U.Kuter, D.Nau, D.Gossink, and J.F. Lemmer. Interactive course-of-action planning using causal models. *Proceedings of the Third International Conference on Knowledge Systems for Coalition Operations (KSCO-2004)*, October 2004 (postponed).

Interactive Course-of-Action Planning Using Causal Models

Ugur Kuter[1], Dana Nau[1], Don Gossink[2], and John F. Lemmer[3]

[1]University of Maryland, Dept. of Computer Science,
College Park, MD 20742, USA
email: {ukuter,nau}@cs.umd.edu

[2]Command and Control Division,
Defense Science and Technology Org.,
Edinburgh, 5111, Australia
email: Don.Gossink@dsto.defence.gov.au

[3]Rome Laboratory,
525 Brooks Road, Rome, NY 13441, USA
email: John.Lemmer@rl.af.mil

Abstract. This paper describes a new technique for interactive planning for coalition operations under conditions of uncertainty. Our approach is based on the use of the Air Force Research Laboratory's Causal Analysis Tool (CAT), a system for creating and analyzing causal models similar to Bayesian networks.

In order to use CAT as a tool for planning for coalition operations, users go through an iterative process in which they use CAT to create and analyze alternative plans. One of the biggest difficulties is that the number of possible plans that must be analyzed is exponential in the number of possible actions that may or may not appear in those plans. In any planning problem of significant size, it is impossible for the user to create and analyze every possible plan; thus users can spend days arguing about which actions to include in their plans.

To solve this problem, we have developed an approach to quickly compute upper and lower bounds on the probabilities of success associated with a partial plan, and use these probabilities to recommend which actions the user should include in the plan in order to get a complete plan. This provides an exponential reduction in the amount of time needed to find a complete plan. In our experiments, our approach generated recommendations that resulted in plans that have the highest probability of success in just a few minutes.

1 Problem and Significance

In planning a coalition's course of action (i.e., a plan for the coalition to execute to achieve a desired objective or objectives), quick and accurate decision making is a very important task and it is very hard. A major source of difficulty is how to deal with uncertainty. This uncertainty has many sources, but perhaps the biggest one is the uncertain relationship between causes and effects. For example:

- *At a tactical level, sorties are flown against a series of bridges to prevent the enemy ground forces from crossing the river. The sorties are intended to prevent the crossing. What is the probability that they will?*
- *At a strategic level, the international coalition's destruction of the Taliban Army was intended ultimately to reduce world-wide terrorism. Did it?*

Such uncertainties are compounded by the size and complexity of most coalition plans—for example, a causal model of Operation Deny Freedom, built by the actual planners, contains over 300 uncertain events interrelated by cause and effect. Moreover, there are often significant delays between cause and effect, and effects may persist for only limited amounts of time: a destroyed bridge can be rebuilt or bypassed. This makes it exceedingly difficult to forecast the possible effects of a coalition operation.

This paper describes the approach we have developed to help analyze this uncertainty in order to generate effective plans. The basis for our approach is the Air Force Research Laboratory's (AFRL's) Causal Analysis Tool (CAT), which is a tool for representing and analyzing causal networks similar to Bayesian networks. From this representation, CAT can compute the probability that any given plan (i.e., any chosen combination of actionable items) will achieve the desired objectives of that coalition.

A major technical difficulty is how to overcome combinatorial blowup during the planning process. If there are n different actionable items, then there are potentially 2^n different plans, making it infeasible for the user to ask CAT to analyze each one. Our approach exploits the conditional-independence relationships within a causal network in order to overcome this combinatorial blowup. In doing so, it quickly computes upper and lower bounds on the probabilities of success associated with a partial plan, and uses these bounds to recommend which actions the user should include in the plan in order to get a complete course of action. This provides an exponential reduction in amount of time needed to find a complete plan. In our experimental evaluation, our approach generated recommendations that resulted in plans that have the highest probability of success in just a few minutes, demonstrating its effectiveness.

2 Background: Causal Analysis Tool (CAT)

Causal Analysis Tool (CAT) is a system developed by the Air Force Research Laboratory (AFRL) for being use in creating, modifying and analyzing causal models. CAT is a development tool that is currently in prototype stage and it has not been deployed in any sort of active use yet. However, to the best of our knowledge, several strategic-level organizations within the US Air Force are testing CAT and giving positive feedback about it.

2.1 Probability Analysis in CAT

The basic function of CAT is to propagate local estimates of uncertainty throughout large models. Its most basic output is the probability, as a function of time, that particular events will be true. Below, we give a brief summary of CAT; for detailed information on the technology that CAT uses, see [9, 10, 8].

Probability analysis in CAT is based on the use of causal models; CAT provides tools to either construct a causal model or load a previously constructed causal model from a file. CAT's causal models are similar to Bayesian Networks (and CAT compiles them into Bayesian Networks in order to do its analysis). However, CAT's causal models incorporate several extensions in order to make Bayesian causal modeling available to users who do not have specialized probability training, and allow sophisticated incremental improvement of these models when more time is available.

In CAT, a *causal model* is a directed graph (e.g., see Figure 1 on the next page) in which each node represents an event that may or may not occur. There are three different kinds of events: *actionable items*, which are actions that the coalition may choose whether or not to perform, *goals* that the coalition may wish to achieve, and other intermediate events that are neither actionable items nor goals. The edges (which are called *mechanisms*) represent causal and inhibitory relationships between events. A mechanism $e_1 \mapsto e_2$ between events e_1 and e_2 is *causal* if the occurrence of e_1 increases e_2's probability of occurrence, and it is *inhibitory* if the occurrence of e_1 reduces e_2's probability of occurrence. Associated with each mechanism is a number between 0 and 1 to indicate the probability with which e_1 causes or inhibits e_2. These numbers are probabilities of causation or inhibition rather than the conditional probabilities used in Bayesian networks—but they can be translated into the latter, and CAT does such a translation in order to perform its calculations.

