STRATEGIES OF COOPERATION IN DISTRIBUTED PROBLEM SOLVING†

Stephanie Cammarata, David McArthur, Randall Steeb

Rand Corporation
1700 Main Street
Santa Monica, California 90406
USA

ABSTRACT

Distributed Artificial Intelligence is concerned with problem solving in which groups solve tasks. In this paper we describe strategies of cooperation that groups require to solve shared tasks effectively. We discuss such strategies in the context of a specific group problem solving application: collision avoidance in air traffic control. Experimental findings with four distinct air-traffic control systems, each implementing a different cooperative strategy, are mentioned.

I. INTRODUCTION

Distributed Artificial Intelligence is concerned with problem solving in which groups solve tasks. Through systems like STRIPS [4], ABSTRIPS [9], BUILD [3], and NOAH [10], we now have some understanding of how a single agent can solve problems. Unfortunately, recent work suggests that the representations of knowledge [5], [l] and planning expertise [8] required of agents in distributed or group problem solving is quite different than that required of single agent problem solvers.

In this paper we focus on one particularly important but little understood topic: The kinds of *strategies of cooperation* that groups require to solve shared tasks effectively. We begin with a discussion of the difficulties facing distributed problem-solving groups. From this analysis we infer a set of requirements on the information-distribution and organizational policies such groups require. We then discuss a set of distributed problem solvers that we have developed in the domain of air traffic control. We concentrate on the particular cooperative strategies they embed, how they are implemented, and how they successfully overcome some of the obstacles that make it difficult to coordinate groups of agents.

II. DISTRIBUTED PROBLEM SOLVING DIFFICULTIES

To understand the difficulties facing groups solving problems it helps to note some important characteristics of distributed problems. The following traits are true of a wide variety of group situations:

- Most situations consist of a collection of agents, each with various *skills,* including sensing, communication (often over limited- bandwidth channels), planning, and acting.
- The group as a whole has a set of assigned *tasks.* As in single agent problem-solving situations, these tasks may need to be decomposed into subtasks, not all of which may be logically independent. The group must somehow assign subtasks to appropriate agents.
- Typically each agent has only *limittd knowledge.* An agent may be subject to several kinds of limitations: limited knowledge of the environment (e.g., because of restricted sensing horizons), limited knowledge of the tasks of the group, or limited knowledge of the intentions of other agents.
- There are often shared *limittd resources* with which each agent can attack tasks. For example, if the agents are in a blocksworld environment, the shared resources are the blocks out of which their constructions must be made.
- Agents typically have differing *appropriateness* for a given task. The appropriateness of an agent for a task is a function of how well the agent's skills match the expertise required to do the task, the extent to which its limited knowledge is adequate for the task, and current processing resources of the agent.

† The work reported here was supported by Contract MDA903-82-C-0061 from the Information Processing Technology Office of the Defense Advanced Projects Agency (DOD)

Several kinds of distributed problem-solving difficulties follow from this characterization. First, there are difficulties with *optimal task assignment.* Many mappings of decomposed subtasks to agents are possible, but, because agents have differing appropriateness with respect to a given task, only a few agents will be acceptable for any task. In many distributed problems it is crucial for agents to adopt the right *role.* In addition to insuring that any given task is assigned to an appropriate agent, the group has to achieve *task coverage.* All subtasks should be assigned to some agent (complete role assignment) and multiple unnecessary agents should not be assigned a task (consistent role assignment). The limited knowledge of agents compounds the difficulty of optimal task assignment and task coverage. Incomplete knowledge prevents consistent and complete role assignment because no one agent may have a global knowledge of all the roles or subtasks that need to be assigned. Optimal task assignment is also threatened since agents may not know about tasks for which they are the most appropriate.

Second, *task coordination problems* arise because tasks assigned to agents may not be independent. For example, if two blocksworld agents are each to build towers (as subtasks of a larger task), the plan that one agent produces might negatively interact with the plan of another if both intended to use the same block. While single-agent problem solvers have difficulties in handling non-independent tasks or subgoals, these difficulties multiply for distributed problem solvers. Again, limited knowledge is the reason. If two agents have only local knowledge—if they know only the local environment, know only their tasks and intentions-then they will not know of, or be able to prevent, negative interactions between their roles and those of other agents.

