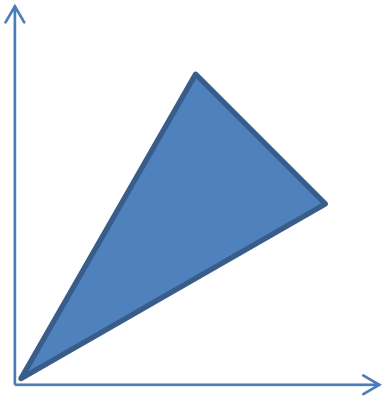
Conic Optimization and Application Algorithm Final Review

Yinyu Ye
Department of Management Science and Engineering
Stanford University
Stanford, CA 94305, U.S.A.

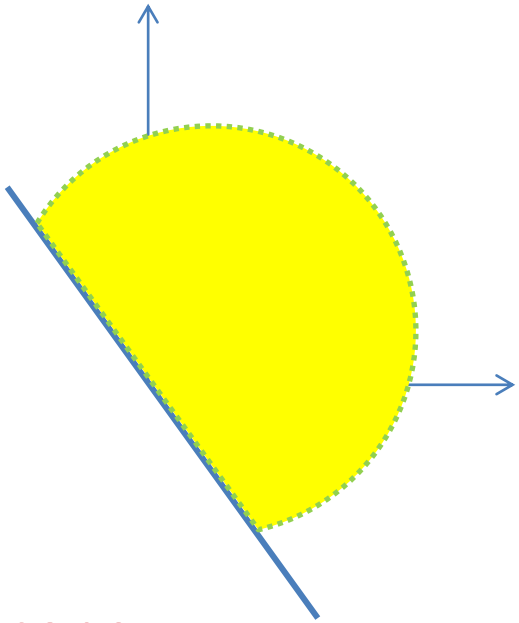
http://www.stanford.edu/~yyye
http://www.stanford.edu/class/msande211/
Chapter 14.1-14.2, Chapter6

Conic Linear Programming

$$(\text{LP}) \quad \min \; c^T x$$
$$\text{s.t.} \; Ax = b,$$
$$x \geq 0$$



$$(\text{CLP}) \quad \min \; c^T x$$
$$\text{s.t.} \; Ax = b,$$
$$x \in C$$



C: a convex and self-dual pointed cone

CLP Examples

$$
\begin{aligned}
(\text{LP}) \quad \min \quad & ax_2 + bx_3 \\
\text{s.t.} \quad & x_2 + x_3 = 1, \\
& x_1 = 1, \\
& (x_1, x_2, x_3) \geq 0
\end{aligned}
$$

Nonnegative Orthant Cone—
Linear or Polyhedral

$$
\begin{aligned}
(\text{SOCP}) \quad \min \quad & ax_2 + bx_3 \\
\text{s.t.} \quad & x_2 + x_3 = 1, \\
& x_1 = 1, \\
& x_1 \geq \sqrt{(x_2)^2 + (x_3)^2}
\end{aligned}
$$

Second-Order Cone -- Nonlinear

$$
\begin{aligned}
(\text{SDP}) \quad \min \quad & ax_2 + bx_3 \\
\text{s.t.} \quad & x_2 + x_3 = 1, \\
& x_1 = 1, \\
& \begin{pmatrix} x_1 & x_2 \\ x_2 & x_3 \end{pmatrix} \text{ is PSD}
\end{aligned}
$$

Positive Semidefinite Cone – Nonlinear

Dual of Conic Linear Programming

$$(\text{LP}) \quad \min \; c^T x \quad \text{s.t.} \; Ax = b, \; x \geq 0$$

$$\Rightarrow$$

$$(\text{LD}) \quad \max \; b^T y \quad \text{s.t.} \; A^T y + r = c, \; r \geq 0$$

$$(\text{CLP}) \quad \min \; c^T x \quad \text{s.t.} \; Ax = b, \; x \in C$$

$$\Rightarrow$$

$$(\text{CLD}) \quad \max \; b^T y \quad \text{s.t.} \; A^T y + r = c, \; r \in C$$

