Introduction to Planning in Multiagent Systems

Mathijs de Weerdt[*], Delft University of Technology,
PO Box 5031, 2600 GA Delft, The Netherlands, and

Brad Clement[†], Jet propulsion Laboratory,
4800 Oak Grove Dr., Pasadena, CA 91750 USA

December 23, 2009

Abstract

In most multiagent systems planning on forehand can help to seriously improve the efficiency of executing actions. The main difference between centrally creating a plan and constructing a plan for a system of agents lies in the fact that in the latter coordination plays the main part. This introduces a number of additional difficulties. This special issue discusses some of these difficulties in detail. To place these in a context, this introduction gives a brief overview of multiagent planning problems, and most multiagent planning techniques.

1 Introduction

Agents can be classified into two categories according to the techniques they employ in their decision making: *reactive* agents (cf. [27]) base their next decision solely on their current sensory input, while *planning* agents, on the other hand, take into account anticipated future situations, possibly as a result of their own actions, to decide on the best course of action [33].

[*]Corresponding author: M.M.deWeerdt@tudelft.nl
[†]Corresponding author: bclement@jpl.nasa.gov

When an agent should plan and when it should be reactive depends on the particular situation it finds itself in. Consider the example where an agent has to plan a route from one place to another. A reactive agent might use a compass to plot its course, whereas a planning agent would consult a map. Clearly, the planning agent will come up with the shortest route in most cases, as it will not be confronted with uncrossable rivers and one-way streets. On the other hand, there are also situations where a reactive agent can be at least as effective, for instance if there are no maps to consult such as in a domain of (Mars) exploration rovers. Nevertheless, the ability to plan ahead is invaluable in many domains. Therefore, this special issue is dedicated to agents that are planning.

In particular the work presented here focuses on systems where a number of such planning agents interact. Such settings where multiple agents plan, often distributedly, introduce additional difficulties over the already hard problem of planning itself: there is the additional need for coordination, and because communication is often limited, the result is consequently less optimal. However, there are a number of good reasons for having multiple agents creating plans. First, the agents may represent real-life entities which mainly have their own interests at heart. Therefore, they appreciate maintaining their privacy and autonomy. Second, a distributed system may already exist, for which centralization would be too costly. Third, creating and maintaining plans locally allows for a more efficient reaction in case of incidents, especially when communication is limited. Finally, dividing the planning problem into smaller pieces and solving those in parallel may sometimes be more efficient, especially when the individual planning problems are loosely coupled.

The five contributions in this special issue expand on these motivations by studying some of the questions that arise when developing a multiagent planning approach.

1. How to place additional constraints upon the agents before planning such that their resulting plans can easily be coordinated?

2. How to efficiently construct plans in a distributed fashion?

3. How to make collaborative decisions when there are multiple options for which each agent has its own preferences?

4. When should a planning agent ask the user for more specific information?

5. How to measure how much privacy is lost in the process of coordinating plans?

This introduction gives some background on the multiagent planning problem, existing approaches to this problem, and it then places these five contributions in this context. Parts of this document are based on an earlier technical report [14].

2 Multiagent planning problems

There are many variants of what is understood as a multiagent planning problem. In general, a multiagent planning problem can be defined as the problem of planning by and for a group of agents. Except for more centralized (multiagent) planning problems, each agent in such a problem has in fact a private, individual planning problem. A typical individual planning problem of an agent includes a set of operations (with some costs attached, and a pre- and post-condition) that it can perform, a set of goals (with reward values), and the current (initial) state of this agent. The solution to a multiagent planning problem is a plan: a partially ordered sequence of actions that, when executed successfully, results in a set of achieved goals for some of the agents. Most techniques can deal with problems where the actions and goals of the agents are only weakly dependent upon each other, where the agents are cooperative, and where communication is reliable. However, in general a multiagent planning approach may encounter a whole variety of situations along these three axes.

- From independent to strongly related
  - Independent: no shared resources, no dependencies
  - Strongly related: joint actions, shared resources
  - E.g. lift a box together, car assembly
- From cooperative to self-interested agents
  - In some settings the participating agents are only interested in optimizing their own utility.
  - E.g. robots in the robocup versus companies in a supply chain

- From no communication possible to reliable communication
  - In hostile environments agents may not or cannot communicate during execution. This may require all coordination to take place before the execution starts.
  - E.g. robots rescuing people in disaster scenarios, or on a planetary exploration mission versus companies in a supply chain

There are benchmark problems with different ranges in the spectra of these properties, such as

- Robocup Rescue [38], where a team of agents of sometimes different types need to coordinate their efforts in dealing with all kinds of disasters,
- DARPA COORDINATORS military team coordination [41, 43, 55, 66], and
- supply chain formation in the Trading Agent Competition [70].