In summary, a main challenge to distributed problem solving is that the solutions which a distributed agent produces must not only be locally acceptable, achieving the assigned tasks, but also they must be interfaced correctly with the actions of other agents solving dependent tasks. The solutions must not only be reasonable with respect to the local task, they must be *globally coherent* and this global coherence must be achieved by *local computation alone.*

III. STRATEGIES FOR COOPERATION

How can groups achieve global coherence in the face of limited knowledge and the requirement that all computation be local? Broadly, the key to coherent distributed problem solving lies in the fact that while distributed agents have greater difficulties in solving a given task, they have potentially more options as well. For example, a distributed agent may plan or act, but he may also request others to do so. In short, much of the power of distributed problem solving comes through cooperation and communication.

Although communication between agents provides the basis for effective cooperative problem solving, it is just another problem-solving tool that may be used poorly or effectively. If the tool is used poorly then group problem-solving performance may be worse than individual problem-solving performance. Considerable expertise is required to use communication effectively. We refer to such expertise as *cooperative strategies.* Our main theoretical and empirical goals have been to understand two distinct classes of such strategies: *organizational policies* and *information-distribution policies.*

A. Organizational Policies

Organizational policies dictate how a larger task should be decomposed into smaller (sub)tasks which can be assigned to individual agents. Typically a given organizational policy assigns specific roles to each of the agents in a group. Such a policy is useful if for some tasks the resulting division of labor enables agents to work independently. For example, the corporate hierarchy is an organizational policy that is particularly effective if the corporate task can be decomposed in such a way that an agent at one level can work independently of others at that level, reporting results only to his immediate superior who takes care of any necessary interfacing.

Organizational policies not only define a task decomposition but prescribe communication paths among agents. They turn a random collection of agents into a network that is fixed, at least for a given task. In the corporate hierarchy, again, the arcs between agents usually indicate which pairs are permitted to talk to one another, and, in addition, determine the nature of the messages that are allowed. Such communication restrictions will be beneficial if they encourage only those agents who should communicate to do so; in particular, agents that have dependent tasks or who may share resources. In general, organizational policies strongly direct and constrain the behavior of distributed agents. If those constraints are appropriate to the task at hand, then the organization is effective, otherwise its performance may be sub-optimal.

In our distributed problem-solving systems and others |Il),[2] groups begin by establishing an organizational policy. To do so, the agents must not only know which policy is appropriate to the current circumstances, but also must know *techniques by which a group can implement the chosen policy in a distributed fashion.* Briefly, any distributed method of implementing an organizational policy must answer a variety of questions, including:

- When does organization structuring take place'
- How is the assignment of roles specified by the policy made to agents? In other words, how is the agent who is most appropriate found for a given task found?
- Are agents "externally-directed" or "data-directed" [7|? That is. does an agent arrive at its roles by being told them or is information relayed, allowing it to make the assignment of roles itself?
- When an agent is requested by another to conform to a role, or take on another subtask for that agent, does the first agent have a right to negotiate? How does an agent weigh the value of competing tasks?

Smith |II] has proposed the contract net as a formalism for implementing organizational policies in a distributed fashion. In Section V.B., we discuss how some organizational policies were imposed in distributed air-traffic control

B. Information-distribution Policies

An information-distribution policy addresses the nature of communication between cooperating agents. Decisions about how agents communicate with each other are, first of all, constrained by the choice of organizational policy, since that policy decides the network of permissible communicators. However, within these constraints, a great number of lower level decisions must be made about how and when communications should occur. Briefly:

- *Broadcast or selective communication.* Are agents discriminating about who they talk to. If so what criteria arc used to select recipients?
- *Unsolicited or on-demand communication.* Assuming you know who you want to communicate with, do you do so only if information is requested, or do you infer the informational needs of other agents and transmit data accordingly?
- *Acknowledged or unacknowledge communication.* Do you indicate that you have received information?
- *Single-transmission or repeated-transmission communication.* Is a piece of information only sent once, or can it be repeated? How frequently? Lesser and Erman [6] refer to a repeated-transmission policy as murmuring.