In a causal model, the user can specify a number of probabilities by filling in the probability tables for each event in the causal model. For example, Figure 1 shows a set of user-specifed causal probabilities for the event "Destroy IADS" in that model. These probabilities tell us that each of the mechanisms "No Communications", "No Sensors", "No Weapons", and "No C2" will cause this event alone with probability 0.76. The user can also specify causal probabilities for the event "Destroy IADS" given various groups of its causes by using the "group" check-boxes.

Furthermore, each event in a causal model is associated with a special type of probability, called the *leak probability* for that event. Intuitively, an event's leak probability specifies the probability that the event will occur even when none of its causes occurs in the world. In other words, a leak probability specifies the causes of an event that are not specified explicitly in the given causal model. Leak probabilities allow CAT users to work with incomplete causal models with unknown events and still be able to reason and compute the probabilities of the events already in the causal model.

To calculate the probabilities of occurrence for the events and mechanisms of a causal model M, CAT first compiles M into a Bayesian Network, say $B(M)$, such that the event and mechanisms in M correspond to the nodes of $B(M)$. CAT computes two different conditional probability tables (CPTs) for each node n in $B(M)$; namely, a *Causal CPT* and a *Inhibitory CPT*. These conditional probability tables model the causal and inhibitory relationships among the events and mechanisms of a causal model, as described above. They are both computed using the *Recursive Noisy-OR (RNOR) rule* reported in [8]. The RNOR rule is a generalization of the traditional *Noisy-OR* rule [12], which is widely used for computing the probability of an event, given the conditional probabilities that describe the dependencies between that event and each of its predecessors. The RNOR rule allows for modeling and reasoning about complex

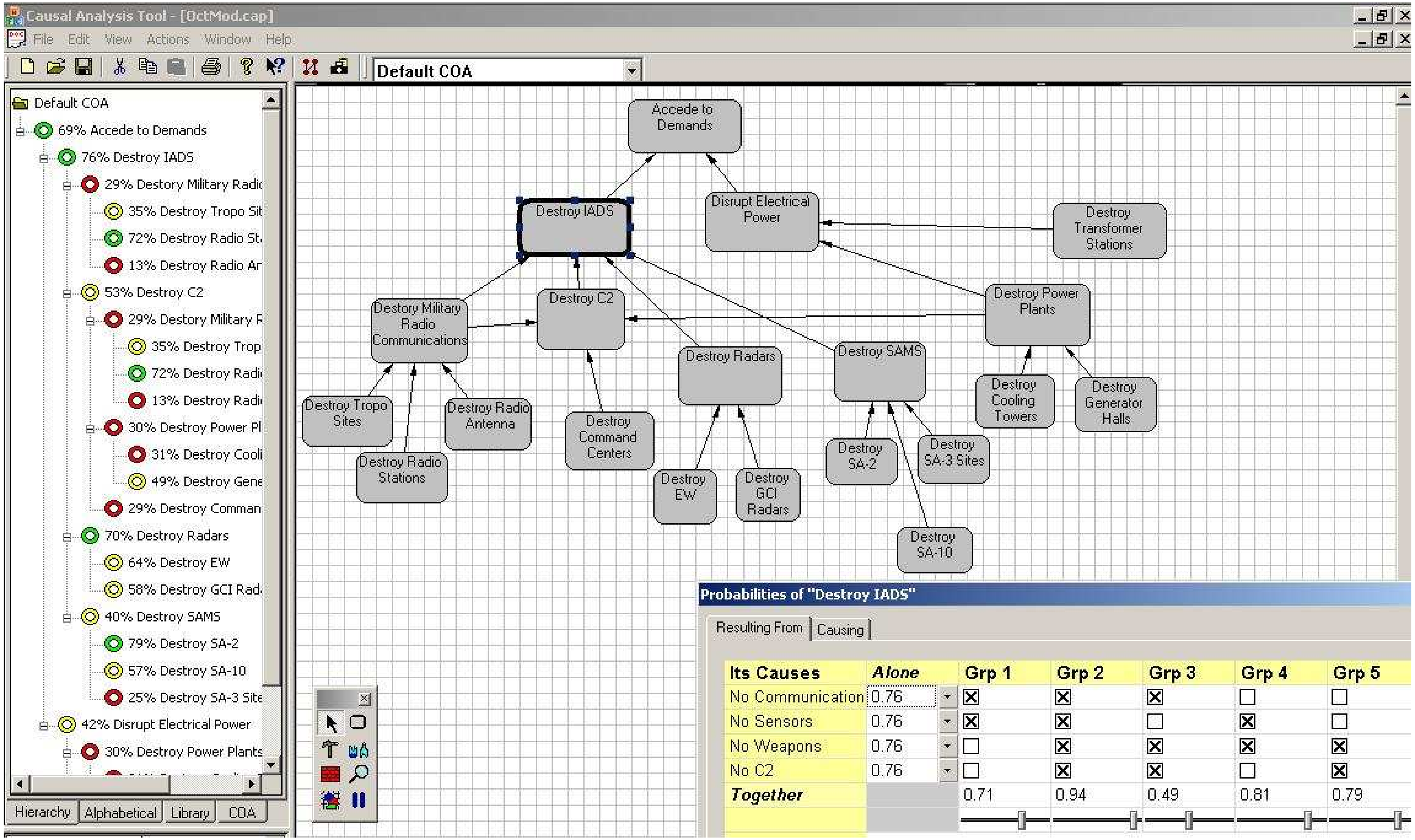

Figure 1: An abstract causal model named “Operation OctMod” which represents the plan that the international coalition used against Milosevic in the Bosnia-Herzegovina war. The window on the right-hand side of the screen shows a portion of the probability table stored in the highlighted node. The actionable items are the twelve nodes at the bottom of the network that have no predecessors.

dependencies between events of a given causal model, which cannot be captured by the Noisy-OR rule. For details on the RNOR rule, see [8].

