Numerical Optimization

A Workshop

At

Department of Mathematics

Chiang Mai University

August 4-15, 2009

Instructor: **Vira Chankong**

Electrical Engineering and Computer Science
Case Western Reserve University

Phone: 216 368 4054, Fax: 216 368 3123
E-mail: **vira@case.edu**

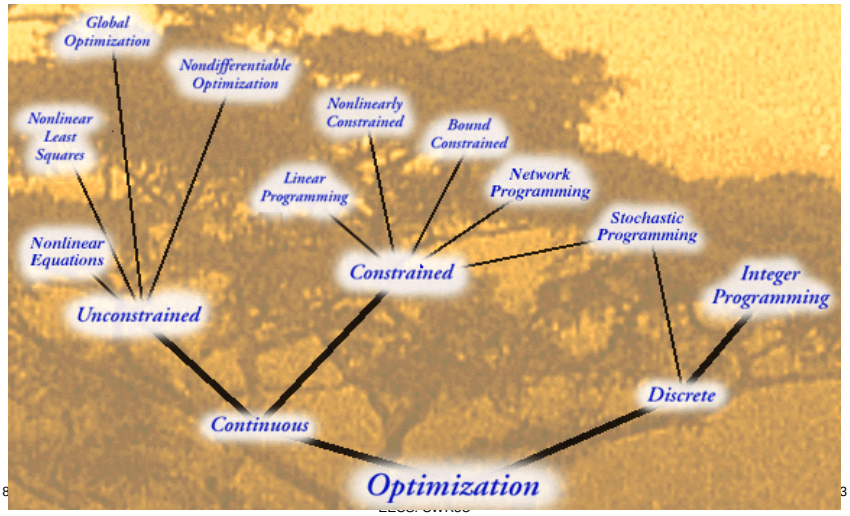
Session:

Methods For Constrained Linear Optimization Problems (LP)

Vira Chankong

Case Western Reserve University
Electrical Engineering and Computer Science

NEOS Guide Optimization Tree



Linear Programming (LP)

The Structure and Geometry of LP:

Standard Form of LP: $\min z = \mathbf{c}^T \mathbf{x}$

$$s.t. \quad \left.\begin{array}{l} \mathbf{Ax} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \end{array}\right\} \; X \subseteq R^n$$

$\mathbf{x} \in R^n$

$\mathbf{A} \in R^{m \times n}$; $\mathbf{b} \in R^m_+$; $\mathbf{c} \in R^n$

Assume: i) $m < n$ (for a meaningful/interesting LP problem)

ii) $rank(\mathbf{A}) = m$ (i.e. no redundant constraints)

LP: The Key Observations

- LP is convex : So any optimal solution found must be global. (LP is convex because X is convex--see below-- and f is linear)
- An LP either has a finite optimal solution or is unbounded (i.e. z can be reduced indefinitely).
- The feasible set X, being an intersections of hyperplanes and half-spaces is a polyhedron or polytope (bounded polyhedron). It is therefore completely characterized by a nonzero finite number of vertices (extreme points) and a finite number of extreme rays (if X is unbounded). A bounded X is a convex hull of its extreme points. In any case. X is always a convex set.

LP: Key Observations (cont.)

- If an LP has a finite optimal value z^*, at least one of its solutions x^* must occur at one of its vertices (extreme points)
- Thus to search for a finite solution of an LP, it one exists, we only need to search among its finite number of vertices. This is the basis of the mighty SIMPLEX method.

Basic Steps of the Simplex Method: The Geometrical View

Search Strategy: Basic Steps of The Simplex Method:

- Start with an **initial vertex**, **if one exists**
- Identify a **better adjacent** vertex and **move** to it
- Repeat until (1) no better adjacent vertex to the current vertex exists. The CURRENT vertex is then a **global optimum** of the LP by virtue of convexity of the LP, or (2) it can be seen that z can be *reduced indefinitely within X, i.e.* ***z is unbounded*** . *(*Note that for z to be unbounded, then so must X. However the converse is not true. That is X can be unbounded but the optimal z^* is still finite)

Convergence of the Simplex Method

Finite Convergence of the Simplex:

If in every move made from one vertex to the next, at least some FINITE amount of improvement can be made on z, then the Simplex method will always terminate at an optimal vertex in a finite number of steps.

This is because for every move made, the new vertex will be different from the current one. And because of the strictly monotone improvement on z, no vertex will be visited more than once. And since there are only a finite number of vertices of X, an optimal vertex must be found in a finite number of moves.

LP: Basic Steps and Convergence of the Simplex Method

To implement the simplex method algebraically, we must be able to answer the following questions efficiently:

- How do we define a vertex algebraically?
  Basic feasible solution (BFS)/Canonical Form/Tableau
- How do we find an initial vertex, if one exists?
  2-phase/Big M method
- How do we identify a better adjacent vertex?
  Most-negative coefficient rule and minimum ration rule
- How do we move from one vertex to an adjacent one?
  Pivot operation (Pivot form or product form of inverse)
- How do we know when to stop? And what conclusion can we make?
  Terminating conditions

An Algebraic View of the Simplex Method

Consider an LP in standard form:

P: $\min \ \mathbf{c}^T\mathbf{x}$

$s.t. \ \mathbf{Ax} = \mathbf{b}$

$\mathbf{x} \geq \mathbf{0}$

where $\mathbf{A} \in R^{m\times n}$, $\mathbf{b} \in R_+^m$, $\mathbf{c} \in R^n$

$\Leftrightarrow$

The corresponding dual is:

D: $\max z = \mathbf{b}^T\mathbf{y}$

$s.t. \ \mathbf{A}^T\mathbf{y} + \mathbf{s} = \mathbf{c}$

$\mathbf{s} \geq \mathbf{0}$

Since both P and D are linear (convex), $\mathbf{x}^*$ is global optimal for P and $(\mathbf{y}^*,\mathbf{s}^*)$ is global optimal for D if they are the KKT point of their respective problem.