Poor decisions at this level result, at best, in the highly inefficient use of limited-bandwidth channels. At worst, such choices endanger global coherence by preventing agents whose tasks may interact from talking to one another. The goal of information-distribution policies is to minimize these possibilities. As with organizational policies, the utility of communication policies depends on current conditions. These include the bandwidth of the communication channel, the reliability of the channel, the load of the channel, the maximum acceptable information turn-around time, and the relative cost (and time) of computation versus communication.

IV. DISTRIBUTED PROBLEM SOLVING IN AIR-TRAFFIC CONTROL

Problem solving in air-traffic control may be distributed in several ways Elsewhere |12) we discuss a variety of architectures of distribution. Currently we have implemented only *object-centered* systems, where one agent is associated with each aircraft. In our air-traffic control task, aircraft enter a rectangular (14 x 23 mile) airspace at any time, either at one of 10 infixes on the borders of the airspace, or from one of two airports. The main goal of the agent associated with each aircraft is to traverse the airspace to an assigned destination—either a boundary outfix, or an airport. Each aircraft has only a limited sensory horizon, hence its knowledge of the world is never complete and it must continually gather information as it moves through the airspace. Information may be accumulated either by sensing or communication. Agents are allowed to communicate over a limited bandwidth channel to other aircraft for purposes of exchanging information and instructions

Distributed ATC is a group problem not only because agents may help one another gather information but also because the goals of one agent may interact with those of another Coal interactions come in the form of shared conflicts A conflict between two or more agents arises when, according to their current plans, the two will violate minimum separation requirements at some point in the future. When shared conflicts arise, agents must negotiate to solve them. In a crowded airspace, such goal conflicts can get particularly complex, and involve several aircraft, thus necessitating a high degree of group cooperation.

In terms of the vocabulary developed in Section II, the detection and resolution of conflicts are the main distributed problem-solving *tasks.* These tasks may be decomposed into several *subtasks* or distinct *roles.* Agents may gather information about a shared conflict, evaluate or interpret the information, develop a plan to avoid a projected conflict, or execute such a plan. Agents may be more or less *appropriate* for such roles depending on their current processing load (Are they currently involved in helping resolve other conflicts?), their state of knowledge (Do they know a lot about the intentions of other agents in the conflict?), or their spatial constraints ((an they see many nearby aircraft? Do they have much excess fuel'').

The issue of *optimal task assignment* arises because a group of aircraft may fail to assign the agent that is the most appropriate to each role in a conflict-task if some of the conflictees do not know about a shared conflict In addition, care must be taken that a complete and consistent set of roles is assigned. Some role inconsistencies can be fatal, Ior example, two agents would be adopting inconsistent roles if one derides to move left to avoid a head-on collision with the second while the second decides to jog right. Severe *task coordination problems* also threaten to arise in distributed ATC. The action of moving to avoid one conflict may create or worsen other conflicts (negative task interactions) or lessen other conflicts (positive task interactions). Both forms of interaction are caused by the fact that while several agents may be dealing with different conflict-tasks, they are nevertheless exploiting shared limited spatial resources.

V. FOUR DISTRIBUTED PROBLEM SOLVERS FOR AIR TRAFFIC CONTROL

We now outline the cooperative strategies embedded in four distinct ATC systems. All four systems are implemented in our framework for constructing distributed agents |8). This in turn is implemented in INTERLISP- D, running on Xerox Dolphins.

A. Information-distribution Policies In ATC

The information-distribution policy common to all four systems prescribes that information should be sent to other aircraft selectively (no broadcasting), without waiting for a request, without expecting an acknowledgement, and without repeating the information a second time. These choices are reasonable since we assume in all systems that communication is error-free. When we add noise to the communication channel, we envision adopting a policy that injects some needed redundancy or safety into communication; for example a policy that includes murmuring I6|. We also assumed a constant effective communication bandwidth for all four systems. Each aircraft was allowed to send a maximum of 5 messages per 15 seconds of time.