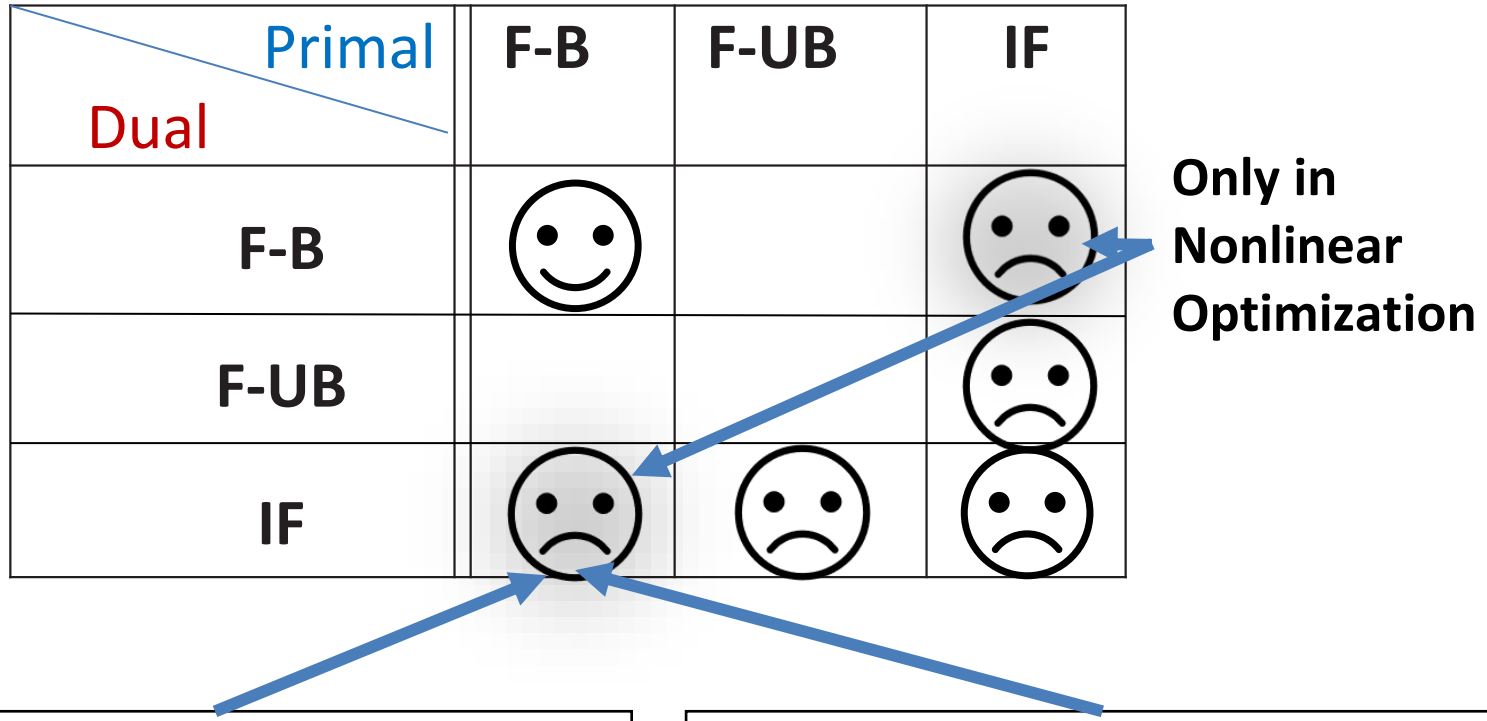
Duality Theorems

Theorem 1 *(Weak duality theorem) Let both primal feasible region F_p and dual feasible region F_d be non-empty. Then,* $\mathbf{c}^T\mathbf{x} \geq \mathbf{b}^T\mathbf{y}$ *for all* $\mathbf{x} \in F_p$, $\mathbf{y} \in F_d$.

This theorem shows that a feasible solution to either problem yields a bound or certificate on the value of the other problem. We again call it the duality gap. Note that the standard KKT condition may fail to establish such certification

Theorem 2 *(Strong duality theorem) Let both primal feasible region F_p and dual feasible region F_d be non-empty and have interior. Then, $\mathbf{x}^* \in F_p$ is optimal for (CLP) and $\mathbf{y}^* \in F_d$ is optimal for (CLD) if and only if the duality gap $\mathbf{c}^T\mathbf{x}^* - \mathbf{b}^T\mathbf{y}^* = 0$.*

