First-Order Constrained Optimization Algorithms

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

Descent Pr
Feasible

http://www.stanford.edu/~yyye

Chapters 4.2, 8.4-5, 9.1-7, 12.3-6

First-Order Algorithms for Conic Constrained Optimization (CCO)

Consider the conic nonlinear optimization problem: $\min\ f(\mathbf{x})$ s.t. $\mathbf{x} \in K$.

$x \succeq 0$

$[x, \nabla f(x)]$ ≥ 0 ≥ 0

- Nonnegative Linear Regression: given data $A \in R^{m\times n}$ and $\mathbf{b} \in R^m$

$$\min\ f(\mathbf{x}) = \frac{1}{2}\|A\mathbf{x} - \mathbf{b}\|^2 \text{ s.t. } \mathbf{x} \geq \mathbf{0}; \quad \text{where } \nabla f(\mathbf{x}) = A^T(A\mathbf{x} - \mathbf{b}).$$

$x_i = u_i^2$

- Semidefinite Linear Regression: given data $A_i \in S^n$ for $i = 1, ..., m$ and $\mathbf{b} \in R^m$

SNL

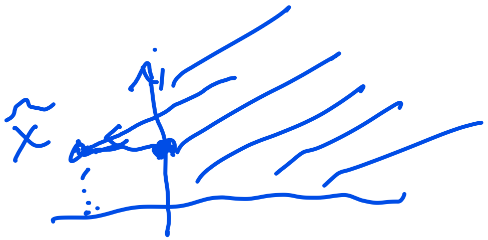
$$\min\ f(X) = \frac{1}{2}\|\mathcal{A}X - \mathbf{b}\|^2 \text{ s.t. } X \succeq \mathbf{0}; \quad \text{where } \nabla f(X) = \mathcal{A}^T(\mathcal{A}X - \mathbf{b}).$$

(UV^T) R^m

$X = UV^T$ $\quad$ $X = UU^T \succeq 0$ $\quad$ $n \times r$

s.t. $U = V$

$$\mathcal{A}X = \begin{pmatrix} A_1 \bullet X \\ \dots \\ A_m \bullet X \end{pmatrix} \quad \text{and} \quad \mathcal{A}^T\mathbf{y} = \sum_{i=1}^m y_i A_i.$$



Suppose we start from a feasible solution $\mathbf{x}^0$ or X^0.

descent-First

feasible-second

SDM Followed by the Conic-Region-Projection

- $\hat{\mathbf{x}}^{k+1} = \mathbf{x}^k - \frac{1}{\beta}\nabla f(\mathbf{x}^k)$
- $\mathbf{x}^{k+1} = \boxed{\text{Proj}_K(\hat{\mathbf{x}}^{k+1})}$: Solve $\min_{\mathbf{x}\in K} \|\mathbf{x} - \hat{\mathbf{x}}^{k+1}\|^2$.

$\|Z - \hat{X}\|_F$

For examples:

- if $K = \{\mathbf{x} : \mathbf{x} \geq \mathbf{0}\}$, then

$$\mathbf{x}^{k+1} = \text{Proj}_K(\hat{\mathbf{x}}^{k+1}) = \max\{\mathbf{0},\ \hat{\mathbf{x}}^{k+1}\}.$$

$Z = LDL^T$

- If $K = \{X : X \succeq \mathbf{0}\}$, then factorize $\hat{X}^{k+1} = \sum_{j=1}^n \lambda_j \mathbf{v}_j \mathbf{v}_j^T$ and let

$$X^{k+1} = \text{Proj}_K(\hat{X}^{k+1}) = \sum_{j:\lambda_j>0} \lambda_j \mathbf{v}_j \mathbf{v}_j^T.$$

(The drawback is that the total eigenvalue-factorization may be costly...)

Does the method converge? What is the convergence speed? See more details in HW3.