B. Organizational Policies In ATC

The organizational policy embedded in three of the four systems may be characterized as task centralization, and the fourth system adheres to a policy of task sharing. Under task centralization, the agents involved in any given conflict task will choose one of their number to play most of the roles. In particular one agent will perform the evaluation role (do all the evaluation of the potential conflict between aircraft), the plan-fixing role (attempt to devise a plan-fix to dissolve the entire conflict) and the actor role (act on the new plan). The selected agent is required to modify only his plan to resolve the conflict, thus the remaining agents perform no planning or actions. Instead these conflictees, having agreed on the choice of a replanner, adopt passive information-gathering roles, merely sending their intentions (plan) to the selected agent. The policy of task centralization, whatever its shortcomings, is worth considering because it enjoys many of the advantages of centralized, single-agent problem solving that it is meant to mimic. Specifically, by centralizing most task roles in a single agent the group has to worry less about negative task interactions such as the threat of two aircraft acting in an inconsistent fashion, noted above.

Although three of our four systems embed a task centralization policy, they differ in how they measure and choose the agent who is the most appropriate for the several centralized roles.

Selection by shared convention. Here each aircraft uses only directly sensed information about the other aircraft (position, heading, and speed) to decide who should plan and who should transmit its current route. The aircraft silently use a common set of conventions for this decision, minimizing communications. Figure 1 shows a prototypic sequence of tasks and communications between two aircraft under this policy.

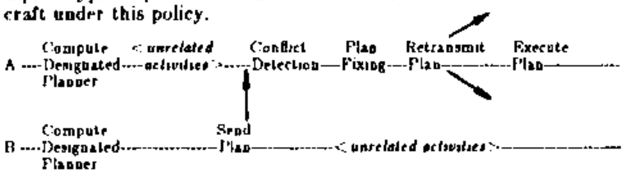

Figure 1 Prototypic task sequence under Shared Convention policy

Time lines for tasks executed by aircraft A and B Solid lines indicate communications

Because of the limited criteria used, the aircraft selected as the replanner is not likely to be the most appropriate. This version mainly serves as a benchmark against which to judge the utility of more intelligent methods of selection which are also more costly in terms of computation and communication.

Selection of the least spatially constrained agent. Here each aircraft in a potential conflict transmits its *constraint factor* to the other aircraft. The constraint factor is an aggregation of such considerations as the number of other nearby aircraft, fuel remaining, distance from destination, and message load. Figure 2 gives an idea of the standard sequence of tasks and communications under this policy.

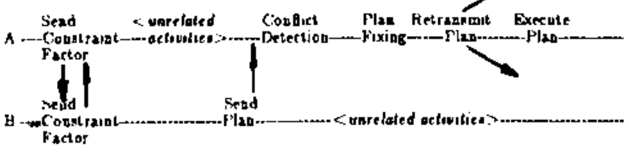

Figure 2. Prototypic task sequence under Least Constrained policy

This method of selection maintains that the agent that is the most appropriate is the one with the most degrees of freedom for modifying its plan. It is a more complex selection than the shared convention, and should result in more effective replanner choices, although at some additional cost in initial communications.

Selection of the most knowledgeable, least committed agent. As above, aircraft share constraint factors with the other conflictees, but here they are computed differently. This method of selection maintains that the best agent to replan is the one who knows the most about other agents' intentions, because, in replanning, a well informed agent can explicitly take account of possible interactions between his intentions and those of other agents. More globally coherent plan-fixes should therefore result. In addition, this method also says that agents whose intentions are known by others should not replan. If such an agent does modify its plan, it will have violated the expectations of cooperating agents, making their knowledge incorrect and in turn making cooperation difficult. Thus, this policy implements a common adage of cooperation: Don't do the unexpected.

In spite of their simplicity, task centralization policies are often ineffective. Although the agent selected to perform the centralized roles may be overall the best, that agent is rarely the best for each of the centralized roles. For example, we still might want to assign the actor role to the agent in a conflict set who is least constrained in the sense defined above. However, that agent might not be the best in the set for fixing his plan — for making a modification to the plan and evaluating the implications of such a change. Presumably the best agent for this role is the (possibly distinct) member of the conflict set that knows most about the environment and intentions of aircraft neighboring the one whose plan is to be fixed. This aircraft is in the best position to determine that any changed plan is not only locally reasonable, solving the conflict, but that it is globally reasonable, not creating new conflicts with other aircraft.

