Midterm exam

Fall 2022

Analysis of Network Data Tuesday 10/25

Please do not turn the page until instructed to do so.

This exam has 6 questions.

You can pick 4 of the 6 questions to submit for a total of 100 points.

Every question you pick will be equally weighted.

If you pick more than 4 questions, 4 of them will be selected at random for grading.

There are 6 pages in the exam.

The exam is **a take-home exam**.

It is due by class time (2:00 pm Central time) on Tuesday 11/01.

You are **not** allowed to collaborate or communicate with others about the questions in this exam.

You are **perfectly allowed** to discuss and collaborate with me!

Answer every question to the best of your knowledge.

Make sure to show your work for **partial credit**.

Some questions (like, for example, Question 2) will ask you to use Python, Gurobi, networkx or the software of your choice. You do **not** have to solve this one if you prefer to avoid coding during the exam.

Good luck!

Question 1: Smaller proofs and algorithms

(a) A matching is a set of edges such that no two edges include the same node. An independent set is a set of nodes such that no two nodes are adjacent (connected by an edge). If M is a matching and $\mathcal{I}$ is an independent set, show that $|M| + |\mathcal{I}| \leq |V|$.

(b) Consider a weighted, directed graph $G(V, E, w)$, where all weights w on the edges are positive (i.e., $w_{ij} > 0$). Then, build two new weighted graphs $\hat{G}(V, E, \hat{w})$ and $\tilde{G}(V, E, \tilde{w})$ such that the weights are the negatives of the ones in G (i.e., $\hat{w}_{ij} = -w_{ij}$) and such that the weights are the reciprocals of the ones in G (i.e., $\tilde{w}_{ij} = \frac{1}{w_{ij}}$), respectively. Prove or disprove the following:

1. Every shortest path in $\hat{G}$ is a longest path in G.
2. Every shortest path in $\tilde{G}$ is a longest path in G.

(c) Given an edge-weighted directed graph with non-negative weights, design an algorithm for finding the shortest path from a given source s to a terminal t when you can change the cost of *any edge* to 0. What is the computational complexity of the algorithm?

(d) In class, we solved the minimum spanning tree with two optimal algorithms: Prim's and Kruskal's. Adapt either one to solve a related problem, where we still find a spanning tree, but one that minimizes the total deviations of every edge cost to the average edge cost. That is, the tree T minimizes the quantity $\sum_{(i,j) \in T} |w_{ij} - \overline{w}|$ where $\overline{w} = \sum_{(i,j) \in T} w_{ij} / (|V| - 1)$ (i.e., the average edge cost in the tree).

Question 2: Emergency crew routing

Last winter, a number of cities in the north states of the US faced threatening ice, snow, and cold weather conditions that caused provision of electricity to become intermittent. As a measure to prevent that and improve societal welfare, state and city authorities decided to locate repair crews and equipment in certain strategic areas in order to quickly assess, respond to, and repair such damages.

We represent this problem below on a graph. Figure 1 represents all candidate locations where crews can be stationed (in red), all arcs are given travel times, and all power system components (in blue) that can fail. If a power component is located in the middle of a street, then only half the traversal time needs to be "paid" before the crew reaches the power station. Assume that no streets are ever closed down, and hence the repair crews can always travel to address the issues arising in any power system component.

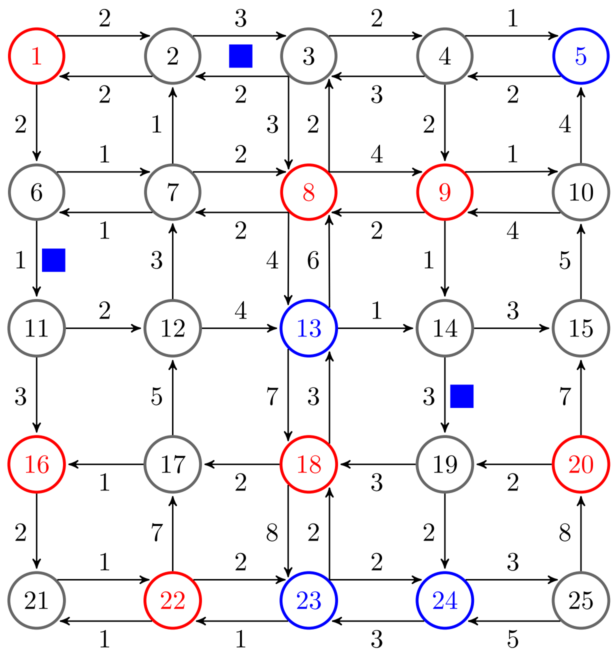


Figure 1: The network for Question 2. Note that the nodes in red are location candidates where your emergency crew can be stationed, while the nodes in blue are ones where damaged power station components can be found. If a blue node exists in the middle of a street, then you only need to "pay" half the arc time to get there from a starting node.

