Interactive Planning under Uncertainty with Causal Modeling and Analysis

Ugur Kuter
University of Maryland,
Department. of Computer Science,
College Park, Maryland, 20742, USA
email: ukuter@cs.umd.edu
phone: 301-405-2716
fax: 301-405-6707

Dana Nau
University of Maryland,
Dept. of Computer Science
and Institute for Systems Research
College Park, MD 20742, USA
email: nau@cs.umd.edu
phone: 301-405-2684
fax: 301-405-6707

John F. Lemmer
Rome Laboratory
525 Brooks Road
Rome, NY 13441, USA
email: John.Lemmer@rl.af.mil
phone: 315-330-3657
fax: 315-330-2563

Abstract

This paper describes a new technique for interactive planning 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 Bayes networks.

In order to use CAT as a tool for planning, 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 is exponential. 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 a way to quickly compute the minimum and maximum 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 the plan that has the highest probability of success. This provides an exponential reduction in amount of time needed to find the best plan.

Problem and Significance

A major feature of military plans is the huge amount of uncertainty they contain. This uncertainty is often referred to as the "fog of war." There are many sources for this uncertainty; perhaps the major source of this uncertainty is the relationship between cause and effect. For example: *At a tactical level, sorties are flow 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, destruction of the Taliban Army was intended ultimately to reduce world-wide terrorism. Did it?*

Due to this uncertainty, the number of possible plans for carrying out a military operation successfully can be quite large. Quick and accurate decision making on which of the series of actions to take is a very important task and it is very hard. The decision making process highly relies on the information gathered about those factors, which is often uncomplete. A typical military plan involves many such sources of uncertainty. 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 bridge destroyed by air power can be rebuilt or bypassed.

This paper describes the tool that we are developing to help manage this uncertainty in order to develop 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. In order to represent plans using CAT's causal networks, all of the *actionable items*, i.e., the actions that might potentially appear in a plan, are represented as nodes within the causal network. Thus, each possible combination of actionable items is a possible plan. 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.

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. One result of this problem is that users of CAT can spend days arguing about which subsets of actionable items to use as their plans.

As described in this paper, we have developed a new approach for overcoming this combinatorial blowup. Our approach exploits conditional independence within the causal network in order to compute quick and accurate feedback to the user about how the best way to extend a partial plan into a complete plan. We summarize the theory underlying our approach, describe how we have implemented it by modifying CAT, and give examples of its operation in order to demonstrate the effectiveness of our approach.

AI Technology

Causal Analysis Tool (*CAT*) is a system developed by the Air Force Research Laboratory (AFRL) for use in creating, modifying and analyzing causal models of military operations. CAT is in active use by several strategic-level organizations within the US Air Force. 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 very brief summary of the theory behind CAT; for details see (Lemmer 1996).

