EE364b Homework 5

1. *Distributed method for bi-commodity network flow problem.* We consider a network (directed graph) with n arcs and p nodes, described by the incidence matrix $A \in \mathbf{R}^{p \times n}$, where

$$A_{ij} = \begin{cases} 1, & \text{if arc } j \text{ enters node } i \\ -1, & \text{if arc } j \text{ leaves node } i \\ 0, & \text{otherwise.} \end{cases}$$

Two commodities flow in the network. Commodity 1 has source vector $s \in \mathbf{R}^p$, and commodity 2 has source vector $t \in \mathbf{R}^p$, which satisfy $\mathbf{1}^T s = \mathbf{1}^T t = 0$. The flow of commodity 1 on arc i is denoted x_i, and the flow of commodity 2 on arc i is denoted y_i. Each of the flows must satisfy flow conservation, which can be expressed as $Ax + s = 0$ (for commodity 1), and $Ay + t = 0$ (for commodity 2).

Arc i has associated flow cost $\phi_i(x_i, y_i)$, where $\phi_i : \mathbf{R}^2 \to \mathbf{R}$ is convex. (We can impose constraints such as nonnegativity of the flows by restricting the domain of ϕ_i to $\mathbf{R}^2_+$.) One natural form for ϕ_i is a function only the total traffic on the arc, *i.e.*, $\phi(x_i, y_i) = f_i(x_i + y_i)$, where $f_i : \mathbf{R} \to \mathbf{R}$ is convex. In this form, however, ϕ is not strictly convex, which will complicate things. To avoid these complications, we will assume that ϕ_i is strictly convex.

The problem of choosing the minimum cost flows that satisfy flow conservation can be expressed as

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^n \phi_i(x_i, y_i) \\ \text{subject to} & Ax + s = 0, \quad Ay + t = 0, \end{array}$$

with variables $x,\ y \in \mathbf{R}^n$. This is the *bi-commodity network flow problem.*

(a) Propose a distributed solution to the bi-commodity flow problem using dual decomposition. Your solution can refer to the conjugate functions ϕ_i^*.

(b) Use your algorithm to solve the particular problem instance with

$$\phi_i(x_i, y_i) = (x_i + y_i)^2 + \epsilon(x_i^2 + y_i^2), \qquad \mathbf{dom}\, \phi_i = \mathbf{R}^2_+,$$

with $\epsilon = 0.1$. The other data for this problem can be found in `bicommodity_data.m`. To check that your method works, compute the optimal value $p^\star$, using `cvx`.

For the subgradient updates use a constant stepsize of 0.1. Run the algorithm for 200 iterations and plot the dual lower bound versus iteration. With a logarithmic vertical axis, plot the norms of the residuals for each of the two flow conservation equations, versus iteration number, on the same plot.

Hint: We have posted a function `[x,y] = quad2_min(eps,alpha,beta)`, which computes

$$(x^*, y^*) = \underset{x\geq 0, y\geq 0}{\operatorname{argmin}} \left((x+y)^2 + \epsilon(x^2+y^2) + \alpha x + \beta y\right)$$

analytically. You might find this function useful.

2. *Minimum eigenvalue via convex-concave procedure.* The (nonconvex) problem

$$\begin{array}{ll} \text{minimize} & x^T P x \\ \text{subject to} & \|x\|_2^2 \geq 1, \end{array}$$

with $P \in \mathbf{S}_+^{n\times n}$, has optimal value $\lambda_{\min}(P)$; x is optimal if and only if it is an eigenvector of P associated with $\lambda_{\min}(P)$. Explain how to use the convex-concave procedure to (try to) solve this problem.

Generate and (possibly) solve a few instances of this problem using the convex-concave procedure, starting from a few (nonzero) initial points. Compare the values found by the convex-concave procedure with the optimal value.

3. *Ellipsoid method for an SDP.* We consider the SDP

$$\begin{array}{ll} \text{maximize} & \mathbf{1}^T x \\ \text{subject to} & x_i \succeq 0, \quad \Sigma - \mathbf{diag}(x) \succeq 0, \end{array}$$

with variable $x \in \mathbf{R}^n$ and data $\Sigma \in \mathbf{S}_{++}^n$. The first inequality is a vector (componentwise) inequality, and the second inequality is a matrix inequality. (This specific SDP arises in several applications.)

Explain how to use the ellipsoid method to solve this problem. Describe your choice of initial ellipsoid and how you determine a subgradient for the objective (expressed as $-\mathbf{1}^T x$, which is to be minimized) or constraint functions (expressed as $\max_i(-x_i) \leq 0$ and $\lambda_{\max}(\mathbf{diag}(x) - \Sigma) \leq 0$). You can describe a basic ellipsoid method; you do not need to use a deep-cut method, or work in the epigraph.

Try out your ellipsoid method on some randomly generated data, with $n \leq 20$. Use a stopping criterion that guarantees 1% accuracy. Compare the result to the solution found using `cvx`. Plot the upper and lower bounds from the ellipsoid method, versus iteration number.