Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute for
Dynamic Systems and Control
IDSC
Institut für Dynamische Systeme
und Regelungstechnik

Dynamic Programming and Optimal Control

Fall 2021

Problem Set:
Deterministic Systems and the Shortest Path Problem

Notes:

- Problems marked with BERTSEKAS are taken from the book *Dynamic Programming and Optimal Control by Dimitri P. Bertsekas, Vol. I, 3rd edition, 2005, 558 pages, hardcover.*
- The solutions were derived by teaching assistants. Please report any error that you may find on Piazza forum.

Problem Set

Problem 1 (BERTSEKAS, p. 98, exercise 2.1)

Find a shortest path from each node to node 6 for the graph of Fig. 1 by using the DP algorithm.

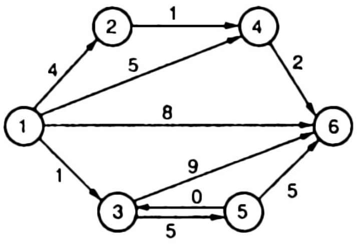

Figure 1: The arc lengths are shown next to the arcs.

Problem 2 (BERTSEKAS, p. 98, exercise 2.2)

Find a shortest path from node 1 to node 5 for the graph of Fig. 2 by using the label correcting method of Section 2.3.1 (see BERTSEKAS)[1].

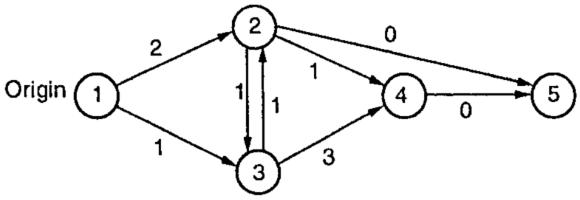

Figure 2: The arc lengths are shown next to the arcs.

Problem 3 (BERTSEKAS, p. 103, exercise 2.14)

Consider the shortest path problem of Section 2.3 (see BERTSEKAS)[2], except that the number of nodes in the graphs may be countably infinite (although the number of outgoing arcs from each node is finite). We assume that the length of each arc is a positive integer. Furthermore, there is at least one path from the origin node s to the destination node t. Consider the label correcting algorithm as stated and initialized in Section 2.3.1[1], except that UPPER[3] is initially set to some integer that is an upper bound to the shortest distance from s to t. Show that the algorithm will terminate in a finite number of steps with UPPER equal to the shortest distance from s to t. *Hint*: Show that there is a finite number of nodes whose shortest distance from s does not exceed the initial value of UPPER.

---

[1] or Section 8.1 from the class notes.

[2] or Section 7.1 from the class notes.

[3] The term UPPER here refers to d_{T} from the class notes. Both are used in literature and can be used interchangeably.

Problem 4 (Matlab Programming Exercise)

Consider a graph with N nodes. The transition costs between node i and j are given by $c_{i,j}$, where

$$c_{i,j} = \begin{cases} 0, & \text{if } i = j, \\ \infty, & \text{if } i \text{ and } j \text{ are not connected,} \\ c_{i,j} \in \mathbb{R}^+, & \text{otherwise.} \end{cases}$$

The objective is to find the shortest path from starting node `S` to terminal node `T` and its associated cost.

a) Implement a Matlab function that solves the shortest path problem using the Label Correcting algorithm.

b) A positive underestimate of the cost to move from node i to j is given by

$$c_{i,j} > |i - j| \, .$$

Modify your implementation of a) by taking the lower bound on the costs into account to obtain an $A*$ algorithm.

A set of Matlab files is provided on the class website (`ProblemSet3_Prob4.zip`). Please use them for solving the above problem.

`script.m`	Matlab script that can be used to load the cost matrix, execute the shortest path algorithms and display the results. By default it computes the shortest path from $s = 1$ to $t = 100$.
`lca.m`	Matlab function template to be used for your implementation of the Label Correcting algorithm for the shortest path problem.
`astar.m`	Matlab function template to be used for your implementation of the $A*$ algorithm for the shortest path problem.
`A.mat`	100×100 cost matrix A containing the costs to move from node i to node j: $A(i, j) = c_{i,j}$.