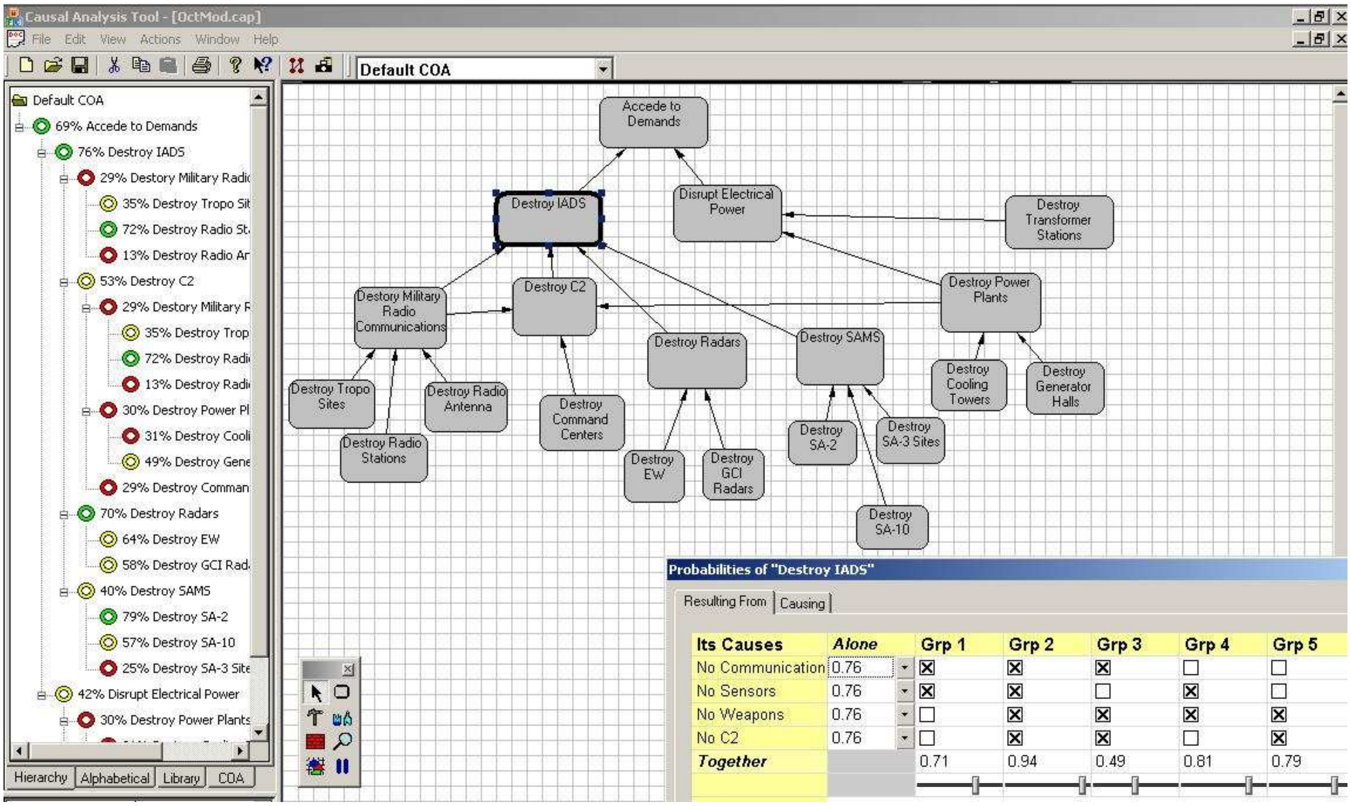

Figure 1: A causal model named "Operation OctMod" which represents the plan that was used against Milosevic in the Bosnia-Herzegovina war. The window on the right-hand side of the screen shows a portion of the probability table stored in the highlighted node. The actionable items are the twelve nodes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

CAT is based on the use of causal models; CAT provides tools to enable 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 directional graph (e.g., see Fig. 1) in which the nodes represent *events* and the edges (which are called *signals*) represent causal and inhibitory relationships between events. A signal is *causal* if it increases an event's probability of occurrence, and *inhibitory* if it reduces an event's probability of occurrence.

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. More specifically, for every event e in a causal model M, let $causes(e)$ be the set of all causal signals affecting e, and $inhibitors(e)$ be the set of all inhibitory signals affecting e. Then, for each e, the user may specify the following probabilities, which are similar but not identical to the probabilities one would specify in a Bayesian network (for the mathematical details, see (Lemmer 1996)):

- A *causal probability* $P_c(e, s_k)$, for each $s_k \in causes(e)$. Intuitively, this represents the extent to which the signal s_k increases e's probability of occurring. The user may also specify causal probabilities of e given various combinations of s_k's.
- A *inhibiting probability* $P_i(e, s_k)$, for each $s_k \in inhibitors(e)$. Intuitively, this represents the extent to which the signal s_k reduces e's probability of occurring.The user may also specify inhibitory probabilities of e given various combinations of s_k's.
- A *leak probability* $P_l(e)$. Intuitively, this is the probability that e will occur if none of the signals occur. It represents the effects of events that are not modeled in M explicitly.
- An *effectual probability* $P_e(s|e)$ for each signal affected by e. Intuitively, this is the probability that the event e will cause the signal s will occur.

In CAT, users can specify these probabilities by filling in the probability tables for every event in the causal model. For example, Figure 1 shows the probability table used for specifying the causal probabilities for the event "Destroy IADS" in that model. This probability table tells us that each of the signals "No Communications", "No Sensors", "No Weapons", and "No C2" will cause this event with probability 0.76. The user can specify causal probabilities for the event "Destroy IADS" given various groups of its causes by using the "group" check-boxes shown in the figure.

Probability Analysis in CAT

When a user gives a causal model M to CAT and asks CAT to perform a probability analysis on M, CAT does so in the following two phases:

Compilation Phase. In this phase, CAT translates M into a Bayesian network $B(M$ whose nodes include the events and the signals of M. CAT computes the following conditional probabilities for the nodes of $B(M)$:

- *Causal conditional probabilities.* For each event e and for each subset $C \subseteq causes(e)$, CAT computes a conditional probability $P(e|C)$ using the causal probabilities $P_c(e, s_k)$ specified by the user for each $s_k \in C$.
- *Inhibiting conditional probabilities.* For each event e and for each subset $I \subseteq inhibitors(e)$, CAT computes a conditional probability $P(e|I)$ using the inhibiting probabilities $P_i(e, s_k)$ specified by the user for each $s_k \in I$.
- *Conditional probabilities of signals.* For each signal s in M, CAT uses $P(s|e) = P_e(s, e)$, where $P_e(s, e)$ is the effectual probability specified in the causal network.

