Lecture 12

Recall the following primal dual standard forms of linear programming,

$$\text{minimize} \quad \langle \boldsymbol{c}, \boldsymbol{x} \rangle \qquad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \geq \boldsymbol{0}. \tag{P}$$

for a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$ and $\boldsymbol{c} \in \mathbb{R}^n$, and

$$\text{maximize} \quad \langle \boldsymbol{b}, \boldsymbol{y} \rangle \qquad \text{subject to} \quad \boldsymbol{A}^\top \boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c},\ \boldsymbol{s} \geq \boldsymbol{0}. \tag{D}$$

For what follows, we assume $m \leq n$, as otherwise (D) is unbounded and (P) empty.

Based on (P) and (D), we get the optimality conditions (see Lecture 11)

$$\begin{aligned} \boldsymbol{A}^\top \boldsymbol{y} + \boldsymbol{s} - \boldsymbol{c} &= \boldsymbol{0} \\ \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} &= \boldsymbol{0} \\ \boldsymbol{X}\boldsymbol{S}\boldsymbol{e} &= \boldsymbol{0} \\ \boldsymbol{x} &\geq \boldsymbol{0} \\ \boldsymbol{s} &\geq \boldsymbol{0}, \end{aligned} \tag{O}$$

where $\boldsymbol{X}$ is the diagonal matrix with the x_i in the diagonal, $\boldsymbol{S}$ the diagonal matrix with the s_i on the diagonal, and $\boldsymbol{e}$ the vector with all ones ($\boldsymbol{X}\boldsymbol{S}\boldsymbol{e}$ is just a compact way of writing the vector with entries $x_i s_i$). If we define the function $\mathbb{R}^{2n+m} \to \mathbb{R}^{2n+m}$

$$F(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s}) = \begin{pmatrix} \boldsymbol{A}^\top \boldsymbol{y} + \boldsymbol{s} - \boldsymbol{c} \\ \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \\ \boldsymbol{X}\boldsymbol{S}\boldsymbol{e} \end{pmatrix} \tag{12.1}$$

then what we are looking for is a *root* $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{s}^*)$ of this function, i.e., a point with $F(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{s}^*) = \boldsymbol{0}$, with the additional property that $\boldsymbol{x}^*$ and $\boldsymbol{s}^*$ be non-negative. For later reference, we record the form of the *Jacobian* matrix of F,

$$\boldsymbol{J}F(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s}) = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{A}^\top & \boldsymbol{I} \\ \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{S} & \boldsymbol{0} & \boldsymbol{X} \end{pmatrix}.$$

12.1 Towards an efficient interior-point method

The challenge is to find an iterative method of solving this rootfinding problem while preserving the non-negativity constraints. We discuss three approaches, in increasing order of sophistication. The third of these approaches forms the basis of *primal dual interior point methods*.

A first attempt

A first attempt at solving the problem $F(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s}) = \mathbf{0}$, with $\boldsymbol{x} \geq \mathbf{0}, \boldsymbol{s} \geq \mathbf{0}$, is to apply Newton's method with really small steps. For that, at each step we solve

$$\begin{pmatrix} \mathbf{0} & \boldsymbol{A}^\top & \boldsymbol{I} \\ \boldsymbol{A} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{S}^{(k)} & \mathbf{0} & \boldsymbol{X}^{(k)} \end{pmatrix} \begin{pmatrix} \Delta \boldsymbol{x} \\ \Delta \boldsymbol{y} \\ \Delta \boldsymbol{s} \end{pmatrix} = \begin{pmatrix} \boldsymbol{c} - \boldsymbol{s}^{(k)} - \boldsymbol{A}^\top \boldsymbol{y}^{(k)} \\ \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k)} \\ -\boldsymbol{X}^{(k)}\boldsymbol{S}^{(k)}\boldsymbol{e} \end{pmatrix}.$$

and then compute

$$\begin{pmatrix} \boldsymbol{x}^{k+1} \\ \boldsymbol{y}^{k+1} \\ \boldsymbol{s}^{k+1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}^{k} \\ \boldsymbol{y}^{k} \\ \boldsymbol{s}^{k} \end{pmatrix} + \alpha_k \begin{pmatrix} \Delta \boldsymbol{x} \\ \Delta \boldsymbol{y} \\ \Delta \boldsymbol{s} \end{pmatrix},$$

choosing the step length α_k so that the non-negativity of $\boldsymbol{x}$ and $\boldsymbol{s}$ remains. Note that above we have a + sign in front of alpha. That is because in the system of equations above, we took the negative $-F(\boldsymbol{x}_k, \boldsymbol{y}_k, \boldsymbol{s}_k)$.