Characterizing an Optimal point of LP and its Dual

That is, (**x***, **y***,**s***) is the primal-dual (global) solution if and only if it satisfies the following joint KKT:

KKT: $\mathbf{A}^T\mathbf{y} + \mathbf{s} = \mathbf{c}$ (1)

$\mathbf{Ax} = \mathbf{b}$ (2)

$\mathbf{x}^T\mathbf{s} = \mathbf{0}$ $(x_i s_i = 0,$ for all $i = 1,..,n)$ (3)

$\mathbf{x} \geq \mathbf{0}, \mathbf{s} \geq \mathbf{0}$ (4)

Or (**x***, **y***,**s***) is the primal-dual optimal if and only if it satisfies

a) Primal feasibility ---Eq. (2) and $\mathbf{x} \geq \mathbf{0}$ in (4)

b) Dual feasibility ---Eq. (1) and $\mathbf{s} \geq \mathbf{0}$ in (4)

c) Complementary Slackness---Eq. (3)

Solving Linear Programs

BV	$\hat{\mathbf{c}}_B^T$	$\hat{\mathbf{c}}_N^T$	RHS	
z	**0**	$\mathbf{c}_N^T - \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{N}$	$\mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{b}$	
$\mathbf{x}_B$	**I**	$\mathbf{B}^{-1}\mathbf{N}$	$\hat{\mathbf{b}}$	$(= \mathbf{B}^{-1}\mathbf{b})$

To solve an LP is therefore to find a way to achieve all three conditions (a), (b) and (c):

•Primal simplex: Maintains (a) and (c) (by keeping canonical forms at all time $\hat{\mathbf{b}} \geq \mathbf{0}$) and works toward achieving (b)—by achieving nonnegative *rcc*'s: $\hat{\mathbf{c}}_N^T \geq \mathbf{0}$

•Dual simplex: Maintains (b): $\hat{\mathbf{c}}_N^T \geq \mathbf{0}$ and (c) by keeping an-almost canonical form, and works toward achieving (b): $\hat{\mathbf{b}} \geq \mathbf{0}$

- Primal-dual simplex: maintains (c) (by keeping tableau forms or basic solutions) and works toward achieving (a)$\hat{\mathbf{b}} \geq \mathbf{0}$ and (b): $\hat{\mathbf{c}}_N^T \geq \mathbf{0}$. This avoids Phase I.

Solving Linear Programs

BV	$\hat{\mathbf{c}}_B^T$	$\hat{\mathbf{c}}_N^T$	RHS
z	**0**	$\mathbf{c}_N^T - \mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{N}$	$\mathbf{c}_B^T\mathbf{B}^{-1}\mathbf{b}$
$\mathbf{x}_B$	**I**	$\mathbf{B}^{-1}\mathbf{N}$	$\hat{\mathbf{b}} \quad \left(=\mathbf{B}^{-1}\mathbf{b}\right)$

But by enforcing (c) in terms of canonical forms (or variants) above, the solution path has to visit vertices of the feasible set. Even though the cost of moving from one vertex to the next is very cheap, but if the number of vertices to visit is very large as in very large LPs, then the total cost may indeed be high. In fact, theoretically, the worst-case complexity of the simplex method is exponential (although in practice it is really rare for the simplex method to experience the worst-case complexity).

An alternative strategy is not to maintain (c) and follow a solution path "interior" to the feasible set— interior-point methods

Interior-Point Methods

for

Large Linear Programs

Vira Chankong
Case Western Reserve University
Electrical Engineering and Computer Science

Alternative way to Solving LPs

Let's relax (c) by relaxing the complementary slackness as shown:

$$\mathbf{A}^T\mathbf{y} + \mathbf{s} = \mathbf{c} \qquad (1)$$

$$\mathbf{A}\mathbf{x} = \mathbf{b} \qquad (2)$$

$$x_i s_i = \tau \text{ for all } i = 1,..,n, \quad \tau > 0 \qquad (3)$$

$$\mathbf{x} > \mathbf{0}, \mathbf{s} > \mathbf{0} \qquad (4)$$

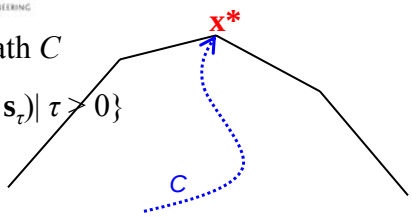
For any $\tau > 0$, let $(\mathbf{x}_\tau, \mathbf{y}_\tau, \mathbf{s}_\tau)$ be a solution of (1)-(4). Then the locus of $(\mathbf{x}_\tau, \mathbf{y}_\tau, \mathbf{s}_\tau)$ as $\tau \to 0$ traces an interior path—named the central path, to the optimal primal-dual solution $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{s}^*)$.

The idea is then to solve (1)-(4) for a series of values of τ_k (making sure that $\tau_k \to 0$), then we have a new class of methods for solving LPs---interior-point methods. This class of methods can be shown to have polynomial complexity.

