Distributed and Multi-Agent Planning: Challenges and Open Issues

Andrea Bonisoli

Dipartimento di Ingegneria dell'Informazione,
Università degli Studi di Brescia,
Via Branze 38, I-25123 Brescia, Italy.
andrea.bonisoli@ing.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. In this work, we survey recent contributions in the fields of distributed and multi-agent planning. We define the problem, briefly outline possible different classifications of the multi-agent planning problem and present the state of the art in this field. Finally, we report open challenges and issues that may be addressed in the following years.

1 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 [1] 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. A first distinction should be made between systems where there is a single planner for all the executing agents and systems where more computational agents are autonomous, rational and have planning abilities. Usually the first are called *centralized* while the latter are called *distributed* or *decentralized*. 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.

The rest of this brief paper is structured as follows. In section 2, we introduce a formalization of the multi-agent planning problem and two possible languages to define it. In section 3 we survey the most important results in this field and describe the main contributions of a set of important recent papers on multi-agent planning. Finally, in section 4, we outline some open challenges and issues that could be addressed in future research. The study presented in this paper has been done in the context of an ongoing research for the PhD thesis of the author.

2 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 [2] and then adopted by several authors (e.g., in [3–6]). Other definition of the problem are also possible [7], nonetheless MA-STRIPS is a simple and effective language to represent the *cooperative* multi-agent planning. The distinction between described in the next section.

An MA-STRIPS problem for a group of agents $\Phi = \{\varphi_i\}_{i=1}^{n}$ is a quadruple $\Pi = \langle P, \{A_i\}_{i=1}^{n}, I, G\rangle$ where:

- P is a finite set of atoms called propositions;
- $I \subseteq P$ encodes the initial state of the system;
- $G \subseteq P$ defines a set of goals;
- A_i is the set of actions that can be performed by agent φ_i, each action has the same syntax as in STRIPS, namely is a triplet of subsets of P which captures the precontitions, additive effects and delete effects of that action.

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.

Another interesting proposal for a standard description language that allow for a more direct comparison between systems and approaches is MA-PDDL [8], 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, but, to the best of our knowledge, no planner uses it yet.

3 State of the Art

Most algorithms and systems from classical planning can be easily adapted to handle centralized multi-agent planning. In this case there is a single planner that can communicate plans to the executing agents. If applicable, a centralized planning approach can be quite efficient.

However, there are few problems that make decentralized or distributed planning more suitable to a wide range of settings. First of all, as noted in [3] and others, allowing concurrent, independent actions for every executing agent leads to exponential blow-up in the action space. This can make impossible to scale up if there is a large number of agents or it leads to poor performance of the centralized approaches [5]. Secondly, there are many settings where the executing agents already have a high degree of autonomy and can plan for themselves. In such settings, where agents can have privacy issues, it's improbable that a central authority can find a plan that every agent accepts.

Due to all different options, a broad categorization is used. The first distinction assesses whether the agents are *cooperative* or *opponent*. We define a pair of *opponent* agents when their goals are different and reaching a goal state for one agent prevents the possibility to reach a goal fact for the other agent. For such opponent agents, strategical analysis and game-theoretical approaches should be taken into account during the planning process. On the contrary, for cooperative agents there should exists at least one solution where every agent reaches his goals. For cooperative agents, a special case is when the set of goals is the same for every agents and the problem can be defined using the `MA-STRIPS` language.

In every cases, we assume that there is at least some level of agents interaction and loosely coupled systems have different properties from tightly coupled [2, 7]. Agents can be forced to cooperate because they cannot achieve their goals alone or because it is more convenient to do so. If agents are not forced, but still willing to cooperate in a joint plan only if given a sufficient reward, they are called *self-interested* or *selfish* agents. The presence of selfish agents requires that the solution joint-plan provides sufficient reward or rational incentives to every involved agents. Finally, communication is also an important factor in multi-agent systems. Communication between agents may be needed to coordinate the actions or to receive the plan from the central planner and agents can send and receive different types of messages. If every agent can communicate freely with every other agent, or if the communication is slow, with limited time or unreliable, is an important distinction that affects the algorithm design.

In case of distributed computation, a cooperative multi-agent planning requires some kind of communication and coordination between agents [1]. By exploiting such ideas, it is possible to find solutions using techniques such as plan merging or plan coordination [9, 10]. While these techniques are well known, they seem not to be well suited for settings where agents are not loosely coupled or there are many interactions between agents, either potential conflicts or cooperative opportunities. Furthermore, such approaches seem to be incapable of solving problems with non-cooperative agents.

To better scale up with the number of agents and being able to deal with non-cooperative agents, Jonsson and Rovatsos proposed an iterative plan refinement approach [3]. In this approach, standard off-the-shelf planning technology is used with a novel best-response planning method. Despite the absence of convergence or optimality guarantees, this approach can be useful to improve multi-agent plans.