Example 12.1. Consider the linear programming problem

$$\begin{aligned} \text{minimize} \quad & x_1 + 2x_2 - 2x_3 \\ & x_1 - 2x_3 \\ & x_2 - x_3 = -1 \\ & x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0. \end{aligned}$$

The data for this problem is given by

$$\boldsymbol{A} = \begin{pmatrix} 1 & 0 & -2 \\ 0 & 1 & -1 \end{pmatrix}, \ \boldsymbol{b} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \ \boldsymbol{c} = \begin{pmatrix} 1 \\ 2 \\ -2 \end{pmatrix}.$$

Based on this data, the function F given by

$$F(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s}) = \begin{pmatrix} y_1 + s_1 - 1 \\ y_2 + s_2 - 2 \\ -2y_1 - y_2 + s_3 + 2 \\ x_1 - 2x_3 - 1 \\ x_2 - x_3 + 1 \\ x_1 s_1 \\ x_2 s_2 \\ x_3 s_3 \end{pmatrix}$$

and the Jacobian matrix

$$\boldsymbol{J}F(\boldsymbol{x},\boldsymbol{y},\boldsymbol{s}) = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -2 & -1 & 0 & 0 & 1 \\ 1 & 0 & -2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ s_1 & 0 & 0 & 0 & 0 & x_1 & 0 & 0 \\ 0 & s_2 & 0 & 0 & 0 & 0 & x_2 & 0 \\ 0 & 0 & s_3 & 0 & 0 & 0 & 0 & x_3 \end{pmatrix}.$$

With this data at hand, we can easily use a Python program to solve the problem for us, making sure that the steplength is small enough. Starting with $\boldsymbol{x}^{(0)} = \boldsymbol{s}^{(0)} = (1,1,1)^\top$, we get the sequence of step lengths

$$\alpha_0 = 0.5455,\ \alpha_1 = 0.5455,\ \alpha_2 = 1,$$

with the corresponding sequence of $\boldsymbol{x}$ vectors

$$\boldsymbol{x}^{(1)} = \begin{pmatrix} 1.1818 \\ 0 \\ 0.5455 \end{pmatrix},\ \boldsymbol{x}^{(2)} = \begin{pmatrix} 2.1736 \\ 0 \\ 0.7934 \end{pmatrix},\ \boldsymbol{x}^{(3)} = \begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix}.$$

One verifies that $\boldsymbol{x}^{(3)}$ is in fact a vertext of the feasible set, and $\langle \boldsymbol{x}^{(3)}, \boldsymbol{c}\rangle = 1$ gives the optimal value.

A second attempt

In general, the method of choosing small step lengths can be slow. A variation would be to solve for

$$F(\boldsymbol{x},\boldsymbol{y},\boldsymbol{s}) = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \tau\boldsymbol{e} \end{pmatrix}, \tag{12.1}$$

for a parameter $\tau > 0$. Given some fixed initial values $\boldsymbol{x} \geq 0$ and $\boldsymbol{s} \geq 0$, compute the *duality measure*

$$\mu = \frac{1}{d}\sum_{i=1}^{d} x_i s_i$$

as the average of the products and using the *centering parameter* $\sigma \in (0,1)$, set $\tau = \sigma\mu$. When aiming to solve (12.1) using Newton's method, we are aiming towards a solution $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{s}^*)$ where instead of asking that $x_i^* s_i^* = 0$, we want that $x_i^* s_i^* = \sigma\mu$, a value that is strictly positive, but smaller than average of the initial value.