CLP Primal-Dual Combinations



(SDP) $\min \quad x_1$

$\text{s.t.} \quad x_2 = 1,$

$\begin{pmatrix} x_1 & x_2 \\ x_2 & x_3 \end{pmatrix}$ is PSD

(SCD) $\max \quad y$

$\text{s.t.} \quad r_1 = 1,$

$y + r_2 = 0,$

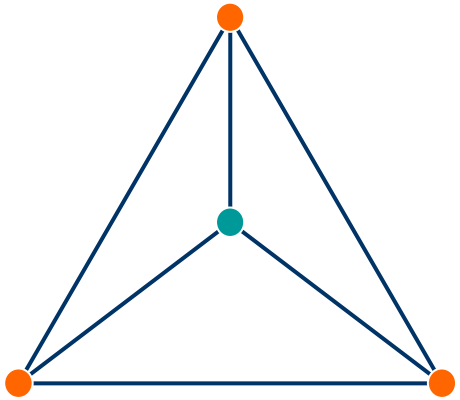
$r_3 = 0,$

$\begin{pmatrix} r_1 & r_2 \\ r_2 & r_3 \end{pmatrix}$ is PSD

Facility Location and its Conic Formulation

Find the facility location that is "close" to the client locations.

$$\min_{y} \ \sum_{i} \left|\left|\boldsymbol{a}_i - \boldsymbol{y}\right|\right|, \quad or$$
$$\min_{y} \ max_{i} \left\{\left|\left|\boldsymbol{a}_i - \boldsymbol{y}\right|\right|\right\}.$$



$$\min_{w_i,y} \ \sum_{i} w_i$$
$$\text{s.t.} \quad \left|\left|\boldsymbol{a}_i - \boldsymbol{y}\right|\right| - \boldsymbol{w}_i \leq \boldsymbol{0}, \ i = 1, \ldots, k \quad or$$

$$\min_{w,y} \ w$$
$$\text{s.t.} \quad \left|\left|\boldsymbol{a}_i - \boldsymbol{y}\right|\right| - \boldsymbol{w} \leq \boldsymbol{0}, \ i = 1, \ldots, k$$

$$(w; \boldsymbol{a}_i - \boldsymbol{y}) \in \boldsymbol{SOC}, \ i = 1, \ldots, k$$

The constraints can be written in conic format.

The optimal dual multipliers can be interpreted as force on the edges.

Sensor Network Localization Statement

Given a graph $G = (V, E)$ and sets of partial distance measurements, say $\{d_{ij} : (i, j) \in E\}$, the goal is to compute a realization of G in the Euclidean space $\mathbf{R}^d$ for a given low dimension d, i.e.

- to place the nodes/vertices of G in $\mathbf{R}^d$ such that
- the Euclidean distance between every pair of adjacent vertices $(i, j) \in E$ equals the measurements $d_{ij} \in E$.

In general the localization may not be fixed since the configuration can rotate and translate. Thus, we assume that the positions of (d+1) sensors are known, and they called anchors.

This problem has wide applications …

Sensor Network Localization Formulation

Find d-dimensional points/vectors $\mathbf{x}_j$, $j=1,2,\ldots,n+d+1$, such that

$$\| x_i - x_j \| = d_{ij} \quad \forall (i,j) \in E$$
$$\text{Anchors}: \quad x_i = a_i, \quad i = 1,2,\ldots,d+1$$

This is a system of quadratic equations (after square both sides) and nonconvex, in contrast to a system of linear equations.

Does the system have a solution/localization of all $\mathbf{x}_j$'s? Is the solution/localization unique? Is there a certification for a solution to make it reliable or trustworthy? Is the system partially localizable with certification?

To get something more tractable, we can consider convex relaxations.

SOCP Relaxation

SOCP: Find d-dimensional vectors $\mathbf{x}_j$, $j=1,2,\ldots,n$, such that

$$\| x_i - x_j \| \leq d_{ij} \quad \forall (i,j) \in E$$
$$x_i = a_i, \quad i = 1,2,\ldots,m$$

This is to find a feasible solution for an SOCP problem and it is a convex feasibility problem, called SOCP relaxation.

$$\| x_i - x_4 \| \leq d_{i4} \quad \forall (i,4) \in E, i = 1,2,3$$
$$x_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, x_2 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, x_3 = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

An important and interesting question is: when is the relaxation exact?
Answer: when the target point is inside the convex hull of anchors.

