Multiagent planning: problem properties that matter

Mathijs de Weerdt and Cees Witteveen
Delft University of Technology
{M.M.deWeerdt, C.Witteveen}@tudelft.nl

Abstract

Many different problems are called distributed or multiagent planning problems. Existing approaches each deal with only a subset of these problems. Which properties are essential in determining a solution method for multiagent planning problems? We argue that mainly the facts that communication is limited, agents have private goals, and the frequency of the dependencies between agents have a significant impact on the solution method.

Introduction

Applying standard (single agent) planning techniques to planning problems that involve multiple agents is often not very successful or sometimes even impossible. Why is this? Which properties require a different approach? This paper aims at answering these questions. First, we study some commonly given arguments for involving agents in the planning process, because standard planning techniques are inadequate. This leads to a list of properties that are relevant for choosing a solution method. In the final section we indicate how known approaches can fit into these dimensions.

Properties of planning with agents

The main difference between planning for a single agent and planning for multiple agents is, obviously, the fact that more agents are involved in the execution of the plan. In principle it is possible to create a plan for multiple agents in a similar way as for one agent. However, many different arguments are usually given for involving these agents also in the *plan computation* phase:

1. Planning is done by multiple agents in parallel to reduce the computation overhead of planning at a central location.
2. A central planning means a central point of failure, and is undesirable in this domain.
3. The agents have limited communication possibilities, so it is impossible to receive all observations and send all actions to the agents within time.
4. The agents have knowledge, goals, priorities, and/or tasks that they do not want others to know.

First observe that reasons 1. and 2. for doing a distributed computation do not require to distribute the plan computation over the agents *themselves*. It is in principle possible to let the plans of all agents be computed by some (clusters of) computers. The fact that the problem involves multiple agents can of course be used in the solution method, just as any other additional information on the domain can help.

The other two reasons for distributing the planning process over the agents have as a consequence that (part of the) computation *for the agent* is also done *by the agent*. In the remainder of this paper we focus on these cases, and consider the impact of their problem properties in more detail.

Limited communication For example in the RoboCup Rescue domain (Kitano & Tadokoro 2001) the rescuing robots act in a hostile environment. It is uncertain whether sent messages will arrive at the receiving side. Clearly, in this case it is a bad idea to use methods that rely on secure communications. In some cases agents should plan and act even when they did not receive confirmations.

Private goals Different branches of an organization, like different army divisions, or public services, may have shared tasks, but usually have different, and possibly even conflicting, goals.

A planning support system for such an organization would need to be controlled by different people, having different goals or at least different priorities. Each of these controlling entities probably claims that their own goals are the most important ones, and would prefer a system that would act accordingly. Consequently, it will be very hard, or in many cases even impossible, to settle down for one planning system dealing with all goals of everyone involved simultaneously (even without considering the complexity of solving such problems). The key issue here is that these agents are primarily *self-interested;* their focus is on attaining their *own* goals. Consider the following example with a shared task, but also with some private goals.

Both the fire brigade and the police would like a fire in a tunnel to be extinguished with the least possible casualties. However, besides the shared task of dealing with such a disaster, the fire brigade would like to show (off) the use of their very large, brand new, and very expensive fire truck. However, due to its size, this truck requires the police to shut off one additional small road near the tunnel. The police on the other hand is interested in minimizing the number of required officers assigned to the tunnel disaster to be able to still assign some to the stadium, where an important baseball game is taking place at the same time.

When multiple *companies* are involved in one task (e.g., from a supply chain to selling the goods), situations with private goals are even easier to see, because in principle, each company is mainly interested in maximizing its own profits.[1] When agents have private goals, a whole set of other properties are relevant to determine applicable solution techniques.

Conflicting goals When agents have private goals, it is not trivial to find out whether some of their goals are conflicting. Which goals should be attained and which not? We believe that this should be the result of some negotiation process between these agents. Whenever agents have private goals they usually cannot send each other details of their plans. So in that case we need some method of coming to an agreement between agents anyhow. Whether goals are conflicting then only makes a marginal difference.

Dependencies between agents A dependency between agents arises when one agent needs others to fulfill a subtask, or when the agents can only fulfill a task if they act on it together, or when one agent needs to wait for others to finish occupying a resource (Malone & Crowston 1994). If (parts of) plans may be exchanged it is straightforward to check the plans of the agents for deadlocks. But what if goals, and thus plans, need to be kept private?

The types of tasks and the abilities of the agents determine the number of dependencies between agents. Problems with many dependencies require much more coordination and thus communication between the agents.

Shared tasks Whether a problem includes agents that have shared tasks or goals is strongly related to the property of dependencies between agents. Suppose we have a method that deals with tasks assigned to one responsible agent. Such a method can be adapted to deal with tasks shared among a group of agents as well, by assigning the task to every agent, and introducing dependencies between all agents in this group.

Dynamism Plans executed in the real world need to be adapted continuously, because of incidents, or just as more information becomes available. Consequently planning agents need to be able to deal with such changes. In a multiagent planning system, agents deal with changes all the time, because of other agents' plans. Therefore, the dynamicity of the domain does not make a significant difference.

Summary Some of the properties discussed above have only a minor impact on the potential solution methods. Leaving those ones out, we remain with the following important dimensions of multiagent planning problems.

1. Limited communication
2. Private goals: ranging from completely opposite goals to different priorities of shared tasks
3. Dependencies between agents: ranging from strongly interdependent to (almost) independent

[1] Note that in some cases it may even be that one agent has a goal of obstructing another agent as much as possible.

Obviously, these dimensions are a bit fuzzy, but they illustrate the diverse kinds of problems we are trying to solve.

Conclusions

We argue that only three problem properties have a significant impact on the solution methods that can be used for a multiagent planning problem. Firstly, when communication is limited, agents should not wait for commitments before acting. Secondly, when agents have private goals, it is not advisable to exchange parts of plans, and commitments should be made through some form of negotiation process. And thirdly, when dependencies between agents are frequent, the process of committing and decommitting should take a much higher pace.

These problem dimensions can now be used to describe the kinds of problems an approach can handle. For example, (Generalized) Partial Global Planning (Durfee & Lesser 1987) works quite well when communication is not very limited, agents do not have private goals, and for both strongly interdependent to independent agents. The same holds for the Cougaar system (Kleinmann, Lazarus, & Tomlinson 2003) where cooperative agents are coordinated by exchanging more and more details of their hierarchical plans until conflicts can be resolved.

We feel that it is also important to look at problems where agents have private goals. Therefore, we recently did an initial study using propositional plan repair (van der Krogt & de Weerdt 2005). Despite some strong assumptions (strict ordering rules for dependencies, and not too many dependencies between agents), this proves that it is possible to create multiagent plans distributedly for agents that do not wish to exchange (parts of) their plans.

Acknowledgments The authors wish to thank Roman van der Krogt for his useful comments.

References

Durfee, E. H., and Lesser, V. R. 1987. Using partial global plans to coordinate distributed problem solvers. In *Proc. of the 10th Int. Joint Conf. on AI*, 875–883. Morgan Kaufmann Publishers.

Kitano, H., and Tadokoro, S. 2001. Robocup rescue: A grand challenge for multiagent and intelligent systems. *AI Magazine* 22(1):39–52.

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

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

van der Krogt, R., and de Weerdt, M. 2005. Coordination through plan repair. In *MICAI 2005: Advances in Artificial Intelligence*, volume 3789 of *LNAI*, 264–274. Springer.