Some questions about interpreting plans as multi-stage influence diagrams

John M. Agosta and Roberto Desimone (SRI International)

"Before the first step is taken, the goal is reached." [0]

This work is motivated by some general questions about working with uncertainty with classical planning tools. We describe the application, and the questions that it has raised, and in the last section, some not-entirely-related formal questions about formulating multi-stage models. In terms of results, this work is preliminary.

1 AI Planning for Oil Spill Response

We have applied the SIPE system (a classical planning tool, see [1]) to the Coast Guard's problem of oil spill response. The Coast Guard has final responsibility for cleanup actions when an oil spill occurs in coastal waters. Oil spill actions fall into two types, "deploying" clean up equipment from the site near the coast where it is stored, and once deployment is complete, "employing" the equipment at the location of the spill. We call this the "oil spill tactical response" problem.

Interestingly, for purposes of comparison in the literature, the tactical response problem has previously been formulated as a dynamic programming problem. [2]. Our intent is to incorporate some of the optimization features from a dynamic programming in the planner's formulation of the problem.

Our formulation of the domain encodes the Coast Guard's operations as a set of planning operators that 1) choose equipment from a catalog of available equipment that is distributed by site, and 2) calculate the time at which the equipment would be deployed. By its nature, the problem is a race against the spread of the oil, so that cleanup actions, to be effective, must be completed within the time predicted for the oil's landfall.

1.1 Planning against a forecast.

To represent the passage of time, the plan is "paced" by a forecast of occurrences (facts about the world) over a sequence of stages. These facts describe the transport of oil from the foundered vessel, over the sea, possibly to the shore. The facts, which express the probability distribution of the extent of the oil's spread, are labeled by the time in which they occur. The facts are generated by outside the planning system by an "oil spill trajectory model."

Each stage enters the planning process in two "modalities;" as a forecast, then as an actual occurrence. We speak of "plan time," which corresponds to "forecast time" and "re-plan" time, which would correspond to the "real-time" planning problem. Replanning is only necessary because the forecast contains uncertainty that is resolved when surveillance actions discover the actual spill trajectory. Thus these two regimes are distinguished when there is uncertainty in the forecast. For re-planning we use SIPE's built-in re-planning capability.

The introduction of surveillance information is captured in the plan by reconnaissance operators, whose effects, if successful, introduce new facts that modify the forecast. Each stage has a surveillance operator as part of the top level goals in its plot. In terms of the dynamic programming formulation, the reconnaissance operator generates the observations of the stage, and all actions taken within a stage are made at the same state of information about the world.

2 General questions about applying the planning formulation to optimization.

The previous discussion provides the general outlines of how the planning problem may be re-formulated as a stochastic dynamic programming problem with a partially observable state. The state variable at each stage, x, describes the location and quantity of spilled oil. Actions, u, are grouped by stage to form the stage decisions. One of the actions at each stage is a surveillance action whose effects determine if the state variable is observable (the variable z) or not. The planner is nonlinear - meaning that the set of actions generated are not strictly time ordered. The division into stages requires "linearization" when observations occur. This may occur in the course of planning when a fact introduced by surveillance that serves as a pre-condition on a parallel branch generates a "phantom node." The final stage occurs when oil is either cleaned up or lands on shore. Costs depend upon the equipment used and the damage done by the oil in the final stage.

2.1 Optimization as foresight

The planner generates a feasible plan, but does not try to search for the best plan. In fact, there is no concept of a cost functional to be minimized in the planner. The user might well ask how will optimization change her behavior. If we for a moment disregard the problem of optimization within a stage, optimization between stages generates "foresight" – the ability to pre-allocate resources for future use. The example in oil spill clean up is dramatic: If the oil is not contained by floating booms within the first few hours, the spread of oil is so rapid that the demand for boom increases beyond the ability to supply it. At this point it is prudent to use the remaining boom to protect sensitive areas along the shore rather than trying to contain the oil on the sea.