To deal with these problems, many different techniques have been put forward. The next section discusses quite a number of these techniques briefly.

3 Multiagent planning techniques

Multi-agent planning techniques cover quite a range of solutions to different phases of the problem. This section structures existing work using these steps in the process of solving a multiagent planning problem. In general, the following phases can be distinguished (generalizing the main steps in task sharing by [20]).

1. Allocate goals to agents.
2. Refine goals into subtasks.
3. Schedule subtasks by adding resource allocation (possibly including the agents) and timing constraints.
4. Communicate planning choices (of prior steps) to recognize and resolve conflicts.

5. Execute the plans.

Planning is a combination of phases 2 and 3, which are often interleaved. Any of these steps could be performed by one agent or some subset. Not all phases of this general multi-agent planning process need to be included. For example, if there are no common or global goals, there is no need for phase 1. Also, some approaches combine different phases. For example, agents can coordinate their plans while constructing their plans (combination of phase 2, 3, and 4), or postpone coordination until the execution phase (combination of phase 4 and 5), as, e.g., robots may do when they unexpectedly encounter each other while following their planned routes.

In general, any interleaving of the five phases may make sense, depending on the problem, indicating a wide variety of possible problem classes. The following subsections describe some well-known approaches to handling issues arising in each of the phases.

3.1 Goal and task allocation

Centralized methods (such as those mentioned in the next section) often take care of the assignment of goals and tasks to agents during planning. There are, however, many other methods to assign tasks in a more distributed way, giving the agents a higher degree of autonomy and privacy. For example, complex task allocation protocols [53] may be used, or auctions and market simulations.

An auction is a way to assign a task to the agent that attaches the highest value or lowest cost (called private value) to it [68, 73]. A Vickrey [62] auction is an example of an auction protocol that is quite often used in multiagent systems. In a Vickrey auction each agent can make one closed bid, and the task is assigned to the highest bidder for the price of the second-highest bidder. This auction protocol has the nice property that bidding agents should simply bid their true private values (i.e., exactly what they think it's worth to them), removing any need for additional reasoning about its worth to others.

Market simulations and economics can also be used to distribute large quantities of resources among agents [67, 71, 72]. For example, in [6] it is shown how costs and money are turned into a coordination device. These methods are not only used for task assignment (phase 2), but can also be used for coordinating agents after plan construction (phase 5). In the context of value-oriented environments, such game-theoretical approaches where agents reason about the cost of their decision making (or communication) become more important. See, for example, work by Sandholm, supported by results from a multiple dispatch center vehicle routing problem [51].

An overview of value-oriented methods to coordinate agents is given in [28]. Among these, Markov decision processes (MDPs) can deal with settings where outcomes are uncertain, and can even be extended to deal with partially observable worlds. Algorithms often use these representations to compute policies that specify the optimal actions for each agent for any possible belief state. In this survey we focus on deterministic approaches to multiagent planning, but there are surveys on the use of MDPs for multiagent planning under uncertainty [45, 52]. These multiagent approaches rely on earlier work on centralized planning/coordination algorithms in the context of uncertainty and/or partial observability [34, 48].

Value-oriented methods for *self-interested agents* lie within the domain of game theory [2]. On the one hand, literature on using auctions, markets, and negotiation protocols to allocate resource or tasks is far too extensive to cover here. On the other hand, however, work relating game theory (and mechanism design) to multiagent planning is surprisingly scarce [see, e.g. 61].

3.2 Goal and task refinement

In the second phase, the global tasks or goals are refined such that each remaining task can be done by a single agent. Apart from single-agent planning techniques using non-linear planning [47, 50] or Hierarchical Task Networks, HTNs [26], special purpose techniques use the classical planning framework to construct multi-agent plans [37, 46]. A number of planners with more sophisticated models of temporal extent can be applied in this fashion, centralizing and combining phases 2 through 4 [1, 5, 12, 39, 42]. See for example the book on automated planning for an overview of such techniques [33].

3.3 Decentralized planning

Instead of one agent planning for the rest, the second and third phases may be implemented by local planning by each of the agents. In principle, any planning technique can be used here, and different agents may even use other techniques. Some approaches integrate individual planning (phases 2 and 3) with coordination of the plans (phase 4). Early in the history of distributed AI, a distributed version of the NOAH planner demonstrated how to integrate phases 1 through 4, each decentralized, to plan for a single agent in parallel [10], highlighting central issues in distributed planning.

