# 15.081J/6.251J Introduction to Mathematical Programming

Lecture 22: Primal-dual Barrier Interior Point Algorithm

### 1 Outline

- 1. The Barrier Problem
- 2. Solving Equations
- 3. The Primal-Dual Barrier Algorithm
- 4. Insight on Behavior
- 5. Computational Aspects
- 6. Conclusions

### 2 The Barrier Problem

Barrier problem:

min 
$$B_{\mu}(\boldsymbol{x}) = \boldsymbol{c}'\boldsymbol{x} - \mu \sum_{j=1}^{n} \log x_j$$
  
s.t.  $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ 

KKT:

$$c - \mu \left(\frac{1}{x_1(\mu)}, \dots, \frac{1}{x_n(\mu)}\right)' + \mathbf{A'} \mathbf{p}(\mu) = \mathbf{0}$$
$$\mathbf{A} \mathbf{x}(\mu) = \mathbf{b}, \qquad \mathbf{x}(\mu) \ge 0$$

2.1 Optimality Conditions

Set 
$$s_j(\mu) = \frac{\mu}{x_j(\mu)}$$
  
 $Ax(\mu) = b$   
 $x(\mu) \ge 0$   
 $A'p(\mu) + s(\mu) = c$   
 $s(\mu) \ge 0$   
 $s_j(\mu)x_j(\mu) = \mu$  or  
 $X(\mu)S(\mu)e = e\mu$ 

 $\boldsymbol{X}(\mu) = \operatorname{diag}(x_1(\mu), \dots, x_n(\mu)), \, \boldsymbol{S}(\mu) = \operatorname{diag}(s_1(\mu), \dots, s_n(\mu))$ 

1

SLIDE 2

SLIDE 3

# 3 Solving Equations

$$F(oldsymbol{z}) = \left[egin{array}{c} oldsymbol{A} x - oldsymbol{b} \ oldsymbol{A}' p + s - oldsymbol{c} \ oldsymbol{X} S e - \mu e \end{array}
ight]$$

 $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{p}, \boldsymbol{s}), \, r = 2n + m$ Solve

$$\boldsymbol{F}(\boldsymbol{z}^*) = \boldsymbol{0}$$

#### 3.1 Newton's method

$$F(\boldsymbol{z}^k + \boldsymbol{d}) \approx F(\boldsymbol{z}^k) + J(\boldsymbol{z}^k)\boldsymbol{d}$$

SLIDE 5

SLIDE 4

Here  $J(z^k)$  is the  $r \times r$  Jacobian matrix whose (i, j)th element is given by

$$egin{aligned} & \left. rac{\partial F_i(oldsymbol{z})}{\partial z_j} 
ight|_{oldsymbol{z}=oldsymbol{z}^k} \ & oldsymbol{F}(oldsymbol{z}^k) + oldsymbol{J}(oldsymbol{z}^k) oldsymbol{d} = oldsymbol{0} \end{aligned}$$

Set  $\boldsymbol{z}^{k+1} = \boldsymbol{z}^k + \boldsymbol{d}$  ( $\boldsymbol{d}$  is the Newton direction) ( $\boldsymbol{x}^k, \boldsymbol{p}^k, \boldsymbol{s}^k$ ) current primal and dual feasible solution Newton direction  $\boldsymbol{d} = (\boldsymbol{d}_x^k, \boldsymbol{d}_p^k, \boldsymbol{d}_s^k)$ 

$$\left[ egin{array}{ccc} m{A} & m{0} & m{0} \ m{0} & m{A'} & m{I} \ m{S}_k & m{0} & m{X}_k \end{array} 
ight] \left[ egin{array}{ccc} m{d}_k^k \ m{d}_p^k \ m{d}_s^k \end{array} 
ight] = - \left[ egin{array}{ccc} m{Ax^k - b} \ m{A'p^k + s^k - c} \ m{X_k S_k e - \mu^k e} \end{array} 
ight]$$

#### 3.2 Step lengths

 $egin{aligned} m{x}^{k+1} &= m{x}^k + eta_P^k m{d}_x^k \ m{p}^{k+1} &= m{p}^k + eta_D^k m{d}_p^k \ m{s}^{k+1} &= m{s}^k + eta_D^k m{d}_s^k \end{aligned}$ 

To preserve nonnegativity, take

$$\begin{split} \beta_P^k &= \min\left\{1, \alpha \min_{\{i \mid (d_x^k)_i < 0\}} \left(-\frac{x_i^k}{(d_x^k)_i}\right)\right\},\\ \beta_D^k &= \min\left\{1, \alpha \min_{\{i \mid (d_s^k)_i < 0\}} \left(-\frac{s_i^k}{(d_s^k)_i}\right)\right\}, \end{split}$$