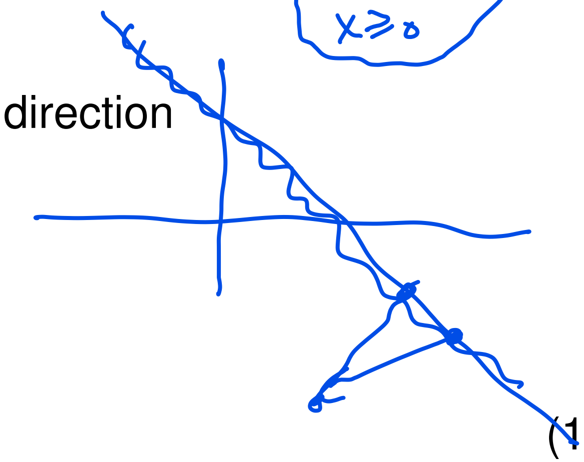
SDM Followed by the Convex-Region-Projection

Consider the convex-region-constrained nonlinear optimization problem: $\min \ f(\mathbf{x}) \quad \text{s.t.} \quad A\mathbf{x} = \mathbf{b}.$ that is $K = \{\mathbf{x} : \ A\mathbf{x} = \mathbf{b}\}$.

$Ax^0 = b$

$x \geq 0$

The projection method becomes, starting from a feasible solution $\mathbf{x}^0$ and let direction



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

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{d}^k; \tag{1}$$

where the stepsize can be chosen from line-search or again simply let

$$\alpha^k = \frac{1}{\beta}$$

and β is the (global) Lipschitz constant.

Does the method converge? What is the convergence speed? See more details in HW3.

min $f(x)$
s.t. $x \in K$

$\begin{bmatrix} -3 \\ 4 \\ 1 \end{bmatrix}$

SDM Followed by the Nonconvex-Region-Projection

$\begin{bmatrix} -3 \\ 4 \\ 0 \end{bmatrix}$

$d=2$

- $K \subset R^n$ whose support size is no more than $d(< n)$: $\mathbf{x} = \text{Proj}_K(\hat{\mathbf{x}})$ contains the largest d absolute entries of $\hat{\mathbf{x}}$ and set the rest of them to zeros.

- $K \subset R^n_+$ and its support size is no more than $d(< n)$: $\mathbf{x} = \text{Proj}_K(\hat{\mathbf{x}})$ contains the largest no more than d positive entries of $\hat{\mathbf{x}}$ and set the rest of them to zeros.

- $K \subset S^n$ whose rank is no more than $d(< n)$: factorize $\hat{X} = \sum_{j=1}^n \lambda_j \mathbf{v}_j \mathbf{v}_j^T$ with $|\lambda_1| \ge |\lambda_2| \ge \ldots \ge |\lambda_n|$ then $\text{Proj}_K(\hat{X}) = \sum_{j=1}^d \lambda_j \mathbf{v}_j \mathbf{v}_j^T$.

- $K \subset S^n_+$ whose rank is no more than $d(< n)$: factorize $\hat{X} = \sum_{j=1}^n \lambda_j \mathbf{v}_j \mathbf{v}_j^T$ with $\lambda_1 \ge \lambda_2 \ge \ldots \ge \lambda_n$ then $\text{Proj}_K(\hat{X}) = \sum_{j=1}^d \max\{0, \lambda_j\} \mathbf{v}_j \mathbf{v}_j^T$.

Does the method converge? What is the convergence speed? What if $f(\cdot)$ is not a convex function?

$x^{k+1} = x^k \cdot d$
$x_i^k \cdot d_i$

$x^{k+1} = x^k - \alpha \nabla f(x^k)$

Multiplicative-Update I: “Mirror” SDM for CCO

$\min f(x)$

$\text{s.t. } x \geq 0$

$x^0 > 0$

At the kth iterate with $\mathbf{x}^k > \mathbf{0}$:

$$\boxed{\mathbf{x}^{k+1} = \mathbf{x}^k . * \exp(-\frac{1}{\beta}\nabla f(\mathbf{x}^k))}$$

$\log x^{k+1} = \log x^k - \frac{1}{\beta}\nabla f(x^k)$

Note that $\mathbf{x}^{k+1}$ remains positive in the updating process.