Simulation Phase. To calculate the probabilities of occurrence the events and signals modeled in $B(M)$, CAT does Monte Carlo simulations in which it repeatedly simulates their occurrence or nonoccurrence.[1]

In this paper, we will assume that M (and thus $B(M)$) is acyclic (although CAT can handle certain kinds of cyclic networks as well). Thus, each node n of $B(M)$ has a height $h(n)$, which is the length of the longest path from any actionable item to n. The simulation proceeds by simulating the occurrence or nonoccurrence of each node at height i, for $i = 1, 2, \ldots, h$, where h is the largest height of any node in $B(M)$.

Here is how CAT decides, for each event e, whether e will occur during the simulation. During a particular run of the simulation, let $C = \{c_1, \ldots, c_j\}$ be the set of all causes of e that have occurred, and $I = \{i_1, \ldots, i_k\}$ be the set of all inhibitors of e that have occurred. If I were empty, then we would want e to occur because of its causes with conditional probability $P(e|C)$, or because of unmodeled external factors with conditional probability $P_l(e)$. However, in either case, if I is nonempty then the occurrence of e may be inhibited with conditional probability $P(e|I)$. Thus, the probability that CAT makes e occur in the simulation is

$$(1 - P(e|I))[1 - (1 - P(e|C))(1 - P_l(e))].$$

Here is how CAT decides, for each signal s, whether s will occur during the simulation. Let e be the event that may cause s. If e does not occur occur during the simulation, then CAT will not make s occur either. However, if e occurs during the simulation, then CAT will make s occur with probability $P(s|e)$.

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

CAT runs such simulations repeatedly. As it does so, keeps track of how frequently each event occurs. These statistics provide an estimate of $P(e)$ for every event e in the network. 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 $P(e)$ get progressively more accurate, and CAT updates its display accordingly. The user may stop the simulations whenever he/she feels that the estimates have become sufficiently accurate.

Planning using CAT

The planning process in CAT starts with creating a causal model in which one or more nodes are *actionable items*, i.e., actions that one may or may not want to perform. In this context, a *plan* corresponds to a set of yes-or-no choices: for each each actionable item, one must choose whether to include it in the plan or exclude it from the plan. In making these choices, the user's objective is to create a plan that causes some set of nodes to occur that represent the *goals* of the plan. For simplicity, in this paper we assume that there is exactly one such goal.

The user's objective is to find a plan that maximizes the probability of the goal. For each plan, one can use CAT to determine the probability that the plan will achieve the goal, in the following manner. First, set the probability of each actionable item to 1 if the item is included in the plan or 0 if it is not included in the plan, and then tell CAT to perform the analysis described in the previous section.

Thus, planning takes place as an iterative process in which users repeatedly make decisions about which actionable items to include and which ones to exclude, tell CAT to determine the probability of achieving the goal, and 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.[2]

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 military operations can be quite large and complex, and since military 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 one (this assumption is rather optimistic: if the network is sufficiently large, CAT might take minutes or even hours). Then the total time needed to analyze all of the plans is about 11,651 hours, which is more than 485 days.

[2]Note that this plan is not necessarily the one that includes all possible actionable items. If the causal model contains inhibitory signals, then some actionable items may reduce the probability of achieving the goal.

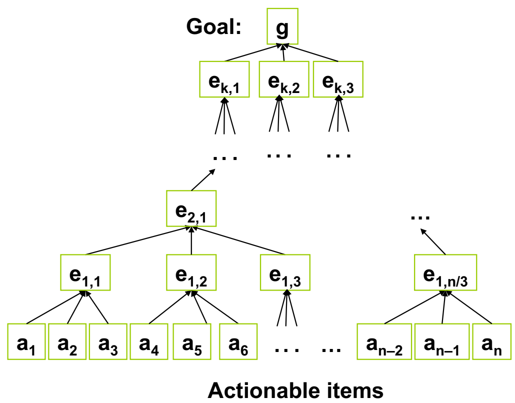

Figure 2: A simple causal model in which there are n actionable items $a_1, \ldots, a_n$, various events $e_{i,j}$ that are caused by the actionable items, and a single goal g. Each event other than $a_1, \ldots, a_n$ has at most three predecessors.

Innovation

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.

In this section, we describe a modified version of CAT that can give the following feedback to the user at each iteration of the process: (1) an evaluation of the minimum and maximum probabilities of success that can be attained with the current partial plan, and (2) a recommendation for what choice to make next in order to get the plan that has the highest possible probability of achieving the goal.

The following subsections describe (1) how to compute the minimum and maximum probabilities of success of a partial plan, and (2) how to use these probabilities to recommend which item to include or exclude next.

Minimum and Maximum Probabilities of Success

Consider a causal model M in which the set of actionable items is $A = \{a_1, \ldots, a_n\}$. Suppose the user has already made yes-or-no choices for some subset $D \subseteq A$. For each $a_i \in D$, if the choice is to include a_i, then this means that the action a_i will be performed, i.e., that $P(a_i) = 1$. If the choice is to exclude a_i, then this means that the action a_i will not be performed, i.e., that $P(a_i) = 0$.

