

# A Decision Making Procedure for Collaborative Planning

Timothy W. Rauenbusch and Barbara J. Grosz  
Division of Engineering and Applied Sciences  
Harvard University  
Cambridge, MA 02138 USA  
{tim,grosz}@eecs.harvard.edu

## ABSTRACT

A team of agents planning to perform a complex task make a number of interrelated decisions as they determine precisely how that complex task will be performed. The decision set includes the choice of recipes and parameters as well as the determination of responsibilities for each agent. This work formally defines a search problem with search operators that correspond to the team planning decisions. It defines rationality for a collaborative team and ensures that a team will abandon performance of a complex task if it is ultimately in its economic best interest to do so. The model respects the constraints on mental states specified by the SharedPlans theory of collaboration.

## Categories and Subject Descriptors

I.2.11 [Computing Methodologies]: Artificial Intelligence — *Multiagent Systems*

## General Terms

Algorithms, Languages, Theory

## 1. INTRODUCTION

Agents working together as a team can accomplish tasks that no agent on the team is able to perform independently. The act of forming a team to accomplish a task is necessary but not sufficient to ensure success. Not all teams are equally effective. For instance, consider two teams of volunteers building a wall of sand bags to fortify a sea wall before a flood. One team can unload 30 bags of sand per minute when working together by passing bags from hand to hand. The other team has each volunteer running back and forth from the truck to the shore unloading a bag at a time, so it unloads only five bags per minute. Each team has the same goal of averting a disaster, but the first team is more likely to achieve its goal because it has been smarter about planning how its members will work together.

The problem faced by a team  $G$  that has agreed to perform action  $A$  is both distributed and complex. It is distributed because the information relevant to making an effective decision is not possessed by any individual, and it is not practical to transfer all relevant information to one central point. It is complex because there is a set of three broad types of interrelated decisions that grows exponentially with the complexity of  $A$ . The three types of decisions are: (a) recipe

selection, (b) parameter binding and (c) assignment. Recipe selection decisions relate to choosing one of the possible ways of performing  $A$ . Parameter binding is the process of further refining an act type taken from a recipe into a concrete action to be performed (e.g., specifying when action execution will begin). Assignment requires a subset of the agents on the team to be assigned responsibility for each constituent action. Complexity results because the decisions are interrelated and because each subact of the recipe selected for  $A$  may be further decomposed, opening up new sets of decisions that need to be made by its responsible (sub)team. Furthermore, these new sets of decisions for subacts are interrelated with the decisions for the parent action  $A$ .

The aim of this work is to provide an algorithm for making the three types of interrelated decisions by recasting the problem as a search problem. To do so, it first defines proposal trees, a new data structure that is similar to Plan Trees [3], but specifically built to represent the team decision making problem. It then defines a search space and provides a precise definition for three search operators: selecting a recipe, binding parameters, and assigning responsible agents.

The objective function for the search is based on an extension of economic individual rationality to team rationality. One major benefit of team rationality is that it requires that a team abandon its plan to perform an action if the costs of performing that action prove to be too high. The costs of actions that have been performed are treated in a rational way when planning present and future actions, based on the economic notion of a sunk cost.

## 2. PROPOSAL TREES

Plan Trees [3] explicitly represent the choices that a group of agents have made when working on a SharedPlan [1, 2]. They are used to summarize the mental states (e.g., goals and intentions) of each agent in a group that is planning to perform a complex action together. Proposal trees have a similar structure to Plan Trees but do not contain planning actions.

Each node  $N$  of a proposal tree contains a triple  $\langle A, C, G \rangle$  where  $A$  is an action,  $C$  is a set of constraints on the parameters of  $A$ , and  $G$  is a set of one or more agents. Each node has zero or more child nodes; the children of a node correspond to the actions in a recipe for  $A$ .

A *fully specified basic node* in a proposal tree is a node  $\langle A, C, G \rangle$  such that  $A$  is an action that is basic and fully specified and  $G$  is a singleton that corresponds to a single agent.

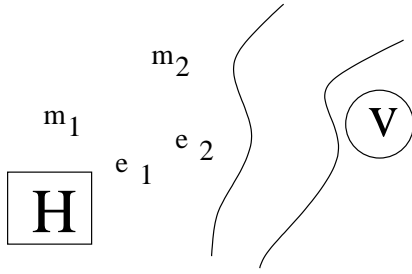


Figure 1: Four-agent rescue domain

A *proposal tree* for  $A$ , denoted  $PT_A$ , is a proposal tree whose root node contains the action  $A$ . A *full proposal tree* (FPT) is a proposal tree in which all of its leaf nodes are fully specified basic nodes. A *partial proposal tree* (PPT) is a proposal tree that is not a FPT. A *minimal proposal tree* for  $A$  is a proposal tree for  $A$  that contains a single node.

