Multistage Negotiation in Distributed Planning

S.E. Conry
R.A. Meyer
V.R. Lesser

COINS Technical Report 86-67
December 1986

Computer and Information Science Department
University of Massachusetts
Amherst, Massachusetts 01003

Abstract

In this paper we describe a multistage negotiation paradigm for planning in a distributed environment with decentralized control and limited interagent communication. The application domain of interest involves the monitoring and control of a complex communications system. In this domain planning for service restoral is performed in the context of incomplete and possibly invalid information which may be updated dynamically during the course of planning. In addition, the goal of the planning activity may not be achievable — the problem may be overconstrained. Through multistage negotiation, a planner is able to recognize when the problem is overconstrained and to find a solution to an acceptable related problem under these conditions. A key element in this process is the ability to detect subgoal interactions in a distributed environment and reason about their impact. Multistage negotiation provides a means by which an agent can acquire enough knowledge to reason about the impact of local activity on nonlocal state and modify its behavior accordingly.

This work was supported, in part, by the Air Force Systems Command, Rome Air Development Center, Griffiss Air Force Base, New York 13441-5700, and the Air Force Office of Scientific Research, Bolling AFB, DC 20332 under Contract No. F30602-85-C-0008. This contract supports the Northeast Artificial Intelligence Consortium (NAIC). This work was also supported, in part, by the National Science Foundation under CER Grant DCR-8500332, and by the Defense Advanced Research Projects Agency (DOD), monitored by the Office of Naval Research under Contract NR049-041. This work was done while S.E. Conry and R. A. Meyer were on sabbatical at the University of Massachusetts.

Multistage Negotiation in Distributed Planning

S. E. Conry and R. A. Meyer
Clarkson University
Potsdam, N. Y. 13676
(315) 268-6510, -6541, -6511
conry @ clvms (bitnet)

V. R. Lesser
University of Massachusetts
Amherst, MA 01003
(413) 545-1322
lesser @ umass-cs.umass.edu (csnet)

December 15, 1986

Paper Type: Full-paper

Topic: Reasoning

Track: Science

Keywords: Distributed problem solving

Abstract

In this paper we describe a multistage negotiation paradigm for planning in a distributed environment with decentralized control and limited interagent communication. The application domain of interest involves the monitoring and control of a complex communications system. In this domain planning for service restoral is performed in the context of incomplete and possibly invalid information which may be updated dynamically during the course of planning. In addition, the goal of the planning activity may not be achievable — the problem may be overconstrained. Through multistage negotiation, a planner is able to recognize when the problem is overconstrained and to find a solution to an acceptable related problem under these conditions. A key element in this process is the ability to detect subgoal interactions in a distributed environment and reason about their impact. Multistage negotiation provides a means by which an agent can acquire enough knowledge to reason about the impact of local activity on nonlocal state and modify its behavior accordingly.

This work was supported, in part, by the Air Force Systems Command, Rome Air Development Center, Griffiss Air Force Base, New York 13441-5700, and the Air Force Office of Scientific Research, Bolling AFB, DC 20332 under Contract No. F30602-85-C-0008. This contract supports the Northeast Artificial Intelligence Consortium (NAIC). This work was also supported, in part, by the National Science Foundation under CER Grant DCR-8500332, and by the Defense Advanced Research Projects Agency (DOD), monitored by the Office of Naval Research under Contract NR049-041. This work was done while S. E. Conry and R. A. Meyer were on sabbatical at the University of Massachusetts.

1 INTRODUCTION

We present a multistage negotiation protocol that is useful for cooperatively resolving resource allocation conflicts which arise in a distributed network of semi-autonomous problem solving nodes. The primary contributions of such a negotiation protocol are that it makes it possible to detect and to resolve subgoal interactions in a distributed environment with limited communication bandwidth and no single locus of control. Furthermore, it permits a distributed problem solving system to detect when it is operating in an overconstrained situation and act to remedy the situation by reaching a satisficing [1] solution.

Multistage negotiation is specifically *not* intended as a mechanism for goal decomposition in the system, though some goal decomposition is a natural result of negotiation in the context of this paradigm. Our protocol may be viewed as a generalization of the contract net protocol [2,3,4]. The contract net was devised as a mechanism for accomplishing task distribution among agents in a distributed problem solving system. Task distribution takes place through a negotiation process involving contractor task announcement followed by bids from competing subcontractors and finally announcement of awards. Multistage negotiation generalizes this protocol by recognizing the need to iteratively exchange inferences drawn by an agent about the impact of its own choice of what local tasks to perform in satisfaction of *global goals*.

Multistage negotiation produces a cooperation strategy similar in character to the Functionally Accurate/Cooperative paradigm [5] in which agents iteratively exchange tentative and high level partial results of their local subtasks. This strategy results in solutions which are incrementally constructed to converge on a set of complete local solutions which are globally consistent. Before describing multistage negotiation in detail, we first motivate the need for a new cooperation paradigm.

2 MOTIVATION FOR MULTISTAGE NEGOTIATION

The distributed environment in which our negotiation takes place is a network of loosely coupled problem solving agents in which no agent has a complete and accurate view of the state of the network. Problem solving activity is initiated through the instantiation of one or more top level goals at agents in the network. Each top level goal is instantiated locally at an agent and is not necessarily known to other agents. Since the conditions which give rise to goal instantiation may be observed at more than one place in the network, the same goal may be instantiated by two or more agents independently. The desired solution to the problem is any one that satisfies all of the top level goals.

In this type of distributed network, it is very expensive to provide a complete global view to each agent in the system. Communication bandwidth is generally limited. Exchange of enough information to permit each agent to construct and maintain its own accurate global view would be prohibitively expensive. In addition, progress in problem solving would be significantly slower due to a decrease in parallelism attributable to the need for synchronization in building a complete view. Multistage negotiation has been devised as a paradigm for cooperation among agents attempting to solve a planning problem in this distributed environment. In the remainder of this section, we explain the contributions of multistage negotiation in solving distributed planning problems.