For each $a_i \in A - D$, no decision has been made yet, so $P(a_i)$ may be either 0 or 1. Thus, the minimum and maximum possible probabilities of 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 that is not an actionable item. One way is the brute-force approach: run CAT's probability analysis on M repeatedly, once for every combination of probabilities $\{P(a_i) \in \{0,1\} : a_i \in A - D\}$. 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 $m = |D|$. For example, in the causal network in Fig. 2, suppose $n = 25$ and $m = 3$. Then there are 2^{22} different plans, the same number as in the previous example. As in the previous example, suppose that CAT takes 10 seconds to analyze each one. Then as before, the total computation time will be 485 days.

In the above example, $P_{\min}(e)$ and $P_{\max}(e)$ can be computed much more quickly by taking advantage of conditional independence in the causal network. We start with computing $P(e_{1,1})$. $P_{\min}(e_{1,1})$ and $P_{\max}(e_{1,1})$ can be computed by calculating $P(e_{1,1})$ eight times: once with $P(a_1) = 0$, $P(a_2) = 0$, and $P(a_3) = 0$, once with $P(a_1) = 0$, $P(a_2) = 0$, and $P(a_3) = 1$, and so forth until we compute $P(e_{1,1})$ of the event $e_{1,1}$ for all possible choices for a_1, a_2 and a_3. Then $P_{\min}(e_{1,1})$ and $P_{\max}(e_{1,1})$ are the minimum and the maximum probability values computed for $P(e_{1,1})$.

Similarly, we can calculate $P(e_{2,1})$ eight times: once with $P(e_{1,1}) = P_{\min}(e_{1,1})$ and $P(e_{1,2}) = P_{\min}(e_{1,2})$ and $P(e_{1,3}) = P_{\min}(e_{1,3})$, once with $P(e_{1,1}) = P_{\min}(e_{1,1})$ and $P(e_{1,2}) = P_{\min}(e_{1,2})$ and $P(e_{1,3}) = P_{\max}(e_{1,3})$, once with $P(e_{1,1}) = P_{\min}(e_{1,1})$ and $P(e_{1,2}) = P_{\max}(e_{1,2})$ and $P(e_{1,3}) = P_{\min}(e_{1,3})$, once with $P(e_{1,1}) = P_{\min}(e_{1,1})$ and $P(e_{1,2}) = P_{\max}(e_{1,2})$ and $P(e_{1,3}) = P_{\max}(e_{1,3})$, and so forth. Then $P_{\min}(e_{2,1})$ and $P_{\max}(e_{2,1})$ are the minimum and the maximum probability values computed for $P(e_{2,1})$.

Continuing in this manner, we can compute $P_{\min}(e_{i,j})$ and $P_{\max}(e_{i,j})$ for every $e_{i,j}$, and thus can compute $P_{\min}(g)$ and $P_{\max}(g)$.

In the above example, we need to consider at most eight combinations at each node. As a result, the total computation time is no more than would be required for eight calls to CAT's probability-analysis routine. Thus if we assume (as before) that CAT takes 10 seconds for each probability analysis of M, it is possible to perform the above computation in about 80 seconds. This is substantially better than the 485 days required by the brute-force approach!

We now describe how to modify CAT to perform the kinds of computations described above. We begin by noting that CAT's simulations can be described using a boolean random variable $x(e) \in \{0,1\}$ for each event e. Thus, rather than saying that an event e occurs during the simulation with probability $P(e)$, we can say that CAT assigns $x(e) \in \{0,1\}$ with probability $P(e)$.

Our modification of CAT is to replace $x(e)$ with *two* random variables $x_{\min}(e)$ and $x_{\max}(e)$ such that CAT assigns $x_{\min}(e) = 1$ with a probability that is the minimum possible value of $P(e)$ over all choices for the actionable items, and it assigns $x_{\max}(e) = 1$ with a probability that is the maximum possible value of $P(e)$ over all choices for the actionable items. We now describe how to accomplish this.

As before, the simulation proceeds by simulating the occurrence and nonoccurrence of each node at height i, for $i = 1, 2, \ldots, h$, where h is the largest height of any node in $B(M)$. The nodes of height $i = 0$ are the actionable items. For each such node e, our modified simulation assigns $x_{\min}(e) = 1$ if the user has included e in the partial plan, and $P_{\min}(e) = 0$ otherwise. Similarly, it assigns $x_{\max}(e) = 0$ if the user has excluded e from the partial plan, and $P_{\max}(e) = 1$ otherwise.

For $i = 1, 2, \ldots$, the simulation proceeds as follows: let s be a signal that may be caused by some event e, and suppose the simulation has progressed far enough to assign values to $x_{\min}(e)$ and $x_{\max}(e)$. From conditional independence, it follows that $P(s)$ depends only on e. Thus, the simulation assigns $x_{\min}(s) = 1$ with probability $P(s|e)$ if $x_{\min}(e) = 1$, and $x_{\min}(s) = 0$ otherwise. Similarly, it assigns $x_{\max}(s) = 1$ with probability $P(s|e)$ if $x_{\max}(e) = 1$, and $x_{\max}(s) = 0$ otherwise.

