Planning With Information States: A Survey

Term Project for cs397sml — Spring 2002

Jason O'Kane
jokane@uiuc.edu

December 11, 2002

1 Introduction

Classical planning generally depends on the assumption that the outcome of each action is known. That is, even though there may be some *a priori* uncertainty about which state will result from a given action, classical planning methods assume that there is no *a postieriori* uncertainty – that after an action is taken, the resulting state is immediately known with certainty. In practice this assumption can be problematic. At best, sensors that report the current state will be subject to some noise; at worst, no state information at all will be available.

This paper surveys work on this imperfect state information problem. In Section 2, we formalize the problem. Section 3 summarizes a number of papers that address specific instances of the problem. Finally, Section 4 is a discussion of several questions that have yet to be satisfactorily answered.

2 Planning with State Uncertainty: The Problem

This section describes a general formalism for the problem of planning under state uncertainty[1]. Let us pose the problem as follows:

- X is a set of states.
- U is a set of actions. It is assumed that all actions are available from each state.
- Θ is a set of "nature actions" that models uncertainty in transitions between states.
- $f : X \times U \times \Theta \to X$ is a state transition equation.
- Y is a set of possible observations that give some information about the current state.
- Φ is a set of "nature observation actions" that models uncertainty in the observations made.
- $h : X \times \Phi \to Y$ is an observation equation that determines observations the decision maker will receive.
- A loss functional L mapping sequences of states and actions $x_1, u_1, \ldots, x_{k-1}, u_{k-1}, x_k$ to $\mathbb{R}$. The loss functional is a general mechanism by which to measure the "goodness" of an action-state sequence. It is typical to assume that the loss function is *stage-additive*. We might define a loss function for a single stage $\ell : X \times U \to \mathbb{R}$ and write $L = \sum_i \ell(x_i, u_i)$.

We assume that the decision maker has some way of characterizing the initial state x_1. For example, x_1 could be known with certainty, or there could be a region of possible starting states X_I, or some probability distribution over X could be known. Let us assume that the system's state changes in discrete steps called stages. At each stage k:

1. The system will be in some state $x_k \in X$, which is unknown to the decision maker.
2. The decision maker chooses an action $u_k \in U$ and nature chooses $\theta_k \in \Theta$. We do not assume that θ_k is chosen independent of u_k; we allow the possibility that nature is "malicious" in that is always chooses that worst θ_k for the chosen u_k.
3. The system transitions from state x_k to state $x_{k+1} = f(x_k, u_k, \theta_k)$.

[1] The notation presented here closely follows the notation used in class, and so is shown here primarily for completeness.

4. Nature chooses $\phi_k \in \Phi$.
5. The decision maker receives the observation $y_k = h(x_k, \phi_k)$.

The problem, then, is to find a strategy that minimizes L. For problems where the state is always known[2], a strategy is a function $\gamma : X \to U$. These problems are well-studied (see, for example, [Ber87]).

Now suppose that state is unknown. Rather than acting based on the current state, the best we can hope for is to act based on the history of actions and observations. Let

$$\eta_k = \{y_1, u_1, y_2, u_2, \ldots, u_{k-1}, y_k\}$$

represent the information state at stage k – the information available to the decision maker in choosing u_k. Let N_k denote the information space – the set of all possible information states at stage k. Our strategies will now have the form $\gamma : N \to U$. This form is, of course, problematic, since the dimension of N_k grows without bound as the number of stages k increases. Two alternate, more manageable representations are quite reasonable, depending on how we choose to model uncertainty:

1. *Probabilistic* – If we assume that θ and ϕ are chosen according to some known probability distribution, then we can describe what is known about the current state as a probability distribution $P(x_k|\eta_k)$ over states.

2. *Nondeterministic* – It may not be reasonable to describe the uncertainty in a problem probabilistically. In these cases, it may be more appropriate to use a set $S_k(\eta_k)$ of states possible, given the information state η_k.

In either case, the advantage is that the original imperfect information problem is rephrased as a perfect information state problem whose state space is the information space of the original problem. In the probabilistic case, we can derive a state-transition equation using Bayes' rule; the nondeterministic case is a parallel based on intersections and unions. In either case, we can (try to) use existing planning techniques to solve this new class of problems.

[2] Perhaps modeled in this framework by setting $y_k = x_k$.

3 Survey of "Classical" Papers

This section will summarize several papers that have made use of the notion of an information state (although almost always under a different name). The goal is not to give a detailed presentation of all of the results from these papers, but rather to give illustrations of research in this area and to begin to unify these approaches under the framework described above.

3.1 Erdmann and Mason: An Exploration of Sensorless Manipulation [EM88]

Erdmann and Mason describe an application that orients polygonal parts in a tray without sensors by a sequence of tilts. The goal is to orient the part in the sense that the number of possible orientations is minimized. Parts with no rotational symmetries will be uniquely oriented. The state space is the set of possible orientation-position combinations for the part. It is assumed that the part can start in any state. The set of actions is the the real interval $[0, 2\pi)$, representing the angle of tilt to apply.