Having computed the special conditional probability tables described above, CAT performs a variant of *probabilistic logic sampling* [5] over the compiled Bayesian Network $B(M)$, in which it repeatedly simulates the occurrence (or nonoccurrence) of the nodes in $B(M)$.[1] In each simulation run, CAT decides whether an event (i.e., a node in $B(M)$) n occurs with the probability computed by the following formula

$$(1.0 - InhibitoryCPT(n, inhibitors(n))) \times [1.0 - (1.0 - CausalCPT(n, causes(n)))(1.0 - Leak(n))],$$

where $Leak(n)$ is the leak probability associated with the node n, and $CausalCPT$ and $InhibitoryCPT$ denote the causal and inhibitory conditional probability tables computed by CAT for the node n during the compilation phase. $causes(n)$ and $inhibitors(n)$ are the sets of predecessors of n in $B(M)$ such that $causes(n)$ denotes the set of nodes in $B(M)$ whose occurrence increases the probability of occurrence for n, and $inhibitors(n)$ denote the set of nodes whose occurrence decreases the probability of occurrence for n.

The formula given above specifies the following probabilistic-reasoning behavior: if a node n had no predecessors in $B(M)$ that inhibit the occurrence of n, then we would want n to occur as a result of its causal dependencies specified in $CausalCPT(n, causes(n))$, and/or as a result of unmodeled external factors with probability $Leak(n)$. However, if n has inhibiting predecessors, then the probability of the occurrence of n due to its causes and its leak probability value may be reduced with the probability specified in $InhibitoryCPT(n, inhibitors(n))$. Thus, in each simulation, n will occur with the probability computed with the formula above, given the occurrences and non-occurrences of each of its predecessors in that simulation run.

In each simulation run, CAT starts with the nodes in $B(M)$ that has no predecessors. For such nodes, the above formula specifies the "a priori" probabilities given as input by the user. The simulation progresses by iteratively considering each node n in $B(M)$ such that the occurrence or non-occurrence of all of the predecessors of n is already probabilistically simulated in this particular run. This way, when CAT considers to simulate the occurrence or nonoccurrence of a node n in a run, it always knows whether the predecessors of n occurred or not in that particular simulation run. In other words, CAT always knows whether the nodes in $causes(n)$ and $inhibitors(n)$ are occured or not in that particular simulation run, when it considers the node n.

CAT runs its simulation repeatedly, for as long as the user wants. As it does so, it keeps statistics on how frequently each node occurs. It uses these statistics to compute an estimate of the probability of occurrence for every event in the original causal network M. CAT displays these estimates to the user as shown in the left-hand pane of Fig. 1. As CAT runs more and more simulations, the estimates of each such probability get progressively more accurate, and CAT updates its display accordingly. The user may stop running simulations whenever he/she feels that the estimates have become sufficiently accurate.

2.2 Planning using CAT

In CAT, planning takes place as an iterative and interactive process in which users repeatedly do the following: (1) they make decisions about some actionable items to include and/or exclude, (2) they use CAT to obtain an estimate of the probability of achieving the goal,[2] and (3) they revise these decisions based on their experience and intuition.

Users may need to try many combinations of actionable items in order to generate the plan that has the highest probability of achieving the goal. This plan is not necessarily the one that includes all possible actionable items: if the causal model contains inhibitory mechanisms, then some actionable items may reduce the probability of achieving the goal. In order to find the plan that maximizes the probability of achieving the goal, in the worst case a user may need to create and analyze exponentially many alternative plans. For example, if there are n actionable items, then there are 2^n different possible combinations of the actionable items, i.e., 2^n different plans. Since the causal models for coalition operations can be quite large and complex, and since the planning often needs to be done in a very limited amount of time under stressful conditions, it clearly is not feasible for the user to generate and examine all of these plans.

As an example, if $n = 22$ then there are 2^{22} different possible plans. Suppose CAT takes 10 seconds to analyze each plan (this assumption is rather optimistic: if the network is sufficiently large, CAT might take minutes or even

[1] The reason why CAT uses probabilistic logic sampling is because of the way in which CAT reasons about time and scheduling; the details are beyond the scope of this paper.

[2] For simplicity, in this paper we assume that there is just one goal g. Situations in which there are several goals $g_1, \ldots, g_k$ can sometimes be modeled by adding a new node g whose causes are $g_1, \ldots, g_k$.

hours). Then the total time needed to analyze all of the plans is approximately 11,651 hours, or more than 485 days. Clearly, this is not acceptable.

3 Our Approach

We have developed a way to overcome the exponential blowup described above. Our approach involves modifying CAT so that it can represent and reason about *partial plans* in which the user has made yes-or-no decisions for some of the actionable items and the others remain *undecided*. This enables the users to carry out the following *iterative plan-development process*: the user begins with a partial plan in which all actionable items are undecided, and gradually makes decisions about more and more of the items until no undecided items remain.

By using our technique, we can give the following feedback to the user at each iteration of the planning process: (1) upper and lower bounds on the probabilities of success that can be attained with the current partial plan, and (2) a recommendation for what choices to make next in order to achieve a complete plan. The following subsections describe how we compute the upper and lower bounds, and how we use these bounds to recommend which actionable items to include or exclude next.

