15.094/SMA5223 Systems Optimization: Models and Computation

Assignment 6 (100 points)

Due May 7, 2004

1 Benders' Decomposition Problem (25 points)

You should download the files bender1.osc, eci_bender1_opl_master.mod, eci_bender1_opl_sub.mod, and eci_opl.dat from the assignment section. These files are an OPLScript based model of the powerplant planning problem discussed in the lecture.

- The file `bender1.osc` is the script file that runs the problem.
- The file `eci_bender1_opl_master.mod` is the model file for the master problem.
- The file `eci_bender1_opl_sub.mod` is the model file for the sub problem.
- The file `eci_opl.dat` is the data file for the problem.

a) The current model uses a five-point probability distribution for the gas turbine operating costs as well as for the coal plant operating costs. Modify the model and/or data file so that the stochastic program instead uses only the one-point average values for the gas turbine and coal plant operating costs. How does use of stochastic operating costs versus average operating costs impact the capacity allocation? Can you explain why?

b) The first column of Table 1 shows the probability distribution of the economic growth rates in the current version of the model. The second, third, fourth, and fifth columns portray ever-cruder approximations of this distribution. Compute the optimal capacity allocation for each one of these distributions. Using these allocations, compute the expected costs assuming that the actual growth rate distribution is the original one given in the first column of Table 1. Compare these costs to the optimal expected cost. What do you observe?

#1		#2		#3		#4		#5	
Growth	Prob.	Growth	Prob.	Growth	Prob.	Growth	Prob.	Growth	Prob.
−1%	20%	−0.6%	20%	−0.2%	30.0%	0.6%	50%	3%	100%
1%	20%	1.8%	25%	3%	35.0%	5.4%	50%		
3%	20%	4.2%	25%	6.2%	35.0%				
5%	20%	6.6%	30%						
7%	20%								

Table 1: Original and ever-cruder growth rate distributions.

2 Another Benders' Decomposition Problem (25 points)

The current version of the powerplant planning model uses the strategy of adding one constraint for every scenario in the model at each iteration of the Benders' decomposition method. This means that $k = 125$ new constraints are added to the model at each outer iteration of the method. As discussed in class, this strategy outperforms the alternative strategy of adding only $k = 1$ new constraint at each outer iteration of the model. In this question, you are asked to explore intermediate strategies that might improve computation time for solving the powerplant planning model using Benders' decomposition.

By adding your own control logic in the region indicated in the script file, `bender1.osc`, experiment with different strategies for limiting and/or controlling the number of new constraints added to the model at each outer iteration. You may also modify the model files or the data file if necessary. (One strategy that you might try would be to add a fixed number of constraints, say $k = 10, 20$, or 30, etc., constraints per outer iteration.)

3 SDP Truss Dynamics Problem (25 points)

Code Preparation:

a) unzip sdptruss.zip. This will create a folder called SDPT3-3.0. Work directly in this folder for the truss dynamics SDP exercises.

b) Create matlab mex files. To do this, start matlab from within the SDPT3-3.0 folder. At the matlab prompt, type `Installmex` You will see a bunch of warnings, but after about 1 minute the script will finish and SDPT3 will be ready to use.

c) Now you are ready to use SDPT3-3.0.

For the truss problem, we have provided wrap-around scripts to avoid having to call SDPT3 directly. For detailed information about these scripts, see the README-TRUSS file in the SDPT3-3.0/ directory. For the homework, the commands needed are explained right in the homework itself though, so you can probably avoid reading this.

a) After downloading and unzipping the truss and SDPT3 packages, you will see a file called bridge1.txt. Download the text file *bridge1.txt* for the bridge shown in Figure 1.

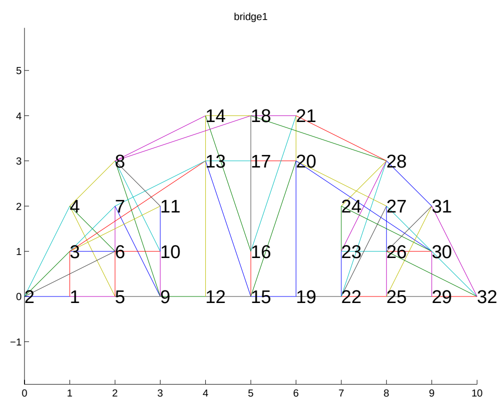

Figure 1: Geometry of a bridge1 with labeled nodes.

