

Dynamic Cooperation Using Resource Based Planning

Utku Erdoğdu
Department of Computer Engineering
Midde East Technical University
Ankara, Turkey
utku@ceng.metu.edu.tr

Faruk Polat
Department of Computer Engineering
Midde East Technical University
Ankara, Turkey
polat@ceng.metu.edu.tr

ABSTRACT

In multi-agent planning, cooperation is a crucial interaction to reduce plan costs and complexities in order to satisfy non-conflicting goals. However, it is not generally natural to sacrifice from the autonomy of agents by letting agents share all of their local plans. Resource based planning is a planning framework similar to logic based planning based on resources and skills that map to states and actions of world state. Resource based planning is very functional in processing plan elements to find cooperation schemas. In this work we extended the simple resource based plan coordination method proposed by Tonino, et al [2]. Our framework provides not only a basis for modifying plan elements, but also functionality for manipulating plans dynamically. A negotiation algorithm using a resource based perspective provides the necessary ground for forming cooperation among agents. We illustrated the effectiveness of the proposed method on a modified postman domain, where agents try to minimize the cost of delivering the letters they are assigned.

Categories and Subject Descriptors

I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search—*plan execution, formation, and generation*; I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Theory, Design

Keywords

Applications of Autonomous Agents and Multi-agent Systems, Multi-agent Planning, Cooperation and coordination Among Agents

1. INTRODUCTION

Planning is one of the classical, but popular research areas in Artificial Intelligence. If we consider a simple autonomous

agent, we desire it to have its own judgement and decision mechanism by the definition of autonomy. Thus a simple autonomous agent should construct her own plan regardless of the framework used or the environment it acts in. Classical AI planners are designed for this sole purpose. However in multi-agent domains individual decisions are rarely enough for producing optimal plans for satisfying all the goals. The global optimum is nearly impossible with pure autonomy, so is individual optimum. Therefore, agents need to cooperate to generate the best multi-agent plan through sharing tentative solutions, exchanging subgoals, or having other agents to satisfy their goals.

The simplest approach to generate a multiagent plan is generating a single plan for all agents. The plan could be generated by a central authority. This perspective might be applicable for agent colonies where a vast number of agents have a single goal, but for autonomous agents with different, possibly conflicting goals, this approach is inadequate.

If the autonomy of the agents is vital for the application, a central planning unit is inappropriate regardless of the success of the plans it produces. It would be more effective if agents generate their plans independently and interact with each other during the execution to revise their plans in case of contingencies, or new goals.

Tonino et al. developed a resource based multi-agent planning method [2] similar to traditional planning. However the representation schema used allows sub-goals and plan elements to be processed as resources that can be exchanged, removed from and inserted into plans.

In this work, we extend the resource based planning framework to handle dynamic goals and complex interaction requirements that exist in many real-life problems. We applied the proposed approach to the modified postman problem where agents (postmen) are required to deliver a predetermined set of letters to some destinations. The domain is dynamic in that, besides the initial letters, agents are required to collect and deliver letters at specified locations in real-time. In order to keep their individual plan costs or total global cost nearly optimal, the agents should cooperate upon new goals by undertaking/exchanging subgoals. We propose a new cooperation mechanism independent of domain properties to handle real-time goals.

2. RESOURCE BASED PLAN FRAMEWORK

The resource based plan framework uses resources and skills instead of states and operators used by traditional planners. A resource is a real or abstract object used or produced by actions of agents. A skill is simply an action of

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS'06 May 8–12 2006, Hakodate, Hokkaido, Japan.
Copyright 2006 ACM 1-59593-303-4/06/0005 ...\$5.00.

an agent. A skill instance in a plan represents an application of a skill. Application of a skill consumes some resources and produces new resources. Consumed and produced resources might be identical.

Using these two concepts a plan can be represented by a bipartite directed graph. Resources are represented by resource nodes and skills are represented by skill nodes. Initial world state and the goal state are represented as two sets of resources. Initial resources are consumed by a plan in order to produce goal resources. Figure 1 shows an example plan graph where circles denote the resource nodes, squares denote the skill nodes and directed arcs denote consumption or production of a resource by a skill.

