Multiagent Planning through Plan Repair*

Roman van der Krogt
Delft University of Technology
r.p.j.vanderkrogt@ewi.tudelft.nl

Mathijs de Weerdt†
Delft University of Technology
m.m.deweerdt@ewi.tudelft.nl

Nico Roos
University of Maastricht
roos@cs.unimaas.nl

Cees Witteveen
Delft University of Technology
c.witteveen@ewi.tudelft.nl

ABSTRACT

We present a novel approach to multiagent planning for self-interested agents. The main idea behind our approach is that multiagent planning systems should be built upon (single-agent) *plan repair* systems. In our system agents can exchange goals and subgoals through an auction, using their own (planning) heuristics and utility functions to determine when to auction and what to bid. Some experimental results for a logistics domain show that this system can be used to support the coordination of self-interested agents.

Categories and Subject Descriptors

I.2.8 [**Problem Solving, Control Methods, and Search**]: Plan execution, formation, and generation; I.2.11 [**Distributed Artificial Intelligence**]: Intelligent agents, Multiagent Systems

General Terms

Algorithms, Design

Keywords

Multiagent Planning

1. INTRODUCTION

Most interesting applications of planning involve more than one agent to plan for. Often these agents are self-interested and require some privacy concerning their plans. We propose a system in which *self-interested* agents can (i) *construct their plans* themselves, (ii) *coordinate* their actions during planning, and do so while (iii) maintaining their *privacy*. With this system we take the challenge of negotiated distributed planning that "methods must be developed for adapting the various [existing] approaches in a way that is consistent with the resource-constrained nature of planning agents: planning should be a continuous, incremental process at both the individual and group level." [4].

Our idea is to combine a dynamic planning method for each agent with an auction for delegating (sub)tasks. However, to coordinate subtasks we should deal with inter-agent dependencies [6] to prevent deadlocks. Currently, multiagent planning methods manage inter-agent dependencies at a central place, or by constructing and communicating a (partial) global plan [3]. Obviously, in many applications, agents are not prepared to share this kind of information.

In our system, we have a number of agents that first concurrently plan for a single goal, after which they take part in an auction (if there is any) to exchange subgoals some of them cannot attain themselves. Then, they apply a plan repair technique to add another goal to their plan, and take part in an auction again. They continue to alternatingly perform these steps of adapting a plan using plan repair and taking part in an auction until a complete and valid plan is computed. When an agent gets a task assigned on which others depend, we use a heuristic that lets the agent schedule it early in its plan to prevent cyclic dependencies. Furthermore, we give the agents some high-level information about the services others can provide to reason about which subgoals they should auction.

2. EXPERIMENTS

To test this type of multiagent planning, we used the following logistics problem: a number of independent planning agents have to transport goods between different locations in different cities. For each of the cities, there is an agent that is capable of transporting goods *within* that city, using trucks. For transport *between* cities, only one agent can transport goods by air from one airport to another. Thus, for a typical transportation order, three agents have to work together: one to bring the goods from their current location to the airport in that city, one to transport the goods to another airport, and a third agent is required for the trans-

*A full version of this paper is available as [9].

†This research is supported by the Technology Foundation STW, applied science division of NWO, and the technology program of the Dutch Ministry of Economic Affairs.

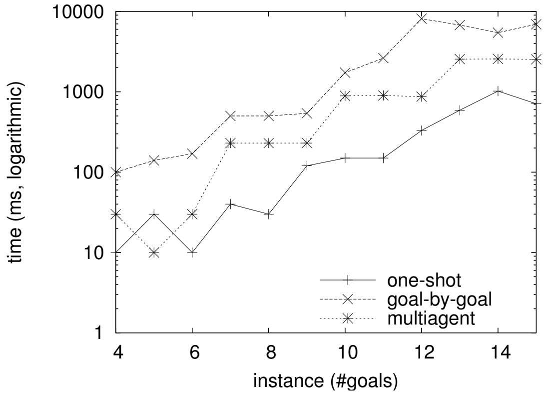

Figure 1: Run times of multi-agent planning compared to single-agent planning

port from that airport to the destination within that same city. As these agents represent different companies, they are self-interested and competitive. However, they are willing to help each other, provided adequate compensation is offered.

