COORDINATION IN OPEN AND UNSTRUCTURED INTELLIGENT AGENT SOCIETIES
Using Distributed Planners on Top of a Semantic Overlay Network

António Luís Lopes and Luís Miguel Botelho
Instituto das Telecomunicações, Av. Rovisco Pais, 1, Lisboa 1049-001, Portugal

Keywords: Multi-agent coordination, Planning algorithms, Graphplan, Peer-to-peer networks, Semantic overlay networks.

Abstract: Collaborative environments, where multiple heterogeneous agents (managing several resources) can cooperate in pursuing common and individual goals, are a step forward in creating real-world agent societies. However, current research in agent negotiation and in service coordination is still not enough for building such an agent-based society, capable of jointly solving complex planning problems and still achieve overall good performance. Most often, current work relies on either some centralised component or pre-defined social structure, which can compromise the system in terms of scalability, openness and robustness, and fails to address general problems. By using efficient network search algorithms and network evolution techniques it is possible to build and maintain a semantic overlay network from a totally unstructured distributed network, which in turn will simplify and optimize the distributed planning process amongst heterogeneous agents. We developed distributed versions of well-known planners that operate on top of the referred semantic overlay network and through a set of tests (using different scenarios) we were able to determine which is the best algorithm.

1 INTRODUCTION

Our main goal is to develop a robust agent architecture that enables agents to freely participate in distributed problem solving in open unstructured agent societies. In such societies, agents receiving requests from other agents are capable of using their own capabilities to handle the part of the problem for which they have competence and resources, and distribute the partially solved problem to other agents that can possibly provide further contributions. This work combines artificial intelligence, distributed problem solving and peer-to-peer computing to address its two main challenges: allow agents to partially contribute to complex problems and to efficiently delegate unsolved sub problems to other agents over unstructured decentralised networks.

Cooperating in distributed problem solving requires agents to be able to discover other agents to delegate the sub problems for which they cannot contribute. In previous research (Lopes and Botelho, 2008) we have shown that it is possible to improve the resource coordination process in multi agent based peer-to-peer networks by building, maintaining and using a powerful semantic overlay network (Cre spo and Garcia-Molina, 2005) that is dynamically learnt and updated by the discovery mechanism itself. The discovery and self-organisation process, in which agents establish semantic connections amongst them, thus fuelling the semantic overlay network, is first carried out by using efficient and robust search mechanisms and network evolution techniques (see (Lopes and Botelho, 2008) for details). Once the semantic overlay network is built, besides using it to easily locate resources, agents can use it to perform more complex tasks such as planning a solution for a specific problem.

In this paper we focus on the planning stage, by analysing and presenting a set of distributed planning algorithms that can be used, on top of the semantic overlay network, to solve planning problems in cooperative environments. Section 2 introduces the problem and presents related work. Section 3 describes our approach and in section 4 we discuss tests and results. Finally, section 5 concludes the paper.
2 MOTIVATION

A typical planner has been seen as a producer of action sequences that requires the following inputs: (i) an initial state of the world; (ii) a description of a goal state; and (iii) a set of possible actions, which can be used in the generated plan to lead the agent from the initial state to the goal state. As the need for more processing power (for increasingly complex problems) arose and the inadequacy of centralised approaches became evident, researchers turned to Distributed Problem Solving. However, taking advantage of the decentralised control of such distributed environments requires that coordination mechanisms exist that are able to avoid conflicts that arise from the concurrent interactions of agents, which otherwise would result in a turmoil.

The coordination process can be undertaken at different times and situations, depending on what is suitable for the specific domain to which it applies. In spite of the variety of approaches, domains and contexts, we have concluded that they have the same kind of limitations. Whether they rely on some sort of centralised component or on a pre-defined structure/knowledge that rules the activity of all entities in the environment, these approaches show signs of compromised scalability and robustness.

Most task refinement and task allocation approaches rely on centralised components, such as blackboards (Wellman, 1993), tables of capabilities (Fung and Chen, 2005) or broker like auctioneers (Wellman et al., 2001). An obvious limitation of these approaches in very large networks is its non scalability. Other approaches (de Weerdt et al., 2007) propose the use of social structures to govern the task allocation process, but again these have to be provided a priori by some human user.

The same limitations apply to planning coordination. Whether coordination is performed before, during or after planning, most approaches rely on some pre defined structure or some centralised component to govern the activity of the agents in the environment. Social laws (Shoham and Tennenholtz, 1992) and other organizational based approaches (Abdallah and Lesser, 2004) (Gaston and Desjardins, 2005) are examples of systems where sets of rules or organizational structures are imposed to the agents societies. Alternative approaches (Cox et al., 2003) use fully connected networks to allow agents to know which and when agents should be contacted. However, this kind of network topology is very difficult to manage in large dynamic networks where high churn rates (the rate at which agents enter or leave a network) make the approach non scalable. Centralisation is often the choice for most domains, whether it is by allowing agents to coordinate themselves through an advertising blackboard (de Weerdt and Van Der Krogt, 2002), or by using specialised central agents to perform post-planning coordination (Cox and Durfee, 2005).

3 TECHNICAL APPROACH

By using a matchmaking process that analyses each agents operators and establishes links between inputs/outputs and pre-conditions/effects of the operations that can be done by other agents (see details in (Lopes and Botelho, 2008)), we dynamically build a powerful Semantic Overlay Network. This abstract layer subsumes the discovery process of the coordination environment by helping agents find each other based on their semantic relationships. However, for an agent to create a plan to solve a specific problem, it still needs a planning algorithm that is capable of dealing with partial knowledge of the domain operators and deriving the potential contributions that can be done by the agent to the referred problem. Our work focuses on finding the appropriate planning algorithm for this distributed environment.