Central Path in the Primal Feasible Set

Central Path C

$= \{(\mathbf{x}_\tau, \mathbf{y}_\tau, \mathbf{s}_\tau) | \tau > 0\}$



$$\mathbf{A}^T\mathbf{y} + \mathbf{s} = \mathbf{c} \qquad (1)$$

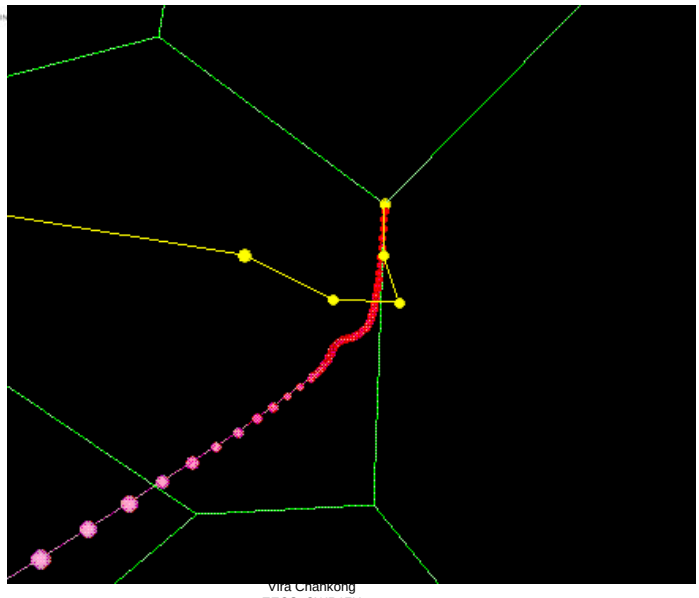
$$\mathbf{A}\mathbf{x} = \mathbf{b} \qquad (2)$$

$$x_i s_i = \tau,\ i = 1,..,n, \quad \tau > 0 \qquad (3)$$

$$\mathbf{x} > \mathbf{0}, \mathbf{s} > \mathbf{0} \qquad (4)$$

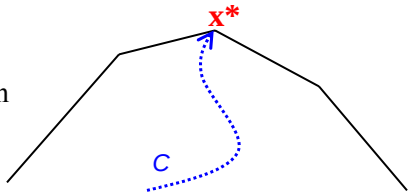
Questions:

1. Does (1)-(4) always have a solution for a given $\tau > 0$?
2. If so, is it always unique?
3. If so, how can we solve (1)-(4) efficiently for a given τ? ---The key questions are how can we maintain (4) and stay in the neighborhood of the central path C at all time, and can we numerically solve (1)-(3) efficiently?
4. How can we vary τ to 0 efficiently?



Central Path and Log-Barrier Function

To answer questions (1) and (2), we first recognize that the central path C can be created by solving the log-barrier function of the primal P to prevent the solution from reaching a boundary corresponding to $x_i = 0$ for some i:



$$P_\tau: \quad \min \ \mathbf{c}^T\mathbf{x} - \tau\sum_{j=1}^{n}\log x_j \qquad \tau > 0$$

$$s.t. \quad \mathbf{Ax} = \mathbf{b}$$

Lagrangian of P_τ is:

$$L(\mathbf{x},\tau) = \mathbf{c}^T\mathbf{x} - \tau\sum_{j=1}^{n}\log x_j + \mathbf{y}^T(\mathbf{b}-\mathbf{Ax})$$

Clearly (i)-(iv) are equivalent to (1)-(4) with $x_j > 0$, $s_j > 0$, for all j

Thus the KKT conditions for P_τ are :

$$\frac{\partial L}{\partial x_j}: \quad c_j - \frac{\tau}{x_j} - \sum_{i=1}^{m} y_i a_{ij} = 0$$

$$\Rightarrow c_j - s_j - \sum_{i=1}^{m} y_i a_{ij} = 0 \ \text{ if } s_j = \frac{\tau}{x_j}$$

$$\Rightarrow \frac{\partial L}{\partial \mathbf{x}}: \quad \mathbf{A}^T\mathbf{y} + \mathbf{s} = \mathbf{c} \qquad \text{(i)}$$

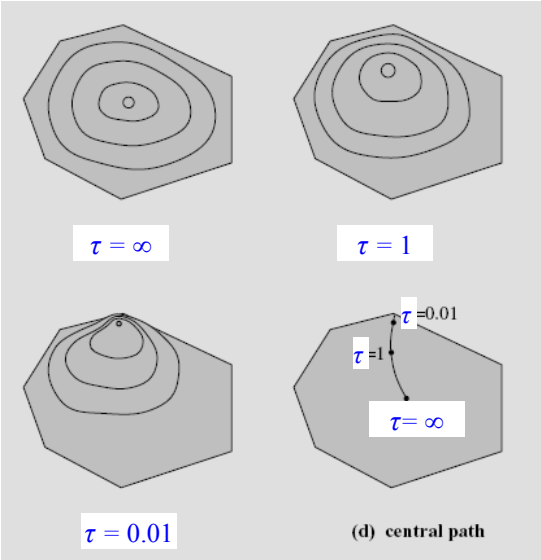
$$\frac{\partial L}{\partial \mathbf{y}}: \quad \mathbf{Ax} = \mathbf{b} \qquad \text{(ii)}$$

and clearly $x_j s_j = \tau$, (iii)

and $x_j > 0,\ s_j > 0$ for j=1,..,n (iv)

Central Path and Log-Barrier Function

Thus solving (1)-(4) for a given $\tau > 0$ is equivalent to solving P_τ. Hence if (1)-(4) has a solution so does P_τ. So the central path C can indeed be generated by solving a family of P_τ as $\tau \to 0$.



Uniqueness of Solution of (1)-(4)

Moreover, if (1)-(4) has a solution then it is unique. This is seen by showing that if P_τ has a minimizer then it is unique.

Assume that (1)-(4), hence (i)-(iv), has a solution $(\mathbf{x}_\tau, \mathbf{y}_\tau, \mathbf{s}_\tau)$ for a given $\tau > 0$, then the Hessian of $L(\mathbf{x}, \mathbf{y})$ with respect to $\mathbf{x}$ at $\mathbf{x}_\tau$ is:

$$\nabla^2 L(\mathbf{x}_\tau, \mathbf{y}_\tau) = \begin{pmatrix} \tau/x_1^2 & . & 0 \\ . & . & . \\ 0 & . & \tau/x_n^2 \end{pmatrix}$$

which is clearly positive definite.

Hence, $(\mathbf{x}_\tau, \mathbf{y}_\tau, \mathbf{s}_\tau)$ is a unique global minimizer of P_τ and is a unique solution of (1)-(4).