In an additional twist to this approach, after starting with an initial guess $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s})$, we perform only *one* Newton step, and then update the duality measure μ with the new values of $\boldsymbol{x}$ and $\boldsymbol{s}$. This way we arrive at the following algorithm:

- Start with $(\boldsymbol{x}^{(0)}, \boldsymbol{y}^{(0)}, \boldsymbol{s}^{(0)})$;
- For each $k \geq 0$, compute

  $$\mu^{(k)} = \frac{1}{d}\sum_{i=1}^{d} x_i s_i$$

  and σ_k. Solve

  $$\begin{pmatrix} \mathbf{0} & \boldsymbol{A}^\top & \boldsymbol{I} \\ \boldsymbol{A} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{S}^{(k)} & \mathbf{0} & \boldsymbol{X}^{(k)} \end{pmatrix} \begin{pmatrix} \Delta\boldsymbol{x} \\ \Delta\boldsymbol{y} \\ \Delta\boldsymbol{s} \end{pmatrix} = \begin{pmatrix} \boldsymbol{c} - \boldsymbol{s}^{(k)} - \boldsymbol{A}^\top\boldsymbol{y}^{(k)} \\ \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(k)} \\ -\boldsymbol{X}^{(k)}\boldsymbol{S}^{(k)}\boldsymbol{e} + \sigma\mu^{(k)} \end{pmatrix}$$

  and compute

  $$\begin{pmatrix} \boldsymbol{x}^{k+1} \\ \boldsymbol{y}^{k+1} \\ \boldsymbol{s}^{k+1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}^{k} \\ \boldsymbol{y}^{k} \\ \boldsymbol{s}^{k} \end{pmatrix} + \alpha_k \begin{pmatrix} \Delta\boldsymbol{x} \\ \Delta\boldsymbol{y} \\ \Delta\boldsymbol{s} \end{pmatrix},$$

  for a small enough $\alpha_k > 0$ to ensure non-negativity.

A third approach

For the purpose of analysis, it is convenient to let an interior point method operate on vectors that satisfy the first two equalities in (O) exactly. Define the feasible and strictly feasible sets as

$$\begin{aligned} \mathcal{F} &= \{(\boldsymbol{y}, \boldsymbol{s}, \boldsymbol{x}) : \boldsymbol{A}^\top\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c},\ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \geq \mathbf{0},\ \boldsymbol{s} \geq \mathbf{0}\} \\ \mathcal{F}^\circ &= \{(\boldsymbol{y}, \boldsymbol{s}, \boldsymbol{x}) : \boldsymbol{A}^\top\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c},\ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} > \mathbf{0},\ \boldsymbol{s} > \mathbf{0}\} \end{aligned}$$

Restricting to points in $\mathcal{F}^\circ$, the computation of the Newton update in the second approach would change to

$$\begin{pmatrix} \mathbf{0} & \boldsymbol{A}^\top & \boldsymbol{I} \\ \boldsymbol{A} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{S}^{(k)} & \mathbf{0} & \boldsymbol{X}^{(k)} \end{pmatrix} \begin{pmatrix} \Delta\boldsymbol{x} \\ \Delta\boldsymbol{y} \\ \Delta\boldsymbol{s} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ -\boldsymbol{X}^{(k)}\boldsymbol{S}^{(k)}\boldsymbol{e} + \sigma\mu^{(k)} \end{pmatrix}$$

In each iteration, a Newton step is taken in the direction of the *central path*. This is a curve in $\mathcal{F}^\circ$ defined as the set of solutions of

$$\begin{aligned} \boldsymbol{A}^\top\boldsymbol{y} + \boldsymbol{s} - \boldsymbol{c} &= \mathbf{0} \\ \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} &= \mathbf{0} \\ \boldsymbol{X}\boldsymbol{S}\boldsymbol{e} &= \tau\boldsymbol{e} \\ \boldsymbol{x} &> \mathbf{0} \\ \boldsymbol{s} &> \mathbf{0}, \end{aligned} \tag{12.2}$$

where $\tau > 0$. As $\tau \to 0$, any solution of this system will converge to an optimal primal-dual vector $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s})$ for the original linear programming problem. As we will see, practical primal-dual interior point methods will try to ensure that we always move within a neighbourhood of the central path. While this third approach lends itself well to analysis, one problem is that a starting point $(\boldsymbol{x}^{(0)}, \boldsymbol{y}^{(0)}, \boldsymbol{s}^{(0)}) \in \mathcal{F}^{\circ}$ may be hard to find.