SHORTEST PATHS WITHOUT A MAP
(Extended Abstract)

Christos H. Papadimitriou\textsuperscript{1} and Mihalis Yannakakis\textsuperscript{2}

**Abstract:** We study several versions of the shortest-path problem when the map is not known in advance, but is specified dynamically. We are seeking dynamic decision rules that optimize the worst-case ratio of the distance covered to the length of the (statically) optimal path. We describe optimal decision rules for two cases: Layered graphs of bounded width, and two-dimensional scenes with unit square obstacles. The optimal rules turn out to be intuitive, common-sense heuristics. For slightly more general graphs and scenes, we show that no bounded ratio is possible. We also show that the computational problem of devising a strategy that achieves a given worst-case ratio to the optimum path in a graph is a universal two-person game, and thus PSPACE-complete, whereas optimizing the expected ratio is \#P-hard.

1. **Introduction**

Finding shortest paths is one of the most well-looked at problems in Computer Science and Operations Research (see, for example, [La], [PS], and the classical survey by Dreyfus [Dr]). More recently, several versions of the shortest-path problem in a geometric setting have been considered (see, for example, [Mi1], [Mi2], [CR], [Pa1], [MMP], [MP], and [Pa2] for a survey with over thirty references). Such variants are motivated by planning the motion of a robot in a scene sprinkled with obstacles, in which the metric is determined by the geometry of the obstacles.

It is sometimes natural to assume, both in the graph-theoretic and the geometric contexts, that the planner initially has incomplete information about the graph or scene, and such information is acquired in a dynamic manner, as the search for a good path evolves (e.g., by acquisition of sensory information in the geometric case, or by timed acquisition of the parameters, when the layer structure of a graph models time). What are appropriate search strategies in such a situation? And what are the right measures for evaluating such strategies?

Besides its inherent interest as a natural extension of a classical problem, this body of problems serves as an important paradigm in decision-making under incomplete information. Since it involves the design and evaluation of search heuristics, the techniques developed will add to the scarce rigorous methodological arsenal of Artificial Intelligence (see [Pea] for an extensive discussion in a more generic context). As it turns out, the heuristics developed in this paper (and shown to be optimal in two important cases) have the flavor of nontrivial, yet natural and common-sense approaches to the problem.

To acquaint the reader with the kind of problems studied, consider the three examples in Figures 1—3. In Figure 1 we have a layered graph of width two. Shortest-path

\textsuperscript{1}Department of Computer Science and Engineering, University of California at San Diego. Research supported by the National Science Foundation.

\textsuperscript{2}AT&T Bell Laboratories.
problems for such graphs model dynamic decisions, as layers from left to right may model stages of the decision-making process, that is, time. Such problems can be solved by specialized dynamic programming techniques (in fact, these were the archetypical applications of this method). Imagine, however, that the graph is given to us one stage at a time. In the beginning, we only know the part shown in Figure 1b. A rational searcher would probably try the lower choice. When the unfortunate information of the next stage (Figure 1c) is revealed, should the searcher persist on this path? (He/she always has the choice of following edges backwards, thus switching paths, but the distance traversed this way is counted in the total score.) Obviously, there is no way to guarantee that the searcher always finds the shortest path (shortest in the static sense, as if the graph were known beforehand). We assume that the goal of the searcher is to devise a strategy, so that the total distance traversed has the best possible ratio to the shortest path. This is a rather familiar notion of performance for dynamic problems, see, e.g., [ST].

For example, a reasonable strategy could be, informally, “persist on this path, unless there is a path on the other side that is less than half this one.” Thus, the searcher should persist in the second stage, but should switch in the next (Figure 1d). We show (Theorem 1) that this strategy is optimal, and achieves a worst-case ratio of distance travelled to shortest path equal to 9, the best possible. We generalize this to layered graphs with more than two nodes (“states”) per layer.

---

Figure 1: Searching Layered Graphs of Width Two