Group Decision and Negotiation, 2:161–191 (1992)
© 1992 Kluwer Academic Publishers

Decentralized Negotiation: An Approach to the Distributed Planning Problem

THERESA A. MOEHLMAN
Department of Computer Science, University of Massachusetts, Amherst, MA 01003

VICTOR R. LESSER
Department of Computer Science, University of Massachusetts, Amherst, MA 01003

BRANDON L. BUTEAU
Planning Research Corp., McLean, VA 22102

Abstract

DENEGOT is a distributed planning framework that bases conflict resolution on decentralized peer negotiation. Negotiation is viewed as a distributed search through potential compromises. The framework assumes that a satisficing solution is acceptable (a reasonable assumption in many complex domains). To estimate the quality of potential solutions, the negotiation search space is structured into a lattice of sets of potential compromise solutions based on hard constraints. A solution in a higher set in the lattice, if it is achievable, will be preferable over a solution in a lower set. Agents first search under the hard constraint level representing the highest quality solution standard achievable in the current situation. By relaxing hard constraints, the set of compromises that qualify as a solution are enlarged. Agents search for a resolution under the relaxed hard constraint set when a solution cannot be found under the current set of constraints. The framework consists of three iterative problem-solving phases: coordinated search, negotiation state analysis, and constraint relaxation. The application of the DENEGOT framework to distributed planning problems in two domains is demonstrated.

Key words: cooperative distributed problem solving, peer negotiation, distributed planning and scheduling, satisficing search, decentralized conflict resolution

1. Introduction

Planning and scheduling in a distributed, multi-agent environment are difficult. A Cooperative Distributive Problem Solving (CDPS) approach can offer many advantages, including faster processing speed, increased system reliability, and decreased communication bandwidth requirements. For many complex, real-world

This work was partly supported by the Defense Advanced Research Projects Agency (DARPA), monitored by the Office of Naval Research under contract N00014-89-J-1877, by the Office of Naval Research under a University Research Initiative grant, number N00014-86-K-0764, and by the PRC Independent Research and Development project, number R00911.DPS.

planning and scheduling applications, these advantages are important if not imperative. In order to realize such benefits, however, a model of interdependent conflict resolution is required. The DENEGOT (*D*istribut*E*d *NEGOT*iation) framework provides a CDPS planning model that bases conflict resolution on decentralized peer negotiation. In this article, we present the DENEGOT framework and demonstrate its application to two distributed planning problems.

When a conflict arises among agents in a decentralized environment, they must work together to reach a mutually acceptable resolution. To achieve such a resolution, agents need to propose, evaluate, and modify compromises so that they are acceptable to all agents involved. For this type of cooperative problem solving, negotiation is a well-suited approach. Negotiation is viewed as a distributed search through potential compromises where each agent brings into the negotiation specific constraints on what it considers an acceptable resolution. If the composite search space (the space made by combining the local search spaces of agents) is large with complex interdependencies among partial solutions, the distributed search to find an optimal compromise can be resource intensive in terms of both the amount of computation used by agents to perform local search and the amount of information exchanged among agents. In the approach developed here, we reduce the level of effort required to find a solution by taking a satisficing view of solution acceptability. We are interested in finding near-optimal compromises that can be achieved with tolerable amounts of computation.

Our approach to achieving a satisficing distributed search involves conceptually partitioning the composite search space into a lattice of sets of potential compromise solutions. A solution in a higher set in the lattice, if it is achievable, will be preferable over a solution in a lower set. Solutions within a set may be further ordered in terms of their preference. This conceptual partitioning of the composite search space is implemented in terms of partitions of each agent's local search space that mirror the structuring of the composite search space. The distributed search for a solution is based on finding the highest set in the lattice in which a compromise solution acceptable to all agents can be constructed. Focusing the search on a set lower in the lattice can be thought of as making additional compromises in order to find a mutually acceptable solution. Lander, Lesser, and Connell (1991a, 1991b), in their work on negotiation among cooperating experts, take a similar view about how to structure the search space.

We call the criteria used to define solution membership in a specific set in the lattice *hard constraints*, and we call those that relate to solution preference within a set *soft constraints*. We use the term hard constraints to indicate that these constraints are not relaxed as an implicit result of the local search process of agents but rather are relaxed only as a result of agents explicitly recognizing that a mutually acceptable solution to all agents cannot be found under these constraints. These hard and soft constraints which are predefined and domain specific can be based on the same, overlapping, or very different characteristics of a solution. In the cases that they are based on some of the same characteristics, the hard constraints specify ranges over these characteristics. Hard constraints need to be sufficiently discriminating in order to partition the search space into a manageable number of discrete solution sets. Keeping the number of solution sets to a manageable level facilitates the search for a compromise solution since it reduces the number of solution sets that need to considered when choosing the next focus of the search.

This is a satisficing view of search since there is no guarantee that (1) the set with the best solution is chosen to be focused on next when there exist multiple sets at the same level in the lattice, (2) a better solution within the same set does not exist, and (3) all solutions in a set will be evaluated before a decision is made to search for a solution in a lower set in the lattice. In addition, for some domains it may not always be true that a higher set always contains a better solution.[1] One of the key issues in implementing such a distributed search involves coordinating the local searches of agents so that the most preferable acceptable solution, if it exists in the current composite set, is found or one close in preference is found. Another important issue involves how the agents decide among themselves when it is likely that a compromise solution cannot be found in the solution set currently being focused on based on the examination of all plausible compromise solutions. A similar issue surrounds the decision of what lower quality solution set should be next focused on in order to find the most reasonable compromise solution. We feel that the exchange among agents of meta-level information about the state of their local searches is an important ingredient in effective solutions to these issues. The DENEGOT framework reflects these key issues since it consists of three iterative problem-solving phases: coordinated search, negotiation state analysis, and constraint relaxation.

Our framework arose out of exploring distributed planning in two complex domains: fire fighting in Phoenix (Cohen et al. 1989; Moehlman and Lesser 1990) and crisis mission planning for the military. In section 2, we expand on the benefits of adopting a decentralized CDPS approach for planning in complex domains. Although few implemented systems use a decentralized organization, many research efforts helped us conceptualize the distributed planning problem. This related research is discussed in section 3. Section 4 presents the effort for the Phoenix domain. This work focused on the communication and coordination aspects of negotiation. After analyzing the results from Phoenix and examining distributed planning as it occurs in a new domain, the analysis phase of the model was expanded. Section 5 outlines the expanded framework. In section 6, the crisis mission planning work is presented. We conclude with a general evaluation of DENEGOT and future research directions.

2. Why decentralized CDPS?

Among the difficulties encountered in the application of artificial intelligence planning and scheduling technology to complex real-world problems are the closed-world assumption and the scaling-up problem. CDPS holds the potential to overcome these obstacles (Durfee, Lesser, and Corkill 1989). A CDPS agent must realize that it is not working in isolation but within an agent community. Hence, it cannot assume that it has a complete view of the problem-solving world nor can it assume that its view is consistent with other agent views. Moreover, planning algorithms tend to be computationally intensive. In a CDPS system, agents can work in parallel on relatively small pieces of the overall problem. Thus, by exploiting parallelism and the natural task decomposition of a domain, CDPS systems can scale up to large, complex planning problems.

To assert that a natural task decomposition typically appears in complex domains does not mean to imply that the tasks will be entirely independent. Invariably, conflicts among tasks will arise during problem solving. When interdependent conflicts occur in decentralized environments, all involved agents must agree on an acceptable resolution. To be practical, the time spent coordinating interactive problem solving in a decentralized system should not remove the gains of parallelism during independent problem solving. Moreover, without a nonlocal view, it is harder to guarantee that the resulting solution is consistent and of high quality. From a Gestalt perspective, the sum of good local solutions may differ from a good overall solution.

While a centralized or hierarchical agent organization can resolve some of the issues arising from decentralization, many advantages gained by decentralization may be lost. In a centralized system, one agent maintains a global view on the problem-solving situation and acts as a central decision maker. If that agent is lost, the system is disabled. In contrast, decentralized agents are peers and work without a global view; if one agent is lost, problem solving can continue. Thus, decentralized systems offer the potential for graceful degradation. Moreover, many application environments have a limited communication bandwidth. To retain a global view, a central agent may need to receive extensive amounts of data. In a decentralized system, agents need only post data relevant to the specific problem-solving interaction. Thus, decentralized agents can limit the amount of information communicated. Finally, a decentralized agent organization fully exploits a domain decomposition by localizing authority for related problem-solving activities. In many domains, this localization expedites development, testing, and maintenance of a system and is consistent with human organization imperatives.

3. Related research

Planning research seems to have always been plagued by the scaling-up problem. In 1974, Sacerdoti called the problem a "combinatorial quagmire." The "combinatorial explosion" is noted as a basic problem of planning in the *Handbook of Artificial Intelligence* (circa 1982). In 1990, Linden referred to the scaling-up problem as the "computational intractability of planning." Many researchers have tried to overcome this problem. The linear assumption avoided many problems by assuming that the order of subgoal achievement is immaterial; however, it proved to be invalid in most domains. Specialized planning methods can be employed but this approach leads to brittleness. Some researchers have turned to reactive planning. However, in domains such as crisis mission planning, where the actions of hundreds of troops must be coordinated, strict reactive planning is not an option.

To overcome some of the limitations of costly planning algorithms, Lansky (1990) has utilized the natural task decomposition of a domain. Lansky's planning system, GEMPLAN, conducts several local searches in smaller spaces (areas of locality) rather than one search in a large space. Through both a complexity analysis and empirical results, Lansky has found that localization can lead to substantial performance gains. Since GEMPLAN is a serial system, implicit assumptions can be made about problem-solving coordination. For instance, by assuming that a global viewpoint exists and problem solving can be serialized, GEMPLAN can guarantee search consistency when subgoal interactions occur by inhibiting problem solving until all affected areas of the local search spaces have been examined. In a decentralized distributed system, where no global viewpoint exists and where agents work concurrently, such assumptions cannot be made.

When interdependent subgoal interactions arise, DENEGOT uses negotiation as a basis for conflict resolution. Negotiation has often been suggested as an approach to CDPS (Adler et al. 1989; Conry, Meyer, and Lesser 1988; Davis and Smith 1983; Laasri, Laasri, and Lesser 1991). Of the systems that apply this approach, many use a centralized mediator (Sathi and Fox 1989; Steeb et al. 1988; Sycara 1989; Werkman 1990). However, as noted above, a centralized agent represents a communication bottleneck and a single point of failure. In recent work on negotiation (Conry et al. 1991; Klein 1991; Lander, Lesser, and Connell 1991a; Sycara et al. 1991; von Martial), a decentralized approach has been employed. While these approaches have similarities to our work, we believe that the domains explored in our work present different issues and hence led to a different approach on how to conduct negotiation and how to control the negotiation process.