3.1 Upper and Lower Bounds

We now discuss how to compute lower and upper bounds $P_{\min}(e)$ and $P_{\max}(e)$ on the probability of each event in a causal model M.

It is simple to put lower and upper bounds on the probabilities of the actionable items. Suppose the set of actionable items is $A = \{a_1, \ldots, a_n\}$, and suppose the user has already chosen some set of actions $D^+ \subseteq A$ to include in the plan and some subset $D^- \subseteq A$ to exclude from the plan, so that the current partial plan is $D = D^+ \cup \{\neg a_i : a_i \in D^-\}$. Then for each $a_i \in D^+$, $P_{\min}(a_i) = P_{\max}(a_i) = 1$; and for each $a_i \in D^-$, $P_{\min}(a_i) = P_{\max}(a_i) = 0$. For each $a_i \in A \setminus (D^+ \cup D^-)$, the user has not yet decided whether to include a_i in the plan, so the tightest lower and upper bounds we can place on $P(a_i)$ are $P_{\min}(a_i) = 0$ and $P_{\max}(a_i) = 1$.

Given the probabilities $\{P_{\min}(a_i), P_{\max}(a_i)\}_{i=1}^{n}$, we want to compute $P_{\min}(e)$ and $P_{\max}(e)$ for every event in M that is not an actionable item. One way would be the brute-force approach: run CAT's probability analysis on M repeatedly, once for every combination of probabilities $\{P(a_i) \in \{0,1\} : a_i \in A \setminus D\}$. However, this approach incurs the same kind of exponential blowup that we discussed earlier, because it requires doing the probability analysis 2^{n-m} times, where $n = |A|$ and $m = |D|$. As we now describe, a quicker computation can be done by taking advantage of conditional independence among the events in M.

During CAT's simulations, the occurrence or non-occurrence of an event e in the Bayesian network $B(M)$ is represented by a boolean random variable $x(e) \in \{0,1\}$. During each simulation run, the probability that CAT assigns $x(e) = 1$ is $P(e)$. In our modified version of CAT, the simulation procedure instead uses *two* random variables $x_{\min}(e)$ and $x_{\max}(e)$ for each event e. Our simulation assigns $x_{\min}(e) = 1$ with a probability that is a lower bound on $P(e)$, and it assigns $x_{\max}(e) = 1$ with a probability that is an upper bound on $P(e)$. This is done as follows.

If e is an actionable item, then there are three cases:

- If the user has chosen to include in the plan, we assign $x_{\min}(e) = x_{\max}(e) = 1$.
- If the user has chosen not to include in the plan, we assign $x_{\min}(e) = x_{\max}(e) = 0$.
- Otherwise we assign $x_{\min}(e) = 0$ and $x_{\max}(e) = 1$.

If e is not an actionable item, then let $e_1, e_2, \ldots, e_b$ be all of the nodes that may affect e, i.e., $e_1, e_2, \ldots, e_b$ are the predecessors of e. Suppose the simulation has progressed far enough to assign values to $x_{\min}(e_i)$ and $x_{\max}(e_i)$ for $i = 1, \ldots, b$. From conditional independence, it follows that $P(e)$ depends only on $e_1, \ldots, e_b$. Thus, the set of possible probabilities for e is

$$\begin{aligned} S = \{P(e|x(e_1), x(e_2), \ldots, x(e_b)) :& \\ & x(e_1) \in \{x_{\min}(e_1), x_{\max}(e_1)\}, \\ & x(e_2) \in \{x_{\min}(e_2), x_{\max}(e_2)\}, \\ & \ldots, \\ & x(e_b) \in \{x_{\min}(e_b), x_{\max}(e_b)\}\}. \end{aligned}$$

Then the simulation assigns $x_{\min}(e) = 1$ with probability $\min(S)$, and assigns $x_{\max}(e) = 1$ with probability $\max(S)$.

3.2 Providing Feedback and Recommendations

Like the original version of CAT, the modified version can keep running simulations for as long as the user wishes. Suppose that the user has made some set of yes-or-no decisions D. For each node e, let $P^k_{\min}(e|D)$ and $P^k_{\max}(e|D)$ be the average values of $x_{\min}(e)$ and $x_{\max}(e)$ over a set of k simulation runs. Our modified version of CAT displays these averages to the user as shown in the left-hand panes of Figures 2, 3, and 4. As the number of runs increases, $P^k_{\min}(e|D)$ and $P^k_{\max}(e|D)$ converge to lower and upper bounds on $P(e|D)$.

Our modified version of CAT uses a hill-climbing approach to provide recommendations for additional actions to include in D^+ and D^-. Suppose g is some *goal event* whose probability the user wants to maximize. In addition to computing $P^k_{\min}(g|D)$ and $P^k_{\max}(g|D)$ as described above, our modified version of CAT also computes $P^k_{\min}(g|D, a_i)$ and $P^k_{\min}(g|D, \neg a_i)$ for every $a_i \in A \setminus (D^- \cup D^+)$. Let

$$P^* = \max \bigcup_i \{P^k_{\min}(g|D, a_i), P^k_{\min}(g|D, \neg a_i)\}.$$

Then P^* is the largest amount by which $P_{\min}(g|D)$ can increase if the user makes a yes-or-no decision about one of the undecided actions. Either there is an a_i such that $P^k_{\min}(g|D, a_i) = P^*$, in which case our modified version of CAT will recommend including a_i in the plan, or else there is an a_i such that $P^k_{\min}(g|D, \neg a_i) = P^*$, in which case our modified version of CAT will recommend excluding a_i from the plan.

