General Information about the Final Exam

The final exam will cover some topics from earlier, those are listed below. We will also cover topics from Chapters 7 and 8, given below. The topics from the last section (after 8.3, including topics from Chapter 12 on nonlinear programming and Chapter 16 on neural nets) will NOT be on the in-class portion of the final.

The in-class portion of the final will be about an exam and a half in length, and we will have about 2 hours to complete it.

Topics from Chapters 7 and 8

1. We looked at the following applications of linear programming:
    - The Transportation Problem.
    - The Assignment Problem.
    - The Transshipment Problem.
    - The Shortest Path.
    - The Maximum Flow (and Minimum Cut)

2. While all of the applications are solvable through using a linear program, special techniques take advantage of the extra structure available in these applications, and can be orders of magnitude faster to solve on a computer. Here is a list of our specialized algorithms:
    - The Transportation Simplex Method (for a transportation tableau, and used to solve transshipment problems). This is a two step method:
        - Find a BFS
            - NW Corner Rule
            - Min Cost Rule
            - Vogel's Approximation Method (VAM)
        - MODI (Modified Distribution Method), or "The $u - v$ method". This tells us if a given BFS is optimal, and if it is not, gives the position of new basic variables.

      Note that we also did some sensitivity analysis: Changing the c_{ij} for a NBV, changing c_{ij} for a BV, then changing supply s_i and demand d_j by Δ in the case that x_{ij} is basic, then non-basic.
    - The Hungarian Method for solving the assignment problem. To initialize the method, subtract row minimums from each row, then column minimums from each column.

  - (For an $n \times n$ grid)
    If the smallest number of lines needed to cover all of the zeros is less than n, then we will not be able to assign zeros to each person, job pair. In that case (numbers are "covered" if a line goes through them), continue.
  - Find the minimum value of the uncovered numbers. Subtract that value from all uncovered numbers. Add that number to all numbers that are covered twice (by the intersection of two lines). Repeat.
- Dijkstra's Algorithm for finding the shortest path.
  List the nodes along the top. To the left, start a row with the first node (where the distance to itself is zero). Box that number and fill in the rest. If a node is not connected to the starting node, write ∞.
  Repeat these steps:
  - Find the minimum value in the row. That node will be used for the next row- Box that value.
  - Find the distance from the current node to all nodes that have not been boxed. If the total distance from the current node to a given node is less than the number given in the previous row, replace that value in the current row. If the distance is greater, leave that distance as written (copy it from the previous row).
  - Repeat until all nodes have boxed values.
- The Ford-Fulkerson Algorithm for finding the maximum flow.
  - Begin with a path of flow zero.
  - Find a path from the source to the sink (each edge must have remaining capacity). Recall that if an edge has a positive flow ($(f_e)c_e$ with $f_e > 0$) then we may move backwards along the path and subtract from f_e (up to f_e).
  - Find the minimum capacity along the path (we called that value b)
  - Add b to each edge in which we moved in a positive direction (subtract from edges where we moved in a negative direction).
  - Repeat until we have no further paths.

  The best way to verify if you have a maximum flow is to find a cut whose capacity is that value.

3. For 8.3, be able to compute: (i) the value of the flow, (ii) the capacity of a cut, (iii) the net flow across a cut. Know the max-flow, min-cut theorem.

Overall, be sure that you can convert each type of problem (transportation, assignment, transhipment, shortest path, maximum flow) to an appropriate linear program. For help/examples, see pg. 404, 393, 418, and 421.

For Ch 7, look at the review problems: 1-4, 8-10, 15, 16. Chapter 8 review is very short and focuses on material we didn't look at- look over the homework that was assigned. Beyond the conversion to LP, just be able to compute shortest path and maximum flow, for example, for the networks shown in fig 19-23, p. 430.

Topics from Exam 1

From the first review, be sure you understand the three theorems at the top of page 2 of the first exam review. In particular, be able to express any feasible point using the representation theorem, and be able to state the **fundamental theorem of linear programming**. On the final exam, we won't be focusing on Chapter 3 (setups), so you might focus your review on:

- Be able to solve an LP using the simplex method. Additionally, be able to state whether or not we have a final tableau, and be able to interpret what kind of solution we have (one solution, multiple solutions, no solution).
- Be able to write one point as a convex combination of other points (that's a skill needed for the representation theorem).

Example questions from the first review: 3, 4, 5, 13, 14, 18. From the chapter 4 review, you might look at questions 1, 2, 3, 7, 15, 17, 18.

Topics from Exam 2

From exam 2, be able to set up big-M. Be able to use the linear algebra notation we set up for the optimal tableau, and use that to do sensivity analysis.

Be able to compute the dual. Use the dual theorem to compute the solution to the dual. Be able to get the solution to the dual using complementary slackness.

I won't ask you to do the Dual Simplex algorithm for the in-class portion.

From Exam 2, a good overall summary is the last link to the material on our class website, where we answered a bunch of questions about sensitivity to a given problem. (The "group work" handout and solutions). In particular, from the exam review:

$$3(\text{set up}), 6, 7, 8, 12, 13$$

From the Chapter 6 review:

$$1, 3, 6, 9, 10, 18, 20, 33, 34$$