The format of the input in this file is "node1x node1y node2x node2y area". For example, the first line of the bridge1.txt file is "1 0 0 0 1.6", which means that there is a bar from (1,0) to (0,0) of area 1.6. (Note that the file contains an extra last line.) Run *truss.m* (just type `truss`) to prepare an input file for the SDPT3 program. Constrain the two end nodes of the bridge (nodes 2 and 32 as you can see in Figure 1) in both the x and y degrees of freedom (you will be prompted for this information).

b) What is the current lowest natural frequency of the bridge (this is output from the program truss.m)?

When prompted for a desired threshold frequency for the optimization, assume that design specifications require the lowest frequency to be 700 Hz. A file called *truss.sdpa* is generated which is the input to the following optimization problem.

c) Optimization. Now run *solvetruss.m* and SDPT3 will compute the optimal areas using the original areas in the file bridge1.txt as lower bounds.

d) What is the maximum increase in the areas among all of the bars of the truss? (Hint: the original areas are stored in the **areas** array (a Matlab row vector), and the new areas are stored in SDPT3 output array **y** (a Matlab column vector).

e) What is the resulting weight of the structure? (Hint: The objective value for the primal and dual are stored in a 2×1 array called **obj**. The negative sign is a result of writing our problem in SDPT3 format.)

f) Sensitivity to nearly infeasible and nearly feasible problems. As it turns out, the highest threshold frequency possible for this truss (above which the model is infeasible) is $\bar{\Omega} = 1012.7$ Hz. Try to create a plot of IPM iterations versus threshold frequency in the range $700 \leq \bar{\Omega} \leq 1012.7$. (Hint: To generate such a plot, rather than re-input the truss geometry every time, just run *reenterOmega.m* and after that rerun *solvetruss.m*.) As frequency is increased, the problem should become harder to solve. What types of messages does the code give indicating this? Nevertheless, the code does well. What is your overall observation from the plot you have created?

g) Now try to create a plot of IPM iterations versus threshold frequency in the range $1012.7 \leq \bar{\Omega} \leq 2000$. As frequency is increased, the software should take fewer iterations to declare the problem to be infeasible. What types of messages does the code give indicating this? What is your overall observation from the plot you have created?

4 Another Semidefinite Optimization Exercise (25 points)

This exercise is designed for you to explore solving ever-larger instances of a truss dynamics SDP problem. We have created a Matlab script for you that automatically:

- generates finer and finer meshed towers such as that shown in Figure 2 and Figure 3,

- determines each tower's natural frequency, and
- runs SDPT3 to find the minimum weight structure with a natural frequency that is 10 percent larger than the original.

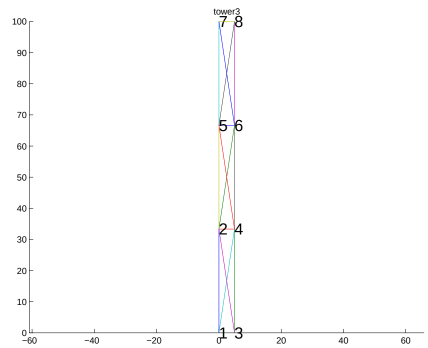

Figure 2: Geometry of a tower with 3 segments with labeled nodes.

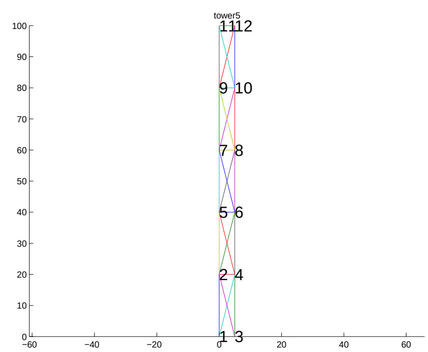

Figure 3: Geometry of a tower with 5 segments with labeled nodes.

a) Solve the SDP optimization problem with each of these finer and finer meshes for the towers by running the script *runtowers.m*.

   Note that generating the SDPA formatted input data will take progressively longer with each mesh refinement. You may want to reduce the number of mesh refinements if you find that your computer runs out of memory.

b) Plot IPM iterations and CPU times versus the number of nodes. What do you observe regarding computational efficiency?