The classical Projected SDM update can be viewed as

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}\geq\mathbf{0}} \nabla f(\mathbf{x}^k)^T\mathbf{x} + \frac{\beta}{2}\|\mathbf{x}-\mathbf{x}^k\|^2.$$

One can choose any strongly convex function $h(\cdot)$ and define

$$\mathcal{D}_h(\mathbf{x},\mathbf{y}) = h(\mathbf{x}) - h(\mathbf{y}) - \nabla h(\mathbf{y})^T(\mathbf{x}-\mathbf{y})$$

and define the update as

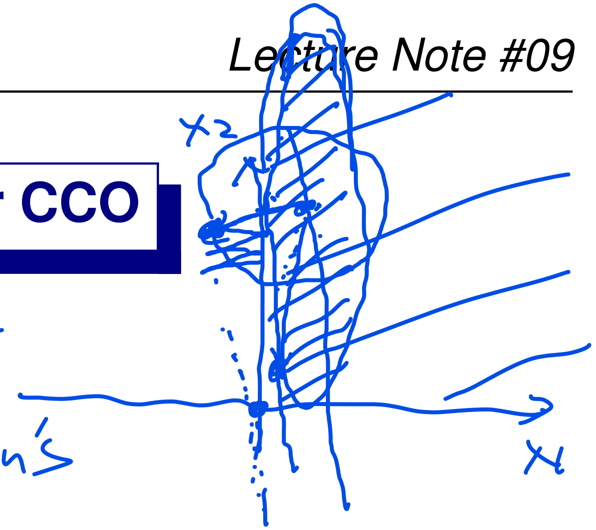
$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}\geq\mathbf{0}} \nabla f(\mathbf{x}^k)^T\mathbf{x} + \beta\mathcal{D}_h(\mathbf{x},\mathbf{x}^k).$$

The update above is the result of choosing (negative) entropy function $h(\mathbf{x}) = \sum_j x_j \log(x_j)$.

min $\nabla f(x)^T x$

s.t. $\|x - x^k\|_2^2 \le \delta$

Multiplicative-Update II: Affine Scaling SDM for CCO



1964

Dikin's

At the kth iterate with $\mathbf{x}^k > \mathbf{0}$, let D^k be a diagonal matrix such that

when $f(x)$ is $x \ge 0$

$$D^k_{jj} = x^k_j, \ \forall j$$

and

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x} \ge \mathbf{0}} \ \nabla f(\mathbf{x}^k)^T \mathbf{x} + \frac{\beta}{2}\|(D^k)^{-1}(\mathbf{x} - \mathbf{x}^k)\|^2,$$

or

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k (D^k)^2 \nabla f(\mathbf{x}^k) = \mathbf{x}^k . * \underbrace{(\mathbf{e} - \alpha_k \nabla f(\mathbf{x}^k) . * \mathbf{x}^k)}_{d}$$

where variable step-sizes can be

$$\alpha^k = \min\{\frac{1}{\beta \max(\mathbf{x}^k)^2}, \ \frac{1}{2\|\mathbf{x}^k . * \nabla f(\mathbf{x}^k)\|_\infty}\}.$$

Is $\mathbf{x}^k > \mathbf{0}, \ \forall k$? Does it converge? What is the convergence speed? See more details in HW3.

Geometric Interpretation: inscribed ball vs inscribed ellipsoid.

Affine Scaling for SDP Cone?

At the kth iterate with $X^k \succ \mathbf{0}$, the new SDM iterate would be

$$X^{k+1} = X^k - \alpha_k X^k \nabla f(X^k) X^k = X^k (I - \alpha_k \nabla f(X^k) X^k).$$

Choose step-size is chosen such that the smallest eigenvalue of X^{k+1} is at most a fraction from the one of X^k?