3.3 Computation Time

The total computation time required by this technique is no greater than the time needed for $n2^b$ calls to the original version of CAT, where b is the maximum number of predecessors of each node and n is the number of actionable items. This is a substantial improvement over 2^n, because b normally remains small even in very large networks. For example, in the OctMod example of Fig. 1, no node has more than four predecessors. Furthermore, if most nodes have fewer than b predecessors (as is true in the OctMod example), then the total computation time will be substantially less than $n2^b$.

For example, let us suppose that we have causal model in which the maximum number of predecessors of each node is $b = 4$, the number of actionable items is $n = 25$ and three of the actionable items has been already decided — i.e., we have $m = 3$. Furthermore, suppose again that CAT needs 10 seconds each time it analyzes the causal network. Then the total time needed for us to get the complete plan is less than 70 minutes. This is substantially better than the 485 days required by the brute-force approach!

4 Implementation and Preliminary Experiments

We have implemented our approach in CAT that computes the probabilities and recommendations described in the previous section, and done some preliminary experiments. For our experiments, we have used unclassified versions of causal models for two coalition-operation scenarios. One is the "Operation SSWOTS," a portion of which is shown in Figure 2, is a "scrubbed" version of a much larger model developed for the war in Afghanistan. The other, called the "Operation OctMod," shown in Figure 1. The "Operation OctMod" model is a representation of the causal model that was used against Milosevic in the Bosnia-Herzegovina war. For each case, it was possible to use our modified version of CAT to develop plans in the order of minutes.

We now describe a sample user session we have performed with the OctMod example. In this example, the maximum and the minimum probabilities of occurrence for the goal event (the "accede to demands" node in Figure 3) are 90% and 0%, respectively. The maximum probability of success is achieved when all of the actions are included in the plan, and the minimum probability of success is achieved when all are excluded.

Initially, we did not specify any decisions on which of the actionable items to include in the plan or exclude from it, so all of the actionable items are marked as *undecided*. We first asked our modified version of CAT to analyze the causal model and make a recommendation. CAT then calculated the maximum and minimum probabilities shown in the left-hand pane in Fig.3. Note that these probabilities computed by CAT are correct estimates of the actual minimum and maximum probabilities of the goal node in this example since our approach enables CAT to compute

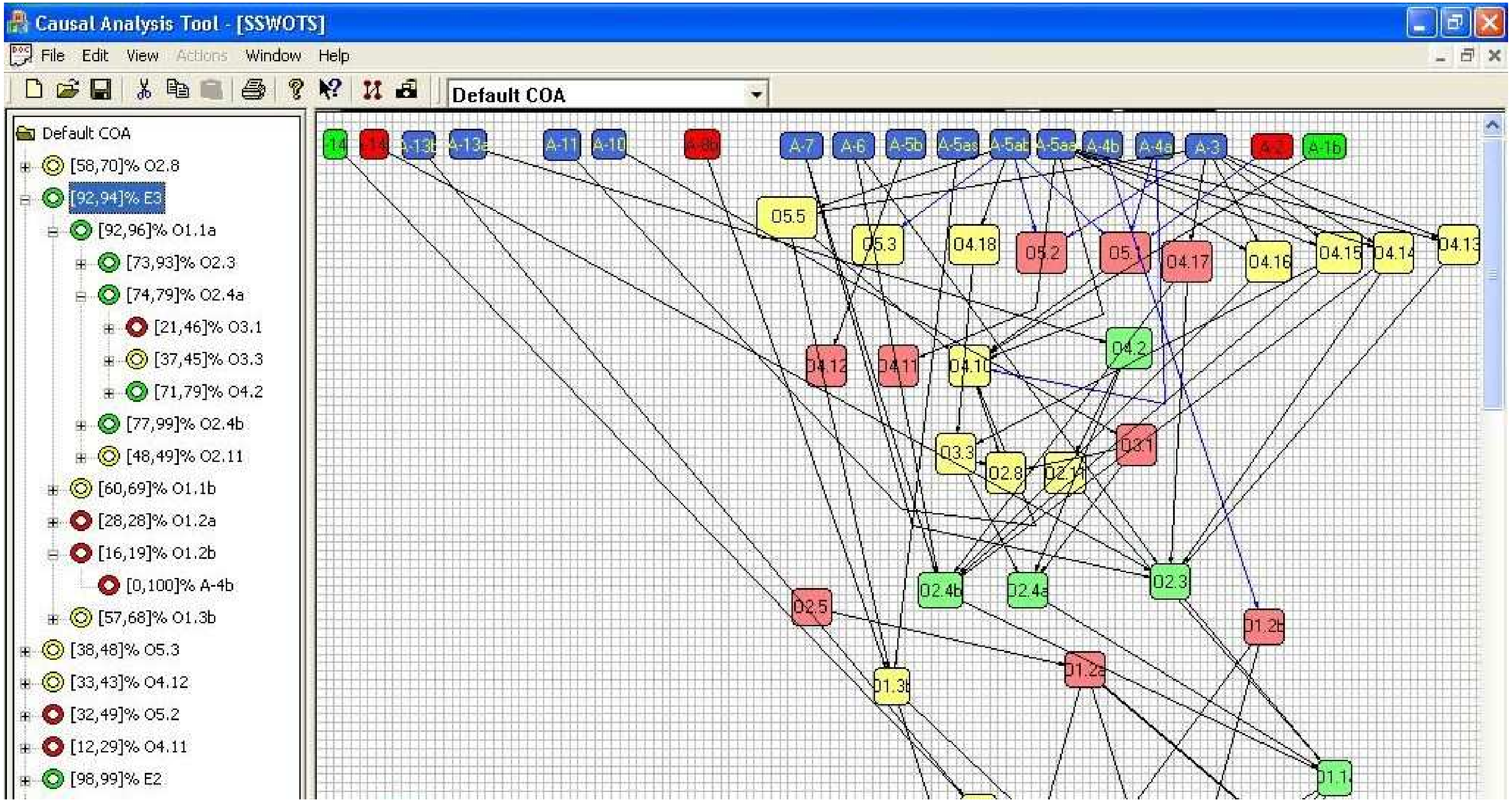

Figure 2: A portion of the causal model for “Operation SSWOTS.”

