From: AAAI Technical Report WS-97-10. Compilation copyright © 1997, AAAI (www.aaai.org). All rights reserved.

Best Information Planning for Unknown, Uncertain, and Changing Domains

Anthony Stentz
Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
U. S. A.

Abstract

Planning operations for an autonomous agent is a much-studied problem. Most solutions require that the agent's domain be fully known at the time of planning and fixed during execution. For many domains, however, this is not possible. One approach to the problem is to construct an initial plan based on the best information a priori and revise the plan during execution as better information becomes available. In this paper, we show that this approach yields good average case performance compared to competing approaches for certain problems of interest. We also discuss the uses for this approach to planning and describe qualitative conditions for making it tractable.

1 Introduction

To solve many realistic problems, autonomous agents must operate in domains for which they have less than complete information or which change over time in unpredictable ways. In such cases, the types of domain information available to the agent might include the following:

- *Partially-Known:* accurate information is available for some part of the domain and no information is available for the rest. Example: a road map that shows just the primary roads but no secondary roads.
- *Outdated:* information was accurate at one time but is now inaccurate. This situation frequently occurs in dynamic environments. Example: bridge closure due to construction not indicated on map.
- *Probabilistic:* distributions covering a range of possibilities. Example: likelihood of traffic congestion along a particular road.
- *Heuristic:* rules of thumb that are frequently correct. Example: avoiding a particular intersection during rush hour is prudent.
- *Assumed:* default information used for lack of better information. Example: all roads are open and free flowing.
- *Estimated:* approximate information possibly calculated from measurements. Example: report from a helicopter on approximate traffic conditions.

Given the lack of complete information, we propose a *best information* approach to planning for autonomous agents. The foundation for this approach was described in earlier papers for single agent problems [Stentz, 94] [Stentz, 96] and later extended to multiple agents [Brumitt and Stentz, 96]. The idea is to construct an initial, optimal plan based on the best information available, including complete information when possible, and one or more of the above types when necessary. As the agent executes the plan, whenever better information becomes available, the agent recomputes an optimal plan from its current state to the goal. We call this approach best information planning, because at all times the agent is executing an optimal plan with respect to the best information available.

This approach is similar to assumptive planning [Nourbakhsh and Genesereth, 96], in which assumptions are made about the agent's domain and actions to avoid the combinatorics of conditional planning [Warren, 76]. We must emphasize, however, that best information planning does not preclude conditional planning, since conditional plans could be constructed which are later revised in response to better information on the possible courses of action.

Whereas inaccurate information can mislead an agent to construct a poor plan [Koenig and Smirnov, 97], we assert that the average case performance of such an approach is quite good for many problems of interest. In the experimental results section of the paper, we validate this assertion for one important problem: a mobile robot seeking a goal in a cluttered environment. But first, in the next section we address the difficult computational issue in best information planning: replanning for each new piece of information.

2 Conditions for Tractability

Best information planning can be difficult computationally. For example, STRIPS planning is PSPACE-hard [Bylander, 94], and in the worst case replanning is no better. But for many problems, replanning can be very efficient such that best information planning is tractable. In this section, we qualitatively describe conditions for making this approach possible.

2.1 Dynamic Programming

Dynamic programming is an approach to algorithm design that solves a problem by constructing increasingly larger subproblems until the given problem is solved [Aho et al, 83]. Often, the subproblems are "memory cached" for re-use to avoid unnecessary calculations. This approach holds promise for best information planning since the new plans produced during replanning are likely to be similar to the original plan. Some of the partial plans that were produced for the original plan can be re-used for the "repaired" plan. The same idea is employed in many incremental algorithms [Ramalingam and Reps, 92].

A* is an example of a search algorithm that uses dynamic programming [Nilsson, 80]. Optimal paths between initial and goal states are found by chaining together optimal paths between substates until the entire optimal sequence is constructed.

2.2 Bounding the Effects of Information Updates

For some planning problems, new information about the domain has a limited, local effect on agent operations. Determining bounds on these effects can serve to prune the search during replanning. For example, consider a mobile robot, equipped with a sonar sensor with a two-meter maximum range, planning paths across a room to a door. As the robot moves toward the door, its sensor detects an obstacle in its path, two meters away. Unless the obstacle completely obstructs a passageway, the robot need only replan those near-term driving operations that take it around the obstacle and piece the partial plan for this subproblem together with another partial plan that moves it the rest of the way to the goal.