Does it converge? What is the convergence speed? See more details in HW3.

c:

Reduced Gradient Method – the Simplex Algorithm for LP

x_B , $|B| = m$

$\{1, \cdots, n\}$

$$\text{LP:} \quad \min \quad \mathbf{c}^T\mathbf{x} \quad \text{s.t. } A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0},$$

where $A \in R^{m \times n}$ has a full row rank m.

$\text{BS} \begin{cases} A_B x_B = b \\ x_N = 0 \end{cases}$ BFS + $x_B \geq 0$

Theorem 1 *(The Fundamental Theorem of LP in Algebraic form) Given (LP) and (LD) where A has full row rank m,*

i) *if there is a feasible solution, there is a basic feasible solution (Carathéodory's theorem);*

ii) *if there is an optimal solution, there is an optimal basic solution.*

High-Level Idea:

1. Initialization Start at a BSF or corner point of the feasible polyhedron.
2. Test for Optimality. Compute the reduced gradient vector at the corner. If no descent and feasible direction can be found, stop and claim optimality at the current corner point; otherwise, select a new corner point and go to Step 2.

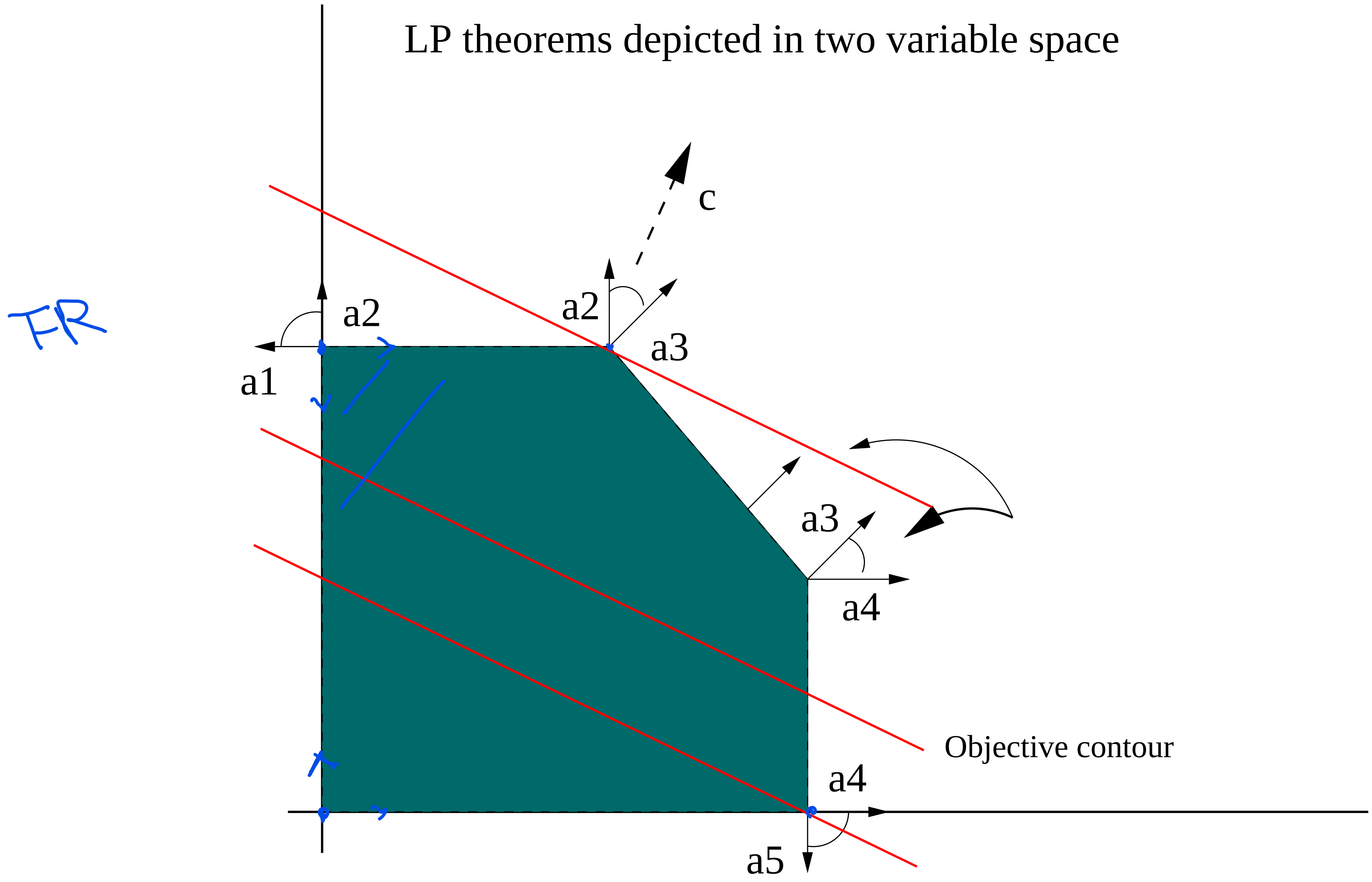


Figure 1: The LP Simplex Method

When a Basic Feasible Solution is Optimal

$|B| = m$

BFS $A_B x_B = b$ $x_N = 0$

Suppose the basis of a basic feasible solution is A_B and the rest is A_N. One can transform the equality constraint to

$$A_B^{-1}A\mathbf{x} = A_B^{-1}\mathbf{b}, \quad \text{so that } \mathbf{x}_B = A_B^{-1}\mathbf{b} - A_B^{-1}A_N\mathbf{x}_N.$$

