# Financial Optimization
# ISE 347/447

## Lecture 11

Dr. Ted Ralphs

# Reading for This Lecture

- C&T Chapter 7

# Iterative Methods for Optimization

- As discussed earlier, many optimization techniques are *iterative* in nature.
- Starting from an initial point, we determine a search direction that will get us to an improved point.
- At the new point, we repeat until a stopping criteria is satisfied.
- The two crucial elements are
  - A measure that can be used to judge improvement.
  - A method for generating a new solution in each iteration.
- Ideally, we should be able to prove that the method will converge.

# One-dimensional Line Search

- One-dimensional line search is the fundamental subproblem for many non-linear algorithms.
- Given a function $f$, a current iterate $\hat{x}$, and a direction $d$, we want to solve the following problem

$$\begin{aligned} &\min f(x + \lambda d) \\ &\text{s.t. } a \le \lambda \le b \end{aligned}$$

# Line Search Methods

- Exact Methods
  - Solve the line search problem analytically.
  - Take the derivative with respect to $\lambda$ and set it to zero.
- Iterative Methods
  - Methods using function evaluations.
  - Methods using derivatives.
  - Generally guaranteed to converge for convex functions.

# The Interval of Uncertainty

- The *interval of uncertainty* is the interval within which the optimal solution has to lie.
- Most derivative-free line search methods are based on iteratively reducing the interval of uncertainty.

**Theorem 1.** *Let $\Theta : \mathbb{R} \to \mathbb{R}$ be strictly convex over the interval $[a, b]$. Let $\lambda, \mu \in [a, b]$ be such that $\lambda < \mu$.*

- *If $\Theta(\lambda) > \Theta(\mu)$, then $\Theta(z) \geq \Theta(\mu)$ for all $z \in [a, \lambda)$.*
- *If $\Theta(\lambda) \leq \Theta(\mu)$, then $\Theta(z) \geq \Theta(\lambda)$ for all $z \in (\mu, b]$.*

# Derivative-free Line Search

- The previous theorem shows that we can reduce the interval of uncertainty through function evaluations.
- There are a number of line search methods based on this idea.
  - Uniform search
  - Dichotomous search
  - Golden section
  - Fibonacci search
- These methods differ essentially in how they choose the points at which to evaluate the function.

# Newton's Method

- Newton's Method is a method for finding roots of an equation of the form $f(x) = 0$ for a continuously differentiable function $f : \mathbb{R} \to \mathbb{R}$.
- The idea is very simple.
  - Start with an initial guess $x^0$, set $k \leftarrow 0$, and repeat the following.
  - Set $x^{k+1}$ (the next guess) to be be the unique root of the first-order approximation of $f$ at $x^k$, which is $f(x) \approx f(x^k) + f'(x^k)(x - x^k)$ and set $k \leftarrow k + 1$.
- Solving the above equation, we get that the guess in iteration $k$ is just $x^{k+1} \leftarrow x^k - f(x^k)/f'(x^k)$.
- By iteratively computing a sequence of guesses, we will (hopefully) converge to a root of the original equation.

# Newton's Method for Minimization

- Newton's Method can also be used to find a point satisfying first order optimality conditions for minimization of the function $f$.
- This is done by applying Newton's Method from the previous slide to the equation $f'(x) = 0$.
- In this case, we can view the method as using a second-order approximation to $f$ at $x^k$.

$$f(x) \approx f(x^k) + f'(x^k)(x - x^k) + \frac{1}{2} f''(x^k)(x - x^k)^2$$

- The next iterate is then taken to be the point at which the derivative of this approximation is zero.

$$\begin{aligned} \Rightarrow \quad & f'(\lambda_k) + f''(\lambda_k)(\lambda_{k+1} - \lambda_k) = 0 \\ \Rightarrow \quad & \lambda_{k+1} = \lambda_k - f'(\lambda_k)/f''(\lambda_k) \end{aligned}$$

- Again, this is just a Newton step applied to the equation $f'(x) = 0$.

# Convergence of Newton's Method

- Newton's method does not always converge.
- There is no measure that is always guaranteed to decrease.
- If the starting point is "close enough," then we can show convergence.
- There is a quadratic fit line search method with global convergence.

# Interior Point Methods for Quadratic Programs

- Consider the quadratic program

$$\begin{aligned} \min\ & \frac{1}{2}x^\top Q x + c^\top x \\ \text{s.t.}\ & Ax = b \\ & x \geq 0 \end{aligned}$$

- Optimality conditions are that there exists a solution to the system

$$F(x, y, s) = \begin{bmatrix} A^\top y - Qx + s - c \\ Ax - b \\ s^\top x \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, (x, s) \geq 0. \quad (1)$$

- Again, we can interpret these as *primal feasibility*, *dual feasibility*, and *complementary slackness*.