Similarly, consider a robot that is following a planned path to the goal believed to be five meters in length (based on best information). If it sees an open door previously believed to be closed that is six meters away, it knows that this new information cannot possibly yield a better plan to the goal. Therefore, the effects of this new information need not be further investigated.

2.3 Backward Chaining

Plans can be constructed by forward-chaining operations from the initial state to the goal state, or by backward-chaining operations from the goal to the initial state [Charniak and McDermott, 85]. This latter approach is better suited for best information planning since from one replanning iteration to the next, the goal remains fixed and the initial state changes.

Therefore, if information updates occur local to the agent, the planned operations in the sequence near the goal state are less likely to be affected than those near the agent. Replanning can be accomplished by searching from affected operations in the middle of the sequence back to the agent's current state, re-using the search space constructed near the goal.

3 D* Search Algorithm

The D* algorithm (Dynamic A*) [Stentz, 94] is an efficient search algorithm for replanning that embodies all of the characteristics in the previous section. Like A*, D* uses a dynamic programming approach to construct plans by piecing together solutions to larger and larger subproblems. D* is capable of detecting when a new piece of information could affect the optimality of the current plan, and it propagates the effects of this new information only far enough in the search space to compute a new optimal plan. D* backward-chains operations from the goal to the agent's current state.

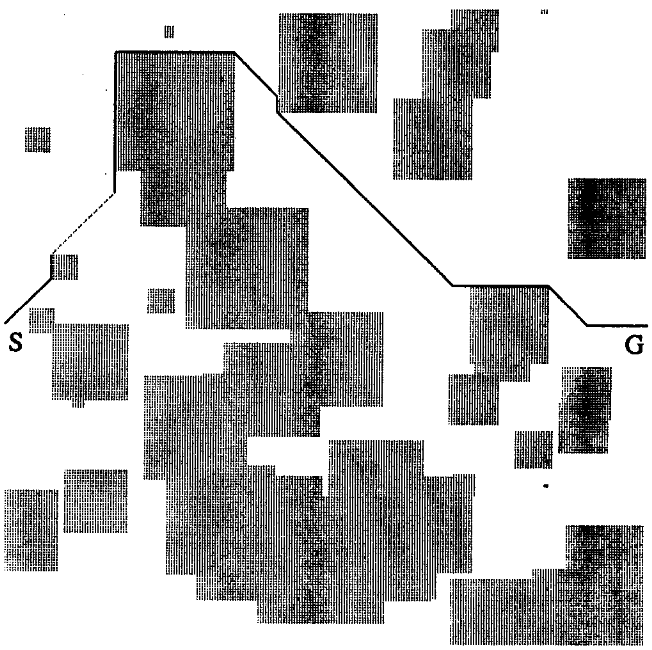
For domains with a large number of states and frequent information updates, D* is able to replan hundreds of times faster than brute-force replanning from scratch. The algorithm was tested both in simulation and using a real mobile robot driving in a cluttered and unknown environment. For each new obstacle detected by the robot's sensor, D* was capable of replanning the entire path to the goal in less than a second on a workstation.

The D* algorithm is described extensively in earlier papers [Stentz, 94] [Stentz, 95] [Stentz, 96], as is its validation using a real robot [Stentz and Hebert, 95].

4 Experimental Results

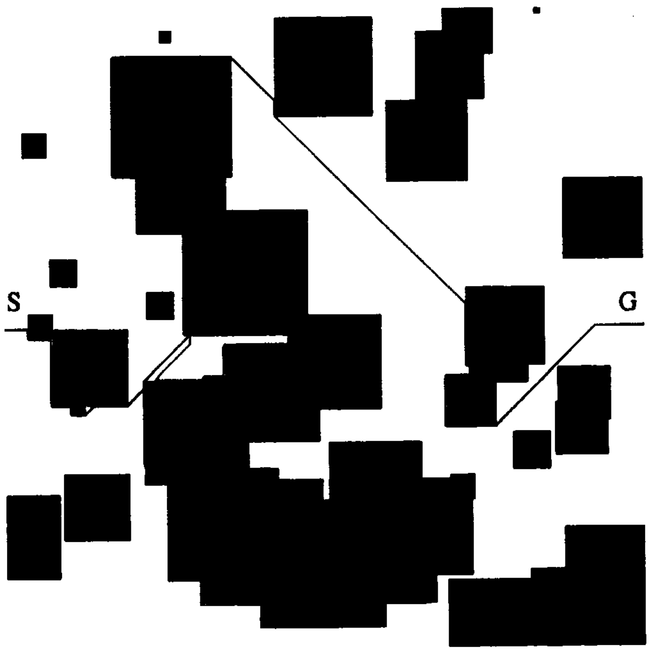
In order to validate the best information approach to planning, the following problem was analyzed. Consider a mobile robot equipped with a contact sensor that is instructed to drive from a start state S to a goal state G along the shortest possible path while avoiding a field of unknown obstacles. This problem is illustrated in Figure 1. For the experiments conducted, the robot's environment was a 500 x 500 grid of 8-connected cells. A total of 50 square obstacles were randomly placed in the environment, each of which ranged (randomly) from 1 to 50 cells in width. Note that obstacles could overlap. The optimal (i.e., shortest length) path from S to G is shown by a black polyline.