We have studied several algorithms and decided to use the well-known Graphplan planner (Blum and Furst, 1997) because it is one of the best planning algorithms and has been used as the basis for other even more efficient planning algorithms. We have created different versions of the planner so we could test its application to different distributed problems. In section 3.1, we present our distributed version of the Graphplan algorithm. Section 3.2 describes an alternative version of the distributed Graphplan algorithm that previously builds an operators-graph through means-ends analysis.

3.1 Distributed Graphplan

In a centralised version of the Graphplan algorithm an agent has full knowledge of the available operators. However, in a distributed setting, each agent may only have knowledge of its own operators and of those of selected neighbours or, in the case of the Semantic Overlay Network, of agents that are semantically-linked to it. So, we modified the algorithm to allow partial contributions to the creation of the planning graph. The process is carried out as follows:

- The agent receiving a request (which includes a description of the initial state and the goal state) will analyse each level and determine which of the operators (that it knows) can contribute to the
current planning-graph (if the agent is the first receiving the request, it has to create the first proposition level, which will include all propositions of the initial state) and adds them accordingly:

- The agent further analyses open propositions (for which it was unable to contribute) and determines (using the Semantic Overlay Network) which are the most adequate agents to receive this partially-complete planning-graph;
- Each agent receiving the planning-graph will execute these same steps up to a point where a level in the graph is reached where all goal propositions exist and none of which are mutex;
- Once a planning-graph reaches this desired stage, the agent holding the planning-graph at that time executes the backward search that will find a solution plan – the agent can also request the assistance of other agents in the backward search, but each agent will use a different heuristic in the process;

The termination of this process in a distributed environment is not trivial. In the centralised version an agent can simply rely on the level-off property of Graphplan 1 to conclude that the problem is impossible to solve. For an agent with only partial knowledge of the world, it is impossible to know if a levelled-off graph means that the problem is impossible or that the agent simply does not have enough knowledge to complete it. This could lead to a near-infinite process of forwarding partially solved problems between agents. To avoid this situation, we use a similar mechanism as the one used in peer-to-peer search algorithms, where a time-to-live setting is used to specify the allowed number of forwards that can be done with a single request.

3.2 Distributed Graphplan with Operators-Graph

In most domains, the size of the goal state is a lot smaller than the initial state, which means that some of the propositions contained in the initial state may be completely irrelevant to reach the goal state. As most forward-based planners, Graphplan suffers from the problem of distraction, where the planner considers all propositions in the initial state even if they will not help reach a solution plan. In fact, these unnecessary propositions can be very time-consuming, thus degrading the performance of the planner. Therefore, they should be avoided.

To cope with this problem, we have used a similar approach to (Kambhampati et al., 1997). We use means-ends analysis in the Graphplan algorithm, by first producing an operators-graph (Smith and Peot, 1996) using a backward-chaining process starting from the goal state. Since it only considers the propositions in the goal state, the operators-graph will produce a graph with only relevant actions.

This planner uses a similar process to the one used in the generation of the planning-graph but in a different direction. It finds operators that can contribute to propositions in the goal state and the pre-conditions of those operators become the new goal propositions. The planner proceeds with this process until it reaches a level in which all propositions are contained in the initial state. After the graph has been generated, the normal forward generation of the planning-graph can take place, except this time it will only consider the operators that are contained in the operators-graph, thus significantly reducing the size of the graph and the number of calculations. The drawback is, obviously, the overhead introduced by the generation of the operators-graph.

4 TESTS AND RESULTS

In order to test the distributed Graphplan and the operators-graph-based version of Graphplan algorithms, we deployed two different testing scenarios but due to space limitations we will only show the results for the Rescue Agents scenario. In this scenario, agents represent entities that participate in a rescue operation after the occurrence of a natural disaster, where they have to perform operations such as clearing roads, putting out fires and providing assistance to injured people. This scenario is characterized as having a small number of different entities but with a high degree of complexity due to the high level of interaction/coordination that is needed between the agents.

We have performed a set of preliminary tests for this scenario using increasingly complex variations (different number of injured people) on both algorithms. As depicted in Figure 1, the results show that the operators-graph-based version of the distributed Graphplan algorithm is more scalable than the distributed Graphplan algorithm. This is strongly linked to the fact that for more complex or large problems, the means-ends analysis is effective in reducing the search scope of the planner, in spite of the introduced overhead.

---

1 When two subsequent levels are equal and non-mutex goal propositions have been reached yet
5 CONCLUSIONS AND FUTURE WORK

In this paper we have described the approach taken in a cooperative planning environment, where we have deployed a distributed network of problem solving agents by using a Semantic Overlay Network and distributed Graphplan-based algorithms. Preliminary results show that the approach can be considered scalable and efficient. The results are still preliminary and we intend to perform a more thorough analysis of the testing scenarios and include other scenarios to address the robustness of the approach in a large variety of problems.

ACKNOWLEDGEMENTS

This work has been supported in part by the Portuguese Foundation for Science and Technology under the scholarship grant SFRH/BD/27533/2006. The authors would also like to thank the support of the Lisbon University Institute and the Instituto de Telecomunicações (IT).

REFERENCES