That is, we express $\mathbf{x}_B$ in terms of $\mathbf{x}_N$, the non-basic variables are are active for constraints $\mathbf{x} \geq \mathbf{0}$.

Then the objective function equivalently becomes

$$\begin{aligned} \mathbf{c}^T\mathbf{x} = \mathbf{c}_B^T\mathbf{x}_B + \mathbf{c}_N^T\mathbf{x}_N &= \mathbf{c}_B^T A_B^{-1}\mathbf{b} - \mathbf{c}_B^T A_B^{-1}A_N\mathbf{x}_N + \mathbf{c}_N^T\mathbf{x}_N \\ &= \boxed{\mathbf{c}_B^T A_B^{-1}\mathbf{b} + (\mathbf{c}_N^T - \mathbf{c}_B^T A_B^{-1}A_N)\mathbf{x}_N.} \end{aligned}$$

$f\begin{pmatrix} x_B \\ x_N \end{pmatrix} = f_N(x_N)$

Vector $\mathbf{r}^T = \mathbf{c}^T - \mathbf{c}_B^T A_B^{-1}A$ is called the Reduced Gradient/Cost Vector where $\mathbf{r}_B = \mathbf{0}$ always.

Theorem 2 *If Reduced Gradient Vector $\mathbf{r}^T = \mathbf{c}^T - \mathbf{c}_B^T A_B^{-1}A \geq \mathbf{0}$, then the BFS is optimal.*

Proof: Let $\mathbf{y}^T = \mathbf{c}_B^T A_B^{-1}$ (called Shadow Price Vector), then $\mathbf{y}$ is a dual feasible solution ($\mathbf{r} = \mathbf{c} - A^T\mathbf{y} \geq \mathbf{0}$) and $\mathbf{c}^T\mathbf{x} = \mathbf{c}_B^T\mathbf{x}_B = \mathbf{c}_B^T A_B^{-1}\mathbf{b} = \mathbf{y}^T\mathbf{b}$, that is, the duality gap is zero.

The Simplex Algorithm Procedures

0. Initialize Start a BFS with basic index set B and let N denote the complementary index set.

1. Test for Optimality: Compute the Reduced Gradient Vector $\mathbf{r}$ at the current BFS and let

$$r_e = \min_{j \in N}\{r_j\}.$$

$e = \arg\min_{j \in N} \{r_j\}$

If $r_e \geq 0$, stop – the current BFS is optimal.