Figure 1: Optimal Path Given Complete Information



Two algorithms were compared: 1) the Best Information Planner (BIP), and 2) the "Bug1" algorithm [Lumelsky and Stepanov, 86]. Initially, BIP assumes that each cell does not contain an obstacle. It begins by planning a straight-line path from S to G and follows it. Whenever the robot attempts to move into an occupied cell, the contact sensor reports the presence of an obstacle. In such cases, BIP stores the obstacle cell in its map and replans a path to the goal using information about all obstacle cells detected so far. The traverse taken by BIP for the environment in Figure 1 is shown in Figure 2. The obstacles are drawn in black to indicate that they are unknown to BIP. As BIP discovers them with its contact sensor, it colors them grey. The path taken by BIP is 91% longer than the optimal path with complete information shown in Figure 1.

Figure 2: BIP Traverse Given No Initial Obstacle Information



Bug is a local algorithm that moves directly toward the goal through freespace and drives the full perimeter of each obstacle encountered to find the closest "leave" point before resuming free motion toward the goal. The traverse taken by Bug for the environment in Figure 1 is shown in Figure 3. The path taken by Bug is 405% longer than the optimal path with complete information. Note that free motion for both algorithms is generally not a single straight line since 8-connectivity is used.

A total of 100 environments were randomly produced and the results were averaged. Normalizing the width of each environment to "1", the average length of the optimal traverse through each environment given complete information was 1.236. The average length of the BIP traverse was 1.873 (52% longer than optimal). The average length of the Bug traverse was 5.578 (351% longer than optimal).

It is important to note that on average, BIP traversed 0.6% of the freespace and the Bug algorithm traversed 2%. Therefore, both algorithms are superior to an undirected approach that systematically explores the freespace until it happens across the goal [Pirzadeh and Snyder, 90].

To show the value in using the best information available, the above set of experiments was repeated for BIP where some of the obstacles where known to the planner initially. Table 1 shows the results for 11 cases ranging from 0% of the obstacles known (full freespace assumption) up to 100% known (complete information). As indicated, performance of the algorithm improved on average as the quality of the initial information improved.

The BIP traverse is shown in Figure 4 for the environment in Figure 1 where 50% of the obstacles where known a priori. The path taken by BIP is 47% longer than optimal with complete information. The unknown obstacles are shown in black and the known and "discovered" obstacles are shown in grey.