# Generalizing Newton's Method

- If not for the nonnegativity constraints, we could simply solve the system of equations yielded by the optimality conditions (1).
- The nonnegativity conditions make the situation a bit more complicated. however.
- To find a solution to the system (1), we can use a variant of Newton's Algorithm.

# The Basic Idea of Interior Point Methods for QP

- Interior point methods are iterative methods for finding a point satisfying the optimality conditions (1).
- We start by finding a point $(x^0, y^0, s^0)$ satisfying PF and DF constraints and for which $x^0 > 0$, $s^0 > 0$.
- Such a point is said to be *strictly feasible* and we will denote the set of all strictly feasible points by $\mathcal{F}_0$.
- Next, we try to find a second strictly feasible point $(x^1, y^1, s^1)$ for which $(s^1)^\top x^1 < (s^2)^\top x^2$.
- By iterating, we try to converge to a point $(x^*, y^*, s^*)$ satisfying $(s^*)^\top x^* = 0$.
- This point will then have to be optimal.

# Newton Steps

- Let's assume we have a strictly feasible point $(x^k, y^k, s^k)$.
- If we apply Newton's Method to the problem of satisfying the QP optimality conditions, the Newton step would be determined by solving

$$J(x^k, y^k, s^k) \begin{bmatrix} \Delta x^k \\ \Delta y^k \\ \Delta s^k \end{bmatrix} = -F(x^k, y^k, s^k). \tag{2}$$

where

$$J(x^k, y^k, s^k) = \begin{bmatrix} -Q & A^\top & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix}$$

and $S^k = \text{Diag}(s^k)$ and $X^k = \text{Diag}(x^k)$.

# Newton Steps

- Since $(x^k, y^k, s^k)$ is strictly feasible, we can reduce (2) to

$$J(x^k, y^k, s^k) \begin{bmatrix} \Delta x^k \\ \Delta y^k \\ \Delta s^k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -X^k S^k e \end{bmatrix} \quad (3)$$

- In the standard Newton's method, the new iterate would then be

$$(x^k, y^k, s^k) + (\Delta x^k, \Delta y^k, \Delta s^k)$$

- However, in this case, we need to take into account the nonnegativity constraints.

# Choosing the Step Size

- Choosing the step size is similar to what we do in the simplex method.
- Using a test similar to the ratio test, we determine how far we can go and remain feasible.
- In contrast to simplex, here we must remain strictly feasible, however.
- We must choose an $\alpha^k$ so that

$$(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k, y^k, s^k) + \alpha_k(\Delta x^k, \Delta y^k, \Delta s^k)$$

is strictly feasible.

# The Central Path

- The central path consists of solutions to the following system

$$F(x^\tau, y^\tau, s^\tau) = \begin{bmatrix} 0 \\ 0 \\ \tau e \end{bmatrix}, (x^\tau, s^\tau) > 0 \qquad (4)$$

for some $\tau > 0$.

- The third set of equations is actually equivalent to

$$x_i^\tau s_i^\tau = \tau$$

- Equation (4) has a unique solution for every $\tau$ as the long as the set of strictly feasible solutions is nonempty.
- More importantly, the path followed by $(x^\tau, y^\tau, s^\tau)$ converges to an optimal solution as $\tau$ goes to zero.

# Path-Following Algorithms

- Path-following algorithms try to improve the convergence rate of the naive approach by generating a sequence of iterates approximating the central path for decreasing values of $\tau$.

- To do so, we use what are called *centered directions*, which are Newton steps for the system

$$\hat{F}(x,y,s) = \begin{bmatrix} Ax - b \\ A^\top y - Qx + s - c \\ XSe - \tau e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{5}$$

  such that $x, s \geq 0$.

- A *centered direction* is then a solution to the system

$$\begin{bmatrix} -Q & A^\top & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x_c^k \\ \Delta y_c^k \\ \Delta s_c^k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \tau e - X^k S^k e \end{bmatrix} \tag{6}$$

# The Duality Gap

- An important question is then what value of $\tau$ to use in this equation.
- The value
$$\mu = \mu(x, s) := \frac{\sum_{i=1}^{n} x_i s_i}{n} = \frac{x^\top s}{n}$$
is called the *duality gap*.
- It is a measure of "closeness to optimality".
- For a point on the central path, $\mu(x, s) = \tau$.
- We can think of the value of $\tau$ as being chosen in relation to the current duality gap.

