**6.251/15.081 Homework 6**
**WEEK 10**

**A.** Computational assignment (next page). For this part, feel free to work in groups of two or three, and hand in identical copies of your writeup.
For those in need of MATLAB®* help, here is a good, short primer.

**B.** Problems 5.14, 6.3, 6.4, 6.6

# Computational Assignment*

**Exercise 1.** *The affine scaling algorithm.*

**(a)** Implement in MATLAB a function `affscale.m` which performs the affine scaling algorithm (as per p.400 in [1]). The prototype for the function is

```
[xstar,ubflag,fconv,xconv] = affscale(A,b,c,x0,tol,beta),
```

where the inputs `A`, `b`, and `c` are the usual (standard form) LP parameters, `x0` is an initial feasible solution, `tol` is the optimality tolerance, and `beta` is the step-length scale factor. The outputs are an optimal solution (if found) `xstar`, as well as the flag `ubflag`, which is `1` if the problem is detected to be unbounded and `0` otherwise. The last outputs are `fconv`, which is a vector storing the objective value of the feasible solution $\boldsymbol{x}^k$ at each iteration, and `xconv`, which is a matrix with column $k$ being $\boldsymbol{x}^k$. Print out and submit a copy of your code.

**(b)** Solve the LP $\min\{-x_1 - 3x_2 \ : \ x_1 + x_2 \leq 1, \ x_1, x_2 \geq 0\}$. Obviously, you must convert to standard form first.

**(c)** For the LP in **(b)**, print out a convergence plot (from your vector `fconv`) of the objective value for `beta=0.01`, `beta=0.50`, and `beta=0.99`. Use the same `x0` and `tol` in each case and provide a qualitative explanation for any observed differences.

**(d)** For the LP in **(b)**, solve it twice with your code, once for `x0=1/3*[1;1;1]` and once for `x0=[0.99;0.005;.005]` (last component is the slack variable). For both runs, make a scatter plot of the first two components of all of the $\boldsymbol{x}^k$ solutions, which are stored in `xconv`. Use the same `beta` and `tol` in both cases and provide a qualitative explanation for any observed differences. [*Hint:* To make a scatter plot of two vectors `x` and `y`, use the command `plot(x,y,'+')`.]

*For those students unfamiliar with MATLAB, please see the short primer linked to on the course website.

**Exercise 2.** *An example LP.*

**(a)** Solve exercise 5.6**(b)** in [1] using your algorithm from exercise 1. Print out a convergence plot of the objective value and make sure to note what values of the various parameters you used, as well as the optimal solution.

**(b)** Solve exercises 5.6**(c)**-**(g)** in [1] using a standard LP solver. The course staff highly recommends the function `linprog.m` in MATLAB, but you are free to choose any available tools.

# References

[1] Bertsimas, D.; Tsitsiklis, J.N. *Introduction to Linear Optimization.* Athena Scientific, 1997.