CDM

2. Determine the Replacement: Increase x_e while keep all other non-basic variables at the zero value (inactive) and maintain the equality constraints:

$x_o \leftarrow x_e$

$$\mathbf{x}_B = A_B^{-1}\mathbf{b} - A_B^{-1}A_{.e}x_e \ (\geq \mathbf{0}).$$

$B^k - \quad B^{k+1}$

If x_e can be increased to ∞, stop – the problem is unbounded below. Otherwise, let the basic variable x_o be the one first becoming 0.

3. Update basis: update B with x_o being replaced by x_e, and return to Step 1.

A Toy Example

$$
\begin{array}{llllllll}
\text{minimize} & -x_1 & -2x_2 & 0x_3 & 0x_4 & 0x_5 & \\
\text{subject to} & x_1 & & +x_3 & & & = 1 \\
 & & x_2 & & +x_4 & & = 1 \\
 & x_1 & +x_2 & & & +x_5 & = 1.5.
\end{array}
$$

$$
A = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 1 \\ 1 \\ 1.5 \end{pmatrix}, \mathbf{c}^T = (-1 \; -2 \;\; 0 \;\; 0 \;\; 0).
$$

Consider initial BFS with basic variables $B = \{3, 4, 5\}$ and $N = \{1, 2\}$.

Iteration 1:

$v_B = 0$

1. $A_B = I$, $A_B^{-1} = I$, $\mathbf{y}^T = (0\,0\,0)$ and $\mathbf{r}_N = (-1 \;\; -2)$ – it's NOT optimal. Let $e = 2$.

x_2

2. Increase x_2 while

$$\mathbf{x}_B = A_B^{-1}\mathbf{b} - A_B^{-1}A_{.2}x_2 = \begin{pmatrix} 1 \\ 1 \\ 1.5 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} x_2.$$

   We see x_4 becomes 0 first.

3. The new basic variables are $B = \{3, 2, 5\}$ and $N = \{1, 4\}$.

Iteration 2:

1.

$$A_B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \quad A_B^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix},$$

   $\mathbf{y}^T = (0 \;\; -2 \;\; 0)$ and $\mathbf{r}_N = (-1 \;\; 2)$ – it's NOT optimal. Let $e = 1$.

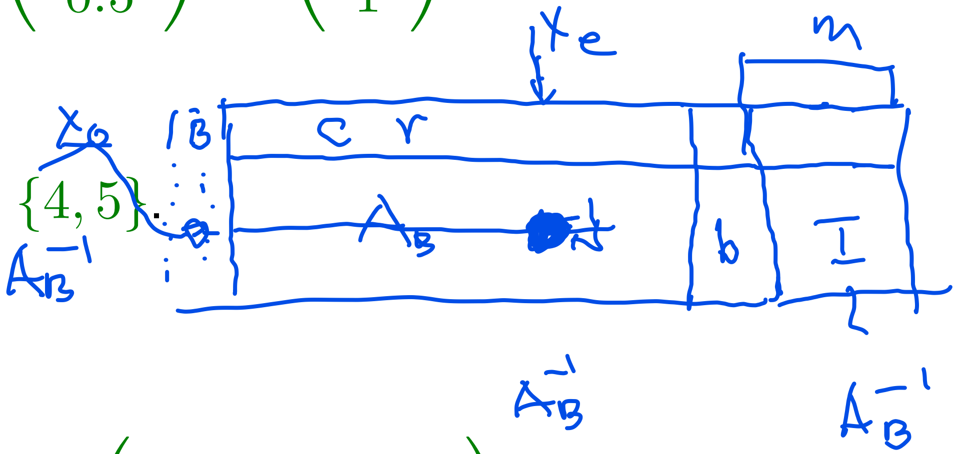
2. Increase x_1 while

$$\mathbf{x}_B = A_B^{-1}\mathbf{b} - A_B^{-1}A_{.1}x_1 = \begin{pmatrix} 1 \\ 1 \\ 0.5 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} x_1.$$

Simplex Tableau

GPU

We see x_5 becomes 0 first.