Although we do not use centralized mediation, we have adopted some of the concepts from the constraint-directed negotiation work of Sathi and Fox (1989). Domain features can determine the quality of a solution. An instantiation of those features (constraints) defines a range of acceptable solutions. If an impasse is reached when trying to find a solution under a particular constraint set, constraints can be relaxed, thereby enlarging the number of solutions that are considered acceptable. Thus, a solution space and, therefore, search levels can be structured by constraint sets.

4. The distributed planning problem in Phoenix

Our first exploration of the decentralized distributed planning problem involved cooperative planning during overconstrained resource situations in the Phoenix fire-fighting domain (Cohen et al. 1989; Moehlman and Lesser 1990). In this section, we first discuss the distributed planning problem as it occurs in Multi-Fireboss Phoenix.[2] We then introduce the DENEGOT framework. Finally, we present an implementation example and an analysis of the results.

4.1. *The multi-fireboss distributed planning problem*

Planning in the Phoenix domain involves bringing about the actions needed to contain fires that occur in a model of Yellowstone National Park. Phoenix consists of a detailed simulation of fire and environmental conditions, agents (firebosses) who plan fire attacks, and resources that implement fire attacks. Bulldozers were the primary fire-fighting resource at the time our research was conducted; they contain fires by building firelines around their perimeters. Plans using multiple bulldozers can be constructed for building a fireline. In Multi-Fireboss Phoenix, each agent is responsible for containing fires that occur in its assigned geographical area and owns a set of bulldozers to use for those fire attacks. Each bulldozer has a unique owner. When a new fire is spotted in an agent's local area, it computes a fire projection that specifies the fireline that must be built and the time by which it must be built to contain the fire. An agent then constructs a bulldozer allocation schedule. A schedule is a solution if it builds the requisite fireline by a certain time. The system objective is to minimize the land loss caused by all fires that occur.

Since Phoenix is a real time domain (i.e., fires continue to burn during problem solving), agents search to quickly find a satisficing solution rather than spending the time to find an optimal one. To help estimate loss in Multi-Fireboss Phoenix and to structure the negotiation process, a fire priority classification has been introduced. Priority classes define ranges of land loss levels. The lower the priority class in which a fire is contained, the less loss caused by the fire. Thus, agents try to contain fires within the lowest priority classes possible. When a new fire is observed by a fireboss agent, a fire projection is constructed by that agent which indicates the lowest priority class in which the fire can be contained. This priority class is locally assigned to the fire by the observing agent. Thus, in this application domain, the hard constraints used in our model to partition the composite search space are defined in terms of priority classes.

An ideal attack plan (schedule) for a fire specifies the minimum number of bulldozers, allocated immediately and for the duration of the attack, needed to contain the fire within its assigned priority class based on the initial fire projection. The assumption behind this ideal attack is that immediately and fully fighting a fire with the necessary resources will minimize the amount of land destroyed by the fire and the total amount of bulldozer time needed for constructing the fireline. While an agent tries to achieve the ideal attack for a fire, other attacks may qualify as a solution in terms of containing the fire before it grows sufficiently large to be classified as a more severe and therefore higher priority fire. These other attacks represent less optimal solutions because of slightly more land loss, resource utilization, and timing considerations. Thus, soft constraints in our model are defined in terms of these considerations. Note that the land loss characteristic of a solution is used for defining both hard and soft constraints.

We recognize two general deviations from the ideal attack: delay the attack start time and begin the attack with a fewer number of bulldozers than required. A third potential deviation, though not implemented, is to begin the attack with more bulldozers than needed and assume that some of those bulldozers will leave the attack before it is completed.

Interaction between agents occurs when a single agent is in a potentially overconstrained resource situation. When a new fire occurs in an agent's area and it does not have enough local bulldozers available to implement the ideal attack, it initiates negotiation by requesting the needed bulldozers. During negotiation, agents search for a solution to the new fire. Search involves modifying the current attack schedules to release bulldozers immediately or in the future, thereby creating bulldozer schedule entries to offer for the new fire attack. Figure 1 provides a composite view on the Phoenix distributed planning problem. A Fire Priority Configuration defines a set of possible solutions in terms of alternative strategies and bulldozer schedules for fighting a set of fires so that they never grow out of specified land loss levels. Agents first search for a bulldozer distribution under the lowest possible loss priority configuration. If the agents cannot find a distribution that qualifies as a solution, they must incur more loss. By allowing some fires to burn into a higher priority class before being contained, an enlarged set of bulldozer distributions qualify as solutions to this less acceptable fire priority configuration for the current fire situation. The agents can then search for a solution under this revised criteria for acceptable priority configuration. Figure 2 displays an agent's local viewpoint. From a local perspective, the higher quality solutions under both the hard and soft constraints may not correspond to those of the composite viewpoint.

In our model, when an agent has insufficient local resources for an optimal fire attack plan, it immediately requests the loan of these missing resources from other agents without considering whether it can achieve a nonoptimal but still acceptable solution using its own local resources. Clearly, if other agents have available resources, an optimal solution can be found quickly. The next question is what actions should be taken if other agents cannot offer those resources. In our model, negotiation is entered. Alternatively, agents could dynamically decide, based on a rough model of local resource availability, if negotiation should be entered (and which agents to engage) or if the initial agent should first search for a nonoptimal solution in its local space. In our model, the solution quality is likely to be higher since all agents contribute but more negotiation may be performed and more agents may participate than is necessary. Depending on the application, negotiation may lead to significant delays in agents' finding mutually acceptable solutions because of time spent waiting for communicated information and time spent on additional local search. The important question is whether these delays lead to resources being inefficiently used during the time spent finding a better solution.

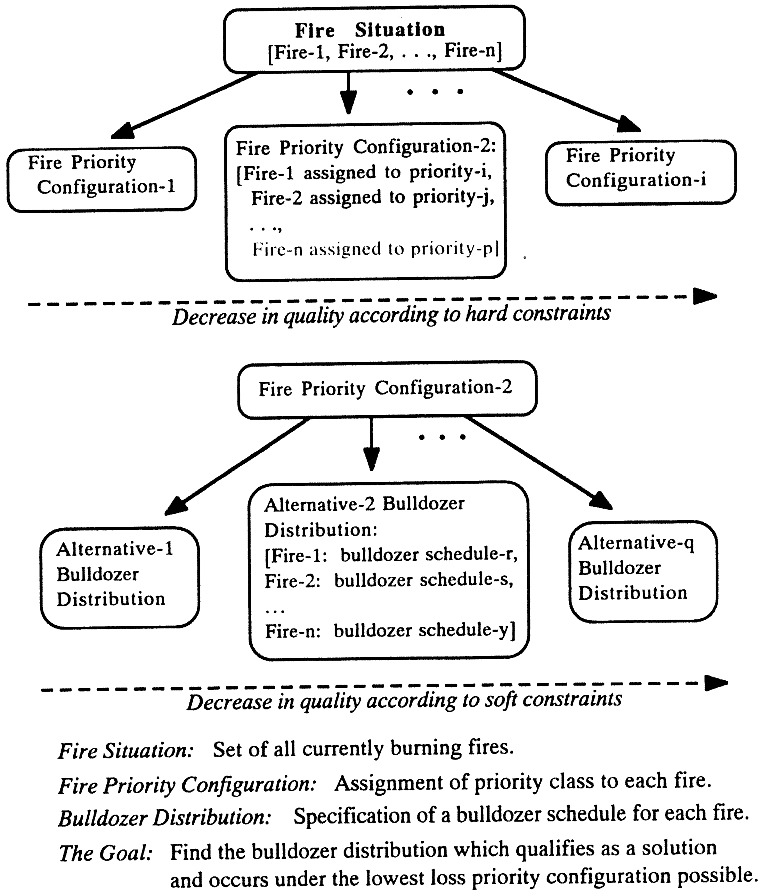


Figure 1. The global view.

If this is the case, then quickly accepting a nonoptimal solution locally may be preferable even though this can lead to inefficient use of nonlocal resources and loss of quality in the overall solution. Therefore, the decision to engage in negotiation in the hope of achieving a better overall resource utilization plan and solution quality must be carefully evaluated based on domain considerations and the current state of the composite search.

4.2. *In search of coherence*

In our distributed planning model, each agent is assigned a set of goals. An agent works to achieve its goals in a local manner. As shown in Figure 3, during this independent problem-solving state, agents work in parallel on their individual

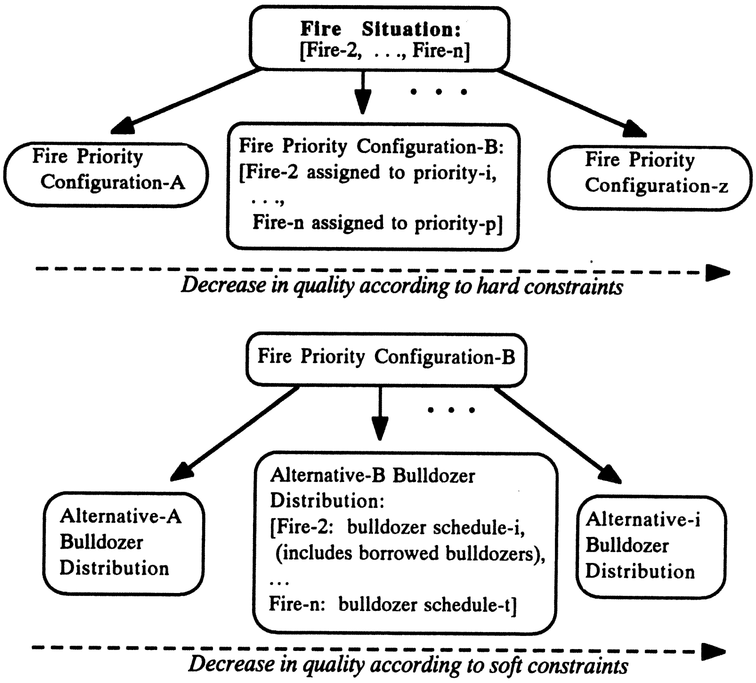


Figure 2. A local view.