One of the major difficulties which arises in planning systems is detecting the presence of subgoal interactions and determining the impact of those interactions. In distributed applications, the problem is exacerbated because no agent has complete knowledge concerning all goals and subgoals present in the problem solving system. For example, subgoals initiated by one node may interact with other subgoals initiated elsewhere, unknown to the first node. These interactions may become quite complex and may not be visible to any single node in the network. *A key objective of our multistage negotiation is to allow nodes to exchange sufficient information so that these interactions are detected and handled in a reasonable manner.* This objective is achieved by exchanging knowledge about the nonlocal impact of an agent's proposed local action without requiring the exchange of detailed local state information.

Another significant issue that arises in planning is recognizing when goals are not attainable. When satisfaction of a goal requires the commitment of resources, conflicts may arise among goals competing for limited resources. A planning problem is overconstrained if satisfaction of one top level goal precludes the satisfaction of others. Detection of an overconstrained situation in a distributed environment is, again, particularly difficult because no agent is aware of all goals, and each agent has only a limited view of the complete set of conflicts. *When a number of alternative choices for goal satisfaction are known, detection of an overconstrained situation is not possible without either multistage negotiation or a global view.*

In an overconstrained problem, a planning system must reformulate what it seeks as a satisfactory solution. Having several equally important top level goals, the planner must decide which ones should be sacrificed to permit satisfaction of others. Since the distributed network has no agent with sufficient knowledge to serve as an intelligent arbitrator, a consensus must be reached. *Multistage negotiation provides a mechanism for reaching a consensus among those nodes with conflicting goals concerning an acceptable satisficing solution.*

In the following sections, we first describe the problem in more detail, discussing the application domain itself as well as an example which illustrates the nature of the planning problem. We then discuss two models of problem solving relevant to this domain: one which is oriented from the perspective of a single goal and one which is node centered. In the fifth section we discuss a multistage negotiation protocol which utilizes these models and has been incorporated in a distributed planner for this problem. We illustrate this protocol with the aid of a simple example. Finally, we discuss ways in which this research extends existing work.

3 APPLICATION DOMAIN

The application domain of interest is the monitoring and control of a complex communications system. This system consists of a network of sites, each containing a variety of communications equipment, interconnected by links. These sites are partitioned into several geographic subregions with a single site in each subregion designated as a control facility. Each control facility has responsibility for communication system monitoring and control within its own subregion and corresponds to a single node in the distributed problem solving network. In order to distinguish between the communication network and the problem solving network, in this paper we reserve the term "site" to mean a physical location in the communication system. The term "node" will be used to refer to those sites at which processing and control reside.

The communication network considered here represents a long-haul, transmission "backbone" of a larger, more complex communications system. From this transmission oriented perspective, each user is provided with a dedicated set of resources (equipment and link bandwidth) which establishes a point-to-point connection, or **circuit** for a significant period of time. Any equipment failure or outage will cause an interruption of service to one or more users.

An overall knowledge-based system to perform the monitoring and control function would employ distributed problem solving agents involving data interpretation, situation assessment, fault diagnosis, and planning [6]. In this context planning is used to find restoral plans for user circuits which have been interrupted as a result of some failure or outage.

A restoral plan consists of a logical sequence of control actions which allocate scarce resources in order to restore end-to-end user connectivity (circuits). These actions allocate or reallocate equipment and link capacity along some route to specific circuits and are subject to a number of constraints. For example, a circuit is assigned to one of several priority categories. In attempting to restore service, resources belonging to circuits of a lower priority may be preempted. Depending upon the type of circuit, there may be special equipment needs which are not necessarily present at all sites. Available routes through the network may be constrained by lack of certain equipment items such as switches or multiplexers. Thus generation of a restoral plan for a single circuit uses conventional route finding algorithms [7] in combination with knowledge about circuit types and priority, needed equipment, network topology, and equipment configuration at all sites along the restoral path. For any specific circuit there will generally be many alternative restoral plans, so the planning system must then attempt to select a combination of alternatives which restores all circuits.

There are a number of features of this planning problem that make it interesting. There is implicit in this domain the assumption that the knowledge of each agent is incomplete. It may also be inaccurate and inconsistent with that of other agents. Restoral plans must be generated in a distributed fashion because no agent has a global view and reliability issues mitigate against delegating the responsibility for planning to a central node. The overall system goal is one of determining plans for restoral of all interrupted service. Although each agent implicitly knows this goal, it generally will not know all of the specific circuits which require restoral. The planning system need not satisfy the overall goal to be successful. In many instances, the overall goal may be infeasible, and thus a satisfactory plan will fall short of reaching this goal.

The distributed planning problem addressed in this paper and our approach to solving it can best be understood with the aid of an example. A simplified diagram of a small network is shown in Figure 1. In this phase of our work, we use a simplified model of a communications system which disregards any constraints arising from equipment configuration at a site. There are five subregions, labeled A, B, C, D, and E, shown. Each site is