3. The new basic variables are $B = \{3, 2, 1\}$ and $N = \{4, 5\}$.

Iteration 3: LP

1.

$$A_B = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \quad A_B^{-1} = \begin{pmatrix} 1 & 1 & -1 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix},$$

$\mathbf{y}^T = (0 \ \ -1 \ \ -1)$ and $\mathbf{r}_N = (1 \ \ 1)$ – it's Optimal.

Is the Simplex Method always convergent to a minimizer? Which condition of the Global Convergence Theorem failed?

The Frank-Wolf Algorithm

$$\text{P:} \quad \min \ \boxed{f(\mathbf{x})} \ \text{ s.t. } A\mathbf{x} = \mathbf{b}, \ \mathbf{x} \geq \mathbf{0},$$

where $A \in R^{m \times n}$ has a full row rank m.

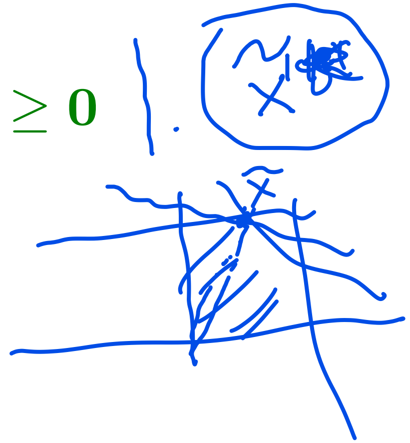
SLP

Start with a feasible solution $\mathbf{x}^0$, and at the kth iterate do:

- Solve the LP problem

$$\min \quad \nabla f(\mathbf{x}^k)^T \mathbf{x} \quad \text{s.t. } A\mathbf{x} = \mathbf{b}, \ \mathbf{x} \geq \mathbf{0}$$

  and let $\tilde{\mathbf{x}}^{k+1}$ be an optimal solution.



- Choose a step-size $0 < \alpha^k \leq 1$ and let

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k (\tilde{\mathbf{x}}^{k+1} - \mathbf{x}^k).$$

This is also called sequential linear programming (SLP) method.

First-Order Method for MDP: Value-Iteration of Fixed-Point Mapping

Let $\mathbf{y} \in \mathbf{R}^m$ represent the cost-to-go values of the m states, ith entry for ith state, of a given policy. The MDP problem entails choosing the optimal value vector $\mathbf{y}^*$ which is a fixed-point of:

$$y_i^* = \min_{j \in \mathcal{A}_i} \{c_j + \gamma \mathbf{p}_j^T \mathbf{y}^*\}, \ \forall i,$$

$y_i^* = \min_{j \in A_i} \{c_j + \gamma p_j^T y^*\}$

The Value-Iteration (VI) Method is, starting from any $\mathbf{y}^0$, the iterative mapping:

$$y_i^{k+1} = A(\mathbf{y}^k)_j = \min_{j \in \mathcal{A}_i} \{c_j + \gamma \mathbf{p}_j^T \mathbf{y}^k\}, \ \forall i.$$

$0 < \gamma < 1$

If the initial $\mathbf{y}^0$ is strictly feasible for state i, that is, $y_i^0 < c_j + \gamma \mathbf{p}_j^T \mathbf{y}^0, \ \forall j \in \mathcal{A}_i$, then y_i^k would be increasing in the VI iteration for all i and k.

On the other hand, if any of the inequalities is violated, then we have to decrease y_i^1 at least to

$$\min_{j \in \mathcal{A}_i} \{c_j + \gamma \mathbf{p}_j^T \mathbf{y}^0\}$$

.