The continuous space of states is discretized by representing only the information required to determine the next state, namely which edge of the polygon is in contact with the side of tray and whether or not the part is in a corner. The action space is also discretized, using an analysis of the physics of the system to divide the range into a finite set of equivalence classes.

Information states are represented nondeterministically. The information space is searched breadth-first, and nodes in the state transition graph are explicitly represented only as they are needed. The search finds the reachable information state of minimal cardinality. The breadth-first search strategy implicitly prefers shorter programs.

3.2 Goldberg and Mason: Bayesian Grasping [GM90]

This is another application: this time parts are oriented without sensors by squeezing them with a parallel-jaw gripper. The same breadth-first search over the information space as in [EM88] is used here. The new twist is that information states are represented in a Bayesian manner: there is initially a uniform distribution over the one-dimensional space of possible orientations; subsequent information states are represented as probability distributions over this space. The actions themselves, however, are assumed to be completely deterministic; that is, given an x and a u, the resulting state x' is uniquely determined. This assumption, combined with the piecewise-constant "squeeze function" that determines x', means that after the first squeeze, there are only finitely many possible orientations.

The stated goal is to find paths that minimize the *expected* number of operations required to orient the part. More precisely, assume that after executing a sequence of squeezes, the part is passed through a filter that accepts only correctly oriented parts and that parts that are rejected by the filter undergo the squeeze sequence again. If each squeeze and the verification step each take one time unit, then the "best" plan is the one that minimizes the expected time required to orient the part. Simply admitting the possibility of failure increases the margin for modeling error. This is a step up on [EM88], were it is simply assumed that their models are perfectly accurate.

3.3 Goldberg: Orienting Polygonal Parts Without Sensors [Gol93]

Here we have one additional take on the problem of sensorless orientation of parts. The noteworthy differences from [GM90] are:

- The notion of a part being oriented "up to symmetry" is formalized.
- The probabilistic flavor of [GM90] is missing. Instead, Goldberg's algorithm implicitly manipulates nondeterministic information states. Although it is not discussed, it would be trivial to reintroduce the expected-case analysis of plans from the earlier paper.
- An efficient (i.e. $O(n^2 \log n)$) algorithm for finding the shortest squeezing plan that will orient a part up to symmetry is given and proved correct.

The algorithm is also generalized to work with a slight variant, namely *push-grasp* actions, in which one of the gripper arms pushes the part into a stable position before the other arm makes contact. Goldberg shows that, because the "push" function has the same peicewise-constant structure as the squeeze function originally used, the algorithm can be used for this new problem essentially unchanged.

3.4 Erdmann: Randomization for Robot Tasks: Using Dynamic Programming in the Space of Knowledge States[Erd93]

Erdmann considers extending the idea of a "strategy" to allow randomized choices: allowing the decision maker to choose randomly from among several actions from certain information states, similar to the concept of a behavioral strategy from game theory.

The essence of Erdmann's approach follows: Consider the problem of using dynamic programming in the space of nondeterministic information states. Suppose that, after j iterations, it is known that for a given information state η_0, no plan can guarantee that the goal state will be reached in j steps. Suppose further that there exists some set of information states $\{\eta_1, \eta_2, \ldots, \eta_m\}$ for each of which, such a plan *is* available and such that $\eta_0 \subseteq \bigcup_{i=1}^{m} \eta_i$ Notice that if η_0 is the true information state, then true state x is in η_0 and therefore in at least one of $\eta_1, \ldots, \eta_k$. So if we randomly choose $i \leq m$ and execute the plan for η_i, there is some probabability that the plan will complete successfully.

In such a case, we can insert a SELECT operation into the dynamic programming table at stage j for information state η_0. When the algorithm completes, we'll have a randomized algorithm: upon encountering a SELECT in execution of the plan, we can choose an information state at random and continue as though that information state were known.

Erdmann presents two very contrived examples to prove his main result: that some tasks whose shortest guaranteed plans are exponentially long in the number of states have randomized plans that, in the expected sense, will complete in a polynomial number of steps.

4 Discussion

Finally, we shall discuss a number of outstanding issues.

4.1 The Problem of Dimensionality

In most of the work we've discussed, there's a tacit assumption made about the nature of the information space being searched. Consider the algorithm of [EM88]: Given a part whose convex hull has n sides, the information space has roughly 2^n information states to search. Generally, one would avoid a breadth-first search on a graph of exponential size. Here, Erdmann and Mason have quietly assumed that only a small part of the information space will need to be searched[3] before finding a plan or determining that no plan exists.

Erdmann's later work makes little improvement in this area. Although [Erd93] is mainly making a theoretical argument – that randomized strategies are, in some sense, more powerful than "pure" strategies – he too chooses not to address the dimensionality problem. The approach as he describes it crucially depends on being able to build a dynamic programming table whose rows are the 2^n information states. This is clearly impractical for all but the smallest state spaces.

[Gol93] is alone in discussing computational complexity, but, perhaps unsurprisingly, that work relies too heavily on properties of information space of the specific problem to be of general interest.