Our *task sharing* policy attempts to avoid such problems by evaluating agents' qualifications with respect to each of the roles associated with a conflict. While in centralized policies a single negotiation determines an overall replanner, in the task sharing policy two rounds of negotiation are necessary, one to determine the plan-fixer and one to determine the actor. Figure 3 gives a detailed description of a prototypical sequence of tasks and communications showing how such a policy is implemented in a distributed fashion.

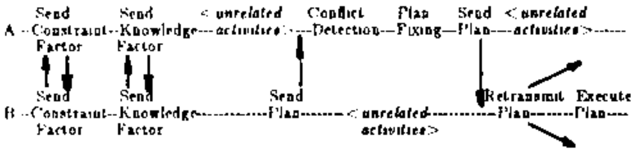

Figure 3 Prototypic task sequence under Task Sharing policy

Performance of groups working under a task sharing policy is potentially superior to groups working under a policy of centralization because in the former the group attempts to optimize on each role. However, in practice this policy has several possible drawbacks. It is communication intensive and may be inappropriate when communication channels are unreliable or costly. Moreover, it risks potential negative interactions, because several agents have to coordinate intimately to achieve a solution.

VI. EXPERIMENTAL STUDIES

We are conducting a series of rudimentary experimental studies on the four policies outlined above. We report here on results pertaining to only the three task centralization policies, since data collection for the task sharing policy is not yet complete. AH three variants were tested on eight distributed scenarios. Each scenario stipulated (i) how many aircraft would enter the airspace in the session, (ii) when and where they would enter, and (iii) where they would exit. This control over the parameters of distributed problem-solving situations allowed us to isolate situation features that uncovered the strengths and weaknesses in performance of our policies. In particular, the scenarios varied considerably in task density, time stress, and task difficulty.

We examined three performance indices when comparing the systems: communication load, processing time, and task effectiveness. Task effectiveness was indicated by two distinct factors: separation errors (more important) and fuel usage (less important). A summary of the main results is given in Table 1.

VII. DISCUSSION OF RESULTS

Examining the individual scenarios, we found the Shared convention policy, relying on essentially arbitrary assignment of planning responsibility, performed well only in low complexity, low difficulty tasks. It minimized communications and responded rapidly compared to the other policies but quickly foundered in 3- and 4-body conflicts.

	Shared Convention	Least Constrained	Most Knowledgeable
Communications (i)	10.9	28.6	28.2
Processing time (ii)	1265	1726	1651
Separation errors (iii)	4.3	1.4	2.3
Fuel usage (iv)	96	108	101

(i) *Communication load* = Mean messages sent per aircraft while flying from infix to outfix
(ii) *Processing time* = mean Xerox 1100 cpu seconds per aircraft while flying
(iii) *Separation errors* = Mean number of near misses or collisions for all aircraft in a scenario
(iv) *Fuel usage* = Mean number of fuel units used for all aircraft

Figure 1. Performance measures of three organizational policies (statistics averaged across 8 scenarios)

The *Least Constrained* policy performed best overall. It did particularly well with high complexity, high difficulty tasks. In such cases the planning aircraft tended to be located at the edge of the fray, able to find more viable solutions than those aircraft in the interior. The policy was time and communication intensive, however, largely because of the high number of messages needed to cooperatively determine the replanner and to maintain consistency after replanning. In any of the three systems, when a replanner is successful it must send *data retransmission messages* to all aircraft to which it had previously sent its intentions. The number of data retransmissions was especially high under the *Least Constrained* policy.

The *Most Knowledgeable* policy was intermediate in performance. It performed best in tasks of low complexity and high difficulty, that is, tasks with primarily 2- and 3-body interactions but having few potential solutions. In complex multi-aircraft situations, if the wrong aircraft was chosen for planning, the effect was often catastrophic. This is because the aircraft that then received replan requests tended to have little knowledge about the routes of other aircraft. By design of the policy, this knowledge was typically concentrated in the initially selected planner.

