A Modular Framework for Decentralised Multi-Agent Planning

(Extended Abstract)

Rafael C. Cardoso, and Rafael H. Bordini
Faculdade de Informática, PUCRS
Porto Alegre – RS, Brazil
rafael.caue@acad.pucrs.br,rafael.bordini@pucrs.br

ABSTRACT

Multi-agent systems often require runtime planning, which remains an open problem due to the existing gap between planning and execution in practice. Extensive research has been carried out in centralised planning for single-agent systems, but so far decentralised multi-agent planning has not been fully explored. In this paper, we extend existing multi-agent platforms to enable decentralised planning at runtime. In particular, we put forward a planning and execution framework called Decentralised Online Multi-Agent Planning (DOMAP). Experiments with a planning domain we developed on flooding disaster scenarios show that DOMAP outperforms 4 other state-of-the-art multi-agent planners, particularly in the most difficult problems.

Keywords

multi-agent planning; multi-agent systems; distributed problem solving; online planning

1. INTRODUCTION

Research on automated planning has been mostly focused on centralised planning. Although it is possible to adapt single-agent techniques to work in a decentralised way, as in [6], distributed computation is not the only advantage of using Multi-Agent Planning (MAP). By allowing agents to do their own individual planning, the search space is effectively pruned, which can potentially decrease planning time on domains that are naturally distributed, and allow agents to keep some privacy within the system. MAP has received increasing attention recently [3, 8, 9, 19, 14], tackling new and complex multi-agent problems that require decentralised solutions. One such problem is to combine planning and execution [13].

Multi-Agent Systems (MAS) development platforms also changed the focus from agent- to organisation-centred approaches. Recent research in MAS, as evidenced in [1, 16], shows that considering other programming dimensions such as environments and organisations as first-class abstractions (along with agents) allow developers to create more complex MAS.

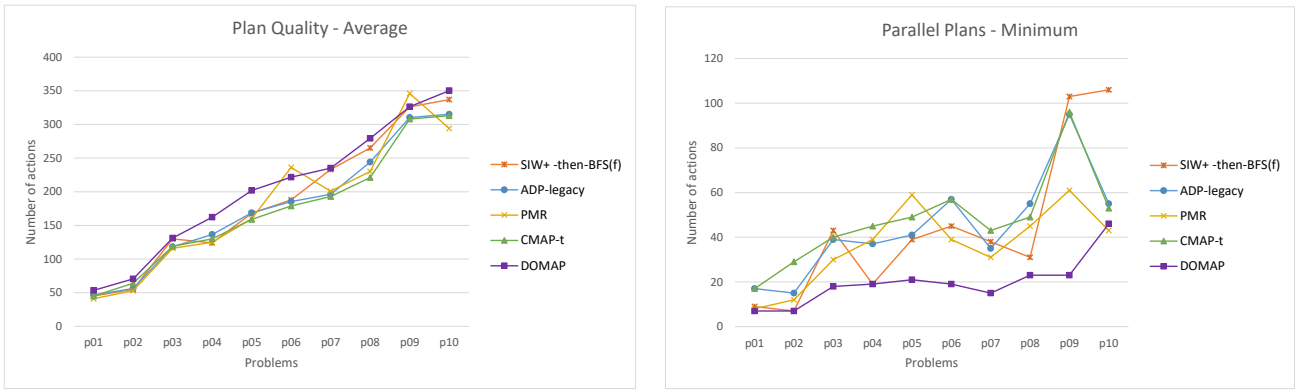
Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

2. PLANNING FRAMEWORK

DOMAP uses a factored representation to interface with a MAS in order to obtain knowledge necessary for planning. It serves as a bridge between planning and execution. DOMAP has three modular components: (*i*) goal allocation mechanism – agents use the contract net protocol [17] to pre-select goals that they believe to be more appropriate for them (see [4]); (*ii*) individual planner – the SHOP2 [12] planner is used by each agent for individual planning so as to make the most of the HTN-like structure of the plan library in typical BDI agents; (*iii*) coordination mechanism – social laws [15] are employed to coordinate agents at execution time, in order to avoid possible conflicts created during (individual) planning. Algorithm 1 gives an overview of how planning in DOMAP works. The solution is then executed and monitored so as to coordinate agents at runtime.