Let e be an event that is not an actionable item. Let $s_1, s_2, \ldots, s_b$ be all of the signals that may affect e, and suppose the simulation has progressed far enough to assign values to $x_{\min}(s_i)$ and $x_{\max}(s_i)$ for each i. From conditional independence, it follows that $P(e)$ depends only on $s_1, \ldots, s_b$. Thus, the set of possible probabilities for e is

$$\begin{aligned}\{P(e|x(s_1), x(s_2), \ldots, x(s_b)) :& \\ & x(s_1) \in \{x_{\min}(s_1), x_{\max}(s_1)\}, \\ & x(s_2) \in \{x_{\min}(s_2), x_{\max}(s_2)\}, \\ & \ldots, \\ & x(s_b) \in \{x_{\min}(s_b), x_{\max}(s_b)\}\}.\end{aligned}$$

Let p_0 and p_1 be the minimum and maximum values in this set. Then the simulation assigns $x_{\min}(s) = 1$ with probability p_0 and it assigns $x_{\max} = 1$ with probability p_1.

Like the original version of CAT, the modified version keeps running its simulations for as long as the user wishes. For each event e, it keeps track of the average values of $x_{\min}(e)$ and $x_{\max}(e)$. These averages are estimates of the minimum and maximum probabilities of e over all possible choices for the actionable items. These probabilities are displayed to the user in the hierarchical display shown in the left-hand pane of Fig. 3.

The Feedback Mechanism

The previous section described an efficient way to compute $P_{\min}(g)$ and $P_{\max}(g)$, given a causal network M and a partial plan. We now describe how this computational technique makes it possible for us to give recommendations to the user about the best choice to make at each step of the iterative planning process, in order to generate a plan that maximizes the probability of achieving the goal.

First, we point out that we can use the computational technique of the previous section to compute $P_{\max}(g|a_i)$: we just do the computation with $P(a_i) = 1$. To get $P_{\max}(g|\neg a_i)$, we can do the same thing with $P(a_i) = 0$.

Next, suppose we compute $P_{\max}(g|a_i)$ and $P_{\max}(g|\neg a_i)$ for every $a_i \in A - D$. Using the conditional independence conditions in the causal network, we can show that for every $a_i \in A - D$, either $P_{\max}(g) = P_{\max}(g|a_i)$ or $P_{\max}(g) = P_{\max}(g|\neg a_i)$. If $P(g) = P_{\max}(g|a_i)$, then our recommendation to the user is to include a_i in the plan; if $P(g) = P_{\max}(g|\neg a_i)$ for some a_i, then our recommendation to the user is to exclude a_i from the plan.

The above technique generates a sequence of recommendations for producing the plan with the largest possible probability of achieving the goal. The total computation time 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. 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, consider again the causal network in Fig. 2. As before, let us suppose that $n = 25$ and $m = 3$, and the original version of 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 30 minutes.

Implementation and Evaluation

We have created a modified version of CAT that computes the probabilities and recommendations described in the previous section.

To evaluate our implementation, we have used unclassified versions of causal models for two military operations scenarios. One is the "Operation OctMod" model shown in Fig. 1, which is a representation of the plan was used against Milosevic in the Bosnia-Herzegovina war. The other is a "scrubbed" version of a much larger model that was developed for the war in Afghanistan. In each case, it was possible to use our modified version of CAT to develop plans in just a few minutes.

We now describe a sample user session with the OctMod example. Initially, the user has not decided which of the actionable items to include in the plan or exclude from it, so all of the actionable items are marked as *unknown*. Suppose the user asks our modified version of CAT to analyze the causal model and make a recommendation. Then CAT calculates the maximum and minimum probabilities shown in the left-hand pane in Fig. 3. Note in particular that the range of probabilities for the goal node (the "accede to demands" node) is quite broad; the minimum and maximum probabilities for this node are 0% and 89%. Furthermore, CAT calculates that the best choice for the user to make next is to include the action "Destroy Transformer Stations," so it highlights this action in black as shown in Fig. 3.

Suppose the user decides to follow CAT's recommendation and include the action in the plan, and then asks 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 Monte Carlo simulation. At this point, the user can again request a recommendation for what to do next.

The iterative planning process continues in this manner until the user has made a decision for every actionable item.