When successful, the *Most Knowledgeable* policy's performance was in some ways better than that of the *Least Constrained* policy. In particular, when an agent found a solution to a local conflict-task under the *Most Knowledgeable* policy its solution was likely to be more globally coherent than under other policies, since the replanning agent was selected partially because of his wide knowledge of other aircraft's plans. This knowledge allowed the agent to more effectively replan without incurring new conflicts. In addition, a successful replanning agent under a *Most Knowledgeable* policy generally needed to issue less data retransmission messages than under the other policies, since it was selected to replan partially because its intentions were known to fewer others (i.e., it was the less committed agent). We had initially anticipated minimizing data retransmissions would be very important for guaranteeing globally coherent performance. We envisioned situations where one retransmission would cause the receiving agent to re-evaluate, possibly finding new conflicts, causing more replanning, further data retransmissions, and so on in a vicious propagation of changes. This did not happen as much as we had expected under the the *Least Constrained* policy, although a few instances were observed.

Another expectation that did not arise was a wide variation in processing times among the aircraft under the *Most Knowledgeable* policy. This policy should tend to bias replanning in the favor of a few agents. If an agent is the replanner once, it gains new knowledge of other's plans making it an even better choice as replanner for later conflict-tasks. We anticipated that this concentration would skew the processing times compared to a more uniform distribution of responsibilities under the other policies. This would have been a disadvantage in a truly distributed system, as some agents would be quiescent much of the time. The expected variation in times did not evidence itself, however, except in the relatively easy scenarios.

VII. CONCLUSIONS

Distributed problem solving is an enigma. Potentially, a group of agents should be able to solve problems more effectively than the same agents working individually. In practice, however, groups often work ineffectively and their joint productivity is less than the sum of the productivities expected of each member. Our aim is to discover the elusive cooperative strategies that enable groups to reach optimum productivity. On the theoretical side we are developing concepts that allow us to simply and formally describe various cooperative strategies. On the empirical side, we are testing such strategies by imposing them on groups and observing the resulting group performance. Both phases are necessary. The theoretical investigations are valuable because most problem solving models presently available describe only the information processing of single agent problem solvers, not distributed problem solvers. The empirical work is necessary because many of the behavioral properties of complex cooperative strategies are not apparent without observing how they actually perform in real or simulated settings.

REFERENCES

[I) Appelt, D. E., Planning Natural-language utterances to satisfy multiple goals, Technical Note 259, SRI International, 1982.

[2] Davis, R., Smith, R.G., Negotiation as a metaphor for distributed problem solving, Memo 624, MIT Artificial Intelligence Lab, 1981.

(3) Fahlman, S., A planning system for robot construction tasks, *Artificial Intelligence,* 5, (1), 1974, 1-49.

[4] Fikes, R.E., Nilsson, N.J., Strips: A new approach to the application of theorem proving to problem solving, *Artificial Intelligence,* 2, (2j, 1971, 189-208.

[5] Konolige, K, A first order formalization of knowledge and action for a multi-agent planning system, *Machine Intelligence 10,* 1981.

[6] Lesser, V. R., Reed, S., Pavlin, J., Quantifying and Simulating the behavior of knowledge-based interpretation systems, *Proceedings of the First Annual National Conference on Artificial Intelligence,* Stanford University, 1980, 111-115.

[7) Lesser, V., A high-level simulation testbed for cooperative problem solving, COINS Technical Report 81-16, University of Massachusetts, Amherst, MA, 1981.

[8) McArlhur, D., Steeb, R., and Cammarata, S., A framework for distributed problem solving, *Proceedings of the National Conference on Artificial Intelligence,* Pittsburg, PA, 1982, 181-184.

[9] Sarerdoti, E., Planning in a hierarchy of abstraction spaces. *Artificial Intelligence,* 5, (2), 115-135, 1974.

(10) Sacerdoti, E., A structure for plans and behavior. New York: Elsevier North-Holland, 1977.

[II] Smith, R.G., A framework for problem solving in a distributed processing environment, STAN-CS-78-700, Stanford University, 1978.

(12] Steeb, R., Cammarata, S., Hayes-Roth, F.A., Thorndyke, P.W., Wesson, R.B., Distributed intelligence for air fleet control, R-2728-ARPA, The Rand Corporation, 1981.