More First-Order Optimization Algorithms

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

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

Chapters 3, 8, 13
The SDM for Unconstrained Convex Lipschitz Optimization

Here we consider $f(x)$ being convex and differentiable everywhere and satisfying the (first-order) $\beta$-Lipschitz condition. Given the knowledge $\beta$, we again adopt the fixed step-size rule:

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

\textbf{Theorem 1} For convex Lipschitz optimization the Steepest Descent Method generates a sequence of solutions such that

$$f(x^{k+1}) - f(x^*) \leq \frac{\beta}{k+2} \|x^0 - x^*\|^2 \text{ and } \min_{l=0, \ldots, k} \|\nabla f(x^l)\|^2 \leq \frac{4\beta^2}{(k + 1)(k + 2)} \|x^0 - x^*\|^2,$$

where $x^*$ is a minimizer of the problem.

\textbf{Proof:} For simplicity, we let $\delta^k = f(x^k) - f(x^*) (\geq 0)$, $g^k = \nabla f(x^k)$, and $\Delta^k = x^k - x^*$ in the rest of proof. As we have proved for general Lipschitz optimization

$$\delta^{k+1} - \delta^k = f(x^{k+1}) - f(x^k) \leq -\frac{1}{2\beta} \|g^k\|^2,$$

that is $\delta^k - \delta^{k+1} \geq \frac{1}{2\beta} \|g^k\|^2$.  \hspace{1cm} (2)
Furthermore, from the convexity,

\[-\delta^k = f(x^*) - f(x^k) \geq (g^k)^T (x^* - x^k) = -(g^k)^T \Delta^k, \quad \text{that is} \quad \delta^k \leq (g^k)^T \Delta^k. \quad (3)\]

Thus, from (2) and (3)

\[
\delta^{k+1} = \delta^{k+1} - \delta^k + \delta^k \\
\leq -\frac{1}{2\beta} \|g^k\|^2 + (g^k)^T \Delta^k \\
= -\frac{\beta}{2} \|x^{k+1} - x^k\|^2 - \beta (x^{k+1} - x^k)^T \Delta^k, \quad \text{(using (1))} \\
= -\frac{\beta}{2} (\|x^{k+1} - x^k\|^2 + 2(x^{k+1} - x^k)^T \Delta^k) \\
= -\frac{\beta}{2} (\|\Delta^{k+1} - \Delta^k\|^2 + 2(\Delta^{k+1} - \Delta^k)^T \Delta^k) \\
= \frac{\beta}{2} (\|\Delta^k\|^2 - \|\Delta^{k+1}\|^2). \quad (4)
\]

Sum up (4) from 1 to $k + 1$, we have

\[
\sum_{l=1}^{k+1} \delta^l \leq \frac{\beta}{2} (\|\Delta^0\|^2 - \|\Delta^{k+1}\|^2) \leq \frac{\beta}{2} \|\Delta^0\|^2.
\]
From the proof of the Corollary 1 of last lecture, we have $\delta^0 \leq \frac{\beta}{2} \|\Delta^0\|^2$. Thus,

$$\sum_{l=0}^{k+1} \delta^l \leq \beta \|\Delta^0\|^2,$$

(5)

and

$$\sum_{l=0}^{k+1} \delta^l = \sum_{l=0}^{k+1} (l + 1 - l) \delta^l$$

$$= \sum_{l=0}^{k+1} (l + 1) \delta^l - \sum_{l=0}^{k+1} l \delta^l$$

$$= \sum_{l=1}^{k+2} l \delta^{l-1} - \sum_{l=1}^{k+1} l \delta^l$$

$$= (k + 2) \delta^{k+1} + \sum_{l=1}^{k+1} l \delta^{l-1} - \sum_{l=1}^{k+1} l \delta^l$$

$$= (k + 2) \delta^{k+1} + \sum_{l=1}^{k+1} l \delta^{l-1} - \delta^l$$

$$\geq (k + 2) \delta^{k+1} + \sum_{l=1}^{k+1} l \frac{1}{2\beta} \|g^{l-1}\|^2,$$

where the first inequality comes from (2). Let $\|g'\| = \min_{l=0,\ldots,k} \|g^l\|$. Then we finally have

$$(k + 2) \delta^{k+1} + \frac{(k + 1)(k + 2)/2}{2\beta} \|g'\|^2 \leq \beta \|\Delta^0\|^2,$$

(6)

which completes the proof.
The Accelerated Steepest Descent Method (ASDM)

There is an accelerated steepest descent method (Nesterov 83) that works as follows:

\[ \lambda^0 = 0, \quad \lambda^{k+1} = \frac{1 + \sqrt{1 + 4(\lambda^k)^2}}{2}, \quad \alpha^k = \frac{1 - \lambda^k}{\lambda^{k+1}}, \]  

(7)

\[ \tilde{x}^{k+1} = x^k - \frac{1}{\beta} \nabla f(x^k), \quad x^{k+1} = (1 - \alpha^k)\tilde{x}^{k+1} + \alpha^k \tilde{x}^k. \]  

(8)

Note that \((\lambda^k)^2 = \lambda^{k+1}(\lambda^{k+1} - 1), \lambda^k > k/2\) and \(\alpha^k \leq 0\).

One can prove:

**Theorem 2**

\[ f(\tilde{x}^{k+1}) - f(x^*) \leq \frac{2\beta}{k^2} \|x^0 - x^*\|^2, \quad \forall k \geq 1. \]
Convergence Analysis of ASDM

Again for simplification, we let $\Delta^k = \lambda^k x^k - (\lambda^k - 1) \bar{x}^k - x^*$, $g^k = \nabla f(x^k)$ and $\delta^k = f(\tilde{x}^k) - f(x^*)(\geq 0)$ in the following.