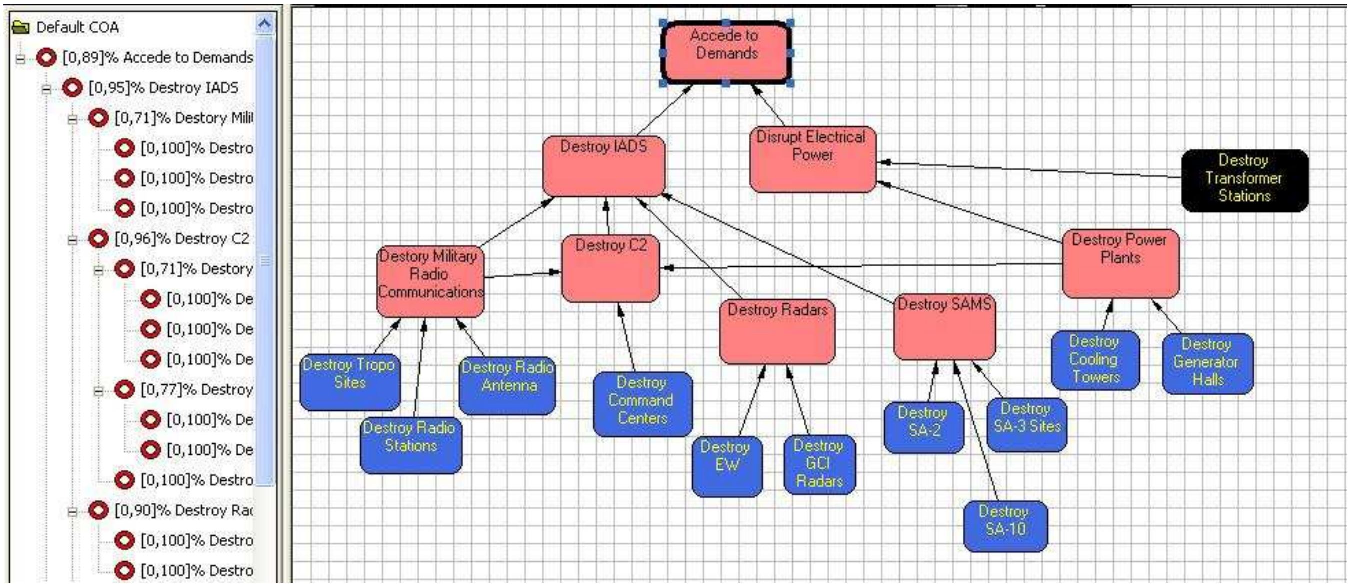

Figure 3: Here is the causal model for Operation OctMod, with the left-hand pane showing the values that our modified version of CAT computes for the minimum and maximum probabilities of each node. Our system recommends performing the rightmost actionable item in the causal network, and indicates this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

If the user follows all of our system's recommendations, the result will be a plan whose probability of success is as high as possible (i.e., about 89% or 90%). The entire process takes just a few minutes.

Related Research

The best known approach for planning under uncertainty is based on Markov Decision Process (MDPs); examples include (Kushmerick, Hanks, & Weld 1995; Hanks & McDermott 1994; Draper, Hanks, & Weld 1993; Poole 1995; Singh & Cohn 1998; Dean & Kanazawa 1989; Howard & Matheson 1984). For a survey of this approach see (Boutilier, Dean, & Hanks 1999). MDPs can be solved by using dynamic programming algorithms ((Barto, Bradtke, & Singh 1995)). To reduce the computational overhead of solving a planning problem represented as an MDP, (Givan, Leach, & Dean 1997) proposes a concept called *Bounded Parameter MDPs*, which assigns probability intervals representing the assertion that the probability of the state of the planning algorithm must be within that interval. The use of probability intervals are similar to our reasons for efficiently computing the minimum and maximum probabilities for events, but they are closed real intervals for probability distributions whereas the minimum and maximum probabilities in our case represent the set of single probability values of occurrence for an event (or a signal). (Littman 1997) reviews several such representations and shows that these approaches are "expressively equivalent", meaning that planning problems formulated in one representation can be converted to another representation in polynomial time.

Several other approaches have been developed for planning under conditions of uncertainty. For example, (Bertoli *et al.* 2001) introduces a planning system that inserts *sensing actions* in the plan it generates, in order to gather information when the plan is being executed. (Cimatti & Roveri 2000) presents a search technique that relies on the use of symbolic model checking, Binary Decision diagrams, and heuristic search. (Bonet & Geffner 2000) introduces the GPT (General Planning Tool), which is a system that provides a high-level language for expressing actions, sensors, and planning goals, along with set of heuristic search algorithms that generate plans based on the problem description given by that language. (Smith & Weld 1998) presents a planning technique that is built on top of the GRAPHPLAN planning algorithm ((Blum & Frust 1997) and is capable of coping with certain kinds of uncertainty in the world. Finally, Bridge Baron (Smith, Nau, & Throop 1998) uses task-decomposition planning techniques to generate game trees in the game of bridge and analyze those game trees probabilistically; this approach worked well enough to win the 1997 world championship of computer bridge.