State A: Concurrent Local Problem Solving
- Agents work independently to achieve their assigned tasks
- Relatively little interaction between agents

State B: Coordinated Problem Solving (Negotiation)
- Agents work together to resolve current conflict
- Sufficient interaction among agents to reach a resolution

Figure 3. Distributed problem-solving states.

tasks. When an agent lacks the local means to accomplish a goal, it initiates negotiation (coordinated problem solving).

Each agent brings into the negotiation hard constraints on solution acceptability. The composition of these constraints defines the negotiation search level (set in the lattice). In Multi-Fireboss Phoenix, each agent has assigned a priority class to each fire in its area. The set of all priority classes defines a priority configuration. Agents first search for a solution under this hard constraint set. Within this set, soft constraints determine the quality of a solution. To reach a solution under a specified set of hard constraints, agents may have to relax soft constraints. For instance, an agent may have to achieve a local goal with less efficiency by requiring more bulldozer time than might otherwise be possible if bulldozer resources were immediately available at the time the fire was spotted. This lack of timely availability of resources could also lead to more land loss. Hopefully there will not be significantly more land loss so as to force the fire into a higher priority class and thus violate the hard constraints associated with the current search level. Since relaxing hard constraints means incurring significantly more land loss, agents first try to exhaust all possibilities of finding a solution at a given search level before resorting to hard constraint relaxation.

Using constraints to structure the negotiation process, three problem-solving phases can be identified: search directed by negotiation, analysis of the negotiation state, and relaxation of negotiation constraints.[3] The framework is outlined in Figure 4. Phases I and II represent a distributed search. During phase I, agents perform a local search. The result of a local search can be a solution (the agent can meet all of the constraints), a partial solution (the agent can meet some but not all of the nonlocal constraints), or no solution (the agent can meet its local constraints but it cannot meet any of the nonlocal constraints). If a solution is offered, the conflict is resolved. If no solution is offered, compositions of partial solutions (i.e., compromises) are examined. If an acceptable compromise is found, the conflict is resolved. If no resolution is generated, the agents enter phase II to assess the space searched before relaxing hard constraints. Since the search is distributed and no global viewpoint exists, an explicit analysis phase

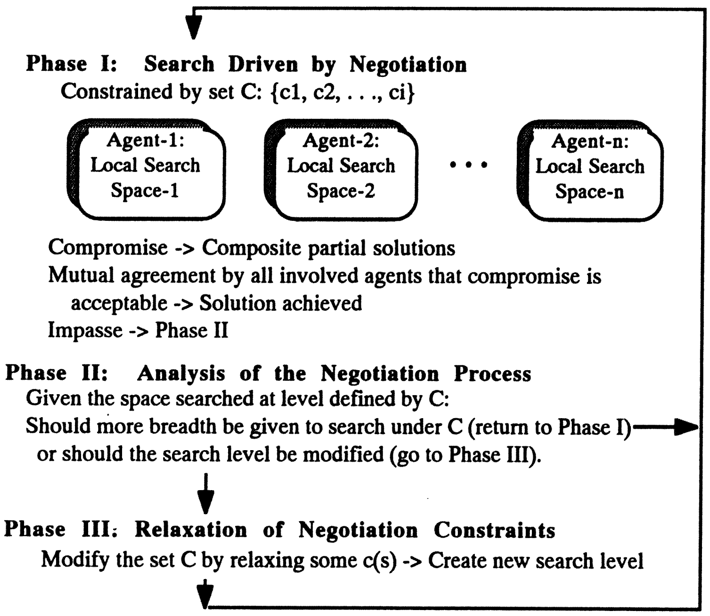


Figure 4. The coordination framework.

enables the agents to evaluate how much of the composite search space has been examined and to direct the local activities of the phase I search.

If the agents have exhausted their means of finding a solution based on the criteria specified in the current search level, they enter into phase III. In this phase, the agents determine which constraints to modify. Depending on the system goals, various strategies can be used in this phase. In Multi-Fireboss Phoenix, agents try to delay the fewest and smallest fires as a strategy to minimize land loss. Once constraints have been relaxed, distributed search is re-entered.

Although we have been studying *cooperative* problem solving, the coordination framework does not assume this type of interaction. Cooperation relates to the willingness of agents both to offer partial solutions and to relax their constraints for a nonlocal benefit. In a *competitive* environment, an agent may place a condition on a loan such as requiring the borrowing agent to commit some of its resources at a later time. Competitive agents evaluate compromises using local criteria whereas cooperative agents evaluate compromises using a combination of local and nonlocal criteria. Thus, a cooperative agent may be willing to incur local loss if it benefits other agents. A future research area involves comparing cooperative and competitive interactions.[4]

4.3. The communication protocol

During phase I, agents search their local spaces for partial solutions. Search is initiated when one agent issues a request. In other work, the communication protocol allowed an agent to make a positive or negative response to a request, perhaps with an explanation (Conry et al. 1991; Sycara 1989). Another option is to allow alternative suggestions. While an agent may not be able to meet a request totally, it may be able to meet it partially.

In Multi-Fireboss Phoenix, agents request the use of bulldozers for fire attacks. One agent issues a request to other agents for a bulldozer loan during a specified time period. While a responding agent may not be able to meet the entire request, it may be able to offer a fewer number of bulldozers, a later start time on the loan, or a restricted completion time on the resource use. Combining partial solutions can lead to a compromise that may not have been found or may have taken longer to find if the requesting agent was required to issue requests iteratively.

In Figure 5, the communication protocol of DENEGOT is shown in the context of phase I problem solving. During phase I, a request is issued to all agents involved in the interaction. The request indicates the activities or resources needed to reach a solution. A responding agent, after searching to meet that request, may offer a positive response ("yes, I can meet that request"), an alternative suggestion ("while I cannot meet the entire request, I can offer a partial solution"), or a negative response ("no, I cannot meet the request"). The composition of the responses may lead to a solution. If the composite response is not a solution and if there is a modification to it that will make it a solution, a modified request is

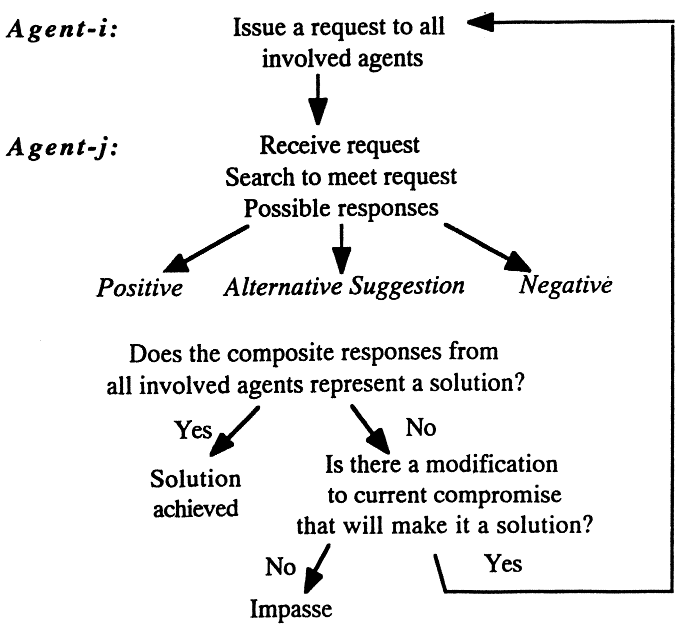


Figure 5. Search dialogue.

issued. Alternative suggestions and modified requests represent a distributed depth first search. An impasse is reached when all possible modifications have been examined. In this case, phase II is entered to determine if additional search under the current hard constraint set is warranted or if phase III should be entered.

4.4. The implementation

In the implementation of DENEGOT for the Phoenix domain, we simplified problem solving by using a two-agent configuration (rather than a multi-agent configuration). Furthermore, we assumed that an agent was involved in only one negotiation session at any given time. In this section, we present an example to illustrate the framework and analysis of the results.

The example shows the negotiation dialogue between the two agents, Disfireboss-1 and Disfireboss-2. There are six bulldozers in the example; bulldozers 1, 2, and 3 are owned by Disfireboss-1 and bulldozers 4, 5, and 6 are owned by Disfireboss-2. Figure 6 shows the initial situation starting on 8/1 at 12:00 which is considered time 0 for the simulator.[5] Two medium priority fires occur: actual-fire.5 (at time 13:04) and actual-fire.6 (at time 13:35). Independently, the agents find solutions for the fires which will take over 24 hours to contain. When actual-

TIME	DISFIREBOSS-1	DISFIREBOSS-2
13:04	New fire - actual-fire.5 spotted	
	Priority: medium	
13:10	Computing Projection for actual-fire.5	
	End time: 8/2 14:07 (94072)	
	Needed Bulldozers: 2	
	Can fight actual-fire.5 with my resources	
13:13	Adding agent-goal.5 for actual-fire.5	
	Bulldozers (bulldozer-2 bulldozer-1)	
	Start time 4404; end-time 94072	
13:35		New fire - actual-fire.6 spotted
		Priority: medium
13:38		Computing Projection for actual-fire.6
		End time: 8/2 14:36 (95775)
		Needed Bulldozers: 2
13:39		Can fight actual-fire.6 with my resources
13:40		Adding agent-goal.6 for actual-fire.6
		Bulldozers (bulldozer-5 bulldozer-4)
13:41		Start time 6044; end-time 95775
14:35	New fire - actual-fire.7 spotted	
	Priority: medium-low	
14:39	Computing Projection for actual-fire.7	
14:40	End time: 8/2 15:37 (99438)	
	Needed Bulldozers: 2	
	Not enough resources for actual-fire.7	
	Trying resource loan	
15:27		Received Request for
		1 bulldozers; medium-low priority
		Can honor request
15:29	Received POSITIVE from disfireboss-2	
15:34		Loaning bulldozer bulldozer-6
15:35	Now in charge of bulldozer-6	
	Adding agent-goal.7 for actual-fire.7	
	Bulldozers (bulldozer-6 bulldozer-3)	
	Start time 12915; end-time 99438	

Disfireboss-1: Current Goals:
[Actual-fire.5, medium, (Bulldozer-2, Bulldozer-1) allocated for complete time]
[Actual-fire.7, medium-low, (Bulldozer-6 [on loan], Bulldozer-3) allocated for complete time]
Idle Bulldozers: 0

Disfireboss-2: Current Goals:
[Actual-fire.6, medium, (Bulldozer-5, Bulldozer-4) allocated for complete time]
Idle Bulldozers: 0

Figure 6. Initial situation of example.

fire.7 occurs (at time 14:35), Disfireboss-1 does not have enough local idle bulldozers to implement the ideal attack. So Disfireboss-1 asks Disfireboss-2 for one bulldozer to start immediately. Since Disfireboss-2 can meet the request, it sends a positive response. Thus, the ideal fire attack for actual-fire.7 has been achieved.