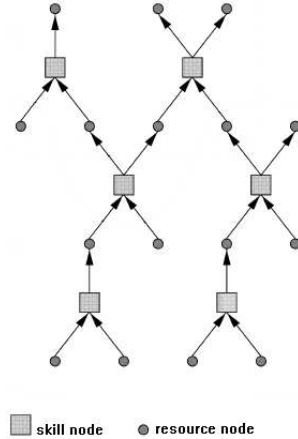


Figure 1: An example single agent Resource Based Plan Graph.

The graph representation is used for determining possibilities of cooperation between agents. Note that each skill produces resources. Some of these resources are goal resources, some are consumed by other skills and some are just side effects of the plan. We can safely remove a skill from a plan graph if we can find a way to generate the same resources that this skill produces. Tonino discusses a plan revision schema based on this principle[2].

They provide a simple protocol for compensation. This framework proves to be successful in establishing cooperation and generating a collaborative plan where total and individual costs of the plan are reduced as much as possible. The framework is applied to a transportation problem where trucks carry goods between cities. In the collaborative plans produced, trucks share routes and goods they carry, thus costs are reduced to minimum for the simple route topologies. However there are some issues that framework is not sufficient.

Although the framework is independent of the problem domain, it is not effective for domains having large state space. The selected application domain (transportation example) possesses a dense and small graph and hence agents produce many side effect resources which can be used by other agents. In most domains, producing extra resources may be cheap, but not trivial. A mechanism to enforce agents producing side effect resources is necessary.

Secondly, the framework mainly transfers the side effect resources of one agent to other relevant agents. There is also

a need to finely tune this resource exchange. The simplest form of a collaboration is to transfer a useless resource from the acceptor to the proposer. There are domains in which no useless resource exists and thus it is not possible to simply transfer the resources, which increases the plan cost. The collaboration should allow the exchange of resources whose cost can be assessed by each of the agents participating in the exchange.

Our framework is designed to provide solution to these problems and improve the original framework in technical and theoretical levels. The details of our framework are given in the following sections.

3. DYNAMIC RESOURCE BASED COOPERATION FRAMEWORK

The postman problem is one of the classical example domains for multi-agent research[1, 3]. We changed the classical postman problem as follows. First, agents are required to collect letters from some locations and deliver them to the desired locations. A letter is represented with a source node (possibly the post office) and a destination node. Moreover, the final destination of agents is changed from post office to any node in the graph. It is shown in [1] that if the final destination node is post office, i.e., delivery of all letters is a cycle, then there may be trivial interaction schemas independent of the letter set. By changing the final destination to be any node, we prevent such trivial interactions.

When a new goal is assigned to an agent, possible choices are

- The goal can be already satisfied by the current plan,
- Another agent can satisfy the goal and
- The agent can modify her plan to satisfy the goal.

It is apparent that a rational agent selects one of the feasible alternatives in the order given above. If the new goal is already satisfied by the current plan without additional cost, then the plan is slightly revised to satisfy this new goal. Otherwise the agent computes the cost of modifying the current plan to satisfy the new goal. If the modification is costly, then the agent interacts with others to see whether she can benefit from any cooperation, or not. The cooperation alternative which does not bring additional cost is selected and agents involved in the cooperation revise their plans accordingly.

When an agent asks for another agent to handle one of her goals, the second agent attempts to handle the whole goal alone if it does not bring an extra cost. If this request cannot be satisfied, the proposer agent breaks the goal into subgoals and repeats the same process. In our framework, we have realized only one level decomposition but the this can further be extended to any level easily.

The acceptor agent could simply accept to satisfy the request, deny the request or accept to satisfy only some of the subgoals of the request. Moreover, the agent may satisfy the request because of two reasons; her plan might already satisfy the request with trivial modification or the requested goal might be inserted into her plan with modification which does not increase agent's cost. This structural decomposition of goals in decision making leads to a more interesting and fine tuned cooperation mechanism as we will illustrate in examples.

The acceptor might accept the request with the condition that the proposer handles one or some of the goals of the acceptor. To do this, agents could adopt their plans to partition the total goal set into meaningful subsets. In our system, an acceptor might request from a proposer to deliver one of her letters that contributes most to the cost of her plan.