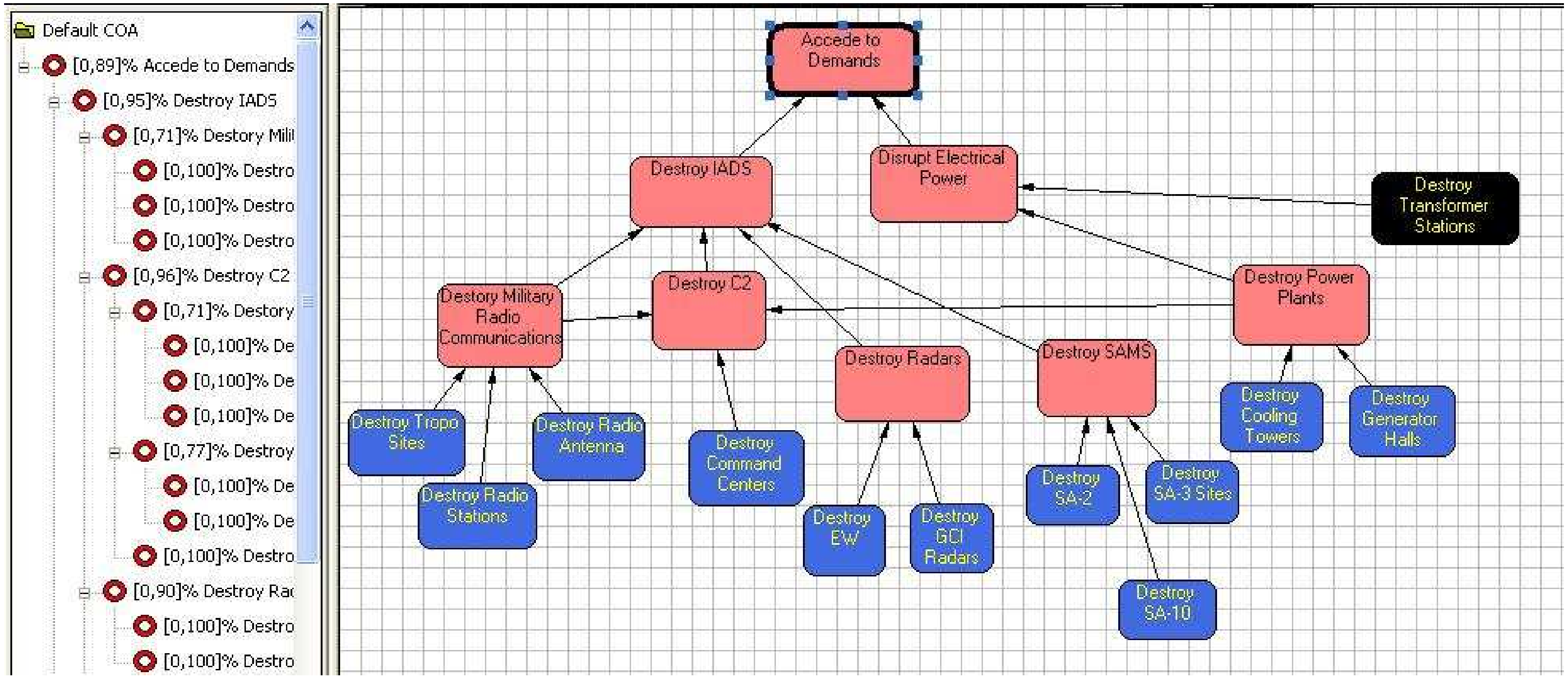

Figure 3: The causal model for "Operation OctMod," with the left-hand pane showing the values that our modified version of CAT computes for the minimum and maximum probabilities of each node. Our system recommends performing the rightmost actionable item in the causal network and indicates this by highlighting the node.