For the case of optimal planning, in which the solution plans must involve the minimum number of actions, a distributed and parallel version of the algorithm A* can be used [5]. For deterministic distributed planning approaches, the solution of the planning problem can be found using a distributed constraint satisfaction problem [2, 6] or a state-of-the-art forward-chaining partial-order planning search process [7]. Further some approaches use merging of hierarchical task networks [11] or an adaptation of the heuristic planner Fast-forward [4].

To deal with partial observability or non-deterministic action effects, Markov decision processes and their generalization POMDP can be used [12]. Moreover, decentralized POMDP algorithms can also be used [13–15] for distributed planning, but the high complexity of Dec-POMDP models limits their applicability to small problems.

4 Challenges, Open Issues and Future Work

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* proposed in [2] 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.

Another interesting field of research is multi-agent plan repair or replanning. While such issues are well studied in classical planning, the presence of multiple agents makes some known techniques unsuitable, as described in [16]. Therefore, new approaches to multi-agent plan repair should be investigated, using experimental insights such as those in [17].

Acknowledgements This preliminary study has been conducted under the supervision of professor Alfonso Emilio Gerevini and the help of Alessandro Saetti and Ivan Serina. The author would like to thank to Anja Roubickova and the anonymous reviewers for their advices and help writing this paper.

References

1. de Weerdt, M., Clement, B.: Introduction to planning in multiagent systems. Multiagent and Grid Systems **5** (2009) 345–355
2. Brafman, R.I., Domshlak, C.: From one to many: Planning for loosely coupled multi-agent systems. In: Proceedings of the 18th International Conference on Automated Planning and Scheduling. ICAPS (2008) 28–35

3. Jonsson, A., Rovatsos, M.: Scaling up multiagent planning: A best-response approach. In: Proceedings of the 21st International Conference on Automated Planning and Scheduling. ICAPS (2011)
4. Štolba, M., Komenda, A.: Fast-forward heuristic for multiagent planning. In: Proceedings of the First Workshop on Distributed and Multi-Agent Planning. ICAPS (2013) 75–83
5. Nissim, R., Brafman, R.I.: Multi-agent A* for parallel and distributed systems. In: Proceedings of the Workshop on Heuristics and Search for Domain-Independent Planning. ICAPS (2012) 43–51
6. Brafman, R.I., Domshlak, C.: On the complexity of planning for agent teams and its implications for single agent planning. Artificial Intelligence **198** (2013) 52–71
7. Torreño, A., Onaindia, E., Sapena, O.: FMAP: a heuristic approach to cooperative multi-agent planning. In: Proceedings of the First Workshop on Distributed and Multi-Agent Planning. ICAPS (2013) 84–92
8. Kovacs, D.L.: A multi-agent extension of PDDL 3.1. In: Proceedings of the 3rd Workshop on the International Planning Competition. ICAPS (2012) 19–27
9. Cox, J.S., Durfee, E.H., Bartold, T.: A distributed framework for solving the multiagent plan coordination problem. In: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems. AAMAS '05, New York, NY, USA, ACM (2005) 821–827
10. Cox, J.S., Durfee, E.H.: An efficient algorithm for multiagent plan coordination. In: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems. AAMAS '05, New York, NY, USA, ACM (2005) 828–835
11. Milla-Millán, G., Fdez-Olivares, J., Sánchez-Garzón, I.: Multi-agent planning based on the dynamic selection and merging of hierarchical task networks. In: Proceedings of the Workshop on Distributed and Multi-Agent Planning. ICAPS (2013) 34–42
12. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. Artificial intelligence **101**(1) (1998) 99–134
13. Seuken, S., Zilberstein, S.: Improved memory-bounded dynamic programming for decentralized pomdps. In: Proceedings of the Twenty-Third Conference Annual Conference on Uncertainty in Artificial Intelligence, AUAI Press (2007) 344–351
14. Bernstein, D.S., Givan, R., Immerman, N., Zilberstein, S.: The complexity of decentralized control of markov decision processes. Mathematics of Operations Research **27**(4) (2002) 819–840
15. Brafman, R.I., Shani, G., Zilberstein, S.: Qualitative planning under partial observability in multi-agent domains. In: Proceedings of the First Workshop on Distributed and Multi-Agent Planning. ICAPS 26–33
16. Talamadupula, K., Smith, D., Cushing, W., Kambhampati, S.: A theory of intra-agent replanning. In: Proceedings of the First Workshop on Distributed and Multi-Agent Planning. ICAPS (2013) 48–56
17. Komenda, A., Novák, P., Pěchouček, M.: How to repair multi-agent plans: Experimental approach. In: Proceedings of the First Workshop on Distributed and Multi-Agent Planning. ICAPS (2013) 66–74