Later, all five phases are interleaved by the Partial Global Planning framework [PGP, 21], and its extension, Generalized PGP [GPGP, 15, 16], where each agent has partial knowledge of the plans of other agents using a specialized plan representation. In this method, coordination is achieved as follows. If an agent A informs another agent B of a part of its own plan, B merges this information into its own partial global plan. Agent B can then try to improve the global plan by, for example, eliminating redundancy it observes. Such an improved plan is shown to other agents, who might accept, reject, or modify it. This process is assumed to run concurrently with the execution of the (first part of the) local plan. PGP has first been applied to the distributed vehicle monitoring test bed, but, later on, an improved version has also been shown to work on a hospital patient scheduling problem. Here Decker and Li [17] used a framework for Task Analysis, Environment Modeling, and Simulation (TÆMS) to model such multi-agent environments in a more general way. Shared Activity Coordination (SHAC) extended GPGP's concept of modeling coordination mechanisms while separating the model and implementation from that of the planning problem and algorithm [9]. An overview of the PGP related approaches is given by [40].

Another approach to agent coordination is through models of mental attitude. The GRATE framework enables agents to coordinate their individual planning by reasoning about their beliefs, desires, intentions, and joint intentions/commitments [36]. Coordination is interleaved with planning by creating and revising commitments through an organizing agent.

3.4 Coordination after planning

A large body of research focused on how to coordinate after plans have been constructed separately (phase 4). These so-called plan merging methods aim at the construction of a joint plan for a set of agents given the individual (sub) plans of each of the participating agents. Georgeff [30, 32] was one of the first to actually propose a plan-synchronization process starting with individual plans. He defined a process model to formalize the actions open to an agent. Parts of such a process model are the correctness conditions, which are defined on the state of the world and must be valid before execution of the plan may succeed. Two agents can help each other by changing the state of the world in such a way that the correctness conditions of the other agent become satisfied. Of course, changing the state of the world may help one agent, but it may also interfere with another agent's correctness conditions [31].

Stuart [56] uses a propositional temporal logic to specify constraints on plans, such that it is guaranteed that only feasible states of the environment can be reached. These constraints are given to a theorem prover to generate sequences of communication actions (in fact, these implement semaphores) that guarantee that no event will fail. To both improve efficiency and resolve conflicts, one can introduce restrictions on individual plans (in phase 3) to ensure efficient merging. This line of action is proposed by Yang et al. [75] and Foulser et al. [29], and can also be used to merge alternative plans to reach the same goal.

Another centralized plan-merging approach addresses problems arising from both conflicts and redundant actions by using the search method A* and a smart cost-based heuristic: Ephrati and Rosenschein [22] showed that, by dividing the work of constructing sub plans over several agents, one can reduce the overall complexity of the merging algorithm [23].

Other work on plan merging propose a distributed polynomial-time algorithm to improve social welfare, the sum of the benefits of all agents [25, 49]. Through a process of group constraint aggregation, agents incrementally construct an improved global plan by voting about joint actions. They even propose algorithms to deal with insincere agents, and to interleave planning, coordination, and execution [24].

The plan merging problem is also blurred with interleaved planning and coordination at multiple levels of abstraction [7]. The idea is that the agents may have partially refined their plans at different levels of detail and can also coordinate them at different levels. Based on a concurrent hierarchical plan (CHiP) representation (adding durative action and consumable/replenishing resources to an HTN), centralized algorithms are given for offline summarization of potential refinements of an abstract task and for exploiting this summary information to more efficiently resolve conflicts in systematic and local planning [8].

This abstract reasoning can also be used by agents to maintain autonomy while exploiting the results of other agents to improve plan efficiency and search performance [11, 13]. In [11] the idea is to add conditional dependencies to the plan: if an agent achieves another's subgoal, the agent can execute a more efficient branch of the plan; otherwise the normal course of action can still be followed. This works succeeds a single-agent approach that uses a conditional simple temporal network (STN) representation to merge redundant actions/subplans across subgoals [59]. In [13] all plans are modeled as resource consuming and producing processes. Such a view allows for efficient plan merging through resource exchanges. The effectivity of this approach is supported by an experimental analysis of applying plan merging to planning data from a taxi company.

3.5 Coordination before planning

Another way agents can coordinate (phase 4) before they even start creating their plans (phases 2 and 3) is by using *social laws*. A social law is a generally accepted convention that each agent has to follow. Such laws restrict the agents in their behavior. They can be used to reduce communication costs and planning and coordination time. In fact, the work of Yang et al. [75] and Foulser et al. [29] about finding restrictions that make the plan merging process easier, as discussed in the previous section, is a special case of this type of coordination. Typical examples of social laws in the real world are traffic rules: because everyone drives on the right side of the road (well, almost everyone), virtually no coordination with oncoming cars is required. Generally, solutions found using social laws are not optimal, but they may be found relatively fast. How social laws can be created in the design phase of a multi-agent system is studied by Shoham and Tennenholtz [54]. Briggs [3] proposed more flexible laws, where agents first try to plan using the strictest laws, but when a solution cannot be found agents are allowed to relax these laws somewhat.