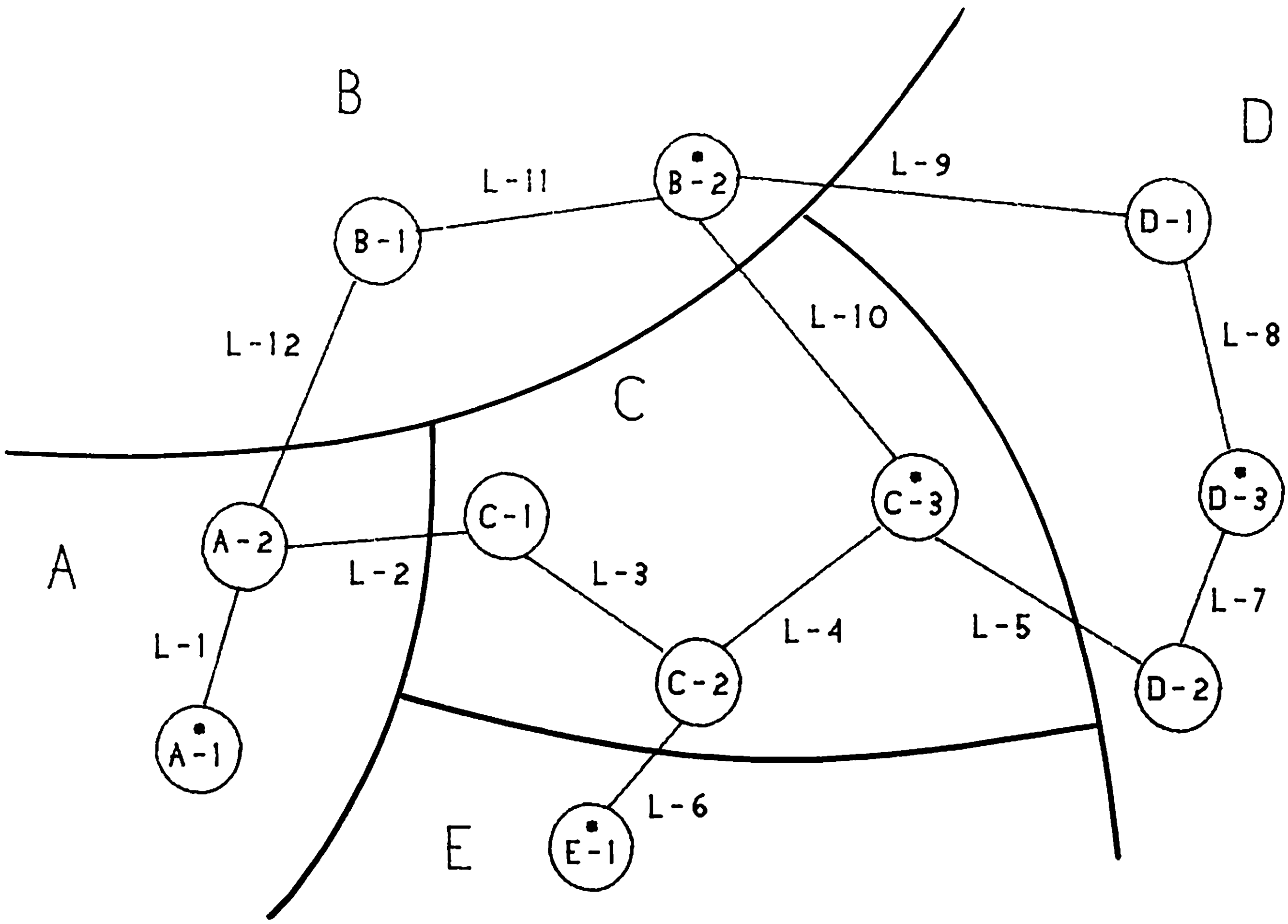

Figure 1: Example Network

designated by a letter-number pair, where the letter indicates the subregion in which the site is located. The communication network links are designated by L-number. The control facility for each subregion is located at the site marked with an "*". Each control facility has a planning agent to restore interrupted service. It should be noted that a separate communication network, of substantially lower bandwidth, is not shown, but is assumed to interconnect the control facilities for the purpose of exchanging messages among the agents.

For the purposes of describing the restoral problem, we assume that there is an equipment malfunction at station B-2 that fails all communication using link L-11. We also assume that each link can handle at most two circuits and that there are four circuits established at the time of the supposed failure. These are described in Table 1 by listing the sites and links along the route of each circuit. To simplify the presentation, these circuits all have the same restoration priority so that none of them should be preferred over the others for restoral in the event of service disruption.

As a result of the presumed failure, two circuits are disrupted, namely ckt-1 and ckt-2 (both use L-11 to get from B-1 to B-2). The planning activity is initiated when an agent observes disruption of a circuit terminating within its subregion and instantiates a restoral

ckt-1	(A-1	:L-1:	A-2	:L-12:	B-1	:L-11:	B-2)		
ckt-2	(B-1	:L-11:	B-2	:L-10:	C-3	:L-5:	D-2)		
ckt-3	(E-1	:L-6:	C-2	:L-4:	C-3	:L-5:	D-2	:L-7:	D-3)
ckt-4	(B-1	:L-12:	A-2	:L-2:	C-1	:L-3:	C-2)		

Table 1: Circuit Descriptions

goal. In this example, the restoral goals are autonomously instantiated in subregion A (for ckt-1), subregion B (for ckt-1 and ckt-2) and subregion D (for ckt-2). Each agent initially has only the following knowledge about ***each circuit terminating in its subregion:***

- a circuit identifier that is unique within the network,
- a priority or degree of urgency for restoral,
- detailed routing of this circuit within this agent's area of responsibility, and
- the end stations of the circuit and the agents responsible for them.

In addition, each agent has detailed knowledge concerning the status of resources resident in its subregion.

The first phase of the planning process is plan generation, and since it uses only one stage of negotiation, as in contract nets [2,3,4], we shall not consider the details of plan generation here. When viewed from a global perspective, plan generation produces two alternative restoral plans for each circuit. Each plan is represented in Table 2 as a list of alternating sites and links, traversing the proposed restoral path. To clarify the example,

Plans for goal g1 to restore ckt-1:									
g1/p1	(A-1	:L-1:	A-2	:L-2:	C-1	:L-3:	C-2	:L-4:	C-3
	:L-5:	D-2	:L-7:	D-3	:L-8:	D-1	:L-9:	B-2)	
g1/p2	(A-1	:L-1:	A-2	:L-2:	C-1	:L-3:	C-2	:L-4:	C-3
	:L-10:	B-2)							
Plans for goal g2 to restore ckt-2:									
g2/p1	(B-1	:L-12:	A-2	:L-2:	C-1	:L-3:	C-2	:L-4:	C-3
	:L-10:	B-2	:L-9:	D-1	:L-8:	D-3	:L-7:	D-2)	
g2/p2	(B-1	:L-12:	A-2	:L-2:	C-1	:L-3:	C-2	:L-4:	C-3
	:L-5:	D-2)							

Table 2: Alternative Plans

we have adopted a naming convention for goals and alternative plans which incorporates the circuit name and plan number; thus the two alternative plans for restoring circuit ckt-1 are designated g1/p1 and g1/p2. It is essential to remember that these are global plans which have been generated in a distributed manner, and no single agent *necessarily* knows of all plans or any one complete plan.

