Foundations of Artificial Intelligence

Prof. Dr. B. Nebel, Prof. Dr. W. Burgard
B. Frank, A. Karwath, G. Röger
University of Freiburg
Department of Computer Science

Summer Term 2009

Exercise Sheet 3

Due: Tuesday, May 19, 2009

Exercise 3.1 (Path planning)

Consider the problem of finding the shortest path between two points on a plane that has convex polygonal obstacles (see Fig. 1). This is an idealization of the problem a robot has to solve to navigate its way around in a crowded environment.

(a) Suppose the state space consists of all positions (x,y) in the plane. How many states are there? How many paths are there to the goal?

(b) Explain briefly why the shortest path from one polygon vertex to any other in the scene must consist of (a) straight-line segments joining (b) vertices of the polygons. Define a good state space now. How large is this state space?

(c) In order to implement the search problem define (in text or pseudo code) a successor function that takes a vertex as input and returns the set of vertices that can be reached in a straight line from the given vertex.

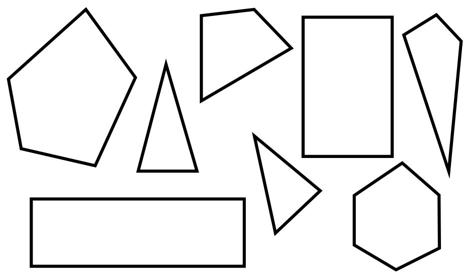

Figure 1: Robot navigation among polygons

Exercise 3.2 (Local search)

We will now examine *hill-climbing* in the same setting (planar robot navigation among polygonal obstacles).

(a) Explain how hill-climbing would work as a method of reaching a particular end point.

(b) Show how nonconvex obstacles can result in a local maximum for the hill-climber, using an example.

(c) Is it possible for it to get stuck with convex obstacles?

(d) Would simulated annealing always escape local maxima on this family of problems? Explain!

Exercise 3.3 (CSPs)

The $SEND + MORE = MONEY$ problem consists in finding distinct digits for the letters D, E, M, N, O, R, S, Y such that S and M are different from zero, i.e. no leading zeros, and the equation

$$SEND + MORE = MONEY$$

is satisfied.

(a) Explain in a nutshell, why it would be good to formulate the problem as a *constraint satisfaction problem*?

(b) Formulate the problem as a *constraint satisfaction problem*, i.e. what are the variables, what constraints do we have, etc.

(c) Find a solution using *forward checking* and *arc consistency*. Give the search tree.
(Hint: consider the letters in the following order: O, M, Y, E, N, D, R, S.)

Exercise 3.4 (Arc consistency)

AC-3 puts back on the queue *every* arc (X_k, X_i) whenever *any* value is deleted from the domain of X_i, even if each value of X_k is consistent with several remaining values of X_i. Suppose that, for every arc (X_k, X_i) and each value of X_k, we keep track of the number of remaining values of X_i that are consistent with this value of X_k. Explain how to update these numbers efficiently and hence show that arc consistency can be enforced in in total time $O(n^2d^2)$.

The exercise sheets may and should be handed in and be worked on in groups of three (3) students. Please fill the cover sheet[1] and attach it to your solution.

[1] `http://www.informatik.uni-freiburg.de/~ki/teaching/ss09/gki/coverSheet-english.pdf`