# The Effects of Cooperation on Multiagent Search in Task-Oriented Domains

José M. Vidal Computer Science and Engineering University of South Carolina Columbia, SC 29208 vidal@sc.edu

## ABSTRACT

We study the benefits of teaming and selflessness when using multiagent search to solve task-oriented problems. We introduce a formal framework for multiagent search and study a specific instantiation of it: task-oriented domains. Our experiments show that better allocations are found when the dynamics of the multiagent system lie between order and chaos, that neither absolute selfishness nor absolute selflessness result in better allocations, and that the formation of small teams usually leads to better allocations, among other results.

## **Categories and Subject Descriptors**

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

#### **General Terms**

Multiagent Search. Task-Oriented Domain.

## 1. INTRODUCTION

Multiagent systems are especially suited to solve problems in which individual decision-makers with localized information are able to affect their local state in the hopes that the system will eventually reach a global state of either optimal or at least satisfactory utility. Classic example scenarios include distributed sensor monitoring, distributed task allocation , and coalition formation. In these problems, each agent perceives some part of the global state and takes actions that modify some part of this state. The agents act to maximize some local utility function. The function's details, e.g., how "selfish" it is, as well as the interaction protocols among the agents are left to the system designer. The designer must engineer these so that locally optimal decisions

Copyright 2002 ACM 1-58113-480-0/02/0007 ..\$5.00

give rise to the best possible global state. We refer to these types of problems as instances of a more general **multiagent search** problem.

In this paper we first introduce a formal framework for multiagent search that, we can show, forms a superset of task-oriented domain, coalition formation, distributed constraint satisfaction, and Kauffman's NK landscapes. The grouping of all these different problems into one framework allows us to leverage results from one domain and use them in another. As an example this power in Section ?? we present our results on the effectiveness of cooperation via team formation and selflessness in task-oriented domains. This approach was inspired by the successful use of "patches" in the search of NK landscapes in a two-dimensional grid instantiation.

From our experiments we were able to derive several interesting results. We show how agents that form teams and engage in limited forms of selfless behavior find solutions that are of a higher global utility. We show that the best solutions are found in systems that exhibit dynamics that are at the phase transition between order and chaos. These results lead us to suggest that further study should be devoted to the study of coordination protocols that do not converge to a stable solution but instead continue to change. We believe that such protocols shall result in better (from a global perspective) emergent behaviors in multiagent systems. We also present several specific findings such as the fact that neither absolute selfishness nor absolute selflessness result in better allocations, and the fact that the formation of small teams usually leads to better allocations.

#### 2. MULTIAGENT SEARCH FRAMEWORK

In this section we present a formal framework for describing multiagent search problems. These problems are characterized by a global state composed of the aggregation of the value of many local variables. Each agent perceives the values some of the variables, modifies the value of some of the variables, and receives a utility that depends on the value of some of the variables. By limiting which variables the agents perceive, modify, or derive utility from, we can instantiate various well-known multiagent problem domains.

The global state is denoted by S. It is formed by the union of a fixed set of local variables  $\{s_1, s_2, \ldots, s_{|S|}\}$ , each one with a finite domain. The set of agents is  $A \equiv \{1, 2, \ldots, n\}$ , where n is the number of agents. Each agent  $i \in A$  has an utility function  $u_i : d_i \to \mathbb{R}$  that provides a mapping between a subset of state variables  $d_i \subseteq S$  and a real number. An

 $<sup>^{*}\</sup>mbox{This}$  work was funded by an NSF CAREER grant number 0092593.

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'02, July 15-19, 2002, Bologna, Italy.

agent's relationship with its environment is captured by the set of local variables on which its utility depends, the set of local variables whose value it modifies, and the set of local variables whose value it views. Specifically, for agent i we define  $d_i$  to be the set of variables upon which its utility depends,  $m_i$  is the set of variables which it can modify, and  $v_i$  is the set of variables it can view. The agents modify the state of the variables in their respective  $m_i$  sets but only if these modifications satisfy the constraints imposed by  $P: S \times S \rightarrow \{0, 1\}$ . For example, agent i can only change S into S' if p(S, S') = 1, where  $p \in P$ , and the state variables it modifies are in  $m_i$ . If a constraint function evaluates to 0 it means that the particular state change is not allowed.

We now define a multiagent search problem as the tuple  $\{A, S, U, D, M, V, P\}$  where  $A \equiv \{1, 2, \ldots, n\}$  is the set of agents,  $G \equiv \{S_1, S_2, \ldots, S_{|G|}\}$  is the set of all possible global states such that  $S \in G$ , U is the set of all agent utility functions where  $u_i \in U$  and  $u_i : d_i \to \mathbb{R}$ ,  $d_i \in D$ ,  $m_i \in M$ ,  $v_i \in V$ , and P is the set of constraints, as defined above.