Another way to coordinate agents is to figure out the exact interdependencies between their tasks beforehand. Prerequisite constraints can be dealt with centrally using existing planning technology (such as partial order planning [69] or those mentioned in Section 3.2) by viewing these tasks as single-agent tasks. The summary information discussed used in PGP has also been proposed to precompute the interferences (such as shared resources) among the goals of one agent or a group [8]. Information about the top level of a plan hierarchy can be exchanged among the agents to determine conflicting and also positive relations, and even to match goals to agents [63–65]. If possible, relations are solved or exploited at this top level. If not, a refinement of the plans is made, and the process is repeated, thus, integrating phase 2 and 4.

Coordination before planning can also be used by competitive agents that insist on their planning autonomy [60]. Here, the problem is that the planning agents have a set of interrelated (sub)goals that they have to reach, and they do not want others to interfere with their planning activity. That is, each of the agents requires full planning autonomy, but at the same time they have to be sure that whatever (sub)plans they construct to solve their part of the problem can be coordinated seamlessly without requiring replanning. Planning problems like these often occur in multi-modal transportation problems: several parties have to ensure that packages are transported from their source locations to their destinations. The planning agents are prepared to carry out their part of the job if it can be guaranteed that they will not be interfered by the activities of other agents [4].

It is clear that most of those planning problems cannot be decomposed into independent subproblems without changing the original planning problem. However, temporal constraints can be added to the agents' STNs up front so that they need not communicate at all during scheduling and execution [35]. Another preplanning coordination method adds a minimal set of additional constraints to the subgoals to be performed in order to ensure a coordinated solution by independent planning [58].

3.6 Plan execution

Distributed Continual Planning (DCP) problems often require agents to break and re-make commitments during execution when there are unexpected events/failures or goal changes [19]. Distributed SIPE [18] and CODA [44] explore approaches to interleaving phases 2 through 5 with a focus on minimizing communication. SHAC, mentioned in Section 3.3, incorporates a simple, general algorithm, for which coordination mechanisms are customized to the problem domain.

Recently, a variety of decentralized planning algorithms for handling uncertainty in real time have been developed for scaling to large (100 agents and over 13000 tasks) problems based on TÆMS as part of the DARPA COORDINATORS program [41, 43, 55, 66, 74]. Some of the challenges of these problems include partial observability, deadlines, uncertain duration, uncertain message delay, and dynamic revision of goals. These algorithms interleave phases 2 through 5 in different ways: by computing metrics to communicate and identify the most critical tasks to execute, by using the timing flexibility of STNs to maintain schedule stability while continually exploring optimizations with others, and by regenerating local MDP policies based on changing commitments.

The STEAM collaborative execution framework [57] focuses just on phases 4 and 5 by building on the concept of joint intention mentioned in Section 3.3. This system enables agents to work together to discover when commitments are broken and how to recover from failures and still meet goals.

4 Contributions in this special issue

Problems associated with agent communication and interaction in planning (the fourth phase introduced in Section 3) are at the heart of multiagent planning. The articles of this issue cover many of the different contexts described above but focus on ways to minimize commmunication or interactions for more efficient planning and execution.

- Steenhuisen and Witteveen extend precedence-based temporal decoupling (coordination before planning) to handle synchronization constraints.
- Cox and Durfee introduce an algorithm and problem reformulation techniques for distributed coordination after planning to efficiently "merge" redundant actions and reuse the results of other agents.
- Purrington and Durfee describe complete and approximate algorithms for finding optimal agreements for self-interested planning agents.
- Rosenfeld, Kraus, and Ortiz demonstrate that an agent can learn when it needs feedback from others based on its confidence in making local planning decisions.
- Van der Krogt describes how an agent can measure how much private information it is communicating to others according to the size of the possible plan space.

Acknowledgments

This work is partially supported by the Technology Foundation STW, applied science division of NWO, and the Ministry of Economic Affairs of the Netherlands. The authors would like to thank Adriaan ter Mors and Cees

Witteveen for their co-authorship of a technical report upon which some parts of this introductory article have been based [14].

References

[1] J. Allen, H. Kautz, R. Pelavin, and J. Tenenberg. *Reasoning about plans*. Morgan Kaufmann, 1991.