Figure 3: Bug Traverse of Same Environment

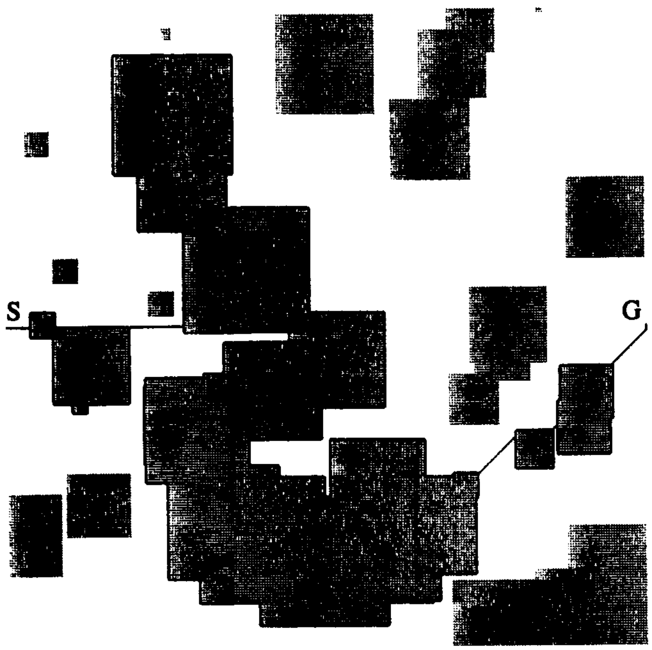


Figure 4: BIP Traverse with 50% of Obstacles Known

Note that the knowledge of 50% of the obstacles a priori led to a shorter traverse when compared to the case of no known information. This traverse, however, was still longer than the case for which all information was known a priori. All three traverses were shorter than Bug's.

Percentage of Obstacles Known	Path Length	Percentage Longer than Optimal Path
0%	1.873	52%
10%	1.855	50%
20%	1.824	48%
30%	1.820	47%
40%	1.775	44%
50%	1.711	38%
60%	1.602	30%
70%	1.488	20%
80%	1.396	13%
90%	1.314	6%
100%	1.236	0%

Table 1: BIP Results from Combining Known and Assumed Initial Information

5 Conclusions

In conclusion, we have empirically shown that the average case performance of best information planning for a mobile robot seeking a goal in an unknown and cluttered environment is superior to locally-directed and exploratory approaches for the environments modelled. We assert that this approach to planning can cover a wide range of problems for which less than complete information is available a priori, and we have qualitatively described the algorithm constructs needed to replan efficiently.

We are currently developing efficient replanning algorithms for STRIPS-like planners, moving goal problems, and sensor coverage.

Acknowledgments

This research was sponsored by DARPA, under contracts "Perception for Outdoor Navigation" (DACA76-89-C-0014), "Unmanned Ground Vehicle System" (DAAE07-90-C-R059), and "Technology Enhancements for Unmanned Ground Vehicles" (DAAE07-96-C-X075).

References

[Aho et al, 83] A. Aho, J. Hopcroft, and J. Ullman. *Data structures and algorithms.* Addison-Wesley Publishing Company, 1983.

[Brumitt and Stentz, 96] B. Brumitt and A. Stentz. Dynamic mission planning for multiple mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, April 1996.

[Bylander, 94] T. Bylander. The computational complexity of propositional STRIPS planning. *Artificial Intelligence*, 69, 1994.

[Charniak and McDermott, 85] E. Charniak and D. McDermott. *Introduction to artificial intelligence.* Addison-Wesley Publishing Company, 1985.

[Koenig and Smirnov, 97] S. Koenig and Y. Smirnov. Sensor-based planning with the freespace assumption. In *Proceedings of the IEEE International Conference on Robotics and Automation*, April 1997.

[Lumelsky and Stepanov, 86] V. J. Lumelsky and A. A. Stepanov. Dynamic path planning for a mobile automaton with limited information on the environment. *IEEE Transactions on Automatic Control*, AC-31(11), November 1986.

[Nilsson, 80] N. J. Nilsson. *Principles of artificial intelligence.*Tioga Publishing Company, 1980.

[Nourbakhsh and Genesereth, 96] I. Nourbakhsh and M. Genesereth. Assumptive planning and execution: a simple, working robot architecture. *Autonomous Robots*, 3(1), 1996.

[Pirzadeh and Snyder, 90] A. Pirzadeh and W. Snyder. A unified solution to coverage and search in explored and unexplored terrains using indirect control. In *Proceedings of the IEEE International Conference on Robotics and Automation*, May 1990.

[Ramalingam and Reps, 92] G. Ramalingam and T. Reps. An incremental algorithm for a generalization of the shortest-path problem. *University of Wisconsin Technical Report #1087*, May 1992.

[Stentz, 94] A. Stentz. Optimal and efficient path planning for partially-known environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, May 1994.

[Stentz, 95] A. Stentz. The focussed D* algorithm for real-time replanning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, August 1995.

[Stentz, 96] A. Stentz. Map-based strategies for robot navigation in unknown environments. In *Proceedings of the AAAI Spring Symposium on Planning with Incomplete Information for Robot Problems*, 1996.

[Stentz and Hebert, 95] A. Stentz and M. Hebert. A complete navigation system for goal acquisition in unknown environments. *Autonomous Robots*, 2(2), 1995.

[Warren, 76] D. Warren. Generating conditional plans and programs. In *Proceedings of the AISB Summer Conference*, 1976.