Convergence of Value-Iteration for MDP

Theorem 3 *Let the VI algorithm mapping be* $A(\mathbf{v})_i = \min_{j \in \mathcal{A}_i}\{c_j + \gamma \mathbf{p}_j^T \mathbf{v}, \ \forall i\}$. *Then, for any two value vectors* $\mathbf{u} \in R^m$ *and* $\mathbf{v} \in R^m$ *and every state* i*:*

$$|A(\mathbf{u})_i - A(\mathbf{v})_i| \le \gamma \|\mathbf{u} - \mathbf{v}\|_\infty, \text{ which implies } \|A(\mathbf{u})_i - A(\mathbf{v})_i\|_\infty \le \gamma \|\mathbf{u} - \mathbf{v}\|_\infty$$

Let j_u and j_v be the two $\arg\min$ actions for value vectors $\mathbf{u}$ and $\mathbf{v}$, respectively. Assume that $A(\mathbf{u})_i - A(\mathbf{v})_i \ge 0$ where the other case can be proved similarly.

$$\begin{array}{rcl} 0 \le A(\mathbf{u})_i - A(\mathbf{v})_i & = & (c_{j_u} + \gamma \mathbf{p}_{j_u}^T \mathbf{u}) - (c_{j_v} + \gamma \mathbf{p}_{j_v}^T \mathbf{v}) \\ & \le & (c_{j_v} + \gamma \mathbf{p}_{j_v}^T \mathbf{u}) - (c_{j_v} + \gamma \mathbf{p}_{j_v}^T \mathbf{v}) \\ & = & \gamma \mathbf{p}_{j_v}^T (\mathbf{u} - \mathbf{v}) \le \gamma \|\mathbf{u} - \mathbf{v}\|_\infty. \end{array}$$

where the first inequality is from that j_u is the $\arg\min$ action for value vector $\mathbf{u}$, and the last inequality follows from the fact that the elements in $\mathbf{p}_{j_v}$ are non-negative and sum-up to 1.

Value-Iteration for MDP II: Other issues

The Value-Iteration (VI) Method for zero-sum game, starting from any $\mathbf{y}^0$, the iterative mapping:

$$y_i^{k+1} = A(\mathbf{y}^k)_j = \min_{j \in \mathcal{A}_i} \{c_j + \gamma \mathbf{p}_j^T \mathbf{y}^k\}, \ \forall i \in I^-$$

and

$$y_i^{k+1} = A(\mathbf{y}^k)_j = \max_{j \in \mathcal{A}_i} \{c_j + \gamma \mathbf{p}_j^T \mathbf{y}^k\}, \ \forall i \in I^+.$$

$0 < \gamma < 1$

Remarks':

- One can choose i at random to update, e.g., follow a random walk.
- Aggregate states if they have similar cost-to-go values
- State-values are updated in a unsynchronized manner: a state is updated after one of its neighbor-states is updated.

Many research issues in a suggested Project.

First-Order Method for Nonlinear Constrained Optimization I

We consider the general constrained optimization:

$$
\text{(GCO)} \qquad \begin{array}{lll} \min & f(\mathbf{x}) & \\ \text{s.t.} & c_i(\mathbf{x}) & = 0, \ i \in \mathcal{E}, \\ & c_i(\mathbf{x}) & \geq 0, \ i \in \mathcal{I}. \end{array}
$$

We can convert it to an unconstrained problem:

$$
\min \quad f(\mathbf{x}) + \lambda \sum_{i \in \mathcal{E}} |c_i(\mathbf{x})| - \mu \sum_{i \in \mathcal{I}} \log(c_i(\mathbf{x}))
$$

where λ is sufficiently large and μ is sufficiently small.

Not robust if a high accuracy is desired...

A remedy strategy is to adjust λ and μ dynamically, or use a projected gradient or reduced gradient first-order method, such as the Simplex Method of Dantzig...

First-Order Method for Nonlinear Constrained Optimization II

Another popular method is again Descent-First and Feasible-Second: linearize the nonlinear constraints using the first-order Taylor expansion and apply the Frank-Wolfe algorithm to compute a solution feasible for the linearized constraints, then project it onto the nonlinear-constrained feasible region.

Summary of the First-Order Methods

- Good global convergence property (e.g. starting from any (feasible) solution under mild technical assumption...).
- Simple to implement and the computation cost is mainly compute the numerical gradient.
- Maybe difficult to decide step-size: simple back-track is popular in practice.
- The convergence speed can be slow: not suitable for high accuracy computation, certain accelerations available.
- Can only guarantee converging to a first-order KKT solution.