[2] Robert J Aumann. Game theory. In John Eatwell, Murray Milgate, and Peter Newman, editors, *Game Theory*, The New Palgrave, pages 1–54. Macmillan, London and Basingstoke, 1987.

[3] Will Briggs. *Modularity and Communication in Multi-Agent Planning*. PhD thesis, University of Texas at Arlington, 1996. URL `http://citeseer.nj.nec.com/briggs96modularity.html`.

[4] P. Buzing, A.W. ter Mors, J.M. Valk, and C. Witteveen. Coordinating self-interested planning agents. *Autonomous Agents and Multi-Agent Systems*, 12(2):199–218, February 2006. URL `http://dx.doi.org/10.1007/s10458-005-6104-4`.

[5] S. Chien, R. Knight, A. Stechert, R. Sherwood, and G. Rabideau. Using iterative repair to improve the responsiveness of planning and scheduling. In *Proceedings of the International Conference on AI Planning and Scheduling*, pages 300–307, 2000.

[6] S.H. Clearwater. *Market-Base Control – a paradigm for distributed resource allocation*. World Scientific Publishing Co., 1996.

[7] B. Clement and E. Durfee. Top-down search for coordinating the hierarchical plans of multiple agents. In *Proceedings of the International Conference on Autonomous Agents*, 1999.

[8] B. Clement, E. Durfee, and A. Barrett. Abstract reasoning for planning and coordination. *Journal of Artificial Intelligence Research*, 28:453–515, 2007. URL `http://www.jair.org/papers/paper2158.html`.

[9] Bradley J. Clement and Anthony C. Barrett. Continual coordination through shared activities. In *Proceedings of the Second International*

Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-03), 2003.

[10] D. Corkill. Hierarchical planning in a distributed environment. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 168–175, 1979.

[11] J. S. Cox and E. H. Durfee. Discovering and exploiting synergy between hierarchical planning agents. In *Proceedings of the International Joint Conference on Autonomous Agents and MultiAgent Systems*, pages 281–288, 2003.

[12] K. Currie and A. Tate. O-Plan: The open planning architecture. *Artificial Intelligence*, 52:49–86, 1991.

[13] Mathijs M. de Weerdt. *Plan Merging in Multi-Agent Systems*. PhD thesis, Delft Technical University, Delft, The Netherlands, 2003. URL `www.pds.twi.tudelft.nl/ mathijs/`.

[14] Mathijs M. de Weerdt, Adriaan ter Mors, and Cees Witteveen. Multi-agent planning – an introduction to planning and coordination. Technical report, Delft University of Technology, 2005. URL `http://www.st.ewi.tudelft.nl/ mathijs/publications/easss05.pdf`.

[15] Keith S. Decker and Victor R. Lesser. Generalizing the partial global planning algorithm. *International Journal of Intelligent and Cooperative Information Systems*, 1(2):319–346, June 1992.

[16] Keith S. Decker and Victor R. Lesser. Designing a family of coordination algorithms. In *Proceedings of the Thirteenth International Workshop on Distributed Artificial Intelligence (DAI-94)*, pages 65–84, 1994. URL `http://citeseer.nj.nec.com/decker95designing.html`.

[17] Keith S. Decker and Jinjiang Li. Coordinating mutually exclusive resources using GPGP. *Autonomous Agents and Multi-Agent Systems*, 3(2):113–157, 2000. URL `http://www.cis.udel.edu/ decker/cv.html#PUBLICATIONS`.

[18] M. desJardins and M. Wolverton. Coordinating a distributed planning system. 20(4):45–53, 1999.

[19] Marie E. DesJardins, Edmund H. Durfee, Charles L. Ortiz, and Michael J. Wolverton. A survey of research in distributed, continual planning. *AI Magazine*, 20(4):13–22, 2000.

[20] Edmund H. Durfee. Distributed problem solving and planning. In Gerhard Weiß, editor, *A Modern Approach to Distributed Artificial Intelligence*, chapter 3. The MIT Press, San Francisco, CA, 1999.

[21] Edmund H. Durfee and Victor R. Lesser. Planning coordinated actions in dynamic domains. In *Proceedings of the DARPA Knowledge-Based Planning Workshop*, pages 18.1–18.10, December 1987.

[22] Eithan Ephrati and Jeffrey S. Rosenschein. Multi-agent planning as the process of merging distributed sub-plans. In *Proceedings of the Twelfth International Workshop on Distributed Artificial Intelligence (DAI-93)*, pages 115–129, May 1993. URL `http://www.cs.huji.ac.il/labs/dai/papers.html`.

[23] Eithan Ephrati and Jeffrey S. Rosenschein. Divide and conquer in multi–agent planning. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, pages 375–380, Menlo Park, CA, 1994. AAAI Press.