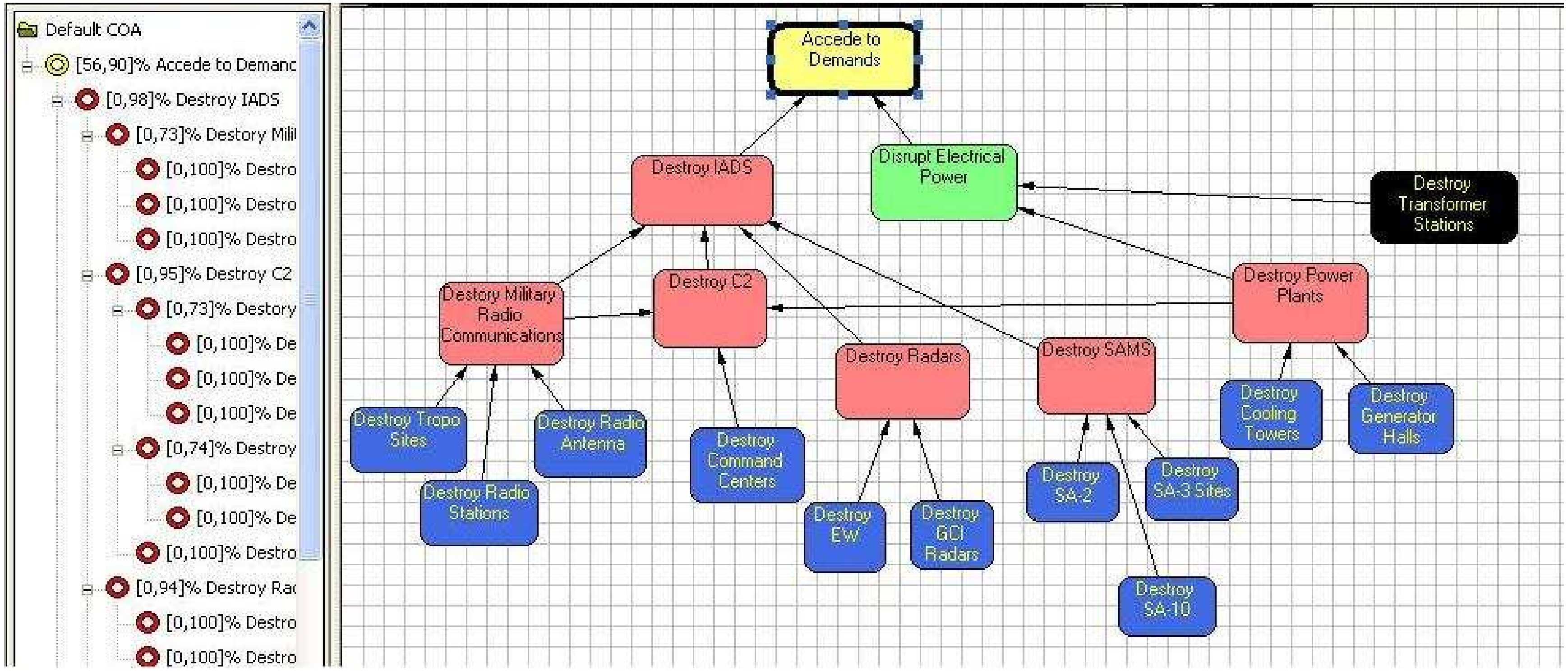

Figure 4: If the user decides to follow the recommendation highlighted in Figure 3, this substantially increases the minimum probability of achieving the goal.

these estimates over every possible combination of the decisions on the undecided actions, virtually in a simultaneous way.

Having computed the estimates of the minimum and maximum probabilities, CAT also calculated that the best choice for us to make next is to include the action "Destroy Transformer Stations," so it highlighted this action in black as shown in Figure 3. This action is the one with the greatest estimate of increasing the probability of the goal node. Then we, following CAT's recommendation, included the action in the plan, and asked CAT to analyze the causal model again. As shown in Fig. 4, including this action in the plan increases the minimum probability of the goal node from 0% to 56%. At 90%, the node's maximum probability is the same as before except for a 1% difference due to random variation in CAT's simulation. The reason for such an increase in the minimum probability of the goal node is that 56% represents an estimate of the probability of the goal when the recommended action is included in the plan, and the rest of the actions are excluded. The maximum probability of the goal does not change because it is the probability of the goal when all of the actions are included in the plan.

At this point, we again requested a recommendation for what to do next. The iterative planning process continued in this manner until we have made a decision for every actionable item. In the case we followed all of our system's recommendations, the result was a plan whose probability of success is as high as possible (i.e., both minimum and maximum probabilities of the goal is about 90%), in which all of the actionable items are included. The entire process took just a few minutes.

5 Related Work

In this section, we describe some of the knowledge systems that are designed to support coalition operations, and compare their action-planning techniques with our approach using CAT. We also describe two knowledge-based systems (namely CYPRESS [14] and HICAP [11]) that were developed for generating courses of actions under certain conditions of uncertainty. Although CYPRESS and HICAP are not originally designed for coalition operations, both systems can easily be extended for that purpose.

The CADET system [4] is a knowledge-based tool planning tool that can automatically generate courses of actions in coalition environments. The system is capable of modeling hetergeneous assets and tasks, coordinating team efforts, and generating team action plans in adverserial environments. In that respect, CADET can be considered as a very useful planning tool for coalition operations. An important difference between our approach with CAT and the CADET system is that, to the best of our knowledge, CADET is not capable of performing probabilistic analyses of cause and effect relationships between the events that may or may not occur during a coalition operation, and therefore, it is not capable of reasoning about optimality in generation of the courses of actions.

[13] describes a knowledged-based system for forming coalitions in order to achieve the given objectives. This system, called CPlanT, is an agent-based system in which the agents form alliances according to the information they have about the world and the information they have about the other agents in the world. In this model, the agents prefer to form coalitions within the particular alliances they are involved with, since allied agents know about each other, and therefore, substantial communication overhead is avoided when the coalition is formed within the allience. Once a coalition is formed, team-action planning is done by determining how each team member will contribute to achieving the goals. This task is accomplished by a coordinator agent, which decomposes the goal into subgoals, creates a course of action for each participant agent, and distributes these subgoals to the agents in a contract proposal.

The Coalition Agents Experiment (the CoAX Project) [1] also aims to design an agent-based system for coalition operations. This project aims to provide a rapid integration of agent systems in order to improve interoperability and support human situation awareness without going through a detailed planning process involving the participating agents. Using this system, the human users can develop action plans in various levels of abstraction, and execute those plans in the world. The system also includes agents that represent the other entities in the world other than the coalition members. The behavior of these agents may have an influence on the action plan generated for the coalition members, so the system allows for revising the generated plans, and deconfliction and adjustment of the revised plans via the human users.

Other examples of the agent-based approaches for supporting coalition operations include [2, 3, 6, 7] and others. To the best of our knowledge, one important issue in distributed agent-based computing for coalition operations is that the substantial computation required for coordination and information sharing among the agents in a multi-agent system. Our approach sidesteps this issue by requiring a single causal model to be built by the human experts for possible actions of all potential participants of a coalition to achieve some objectives in the world. Like the CoAX project, our approach is a mixed-initiative technique; human experts are responsible and in control of making the decisions, and our system helps them analyze the uncertainty in the world so that they can generate optimal or near-optimal plans.[3]

A difficulty in our approach might be a practical one: the causal models developed for real coalition operations could be so huge as to incur too much computational overheadin CAT's probabilistic analysis algorithms. Although this was not the case in our preliminary experiments, the causal models in those examples were rather small. It will be really interesting to test our system with real scenarios and real users, and we intend to do so in the near future.