As a result of plan generation, a node produces local alternative plan fragments which may be used to satisfy global goals. Each global plan listed in Table 2 is composed of several fragments distributed over a subset of the agents. This is illustrated in Table 3 which summarizes the knowledge each agent has about goals, alternative plan fragments,

Goal	Plan Frag.	Resources Used	Cost
g1	1A	L-1, L-2	9
g2	7A	L-2, L-12	9
		Agent A	
g1	2B	L-9	9
	5B	L-10	6
g2	8B	L-9, L-10, L-12	9
	11B	L-12	6
		Agent B	
g1	3C	L-2, L-3, L-4, L-5	9
	6C	L-2, L-3, L-4, L-10	6
g2	9C	L-2, L-3, L-4, L-10	9
	12C	L-2, L-3, L-4, L-5	6
		Agent C	
g1	4D	L-5, L-7, L-8, L-9	9
g2	10D	L-7, L-8, L-9	9
	13D	L-5	6
		Agent D	

Table 3: Local Knowledge About Plan Fragments

and local resources. Plan fragments are numbered and each is identified by a letter indicating the responsible agent. Note that agents are not *explicitly* aware of global alternative plans, but are only aware of local alternatives. For example, even though Agent A has resources needed by both g1/p1 and g1/p2, the local plan fragment is the same in both cases, and thus Agent A "sees" only one alternative plan for goal g1.

This example is considerably oversimplified in order to focus attention on the significant characteristics of this planning problem and to illustrate the cooperation strategy which results from multistage negotiation. The communication network has been simplified so that link capacity is the only resource, and thus there are no constraints arising from local equipment configurations. The number of circuits and link capacities are also much smaller than is typical. Since only two top level goals exist, the subgoal interactions are simple and can be recognized in only one step. In a more realistic problem, subgoal interactions often involve multiple dependencies and may require several steps of negotiation to detect and resolve.

The features of the planning problem which are important for the discussion of multistage negotiation in this paper are summarized below:

- Goals are autonomously generated at nodes in the system.
- The same system goal may be generated at more than one node, independently.
- Knowledge about local resource availability and potential goal interactions at each node differs from that at other nodes.
- Goal satisfaction in general requires nonlocal resources.
- The planning problem being addressed is, in general, overconstrained. A choice to satisfy some goals may preclude the satisfaction of others, so choice heuristics are necessary.
- Goals are prioritized, but this does not imply a total ordering with respect to priority.

4 MODEL OF PROBLEM SOLVING

The planning problem discussed in the previous section can be viewed in a broader context. In this section we characterize a problem solving model in which multistage negotiation is useful. The search space for a problem of this kind can be considered from two points of view: a task or goal centered perspective and a node centered perspective. Each of these ways of viewing the search space provides a different set of insights with respect to problem solving.

When viewed from the perspective of the system goal, the global problem appears as an AND-OR tree progressing from the system goal (at the root), down through goals and plans, to local plan fragments distributed among the agents. A goal centered view of our example problem is illustrated in Figure 2. Two goals have been instantiated, with four alternative plans and several local plan fragments. Of course, since this is a distributed environment, no single agent has a complete view of this tree. Observe that each agent is aware of both goals g1 and g2, but agent D is only aware of one plan fragment for g1, the one which is a component of g1/p1.

An agent may not simply satisfy a local goal by choosing any plan fragment, but must coordinate its choice so that it is compatible with those of other agents. Formulation of a plan as a conjunction of plan fragments induces a set of compatibility constraints on the local choices an agent makes in satisfaction of global goals. In Figure 2, we show the plan fragments interconnected by dashed lines. These dashed lines indicate the local knowledge an agent has about which other agents are involved in compatibility constraint relations with its own plan fragments. Observe that an agent generally does not have *complete* knowledge about these compatibility sets. In our application domain, these constraints involved shared resources between two agents.

From a node centered perspective, plan fragment selection is constrained by local resource availability. An agent cannot choose to execute a set of alternative plan fragments that require more local resources than are available. For example, agent B's local resources permit selection of any pair of its own plan fragments in satisfaction of g1 and g2, whereas