Algorithm 1 DOMAP overview.

```
1: function DOMAP(Social_Goals)
2: for each agent ∈ Agents do
3:     create own factored representation
4: end for
5: banned ← ∅
6: repeat
7:     announce(Social_Goals, banned(List))
8:     for each agent ∈ Agents do
9:         for each sg ∈ Social_Goals do
10:            if (agent, sg) ∉ banned(List) then
11:                agent calculates and places a bid for sg
12:            end if
13:        end for
14:    end for
15:    award(Social_Goals, bids)
16:    // goals_agent are SGs allocated to agent
17:    for each agent ∈ Agents do
18:        individual_planning(goals_agent)
19:        if planning_failed(goals_agent) then
20:            for each sg ∈ goals_agent do
21:                banned ← banned ∪ {(agent, sg)}
22:            end for
23:        end if
24:    end for
25: until banned ≠ ∅
```



(a) (b)

Figure 1: (a) Average plan cost across 10 runs; (b) Shortest parallel plan found across 10 runs.

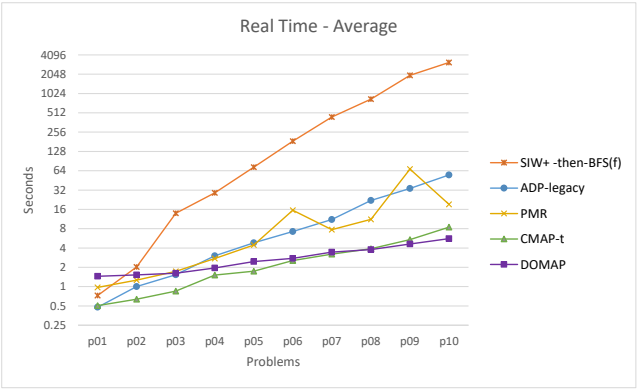


Figure 2: Average planning time across 10 runs.

3. EXPERIMENTS AND RESULTS

To evaluate our framework we implemented DOMAP in JaCaMo [1], a platform for the development of MAS. To the best of our knowledge, DOMAP is the only recently developed multi-agent online planner that is able to take advantage of the different programming abstractions in MAS (agent, environment, and organisation). Thus, in order to proper evaluate our framework, we isolated the online-planning components of DOMAP, and ran offline experiments using a novel multi-agent domain, the Floods domain, to compare DOMAP against four state-of-the-art planners that took part in the 2015 Competition of Distributed and Multi-Agent Planners (CoDMAP-15) [18].

One of our research projects aiming to address real-world problems related to (flooding) disasters led us to design a new domain with the essential characteristics of MAP and MAS, such as heterogeneous agents and multiple types of goals. In the Floods domain, a team of heterogeneous autonomous robots are dispatched to monitor flooding activity and execute search and rescue operations within a geographical region. We generated 10 problem variations, increasing the number of goals and agents for each problem. We gave all problems a time limit of 60 minutes. We ran each planner 10 times for each of the 10 problems, resulting in a total of 500 executions, 100 for each planner. The computer used to run the experiments has the following specification: Intel Xeon Processor E5645 (12M Cache, 2.40 GHz, 6 cores, 12 threads), 32 GB of memory, Ubuntu 16.04 operating system.

Average plan cost is shown in Figure 1a, and shortest parallel plans in Figure 1b. CMAP-t [2], ADP-legacy [7, 5], and PMR [10] found some of the lowest-cost plans. PMR does much worse on problems 6 and 9, when it has to use multiple planners, but it is able to find the lowest cost plan for the hardest problem (p10). DOMAP and SIW+ -then-BFS(f) [11] have mediocre average plan lengths when compared to other planners for the most difficult problems. Prioritising fairness among agents in goal distribution pays off when we consider plan parallelism, where DOMAP shows excellent parallel solutions for all problems.

In Figure 2, we show the results for planning time for each problem. Time in seconds is on the y-axis; it is in logarithmic scale to improve readability. Our results in this new domain show a larger gap between planners when compared to the results reported in CoDMAP-15. We suspect that this discrepancy occurred because the domains from the competition were too simplistic, with a low number of agents and goals, resulting in planning times low enough that the winner could have been any of the top performing planners. SIW+ -then-BFS(f), the winner of CoDMAP-15 w.r.t. planning time, performed the worst in our experiments (it does not separate goal allocation from planning).