Existence of Solution of (1)-(4)

It can also be shown that, solution of (1)-(4) always exists for any given $\tau > 0$ if and only if the primal problem P and the dual D both have nonempty interiors.

See a sketch of the proof next slide or in Vanderbei, *Linear Programming: Foundations and Extensions*, second edition, Kluwer, 2001.

Existence of Solution of (1)-(4)

The "only if" part is trivial and not important. So we prove the "if" part.

First, for any given $\tau > 0$, solution of (1)-(4) exists if and only if solution of the corresponding log-barrier problem exists.

$$P: \min_{x \in C} z(x) = c^T x - \tau \sum_{j=1}^{n} \log(x_j), \text{ where } C = \{x \in R^n \mid Ax = b, x \geq 0\}$$

Now since there exist $(\hat{x}, \hat{y}, \hat{s})$ such that $\hat{x} > 0, A\hat{x} = b$, and $A^T \hat{y} + \hat{s} = c, \hat{s} > 0$, we can rewrite $z(x)$ for any $x \in C$ as :

$$z(x) = (A^T \hat{y} + \hat{s})^T x - \tau \sum_{j=1}^{n} \log(x_j) = \sum_{j=1}^{n} \left(\hat{s}_j x_j - \tau \log(x_j)\right) + b^T \hat{y} \quad (1)$$

Thus solving P is equivalent to solving : $\min z(x) = \sum_{j=1}^{n} \left(\hat{s}_j x_j - \tau \log(x_j)\right) + b^T \hat{y}$

$$s.t. \; x \in \hat{C} = C \cap \{x \in R^n \mid z(x) \leq z(\hat{x})\}$$

Since $\hat{b}^T \hat{y}$ is constant and each term in the summand $z_j(x_j)$ is unimodal $\left(\frac{dz_j}{dx_j} = \hat{s}_j - \frac{\tau}{x_j}\right)$ with a unique minimizer at $\tau/\hat{s}_j > 0$. Hence $\hat{C}$ is closed and bounded. Since $z(x)$ is continuous, it must have a minimizer in $\hat{C}$ as required to be shown.

Solving (1)-(4)—Following the Central Path

It is clear then that we can follow the central path to a solution of P and D by solving a family of (1)-(4) for $\tau \to 0$.

Now we turn to Questions (3) and (4). The following predictor-corrector Primal-Dual method by Mehrotra is most popular:

Predictor-Corrector Primal-Dual Version

1. Given $(\mathbf{x}^{(0)}, \mathbf{y}^{(0)}, \mathbf{s}^{(0)})$ with $(\mathbf{x}^{(0)}, \mathbf{s}^{(0)}) > \mathbf{0}$, set $k = 0$.
2. Check for optimality: STOP if all of the following are true:
   - primal feasibility: $\left\|\mathbf{r}_b^{(k)}\right\| = \left\|\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}\right\| \le \varepsilon_1 \left(1 + \|\mathbf{b}\|\right)$
   - dual feasibility: $\left\|\mathbf{r}_c^{(k)}\right\| = \left\|\mathbf{A}^T\mathbf{y}^{(k)} + \mathbf{s}^{(k)} - \mathbf{c}\right\| \le \varepsilon_2 \left(1 + \|\mathbf{c}\|\right)$
   - duality gap: $\left(\mathbf{x}^{(k)}\right)^T \mathbf{s}^{(k)} \le \varepsilon_3$
3. Solve

$$\begin{pmatrix} \mathbf{0} & \mathbf{A}^T & \mathbf{I} \\ \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{S} & \mathbf{0} & \mathbf{X} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{x}^{aff} \\ \Delta\mathbf{y}^{aff} \\ \Delta\mathbf{s}^{aff} \end{pmatrix} = \begin{pmatrix} -\mathbf{r}_c^{(k)} \\ -\mathbf{r}_b^{(k)} \\ -\mathbf{X}\mathbf{S}\mathbf{e} \end{pmatrix}$$

to get predicted Newton's direction $\begin{pmatrix} \Delta\mathbf{x}^{aff} \\ \Delta\mathbf{y}^{aff} \\ \Delta\mathbf{s}^{aff} \end{pmatrix}$

Predictor-Corrector Primal-Dual Version

CASE SCHOOL OF ENGINEERING

4. Compute predicted stepsizes: $\alpha_{aff}^{primal} = \min\left(1, \min_{i:\Delta x_i^{aff}<0} \frac{-x_i^{(k)}}{\Delta x_i^{aff}}\right)$; $\alpha_{aff}^{dual} = \min\left(1, \min_{i:\Delta s_i^{aff}<0} \frac{-s_i^{(k)}}{\Delta s_i^{aff}}\right)$

Compute $\mathbf{x}^{aff} = \mathbf{x}^{(k)} + \alpha_{aff}^{primal} \Delta \mathbf{x}^{aff})$

$\mathbf{y}^{aff} = \mathbf{y}^{(k)} + \alpha_{aff}^{dual} \Delta \mathbf{y}^{aff}$

$\mathbf{s}^{aff} = \mathbf{s}^{(k)} + \alpha_{aff}^{dual} \Delta \mathbf{s}^{aff}$

Compute estimated duality gap measure $\mu_{aff} = \frac{\left(\mathbf{x}^{aff}\right)^T \mathbf{s}^{aff}}{n}$; and $\mu_k = \frac{\left(\mathbf{x}^{(k)}\right)^T \mathbf{s}^{(k)}}{n}$

and estimated centering parameter $\sigma_k = \left(\frac{\mu_{aff}}{\mu_k}\right)^3$