[24] Eithan Ephrati and Jeffrey S. Rosenschein. A framework for the interleaving of execution and planning for dynamic tasks by multiple agents. In C. Castelfranchi and J.P. Müller, editors, *From Reaction to Cognition — Fifth European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-93), LNAI Volume 957*, pages 139–156, Berlin, 1995. Springer Verlag.

[25] Eithan Ephrati, Martha Pollack, and Jeffrey S. Rosenschein. A tractable heuristic that maximizes global utility through local plan combination. In Victor Lesser, editor, *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 94–101, San Francisco, CA, 1995. AAAI Press, distributed by The MIT Press.

[26] Kutluhan Erol, James Hendler, and Dana S. Nau. HTN planning: Complexity and expressivity. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, volume 2, pages 1123–1128,

Seattle, Washington, USA, 1994. AAAI Press/MIT Press. ISBN 0-262-51078-2. URL `citeseer.nj.nec.com/24780.html`.

[27] Jacques Ferber and Alexis Drogoul. Using reactive multi-agent systems in simulation and problem solving. In Nicholas M. Avouris and Les Gasser, editors, *Distributed Artificial Intelligence: Theory and Praxis*, volume 5 of *Euro Courses: Computer and Information Science*, pages 53–80. Kluwer Academic, The Netherlands, 1992. ISBN 0792315855.

[28] Klaus Fischer, Christian Ruß, and Gero Vierke. Decision theory and coordination in multi-agent systems. Technical Report RR-98-02, DFKI GmbH: German Research Center for Artificial Intelligence, September 1998.

[29] D.E. Foulser, Ming Li, and Qiang Yang. Theory and algorithms for plan merging. *Artificial Intelligence Journal*, 57(2–3):143–182, 1992. URL `http://www.cs.sfu.ca/ isa/pubs/`.

[30] Michael P. Georgeff. Communication and interaction in multi-agent planning. In *Proceedings of the Third National Conference on Artificial Intelligence (AAAI-83)*, pages 125–129, Menlo Park, CA, August 1983. AAAI Press. Also published in [**?**], pages 200–204.

[31] Michael P. Georgeff. A theory of action for multiagent planning. In *Proceedings of the Fourth National Conference on Artificial Intelligence (AAAI-84)*, pages 121–125, Menlo Park, CA, August 1984. AAAI Press. Also published in [**?**], pages 205–209.

[32] Michael P. Georgeff. Communication and interaction in multi-agent planning. In A. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 200–204. Morgan Kaufmann Publishers, San Mateo, CA, 1988. Also published as [30].

[33] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning, theory and practice*. Morgan Kaufmann Publishers, 2004. ISBN 1-55860-856-7.

[34] Eric A. Hansen and Shlomo Zilberstein. Lao*: A heuristic search algorithm that finds solutions with loops. *Artificial Intelligence*, 129(1-2): 35–62, 2001.

[35] L. Hunsberger. Distributing the control of a temporal network among multiple agents. In *Proceedings of the National Conference on Artificial Intelligence*, 2002.

[36] N. R. Jennings. Specification and implementation of belief, desire, joint-intention architecture for collaborative problem solving. *Journal of Intelligent and Cooperative Information Systems*, 2(3):289–318, 1993.

[37] Matthew J. Katz and Jeffrey S. Rosenschein. Plans for multiple agents. In Les Gasser and Michael N. Huhns, editors, *Distributed Artificial Intelligence*, volume 2 of *Research Notes in Artificial Intelligence*, pages 197–228. Pitman Publishing and Morgan Kaufmann Publishers, London, UK, 1989.

[38] Hiroaki Kitano and Satoshi Tadokoro. Robocup rescue: A grand challenge for multiagent and intelligent systems. *AI Magazine*, 22(1):39–52, 2001.

[39] P. Laborie and M. Ghallab. Planning with sharable resource constraints. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1643–1649, 1995.

[40] Victor Lesser, Keith Decker, N. Carver, A. Garvey, D. Neimen, M.V. Prassad, and Thomas Wagner. Evolution of the GPGP domain independent coordination framework. Technical Report UMASS CS TR 1998-005, University of Massachusetts, 1998.

[41] R. T. Maheswaran, P. Szekely, M. Becker, S. Fitzpatrick, G. Gati, J. Jin, R. Neches, N. Noori, C. Rogers, R. Sanchez, K. Smyth, and C. Van-Buskirk. Predictability & criticality metrics for coordination in complex environments. In *7th International Joint Conference on Autonomous Agents and Multiagent Systems*, Estoril, Portugal, May 2008.

