Homework Assignment #1

Due date: 2/14/2012, in class.

1. Turn in your solution to Quiz #1, which is now posted on the class web site.
2. A two-dimensional skier must slalom down a slope by going through n parallel gates of known position (x_i, y_i), and of width c_i, $i = 1, \ldots, n$. The initial position $(x_0, 0)$ is given, as well as the final one, $(x_{n+1}, 0)$. (Here the x-axis represents the direction down the slope, from left to right.)

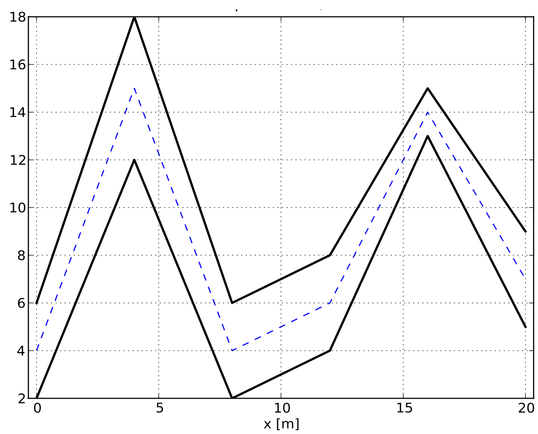

Figure 1: *Slalom problem with $n = 5$ obstacles. "Uphill" (resp. "downhill") is on the left (resp. right) side. Middle path in blue. Initial and final positions are not shown.*

(a) Find the path that minimizes the total length of the path. Your answer should come in the form of an optimization problem. If it falls within a problem class (LP, QP, SOCP, SDP) seen in lecture, state so.

(b) Try solving it via CVX, with the following problem data:

i	x_i	y_i	c_i
0	0	4	2
1	4	5	3
2	8	4	2
3	12	6	2
4	16	5	1
5	20	7	2

Include your code printout in the homework.

3. *Multi-period investments.* We consider an investment decision problem where an investor needs to decide how much money to allocate in a single financial instrument (asset) at each stage of a decision horizon composed of $n > 1$ periods. Let p_i denote the market value or price of this asset at time i and let u_i be the "bet" made by the investor (i.e. the amount of money invested in the instrument) at time i, for $i = 1, \ldots, n$, u_0 being the given initial position. The outcome of the bet is dictated by the underlying price fluctuation, which is random: if u_i is invested at i, the value of this investment at time $i+1$ is $\frac{p_{i+1}}{p_i} u_i$, hence the net return of this period is $y_i u_i$ where $y_i = (p_{i+1} - p_i)/p_i$ denotes the random return in period i.

   One basic goal of the investor is clearly to maximize the cumulative returns, i.e. $\sum_{i=1}^{n} y_i u_i$. However, a sensible investment the problem is really a multi-criterion one, since the investor also needs to take into some account the cost of transactions and the exposure to risk. To this end, we have following three terms:

$$
\begin{aligned}
\mathcal{P} &= \sum_{i=1}^{n+1} y_i u_i && : \text{cumulative profit} \\
\mathcal{R} &= \sum_{i=1}^{n+1} \sigma_i^2 u_i^2 && : \text{risk term} \\
\mathcal{C} &= \sum_{i=1}^{n+1} c|u_i - u_{i-1}| && : \text{approximate transaction costs term}
\end{aligned}
$$

   where $u_{n+1} = 0$, σ_i is the given variance of y_i and $c > 0$ is the unit transaction cost.

   The optimization problem is thus

$$
\phi(u_0, \mathbf{y}, \sigma, c) := \max_{u_{n+1}=0} \sum_{i=1}^{n+1} y_i u_i - \sigma_i^2 u_i^2 - c|u_i - u_{i-1}| \tag{1}
$$

   (a) Find the dual of the formulated problem in (1).

   (b) What is the sensitivity issue of ϕ with respect to u_0? In other words, provide a tight upper bound on $|\phi(u_0 + \epsilon) - \phi(u_0)|$ for arbitrary $\epsilon > 0$ (assuming that y, σ, c are fixed).

   (c) Implement the code to solve the primal and the dual problem in CVX. *Hint:* you could use the dual variables mode to allow to compute dual variables without coding both of them.

4. *Boolean least-squares.* Consider the following problem known as *Boolean Least Squares*:

$$
\phi = \min ||Ax - b||_2^2 \quad : x_i \in \{-1, 1\}, \quad i = 1, \ldots, n
$$

   in the variable $x \in \mathbf{R}^n$, where $A \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^m$ are given. This is a basic problem in digital communications. A brute force solution is to check all 2^n possible values of x which is usually impractical.

