapena, and Eva Onaindia. Global heuristics for distributed cooperative multi-agent planning. In *ICAPS*, pages 225–233, 2015.

[Tozicka *et al.*, 2014] Jan Tozicka, Jan Jakubuv, and Antonín Komenda. Generating multi-agent plans by distributed intersection of finite state machines. In *ECAI*, 2014.

[Tozicka *et al.*, 2017] Jan Tozicka, Michal Štolba, and Antonín Komenda. The limits of strong privacy preserving multi-agent planning. In *ICAPS*, 2017.

[Tožička *et al.*, 2018] Jan Tožička, Jan Jakubuv, and Antonín Komenda. Recursive reductions of action dependencies for coordination-based multiagent planning. In *TCCI 28*, pages 66–92. Springer, 2018.