Homework Assignment 2

Fall 2022

IE 532 Analysis of Network Data Due on Friday 10/14 (at 11:59 pm)

There are 4 questions in this assignment.

Please answer every question to the best of your knowledge and make sure to show your work for partial credit.

Make sure to cite your sources, if you decide to use more material to help you solve the exercises.

Remember that collaboration between students is allowed and encouraged, but please give your collaborators proper reference by letting me know who you worked with and what they contributed.

Only electronic submissions are accepted for this assignment.

You may either type your answers using LaTeX or other word processing software, or scan your handwritten answers and submit them. If the latter, please make sure the scanned copy is legible.

If submitting multiple files, create a folder and compress it (in either .zip, .rar, or .7z format, among others) before submitting.

Question 1: Scheduling and shortest paths

(a) When timetabling (scheduling shifts), we are interested in having at least one available resource at each of the operating hours. For example, in a bus company, we may be interested in having at least one available driver at each hour of the operations for a specific bus route/line. As an example, we offer Table 1, which shows all possible shifts for a bus company. Say, we need to ensure that at least one driver is on duty at each of the shifts (from 9am to 5pm) at the total lowest cost.

Design this scheduling problem as a shortest path problem and solve it with an appropriate algorithm. Explain why you chose the algorithm you did. The example is offered to help you check your approach; that said, please try to offer how to create a shortest path problem for a general case with multiple available shifts.

Hours	9am-1pm	9am-11am	noon-3pm	noon-5pm	2pm-5pm	1pm-4pm	4pm-5pm
Cost	30	18	21	38	20	22	9

Table 1: Duty hours for a bus company.

(b) A company needs to schedule the following tasks (shown in Table 2), numbered from 1 to 7, each of which has a duration and a list of prerequisite tasks that need to have been completed before they are started. For example, Task 4 will take 10 hours and needs Tasks 2 and 3 to be over before it is taken up.

Task	Duration	Prerequisites
Task 1	8 hrs	None
Task 2	4 hrs	Task 1
Task 3	3 hrs	Task 1
Task 4	10 hrs	Tasks 2, 3
Task 5	5 hrs	Task 2
Task 6	4 hrs	Task 3
Task 7	3 hrs	Tasks 4, 5, 6

Table 2: The tasks and their details.

Let t_i be a decision variable that signals the *starting time* of Task i, and assume that there is a fake "starting task", Task 0, with a duration of 0 hours, that starts at time 0, and a fake "ending task", Task 8, with a duration of 0 hours, and a starting time of t_8. Clearly, the smallest the time t_8 the earliest the ending of the overall project, so our problem would have the goal of min t_8. Formulate the problem of minimizing the duration of the whole project, while satisfying all the prerequisite requirements. Then, using the theory discussed during Lecture 10, formulate its dual. Can it be written as a shortest path problem? And, if so, recommend a method that solves the problem.

Question 2: Shortest path deeper cuts

(a) Consider the following statements and answer whether they are true or false. Justify your answer by providing a proof to the statement (if true) or by showing a counter example (when false).

> **Statement 1**. A network where all arc costs are different has a unique shortest path connecting two nodes.
>
> **Statement 2**. In a directed network, if we drop all directions (that is, every arc can be traversed in every direction no matter its original sense), the shortest path will not change.
>
> **Statement 3**. Assume we solved the shortest path problem but we underestimated each arc cost by $k > 0$ units (that is, instead of c_{ij} we should have used $\hat{c}_{ij} = c_{ij} + k$. Then, the solution to the two problems should be the same.
>
> **Statement 4**. Assume we solved the shortest path problem but we underestimated each arc cost by a factor of $k > 0$ (that is, instead of c_{ij} we should have used $\hat{c}_{ij} = c_{ij} \cdot k$). Then, the solution to the two problems should be the same.

(b) In some instances, we may be interested in finding **two shortest paths**. Why? Well, in cases where we are focused on the robustness of the proposed solution, we may want to focus on having a "Plan B". Specifically, assume that we are interested in solving the **2-disjointed shortest paths problem** for the same source and terminal nodes. We call two paths disjointed if they share no arcs (they are allowed to share nodes, though).

An algorithm to tackle this problem is the ingenious Suurballe's algorithm. A big-picture description is offered here:

1. Compute a shortest path tree from the source s to every other node (including the terminal t). Let d_{si} be the distance from s to i in the shortest path tree.
2. Modify all arc costs to be $c'_{ij} = c_{ij} + d_{si} - d_{sj}$. Note how this modification turns all arcs that are in the shortest path tree to have a cost of 0.
3. Construct a residual network by reversing all arcs in the current shortest path from s to t as well as removing all arcs that are opposite to the shortest path from s to t (in the reverse direction than the shortest path from s to t).
4. Find a shortest path from s to t in the residual network.
5. Discard any arc that appears in both shortest paths in opposite directions. The remaining arcs form two paths from s to t that share no arcs.

Implement this algorithm using networkx and then solve the network that appears in Lecture 6 Slide 21.

Question 3: Spanning and Steiner trees

(a) Prove or provide a counterexample to the following statement: *If all arcs of a network have distinct costs (that is, no two arc costs are equal), then the network has a unique minimum spanning tree.*

(b) Consider the problem of the *minimum cost spanning forest*, in which the goal is to identify k trees that span all nodes of a network in the minimum cost. Recommend an approach to solve this problem when k is given in advance. Then, use networkx to code this approach and solve the minimum cost spanning forest for $k = 2, 3$ in the network provided in the file *phylo.csv*.

(c) Write code in networkx that reads the illinois.csv zip codes, randomly selects k of them (you may try $k \in \{5, 10, 15\}$) and then returns the minimum cost **Steiner tree** connecting them. You may assume the distances between two zip codes are the Haversine distance based on their longitudes and latitudes. You may also assume that we may only connect two zip codes by an edge if the distance between them is below 40 units.

Question 4: Connectedness

(a) A network is said to be k-edge-connected if it remains connected after removing $k-1$ edges. Here, we ask you to focus on a specific pair of nodes s, t. Then, the network is said to be k-edge-connected as far as $s-t$ are concerned if s can send flow to t upon removal of $k-1$ edges. Formulate this problem as a maximum flow problem.

(b) A network is said to be k-node-connected if it remains connected after removing $k-1$ nodes. Here, we ask you to focus on a specific pair of nodes s, t. Then, the network is said to be k-node-connected as far as $s-t$ are concerned if s can send flow to t upon removal of $k-1$ nodes, other than the source and the terminal themselves. Formulate this problem as a maximum flow problem.