 $0 < \alpha < 1$ 

SLIDE 6

# 4 The Primal-Dual Barrier Algorithm

- 1. (Initialization) Start with  $x^0 > 0$ ,  $s^0 > 0$ ,  $p^0$ , and set k = 0
- **2.** (Optimality test) If  $(s^k)'x^k < \epsilon$  stop; else go to Step 3.
- **3.** (Computation of Newton directions)

$$\mu^{k} = \frac{(s^{k})' \boldsymbol{x}^{k}}{n}$$
$$\boldsymbol{X}_{k} = \operatorname{diag}(x_{1}^{k}, \dots, x_{n}^{k})$$
$$\boldsymbol{S}_{k} = \operatorname{diag}(s_{1}^{k}, \dots, s_{n}^{k})$$

Solve linear system

$$\left[egin{array}{ccc}oldsymbol{A}&oldsymbol{0}&oldsymbol{a}'&I\ oldsymbol{S}_k&oldsymbol{0}&oldsymbol{X}_k\end{array}
ight] \left[egin{array}{ccc}oldsymbol{d}_x^k\ oldsymbol{d}_p^k\ oldsymbol{d}_p^k\ oldsymbol{d}_s^k\end{array}
ight] = - \left[egin{array}{ccc}oldsymbol{A}x^k-b\ oldsymbol{A}'p^k+s^k-c\ oldsymbol{X}_koldsymbol{S}_ke-\mu^ke\end{array}
ight]$$

SLIDE 9

4. (Find step lengths)

$$\begin{split} \beta_P^k \ &= \ \min\left\{1, \alpha \ \min_{\{i \mid (d_x^k)_i < 0\}} \left(-\frac{x_i^k}{(d_x^k)_i}\right)\right\} \\ \beta_D^k \ &= \ \min\left\{1, \alpha \ \min_{\{i \mid (d_s^k)_i < 0\}} \left(-\frac{s_i^k}{(d_s^k)_i}\right)\right\} \end{split}$$

5. (Solution update)

$$egin{aligned} &oldsymbol{x}^{k+1}\,=\,oldsymbol{x}^k+eta_P^koldsymbol{d}_x^k\ &oldsymbol{p}^{k+1}\,=\,oldsymbol{p}^k+eta_D^koldsymbol{d}_p^k\ &oldsymbol{s}^{k+1}\,=\,oldsymbol{s}^k+eta_D^koldsymbol{d}_s^k \end{aligned}$$

6. Let k := k + 1 and go to Step 2

# 5 Insight on behavior

• Affine Scaling

$$d_{ ext{affine}} = - oldsymbol{X}^2 \Big( oldsymbol{I} - oldsymbol{A}' (oldsymbol{A}oldsymbol{X}^2 oldsymbol{A}')^{-1} oldsymbol{A}oldsymbol{X}^2 \Big) oldsymbol{c}$$

• Primal barrier

$$oldsymbol{d}_{ ext{primal-barrier}} = igg( oldsymbol{I} - oldsymbol{X}^2 oldsymbol{A}')^{-1} oldsymbol{A} igg) igg( oldsymbol{X} oldsymbol{e} - rac{1}{\mu} oldsymbol{X}^2 oldsymbol{c} igg)$$

SLIDE 10

• For  $\mu = \infty$ 

$$oldsymbol{d}_{ ext{centering}} = igg( oldsymbol{I} - oldsymbol{X}^2 oldsymbol{A}' (oldsymbol{A} oldsymbol{X}^2 oldsymbol{A}')^{-1} oldsymbol{A} igg) oldsymbol{X} oldsymbol{e}$$

• Note that

$$m{d}_{ ext{primal-barrier}} = m{d}_{ ext{centering}} + rac{1}{\mu}m{d}_{ ext{affine}}$$

- When  $\mu$  is large, then the centering direction dominates, i.e., in the beginning, the barrier algorithm takes steps towards the analytic center
- When  $\mu$  is small, then the affine scaling direction dominates, i.e., towards the end, the barrier algorithm behaves like the affine scaling algorithm

#### 6 Computational aspects of IPMs

Simplex vs. Interior point methods (IPMs)

- Simplex method tends to perform poorly on large, massively degenerate problems, whereas IP methods are much less affected.
- Key step in IPMs

$$(AX_k^2A')d = f$$

• In implementations of IPMs  $AX_k^2A'$  is usually written as

$$AX_k^2A' = LL',$$

where L is a square lower triangular matrix called the *Cholesky factor* 

• Solve system

$$(AX_k^2A')d = f$$

by solving the triangular systems

$$Ly = f, \qquad L'd = y$$

- The construction of L requires  $O(n^3)$  operations; but the actual computational effort is highly dependent on the sparsity (number of nonzero entries) of L
- Large scale implementations employ heuristics (reorder rows and columns of A) to improve sparsity of L. If L is sparse, IPMs are stronger.

### 7 Conclusions

- IPMs represent the present and future of Optimization.
- Very successful in solving very large problems.
- Extend to general convex problems

SLIDE 12

6.251J / 15.081J Introduction to Mathematical Programming Fall 2009

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.