The example continues in Figure 7. Another fire, actual-fire.8, is spotted in Disfireboss-2's area (at time 17:40). Since Disfireboss-2 has no local resources available, it initiates negotiation by requesting 2 bulldozers to start immediately. Disfireboss-1, who has no idle bulldozers, offers an alternative suggestion of two bulldozers starting at a later time (94072). These bulldozers will be available when

TIME	DISFIREBOSS-1	DISFIREBOSS-2
17:40		New fire - actual-fire.8 spotted Priority: medium-high
17:44		Computing Projection for actual-fire.8 End time: 8/3 3:01 (140493) Needed Bulldozers: 2
17:45		Not enough resources for actual-fire.8 Trying resource loan
17:48	Received Request for 2 bulldozers; medium-high priority Alternative - lower priority goal Delay Start time to 94072	
17:49		Received Alternative from disfireboss-1 Schedule will not work
17:50		Getting earliest time bulldozers released Bulldozers (bulldozer-4 bulldozer-5) Time available 95775 Evaluating Schedule ((2 94072 97240) (2 95775 140493)) Schedule inadequate - Planning modified request
17:51		Needed build time 23740.602 to make schedule a solution Modified Request: 1 bulldozers; start-time 77807 Characterized as Delaying the start time
17:55	Received Modified Request	
17:56	Sending reply of negative	
17:57		Received NEGATIVE on modified request
17:58		ENTERING Phase II
17:59		Phase 2 Trying to create base schedule starting with fewer bulldozers Can release 0 bulldozers now
18:01	Received Enter Phase II message - fewer characterization	
18:02	Can release 0 bulldozers now	
18:03		Entering Phase III

Figure 7. Phase I and II search.

the attack on actual-fire.5 is complete. However, the schedule produced by the alternative suggestion does not qualify as a solution (refer to time 17:49). Disfireboss-2 amends the schedule by adding the bulldozers from actual-fire.6; when they complete that attack, however, the schedule still does not qualify as a solution. Disfireboss-2 computes that adding one bulldozer at time 77807 will make the schedule a solution, and it issues a modified request to Disfireboss-1 (refer to time 17:51). Disfireboss-1, unable to meet the request, sends a negative reply. At this point, the agents have reached an impasse and they enter into phase II.

In phase II, the agents determine whether another means exists to find a solution under the current constraint set. In the previous search, the agents searched for a solution characterized by delaying the attack start time. Since a solution type characterized by starting the attack with fewer bulldozers has not yet been attempted, the agents determine if that solution type is possible in the current situation. However, neither agent can release any bulldozers immediately. Since these two characterizations were the only ones implemented, the agents have exhausted their known possibilities of finding a solution under the current constraint set. Phase III is entered.

The example concludes in Figure 8. The agents delay fighting the fewest and lowest priority fires in an attempt to achieve a minimal loss solution. If the new fire is not delayed, a solution must be found to it before another fire is delayed. This strategy tries to prevent bulldozer thrashing (bulldozers spending most of their time traveling to fires without much useful work being done). The agents find that if they delay the lowest priority fire, actual-fire.7, a solution to the new fire is achieved. Hence, actual-fire.7 is allowed to burn into a medium priority class, and the bulldozers assigned to it are re-allocated to actual-fire.8. Disfire-boss-1 then finds a solution for the delayed actual-fire.7 within its own goal set. Thus, a solution to the conflict has been found.

4.5. What did we learn?

After implementing the framework and examining the cooperative problem-solving behavior, it was apparent that a decentralized distributed approach to planning was feasible even though few implemented systems used this type of organization. Using the communication protocol, both agents could actively par-

TIME	DISFIREBOSS-1	DISFIREBOSS-2
18:05	Received Enter Phase III message	
18:09		Neighbor lowest priority: medium-low; Bulldozers freed: 2 My lowest priority: medium; Bulldozers freed: 2 Bulldozers I have available 0 Delay neighbor lower goal
18:18	Loaning bulldozer bulldozer-3 Returning bulldozer bulldozer-6	Back in charge of bulldozer-6 Now in charge of bulldozer-3
18:19	Deleting goal agent-goal.7 Updated goal list: (agent-goal.5)	Adding agent-goal.8 for actual-fire.8 Bulldozers (bulldozer-3 bulldozer-6) Start time 22736; end-time 140493
18:26	Computing Projection for actual-fire.7 End time: 8/3 21:31 (207112) Needed Bulldozers: 1	
18:27	Not enough resources for actual-fire.7 Trying resource loan	
18:48		Received Request for 1 bulldozers; medium priority Negative reply to request
19:09	Received DENIAL from disfireboss-2	
19:10	Getting earliest time bulldozers released	
19:11	Bulldozers (bulldozer-1 bulldozer-2) Time available 94072 Evaluating Schedule ((2 94072 207112)) Schedule works - SOLUTION Using the following of my resources ... bulldozer-1: start 94072; end 207112 bulldozer-2: start 94072: end 207112	

Figure 8. Conclusion of the example.

ticipate in compromise construction. The compromises achieved through negotiation were suitable satisficing solutions. Hence, structuring the solution space in terms of hard and soft constraints is an appropriate approach under the satisficing standard for this domain.

Upon deeper analysis of the cooperative problem-solving behavior, several problems were noted. Agents searched for alternative suggestions in a fixed order rather than taking into account the current situation. Moreover, phase II retroactively examined the distributed search rather than proactively directing the local search activities. In general, the need to expand the distributed search model of the framework was recognized.

5. The expanded DENEGOT framework

After analyzing the results from the Phoenix domain, the need to dynamically direct the local search activities of agents during the distributed search was recognized. In this section, we present the expanded distributed search model. We then describe how the model was incorporated into the DENEGOT framework.

5.1. Decentralized heuristic distributed search

In a heuristic search, features of the problem-solving situation guide the search. In a decentralized environment, no global view exists. If agents are to use heuristics in these environments, they must first gain a more comprehensive view on the problem-solving context. Since there may be communication restrictions and timing constraints, agents cannot simply send raw data across the network. Thus, the agents must abstract information about their local states. The need to abstract local state characterizations has been recognized before by Durfee and Lesser in their work on partial global planning (1986, 1991; Lesser 1991) and by Sycara and her colleagues in their work on distributed scheduling (Sycara et al. 1990, 1991). Lander is also exploring the use of local state characterizations to direct search (August 1991). To determine what information is relevant, an agent can consult its local version of a domain characterization map. Figure 9 shows the general form of a characterization map.[6]

When coordinated problem solving is entered, each agent, from its local viewpoint, creates a local situation assessment report. By combining these reports, agents can gain a nonlocal perspective on the problem context. They can then decide, through negotiation or nonlocal strategies, which solution types are likely to be achieved in the current situation. At the same time, they can eliminate unlikely possibilities from consideration and thus avoid wasting time searching to achieve those solutions. Since there may be many possible ways to achieve a specific solution type, agents must also determine what set of problem-solving actions to take. Based on the situation assessment reports, agents can elect a set

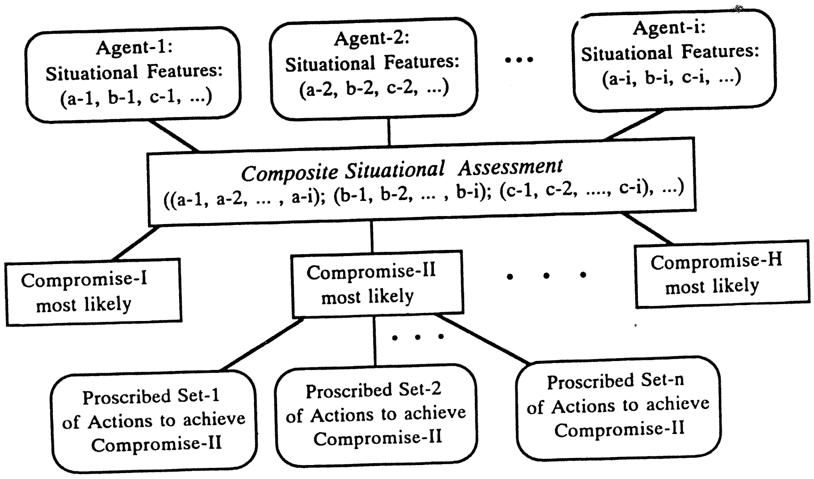


Figure 9. Characterization map.

of local problem-solving activities that directs the distributed search toward high-quality solutions under the soft constraint set. One agent may try to take the complete responsibility for a solution. In some cases, localizing the responsibility for building the solution and managing its execution may be beneficial. On the other hand, responsibility may be shared among agents. In this case, the separation of responsibility must be determined.

Obviously, a solution may not be found during the initial set of uncoordinated problem-solving activities. However, the basis of a solution can be built. Using this base, the agents can enter into phase I and search in more depth on that base. In Multi-Fireboss Phoenix, a base schedule specifies the attack start time and an initial bulldozer allocation. If the base schedule is not a solution, the agents can search to amend the schedule by adding more bulldozers at a later time. If enough bulldozers can be brought to the fire to build the requisite fireline, a solution will be reached.

The characterization map provides the basis for a distributed heuristic search, without a centralized controller or agents' having an in-depth view of other agents activities. It gives the agents a common language, a method to coordinate concurrent local search activities, and a mechanism for monitoring the search state. Thus, without a complete and detailed view of the composite search space, agents have a means to organize local problem-solving actions toward finding an acceptable overall solution.

5.2. *The framework revisited*

To gain an understanding of how the DENEGOT model incorporates the domain characterization map, we present an abstract problem-solving example illustrated with specific instances from Multi-Fireboss Phoenix and the multi-agent crisis mission planning domain (CMP). In CMP, the goal of the distributed planning system is to create a mission plan to resolve a given crisis. A mission plan consists of a set of detailed operation plans; a CMP agent is assigned the responsibility of constructing one of those detailed operation plans which may involve rescuing captives or neutralizing a potential enemy threat that could interfere with the successful rescue. Monitor agents are used to manage global databases and to detect potential conflicts between agent plans. (A more detailed discussion of the CMP domain follows this section.)