# Reducing the Duality Gap

- Rewriting (6) to emphasize this using a new parameter $\sigma_k$, our direction can be described as the solution to the equation

$$\begin{bmatrix} -Q & A^\top & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x_c^k \\ \Delta y_c^k \\ \Delta s_c^k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sigma_k \mu^k e - X^k S^k e \end{bmatrix} \tag{7}$$

- If we choose $\sigma_k = 0$, this corresponds to the original *pure Newton step* and is focused purely on decreasing the gap.
- On the other hand, choosing $\sigma_k = 1$ corresponds to moving towards the central path without decreasing the duality gap.
- In practice, a balance must be struck between these two strategies.

# Neighborhoods of the Central Path

- Variants of the interior point methods differ in how the step-size parameter $\sigma^k$ is chosen.
- In general, the idea is to keep the iterates in a neighborhood of the central path.
- Ideally, we would like iterates to be a good "approximation" to a point on the central path, i.e., be within a distance $\epsilon$ of the central path.
- This is difficult to enforce algorithmically.
- Instead, we can try to ensure that the points lie in certain *neighborhoods*.

# Common Neighborhoods

Two of the most commonly used neighborhoods are

$$\mathcal{N}_2(\theta) = \{(x,y,s) \in \mathcal{F}_0 \mid \|XSe - \mu e\| \leq \theta\mu, \mu = \frac{x^\top s}{n}\}$$

for $\theta \in (0,1)$ (the *2-norm neighborhood*) and

$$\mathcal{N}_{-\infty}(\gamma) = \{(x,y,s) \in \mathcal{F}_0 \mid x_i s_i \geq \gamma\mu \ \forall i \in 1,\ldots,m, \mu = \frac{x^\top s}{n}\}$$

for $\gamma \in (0,1)$ (the $-\infty$*-norm neighborhood*). Note that $\theta = 0$ and $\gamma = 1$ correspond to the central path itself.

# Short-Step Versus Long-Step Methods

- For typical values of $\gamma$ and $\theta$, the 2-norm neighborhood is usually much smaller than the $-\infty$-norm neighborhood.
- Requiring iterates to be in the 2-norm neighborhood results in a much more restrictive algorithm, called a *short step algorithm*.
- Requiring iterates to be in the $-\infty$-norm neighborhood results in a less restrictive algorithm.
- The main difference between these two classes of methods is in the theoretical worst-case performance.

# A Generic Long-Step Method

1. Given $\gamma \in (0,1)$ and $0 < \sigma_{\min} < \sigma_{\max} < 1$, choose $(x^0, y^0, s^0) \in \mathcal{N}_{-\infty}(\gamma)$. For $k = 0, 1, \ldots,$ repeat the following steps.

2. Choose $\sigma^k \in [\sigma_{\min}, \sigma_{\max}]$, let $\mu^k = \frac{(x^k)^\top s^k}{n}$. Solve

$$\begin{bmatrix} -Q & A^\top & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x_c^k \\ \Delta y_c^k \\ \Delta s_c^k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sigma_k \mu^k e - X^k S^k e \end{bmatrix}.$$

3. Choose $\alpha^k$ such that $(x^k, y^k, s^k) + \alpha_k(\Delta x^k, \Delta y^k, \Delta s^k) \in \mathcal{N}_{-\infty}(\gamma)$.

4. Set
$$(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k, y^k, s^k) + \alpha_k(\Delta x^k, \Delta y^k, \Delta s^k)$$
   and $k = k + 1$.

# Starting from an Infeasible Point

- Note that the algorithms we discussed assume that we can find a strictly feasible point to initialize the algorithm.
- We can modify the basic algorithm to accommodate points that don't satisfy the equality constraints, as long as we still have $x^0 > 0$, $s^0 > 0$.
- As before, we are still trying to solve the system

$$\hat{F}(x, y, s) = \begin{bmatrix} Ax - b \\ A^\top y - Qx + s - c \\ XSe - \tau e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

such that $x, s \geq 0$.

# Newton Step From an Infeasible Point

- The Newton step is still determined by solving the following system of linear inequalities:

$$J(x^k, y^k, s^k) \begin{bmatrix} \Delta x^k \\ \Delta y^k \\ \Delta s^k \end{bmatrix} = -\hat{F}(x^k, y^k, s^k).$$

- The Newton step is then

$$\begin{bmatrix} -Q & A^\top & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x_c^k \\ \Delta y_c^k \\ \Delta s_c^k \end{bmatrix} = \begin{bmatrix} c + Qx^k - A^\top y^k - s^k \\ b - Ax^k \\ \tau e - X^k S^k e \end{bmatrix}$$