IERG 6120 Project #1

due: 2:30pm, Monday, November 6, 2023

Instructions:

- You may discuss the problems with other students. However, everything in your submitted work must be your own (including code, etc.).
- You are allowed to use whatever book or paper that may help you complete this project. However, you must provide full references clearly explaining how you used the book and/or papers in your project.
- You are allowed to use any of the following programming languages C/C++/MATLAB/Python. If you want to use any other programming language, please contact the instructor at xjlin@ie.cuhk.edu.hk.
- If you use MATLAB, the configuration file for the following problems can be downloaded from the course-website. You may want to utilize the variables/matrices defined there as much as possible. If you use other programming languages, please create configureation files simlar to these.
- Please write a report that answers all the problems below. Your report can simply follow the order of the problems. For each problem, briefly describe how you design your program for the question. Then, report your results with figures and discussions. All the results, figures, comments, and conclusions should be clearly stated in the report.
- The source code of the project and the pdf file of your report should be compressed into one single ZIPPED file, and submitted via Brightspace (through the menu "Course Contents → Project 1") by the due date. Please also turn in a hard copy of the report in class.

(1) The first set of problems deal with simpler forms of convex optimization problems.

(a) Consider the following objective function

$$f(x_1, x_2) = x_1^2 + 3x_1x_2 + 9x_2^2 + 2x_1 - 5x_2$$

Use a 3D plotting tool (such as the mesh() function in Matlab) to draw a picture of the function, so that you can see whether the function is convex. Further, use the starting point $[x_1, x_2] = [0, 0]$. Pick a few random directions, and use a 2D plotting tool (such as the plot() function in Matlab) to draw a picture of the function restricting to a line through the above starting point. Repeat for 3 random directions of this line, so that you can see whether the function (restricted to the line) is convex.

(b) Use a gradient descent algorithm with step-size γ to find the minimum point of the above function.

Plot a figure showing how $x_1(k)$ and $x_2(k)$ evolve as the number of iterations k increases. Repeat this three times for three different step sizes: one that is small (so convergence is too slow), one that is large (so it does not converge), and one that is "just right" (so it converges fast). Plot the contour plot of the objective function (e.g., using the contour() function in Matlab). On top of the contour plot, show a trajectory of how the points $(x_1(k), x_2(k))$ move throughout the iterations. Repeat for the three different step-sizes chosen above.

(c) Now let us add the constraints. Consider the following convex optimization problem:

$$\begin{aligned} \min \quad & x_1^2 + 3x_1x_2 + 9x_2^2 + 2x_1 - 5x_2 \\ \text{subject to} \quad & 2x_1 + x_2 \geq 3 \\ & x_1 + 2x_2 \geq 3 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{aligned}$$

Write the code for the gradient projection algorithm that finds the optimal solution. Note that in general, finding the projection of a vector to the constraint set can by itself be a non-trivial optimization problem. However, since the above constraints are relatively simple, you can write a small subroutine to find the projection, by considering a few cases of where $[x_1, x_2]$ lies.

Plot a figure showing how x_1 and x_2 change as the number of iterations increases. Plot the trajectory on the contour plot too. Again, choose three different stepsizes to show how the algorithm converges very slowly, diverges, and converges "just right."

(d) Now, use a dual gradient projection algorithm to solve the same problem in part (c) . Recall that the projection of the dual variables should be a lot easier. However, since the two constraints $x_1 \geq 0$ and $x_2 \geq 0$ are pretty simple already, please do not relax them. Instead, associate the Lagrange multipliers only to the other two more-complex constraints.

Plot two figures showing how the primal variables and the dual variables, respectively, change as the number of iterations increases. Plot the trajectory of (x_1, x_2) on the contour plot too. Again, choose three different stepsizes to show how the algorithm converges very slowly, diverges, and converges "just right."

(e) For parts (b)-(d), it may be possible that the optimal solution that you find is incorrect. Hence, please write a small piece of code to verify that each of the solution is indeed the optimal solution. (Hint: use the first-order condition and/or the KKT condition, depending on the setting.)

(2) The second set of problems deal with a rate-control problem in a network. Consider the following network with 7 nodes, 12 links, and 7 flows:

The link capacities are $[1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2]$. The utility function of each flow i is of the form $U_i(x) = w_i \ln x$, where w_i is a weight parameter. The weights of the 7 flows

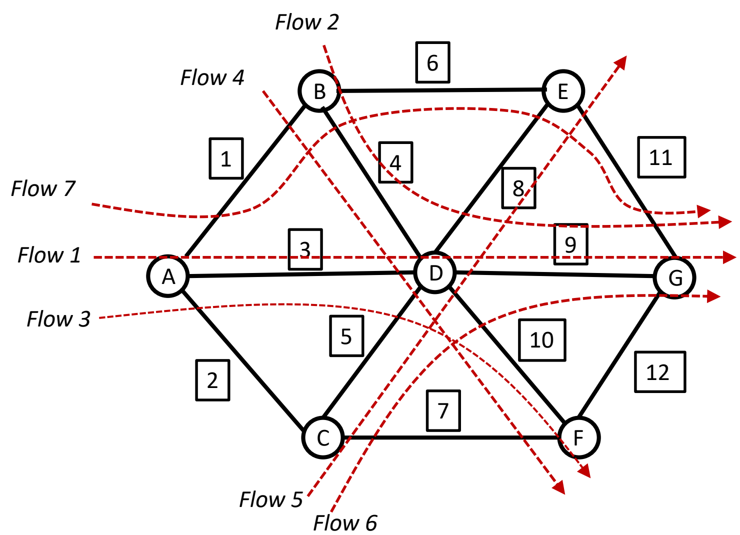

Figure 1: The network topology: 7 nodes, 12 links and 7 flows.

are $[1, 1, 1, 1, 2, 2, 2]$. Again, please refer to the configuration file to see how the topology information is encoded in MATLAB vectors/matrices.

(a) Formulate the rate control problem as a utility maximization problem, i.e., that maximizes the total utility of all flows, subject to the constraint that the total flow rate at each link is no greater than its capacity. Derive the dual gradient algorithm for solving it.

(b) Write the code to implement the dual gradient algorithm. Choose a suitable stepsize so that the convergence is relatively fast. Show two plots of how the flow rates and the dual varaibles, respectively, converge as a function of the number of iterations.

(c) Write the code to verify that the final flow rates are indeed optimal.