In a distributed planning system, agents can concurrently build plans for their given task. Little interaction occurs during this independent problem-solving stage. When an agent's local problem-solving activities potentially have a nonlocal effect, the agent initiates negotiation. In Multi-Fireboss Phoenix, negotiation is initiated by an agent when a new fire occurs in the agent's area and the agent locally lacks the needed bulldozers to implement an ideal attack. In CMP, negotiation is initiated by an agent when it assumes the responsibility for accomplishing a specific plan subgoal or the use of a resource that has previously been reserved by other agents.

Each agent brings into the negotiation hard constraints on solution acceptability. Given these constraints, typically, there is some ideal solution type. In Multi-Fireboss Phoenix, a fire attack that begins immediately at the time the fire is spotted using the maximum resources needed to efficiently contain the fire represents this solution type. In CMP, an ideal solution for a threat conflict is for one agent, using the most severe means (most effective) at the earliest time, to take responsibility for all targets in the threat. Under a hard constraint set, the ideal solution type represents the best solution that can be achieved according to the soft constraints. Thus, agents first search to find this solution type. If that solution cannot be achieved, the agents enter into phase II. Based on situation assessment information collected during the phase I search, the agents can determine which solution types are likely to be achieved and which types can be eliminated from further consideration.

In Multi-Fireboss Phoenix, the domain features that characterize a situation include the start and end times of goals and the bulldozer availability. Figure 10 displays one possible characterization of the Phoenix domain. During situation analysis, agents decide which solution type to try to achieve. If most of the fire attacks have just begun, characterization 4 is likely to lead to a solution whereas characterization 2 is unlikely to lead to a solution and can be eliminated from consideration.

Once the agents agree on a solution type, they then chose a set of local problem-solving activities to take. In most cases, there will be several means to achieve a particular solution type. This step may be relatively straightforward. In Multi-Fireboss Phoenix, agents can concurrently search their local spaces, guided by the solution type, for bulldozer schedule entries. On the other hand, this step may be more complex.

1.	*Characterization:*	Ideal bulldozer schedule.
	Situation:	First starting negotiation.
	Space to be Searched:	Immediately releasing bulldozers from goals.
2.	*Characterization:*	Delay start time of new fire attack.
	Situation:	Goals exist that will be completed relatively soon.
	Space to be Searched:	Using bulldozers upon completion of current assignments.
3.	*Characterization:*	Start new fire attack with fewer bulldozers.
	Situation:	Some bulldozers are currently available.
	Space to be Searched:	Use available bulldozers and search to add bulldozers at a later time.
4.	*Characterization:*	Start new fire attack with more bulldozers.
	Situation:	Some goals are just starting.
	Space to be Searched:	Searching the effects of delaying other goal start times (temporarily loaning bulldozers).

Figure 10. Domain characterization of Phoenix.

For example, in CMP, a timing conflict may have been detected that involves several agent plans. To reach a solution, one agent may need to determine the timing of pivotal activities before other agents can search to modify the timing of their activities.

In this first step, the basis for a compromise is built. If that base qualifies as a solution, the conflict is resolved. If the base does not qualify as a solution, the agents enter into phase I and search in more depth on that base. If an impasse is reached in the search, agents enter into phase II and evaluate the negotiation state. Information uncovered during search may eliminate other solutions types from consideration. The agents may search to achieve another solution type. When the agents have searched for or eliminated all possible solution types, they enter phase III.

Depending on the agent organization and the system goals, different strategies can be applied during constraint relaxation in phase III. For instance, a conservative strategy minimally modifies the hard constraint set as a means to achieve the highest quality solution possible (find the next lowest set in the lattice). In contrast, a more liberal strategy may relax many constraints at once in an attempt to limit the time spent negotiating. Once the agents agree on the constraints to relax, they re-enter the distributed search.

DENEGOT provides a model for decentralized negotiation without dictating specific problem-solving strategies. This flexibility makes the framework applicable to a wide range of planning problems, agent organizations, and problem-solving methods. The framework can accommodate both homogeneous and heterogeneous agent organizations. In Multi-Fireboss Phoenix, planning consists mostly of mathematical computations. In CMP, knowledge-based methods are used. In Multi-Fireboss Phoenix, agents act in a somewhat reactive manner, whereas in CMP agents concentrate on pre-planning. In both domains, however, the DENEGOT model enables the agents to accomplish the necessary interactive problem solving for conflict resolution.

6. The military crisis mission planning problem

The second application for the DENEGOT framework was military crisis mission planning. When a military crisis occurs, a planning team is tasked to construct a mission plan that, when executed, will successfully resolve the crisis. While the military maintains a library of contingency plans, a contingency plan is often a skeletal outline of a mission plan or a mission plan that was successfully implemented in a similar situation. Hence, planning involves expanding a skeletal plan into an executable one or adapting a mission plan to the current situation. Typically, a quick response to the crisis is warranted since human lives may be at stake. Thus, the primary aim is to construct rapidly a plan that will successfully resolve the crisis without concern for creating the optimal plan for the particular situation. In other words, the planners search for a satisficing solution.[7]

Detailed planning in this domain is quite complex. The movement and actions of ground troops, aircraft, and naval vessels all need to be coordinated. In addition, ground equipment may need to be specially transported to the theater of operations, aircraft may need to refuel during the operation, and margins for error must be taken into account. Many activities occur concurrently while others must take place sequentially. Moreover, physical resources are limited to those that can be brought to the theater of operations within the time frame of the mission. Resources such as air space are also limited. While our implementation is modeled after the human military planning, it is necessarily different and is a simplified version of true crisis mission planning.

6.1. The implementation

A mission outline consists of a set of tasks. A planner is assigned to each task. Task plans are interdependent in three main ways: timing, resources, and threat responsibility. The activities of one task plan may restrict the timing of activities in other tasks. Resources may be limited or resource support may be limited (e.g., while transport aircraft may be available, only a limited number can land and unload in any given time period). Finally, no unintended redundancy in threat responsibility should occur between tasks (undetected conflicts in threat responsibility can lead to unnecessary injuries and wasteful resource use).

The planning system is implemented in CAIBL (Buteau 1990), a distributed blackboard environment. Currently, the system runs on a three-machine configuration. An agent in CAIBL is represented by a set of knowledge sources and private data blackboards. There are two basic types of agents: knowledge-based planning agents (specialists) who construct detailed plans for an assigned task and monitor agents who manage global databases and detect potential conflicts between agents' plans. The specialists are heterogeneous: different agents may place different priorities on the timing of activities, the use of overconstrained resources, and the method of dealing with threats.

The particular planning scenario for the implementation concerns the following situation:

> A PAN AM 747 jet, carrying over 300 passengers, was hijacked by a small group of terrorists and flown to Al Bushia, Khaman. The United States military attempted a commando style rescue which failed when the Khamani military intervened. During this operation, approximately 20 military captives were taken prisoner and are now being held at Yukehr Air Base in Khaman. The distributed planning system is tasked to plan two full scale rescue operations and any support operations necessary for a successful release of the hostages and captives.

The system, on being given its assignment, first constructs a set of tasks. As shown in Figure 11, eight tasks are created to accomplish this rescue (the leaves of the tree). A specialist is then assigned to each task. The specialist is to develop a detailed plan to accomplish its task. The planning varies according to the spe-

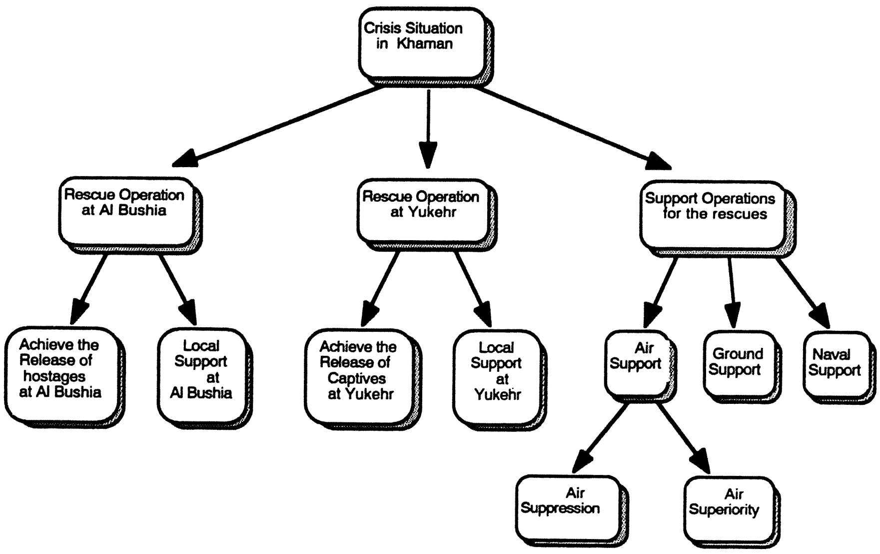


Figure 11. Mission plan outline.

cific situational features. Different resources may be available, different threats may be present, and various rules of engagement can be given to the system.

A specialist first builds a skeletal plan for its task and then employs an opportunistic approach to the detailed planning. Detailed planning involves expanding the skeletal plan steps into executable activities. As part of this process, agents need to allocate resources, restrict the timing of various activities, and target threats. Since these activities potentially involve other agents, planners make reservations with the appropriate monitor. When a monitor detects a conflict, it notifies the agent that a conflict may exist and provides that agent with a list of the involved agents. Our monitor is a simplified version of the resource monitor used by Sycara and her colleagues (1990). If an agent receives a conflict notice from the monitor, it initiates negotiation by announcing to all involved agents that a threat conflict has occurred and indicates the threat in question.

For the purposes of this article, we focus our presentation on threat responsibility conflicts. Figure 12 displays the features of a threat responsibility conflict. A threat consists of a group of related threat-clusters. H-hour refers to the start of the rescue operations. Hard constraints relate to the timing and means of dealing with threat-clusters. Soft constraints relate to the type of solution employed. It is more desirable for one agent to take the responsibility for all threat-clusters of a threat since localization of command reduces communication requirements during plan execution and eases coordination requirements. Redundancy in threat responsibility is least desirable since resources may be limited and timing is severely constrained during plan execution.