(a) Assume that your 2 crews are located in nodes 1 and 20. Formulate an optimization problem to evaluate the quality of the site selections, by deciding on which power systems get repaired by each crew, and the route that a crew would select to get to the damage site the fastest. Define your parameters, your variables, your constraints, and your objective function.

(b) Optimizing the same decisions as before, consider the case where the location crews are not known in advance, but instead are another decision variable. Formulate the problem under the assumption that the **2** crews are to be located on any node. Once more carefully define all your parameters, variables, constraints, and your objective function.

(c) Formulate and solve problem (b) in Gurobi.

Question 3: A crew scheduling problem

Consider the following airline crew scheduling problem. A crew can serve at most one flight at a time. When a crew serves a flight, it cannot serve another flight unless they arrive at their destination. As soon as the crew arrives in their destination airport, they can be scheduled to any flight starting from that destination or they can fly using an available flight to another airport from where they can get their next assignment.

We further assume that every flight has the following details:

$$(\text{Origin airport}, \text{Destination airport}, \text{Start time}, \text{Arrival time}) .$$

To help you produce instances we provide you with a code called "flights.ipynb": a screenshot with some indicative flights is provided in Figure 2. You may assume that times are all given in a universal clock setting; that is a number between 0 and 24.

Figure 2: A toy example for your crew scheduling problem.

#	Origin	Dest	Start	End
1	ATL	DCA	11.14	12.14
2	ATL	DFW	13.18	15.18
3	LGA	MCO	10.57	13.57
4	MCO	PHX	9.4	13.4
5	LAX	DFW	13.83	16.83
6	PHX	LGA	1.93	5.93
7	LGA	LAS	12.13	16.13
8	MCO	DEN	0.76	4.76
9	SFO	DCA	5.57	10.57
10	SFO	JFK	16.92	21.92
11	LAS	LGA	12.72	16.72
12	CLT	ORD	8.51	10.51
13	DFW	ATL	1.91	3.91
14	LAS	CLT	4.98	8.98
15	SFO	PHX	8.33	9.33
16	SFO	ORD	17.81	21.81

(a) Recommend an approach to solving the problem as a network flow problem when the goal is to *minimize the number of crews you need to use throughout the day.*

(b) Use the instance generator provided to solve a problem with $n = 50$ flights in networkx and report how many crews you need to accommodate all flights.

(c) Assume that some flights are bigger than others and need two or more crews. Based on the network flow problem you formulated in part (a), recommend an extension that you would implement to accommodate for that.

Question 4: Exploring a grid

Assume you have been given a $n \times n$ grid where some points have been marked, like the one in Figure 3. You begin your path from the top right corner, and need to get to the bottom left one. Before you do that, though, you want to visit as many of the marked areas as possible.

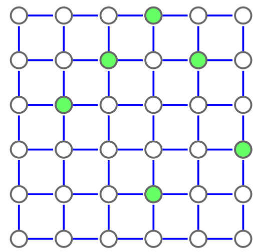


Figure 3: An example of the grid.

What is the catch? You can only move down or right. Hence, there is no way to visit all marked points with one path. Answer the following questions.

(a) Devise a polynomial time algorithm to pass through the maximum number of marked points. An example of what a path that covers three marked points would look like is shown in Figure 4. You may use networkx to produce more instances to test your algorithm on.

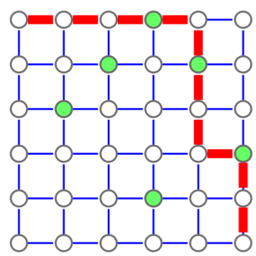


Figure 4: An example of visiting three marked nodes with one path.

(b) It will often be impossible to visit all marked points with one such path. Devise an algorithm to compute the smallest set of paths that pass through all marked points. What is the computational complexity of the algorithm?

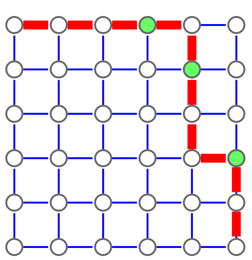


Figure 5: Path 1.

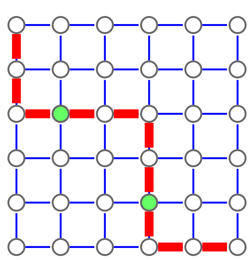


Figure 6: Path 2.

