ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

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

151-0563-00 **Dynamic Programming and Optimal Control** (Fall 2009)

Programming Exercise #1 Topic: Deterministic Systems and the Shortest Path Problem

Issued: Oct 21, 2009 Due: Nov 04, 2009

Sebastian Trimpe (strimpe@ethz.ch) , 20. Oktober 2009

Find a shortest path using Dynamic Programming and the Label Correcting Method

The shortest path problem is to be solved for a finite graph with M nodes. The graph is given in the form of a square matrix P with dimension M by M. The (i, j)-th element of the matrix P denotes the cost of the transition from node i to the node j. All elements of P are nonnegative, finite integers.

The objective is to find a shortest path for any given starting node $s \in \{1, 2, \ldots, M\}$ and terminal node $t \in \{1, 2, \ldots, M\}$, i.e. to find the minimum cost of transition from s to t and a path from s to t with this cost. If there are several shortest paths, only one needs to be given. The shortest path problem is to be solved by

(a) converting the shortest path problem to a deterministic finite-state problem and solving it with the Dynamic Programming algorithm (see class textbook, p. 67/68), **and**

(b) applying the Label Correcting algorithm (see class textbook, Sec. 2.3.1) with Dijkstra's method for selecting a node from the candidate list at each step (see textbook p. 86).

The two methods are to be compared.

Provided Matlab Files

A set of Matlab files is provided on the class website. Please use them for solving the above problem.

`script.m`	Matlab script that can be used to load the problem data, execute the shortest path algorithms and display the results.
`sp_dpa.m`	Matlab function template to be used for your implementation of the Dynamic Programming algorithm for the shortest path problem.
`sp_lca.m`	Matlab function template to be used for your implementation of the Label Correcting algorithm for the shortest path problem.
`exampleProblemData.mat`	Problem data corresponding to the problem shown in Fig. 2.1.2 in the class textbook on page 68
`origProblemData.mat`	Problem matrix P specifying the problem to be solved
`generateProblemData.m`	Matlab script to generate random problem matrix P

Tasks

Implement your solutions for problem (a) and (b) in the files `sp_dpa.m` and `sp_lca.m`, respectively. Your code has to be able to be run with the Matlab script `script.m`. For your code development, you may find it helpful to consider the example `exampleProblemData.mat` and compare your solution with the solution in the book.

For evaluating your solution, we will test it on the given problem `origProblemData.mat` as well as on other random examples generated with `generateProblemData.m`.

Deliverables

Please hand in by e-mail

- your implementation of the DP algorithm `sp_dpa.m`;
- your implementation of the Label Correcting algorithm `sp_lca.m`;
- in a `pdf`-file, answers to the following questions

    1. What is the shortest path (cost and path) for the problem given in `origProblemData.mat` for starting node $s = 1$ and terminal node $t = 500$?
    2. Which algorithm takes longer to compute the shortest path? Why is this the case?

Please include all three files into one `zip`-file, which you name `DPOCEx1_Names.zip`, where *Names* is a list of the surnames of all students who have worked on the solution.[1]

Send your file to Sebastian (`strimpe@ethz.ch`) until the due date indicated above. We will send a confirmation e-mail upon receiving your e-mail. You are ultimately responsible that we receive your solution in time.

[1] Up to three students are allowed to work together on the problem. They will all receive the same grade.