5. Solve

$$\begin{pmatrix} \mathbf{0} & \mathbf{A}^T & \mathbf{I} \\ \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{S} & \mathbf{0} & \mathbf{X} \end{pmatrix} \begin{pmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \\ \Delta \mathbf{s} \end{pmatrix} = \begin{pmatrix} -\mathbf{r}_c^{(k)} \\ -\mathbf{r}_b^{(k)} \\ -\mathbf{XSe} - \Delta \mathbf{X}_{aff} \Delta \mathbf{S}_{aff} \mathbf{e} + \sigma_k \mu_k \mathbf{e} \end{pmatrix}$$

to get corrected centering direction $\begin{pmatrix} \Delta \mathbf{x}^{(k)} \\ \Delta \mathbf{y}^{(k)} \\ \Delta \mathbf{s}^{(k)} \end{pmatrix}$

Predictor-Corrector Primal-Dual Version

CASE SCHOOL OF ENGINEERING

6. Compute full stepsizes: $\alpha_{\max}^{primal} = \min\left(1, \min_{i:\Delta x_i^{aff}<0} \frac{-x_i^{(k)}}{\Delta x_i^{(k)}}\right)$; $\alpha_{aff}^{dual} = \min\left(1, \min_{i:\Delta s_i^{aff}<0} \frac{-s_i^{(k)}}{\Delta s_i^{(k)}}\right)$

Compute the shortened stepsizes to ensure strict interior (i.e. $\mathbf{x}^{(k)} > 0$ and $\mathbf{s}^{(k)} > 0$):

$\alpha_k^{primal} = \min\left(1, \eta \alpha_{\max}^{primal}\right)$; $\alpha_k^{dual} = \min\left(1, \eta \alpha_{\max}^{primal}\right)$; where $0.9 \le \eta < 1$

Compute $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k^{primal} \Delta \mathbf{x}^{(k)}$

$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \alpha_k^{dual} \Delta \mathbf{y}^{(k)}$

$\mathbf{s}^{(k+1)} = \mathbf{s}^{(k)} + \alpha_k^{dual} \Delta \mathbf{s}^{(k)}$

Repeat Step 2.

Notes:

1. It can be shown that a simpler version of interior-point methods is a polynomial algorithm.
2. Predictor-corrector algorithms as above perform very well in practice. In fact they are competitive (sometime even better than) with the simplex method for very large LPs.

Implementation

CASE SCHOOL OF ENGINEERING

The most expensive steps are Steps 3 and 5, which involve solving a system of linear equations of the form:

$$\begin{pmatrix} \mathbf{0} & \mathbf{A}^T & \mathbf{I} \\ \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{S} & \mathbf{0} & \mathbf{X} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \\ \Delta\mathbf{s} \end{pmatrix} = \begin{pmatrix} -\mathbf{r}_1 \\ -\mathbf{r}_2 \\ -\mathbf{r}_3 \end{pmatrix}$$

Two most effective ways to solve the above system begin with the following reduction step (eliminate $\Delta\mathbf{s}$):

$$\begin{aligned} \mathbf{A}^T\Delta\mathbf{y} + \Delta\mathbf{s} &= -\mathbf{r}_1 \quad (1) \\ \mathbf{A}\Delta\mathbf{x} &= -\mathbf{r}_2 \quad (2) \\ \mathbf{S}\Delta\mathbf{x} + \mathbf{X}\Delta\mathbf{s} &= -\mathbf{r}_3 \quad (3) \end{aligned}$$

Implementation

CASE SCHOOL OF ENGINEERING

$$\begin{pmatrix} \mathbf{0} & \mathbf{A}^T & \mathbf{I} \\ \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{S} & \mathbf{0} & \mathbf{X} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \\ \Delta\mathbf{s} \end{pmatrix} = \begin{pmatrix} -\mathbf{r}_1 \\ -\mathbf{r}_2 \\ -\mathbf{r}_3 \end{pmatrix} \Rightarrow \begin{aligned} \mathbf{A}^T\Delta\mathbf{y} + \Delta\mathbf{s} &= -\mathbf{r}_1 \quad (1) \\ \mathbf{A}\Delta\mathbf{x} &= -\mathbf{r}_2 \quad (2) \\ \mathbf{S}\Delta\mathbf{x} + \mathbf{X}\Delta\mathbf{s} &= -\mathbf{r}_3 \quad (3) \end{aligned}$$

To solve (1)-(3) we first eliminate $\Delta\mathbf{s}$.

From (3), $\Delta\mathbf{s} = -\mathbf{X}^{-1}\mathbf{S}\Delta\mathbf{x} - \mathbf{X}^{-1}\mathbf{r}_3$

Substituting in (1) and rearranging,

$$\mathbf{A}^T\Delta\mathbf{y} - \mathbf{X}^{-1}\mathbf{S}\Delta\mathbf{x} = -\mathbf{r}_1 + \mathbf{X}^{-1}\mathbf{r}_3 \Rightarrow \mathbf{X}^{-1}\mathbf{S}\Delta\mathbf{x} - \mathbf{A}^T\Delta\mathbf{y} = \mathbf{r}_1 - \mathbf{X}^{-1}\mathbf{r}_3$$

$$\mathbf{A}\Delta\mathbf{x} = -\mathbf{r}_2 \qquad \mathbf{A}\Delta\mathbf{x} = -\mathbf{r}_2$$

This is called the *augmented system*.

$$\Rightarrow \begin{pmatrix} \mathbf{D}^{-2} & -\mathbf{A}^T \\ -\mathbf{A} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{r}_1 - \mathbf{X}^{-1}\mathbf{r}_3 \\ \mathbf{r}_2 \end{pmatrix} \text{ where } \mathbf{D}^2 = \mathbf{S}^{-1}\mathbf{X} \text{ (which is } pd\text{)}$$

Clearly $\begin{pmatrix} \mathbf{D}^{-2} & -\mathbf{A}^T \\ -\mathbf{A} & \mathbf{0} \end{pmatrix}$ is symmetric and sparse (if $\mathbf{A}$ is).

The augmented system can be solved efficiently using the *sparse symmetric indefinite factorization* (discussed elsewhere).

It is however less stable than the next method.