[42] N. Muscettola. HSTS: Integrating planning scheduling. *Intelligent Scheduling*, pages 169–212, 1994.

[43] David J. Musliner, Edmund H. Durfee, Jianhui Wu, Dmitri A. Dolgov, Robert P. Goldman, and Mark S. Boddy. Coordinated plan management using multiagent MDPs. In *Working Notes of the AAAI Spring Symposium on Distributed Plan and Schedule Management*, March 2006.

[44] K. Myers, P. Jarvis, and T. Lee. CODA: Coordinating human planners. In *Proceedings of the European Conference on Planning*, 2001.

[45] F. A. Oliehoek, M. T. J. Spaan, and N. Vlassis. Optimal and approximate q-value functions for decentralized pomdps. *Journal of Artificial Intelligence Research*, 32:289–353, 2008.

[46] Edwin P.D. Pednault. Formulating multi-agent dynamic-world problems in the classical planning framework. In Michael P. Georgeff and Amy L. Lansky, editors, *Reasoning About Actions and Plans — Proceedings of the 1986 Workshop*, pages 47–82, San Mateo, CA, 1987. Morgan Kaufmann Publishers. Also published in [**?**].

[47] J. Penberthy and Daniel S. Weld. UCPOP: A sound, complete, partial order planner for ADL. In *Proceedings of the Third International Conference on Knowledge Representation and Reasoning (KR&R-92)*, pages 103–114. Morgan Kaufmann Publishers, October 1992.

[48] D.V. Pynadath and M. Tambe. The communicative multiagent team decision problem: Analyzing teamwork theories and models. *Journal of AI Research*, 16:389–423, 2002.

[49] Jeffrey S. Rosenschein. Multiagent planning as a social process: Voting, privacy, and manipulation. In Victor R. Lesser, editor, *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, page 431, San Francisco, CA, 1995. AAAI Press, distributed by The MIT Press. URL `http://www.cs.huji.ac.il/labs/dai/papers.html`.

[50] Earl D. Sacerdoti. The nonlinear nature of plans. In *Proceedings of the Fourth International Joint Conference on Artificial Intelligence (IJCAI-75)*, pages 206–214, San Mateo, CA, 1975. Morgan Kaufmann Publishers.

[51] Tuomas W. Sandholm and Victor R. Lesser. Coalitions among computationally bounded agents. *Artificial Intelligence*, 94(1):99–137, 1997. URL `http://www.cs.wustl.edu/ sandholm/`.

[52] S. Seuken and S. Zilberstein. Formal models and algorithms for decentralized decision making under uncertainty. *Autonomous Agents and Multi-Agent Systems*, 17(2):190–250, 2008.

[53] O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1–2):165–200, May 1998.

[54] Yoav Shoham and Moshe Tennenholtz. On social laws for artificial agent societies: Off-line design. *Artificial Intelligence*, 73(1–2):231–252, 1995. URL `http://citeseer.nj.nec.com/shoham95social.html`.

[55] S. Smith, A.T. Gallagher, T.L. Zimmerman, L. Barbulescu, , and Z. Rubinstein. Distributed management of flexible times schedules. In *Proceedings of the International Joint Conference on Autonomous Agents and MultiAgent Systems*, pages 1–8, New York, NY, USA, 2007. ACM. ISBN 978-81-904262-7-5. doi: http://doi.acm.org/10.1145/1329125.1329215.

[56] Christopher J. Stuart. An implementation of a multi-agent plan synchronizer. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85)*, pages 1031–1033, San Mateo, CA, August 1985. Morgan Kaufmann Publishers. Also published in [**?**], pages 216–219.

[57] M. Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124, 1997.

[58] Adriaan ter Mors, Jeroen Valk, and Cees Witteveen. Coordinating autonomous planners. In *Proceedings of the international conference on artificial intelligence*, pages 795–801. CSREA Press, June 2004.

[59] I. Tsamardinos, M. E. Pollack, and J. F. Horty. Merging plans with quantitative temporal constraints, temporally extended actions, and conditional branches. In *Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems (AIPS-00)*, pages 264–272, Menlo Park, CA, April 2000. AAAI Press.

[60] Jeroen M. Valk, Mathijs M. de Weerdt, and Cees Witteveen. Algorithms for coordination in multi-agent planning. In Vlahavas and Vrakas, editors, *Intelligent Techniques for Planning*, pages 194–224. Idea Group Inc., 2005. URL `http://www.idea-group.com/books/details.asp?id=4496`.

[61] Roman P.J. van der Krogt, Mathijs M. de Weerdt, and Yingqian Zhang. Of mechanism design and multiagent planning. In Malik Ghallab, Constantine D. Spyropoulos, Nikos Fakotakis, and Nikos Avouris, editors, *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI-08)*, pages 423–427. IOS Press, 2008. ISBN 978-1-58603-891-5.