Our results show DOMAP scales rather well, provides excellent parallel solutions, and it is the fastest multi-agent planner for the most difficult problems, while maintaining a reasonable plan length compared to other planners. Our results also indicate that allocating goals before planning can lead to significant improvement in planning time for some problems. Moreover, decentralising planning and running in computers with multiple cores can also lead to faster planning times (only DOMAP and PMR do so). The planners ADP-legacy, CMAP-T, and PMR all use relaxed planning graphs for goal allocation, which could be added to DOMAP to improve plan length. Planning times of decentralised planners could be potentially improved when running in distributed settings, as planned for future experiments.

Acknowledgments

We are grateful for the support given by CNPq and CAPES.

REFERENCES

[1] O. Boissier, R. H. Bordini, J. F. Hübner, A. Ricci, and A. Santi. Multi-agent oriented programming with JaCaMo. *Science of Computer Programming*, 2011.

[2] D. Borrajo and S. Fernandez. MAPR and CMAP. In *Competition of Distributed and Multi-Agent Planners (CoDMAP-15)*, 2015.

[3] M. Brenner and B. Nebel. Continual planning and acting in dynamic multiagent environments. *Autonomous Agents and Multi-Agent Systems*, 19(3):297–331, Dec. 2009.

[4] R. C. Cardoso and R. H. Bordini. Allocating social goals using the contract net protocol in online multi-agent planning. In *5th Brazilian Conference on Intelligent System (BRACIS-16)*, Recife, Pernambuco, Brazil, 2016.

[5] M. Crosby. Adp an agent decomposition planner codmap 2015. In *Competition of Distributed and Multi-Agent Planners (CoDMAP-15)*, 2015.

[6] M. Crosby, A. Jonsson, and M. Rovatsos. A single-agent approach to multiagent planning. In *21st European Conf. on Artificial Intelligence (ECAI'14)*, Prague, Czech Republic, 2014.

[7] M. Crosby, M. Rovatsos, and R. P. A. Petrick. Automated agent decomposition for classical planning. In *Proceedings of the Twenty-Third International Conference on Automated Planning and Scheduling, ICAPS*, Rome, Italy, 2013.

[8] M. de Weerdt and B. Clement. Introduction to Planning in Multiagent Systems. *Multiagent Grid Syst.*, 5(4):345–355, 2009.

[9] E. H. Durfee and S. Zilberstein. Multiagent planning, control, and execution. In G. Weiss, editor, *Multiagent Systems 2nd Edition*, chapter 11, pages 485–545. MIT Press, 2013.

[10] N. Luis and D. Borrajo. PMR: Plan merging by reuse. In *Competition of Distributed and Multi-Agent Planners (CoDMAP-15)*, 2015.

[11] C. Muise, N. Lipovetzky, and M. Ramirez. MAP-LAPKT: Omnipotent multi-agent planning via compilation to classical planning. In *Competition of Distributed and Multi-Agent Planners (CoDMAP-15)*, 2015.

[12] D. Nau, O. Ilghami, U. Kuter, J. W. Murdock, D. Wu, and F. Yaman. Shop2: An htn planning system. *Journal of Artificial Intelligence Research*, 20:379–404, 2003.

[13] D. S. Nau, M. Ghallab, and P. Traverso. Blended planning and acting: Preliminary approach, research challenges. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 4047–4051. AAAI Press, 2015.

[14] R. Nissim and R. I. Brafman. Distributed heuristic forward search for multi-agent planning. *J. Artif. Intell. Res. (JAIR)*, 51:293–332, 2014.

[15] Y. Shoham and M. Tennenholtz. On social laws for artificial agent societies: Off-line design. *Artif. Intell.*, 73(1-2):231–252, Feb. 1995.

[16] M. Singh and A. Chopra. Programming multiagent systems without programming agents. In L. Braubach, J.-P. Briot, and J. Thangarajah, editors, *Programming Multi-Agent Systems*, volume 5919 of *Lecture Notes in Computer Science*, pages 1–14. Springer Berlin Heidelberg, 2010.

[17] R. G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Trans. Comput.*, 29(12):1104–1113, Dec. 1980.

[18] M. Stolba, A. Komenda, and D. L. Kovacs, editors. *Competition of Distributed and Multi-Agent Planners (CoDMAP-15)*, Jerusalem, Israel, 2015.

[19] A. Torreño, E. Onaindia, and O. Sapena. FMAP: distributed cooperative multi-agent planning. *Appl. Intell.*, 41(2):606–626, 2014.