Two of the biggest differences between our problem requirements and the ones addressed in the traditional literature on AI planning are our domain's requirements for user interaction and causal modeling:

- *User Interaction.* Our problem domain requires *mixed-initiative planning* (Burstein & McDermott 1994). It is important for users to be control the planning process and be able to modify the plan being developed at any time during the planning process.

  As a similar approach to ours, (Wilkins 1995) describes the CYPRESS system, a domain-independent framework for planning with distributed agents in dynamic and uncertain environments. The Gister-CL component integrated in CYPRESS implements a suite for analyzing uncertain information about the world and possible ac-

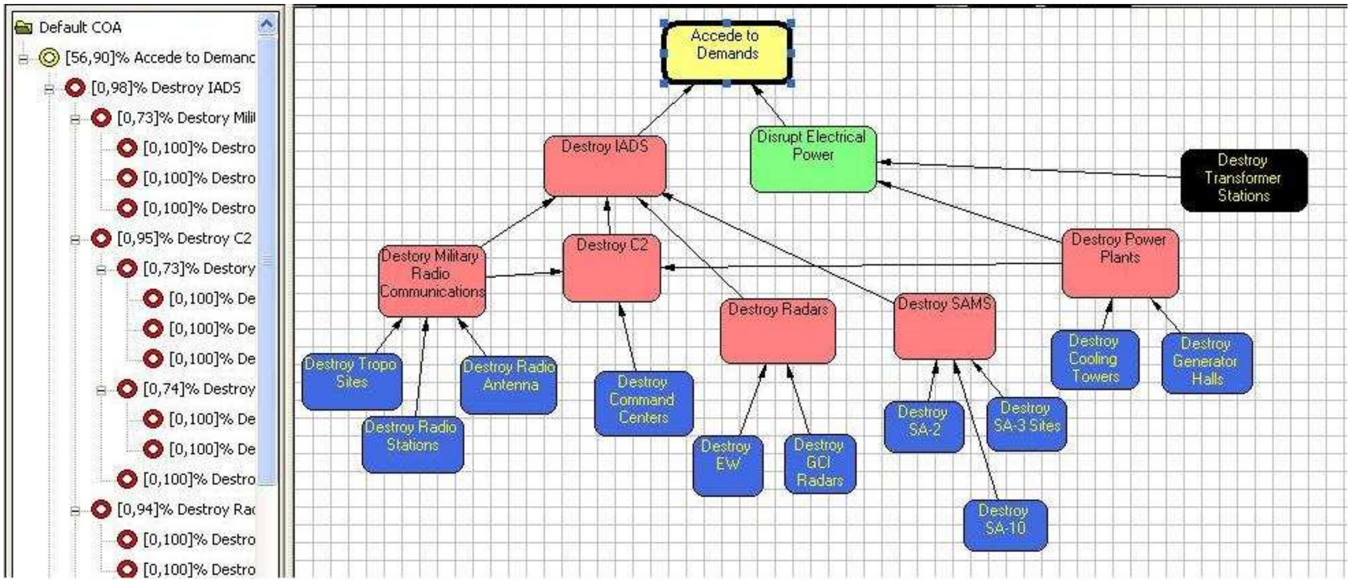

Figure 4: Here, the user has decided to follow the recommendation given by our system in Fig. 3. As shown in the left-hand pane, this substantially increases the minimum probability of achieving the goal.

tions during both planning and execution. Cypress has been used for military operations as well as for fault diagnosis problems. The most important difference between our planning approach and CYPRESS as follows: CYPRESS incorporates the automated planning system SIPE-2 (Wilkins 1988; 1990) to generate plans whereas, in our approach, users are generating plans themselves possibly by using the feedback provided by the system.

(Muñoz Avila *et al.* 1999) describes the HICAP system, which is an interactive cased-based plan authoring system developed for a special kind of military operations called Noncombatant Evacuation Operations (NEOs). HICAP can handle certain kinds of uncertainties by using its conversational case retrieval mechanisms, but has no way to reason about probabilities.

- *Causal Modeling.* Our problem domain requires a way for users to explicitly model probabilistically the cause-effect relationships among actionable items, events that can be caused by those actions, and events that are not under control of the users. There is a huge and well known literature on probabilistic reasoning using Bayesian networks; see (Pearl 1988; 2000; Jensen & Lauritzen 1997) for an overview to the subject. Bayesian techniques have been used in a number of applications ranging from medical decision-support (e.g. HUGIN Advisor (`http://www.hugin.com`) and DIAVAL (Diez *et al.* 1997)) to Microsoft's "Clippy" (Horvitz *et al.* 1998).

  Probabilistic reasoning techniques are also used in a number of military organizations. For this purpose, probably the most widely used probabilistic reasoning system is one called SIAM. However, not much documentation is readily available for SIAM in the open literature. The primary limitation of SIAM relative to CAT is that SIAM is not capable of doing Bayesian inference for evidential reasoning; instead, it requires independence assumptions much stricter than the ones normally needed in Bayesian networks.