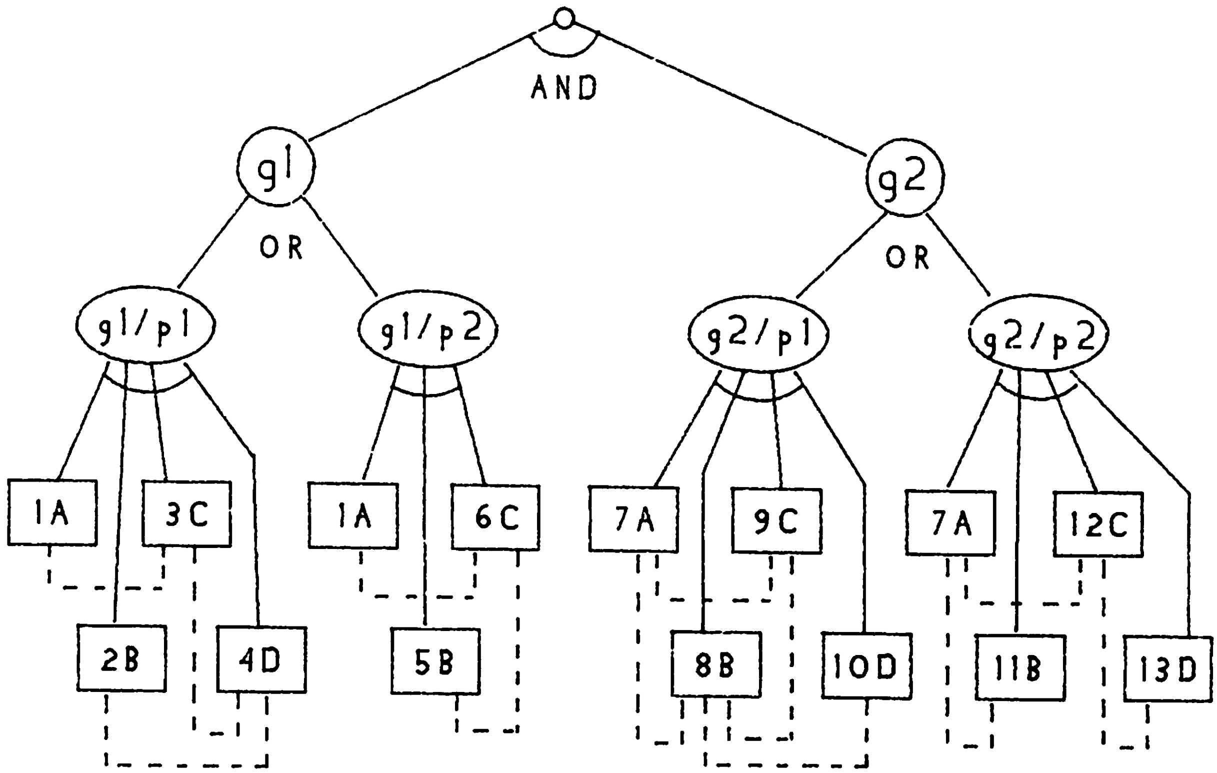

Figure 2: Global Search Space

agents A, C, and D each can select only one plan fragment. The resulting feasibility tree known to each agent is shown in Figure 3. In this figure, resource constraints associated with goals and plan fragments are enclosed by ovals and connected to the appropriate objects with dashed lines. Restoral goals initiated in a subregion are designated with an "*".

From each agent's perspective, the search is over a group of alternatives subject to a set of local resource constraints and a set of compatibility constraints imposed by actions of other agents. Multistage negotiation provides a mechanism by which agents coordinate their actions in selecting plans subject to both resource and compatibility constraints. As additional constraints are added to an agent's base of knowledge, its local feasibility tree is augmented to reflect what it has learned.

5 MULTISTAGE NEGOTIATION

In this section, we describe the multistage negotiation protocol we have developed and give an example of its application in the distributed planning problem which has been discussed. We first treat the protocol at a very high level, discussing the general strategy. We then provide more detail as to phases of planning and the role of negotiation in each. The section is concluded with a detailed trace of negotiation and reasoning in each agent pertinent to our simple example.

