EECS 868 Mathematical Optimization with Applications
Spring Semester 2019
Project **#3** Due **7 May 2019**

This project concerns the network shown on the back of this page. In that network, all links are bi-directional and each link label indicates the number of bi-directional telephone conversations that the link can carry at any given time.

I have no specific format that you should use for your report on this project; just make sure it is professional and complete, yet concise.

1. In this first part of the project, figure out how use the optimization toolbox of MATLAB, in particular, the "linprog" command, to find the maximum number of $bi-directional$ telephone conversations that can be active at any given time between any pair of nodes in a network in which links and their bi-directional capacities are specified (as here). This is similar to a conventional maximum flow problem, but with a bit of a twist: the links are bi-directional, *as are the conversations (paths).* You will need to carefully consider the implications of this "twist" when setting up the problem. You should strive to make your MATLAB script as easy to understand and work with as possible. File listings must be included in your report as appendices. I would strongly recommend that you test/debug your MATLAB script with a much smaller problem first.

   Use your MATLAB script to find the maximum number of bi-directional telephone conversations that can be active at any given time between node c and node f. Use the "optimset" command to force MATLAB to use the simplex algorithm. In addition to the maximum number of bi-directional telephone conversations, give the paths for this maximum number of conversations. These paths are probably best illustrated by copying the network figure and adding (by hand is fine) the number of conversations between nodes c and f that would be carried on each link for your solution. Also give the link or links that are the bottlenecks (that is, the link or links that would allow a greater number of conversations between c and f if their capacity were to be increased), and identify the "cut" that results from your solution (with reference to the min-cut, max-flow theorem). Finally, give the number of variables, equality constraints, and inequality constraints that you used, and the number of iterations that the solver required to find the solution.

   Try solving the same problem again with the "simplex" option turned off. Indicate whether the solution (not just the objective function value) is the same or not as well as any differences in number of iterations required, etc, but you need not go into great detail in discussing this second solution.

2. In this second part of the project, generalize your MATLAB script to find the maximum *total* number of bi-directional conversations that can be active at any given time between any given 3-node subset of the nodes. To explain what this means, let the 3 nodes be A, B, and C. Also let the number of conversations between A and B at a given time be x_{AB}, the number of conversations between B and C at the *same* time be x_{BC}, and the number of conversations between A and C at the *same* time be x_{AC}. Your goal is then to maximize $x_{AB} + x_{BC} + x_{AC}$.

   In your report, *explain* the approach that you used for this part and discuss how you would extend it to a subset of an arbitrary number of nodes. Include the capability to specify a minimum required number of conversations between each pair of nodes in the subset. You may use the simplex method or not; just specify whether or not you are using it. Again, take care to make your MATLAB script as easy to understand and work with as possible.

Then use your MATLAB script to find the maximum number of simultaneous bi-directional conversations that can be active at any given time between nodes b, d, and e (that is, the maximum *total* number of simultaneous bi-directional conversations between b and d, between b and e, and between d and e) for the following two cases:

— No minimum required number of conversations between nodes.

— A minimum of x conversations between each pair of nodes, with x ranging from 25 to 30.

Show resulting paths between *each* pair of nodes (either on separate diagrams or with different colors for the different pairs). Also, state how many variables, equality constraints, and inequality constraints were required for your approach, for this problem. Discuss your results.