General Information about the Final Exam

The final exam will cover topics from Chapters 4, 6, 7, and 8. Between a third and a half of the exam will be exclusively from 8, then the rest from the earlier material. Actually, the material from 4 and 6 go together well, as does material from 7 and 8.

I'll allow you to have one page of notes (only one side please!). 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)
   - Critical Path Management (CPM)
   - Translating problems to MCNFP.
   - Minimum Spanning Tree

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.
   - **Network Simplex** I'm putting section 8.7 here because it is very comparable to the transportation simplex algorithm. We are given a network with $b(i)$ values (for supply/demand), and costs. We need to determine the flow that minimizes the cost using the network simplex method:
     - Given a feasible spanning tree (which represents a BFS), be able to compute the solution to the dual- to do so, set $y(1) = 0$, then moving along the BFS, compute $C_{ij} = y_i - y_j$.
     - See if the BFS is optimal by computing the "Row 0" values corresponding to the NBVs. We use the formula: $C_{ij} - y_i + y_j$. If all these values are positive, then the current BFS is optimal.
       * If a "Row 0" value for x_{ij} (that is, arc (i, j)) gives a negative values, take the edge with the most negative value. This is the incoming edge.
       * Bringing in the edge from the last step creates a cycle. If (i, j) is incoming, and an edge of the cycle moves in the same direction as (i, j), add θ, and otherwise subtract θ. There should be one value of θ that zeros out one of the edges of the loop. Use that value of θ and remove the edge with zero.

- Repeat from the first step.

- 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.

4. For CPM (8.4), we have a table of 6 values (Activity, Duration, ES, EF, LS, LF). Be able to compute ES and EF during a forward pass, then LS and LF in a backward pass across the network. Be able to compute the slack for each activity, then determine the path that is the critical path.

5. For the minimum spanning tree, we're creating the tree by adding the appropriate lowest cost edges one by one (an edge would not be used if its introduction would cause a loop).

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 Earlier

The focus of the exam is on chapters 4, 6, 7 and 8.

From Chapter 4, focus on being able to interpret "the final tableau" in terms of the solution to the given LP. It's a good idea to quickly review "Big-M" and the two-phase method (remember why we would use them).

In Chapter 6, review the linear algebra notation for the max LP (using A, B, etc). Be able to perform sensitivity analysis for a given problem (the "group work" handout from Ch 6 is good to review for this). Be able to compute the dual, and compute shadow prices. I won't ask you to perform the dual simplex method.

Chapters 7 and 8 are outlined above.

Be sure you review and understand the questions from the old exams.

Weighting of the Topics

About $1/3 - 1/2$ of the exam will be on Chapter 8, then the remaining will be over 4, 6 and 7.