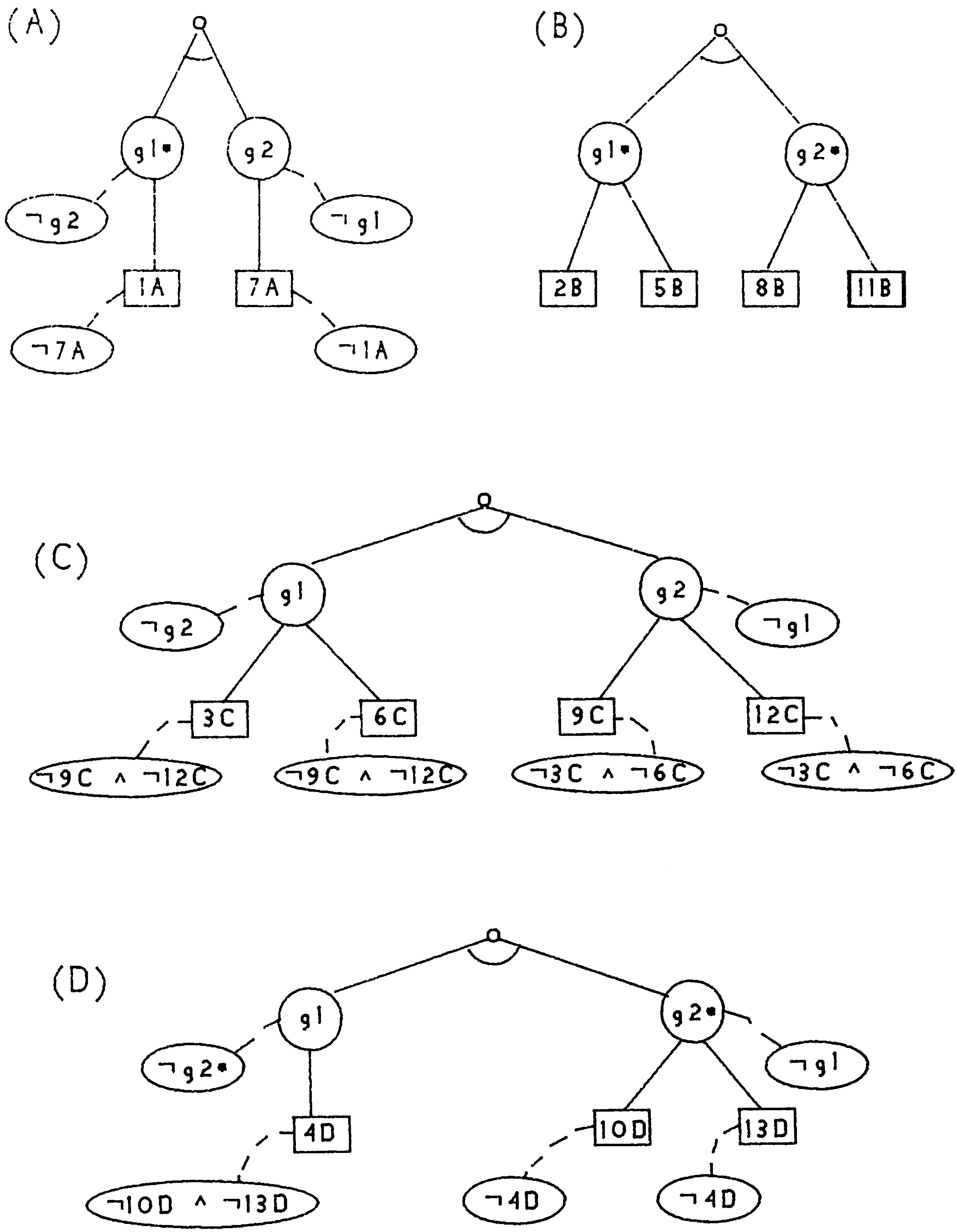

Figure 3: Local Feasibility Trees

High Level Protocol

Multistage negotiation provides a means by which an agent can acquire enough knowledge to reason about the impact of local activity on nonlocal state and modify its behavior accordingly. When problem solving activity is initiated, agents first engage in a phase of plan generation. Each agent ascertains what alternatives for partial goal satisfaction are locally possible and tenders contracts to appropriate agents for furthering satisfaction of the goals needed to complete these plans. On completion of this phase, a space of alternative plans has been constructed which is distributed among the agents, with each agent only having knowledge about its local plan fragments. An agent then examines the goals it instantiated and makes a tentative commitment to the highest rated feasible set of plan fragments relative to these goals. It subsequently issues requests for confirmation of that commitment to agents who hold the contracts for completion of these plan fragments.

Each agent may receive two kinds of communications from other agents: 1) requests for confirmation of other agents' tentative commitments, and 2) responses concerning the impact of its own proposed commitments on others. Impact of local actions is reported as confirmation that a tentative local choice is a good one or as negative information reflecting nonlocal resource conflict. The agent incorporates this new knowledge into its local feasibility tree. It rerates its own local goals using the new knowledge and possibly retracts its tentative resource commitment in order to make a more informed choice. This process of information exchange continues until a consistent set of choices can be confirmed.

Termination of the negotiation process can be done using system-wide criteria or it can be accomplished in a diffuse manner. If global termination criteria are desired in an application, some form of token passing mechanism [8,9,10] can be used to detect that the applicable termination criteria have been met. When synchronized global termination is not required in an application, the negotiation can be terminated by an "irrevocable" commitment of resources. A node initiates plan execution in accordance with its negotiated tentative commitment at some time after it has no pending activities and no work to do for other agents.

Mechanics of Negotiation

When a node begins its planning activity, it has knowledge of a set of top level goals which have been locally instantiated. A space of plans to satisfy each of these goals is formulated during plan generation without regard for any subgoal interaction problems. After plan generation, each node is aware of two kinds of goals: ***primary goals*** (or p-goals) and ***secondary goals*** (or s-goals). In our application, p-goals are those instantiated locally by an agent in response to an observed outage of a circuit for which the agent has primary responsibility (because the circuit terminates in the agent's subregion). These are of enhanced importance to this agent because they relate to system goals which ***must*** be satisfied by this particular agent if they are to be satisfied at all. An agent's s-goals are those which have been instantiated as a result of a contract with some other agent. An agent regards each of its s-goals as a possible alternative to be utilized in satisfaction of some other agent's p-goal.

A plan commitment phase involving multistage negotiation is initiated next. As this phase begins, each node has knowledge about all of the p-goals and s-goals it has instantiated. Relative to each of its goals, it knows a number of alternatives for goal satisfaction. An alternative is comprised of a local plan fragment, points of interaction with other agents (relative to that plan fragment), and a measure of the cost of the alternative (to be used in making heuristic decisions). Negotiation leading to a commitment proceeds along the following lines.

1. Each node examines its own p-goals, making a tentative commitment to the highest rated set of locally feasible plan fragments for p-goals (s-goals are not considered at this point because some other agent has corresponding p-goals).

2. Each node requests that other agents attempt to confirm a plan choice consistent with its commitment. Note that an agent need only communicate with agents who can provide input relevant to this tentative commitment.

3. A node examines its incoming message queue for communications from other nodes. Requests for confirmation of other agents' tentative commitments are handled by adding the relevant s-goals to a set of active goals. Responses to this agent's own requests are incorporated in the local feasibility tree and used as additional knowledge in making revisions to its tentative commitment.

4. The set of **active goals** consists of all the local p-goals together with those s-goals that have been added (in step 3). The agent rates the alternatives associated with active goals based on their cost, any confirming evidence that the alternative is a good choice, any negative evidence in the form of nonlocal conflict information, and the importance of the goal (p-goal, s-goal, etc.). A revised tentative commitment is made to a highest rated set of locally consistent alternatives for active goals. In general, this may involve decisions to *add* plan fragments to the tentative commitment and to *delete* plan fragments from the old tentative commitment. Messages reflecting any changes in the tentative commitment and perceived conflicts with that commitment are transmitted to the appropriate agents.

5. The incoming message queue is examined again and activity proceeds as described above (from step 3). The process of aggregating knowledge about nonlocal conflicts continues until a node is aware of all conflicts in which its plan fragments are a contributing factor.

Two issues need clarification at this point. One deals with the question of termination and the other is concerned with the quality of the result obtained through negotiation (relative to optimality).

Negotiation in this framework continues as long as there are any pending activities in an agent. The only way a situation leading to nontermination could arise involves an agent's making a tentative commitment and subsequently entering a cycle of retracting and remaking that commitment indefinitely. It is not reasonable to expect that an agent should never retract a tentative commitment. It is also not reasonable to expect that an agent would never decide, based on new knowledge, to recommit to an alternative it had previously rejected. An agent's local reasoning must be able to detect when it is making a tentative commitment it has previously made with no new knowledge. Negotiation activity in an agent terminates either when it has no pending activity and no incoming communications or if an attempt is made to return to a previous commitment with no new knowledge from other agents. Endless loops of commitment and decommitment are prevented through this mechanism.