This formalization of multiagent search states the problem but does not provide a solution. The goal of an agentbased software engineer is to implement agent behaviors that will enable the quick discovery of the globally optimal solution. That is, the system should converge to  $s^* = \arg_{S \in G} \max \sum_{i \in A} u_i(S)$ . A common approach is the use of individual hill-climbing. In it, each agent modifies its local variables  $m_i$  to maximize its utility  $u_i$ . It is expected that doing so will also increase the sum of everyone's utility. Unfortunately, this approach usually leads to sub-optimal states. In Section ?? we extend this idea by allowing the formation of teams and the use of partially selfless agents and show the benefits of that approach.

#### 3. TOD PROBLEM SPECIFICATION

We set out to study the benefits of cooperation in a TOD problem, by randomly grouping agents into teams. The teams are non-overlapping and of a fixed and equal size. The team sizes vary from individual teams where each agent is a team to the grand team where all agents belong to the same team. Agents in a team take actions that maximize the team's utility. We define team(i) to be the set of agents in i's team, including i. We then define the utility that agent i receives in global state S as

$$\operatorname{teamUtil}(i, S) = \frac{1}{|\operatorname{team}(i)|} \sum_{j \in \operatorname{team}(i)} u_j(S).$$
(1)

Finally, we also vary the number of tasks that each agent can do. We limit the set of tasks an agent can do by modifying  $m_i$ . At one extreme every task can be done by only one agent, in which case the task allocation problem is trivial. At the other extreme all the agents are able to do all the tasks thereby expanding the size of the search space. As such, it is very time-consuming to find an optimal solution for this case.

In order to determine the effectiveness of team formation in TOD we developed a simulator that searches the space of possible states S. For each run we randomly generate a new cost function and new starting state. Each step in a run consists of first randomly selecting one agent. This agent then determines which is the best action it can take. The available actions to the agent are to either give one of its tasks to another agent or to take one task from another agent. The agent will consider all possibilities and choose the one with the highest team utility for the agent's team. Also, an agent can only give a task to or take a task from another agent if that agent's utility loss is no greater than the maximum loss L, a parameter which we vary from zero, for purely selfish agents, to one, for agents that are willing to take whatever deal is offered to them. That is, if the system is on state S then agent i will only accept a new state S' if willingToDo(i, S, S') is true, which we define as

willingToDo(i, S, S') = teamUtil(i, S) - teamUtil $(i, S') \ge L$ . (2)

Agents with a maximum loss of zero (L = 0) are not willing to accept any deal where their new team utility is less than their current team utility. These are the rational agents from [1]. On the other hand, agents with a maximum loss of one (L = 1) could be said to be completely selfless.

## 4. SUMMARY OF RESULTS

Our first experiments involve 16 agents and 32 tasks. For each experiment we changed the number of tasks that each agent can do. Within each experiment we varied the maximum loss parameter (L), as well as the number of teams allowed.

The experimental results showed several interesting results. We found that the best solution is usually attained with a maximum loss of between .4 and .6, that is, when the agents act somewhat selflessly. These parameter values allow the search to make more exploratory moves. These values seem to have similar effects to the temperature parameter in simulated annealing. We found that an even better predictor of the effectiveness of the multiagent search is the dynamics of the agents' behavior. Specifically, we found that the best global solution is always found when the systems' dynamics lie between the ordered and chaotic regimes. That is, as we vary the value of the parameters that represent the maximum loss, the number of teams, and the number of tasks that agents can do, the systems dynamics vary from ordered, where most of the runs quickly converge to some state, to chaotic, where none of the runs seems to ever converge. The best solutions were found for those cases where only a small percentage of the runs converge. We also found that small teams generally lead to better solutions and that teaming, in general, improves the quality of the result. Finally, we showed that neither complete selfishness nor selflessness are the best solution in almost all cases.

These results are important for the design of multiagent systems. Specifically, our results on the dynamics of multiagent systems seem to suggest that further research into multiagent coordination protocols should not concentrate on protocols that lead to a "clean" fixed solution but should instead study open protocols whose interactions might never end. Open-ended interaction protocols seem more likely to enable the system arrive at a better global solution. Of course, the computational and communications cost might make this a sub-optimal solutions. The final tradeoff would seem to be domain dependent.

#### 5. **REFERENCES**

 J. S. Rosenschein and G. Zlotkin. *Rules of Encounter*. The MIT Press, Cambridge, MA, 1994.