The root of the problem appears to be that the general approach – that of using information states to deal with uncertainty – results in search spaces that are significantly more complex that the original state spaces. In the nondeterministic case, the size of the information space is exponential in the size of the state space. In probabilistic models, the situation is considerably worse: a (discrete) space of n states corresponds to a (continuous) $n-1$-dimensional space of probability distributions.

There are two apparent possibilities for dealing with this difficulty. First, for some problems, we may be able to develop some rigorous assurance that some parts of the information space are unreachable. These regions can then be safely ignored. For example, it is not difficult to imagine a situation in which a nondeterministic information space is limited by simple propositions that must hold for every reachable information state. To be a bit more concrete, we might reasonably design a system in which there will never be any uncertainty in distinguishing between a certain pair of states. If this is the case, we can surely ignore the portion of the information space in which both of these states are possibilities. Certainly more complex examples are also feasible.

Second, there may be some effective way of sampling the information space, rather than searching it exhaustively. This approach has proven successful in other search problems in high-dimensional spaces, but detailed study of this approach has yet to be done.

[3] In fairness, we may alternatively suggest that they merely were not approaching the problem as algorithmists. Still, it is a fact that exponential-time algorithms do not perform well on large inputs, regardless of one's notion of what's important.

4.2 The Problem of Modeling

It seems self-evident that any practical system must be equipped with some model that describes how that system can interact with the world. Such a model could be programmed by hand, learned over time, or acquired by some combination of these two. In any case, the performance of any system is strongly dependent on the quality of its world models.

None of the papers we've discussed adequately addresses this question. Perhaps understandably, world models are often idealized, in particular for physical systems. In fact, Goldberg and Mason ([GM90]) reported disappointing experimental results because their physical models were inadequate: "We believe that the failures can be attributed to violations of the basic mechanical assumptions."

This area appears to be an excellent place to join learning and planning, particularly for applications to "agent"-type systems that are at least partly autonomous. Techniques for reinforcement learning represent progress already made toward this end.

An alternative to developing techniques for building more accurate models is to develop systems that are more robust to errors in the model. This approach is discussed to a degree in [GM90] and [Erd93].

4.3 Randomization

For the work presented here, only [Erd93] addresses the concept of a randomized strategy, but a number of questions are left unanswered. Among them:

- *Does randomization really add anything?* Are there nontrivial problems for which no pure strategy can guarantee reaching the goal, but for which some randomized strategy eventually will? Note that this is a much stronger proposition than that of [Erd93] – there it is only argued that randomization can give us *shorter* plans.
- *Does it help with problems that people actually want to solve?* Erdmann's examples are contrived specifically to prove his point; as such, it remains in question whether using randomization makes a significant difference for practical problems.

4.4 The Role of Observations

It is obvious that in some cases, sensors (observations) are required. But we've also seen several examples of problems that can be solved without any sensing at all. We are led to ask: What assumptions must be made about a problem to guarantee that is has a sensorless solution? Does the existence of a sensorless strategy imply that such a strategy can be found easily? If sensors are required, how powerful need they be?

These questions appear to be, for the most part, still open. Erdmann and Mason propose that sensorless and sensor-based planning ought to be integrated [EM88]:

> In general, one should view the uncertainty-reducing property of sensorless motions as just another mode of gathering information. Whether a sensorless strategy is superior, inferior, or complementary to sensor-based strategy depends on the nature of the task. For complex tasks, a sensor-based system would employ a sensorless strategy as one step in an overall plan.

Donald takes a bit of a different approach: He introduces the notion of an "information invariant": a rigorous way of stating that two sensor systems are equivalent [Don95].

Another important consideration is that, for a given sensor system, there may be several states that are indistinguishable, thereby partitioning that state space into "perceptual equivalence classes" [DJ91]. Not surprisingly, the sensor history becomes important here.

5 Conclusion

In this paper, we've studied the problem of planning with imperfect state information by formalizing the problem and reviewing some early progress on specific problems of this type. Many questions remain to be answered along the way to the development of a coherent theory for planning with information states.

References

[Ber87] D. P. Bertsekas. *Dynamic Programming: Deterministic and Stochastic Models.* Prentice-Hall, 1987.

[DJ91] B. R. Donald and J. Jennings. Sensor interpretation and task-directed planning using perceptual equivalence classes. pages 190–197, Sacramento, CA, April 1991.

[Don95] B. R. Donald. On information invariants in robotics. 72:217–304, 1995.

[EM88] M. A. Erdmann and M. T. Mason. An exploration of sensorless manipulation. *IEEE Journal of Robotics and Automation*, 4(4):369–379, August 1988.

[Erd93] M. Erdmann. Randomization for robot tasks: Using dynamic programming in the space of knowledge states. *Algorithmica*, 10:248–291, 1993.

[GM90] K. Y. Goldberg and M. T. Mason. Bayesian grasping. In *IEEE Int. Conf. Robot. & Autom.*, 1990.

[Gol93] K. Y. Goldberg. Orienting polygonal parts without sensors. *Algorithmica*, 10:201–225, 1993.