In Figure 13, the composite view of the distributed CMP problem is displayed. A negotiation search level is defined by a timing-means configuration. Within a *timing-means* configuration, each threat-cluster is assigned a time and a means to

Threat: {threat-cluster-1, threat-cluster-2, . . ., threat-cluster-n}

Hard Constraints:

Timing: H-hour-, H-hour, H-hour+
(earliest -> latest)

Means: Deception, Visibility, Disability, Neutralization
(least severe -> most severe)

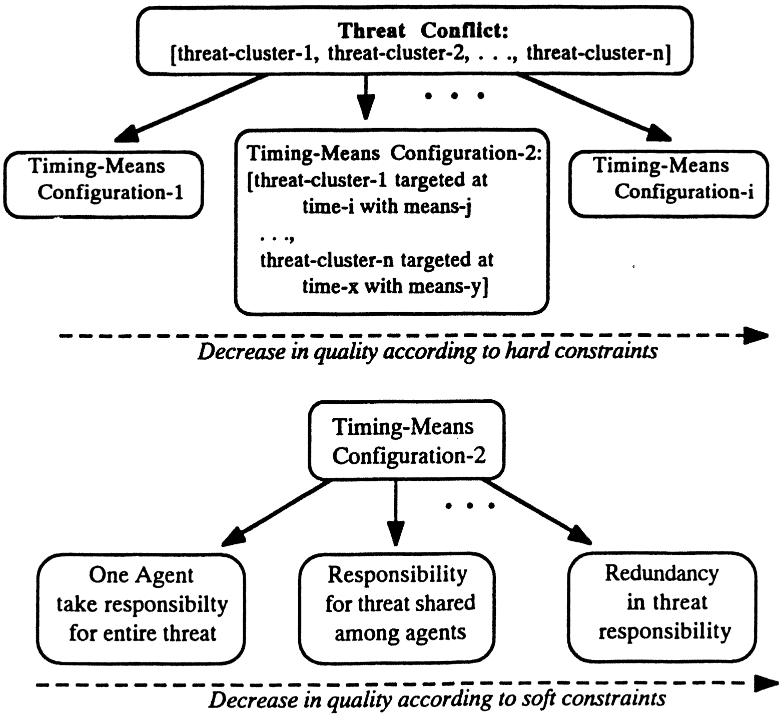
Solution Types (Soft Constraints):

- Single Agent Responsible
- Responsibility Shared Among Agents
- Redundancy in Responsibility

Constraint Relaxation:

- Relax Time on some threat-clusters of threat (Use later time)
- Relax Means on some threat-clusters of threat (Use less severe means)

Figure 12. Threat responsibility conflicts.



Threat Conflict: **Set of threat-clusters.**

Timing-Means Configuration.: **Assignment of time and means to each threat-cluster.**

Solution Types: **Determination of agent responsibility to each threat-cluster.**

The Goal: **Achieve the highest quality solution type under the highest quality timing-means configuration possible.**

Figure 13. The global view.

be targeted. Upon entering into negotiation, each agent announces to the other agents its timing and means constraints on the threat conflict. In Multi-Fireboss Phoenix, the hard constraints did not need to be announced as they were implicit in the bulldozer schedule entries an agent was willing to offer. In CMP, soft constraints are specified in terms of the solution type used. Under a specific timing-means configuration, agents search to achieve the highest quality solution type according to the soft constraints. A solution specifies an agent responsibility for each threat-cluster and the timing and means in which that threat-cluster is targeted. If a solution cannot be achieved under a timing-means configuration, hard constraints are relaxed. When the time or means constraint are relaxed, more agents are able to take the responsibility for threat-clusters, and therefore the space of acceptance solutions is enlarged.

Upon entering into negotiation, each agent sends a report to the other agents indicating the time and means requirements that it has placed on the threat responsibility. The agents first search to achieve the ideal solution for this type of conflict: one agent takes the responsibility for all threat-clusters in the threat using the earliest time and severest means specified in the agent reports. A positive search response indicates that the agent can take the responsibility for all threat-clusters. An alternative suggestion indicates that the agent can take on the responsibility for some of the threat-clusters. A negative search response indicates that the agent can take no threat-cluster responsibility under the requirements.

The example threat conflict concerns a threat consisting of two threat-clusters: air-group-1 and sams-yukehr-1. Three agents are involved: Air-Specialist-1, assigned to the air defense suppression task; Special-Operations-2, assigned to the Yukehr rescue operation; and Support-Operations-2, assigned to the local support task at Yukehr (see Figure 11). Each agent represents the negotiation state with a local conflict object. The initial activity of the conflict is Figure 14. The example is illustrated from the view of Special-Operations-2. One of the agents has received a conflict notice from the monitor and has announced the conflict to the involved agents. Each agent then announced its time and means requirements (refer to the timing and means slots): Air-Specialist-1 indicated disable at H-hour; Special-Operations-2 indicated neutralize at H-hour; and Support-Operations-2 indicated disable at H-hour+. Hence, the earliest time is H-hour and the severest means is neutralize. The current-characterization of the negotiation state is therefore "earliest-time-severest-means."

Since the ideal solution type is for one agent to take complete responsibility for

```
CONFLICTS1 is at the SPECIAL-OPERATIONS-SOLUTION-2.CONFLICTS level.

            Attributes:
              AGENTS-INVOLVED:  (AIR-SPECIALIST-1 SUPPORT-OPERATIONS-2 SPECIAL-OPERATIONS-2)
                CONFLICT-TYPE:  THREAT-RESPONSIBILITY
   CURRENT-CHARACTERIZATION:  EARLIEST-TIME-HARDEST-MEANS
                EARLIEST-TIME:  H-HOUR
                HARDEST-MEANS:  NEUTRALIZE
                        MEANS:  ((AIR-SPECIALIST-1 DISABLE) (SUPPORT-OPERATIONS-2 DISABLE)
                                                            (SPECIAL-OPERATIONS-2 NEUTRALIZE))
                       NUMBER:  0
                          REF:  CONFLICTS1
                      RES-AIR:  0
                   RES-GROUND:  1
                    RES-NAVAL:  0
                     RESPONSE:  ((AIR-SPECIALIST-1 NEGATIVE TIMING) (SUPPORT-OPERATIONS-2 NEGATIVE TIMING)
                                                                    (SPECIAL-OPERATIONS-2 ALTERNATIVE
                                                                                          1
                                                                                          THREAT-CLUSTERS))
                       STATUS:  EXAMINE-ALTERNATIVES
              THREAT-CLUSTERS:  ((AIR-GROUP-1 (AIR-TO-GROUND AIR-TO-AIR) ON-GROUND) (SAMS-YUKEHR-1 GROUND-TO-Ai
                                                                                                 NIL))
                       TIMING:  ((AIR-SPECIALIST-1 H-HOUR) (SUPPORT-OPERATIONS-2 H-HOUR+)
                                                           (SPECIAL-OPERATIONS-2 H-HOUR))
   TRIED-CHARACTERIZATIONS:  NIL

              Links:
                CONFLICT-PART:  (SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS50
                                SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS49
                                SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS48
                                SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS15
                                SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS14
                                SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS13)

              $Links: <none>
Type a space to erase this display:
```

Figure 14. Initial activity of the example.

the entire threat, each agent first determines if it can achieve that solution type. Referring to the response slot in Figure 14, Special-Operations-2 can take responsibility for one threat-cluster and hence offers an alternative suggestion. The other agents offer negative responses. Special-Operations-2 decides not to take on the responsibility for a threat-cluster. Phase II is then entered. Since no other agent can offer to take on threat-cluster responsibility, the agents have reached an impasse under the hard constraint set. No single agent can take responsibility for all threat-clusters, and the threat responsibility cannot be divided sufficiently. Thus, phase III of the framework is entered.

The example continues in Figure 15. During phase III, all agents suggest relaxing the time constraint (refer to the constraint-relaxation slot) and are therefore in mutual agreement. In an effort to find the best solution possible, a conservative strategy is employed that allows only one constraint to be relaxed by one step at any given time. Hence, the time is relaxed to H-hour+ while the means remains neutralize. Since the best solution type is for one agent to take on complete threat responsibility, phase I search is entered where each agent searches to take on the complete responsibility for the threat using the means of neutralize at time H-hour+. Under the requirements of H-hour+ with neutralize, Support-Operations-2 can offer a partial solution. Air-Specialist-1 and Special-Operations-2 respond with a negative reply (refer to the response slot of the object). Support-Operations-2 determines that it will take on the responsibility for sams-yukehr-1. Hence, only one threat cluster remains in the conflict. Since no other agent can take on the remaining target, an impasse is reached. Since all solution types have been eliminated, phase III is entered.

```
CONFLICTS1 is at the SPECIAL-OPERATIONS-SOLUTION-2.CONFLICTS level.

        Attributes:
          AGENTS-INVOLVED:  (AIR-SPECIALIST-1 SUPPORT-OPERATIONS-2 SPECIAL-OPERATIONS-2)
            CONFLICT-TYPE:  THREAT-RESPONSIBILITY
    CONSTRAINT-RELAXATION:  ((AIR-SPECIALIST-1 LATER-TIMING) (SUPPORT-OPERATIONS-2 LATER-TIMING)
                                                             (SPECIAL-OPERATIONS-2 LATER-TIMING))
 CURRENT-CHARACTERIZATION:  LATER-TIMING
            EARLIEST-TIME:  H-HOUR
            HARDEST-MEANS:  NEUTRALIZE
                    MEANS:  ((AIR-SPECIALIST-1 DISABLE) (SUPPORT-OPERATIONS-2 DISABLE)
                                                        (SPECIAL-OPERATIONS-2 NEUTRALIZE))
                   NUMBER:  0
                      REF:  CONFLICTS1
                  RES-AIR:  0
               RES-GROUND:  1
                RES-NAVAL:  0
                 RESPONSE:  ((AIR-SPECIALIST-1 NEGATIVE MEANS) (SUPPORT-OPERATIONS-2 ALTERNATIVE 1 GROUND)
                                                               (SPECIAL-OPERATIONS-2 NEGATIVE UNSPECIFIED))
                   STATUS:  EXAMINE-SEARCH-RESPONSES
          THREAT-CLUSTERS:  ((AIR-GROUP-1 (AIR-TO-GROUND AIR-TO-AIR) ON-GROUND) (SAMS-YUKEHR-1 GROUND-TO-AIR
                                                                                             NIL))
                   TIMING:  ((AIR-SPECIALIST-1 H-HOUR) (SUPPORT-OPERATIONS-2 H-HOUR+)
                                                       (SPECIAL-OPERATIONS-2 H-HOUR))
   TRIED-CHARACTERIZATIONS:  (EARLIEST-TIME-HARDEST-MEANS)

           Links:
           CONFLICT-PART:  (SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS50
                            SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS49
                            SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS48
                            SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS15
                            SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS14
                            SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS13)

          $Links: <none>
Type a space to erase this display:
```

Figure 15. First constraint relaxation.