SDP Relaxation I

The problem can be written as follows:

$$\|x_i - x_j\|^2 = d_{ij}^2 \qquad (i,j) \in E_{ss}$$

$$\|a_k - x_j\|^2 = \bar{d}_{kj}^2 \qquad (k,j) \in E_{sa}$$

$$x_i \in R^d$$

$\{a_k\}$ are the positions of "anchors".

Step 1: Linearize

$$\|x_i - x_j\|^2 = x_i^T x_i - 2x_i^T x_j + x_j^T x_j$$

$$Y_{ii} \qquad Y_{ij} \qquad Y_{jj}$$

$$\|a_k - x_j\|^2 = a_k^T a_k - 2a_k^T x_j + x_j^T x_j$$

$$Y_{jj}$$

SDP Relaxation II

Step 2: Tighten from relation of Y and $X=[x_1\ x_2\ \ldots\ x_n]$:

$$Y \succ X^T X \Leftrightarrow Z = \begin{bmatrix} I & X \\ X^T & Y \end{bmatrix} \succ 0$$

Step 3: Use linear algebra trick and put things together:

$$\begin{aligned}
&(0; e_i - e_j)(0; e_i - e_j)^T \bullet Z = d_{ij}^2 && (i, j) \in E_{ss} \\
&(a_k; -e_j)(a_k; -e_j)^T \bullet Z = \bar{d}_{kj}^2 && (k, j) \in E_{sa} \\
&Z_{1:d,1:d} = I_d, && Z \succ 0;
\end{aligned}$$

where e_i is the unit vector of zeros everywhere but 1 for the ith entry.

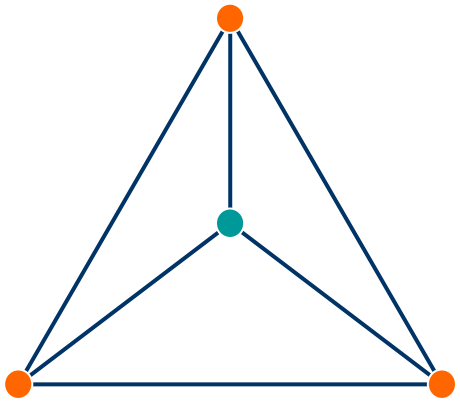
This is an instance of semidefinite programming (SDP) and it is convex.

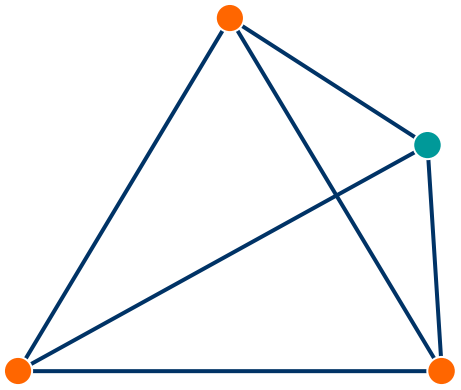
SDP Relaxation III

An important and interesting question is: when is the relaxation exact?

Answer: The relaxation is exact iff the input data satisfies a uniqueness property called Universal d-Rigidity: The input has a unique realization in R^d, and does not have any non-trivial realization in R^h for h>d.

E.g., SDP is always exact with one target and three anchors in R^2.





The optimal dual multipliers can be interpreted as tension on the edges.

SDP Relaxation IV

SDP: When the distance measurements have noise, one can develop an SDP relaxation to minimize an error function in the form

$$\begin{array}{ll} \min & C \bullet Z \\ \text{s.t.} & A_{ij} \bullet Z = (d_{ij})^2, (i,j) \in E, \\ & Z \succcurlyeq 0 \end{array}$$

Interior-Point Algorithms have been extended for SDP using the barrier function regularization: *–log(det(Z))*

$$\begin{array}{ll} \min & C \bullet Z - \mu \log(\det(Z)) \\ \text{s.t.} & A_{ij} \bullet Z = (d_{ij})^2, (i,j) \in E \end{array}$$

Algorithm/Method: facts of optimization

An optimization problem falls in one of three cases:

- Problem is *infeasible*: Feasible region is empty.
- Problem is *unbounded*: Feasible region is non-empty and the objective value is unbounded.
- Problem is *feasible* and *bounded*.

When the problem is feasible and bounded,

- There may be an optimal solution or it may not attainable
  - min e^{-x}, s.t. $x \geq 0$
- Optimal point may be unique or not (alternative optima)
- Optimal point may be on the boundary or inside of the feasible region.