The other issue of importance at this point is related to the quality of the result obtained through negotiation. In the initial negotiation stage, each agent examines only its p-goals and makes a tentative commitment to a locally feasible set of plan fragments in partial satisfaction of those goals. Since each agent is considering just its p-goals at this stage, the only reason for an agent's electing not to attempt satisfaction of some top level goal is that two or more of these goals are locally known to be infeasible. (This corresponds to an overconstrained problem.)

In subsequent stages of negotiation, both p-goals and relevent s-goals are considered in making new tentative commitments. The reasoning strategy employed at each agent will only decide to forego commitment to one of its p-goals if it has learned that satisfaction of this p-goal precludes the satisfaction of one or more other p-goals elsewhere in the system. *If the system goal of satisfying all of the p-goals instantiated by agents in the network is feasible, no agent will ever be forced to forego satisfaction of one of its p-goals (because no agent will ever learn that its p-goal precludes others), and a desired solution will be found.* If, on the other hand, the problem is overconstrained, some set of p-goals cannot be satisfied and the system tries to satisfy as many as it can. While there is no guarantee of optimality, the heuristics employed should ensure that a reasonably thorough search is made.

To make these concepts concrete, multistage negotiation is applied to the simplified planning problem discussed in the previous sections.

Example

We return to our example of planning activity, assuming that each agent has the knowledge depicted in the appropriate part of Figure 3. A summary of the transactions that occur during negotiation to achieve plan selection is shown in Table 4. This table is segmented by agent and by "time slice" to convey a sense of progress in problem solving through negotiation. The notational conventions are relatively simple. Tentative commitment to a locally known activity and the associated communication issued to an appropriate agent is denoted in the form (plan fragment name; message → agent). Exchange of conflict information is indicated in the form (conflict; type of conflict → agent). To make the trace easier to follow, each received message is noted in the form (source agent → message).
As is evident in Table 4, negotiation begins with tentative commitments to alternatives in agents A, B, and D. Though the problem is overconstrained (it is not possible to restore both ckt-1 and ckt-2), no agent is yet aware of that fact. In response to the initial tentative commitments, there is activity in agents A and C. Agent A knows that it cannot act to satisfy both g1 and g2, but it does *not* know if this precludes satisfaction of g2 (since g2 is an s-goal, there might exist another global plan not requiring any action by A). Since A recognizes the need to *attempt* satisfaction of its own p-goal first, agent A informs agent B there is a conflict between what B requested and satisfaction of one of A's p-goals. Thus A has given B the knowledge that the plan fragment B selected would force A to forego one

A	B	C	D
1A; OK? → C	11B; OK? → A 5B; OK? → C		13D; OK? → C
B → OK? 11B conflict; (11B AND ¬ p-goal g1) → B		A → OK? 1A B → OK? 5B D → OK? 13D match 6C with 1A and 5B 1A is OK → A 5B is OK → B conflict; (13D AND ¬ g1 via C) → D	
C → 1A is OK	A → (11B AND ¬ p-goal g1) C → 5B is OK 8B; OK? → A		C → (13D AND ¬ g1 via C) 10D; OK? → B
B → OK? 8B conflict; (8B AND ¬ p-goal g1) → B	D → OK? 10D 8B; OK? → C		
	A → (8B AND ¬ p-goal g1) B knows g1 and g2 not both possible (not both g1 and g2) → D	B → OK? 8B conflict; (8B AND ¬ g1 via C) → B	
	C → (8B AND ¬ g1 via C)		B → (not both g1 and g2)

Table 4: Summary of transactions during negotiation

of its p-goals.

Agent C has now received three communications requesting that plan fragments be extended. It observes that it can effect a plan completion for g1, satisfying both the request from A and the request from B. It also observes that it cannot satisfy both g1 and g2 with use of its locally known plan fragments due to local resource constraints. Since it has the opportunity to complete a plan for ckt-1 and not for ckt-2, it elects to tentatively commit its resources to plan fragment 6C. Messages reflecting this commitment are formulated and transmitted to A and B, while a message indicating the conflict in C is sent to D.

As a result of this second round of communications, activity in subregions B and D is concerned with exploring the remaining alternatives they have for restoral of ckt-2. An acceptable plan for ckt-1 is already reflected in tentative commitments. Agent B elects to try plan fragment 8B and agent D elects to try 10D. Agent B learns that an attempt to satisfy g2 via 8B also fails in A, so it now knows that the problem is overconstrained. Based on the fact that a way of satisfying g1 has already been located, B elects to forego satisfaction of g2 and advises D that it should also give up on g2. Negotiation terminates with tentative commitments reflecting a plan choice for g1.

In concluding this section we summarize, by "time slice", changes to the local feasibility trees that take place during the negotiation illustrated in Table 4.

Slice 1:

- No changes.

Slice 2:

- No changes in constraints by A.
- 6C is tentatively committed to a *complete* plan by C.

Slice 3:

- 1A is marked as tentatively satisfying g1 by A.
- 5B is marked as tentatively satisfying g1 by B.
- Agent B adds the constraint (¬ g1) to 11B.
- Agent D adds the constraint (¬ g1 via C) to 13D. (Note that in this example, the new constraint on 13D is, in fact, redundant. In other examples, with a more complex set of goals, new constraints propagated in this way often provide additional information.)

Slice 4:

- No changes.

Slice 5:

- Agent B adds the constraint (¬ g1) to 8B.
- Agent B propagates the constraint (¬ g1) on 8B and 11B to their parent, g2. *Agent B now knows the problem is overconstrained.*

Slice 6:

- Agent D modifies the constraint (¬ g1) on goal g2* to (¬ g1*). *Agent D now knows the problem is overconstrained.*

This example illustrates ways in which knowledge is integrated into the local feasibility tree as it is acquired through negotiation. It shows how knowledge aggregated at the level of plan segments can be propagated in drawing inferences concerning interactions at the goal level. It also shows how the network of agents can become aware that it has an overconstrained problem.

