Math 482: Topics Covered in Exam 3

Misha Lavrov

April 9, 2020

The third exam will primarily cover the graph theory section of the class; material between Wednesday, March 25^{th} and Friday, April 10^{th} inclusive. There may be a question on Fourier–Motzkin elimination from Friday, March 15^{th} (the one earlier topic that didn't make its way onto the second exam).

The exam will not be cumulative: I won't ask problems that are about earlier topics. But you'll need to use many of the earlier ideas to solve problems about the new topics.

Below I try to summarize the important things we've covered that will be on the exam. It's possible that I've missed a few topics, and it's hard to summarize three weeks of content in two pages anyway; if you're not sure about something, ask me.

1 Things you should know

Here are the definitions you should know:

- **Totally unimodular** matrices.
- **Graphs**, **bipartite graphs**, **matchings**, and **vertex covers**.
- **Networks** and **arc capacities**.
- Things you can try to find in a network: **feasible** s, t**-flow**, s, t**-cut**.
- The **value** of a flow and the **maximum flow**; the **capacity** of a cut and the **minimum cut**.
- The **residual graph** of a flow and things related to the residual graph: **residual capacity**, **augmenting paths**.
- The **excess flow** at a vertex, and networks with **supply/demand constraints** on vertices.
- For the min-cost flow problem: **spanning trees** and **connected** networks.

You should know and understand the following results:

- Farkas's lemma on the existence of solutions to $A\mathbf{x} \leq \mathbf{b}$.
- Linear programs with totally unimodular constraints have integer solutions.
- The size of a maximum matching in a bipartite graph equals the number of vertices in a minimum vertex cover.

- The max-flow min-cut theorem.
- The residual graph theorem: if there is no s, t-path in the residual graph, we can use it to get a minimum cut.

2 Things you should be able to do

- Use Fourier–Motzkin elimination to find a feasible solution to a system of linear inequalities.
- Write down a linear program for each of the problems we talked about: bipartite matching, maximum flow, minimum cut, and so on.

  These are generally very big and so I will not be asking you to solve these linear programs.
- Determine whether a matrix is totally unimodular.
- Use Ford–Fulkerson to find a maximum flow:
  - Draw the residual graph for a network and a flow in it.
  - Use the residual graph to find augmenting paths.
  - Use an augmenting path to increase the value of a flow.
  - Use the residual graph to find a minimum cut when there is no augmenting path.
- Reduce other problems to standard maximum flow problems: circulations with demands, circulations with lower bounds, finding matchings in bipartite graphs.
- Use the simplex method to find a min-cost flow:
  - Determine if a set of arcs forms a spanning tree.
  - For a set of arcs that does form a spanning tree, find a basic solution (a balanced flow), not necessarily feasible.
  - Given a spanning tree and a feasible flow on that tree, pivot to bring in an arc outside the tree and get a new spanning tree.
  - Compute the reduced cost of an arc outside the spanning tree.
  - Use the two-phase simplex method to find a spanning tree with a feasible flow.