Alternative method: Normal Equations

From the augmented system:

$$\begin{pmatrix} \mathbf{D}^{-2} & -\mathbf{A}^T \\ -\mathbf{A} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{r}_1 - \mathbf{X}^{-1}\mathbf{r}_3 \\ \mathbf{r}_2 \end{pmatrix}$$

We can use the top portion to eliminate $\Delta\mathbf{x}$:

$$\Delta\mathbf{x} = \mathbf{D}^2\mathbf{A}^T\Delta\mathbf{y} + \mathbf{D}^2\left(\mathbf{r}_1 - \mathbf{X}^{-1}\mathbf{r}_3\right)$$

Substituting this in the bottom portion we obtain the *normal equations* :

$$\left(\mathbf{A}\mathbf{D}^2\mathbf{A}^T\right)\Delta\mathbf{y} = -\mathbf{r}_2 - \mathbf{A}\mathbf{D}^2\left(\mathbf{r}_1 - \mathbf{X}^{-1}\mathbf{r}_3\right)$$

Assuming **A** has a full rank, i.e. *rank*(**A**) = *m*, and let $\mathbf{B} = \mathbf{A}\mathbf{D}^2\mathbf{A}^T$, and $\mathbf{b} = -\mathbf{r}_2 - \mathbf{A}\mathbf{D}^2(\mathbf{r}_1 - \mathbf{X}^{-1}\mathbf{r}_3)$, the normal equations to be solved are:

$$\mathbf{B}\Delta\mathbf{y} = \mathbf{b} \qquad (1)$$

Note: **B** is *pd*. It is also sparse--if **A** is and *does not have a dense column*. Thus (1) can be solved by

- Sparse Cholesky Factorization --for large LPs (in all general-purpose solvers)
- Conjugate Gradient method ---for very large dense LPs

Recalling Cholesky Factorization

1. Find the Cholesky Factororization of *pd* **B** :

$$\mathbf{B} = \mathbf{L}\mathbf{L}^T \quad (\text{or } \mathbf{L}\mathbf{D}\mathbf{L}^T)$$

where **L** is lower traingular with $l_{ii} > 0$.

(If **B** is dense, this will cost about $O(n^3)$.)

2. Solve

$$\mathbf{L}\mathbf{z} = \mathbf{b} \text{ (forward substitution)}$$

$$\mathbf{L}^T\Delta\mathbf{y} = \mathbf{z} \text{ (backward substitution)}$$

Example of Cholesky Factorization

$$\mathbf{B} = \begin{pmatrix} 2 & -1 & & & -1 \\ -1 & 3 & -1 & -1 & \\ & -1 & 2 & -1 & \\ & -1 & -1 & 3 & -1 \\ -1 & & & -1 & 3 \end{pmatrix} \Rightarrow \begin{pmatrix} 2 & -1 & & & -1 \\ -1 & \frac{5}{2} & -1 & -1 & -\frac{1}{2} \\ & -1 & 2 & -1 & \\ & -1 & -1 & 3 & -1 \\ -1 & -\frac{1}{2} & & -1 & \frac{5}{2} \end{pmatrix} \Rightarrow \begin{pmatrix} 2 & -1 & & & -1 \\ -1 & \frac{5}{2} & -1 & -1 & -\frac{1}{2} \\ & -1 & \frac{8}{5} & -\frac{7}{5} & -\frac{1}{5} \\ & -1 & -\frac{7}{5} & \frac{13}{5} & -\frac{6}{5} \\ -1 & -\frac{1}{2} & -\frac{1}{5} & -\frac{6}{5} & \frac{5}{2} \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} 2 & -1 & & & -1 \\ -1 & \frac{5}{2} & -1 & -1 & -\frac{1}{2} \\ & -1 & \frac{8}{5} & -\frac{7}{5} & -\frac{1}{5} \\ & -1 & -\frac{7}{5} & \frac{11}{8} & -\frac{11}{8} \\ -1 & -\frac{1}{2} & -\frac{1}{5} & -\frac{11}{8} & \frac{19}{8} \end{pmatrix} \Rightarrow \begin{pmatrix} 2 & -1 & & & -1 \\ -1 & \frac{5}{2} & -1 & -1 & -\frac{1}{2} \\ & -1 & \frac{8}{5} & -\frac{7}{5} & -\frac{1}{5} \\ & -1 & -\frac{7}{5} & \frac{11}{8} & -\frac{11}{8} \\ -1 & -\frac{1}{2} & -\frac{1}{5} & -\frac{11}{8} & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 2 & -1 & & & -1 \\ -1 & \frac{5}{2} & -1 & -1 & -\frac{1}{2} \\ & -1 & \frac{8}{5} & -\frac{7}{5} & -\frac{1}{5} \\ & -1 & -\frac{7}{5} & \frac{11}{8} & -\frac{11}{8} \\ -1 & -\frac{1}{2} & -\frac{1}{5} & -\frac{11}{8} & 1 \end{pmatrix}$$

Bhat

Example of Cholesky Factorization

$$\mathbf{B} = \begin{pmatrix} 2 & -1 & & & -1 \\ -1 & 3 & -1 & -1 & \\ & -1 & 2 & -1 & \\ & -1 & -1 & 3 & -1 \\ -1 & & & -1 & 3 \end{pmatrix}$$

Note: For each column j, j=1,..,n

$L(i,j)$ = Bhat(i,j)/Bhat(j,j), for each $i = j+1,..,n$

$D(j,j)$ = Bhat(j,j)

$$= \begin{pmatrix} 1 & & & & \\ -\frac{1}{2} & 1 & & & \\ & -\frac{2}{5} & 1 & & \\ & -\frac{2}{5} & -\frac{7}{8} & 1 & \\ -\frac{1}{2} & -\frac{1}{5} & -\frac{1}{8} & -1 & 1 \end{pmatrix} \begin{pmatrix} 2 & & & & \\ & \frac{5}{2} & & & \\ & & \frac{8}{5} & & \\ & & & \frac{11}{8} & \\ & & & & 1 \end{pmatrix} \begin{pmatrix} 1 & -\frac{1}{2} & & & -\frac{1}{2} \\ & 1 & -\frac{2}{5} & -\frac{2}{5} & -\frac{1}{2} \\ & & 1 & -\frac{7}{8} & -\frac{1}{8} \\ & & & 1 & -1 \\ & & & & 1 \end{pmatrix}$$