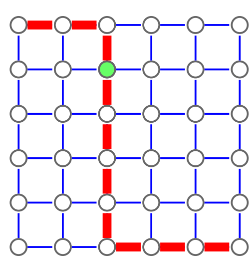
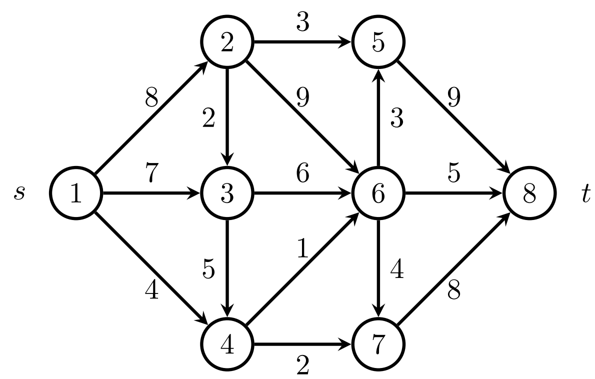


Figure 7: Path 3.

Question 5: Shortest paths and side constraints

(a) In the network of Figure 8, report the shortest path from node 1 to node 8 using the algorithm of your choice.

Figure 8: The network for Question 5 part (b). All arcs are given a cost.



(b) Now, consider the following problem, which is an extension to the original shortest path problem. Assume we seek a shortest path from node 1 to node 5, but with two "side objectives":

1. Each arc that is traversed provides you with a "prize": the total amount of prizes you collect has to be bigger than or equal to a threshold P;
2. Each arc that is traversed costs you a "cost": the total cost you can pay is restricted by your total budget C.

Your overall goal is to traverse the network in the smallest amount of time, such that you satisfy both side constraints. The numbers on the arcs in three networks below represent the *time* (Figure 9), *prize* (Figure 10), and *cost* (Figure 11) parameters of every arc.

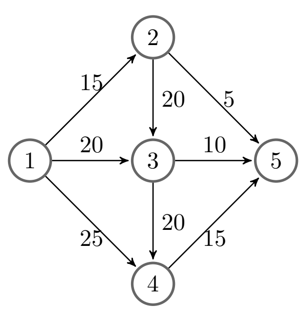


Figure 9: The times of traversal for each arc.

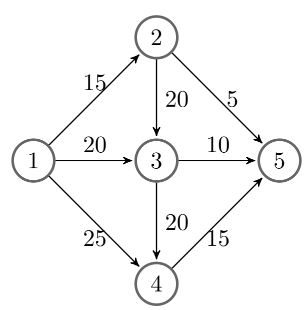


Figure 10: The "prizes" of traversal for each arc.

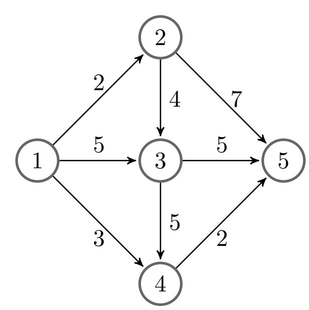


Figure 11: The costs of traversal for each arc.

Formulate the problem with both side constraints. For the problem of Figures 9,10), 11, obtain the optimal solution using Gurobi and networkx.

(c) Write the Lagrange dual of the problem, dualizing both side constraints. Using networkx to solve a shortest path problem iteratively, solve the Lagrangian relaxation of the problem and obtain the optimal solution.

Question 6: An evacuation planning problem

During an evacuation, people living in each of the locations in a network are to be routed towards one or many shelter (safe) locations. Assume you are provided with an undirected transportation network $G(V, E)$ (as an example, you can use the provided network emulating the main street arteries of West Virginia). Further assume that your shelter is randomly located somewhere in the state (i.e., simply select a random node s anywhere in the network to serve as your shelter). Answer the following questions.

(a) Consider the following problem: first, identify a minimum cost spanning tree in the whole network. This is ensured to lead everyone (eventually) to the shelter. Then, calculate the average time it takes every node to get to the shelter. Report this average time using networkx and the West Virginia network provided.

(b) The minimum spanning tree is a little too strict, in the sense that all nodes are equally important. Instead, assume that a set of nodes $S \subset V$ are the most important locations in the network (e.g., they have the biggest population; they are located in especially dangerous locations; they have more vulnerable populations; etc.). Solve a Steiner tree problem where you connect the set of nodes $S \cup s$ (i.e., the important locations and the shelter). Then, identify the shortest path from each of the non-important locations to any of the nodes in the Steiner tree. Again, report the average time it takes for each location to evacuation in the West Virginia network.

(c) Formulate the optimization problem of identifying an evacuation tree of minimum cost. This would be a tree, rooted at the shelter, where every node has **only one path to safety**. Formulate the problem and solve it using Gurobi and networkx. Report the average time it takes every node to get to safety for the West Virginia network provided.