Distributed Planning
based on [1]

Artur Daudrich

University Koblenz-Landau

1 Introduction

To address problems which cannot be solved by one individual agent, like monitoring a large area, distributed problem solving and planning is needed. But there are further benefits. In distributed problem solving and planning tasks are addressed which can be accomplished better, more quickly, more precisely and more certainly by multiple agents than by one individual agent.

2 Task Sharing

If a large task is to be solved by an agent, there is a general strategy to tackle this. At first all requirements for a feasible solution must be defined. This is necessary to check if a found solution solves the problem in the intended way. After that the agent searches for an solution for the problem. If this solution satisfies the requirements the agent solves the problem. Otherwise the agent searches for another solution.

In this article I will use an example of garden agents which solve different tasks in a garden environment. So if there exist one lawn mower agent which has the task to mow the lawn, the agent would make a plan how to mow the lawn in the most efficient way. Most efficient with the meaning of covering the whole area, but trying to reduce areas where the lawn is mowed twice. As one can see this task can be solved faster if more than one agent is mowing the lawn and the task is distributed among the agents. If all agents in a system have the same abilities, expertise and capabilities, the system is called a "homogenous system" and the agents are called "omni-capable agents".

In a homogenous system an agent can enlist the help of other agents if it is assigned with too many or a too large task. The agent first decomposes the task into smaller subtasks and to allocate the smaller subtasks to the agents. Each agent tackles now its subtask. After the subtasks are finished the results are synthesized and the bigger task is solved. If an agents subtask is to large it also can enlist the help of other agents.

The problem with omni-capable agents is that they are mostly overkill and most capabilities are wasted. In the garden scenario imagine there is a pond and a hedge. The pond has to be cleaned and the hedge has to be cut. With omni-capable agents every agents would have the capabilities to do so. But most of the agents won't ever need to cut the hedge or clean the pond. So these capabilities are wasted. This leads to a system of specialists, called "heterogeneous system".

In this system task allocation is not trivial anymore, because every agent has other capabilities. The assigning agent cannot just assign a task to a random agent. To tackle this problem there exist a lookup table where all agents are listed with their capabilities. This lookout table can be located in an agents memory or somewhere centralized. If an agent wants to assign a task it looks into the table and calls one capable agent. This approach is called "directed contract". If an agent calls all capable agents for one task we speak of "focussed addressing". To fill the table with information an agent has to broadcast and ask for capabilities. If it gets positive replies the agent can update the table with the new gained information.

If the agent assigns a task and gets a acknowledgement, all is good. But there are cases where an addressed agent is busy. The announcing agent has now the possibility to wait and retry assigning the task until the busy agent is free again. Otherwise it can try to assign the task to another agent. In some cases the assigned task is to complex or to large and the assigning agent won't get a feasible agent. To address this issue the agent has to revise its announcement, by lowering the expectations (Announcement Revision). Last the announcing agent can try to decompose a large task in other ways if it gets no feasible agents for solving the subtasks (Alternative Decomposition).

3 Result Sharing

In most agent systems actions of agents depend on actions from other agents. To not end up in a conflicted state result sharing is necessary. But result sharing is not only to address the issue of interdependencies. With result sharing more precise solutions are possible, the confidence of correct results is higher and tasks can be accomplished more complete.

One approach of result sharing is called "functional accurate cooperation". Each agent formulates tentative results (functional accurate) and iteratively exchanges its partial solutions with the other agents (cooperation). This has an impact on the completeness, precision and confidence of the partial solutions and leads to an overall solution. One big issue is the overhead in communication. Too many shared results can also lead to distraction, this means all agents do the same problem solving actions.

To address this issue communication has to be limited. One approach is to use a single shared repository. In a shared repository every agent can store its partial results, search through all results, extend and revise results. By revising results can be improved, rejected or rejected results again stored if expectations are relaxed. Another approach is to use communication strategies. Instead of sending all partial results to everyone, results could only be send if they are complete. Furthermore results have to be send only to interested agents and at the right time. Timing is an issue, because if a result is send too late, it leads to delayed actions or is not useful anymore. Otherwise if it is send to early it clutters the memory of the receiving agents. Last option is to send results only if they are requested.

Another important issue is the detection of lost messages. To solve this issue result sending is repeated until an acknowledgement is received or a timeout is reached. Further the sending agent can predict and observe a change in the behavior of the recipient.

4 Distributed Planning

Distributed Planning is a specialization of Distributed Problem Solving. The Problem to solve is to construct a plan. There exist three types of distributed planning. First of all a plan can be distributed on an execution system. Second the planning process can be distributed. And last the whole planning and execution process can be distributed.

In the centralized planning for distributed plans there exist one centralized coordinator agent. This agent has to find a plan with few dependencies or ordering constraints, split up the plan, synchronize the actions of the agents and assign the plan pieces to the individual agents. After that it has to trigger and monitor the plan execution.

In distributed planning for centralized plans there exist one plan and all agents try to cooperate in the planning process. They generate partial-specified plans in parallel, they exchange and share plans. To complete the overall plan partial plans have to be merged. If the planning stucks on a subtask backtracking is used.

In distributed planning for distributed plans multiple agents formulate plans for themselves. Each agent tries to achieve its individual goals. The agents have to ensure that the overall plan can be executed without conflict. Agents should preferably help each other. The biggest challenge is to identify and resolve conflicts. There exist a centralized plan coordination approach, where one agent plans the planning.

When merging plans interactions between pairs of actions of different agents have to be analyzed. If the actions are independent actions there is no conflict. If there exist dependent actions the execution order has to be switched. When actions conflict these actions have to be restricted or suspended. When solving conflicts the most important question is which agent should revise its plan. First all agents have to exchange the description of their options. The agent with most other options could then change its plan. Or the agent which has the least effort has to change its plan. Or the agent has to change its plan when the plan change has the least negative influence on other agents.

Plan synchronization is done by messages as signals or in a scheduled approach with time frames.

4.1 Iterative Plan Formation

In Iterative Plan Formation every agent constructs a set of feasible plans to reach its goal. In the first step each agent proposes a single action to all other agents.

All proposed actions will be rated by all agents. The best rated action will be chosen. Based on the chosen action every agent refines its subset. The proposing process is repeated until no further changes are needed.

4.2 Pre-/Post-Planning

In the Pre-Planning phase planning can be optimized by trying to detect and avoid undesirable states of the world. If undesirable states are detected one has to backtrack from there and restrict the actions which led to these states. The challenge is here to find restrictions without restricting to much. When building a plan action should be preferred from which most agents benefit, even if these actions are not directly relevant to the goals of the agent. When decomposing tasks overloading of critical resources should be avoided. Tasks should be assigned with matching capabilities without wasting to much resources. Agents with a wide view should assign tasks. To ensure coherence assign overlapping responsibilities. If tasks are highly interdependent assign them to agents in proximity.

In the Post-Planning phase it is important to assure that the system works, even if agents fail in their tasks. Each agent has to formulate alternative plans and to monitor its own plan execution. If it fails by executing a task, the agent has to stop all progress. Then it has to plan, coordinate and execute the plan again. Problems should be addressed at local level, so there is no need for coordination with others. To complete urgent tasks tasks can be re-assigned if necessary.

References

1. Durfee, E.H.: Distributed problem solving and planning. Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence pp. 121–164 (1999), http://www.sci.brooklyn.cuny.edu/~parsons/courses/716-spring-2010/papers/durfee-dps.pdf