L **D** **L**^{T}

Sparse Cholesky Factorization Method

If *pd* $\mathbf{B} = \mathbf{A}\mathbf{D}^2\mathbf{A}^T$ is sparse, then we can use sparse Cholesky :

1. Apply heuristics (e.g. minimum-degree heuristics) to do symmetric reordering of rows/columns to increase sparsity of $\mathbf{L}$ based on the sparsity pattern $\mathbf{B}$, which is the same as the sparsity pattern of $\mathbf{A}\mathbf{A}^T$

$$\mathbf{P}^T\mathbf{B}\mathbf{P} = \mathbf{L}\mathbf{D}\mathbf{L}^T \quad (\text{or } \mathbf{L}\mathbf{L}^T)$$

where $\mathbf{L}$ is a unit lower traingular with $l_{ii} = 1$.

(For sparse $\mathbf{B}$, this could cost about $O(n)$.)

2. Solve $\mathbf{L}\mathbf{z} = \mathbf{P}^T\mathbf{b}$ to get $\mathbf{z}$

   Solve $\mathbf{L}^T\bar{\mathbf{z}} = \mathbf{D}^{-1}\mathbf{z}$ to get $\bar{\mathbf{z}}$

   Set $\Delta\mathbf{y} = \mathbf{P}\bar{\mathbf{z}}$

Example of Sparse Cholesky Factorization

First we reorder (symmetrically permute) the row/column with the least number of nonzero elements (minimum degree) to the first row/column and pivot:

$$\begin{pmatrix} 2 & -1 & & & -1 \\ -1 & 3 & -1 & -1 & \\ & -1 & 2 & -1 & \\ & -1 & -1 & 3 & -1 \\ -1 & & & -1 & 3 \end{pmatrix} \Rightarrow \begin{pmatrix} 2 & -1 & & & -1 \\ -1 & \frac{5}{2} & -1 & -1 & -\frac{1}{2} \\ & -1 & 2 & -1 & \\ & -1 & -1 & 3 & -1 \\ -1 & -\frac{1}{2} & & -1 & \frac{5}{2} \end{pmatrix}$$

Next we repeat the above process on the remaining rows/columns (the bottom left block of the (red) lines): Here we see row/column 3 and row/column 5 having the lowest degree of 3. We select row/column 3 and permute them to the top/left of the remaining block (swapping row/column 2 and row/column 3 in this case). After we perform the pivoting:

$$\Rightarrow \begin{pmatrix} 2 & & -1 & & -1 \\ & 2 & -1 & -1 & \\ -1 & -1 & \frac{5}{2} & -1 & -\frac{1}{2} \\ & -1 & -1 & 3 & -1 \\ -1 & & -\frac{1}{2} & -1 & \frac{5}{2} \end{pmatrix} \Rightarrow \begin{pmatrix} 2 & & -1 & & -1 \\ & 2 & -1 & -1 & \\ -1 & -1 & 2 & -\frac{3}{2} & -\frac{1}{2} \\ & -1 & -\frac{3}{2} & \frac{5}{2} & -1 \\ -1 & & -\frac{1}{2} & -1 & \frac{5}{2} \end{pmatrix}$$

Example of Sparse Cholesky Factorization

CASE SCHOOL OF ENGINEERING

Now we repeat the process. In the remaining block (bottom left block below/left of the (red) lines. Since each row/column has the same degree 3, we don't need to do any permutation for now:

$$\Rightarrow \begin{pmatrix} 2 & & -1 & & -1 \\ & 2 & -1 & -1 & \\ -1 & -1 & 2 & -\frac{3}{2} & -\frac{1}{2} \\ & -1 & -\frac{3}{2} & \frac{5}{2} & -1 \\ -1 & & -\frac{1}{2} & -1 & \frac{5}{2} \end{pmatrix} \Rightarrow \begin{pmatrix} 2 & & -1 & & -1 \\ & 2 & -1 & -1 & \\ -1 & -1 & 2 & -\frac{3}{2} & -\frac{1}{2} \\ & -1 & -\frac{3}{2} & \frac{11}{8} & -\frac{11}{8} \\ -1 & & -\frac{1}{2} & -\frac{11}{8} & \frac{19}{8} \end{pmatrix} \Rightarrow \begin{pmatrix} 2 & & -1 & & -1 \\ & 2 & -1 & -1 & \\ -1 & -1 & 2 & -\frac{3}{2} & -\frac{1}{2} \\ & -1 & -\frac{3}{2} & \frac{11}{8} & -\frac{11}{8} \\ -1 & & -\frac{1}{2} & -\frac{11}{8} & 1 \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} 2 & & -1 & & -1 \\ & 2 & -1 & -1 & \\ -1 & -1 & 2 & -\frac{3}{2} & -\frac{1}{2} \\ & -1 & -\frac{3}{2} & \frac{11}{8} & -\frac{11}{8} \\ -1 & & -\frac{1}{2} & -\frac{11}{8} & 1 \end{pmatrix}$$