In our experiments on this problem from the AIPS competition [1] we compared three algorithms: (i) our proposed *multiagent* method, where we allow such companies to construct their plans individually, while coordinating (some of) their actions and maintaining their privacy, (ii) a central *one-shot* planning algorithm (for all goals of all agents), and (iii) a central *goal-by-goal* planning where goals are added one by one, like in our multiagent method. We took 11 different problems with 4 to 15 goals. For a problem with n goals, we used $\lceil \frac{n}{3} \rceil$ cities. The dynamic planning method we used in our experiments is the POPR plan repair system [8], which is an adaptation of the VHPOP planner by [11]. The run times for the three methods are shown in Figure 1. Here we can see that the multiagent method (due to its parallelism) outperforms central planning using the goal-by-goal approach by almost an order of magnitude. On the downside, our multiagent method produces plans that are about 50% longer than (one-shot) centrally produced plans.

3. DISCUSSION

We gave experimental evidence that agents can plan individually, and coordinate their plans by exchanging subgoals. Our method should work with any plan repair algorithm, allowing agents to choose their own dynamic planner.

The distribution of the planning problem in a multiagent planning system leads to an improvement of planning performance compared to a single-agent solving a planning problem goal-by-goal. We expect that for more realistic and more complicated domains the difference may be even larger, since agents can do a lot of work in parallel. Summarizing, from the experiments we conclude that it is indeed possible to use multiple single-agent plan repair systems to let self-interested agents plan for their goals individually, and request (or provide) help when necessary.

This system for coordinating self-interested agents using propositional plan repair is unique in that we do not assume that the agents are *collaborating*. Agents may even be each other's competitors. Previous work on multiagent planning, although often more advanced in modeling problems realistically (by involving time constraints, minimizing costs, and efficient use of resources) assumes that the agents are collaborative. For example, in the Cougaar system [5] and the Generalized Partial Global Planning (GPGP) method [3] cooperative agents are coordinated by exchanging more and more details of their plans until conflicts can be resolved.

Next to work on coordinating multiagent plans, there is also a substantial body of work on *task allocation* for self-interested agents. For example using market mechanisms [10], or using extensions of the contract-net protocol [2]. Ideas from this work may be used to improve the simple auction of our approach, for example to enable parallel or combinatorial auctions. Task (re)allocation, however, cannot completely be disconnected from planning. In our work we focus not so much on task allocation, but on coordinating the agents' *planning* and *plan repair* behavior (without the construction of a global set of constraints).

Since our initial experiments showed promising results, we intend to continue this line of research towards a fully equipped multiagent planning system. First, we would like to have a method to estimate the costs of subgoals to be auctioned, to make more informed decisions on what to auction. Another important topic for future study is using a different type of auction and (de)committing mechanism (e.g. [7]) that matches the specific requirements of efficiently allocating sets of subtasks to self-interested planning agents.

Furthermore, the algorithm for each agent is currently sequential: it adapts its plan to include a new goal, then reasons to bid for an auction, then plans a goal again, and so on. We would like to have two independent subprocesses per agent taking care of each of these tasks.

4. REFERENCES

[1] F. Bacchus, H. Kautz, D. Smith, D. Long, H. Geffner, and J. Koehler. The 5th AIPS Planning Competition. http://www.cs.toronto.edu/aips2000/, 2000.

[2] J. Collins, M. Tsvetovatyy, M. Gini, and B. Mobasher. MAGNET: A multi-agent contracting system for plan execution. In *Proc. of SIGMAN*, 1998.

[3] K. Decker and J. Li. Coordinating mutually exclusive resources using GPGP. *Autonomous Agents and Multi-Agent Systems*, 3(2):113–157, 2000.

[4] M. DesJardins, E. Durfee, C. Ortiz, and M. Wolverton. A survey of research in distributed, continual planning. *AI Magazine*, 20(4):13–22, 2000.

[5] K. Kleinmann, R. Lazarus, and R. Tomlinson. An infrastructure for adaptive control of multi-agent systems. In *IEEE Int. Conf. on Integration of Knowledge Intensive Multi-Agent Systems*, pages 230–236, 2003.

[6] T. W. Malone and K. Crowston. The interdisciplinary study of coordination. *ACM Computing Surveys*, 21(1):87–119, 1994.

[7] T. W. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1–2):1–54, 2002.

[8] R. van der Krogt and M. de Weerdt. Plan repair as an extension of planning. In *Proc. of the Int. Conf. on Automated Planning and Scheduling*, 2005.

[9] R. van der Krogt and M. de Weerdt. Self-interested planning agents using plan repair. In *Proc. of the ICAPS Workshop on Multiagent Planning and Scheduling*, 2005.

[10] W. Walsh and Wellman. A market protocol for decentralized task allocation and scheduling with hierarchical dependencies. In *Proc. of the 3rd Int. Conf. on Multi-Agent Systems*, pages 325–332, 1999.

[11] H. L. S. Younes and R. G. Simmons. VHPOP: Versatile heuristic partial order planner. *JAIR*, 20:405–430, 2003.