We are also aware of two knowledge-based systems in which users can perform course-of-action planning in a mixed-initiative way and under certain conditions of uncertainty. CYPRESS [14] is a domain-independent framework for planning in dynamic and uncertain environments. The system is composed of several components responsible for generative planning, reasoning about uncertainty, and plan execution. It is capable of performing both probabilistic and possibilistic (fuzzy-logic based) uncertainty analyses. CYPRESS is similar to our approach in that it uses simulation techniques to compute lower and upper bounds on probabilities. HICAP [11] is an interactive cased-based plan authoring system developed for Noncombatant Evacuation Operations (NEOs). HICAP's representation of cases provides a way to reason about certain kinds of uncertainties, but not to reason about probabilities in the way our approach does. CYPRESS and HICAP are not originally designed for coalition operations; however, they can be easily extended to operate in distributed environments and generate courses of actions for coalition operations.

6 Conclusions

In this paper, we have described a new technique for interactive course-of-action planning under conditions of uncertainty. Our approach is based on the use of CAT (Causal Analysis Tool). CAT was developed by the Air Force Research Laboratory and is in use by a number of military organizations for creating and analyzing causal models.

To do planning in CAT, a user begins with a causal model of the domain in which some of the nodes represent actionable items, and makes decisions about which actions to include in the plan and which not. One of the biggest problems is the exponentially large number of combinations of actionable items: there are far too many of them for users to analyze each one.

To provide a solution to this problem, we have developed a way to quickly compute estimates for the minimum and maximum probabilities of success associated with a partial plan, and use these probabilities to make recommendations about which actions should be included and excluded in order to produce a complete plan with an exponential reduction in the amount of time required. We have implemented this approach in CAT. Our preliminary experiments with this version of CAT showed that our approach looks promising: CAT generated recommendations that produced complete plans with the highest possible probability of success.

We are currently performing an extensive theoretical and experimental analysis of our technique to determine its strengths and its weaknesses. Furthermore, we also intend to extend the technique for reasoning about time. In that respect, we already started extending our implementation in CAT to evaluate our preliminary ideas on probabilistic planning with time. Our ultimate objective is to develop a comprehensive theory of planning with probabilities and time.

Acknowledgments

This work was supported in part by the following grants, contracts, and awards: Air Force Research Laboratory F30602-00-2-0505, Army Research Laboratory DAAL0197K0135, and Naval Research Laboratory N00173021G005. The opinions expressed in this paper are those of authors and do not necessarily reflect the opinions of the funders.

References

[1] D.N. Allsopp, P. Beautement, J. M. Bradshaw, E. H. Durfee, M. Kirton, C. A. Knoblock, N. Suri, A. Tate, and C. W. Thompson. Coalition agents experiment: Multiagent cooperation in international coalitions. *IEEE Intelligent Systems*, 17(3):26–35, 2002.

[3] However, our system is also capable of generating a complete and near-optimal course of action, if it is asked to.

[2] G. Edwards, B. Kettler, K. Olin, and B. Tsurutani. Agents on the semantic object web: Information management for coalition operations. In *Proceedings of the Second International Conference on Knowledge Systems for Coalition Operations*, pages 42–48, Toulouse, France, 2002.

[3] M. Fletcher. Jack: A system for building holonic coalitions. In *Proceedings of the Second International Conference on Knowledge Systems for Coalition Operations*, pages 49–60, Toulouse, France, 2002.

[4] L. Ground, A. Kott, and R. Budd. A knowledge-based tool for planning of military operations: The coalition perspective. In *Proceedings of the Second International Conference on Knowledge Systems for Coalition Operations*, pages 195–203, Toulouse, France, 2002.

[5] M. Henrion. Propagating uncertainty by logic sampling in bayesian networks. Technical report, Department of Engineering and Public Policy, Carnegie-Mellon University, 1986.

[6] E. Hsu. A group-oriented framework for coalitions. In *Proceedings of the Second International Conference on Knowledge Systems for Coalition Operations*, pages 62–72, Toulouse, France, 2002.

[7] M. Klusch and A. Gerber. Dynamic coalition formation among rational agents. *IEEE Intelligent Systems*, 17(3):36–47, 2002.

[8] J. F. Lemmer and D. Gossink. Recursive noisy-or: A rule for estimating complex probabilistic causal interactions. *IEEE Transactions on Systems, Man, and Cybernatics*, 2004. Reference Number: SMCB-E-10152003-0493.R1, To appear.

[9] John F. Lemmer. Causal modeling. In *Ninth International Conference on Uncertainty In AI (UAI-93)*. Morgan Kaufmann, 1993.

[10] John F. Lemmer. The causal markov condition: Fact or artifact? *SIGART*, 7(3), 1996.

[11] H. Muñoz Avila, D. Aha, L. Breslow, and D. Nau. HICAP: an interactive case-based planning architecture and its application to noncombatant evacuation operations. In *IAAI-99*, pages 870–875, 1999.

[12] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Fransisco, CA, 1988.

[13] M. Pechoucek, V. Marik, and J. Barta. Knowledge based approach to ootw coalition formation. *IEEE Intelligent Systems*, 17(3):17–25, 2002.

[14] D. E. Wilkins. Planning and reacting in uncertain and dynamic environments. *Journal of Experimental and Theoretical AI*, 7(1):197–227, 1995.