General Feasible-Descent Direction Methods

Starting with a feasible solution $\mathbf{x}^k$, we consider an iterative scheme of the form

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{d}^k$$

where $\mathbf{d}^k$ is a search direction vector, both feasible and descent, and scalar α_k is again the step-size.

In fact, once the search direction is chosen, the objective function can be written as $f(\mathbf{x}^k + \alpha \mathbf{d}^k)$, which is just function of α. Thus one can choose step-size α as α_k according to some line (one-variable optimization) search to minimize the function WHILE keep $\mathbf{x}^k + \alpha_k \mathbf{d}^k$ feasible.

The Optimization Algorithms/Methods

- Univariate Optimization Problem:
  - The Bisection (first-order) method for finding a root the derivative function of a univariate problem
  - The Golden-Section (zero-order) method for finding a minimizer of unimodal function of one variable
- Unconstrained Multivariate Optimization Problem (UCOP):
  - The Gradient or Steepest Descent (SDM, first-order) method
    - It converges as long as the level set of the function is bounded. The converges speed depends on the shape of the level set and the starting point.
  - The Newton method (second-order, also applicable for equality constrained optimization (ECOP))
    - It may not converge if the starting point is far from the root point, but super fast if it is "close" to the target.
- General Constrained Multivariate Optimization Problem (GCOP):
  - The Reduced-Gradient Method (RGM): Simplex Method for LP
  - The Gradient-Projection Methods (first-order, for linear equality or nonnegative constraints)
  - The "Path-Following" Methods (sequential Newton, second-order)
  - The Augmented Lagrangian and ADMM methods for constrained optimization

Summary of the Methods for UCOP

1. Take an initial point $\boldsymbol{x}^0$. Set $k = 0$.
2. Evaluate $\nabla f(\boldsymbol{x}^k)$. If $|\nabla f(\boldsymbol{x}^k)| \leq \varepsilon$ where ε is a given tolerance, stop.
3. Update $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k$.

   set $k=k+1$ and go to 2.

* SDM: $\boldsymbol{d}^k = -\nabla f(\boldsymbol{x}^k)$ (works globally, but slow)

Newton: $\boldsymbol{d}^k = -\nabla^2 f(\boldsymbol{x}^k)^{-1} \nabla f(\boldsymbol{x}^k)$ (works locally, but fast)

Quasi-Newton: $\boldsymbol{d}^k = -Q \nabla f(\boldsymbol{x}^k)$ (combined)

where Q^k is a PD matrix

** Fix a step-size α_k according to the Lipchitz constant or use the variable step-size such as one-dimension search to find α_k that minimizes

$$\varphi(\alpha) = f(\boldsymbol{x}^k + \alpha \mathbf{d}^k).$$

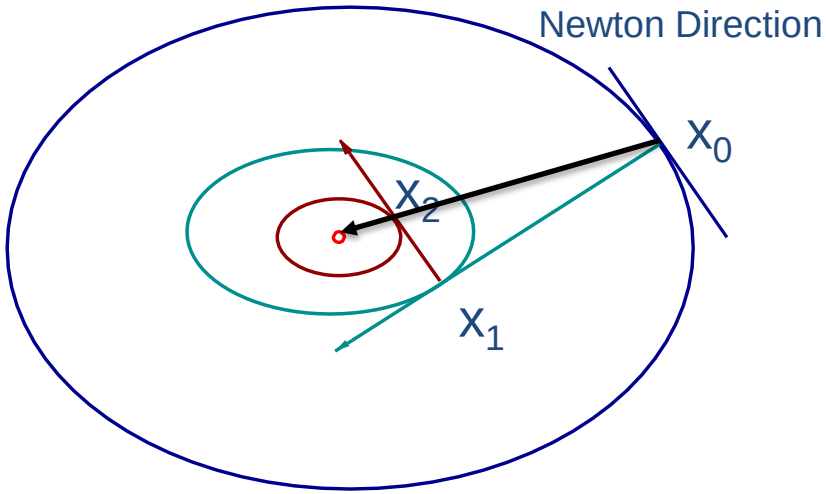
Illustration of the Steepest Descent Method (SDM)

The SDM chooses $\mathbf{d}^k = -\nabla f(\mathbf{x}^k)$ as the search direction at each step and selects step-size $\alpha_k = \arg\min f(\mathbf{x}^k + \alpha \mathbf{d}^k)$.