In the previous section, we mentioned that there is need for a mechanism that is capable of producing side effect resources. We make use of a credibility metric in order to have this effect. Agents have credibility on each other. In case the proposer requests a new task from the acceptor, the acceptor will accept to satisfy this request if the cost of the modification in the acceptors' plan is less than or equal to the credibility of the proposer on the acceptor.

These are the salient features added into the original resource based framework [2]. The following sections describe and exemplify the proposed framework taking into consideration the domain properties.

4. APPLICATION OF DYNAMIC RESOURCE BASED PLANNING TO POSTMAN DOMAIN

Our framework starts with an initial planning phase where each agent constructs a plan for collecting and delivering the letters assigned to her. These plans are constructed independently and hence they do not take into consideration potential cooperations.

The initial plan computation algorithm has exponential complexity in terms of number of letters. The calculation of all pair shortest path and constructing the final path are all polynomial. However, an exhaustive search is necessary to find a minimum hamiltonian path on the constructed graph, which takes exponential time in terms of the number of nodes in the temporal ordering graph which is roughly equal to two times the number of letters.

In the plan graph each skill has a cost. Delivery and collection skills have unit cost, while cost of going from one node to another is naturally the weight of the edge between two nodes. In our framework we used four resources and five skills. Resources are:

- **Connected-Resource** that represents connectivity of graph nodes.
- **Location-Resource** that represents location of an agent.
- **Have-Letter-Resource** that represents having a letter.
- **Delivered-Letter** that represents having delivered a letter.

Skills are:

- **Go-Skill** that represents going from one node to another.
- **Collect-Skill** that represents collecting a letter.
- **Deliver-Skill** that represents delivering a letter.
- **Exchange-From-Skill** that represents getting a letter at rendezvous.

- **Exchange-To-Skill** that represents giving a letter at rendezvous.

Agents have initial credibility on each other. This credibility corresponds to utility concept as we discussed previously. We used a simple scalar credibility measure and set it to an empirical value. If an agent satisfies the request of another agent, she lowers other agent's credibility. The reduction amount is simply the cost of the request for the acceptor agent. The credibility value is chosen such that a given number of trivial deliveries could be handled.

In our work we used a negotiation protocol where agents propose bids for requests. The bids may also include counter proposals and counter requests. By negotiating, agents try to undertake requests for tasks if it is profitable for them. Since negotiation involves counter requests and the agent's utility is based on credibility measure, there are several possible outcomes of the negotiation, some of them increasing cost of a party slightly. For example if the whole task cannot be accepted by any other agent, the agent seeks alternatives to decompose the goal into subgoals.

If the agents agree on cooperation through exchanging letters, a rendezvous should be arranged to transfer the letter. We used an explicit rendezvous concept to build parts of the plan graph to denote that the agents meet at an agreed location and exchange the letter. The negotiation phase includes the arrangement of this rendezvous.

5. CONCLUSION AND FUTURE WORK

We attempted to extend the resource based planning framework of Tonino et.al. Our primary motivation was to extend this framework to allow contingency planning especially for realistic real life problems. The most important feature of this work is to use negotiation to revise plans to handle new goals. Refining the plans in real-time is vital if agents are faced with new goals in real-time. The framework is also designed to be robust, thus more complex interaction schemas can be introduced. For example in an auction based domain, agents might prefer to participate in auctions rather than negotiation. Different interaction mechanisms would result in interesting variations of the framework.

The proposed framework uses a novel representation and modeling schemas for effectively capturing and using distinguished features of more realistic domains. We used a simple utility measure to realize credibility of agents which could be generalized further. This simple measure is sufficiently functional for our example domain. If the agent interactions are more complex, each agent should have better models of others.

6. REFERENCES

- [1] J. S. Rosenschein and G. Zlotkin. Designing conventions for automated negotiation. *AI Mag.*, 15(3):29–46, 1994.
- [2] J. Tonino, A. Bos, M. M. de Weerd, and C. Witteveen. Plan coordination by revision in collective agent-based systems. *Artificial Intelligence*, 142(2):121–145, 2002.
- [3] H. Ulusoy. Forming coalitions through negotiation in multi-agent environments. Master's thesis, Middle East Technical University, Department Of Computer Engineering, December 1999.