(a) Show that the problem is equivalent to

$$\phi = \min_{X,x} \mathbf{Tr}(A^T AX) - 2b^T Ax + b^T b \;:\; X = xx^T, \;\; X_{ii} = 1, \;\; i = 1, \ldots, n,$$

in the variables $X = X^T \in \mathbf{R}^{n\times n}$ and $x \in \mathbf{R}^n$.

(b) The constraint $X = xx^T$, i.e., the set of rank-1 matrices is not convex, therefore the problem is still hard. However an efficient approximation can be obtained by relaxing this constraint to $X \succeq xx^T$ which is usually referred as a *semi-definite relaxation* (SDR):

$$\phi \geq \phi_{\text{SDP}} := \min_{X} \mathbf{Tr}(A^T AX) - 2b^T Ax + b^T b \;:\; \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succeq 0, \;\; X_{ii} = 1, \;\; i = 1, \ldots, n.$$

The relaxation produces a lower-bound to the original problem. Generate random data $A \in \mathbf{R}^{10\times 10}$ and $b \in \mathbf{R}^{10}$ and solve the above problem in CVX. Now an approximate solution to the original problem can be obtained by rounding the solution: $x_{sdp} = \text{sign}\,(x^*)$, where x^* is the optimal solution of the SDR in (b). Calculate the original objective value $||Ax_{sdp} - b||_2^2$, which serves as a benchmark of our approximation.

(c) Another approximation method is to relax the non-convex constraints $x_i \in \{-1, 1\}$ to convex interval constraints $-1 \leq x_i \leq 1$ for all i, which can be written $\|x\|_\infty \leq 1$. Therefore a different lower bound is given by:

$$\phi \geq \phi_{\text{int}} := \min ||Ax - b||_2^2 \;:\; \|x\|_\infty \leq 1.$$

Using the data you generated in part 4b, solve the above problem in CVX. Round the solution by $x_{int} = \text{sign}\,(x^*)$ and compare the original objective value $||Ax_{int} - b||_2^2$.

(d) Which one of ϕ_{SDP} and ϕ_{int} produces the closest approximation to ϕ? Justify carefully your answer.

(e) Now in 100 independent realizations of normally distributed data, $A \in \mathbf{R}^{10\times 10}$ (independent entries with mean zero) and $b \in \mathbf{R}^{10}$ (independent entries with mean 1), plot and compare the histograms of $||Ax_{sdp} - b||_2^2$ of part 4b, $||Ax_{int} - b||_2^2$ of part 4c, and the objective corresponding to a naïve method $||Ax_{LS} - b||_2^2$, where $x_{LS} = \text{sign}\left((A^T A)^{-1} A^T b\right)$ is the rounded ordinary Least Squares solution. Briefly discuss accuracy and computation time (in seconds) of the three methods.

(f) Suppose that for some instance the optimal solution of part 4c turns out to be in the original non-convex constraint set $\{x : x_i \in \{-1, 1\}, \;\; i = 1, \ldots, n\}$. What can you say about the optimal solution of the original problem and 4c?

5. *Reformulating constaints in* `cvx`. Each of the following `cvx` code fragments describes a convex constraint on the scalar variables `x`, `y`, and `z`, but violates the `cvx` rule set, and so is invalid. Briefly explain why each fragment is invalid. Then, rewrite each one in an equivalent form that conforms to the `cvx` rule set. In your reformulations, you

can use linear equality and inequality constraints, and inequalities constructed using `cvx` functions.

You can also introduce additional variables, or use linear matrix inequalities. Be sure to explain (briefly) why your reformulation is equivalent to the original constraint, if it is not obvious.

Check your reformulations by creating a small problem that includes these constraints, and solving it using `cvx`. Your test problem doesn't have to be feasible; it's enough to verify that `cvx` processes your constraints without error.

Remark. This *looks* like a problem about 'how to use `cvx` software', or 'tricks for using `cvx`'. But it really checks whether you understand the various composition rules, convex analysis, and constraint reformulation rules.

(a) `norm( [ x + 2*y , x - y ] ) == 0`

(b) `square( square( x + y ) ) <= x - y`

(c) `1/x + 1/y <= 1; x >= 0; y >= 0`

(d) `norm([ max( x , 1 ) , max( y , 2 ) ]) <= 3*x + y`

(e) `x*y >= 1; x >= 0; y >= 0`

(f) `( x + y )^2 / sqrt( y ) <= x - y + 5`

(g) `x^3 + y^3 <= 1; x>=0; y>=0`

(h) `x+z <= 1+sqrt(x*y-z^2); x>=0; y>=0`

(i) `[x y z; y x+2*z x; z x 3*z-x^2] == semidefinite(3)`