Then the new iterate is defined as $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{d}^k$.

QP Example: Let $f(\mathbf{x}) = \mathbf{c}^T\mathbf{x} + 0.5\mathbf{x}^T Q\mathbf{x}$ where $Q \in \mathbf{R}^{n\times n}$ is symmetric and positive definite. Then, $\nabla f(\mathbf{x}^k) = \mathbf{c} + Q\mathbf{x}^k$, and the step size has a close form formula

$$\alpha_k = -\frac{(\mathbf{c}^T + (\mathbf{x}^k)^T Q)\mathbf{d}^k}{(\mathbf{d}^k)^T Q\mathbf{d}^k} = \frac{(\mathbf{d}^k)^T\mathbf{d}^k}{(\mathbf{d}^k)^T Q\mathbf{d}^k}$$



For quadratic minimization with nonsingular Hessian, the Newton method with unit step-size converges in one step to the KKT solution.

Newton's Method for ECOP

min $f(\boldsymbol{x})$

s.t. $\boldsymbol{Ax-b=0}$

The KKT Equations and its Jacobian Matrix:

$$\mathbf{g}(x,y)=\begin{pmatrix}\nabla f(x)-A^T y\\ Ax-b\end{pmatrix}(=0)$$

$$\nabla\mathbf{g}(x,y)=\begin{pmatrix}\nabla^2 f(x) & -A^T\\ A & 0\end{pmatrix}$$

$$\begin{pmatrix}x^{k+1}\\ y^{k+1}\end{pmatrix}=\begin{pmatrix}x^{k}\\ y^{k}\end{pmatrix}-\nabla\mathbf{g}(x^k,y^k)^{-1}\mathbf{g}(x^k,y^k)$$

Newton Direction Vector

Description of the LP Simplex Method

1. Initialize: with a minimization problem with respect to a BFS with basis index set B and let N denote the rest index set:

$$\mathbf{x}_B = (A_B)^{-1}\mathbf{b} (\geq \mathbf{0}),\ \mathbf{x}_N = \mathbf{0}$$

2. Pricing: Compute the corresponding shadow-dual price vector **y** and the reduced vector **r**:

$$\mathbf{y}^T = \mathbf{c}^T_B (A_B)^{-1} \text{ or solve } \mathbf{y}^T A_B = \mathbf{c}^T_B \text{ ,then let } \mathbf{r} = \mathbf{c}^T - \mathbf{y}^T A$$

and find (Dantzig rule): $r_e = \min_{j \in N} \{r_j\}$. (break ties arbitrarily)

3. Test of Termination: If $r_e \geq 0$, Stop -- the solution is already optimal. Otherwise let x_e be the incoming and determine whether the vector $(A_B)^{-1}A_e$ contains a positive entry. If not, the objective function is unbounded below -- Stop.

4. Step Sizing: Perform the Min-Ratio-Test to determine the step size:

$$\alpha = \min\{\ \mathbf{x}_B \,./ [(A_B)^{-1}A_e]_+\ \}.$$

5. Basis Update: Set $x_e = \alpha$ and elect a current basic variables with zero value (break ties arbitrarily) be the outgoing; so that we reach a new (adjacent) BFS – Go To Step 1.

Theorem: If the reduced cost coefficient is positive for every nonbasic variable, then the optimal BFS is unique.

GCOP: A Gradient Projection Method I

$$\min \; f(\boldsymbol{x}) \qquad \text{s.t.} \; \boldsymbol{X \geq 0}$$

Project the Gradient-Step solution on to the feasible region:

$$\mathbf{x}^{k+1} = \max\{ \boldsymbol{0},\ \mathbf{x}^k - \beta^{-1} \nabla f(\mathbf{x}^k) \}$$

$$\boldsymbol{y}^{k+1} = \max\{ \boldsymbol{0},\ \nabla f(\mathbf{x}^k) \} \text{ (dual)}$$

The step-size could be chosen to minimize the function value. The method works for other simple bound constraints and it needs to start from a feasible solution.