As shown in Figure 16, the agents disagree on the constraint to relax (refer to the constraint-relaxation slot of the object). Air-Specialist-1 and Support-Operations-2 suggest easier means while Special-Operations-2 responds with "NONE." An agent responds with NONE when it believes backtracking prior to the point of conflict detection is needed. When the agents disagree, they post a justification for their response (refer to the constraint-reason slot of the conflict object). For their responses, Air-Specialist-1 and Support-Operations-2 point to the rules of engagement (MIN-ROE). One of these rules states that minimum damage to Khamani military equipment is required unless it would diminish the success of the operation. Since air-group-1 contains MIGs and the means would be relaxed to disable, this rule is particularly applicable. If aircraft are prevented (disabled) from taking off (e.g., by cratering the runways), the minimum damage rule can be fulfilled without diminishing the operation's success.

To resolve disparity in the conflict resolution phase, we employ a systemwide preference ordering on the justifications for a relaxation suggestion. In the preference ordering, MIN-ROE is preferred to backtracking. If Special-Operations-2 could justify backtracking by claiming that relaxing the means would diminish the success of the Yukehr rescue operation, a more preferred justification, the means would not be relaxed. In this case, however, Special-Operations-2 accepts the justification for relaxing the means. Figure 17 shows the conclusion of the example. During a phase I search under the easier-means characterization, both Air-Specialist-1 and Support-Operations-2 respond positively. Hence, a resolution to the conflict has been reached.

```
CONFLICTS1 is at the SPECIAL-OPERATIONS-SOLUTION-2.CONFLICTS level.

             Attributes:
               AGENTS-INVOLVED:  (AIR-SPECIALIST-1 SUPPORT-OPERATIONS-2 SPECIAL-OPERATIONS-2)
                 CONFLICT-TYPE:  THREAT-RESPONSIBILITY
             CONSTRAINT-REASON:  ((AIR-SPECIALIST-1 EASIER-MEANS MIN-ROE) (SUPPORT-OPERATIONS-2 EASIER-MEANS
                                                                                               MIN-ROE)
                                                                          (SPECIAL-OPERATIONS-2 NONE BACKTRACK))
         CONSTRAINT-RELAXATION:  ((AIR-SPECIALIST-1 EASIER-MEANS) (SUPPORT-OPERATIONS-2 EASIER-MEANS)
                                                                  (SPECIAL-OPERATIONS-2 NONE))
      CURRENT-CHARACTERIZATION:  NIL
                 EARLIEST-TIME:  H-HOUR
                 HARDEST-MEANS:  NEUTRALIZE
                         MEANS:  ((AIR-SPECIALIST-1 DISABLE) (SUPPORT-OPERATIONS-2 DISABLE)
                                                              (SPECIAL-OPERATIONS-2 NEUTRALIZE))
                        NUMBER:  0
                           REF:  CONFLICTS1
                       RES-AIR:  0
                    RES-GROUND:  1
                     RES-NAVAL:  0
                      RESPONSE:  ((AIR-SPECIALIST-1 NEGATIVE MEANS) (SUPPORT-OPERATIONS-2 ALTERNATIVE 1 GROUND)
                                                                     (SPECIAL-OPERATIONS-2 NEGATIVE UNSPECIFIED))
                        STATUS:  EXAMINE-CR-REASONING
               THREAT-CLUSTERS:  ((AIR-GROUP-1 (AIR-TO-GROUND AIR-TO-AIR) ON-GROUND))
                        TIMING:  ((AIR-SPECIALIST-1 H-HOUR) (SUPPORT-OPERATIONS-2 H-HOUR+)
                                                             (SPECIAL-OPERATIONS-2 H-HOUR))
       TRIED-CHARACTERIZATIONS:  (LATER-TIMING EARLIEST-TIME-HARDEST-MEANS)

                   Links:
                 CONFLICT-PART:  (SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS50
                                 SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS49
                                 SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS48
                                 SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS15
                                 SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS14
                                 SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS13)

                 $Links: <none>
Type a space to erase this display:
```

Figure 16. Second constraint relaxation.

```
CONFLICTS1 is at the SPECIAL-OPERATIONS-SOLUTION-2.CONFLICTS level.

                 Attributes:
            AGENTS-INVOLVED:  (AIR-SPECIALIST-1 SUPPORT-OPERATIONS-2 SPECIAL-OPERATIONS-2)
              CONFLICT-TYPE:  THREAT-RESPONSIBILITY
          CONSTRAINT-REASON:  ((AIR-SPECIALIST-1 EASIER-MEANS MIN-ROE) (SUPPORT-OPERATIONS-2 EASIER-MEANS
                                                                                   MIN-ROE)
                                                               (SPECIAL-OPERATIONS-2 NONE BACKTRACK))

      CONSTRAINT-RELAXATION:  ((AIR-SPECIALIST-1 EASIER-MEANS) (SUPPORT-OPERATIONS-2 EASIER-MEANS)
                                                               (SPECIAL-OPERATIONS-2 NONE))

 CURRENT-CHARACTERIZATION:  EASIER-MEANS
             EARLIEST-TIME:  H-HOUR
             HARDEST-MEANS:  NEUTRALIZE
                     MEANS:  ((AIR-SPECIALIST-1 DISABLE) (SUPPORT-OPERATIONS-2 DISABLE)
                                                         (SPECIAL-OPERATIONS-2 NEUTRALIZE))

                    NUMBER:  0
                       REF:  CONFLICTS1
                   RES-AIR:  0
                RES-GROUND:  1
                 RES-NAVAL:  0
                  RESPONSE:  ((AIR-SPECIALIST-1 POSITIVE) (SUPPORT-OPERATIONS-2 POSITIVE)
                                                          (SPECIAL-OPERATIONS-2 NEGATIVE UNSPECIFIED))

                    STATUS:  RESOLVED
           THREAT-CLUSTERS:  ((AIR-GROUP-1 (AIR-TO-GROUND AIR-TO-AIR) ON-GROUND))
                    TIMING:  ((AIR-SPECIALIST-1 H-HOUR) (SUPPORT-OPERATIONS-2 H-HOUR+)
                                                        (SPECIAL-OPERATIONS-2 H-HOUR))

  TRIED-CHARACTERIZATIONS:  (LATER-TIMING EARLIEST-TIME-HARDEST-MEANS)

                    Links:
             CONFLICT-PART:  (SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS50
                              SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS49
                              SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS48
                              SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS15
                              SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS14
                              SPECIAL-OPERATIONS-SOLUTION-2.THREATS.THREATS13)

                  $Links: <none>
Type a space to erase this display:
```

Figure 17. Conclusion of the example

6.2. *Evaluating the implementation*

In general, the demonstrations to the expert on the project have been positive. The negotiation process and conflict resolutions were considered reasonable and appropriate. One noted problem relates to the use of a single systemwide preference ordering for constraint relaxation. Rarely, if ever, would an expert commit to an "always" qualification. Thus, always applying a single preference ordering is insufficient. It is unclear whether a more elaborate justification process (as in Sycara's PERSUADER system (1990) where an agent may need to be convinced to accept a compromise) is needed or whether a dynamic application of different preference orderings can be used. Of course, one may say that the latter begs the question as an agent may need to be convinced to accept the use of a preference ordering unless a "meta" preference ordering is used. In practice, however, if classes of applicable circumstances can be delineated, the mapping of preference orderings to circumstance classes may be obvious.

Further comments are warranted. One expert created the model plan for the scenario. During the consultations, several undetected conflicts in his original plan were found. While this observation may not directly relate to automated agents, it does seem to indicate the value of different perspectives during complex planning. Moreover, we experienced close to a parallel speed-up, using clock time, when a third machine was added to a two-machine configuration. It is unclear whether the speed-up relates to actual problem-solving activities, the overhead of inference process, or the varying processing speeds of the machines (although in all cases, the third machine significantly increased performance).

7. Concluding remarks

Our research focused on using decentralized negotiation for interdependent conflict resolution in distributed planning systems. The resulting model arose out of two different types of studies. In Phoenix, the computational feasibility of negotiation was explored. In the CMP domain, an expert was consulted. During this effort, the validity of using hard and soft constraints to structure the negotiation process was confirmed. While the three problem-solving phases of search, analysis, and constraint relaxation could be identified in the human model, it is doubtful that the specific process is similar. However, even though the reasoning of an expert system does not necessarily parallel the reasoning of an expert, knowledge-based problem solving is a valid and useful method.

While we do not make any claims about the specific negotiation strategies employed in our implementations, it is apparent that the DENEGOT framework is a valid approach to interdependent conflict resolution in many complex domains. The major characteristics of these domains that will allow our framework to be employed is that their composite search spaces can be structured into a lattice of sets of potential compromise solutions based on hard constraints such that a solution in a higher set in the lattice, if it is achievable, will most of the time be preferable over a solution in a lower set. Solutions within a set may be further ordered in terms of their preference based on soft constraints. Another requirement is that this structuring of the composite search space be able to be mirrored in the structuring of agents' local search spaces.

Multi-Fireboss Phoenix and CMP presented very different types of distributed planning problems. In Multi-Fireboss Phoenix, agents are homogeneous, solutions are evaluated numerically, and agents react to changes in the environment. In CMP, agents are heterogeneous, solutions are evaluated symbolically, and agents concentrate on pre-planning. In both domains, however, the distributed search space can be structured by hard and soft constraints; thus the DENEGOT framework is applicable.

One future area of study involves comparing various negotiation strategies. For example, agents can be more competitive about which partial solutions they are willing to offer, various strategies can be used to direct the local search activities, and a more comprehensive evaluation can be used to determine the constraints to relax and the degree of relaxation. In addition, the model can be expanded to include a more complete evaluation of potential solutions. Rather than using the first solution that is discovered, a more extensive analysis could be performed to find an optimal or near-optimal solution. For example, an iterative deepening approach could be employed to first search for an acceptable solution and then to optimize that solution as time permits.