There are "night" stages followed by "daylight" stages, so the number of stages is roughly twice the number of days in the incident. An spill incident may run for a week or two, so there may be tens of stages in the problem. With this number of stages, optimization is only practical if the "curse of dimensionality" can be avoided, which is the motivation for going to a dynamic programming (DP) formulation.

2.2 When must "no-forgetting" be enforced?

To gain the efficiencies of DP, the stages must be "separable," in a sense that has been precisely defined elsewhere. [3] In this example, this becomes a concern only with the decision variable: The location of oil on the sea is a Markov state variable with respect to the future dispersion of the oil. And the costs are additive over the stages.

Since the state is only partially observable, the optimal decision at one stage may depend upon decisions at more than just the previous stage. This condition has been termed "no-forgetting" by Howard. To be precise, we define no-forgetting by defining the information state I_k of the decision at stage k: (This follows [3], p.100)

$$I_k = \{z_0 \dots z_k, u_0 \dots u_{k-1}\}$$

or recursively:

$$I_k = \{I_{k-1}, z_k, u_{k-1}\},\ I_0 = z_0$$

In general, (e.g., when cost of computation is not in the cost function) more information cannot make a decision-maker worse off, it just generates more cases for the decision maker to consider, which may be irrelevant to her optimal choice. If conditions can be found when no-forgetting arcs are not needed, then solution complexity would be linear in the number of stages. When the stage's state is known at the time the stage's decision is made, then there is no no-forgetting, i.e., previous choices can be disregarded, and the decision does not depend upon previous decisions made. In the stationary, infinite time-horizon case this is equivalent to a "Markov decision process." The interesting question in this program of research is how can no-forgetting be treated, and what are techniques and costs of relaxing it? This remain a question of practical and research interest.

Drawing again on the literature on stochastic, partially observable DP, the standard technique to preserve linear complexity and no-forgetting is called "state augmentation." The original problem is re-formulated to replace the unobservable state variable with new state variables that are the sufficient statistic(s) of the unobservable state. Unfortunately, the reformulated problem may not be computationally practical.

There are often domain-specific simplifications that can be made. (See, for instance [4], p. 330 and following) Here is one simplification of the oil spill problem: Since surveillance steps provide complete information about the state, the DP is Markov between the stages when observation succeeds. Thus the problem is linear in these "aggregated" states.

Similarly there are other approaches to preserving linear complexity by finding or assuming a Markov property in the decision function. In his thesis, Tatman [5] studies complexity in multi-stage influence diagrams by looking at the growth in arcs formed by arc-reversals when stages are removed. In [4] an analogy is made to the Markov decision assumption and the STRIPS planning assumption. Zhang, Qi and Poole [6] downplay the importance of no-forgetting by what they call "stepwise decomposable influence diagrams," and offer rationales for assuming forgetting between decisions.

In summary, we believe that the emerging results in multi-stage influence diagrams about decision information states will advance both representation and generation of optimal plans.

3 References

[0] From "One Road to Kembo" , p.127, in Paul Reps, "Zen Flesh, Zen Bones," (NY: Doubleday, no copyright).

[1] David Wilkins, Practical Planning (San Mateo, CA: Morgan Kaufman, 1988)

[2] Psarftis, Harilaos and Babis O. Ziogas (MIT) "A Tactical Decision Algorithm for the Optimal Dispatching of Oil Spill Cleanup Equipment" Management Science, V. 31 N. 12 (December 1985), 1475-1491.

[3] Dimitri P. Bertsekas, "Dynamic Programming," (Englewood Cliffs, NJ: Prentice Hall, 1976).

[4] T.L. Dean. , Michael P. Wellman, "Planning and Control" (San Mateo, CA: Morgan Kaufmann, 1991).

[5] J. A. Tatman, "Decision Processes in Influence Diagrams: Formulation and Analysis" (PhD thesis, EES Department, Stanford University, CA, 1985).

[6] L. Zhang, R. Qi and D. Poole, "Incremental Computation of the value of perfect information in stepwise decomposable influence diagrams" Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence, (San Mateo, CA: Morgan Kaufmann, 1993, p 400-407).