Conclusions

In this paper, we have described a new technique for interactive 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 of military operations.

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 for of them users to analyze each one, and users can spend days arguing about which actionable items to include in their plans.

To provide a solution to this problem, we have developed a way to quickly compute 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 get the highest possible probability of success. This provides an exponential reduction in the amount of time required. For problems in which finding the best plan would have required days, it can now be found in minutes.

Our modified version of CAT is attracting interest among several potential users. Currently it is still a prototype that does not include the full functionality of CAT. We intend to add the missing functionality during the next few months. Once we have done so, we anticipate that our modifications will become part of the standard CAT distribution.

Acknowledgments

This work was supported in part by Air Force Research Laboratory grant F30602-00-2-0505. The opinions expressed in this paper are those of authors and do not necessarily reflect the opinions of the funders.

References

Barto, A.; Bradtke, S.; and Singh, S. 1995. Learning to act using real-time dynamic programming. *Artificial Intelligence* 72(1–2):81–138.

Bertoli, P.; Cimatti, A.; Roveri, M.; and Traverso, P. 2001. Planning in nondeterministic domains under partial observability via symbolic model checking. In *IJCAI 2001*, 473–478.

Blum, A., and Frust, M. 1997. Fast planning through planning graph analysis. *Artificial Intelligence* 90(1–2):281–300.

Bonet, B., and Geffner, H. 2000. Planning with incomplete information as heuristic search in belief space. In Chien, S.; Kambhampati, S.; and Knoblock, C., eds., 5^{th} *International Conference on Artificial Intelligence Planning and Scheduling (AIPS-2000)*, 52–61. AAAI Press.

Boutilier, C.; Dean, T.; and Hanks, S. 1999. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research* 11:1–94.

Burstein, M., and McDermott, D. 1994. Issues in the development of human-computer mixed-initiative planning systems. In *Proceedings of the ARPA/Rome Laboratory Knowledge-Based Planning and Scheduling Initiative*, 467–483. San Fransisco, CA: Morgan Kaufmann.

Cimatti, A., and Roveri, M. 2000. Conformant planning via symbolic model checking. *Journal of Artificial Intelligence Research* 13:305–337.

Dean, T., and Kanazawa, K. 1989. A model for reasoning about persistence and causation. *Computational Intelligence* 5(3):142–150.

Diez, F.; Mira, J.; Iturralde, E.; and Zubillage, S. 1997. Diaval: a bayesian expert system for echocardiography. *Articial Intelligence in Medicine* 10:59–73.

Draper, D.; Hanks, S.; and Weld, D. 1993. Probabilistic planning with information gathering and contingent execution. Technical Report 93-12-04, University of Washington, Seattle, WA.

Givan, R.; Leach, S.; and Dean, T. 1997. Bounded-parameter markov decision processes. In *Proceedings of the Fourth European Conference on Planning (ECP'97)*, 234–246.

Hanks, S., and McDermott, D. 1994. Modeling a dynamic and uncertain world i: Symbolic and probabilistic reasoning about change. *Artificial Intelligence* 66(1):1–55.

Horvitz, E.; Breese, J.; Heckerman, D.; Hovel, D.; ; and Rommelse, K. 1998. The lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, 256–265. Madison, WI: Morgan Kaufmann.

Howard, R., and Matheson, J. 1984. Influence diagrams. In *The Principles and Applications of Decision Analysis*.

Jensen, F., and Lauritzen, S. 1997. Probabilistic networks.

Kushmerick, N.; Hanks, S.; and Weld, D. 1995. An algorithm for probabilistic planning. *Artificial Intelligence* 76(1–2):239–286. Special Volume on Planning and Scheduling.

Lemmer, J. F. 1996. Causal modeling. In *Uncertainty In AI (UAI-96)*.

Littman, M. 1997. Probabilistic propositional planning: Representations and complexity. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, 748–754.

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

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

Pearl, J. 2000. *Causality: Models, Reasoning and Inference*. Cambridge, UK: Cambridge University Press.

Poole, D. 1995. Exploiting the rule structure for decision making within the independent choice logic. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, 454–463.

Singh, S., and Cohn, D. 1998. How to dynamically merge markov decision processes. In *Advances in Neural Information Processing Systems 10*, 1057–1063.

Smith, D. E., and Weld, D. S. 1998. Conformant graphplan. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98) and of the 10th Conference on Innovative Applications of Artificial Intelligence (IAAI-98)*, 889–896. Menlo Park: AAAI Press.

Smith, S. J. J.; Nau, D. S.; and Throop, T. 1998. Computer Bridge: A Big Win for AI Planning. *AI Magazine* 19(2):93–105.

Wilkins, D. E. 1988. *Practical Planning: Extending the Classical AI Planning Paradigm*. San Mateo, CA: Morgan Kaufmann.

Wilkins, D. E. 1990. Can ai planners solve practical problems? *Computational Intelligence* 6(4):232–246.

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