The Gradient-Projection Method II

$$\min \ f(\boldsymbol{x})$$
$$\text{s.t.} \ \boldsymbol{Ax-b=0}$$

Project the Gradient on to the null space of the constrained matrix so that it is both feasible and descent direction

$$\mathbf{d}^k=-(I-A^T(AA^T)^{-1}A)\nabla f(\mathbf{x}^k)$$

or

$$y^k=(AA^T)^{-1}A\nabla f(\mathbf{x}^k)$$
$$\mathbf{d}^k=-(\nabla f(\mathbf{x}^k)-A^Ty^k)$$

The step-size would be chosen to minimize the function value or fixed according to the Lipschitz constant.

Need to start from a feasible solution.

UCOP: The Path-Following Method I

$\min \quad f(\mathbf{x})$ ⟹ $\min \quad f_\mu(\mathbf{x}) := f(\mathbf{x}) + 0.5\mu \|\mathbf{x}\|^2$

We start from a good approximate of $\mathbf{x}(\mu^k)$ then we reduce μ^k to μ^{k+1} by a factor such that this good approximate is still close to $\mathbf{x}(\mu^{k+1})$. Then we apply the Newton method with this good approximate of $\mathbf{x}(\mu^k)$ as the initial solution so that one Newton step would produce a good approximate of $\mathbf{x}(\mu^{k+1})$. Then the process repeat where μ is fixed in each inner problem.

The KKT equations:

$$\nabla f(\mathbf{x}) + \mu\, \mathbf{x} = \mathbf{0}$$

At the kth step, we aim to find a good approximate of $\mathbf{x}$ to satisfy

$$\nabla f(\mathbf{x}) + \mu^{k+1}\mathbf{x} = \mathbf{0}$$

We start from $\mathbf{x}^k$ and apply the Newton iteration: compute direction vector $\boldsymbol{d}$ from

$$(\nabla^2 f(\mathbf{x}^k) + \mu^{k+1}\boldsymbol{I})\mathbf{d} = -\nabla f(\mathbf{x}^k) - \mu^{k+1}\mathbf{x}^k$$

then let $\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{d}$.

GCOP: The Path-Following Method II

The same idea can be applied to the inequality constrained optimization problem by constructing a "barriered" objective function with a fixed parameter $\mu > 0$.

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & A\mathbf{x}=\mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

(BOP)

$$\begin{aligned} \min \quad & f_\mu(x) := f(x) - \mu \sum_{j=1}^{n} \ln(x_j) \\ \text{s.t.} \quad & Ax = b, (x > 0) \end{aligned}$$

Denote by $\boldsymbol{x}(\mu)$, the minimizer of the barriered problem, and consider μ reduces from ∞ to zero. Then, $\boldsymbol{x}(\mu)$ form a continuous path strictly inside of the feasible region that leads to an optimal solution of the original problem from the analytic center. This path is called the central path. For simple problem one can use the KKT conditions to construct $\boldsymbol{x}(\mu)$ explicitly.

For fixed μ, the problem has only equality constraints (ECOP) so that the Newton method is applicable. Needs to start from a strictly feasible solution.

Comments on Path-Following Method

In both path-following cases I and II, denote by $\boldsymbol{x}(\mu)$, the minimizer of the parametrized problem, and consider μ reduces from ∞ to zero. Then, $\boldsymbol{x}(\mu)$ form a continuous path that leads to an optimal solution of the original problem.

Numerically, we start from a good approximate of $\mathbf{x}(\mu^k)$ then we reduce μ^k to μ^{k+1} such that this good approximate is still close to $\mathbf{x}(\mu^{k+1})$. Then we apply the Newton method with this good approximate of $\mathbf{x}(\mu^k)$ as the initial solution so that one Newton step would produce a good approximate of $\mathbf{x}(\mu^{k+1})$. Then the process repeat where μ is fixed in each inner problem.

In each inner problem, we solve an unconstrained or equality-constrained-only optimization problem so that the Newton method or any other method are applicable. For example, first minimize $f_1(\mathbf{x})$ then, $f_{1/2}(\mathbf{x})$, then $f_{1/4}(\mathbf{x})$... where each problem needs only be solved approximately. They are typically difficult to be implemented.