### 3. SEARCH PROBLEM

A multi-agent planning problem is defined by a triple

$$\langle A, C, G \rangle$$

where  $A$  is an action to be performed,  $C$  is a set of constraints among those actions, and  $G$  is the set of agents that form a group that is responsible for performing the action.

The search space is represented as a tree. Each node  $S$  of the search tree contains a proposal tree. The root node of the search tree contains a minimal proposal tree. The child nodes of each node  $S$  of the search tree contain the proposal tree attained by applying an operator to the proposal tree contained by  $S$ . Each leaf node of the search tree contains an FPT.

The operators SELECTREC, ADDAGENT, and BINDPARAMETER transform proposal trees. Successors of each search state correspond to search nodes containing those transformed proposal trees.

A team of rational agent should perform action  $A$  if and only if the benefits of performing  $A$  exceed the costs of performing it.  $V(A)$  denotes the benefit, or intrinsic value of performance of  $A$ . The cost of performing an action is a function of the resources consumed in performing it. The cost to agent  $G$  of performing basic action  $B$  is given by  $C_G(B)$ . The function GETCOST shown in returns the cost to a group of agents of performing a the tasks specified in a proposal tree.

The *surplus*  $\pi = V(A) - \text{GETCOST}(PT_A)$  quantifies the net benefit to a team that would perform  $A$  according to proposal tree  $PT_A$ . A *rational team* is one that will not perform  $A$  if  $\pi < 0$ .

### 4. VICTIM RESCUE

The rescue domain is illustrated in Figure 1. Four agents (two medics and two engineers) labeled  $e_1, e_2, m_1$  and  $m_2$  and a hospital (marked ‘H’) are located on the west bank of a river. A victim (labeled ‘v’) lies on the east bank and awaits medical attention and transportation to the hospital. The agents form a team with the goal of rescuing the victim and transporting her to the hospital.

The search for a plan for rescuing the victim begins with its root node which contains the minimal proposal tree,

which contains the single action `rescue-victim`, the group of four agents, and perhaps a constraint that the action must be completed by 4pm.

The SELECTREC operator is chosen to expand the search state. It would choose the one node of the proposal tree labeled `rescue-victim` which is the only leaf. Three nodes that represent constituent acts `move-to-loc`, `assess-victim`, and `transport-victim` would be created under the `rescue-victim` node. The proposal tree of these four nodes would form a child node in the search tree of the root node.

Construction of the search tree continues in this way. Any of the three operators may expand the search tree at the second level. Even in a small problem like the rescue domain that the search space can be large because of the large number of decisions that must be made.

### 5. RELATED WORK

Much research in the intersection of AI and economics has related to strategic interactions among agents [4; 5, inter alia]. Such work makes the assumption of individual rationality of agents — that an agent will make the decision that is best for itself. Team rationality does not explicitly model an agent’s individual decision of whether to join a team. The model assumes that agents in a team are committed to performance of the group action, unless it turns out that it is in the team’s best interest to abandon performance of the action.

### 6. CURRENT WORK

Our current work is on developing algorithms and heuristics for traversing the search tree. The main objective is exploring the tradeoff between the amount of communication among agents and the quality of the outcome of the search. We have developed structures to encode cost and value information that an agent communicates. We are investigating ways to use this knowledge to prune the search space and focus the search on parts of the tree that are most likely to yield outcomes with high surplus.

### 7. REFERENCES

- [1] Barbara Grosz and Sarit Kraus. Collaborative plans for complex group activities. *Artificial Intelligence Journal*, 86(2):269–357, 1996.
- [2] Barbara Grosz and Sarit Kraus. The evolution of SharedPlans. In Michael Wooldridge and Anand Rao, editors, *Foundations and Theories of Rational Agency*, number 14 in Applied Logic Series, pages 227–262. Kluwer Academic Publishers, The Netherlands, 1999.
- [3] Luke Hunsberger. Making shared plans more concise and easier to reason about. In Jörg Müller, Munindar P. Singh, and Anand S. Rao, editors, *Proceedings of the 5th International Workshop on Intelligent Agents V : Agent Theories, Architectures, and Languages (ATAL-98)*, volume 1555, pages 81–98. Springer-Verlag: Heidelberg, Germany, 1999.
- [4] Sarit Kraus. *Strategic Negotiation in Multi-Agent Environments*. MIT Press, Cambridge, MA, 2001.
- [5] Tuomas W. Sandholm. Distributed rational decision making. In Weiss, editor, *Multiagent Systems*, chapter 5. MIT Press, 1999.