Example of Sparse Cholesky Factorization

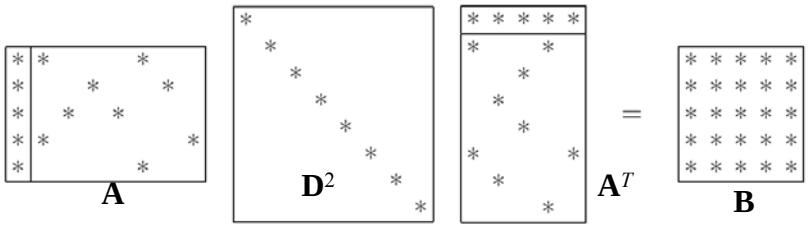
CASE SCHOOL OF ENGINEERING

$$\underbrace{\begin{pmatrix} 1 & & & & \\ & & 1 & & \\ & 1 & & & \\ & & & 1 & \\ & & & & 1 \end{pmatrix}}_{\mathbf{P}^T} \underbrace{\begin{pmatrix} 2 & -1 & & & -1 \\ -1 & 3 & -1 & -1 & \\ & -1 & 2 & -1 & \\ & -1 & -1 & 3 & -1 \\ -1 & & & -1 & 3 \end{pmatrix}}_{\mathbf{B}} \underbrace{\begin{pmatrix} 1 & & & & \\ & & & 1 & \\ & & 1 & & \\ & 1 & & & \\ & & & & 1 \end{pmatrix}}_{\mathbf{P}} = \begin{pmatrix} 2 & & -1 & & -1 \\ -1 & 2 & -1 & -1 & \\ & -1 & 3 & -1 & \\ & -1 & -1 & 3 & -1 \\ -1 & & & -1 & 3 \end{pmatrix}$$

$$= \underbrace{\begin{pmatrix} 1 & & & & \\ & 1 & & & \\ -\frac{1}{2} & -\frac{1}{2} & 1 & & \\ & -\frac{1}{2} & -\frac{3}{4} & 1 & \\ -\frac{1}{2} & & -\frac{1}{4} & -1 & 1 \end{pmatrix}}_{\mathbf{L}} \underbrace{\begin{pmatrix} 2 & & & & \\ & 2 & & & \\ & & 2 & & \\ & & & \frac{11}{8} & \\ & & & & 1 \end{pmatrix}}_{\mathbf{D}} \underbrace{\begin{pmatrix} 1 & -\frac{1}{2} & & -\frac{1}{2} & \\ & 1 & -\frac{1}{2} & -\frac{1}{2} & \\ & & 1 & -\frac{3}{4} & -\frac{1}{4} \\ & & & 1 & -1 \\ & & & & 1 \end{pmatrix}}_{\mathbf{L}^T}$$

What to do if B is dense?

It is known that if **A** has a dense column, then $\mathbf{B} = \mathbf{A}\mathbf{D}^2\mathbf{A}^T$ would be dense as illustrated below:



In this case, the sparse Cholesky would be worst than the regular Cholesky factorization. Also in this case, it may be better to work with the *augmented system* and use the symmetric indefinite factorization method, even though it may be less stable numerically (practical experience has shown that such a scheme works well).

Conjugate Gradients for Very Large $\mathbf{B}\Delta\mathbf{y} = \mathbf{b}$

This is just an application of the CG method to minimize the following strictly convex quadratic function:

$$\min\ 0.5\Delta\mathbf{y}^T\mathbf{B}\Delta\mathbf{y} - \mathbf{b}^T\Delta\mathbf{y} \qquad \mathbf{B} = \mathbf{A}\mathbf{D}^2\mathbf{A}^T$$

- Select an initial $\Delta\mathbf{y}^{(0)}=0$ and $\mathbf{d}^{(0)} = -\mathbf{g}^{(0)}= \mathbf{b}$, $\mu_0 = \mathbf{b}^T\mathbf{b}$. Set $k = 0$.
- While $k \le n$-1 (at this point we have $\mathbf{d}^{(k)}$ and $\mathbf{g}^{(k)}$)
  - If $\| \mathbf{g}^{(k)}\| < \varepsilon(1+\|\mathbf{b}\|)$, a solution Δy^* has been found.
  - Otherwise, compute

    $\mathbf{q}^{(k)} = \mathbf{B}\mathbf{d}^{(k)}$, $r_k = (\mathbf{d}^{(k)})^T\mathbf{q}^{(k)}$, $\mu_k = (\mathbf{g}^{(k)})^T\mathbf{g}^{(k)}$, and $\alpha_k = \mu_k/r_k$
  - Update $\Delta\mathbf{y}^{(k+1)} = \Delta\mathbf{y}^{(k)} + \alpha_k\mathbf{d}^{(k)}$ and $\mathbf{g}^{(k+1)} = \mathbf{g}^{(k)}+\alpha_k\mathbf{q}^{(k)}$
  - Compute $\beta_k = (\mathbf{g}^{(k+1)})^T\mathbf{q}^{(k)}/r_k$ and update $\mathbf{d}^{(k+1)} = -\mathbf{g}^{(k+1)}+ \beta_k\mathbf{d}^{(k)}$
  - Set $k = k+1$, and repeat.

Conjugate Gradients for Very Large $B\Delta y = b$

- Due to the Quadratic Convergent property, the iterative process should not take more than n iteration to find the unique solution $\Delta \mathbf{y}^*$.
- Require much less storage than the Cholesky Methods
- Faster than Cholesky for very large $\mathbf{B} = \mathbf{A}\mathbf{D}^2\mathbf{A}^T$, if $\mathbf{B}\mathbf{d}^{(k)}$ is cheap (i.e. $\mathbf{A}^T\mathbf{d}^{(k)}$ and $\mathbf{A}(\mathbf{D}^2\,\mathbf{A}^T\mathbf{d}^{(k)})$ are cheap, which is true if $\mathbf{A}$ is sparse).
- However, may be *less accurate* and *less robust*

Available Codes for Interior-Point Method

- A free MATLAB-based software

  LIPSOL: Linear Programming Interior Point Solver

  is available from

  http://www.caam.rice.edu/~zhang/lipsol/
- Most commercial software (e.g. CPLEX) now have versions of interior-point solvers
- Codes are easier to write than the Simplex method. So try it.