Interior-Point Methods: Primal-Dual Potential Reduction

Another approach is to reduce the primal-dual potential function (that is parameter-free) and able to take large step size. One can apply the first or second order method to minimize the potential function

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x}=\mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

$$\begin{aligned} \max \quad & \mathbf{b}^T\mathbf{y} \\ \text{s.t.} \quad & A\mathbf{y}+\mathbf{r}=\mathbf{c}, \\ & \mathbf{r} \geq \mathbf{0} \end{aligned}$$

$$\begin{aligned} \min \quad & (2n)ln(c^Tx - b^Ty) + \sum_{j=1}^{n} \ln(x_j r_j) \\ \text{s.t.} \quad & Ax = b, (x>0) \\ & A^Ty + r = c (r > 0) \end{aligned}$$

The upper potential level set contains the optimal solution set for both the primal and dual. Starting from primal and dual interior solution $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{r}^0)$, the potential function can be reduced by a constant in each Newton step resulting a linear/geometric convergent algorithm.

(Matlab Demo LP and Indefinite QP.)

Augmented Lagrangian Method (ALM)

$$\begin{aligned} \min\ & f(\boldsymbol{x}) \\ \text{s.t.}\ & \boldsymbol{Ax-b = 0} \\ & \boldsymbol{X \geq 0} \end{aligned}$$

Augmented Lagrange Function

$L_A(\mathbf{x},\mathbf{y}) = f(\boldsymbol{x}) - (A\boldsymbol{x} - \boldsymbol{b})^T\boldsymbol{y} + 0.5\beta||A\boldsymbol{x}-\boldsymbol{b}||^2$, $\boldsymbol{x} \geq \mathbf{0}$

Augmented Largrangian Method:

$$\begin{aligned} x^{k+1} &= \arg\min_{x\geq 0} L_A(x, y^k); \\ y^{k+1} &= y^k - \beta\cdot(Ax^{k+1} - b) \end{aligned}$$

Theorem: Let $f(\boldsymbol{x})$ be a convex function. Then the dual objective function $\phi(\boldsymbol{y})$ is Lipschitz $1/\beta$ and $\nabla\phi(\boldsymbol{y}^k) = -(\boldsymbol{Ax}^{k+1} - \boldsymbol{b})$.

It is an implicit dual steepest-ascent algorithm, and the algorithm complexity is an $O(1/\varepsilon)$ for the case that $f(\boldsymbol{x})$ is a convex function. It does not need to start from a feasible solution.

Alternating Direction Method with Multipliers (ADMM)

$$\begin{aligned} \min \quad & f_1(\boldsymbol{x}_1)+f_2(\boldsymbol{x}_2) \\ \text{s.t.} \quad & \boldsymbol{A}_1\boldsymbol{x}_1+\boldsymbol{A}_2\boldsymbol{x}_2-\boldsymbol{b}=\boldsymbol{0} \\ & \boldsymbol{x}_2 \geq \boldsymbol{0} \end{aligned}$$

Augmented Lagrange Function

$$L_A(\mathbf{x}_1,\mathbf{x}_2\geq\mathbf{0},\mathbf{y}) = f_1(\boldsymbol{x}_1)+f_2(\boldsymbol{x}_2)-(A_1\boldsymbol{x}_1+A_2\boldsymbol{x}_2-\boldsymbol{b})^T\boldsymbol{y}+0.5\beta||A_1\boldsymbol{x}_1+A_2\boldsymbol{x}_2-\boldsymbol{b}||^2$$

ADMM:

$$\begin{aligned} x_1^{k+1} &= \arg\min \quad L_A(x_1, x_2^k, y^k); \\ x_2^{k+1} &= \arg\min_{x_2\geq 0} L_A(x_1^{k+1}, x_2, y^k); \\ y^{k+1} &= y^k - \beta\cdot(A_1x^{k+1} + A_2x_2^{k+1} - b) \end{aligned}$$

The algorithm complexity is an $O(1/\varepsilon)$ for the case that $f_1(\boldsymbol{x}_1)+f_2(\boldsymbol{x}_2)$ are convex functions. By choosing $\mathbf{x}_1$ and $\mathbf{x}_2$ suitably, the minimizers may have a simple close-form expressions – easy to be implemented.

Final Comments About Algorithms

- All of the algorithms we have discussed can terminate at an KKT point for nonconvex optimization, each of them has pros and cons…
- Thus, at termination they are only guaranteed to have found an optimal solution in the case of convex optimization.
- Randomization sometime can help!
- So how do we find a better solution in the case of Non-convex Optimization?
  - Generally, we simply just rerun the algorithm starting at a number of different starting points.
  - Enumerate all KKT points, or perturb the KKT solution to get out of the "local trap"
  - Convex reformulation or relaxations
- More advanced optimization course, CME307/MS&E311 etc.