6 CONCLUDING REMARKS

In this paper, we have presented a new paradigm for cooperation in distributed problem solving systems. This paradigm incorporates features found in two cooperation strategies treated in the literature: the contract net protocol [2,3,4] and the FA/C paradigm [5]. It has been devised to permit an agent in a distributed problem solving system to acquire enough knowledge to reason about the impact of local activity on nonlocal state and to modify its behavior accordingly.

Three characteristics of distributed planning problems motivate development of a more general cooperation paradigm. First, subgoal interaction problems that arise in the context of a distributed planning system in which agents do not have a global view are very difficult to detect and even more difficult to handle in a reasonable way. Second, many application domains embody planning problems that are overconstrained. When these planning problems are addressed by a network of planning agents, it is essential that the system be able to determine whether or not the problem is overconstrained. Third, when the planning problem is overconstrained, it is necessary for the agents involved to arrive at an agreement as to a set of goals whose satisfaction is regarded as an acceptable solution to the problem at hand. None of these issues can be resolved in the context of the previously proposed cooperation paradigms without the exchange of sufficient knowledge as to permit each agent to construct a global view.

Another factor motivating formulation of a more general cooperation paradigm is the observation that many application domains have characteristics that distinguish them from other multi-agent planning problems which have been investigated. The strategies suggested by Lansky [11] and Georgoff [12] dealing with planning for a multiple agent domain by a centralized planner are not applicable in situations where there is no central planner. In addition, the agents in our networks are not motivated purely by self interest. They are interested in cooperating to achieve some goals pertinent to system performance. For this reason, the metaphor proposed by Genesereth and others [13] does not represent the domain characteristics. It should be noted, however, that our metaphor can be adapted for use in networks of agents which are selfish (as long as they do not lie a great deal).

The mechanisms presented in this paper are related to the techniques that have been utilized in conventional planning systems. Each agent in our system builds a data structure analogous to the Table of Multiple Effects used by NOAH [14] and NONLIN [15] in detecting subgoal interactions. This structure is incrementally built using knowledge gleaned through negotiation. In detecting and resolving conflicts, a form of criticism analogous to that performed by NOAH's Resolve Conflicts critic is employed. Criticism is necessary in our distributed problem solving systems for the same reason it was needed in NOAH - decisions are made initially based on local criteria, whereas nonlocal conditions affect the viability of those decisions. Unlike NOAH (and like NONLIN), alternatives are not discarded after they have been rejected. Backtracking in the form of revised tentative commitment is a feature of the protocol.

In many planning problems, the constraints arising from resource availability are very important in determining a satisfactory solution to the planning problem. We have found that resource constraints play a crucial role in our system as well. The ability to reason about resources is critical in determining adequate solutions. This was recognized in the design of SIPE [16]. Since we have no central planner, the mechanisms for reasoning about resources are somewhat different from those employed in SIPE, but resources as a factor in problem solving are just as important to multistage negotiation as they were in SIPE.

The distributed planning system discussed in this paper is currently in the final stages of implementation on an existing distributed system simulation facility [17].

References

[1] J. G. March and H. A. Simon, **Organizations**, Wiley, 1958.

[2] Reid G. Smith "The Contract Net Protocol: High Level Communication and Control in a Distributed Problem Solver," **IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-10, no. 12**, December 1980.

[3] R. G. Smith and R. Davis "Frameworks for Cooperation in Distributed Problem Solving," **IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-11, no. 1**, January 1981, pp. 61-70.

[4] R. Davis, and R. G. Smith "Negotiation as a Metaphor for Distributed Problem Solving," **Artificial Intelligence, vol. 20, no. 1**, January 1983, pp. 63-109.

[5] V. R. Lesser and D. D. Corkill "Functionally Accurate, Cooperative Distributed Systems," **IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-11, no. 1**, January 1981, pp. 81-96.

[6] S. E. Conry, R. A. Meyer, and J. E. Searleman "A Shared Knowledge Base for Independent Problem Solving Agents," **Proceedings of the Expert Systems in Government Symposium**, IEEE Computer Society, McLean, Virginia, October 1985.

[7] A. S. Tanenbaum, **Computer Networks**, Prentice-Hall, 1981.

[8] S. Vinter, K. Ramamritham, and D. Stemple "Recoverable Communicating Actions in Gutemberg," **Proceedings of the International Conference on Distributed Computing Systems**, May 1986.

[9] E. W. Dijkstra and C. S. Scholten "Termination Detection for Diffusing Computations," **Information Processing Letters, vol. 11, no. 1**, August 1980, pp. 1-4.

[10] N. Francez "Distributed Termination," **ACM Transactions on Programming Languages and Systems, vol. 2, no. 1**, January 1980, pp. 42-55.

[11] A. L. Lansky "Behavioral Specification and Planning for Multiagent Domains," Technical Note 360, Artificial Intelligence Center, SRI International, Menlo Park, California, November 1985.

[12] M. Georgeoff, "Communication and Interaction in Multi-Agent Planning," **Proceedings of the National Conference on Artificial Intelligence (AAAI-84)**, August 1984, pp. 125-129.

[13] M. R. Genesereth, M. L. Ginsberg, and J. S. Rosenschein "Cooperation Without Communication," **Proceedings of the National Conference on Artificial Intelligence (AAAI-86)**, August 1986, pp. 51-57.

[14] E. D. Sacerdoti **A Structure for Plans and Behavior**, Elsevier-North Holland, New York, 1977.

[15] A. Tate "Project Plannig using a Hierarchic Non-Linear Planner," Research Report No. 25, Department of Artificial Intelligence, University of Edinburgh, August 1976.

[16] D. E. Wilkins, "Domain-Independent Planning: Representation and Plan Generation," **Artificial Intelligence, vol 22**, pp. 269-301.

[17] D. J. MacIntosh and S. E. Conry, "SIMULACT: A Generic Tool for Simulating Distributed Systems" to appear in the **Proceedings of the Eastern Simulation Conference**, April 1987.