[62] W. Vickrey. Computer speculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

[63] Frank von Martial. Multiagent plan relationships. In *Proceedings of the Ninth International Workshop on Distributed Artificial Intelligence (DAI-89)*, pages 59–72, September 1989.

[64] Frank von Martial. Coordination of plans in multiagent worlds by taking advantage of the favor relation. In M. Huhns, editor, *Proceedings of the Tenth International Workshop on Distributed Artificial Intelligence (DAI-90)*, number ACT-AI-355-90 in MCC Technical Report, Austin, TX, October 1990.

[65] Frank von Martial. *Coordinating Plans of Autonomous Agents*, volume 610 of *Lecture Notes on Artificial Intelligence*. Springer Verlag, Berlin, 1992.

[66] Thomas A. Wagner, Anita Raja, and Victor R. Lesser. Modeling uncertainty and its implications to sophisticated control in TÆMS agents. *Journal of Autonomous Agents and Multi-Agent Systems*, 13(3):235–292, 2006. ISSN 1387-2532. doi: http://dx.doi.org/10.1007/s10458-006-7669-2.

[67] William E. Walsh and Michael P. Wellman. A market protocol for decentralized task allocation and scheduling with hierarchical dependencies. In *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS-98)*, pages 325–332, 1999. URL `http://www-personal.engin.umich.edu/ wew/research.html`. An extended version of this paper is also available.

[68] William E. Walsh, Michael P. Wellman, and F. Ygge. Combinatorial auctions for supply chain formation. In *Second ACM Conference on Electronic Commerce*, pages 260–269. ACM Press, 2000. URL `http://www-personal.engin.umich.edu/ wew/research.html`.

[69] Daniel S. Weld. An introduction to least-commitment planning. *AI Magazine*, 15(4):27–61, 1994. URL `http://www.cc.gatech.edu/ jimmyd/summaries/weld1994.html`.

[70] M. Wellman, A. Greenwald, P. Stone, and P. Wurman. The 2001 trading agent competition. *Electronic Markets*, 13(1):4–12, Jan 2003. URL `http://www.informaworld.com/index/EDY5MFAR5DPHEC2E.pdf`.

[71] Michael P. Wellman. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research*, 1:1–23, 1993.

[72] Michael P. Wellman, William E. Walsh, P.R. Wurman, and J.K. MacKie-Mason. Auction protocols for decentralized scheduling. In *Proceedings of the Eighteenth International Conference on Distributed Computing Systems*, May 1998.

[73] Michael P. Wellman, William E. Walsh, Peter R. Wurman, and Jeffrey K. MacKie-Mason. Auction protocols for decentralized scheduling. *Games and Economic Behavior*, 35(1–2):271–303, 2001. URL `http://www-personal.engin.umich.edu/ wew/research.html`.

[74] Jianhui Wu and Edmund H. Durfee. Solving large TÆMS problems efficiently by selective exploration and decomposition. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1–8, New York, NY, USA, 2007. ACM. ISBN 978-81-904262-7-5. doi: http://doi.acm.org/10.1145/1329125.1329191.

[75] Qiang Yang, Dana S. Nau, and James Hendler. Merging separately generated plans with restricted interactions. *Computational Intelligence*, 8(4):648–676, November 1992. URL `http://www.cs.sfu.ca/ isa/pubs/index.html`.

Biography of the authors

Mathijs de Weerdt completed his Master's in computer science at the Utrecht University. After that he did his PhD on "Plan Merging in Multiagent Systems" at the Delft University of Technology. Since then he is an assistant professor in Algorithmics. In 2004 he obtained a VENI grant to study the interaction of efficient planning and task allocation algorithms with coordination mechanisms for self-interested agents. He has given tutorials on multiagent planning in previous editions of the EASSS and at the AAMAS, and he has organized international workshops on multi-agent planning. His primary research interests lie in multiagent planning and mechanism design.

Brad Clement received a bachelor degree in computer engineering from the Georgia Institute of Technology and M.S. and Ph.D. degrees in computer science and engineering from the University of Michigan, Ann Arbor. He is a senior member of the Artificial Intelligence Group at the Jet Propulsion Laboratory where he is developing methods for coordinating planning and scheduling for single and multiple spacecraft/missions. His research is centered on developing techniques and software that leverage models of human domain knowledge to coordinate (software, robotic, or human) agents in performing real-time planning, scheduling, and execution tasks in uncertain or partially known environments. He has organized several tutorials and workshops on multiagent planning.