We have been assuming that individual agents are only involved in one negotiation session at a time. The negotiation process becomes much more complex if agents are participating in several sessions during any single time period. For example, agents must guarantee that partial solutions offered in one session are consistent with those offered in other sessions. The phases of the framework may have to be expanded to incorporate this multi-session negotiation paradigm. Moreover, in our model, a negotiation session must end before another agent can become involved. A future research area involves determining how to include an agent in a negotiation session if it becomes clear that the agent is involved in the conflict after a session has commenced.

Another future research area involves exploring how the DENEGOT model can be used to achieve increased system reliability, fuller exploitation of parallelism, and reduced communication requirements. For example, how can a team of agents dynamically reorganize problem-solving responsibility if an agent is lost or if the workload is distributed unevenly? In addition, during negotiation, how can agents optimize the amount of time spent locally searching for partial solutions and the amount of time spent in discussions with other agents? In other words, how can agents limit the amount of information communicated without sacrificing solution quality?

DENEGOT provides a flexible model for decentralized interdependent conflict resolution. The two domains of study provided complex planning problems which raised many issues involved in real-world planning. The resulting model is flexible in that it does not dictate specific problem-solving techniques or strategies and general in that it can be applied to various agent organizations. This flexibility and generality make the model applicable to a wide range of complex planning problems.

Acknowledgments

Many thanks to Paul Cohen and the members of the EKSL lab for the use and support of Phoenix. A very special thanks to Chuck Shankland for consistently remaining patient and good natured when confronted with a persistent AI researcher who was completely naive about all things concerning the military. Thanks also to Norman Carver, Keith Decker, Les Gasser, Dan Neiman, and the anonymous reviewers for their insightful comments on earlier drafts of this article.

Notes

1. Though we have talked about a strict ordering of solution sets based on hard constraints where a solution in a higher set is always preferable to a solution in a lower set, the need in some domains to keep the number of sets to a manageable level may preclude this strict ordering. Instead a looser criterion can be used where it is likely rather than necessary for a solution in a higher set to be preferable over a solution in a lower set.
2. Multi-Fireboss Phoenix is based on a 1989 version of Phoenix.
3. The need to distinguish problem-solving phases within a negotiation session in a decentralized environment has been recognized by Conry and colleagues (1991) and recently by Lander (August 1991).

4. If human models are to be taken as a guide, there are indications that in truly competitive environments, decentralized negotiation has a high cost and may be unmanageable. When agents perform cost-benefit analyses that only consider factors directly related to their local state and do not consider more global implications, mutual agreement may be difficult to achieve as no agent may be willing to take on external costs that are inherent to resource use. In addition, resources are likely to be used inefficiently (e.g., even if aggregate benefits significantly increase by modified resource use, an agent may be unwilling to make that change if it itself suffers some loss). Interested readers are referred to Demsetz's (1967) economic theory of property rights.
5. The simulator steps time each second so that, for instance, the number 94072 represents the simulated elapsed time in seconds (approximately 26 hours and 8 minutes) since the start of the simulation at 12:00 on 8/1.
6. The reasoning needed to process the knowledge of the domain map seems similar in character to the type of reasoning employed in the case-based reasoning approach of Sycara (1989) and in the work of Klein (1991).
7. In fact, optimality may not even be clearly defined in this domain.

References

Adler, Mark R., Davis, Alvah B., Weihmayer, Robert and Worrest, Ralph W.: 1989, "Conflict-Resolution Strategies for Nonhierarchical Distributed Agents," pp. 139–161 in *Distributed Artificial Intelligence*, Vol. 2 (eds. Les Gasser and Michael N. Huhns), Pitman Publishers, London, and Morgan Kaufmann Publishers, San Mateo, CA.

Buteau, Brandon: 1990, "A Generic Framework for Distributed, Cooperating Blackboard Systems," *Proceedings of the Eighteenth Annual ACM Computer Science Conference*, Washington, DC, February 20–22, 358–365.

Cohen, Paul and Feigenbaum, Edward, eds.: 1982, *Handbook of Artificial Intelligence*, Vol. 3, Addison-Wesley, Reading, MA.

Cohen, Paul R., Greenberg, Michael L., Hart, David M. and Howe, Adele E.: 1989, "Trial by Fire: Understanding the Design Requirements for Agents in Complex Environments," *AI Magazine* **10**(3), 34–48.

Conry, Susan E., Kuwabara, Kazuhiro, Lesser, Victor R. and Meyer, Robert A.: 1991, "Multistage Negotiation for Distributed Constraint Satisfaction," *IEEE Transactions on Systems, Man, and Cybernetics*, Special issue on Distributed Artificial Intelligence, **21**(6): 1462–1477, New York.

Conry, Susan E., Meyer, Robert A. and Lesser, Victor R.: 1988, "Multistage Negotiation in Distributed Planning," pp. 367–384, in *Readings in Distributed Artificial Intelligence* (eds. Alan H. Bond and Les Gasser), Morgan Kaufmann.

Davis, Randall and Smith, Reid G.: 1983, "Negotiation as a Metaphor for Distributed Problem Solving," *Artificial Intelligence* **20,** 63–109.

Demsetz, Harold: 1967, "Toward a Theory of Property Rights," 57 *American Economic Review*, **347** (Paper and Proceedings).

Durfee, Edmund H. and Lesser, Victor R.: 1986, "Incremental Planning to Control a Blackboard-Based Problem Solver," *Proceedings of the National Conference on Artificial Intelligence*, Philadelphia, PA, 58–64.

Durfee, Edmund H. and Lesser, Victor R.: 1991, "Partial Global Planning: A Coordination Framework for Distributed Hypothesis Formation," *IEEE Transactions on Systems, Man, and Cybernetics*, **21**(5): 1167–1183, Sept/Oct. 1991.

Durfee, Edmund H., Lesser, Victor R. and Corkill, Daniel D.: 1989, "Cooperative Distributed Problem Solving," in pp. 83–147, *Handbook of Artificial Intelligence*, Vol. 4 (eds. Avron B. Barr, Paul R. Cohen, and Edward A. Feigenbaum), Addison-Wesley, Reading, MA.

Laasri, Brigitte, Laasri, Hassan and Lesser, Victor R.: 1991, "An Analysis of Negotiation and Its

Role for Coordinating Cooperative Distributed Problem Solvers," *Proceedings of the General Conference on Second Generation Expert Systems,* Vol. 2, Eleventh International Conference on Expert Systems and Their Applications, Avignon, France, 81–94.

Klein, Mark: 1991, "Supporting Conflict Resolution in Cooperative Design Systems," *IEEE Transactions on Systems, Man and Cybernetics,* Special Issue on Distributed Artificial Intelligence, **21**(6): 1379–1390, Nov/Dec 1991.

Lander, Susan E. and Lesser, Victor R.: 1991, "Negotiated Search: A Framework for Cooperative Design," Department of Computer Science Technical Report #91-79, University of Massachusetts, Amherst.

Lander, Susan E., Lesser, Victor R. and Connell, M.E.: 1991a, "Conflict.Resolution Strategies for Cooperating Expert Agents," in pp. 183–198, *Cooperating Knowledge-Based Systems 1990,* (ed. S.M. Deen), Springer-Verlag, New York.

Lander, Susan E., Lesser, Victor R. and Connell, M.E.: 19(1b, "Knowledge-Based Conflict Resolution for Cooperation Among Expert Agents," in *Computer-Aided Cooperative Product Development* (eds., D. Sriram, R. Logcher, and S. Fukuda), Springer-Verlag, New York.

Lansky, Amy L.: 1990, "Localized Search for Controlling Automated Reasoning," *Workshop on Innovative Approaches to Planning, Scheduling, and Control,* San Diego, CA, 115–125.

Lesser, Victor R.: 1991, forthcoming, "A Retrospective View of FA/C Distributed Problem Solving," *IEEE Transactions on Systems, Man and Cybernetics,* Special Issue on Distributed Artificial Intelligence, **21**(6): 1347–1362, Nov/Dec 1991.

Linden, Theodore Al: 1990, "Transformational Synthesis: An Approach to Large-Scale Planning Applications," *Workshop on Innovative Approaches to Planning, Scheduling, and Control,* San Diego, CA, 126–130.

Moehlman, Theresa and Lesser, Victor: 1990, "Cooperative Planning and Decentralized Negotiation in Multi-Fireboss Phoenix," *Workshop on Innovative Approaches to Planning, Scheduling, and Control,* San Diego, CA, 144–159.

Sacerdoti, Earl D.: 1974, "Planning in a Hierarchy of Abstraction Spaces," *Artificial Intelligence* **5**, 115–135.

Sathi, Arvind and Fox, Mark S.: 1989, "Constraint-Directed Negotiation of Resource Reallocations," in pp. 163–193, *Distributed Artificial Intelligence,* Vol. 2, (eds. Les Gasser and Michael N. Huhns), Pitman Publishers, London and Morgan Kaufmann Publishers, San Mateo, CA.

Steeb, Randall, Cammarata, Stephanie, Haynes-Roth, Frederick A., Thorndyke, Perry W. and Wesson, Robert B.: 1988, "Distributed Intelligence for Air Fleet Control," in pp. 90–101, *Readings in Distributed Artificial Intelligence,* (eds., Alan H. Bond and Les Gasser), Morgan Kaufmann.

Sycara, Katia: 1989, "Multiagent compromise via negotiation," in pp. 119–137, *Distributed Artificial Intelligence,* Vol. 2 (eds. Les Gasser and Michael N. Huhns), Pitman Publishers, London and Morgan Kaufmann Publishers, San Mateo, CA.

Sycara, Katia, Roth, Steve, Sadeh, Norman and Fox, Mark S.: 1990, "Managing Resource Allocation in Multi-Agent Time-Constrained Domains," *Workshop on Innovative Aproaches to Planning, Scheduling, and Control,* San Diego, CA, 240–250.

Sycara, Katia, Roth, Steve, Sadeh, Norman and Fox, Mark S.: 1991, "Distributed Constrained Heuristic Search," *IEEE Transactions on Systems, Man and Cybernetics,* Special Issue on Distributed Artificial Intelligence, **21**(6): 1446–1461, Nov/Dec 1991.

von Martial, Frank: forthcoming, "Coordinating Plans of Autonomous Agents," in *Lecture Notes in Artificial Intelligence,* Springer-Verlag.

Werkman, Keith J.: 1990, "Multiagent Cooperative Problem Solving through Negotiation and Perspective Sharing," Unpublished doctoral dissertation, Lehigh University.