Coordinating Self-Interested Autonomous Planning Agents

Adriaan ter Mors and Cees Witteveen

Faculty of Electrical Engineering, Mathematics and Computer Science,
Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands,
{a.w.termors, c.witteveen}@ewi.tudelft.nl

This paper will appear in the *Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology (IAT05)*, IEEE Computer Science Press, Los Alamitos, CA, 2005.

Coordination in multi-agent planning systems aims at ensuring that the plans of the participating agents do not conflict and the individual as well as the common goals of the agents can be achieved. In the multi-agent planning literature, one can distinguish three main approaches to coordination. In the first approach coordination between the agents is established *after* the completion of the individual planning processes. It is assumed that agents independently work on their own part of the planning problem and achieve a solution for it. Then, in an after-planning coordination phase, possible conflicts between independently generated individual plans are resolved and positive interactions between them are exploited by exchanging and revising parts of the individual plans. The second approach treats coordination and planning as *intertwined* processes where the agents continuously exchange planning information to arrive at a joint solution. In the third approach coordination takes place *before* the agents make their plans. Well-known examples of the *pre-planning coordination* approach use *implicit* coordination techniques such as social laws and conventions, and (negotiation) protocols such as the Contract Net protocol.

We focus on coordination between agents that are self-interested, do not want to be interfered with *during* their individual planning processes and do not want to *revise* their plans when a joint plan has to be composed. Examples are (*i*) planning for multi-modal transportation tasks by several independent and competitive transportation companies, (*ii*) manufacturing tasks, and (*iii*) patient-centered health-care systems. It can easily be seen that the requirements of autonomous planning and revision-free combination remove the first two coordination approaches from consideration: post-planning coordination will, in general, require agents to revise their individual plans, while coordination during planning violates the requirement of autonomous planning. Therefore, we need to coordinate the agents *before* the planning phase. While existing pre-planning coordination research focuses on implicit coordination approaches — constraints are imposed independently of the particular goals or tasks the agents have to solve — we are looking for an *explicit* coordination approach: based on the specific set of tasks to be achieved, the assignment policies, and the (inter-agent) dependencies, we specify which constraints have to be imposed on the tasks to achieve revision-free coordination between the agents.

To analyze this coordination problem, we use a task-based framework where a number of agents, each having different capacities, have to solve a complex task. Here, a complex task consists of a hierarchically specified task network where two kinds of relations are distinguished: a *refinement* relation specifying which (set of) tasks constitute a refinement of a certain task t and a partially ordered *dependency* relation, specifying which tasks have to be completed before a given task t can be started. Each task requires specific (agent) capabilities and has to be assigned to an agent with sufficient capabilities. Since an agent might be assigned a task that depends on the completion of tasks assigned to other agents, it cannot plan the execution of this task independently of the other agents. The only way to decouple the planning activities of the different agents is to find a *minimal* set of additional dependency constraints that guarantees that, whatever plans are chosen by the individual agents, these plans can be combined into a feasible joint plan.

In this paper we address the computational complexity of some aspects of this coordination problem and we show that in general, even the easiest problems are intractable. Our results can be summarized as follows:

1. *checking coordination*: Assuming that the tasks have already been assigned to agents, to detect whether or not additional dependency constraints are needed in order to guarantee coordination is a co-NP complete problem in general (in the number of agents), even if each agent has to make a plan for achieving at most four tasks. Only if an agent has at most three tasks to plan, this problem can be solved efficiently. If the agents are not assigned to tasks and we ask whether or not there exists an assignment of tasks to agents such that no additional constraints are needed for coordination, this problem is Σ_2^p-complete.
2. *finding a minimal set of constraints*: the problem of deciding whether k additional constraints suffice is a Σ_2^p-complete problem, even if agents have to plan for a constant ($\geq$8) number of tasks. Remarkably, it makes no essential difference in computational complexity whether tasks have already been assigned or not. Only if we want to find out whether adding k constraints suffices for all possible assignments of tasks to agents does the complexity increase: this problem is Π_3^p-complete.

Coordination before planning has many advantages: agents can plan independently of other agents, and there is no need for advanced multi-agent planning tools because an agent has only its own planning problem to solve. The drawback of pre-planning coordination is that it is computationally expensive, even if the planning task of each of the agents is trivial. Practical application of pre-planning coordination therefore has to focus on the development of approximation algorithms. Future work will concentrate on investigating simple domain-independent heuristics as well as domain-specific heuristics.