Applying Lemma 1 for $x = \tilde{x}^{k+1}$ and $y = \tilde{x}^k$, convexity of $f$ and (8) we have

$$
\delta^{k+1} - \delta^k = f(\tilde{x}^{k+1}) - f(x^k) + f(x^k) - f(\tilde{x}^k)
\leq -\frac{\beta}{2} \|\tilde{x}^{k+1} - x^k\|^2 + f(x^k) - f(\tilde{x}^k)
\leq -\frac{\beta}{2} \|\tilde{x}^{k+1} - x^k\|^2 + (g^k)^T (x^k - \tilde{x}^k)
= -\frac{\beta}{2} \|\tilde{x}^{k+1} - x^k\|^2 - \beta(\tilde{x}^{k+1} - x^k)^T (x^k - \tilde{x}^k).
$$

(9)

Applying Lemma 1 for $x = \tilde{x}^{k+1}$ and $y = x^*$, convexity of $f$ and (8) we have

$$
\delta^{k+1} = f(\tilde{x}^{k+1}) - f(x^k) + f(x^k) - f(x^*)
\leq -\frac{\beta}{2} \|\tilde{x}^{k+1} - x^k\|^2 + f(x^k) - f(x^*)
\leq -\frac{\beta}{2} \|\tilde{x}^{k+1} - x^k\|^2 + (g^k)^T (x^k - x^*)
= -\frac{\beta}{2} \|\tilde{x}^{k+1} - x^k\|^2 - \beta(\tilde{x}^{k+1} - x^k)^T (x^k - x^*).
$$

(10)
Multiplying (9) by \( \lambda^k(\lambda^k - 1) \) and (10) by \( \lambda^k \) respectively, and summing the two, we have

\[
\begin{align*}
(\lambda^k)^2 \delta^{k+1} - (\lambda^{k-1})^2 \delta^k & \leq - (\lambda^k)^2 \frac{\beta}{2} \| \tilde{x}^{k+1} - x^k \|^2 - \lambda^k \beta (\tilde{x}^{k+1} - x^k)^T \Delta^k \\
& = - \frac{\beta}{2} ((\lambda^k)^2 \| \tilde{x}^{k+1} - x^k \|^2 + 2\lambda^k (\tilde{x}^{k+1} - x^k)^T \Delta^k) \\
& = - \frac{\beta}{2} (\| \lambda^k \tilde{x}^{k+1} - (\lambda^k - 1) \tilde{x}^k - x^* \|^2 - \| \Delta^k \|^2) \\
& = \frac{\beta}{2} (\| \Delta^k \|^2 - \| \lambda^k \tilde{x}^{k+1} - (\lambda^k - 1) \tilde{x}^k - x^* \|^2).
\end{align*}
\]

Using (7) and (8) we can derive

\[
\lambda^k \tilde{x}^{k+1} - (\lambda^k - 1) \tilde{x}^k = \lambda^{k+1} x^{k+1} - (\lambda^{k+1} - 1) \tilde{x}^{k+1}.
\]

Thus,

\[
(\lambda^k)^2 \delta^{k+1} - (\lambda^{k-1})^2 \delta^k \leq \frac{\beta}{2} (\| \Delta^k \|^2 - \| \Delta^{k+1} \|^2).
\] (11)

Sum up (11) from 1 to \( k \) we have

\[
\delta^{k+1} \leq \frac{\beta}{2(\lambda^k)^2} \| \Delta^1 \|^2 \leq \frac{2\beta}{k^2} \| \Delta^0 \|^2
\]

since \( \lambda^k \geq k/2 \) and \( \| \Delta^1 \| \leq \| \Delta^0 \| \).
First-Order Algorithms for Conic Constrained Optimization (CCO)

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

- Nonnegative Linear Regression: given data \( A \in \mathbb{R}^{m \times n} \) and \( b \in \mathbb{R}^m \)
  \[
  \min f(x) = \frac{1}{2} \| Ax - b \|^2 \text{ s.t. } x \geq 0; \quad \text{where } \nabla f(x) = A^T (Ax - b).
  \]

- Semidefinite Linear Regression: given data \( A_i \in S^n \) for \( i = 1, \ldots, m \) and \( b \in \mathbb{R}^m \)
  \[
  \min f(X) = \frac{1}{2} \| AX - b \|^2 \text{ s.t. } X \succeq 0; \quad \text{where } \nabla f(X) = A^T (AX - b).
  \]

\[
AX = \begin{pmatrix}
A_1 \cdot X \\
\vdots \\
A_m \cdot X
\end{pmatrix}
\quad \text{and} \quad A^T y = \sum_{i=1} y_i A_i.
\]

Suppose we start from a feasible solution \( x^0 \) or \( X^0 \).
\[ \hat{x}^{k+1} = x^k - \frac{1}{\beta} \nabla f(x^k) \]

\[ x^{k+1} = \text{Proj}_K(\hat{x}^{k+1}): \text{Solve } \min_{x \in K} \| x - \hat{x}^{k+1} \|^2. \]

For examples:

- If \( K = \{x : x \geq 0\} \), then
  \[ x^{k+1} = \text{Proj}_K(\hat{x}^{k+1}) = \max\{0, \hat{x}^{k+1}\}. \]

- If \( K = \{X : X \succeq 0\} \), then factorize \( \hat{X}^{k+1} = \sum_{j=1}^{n} \lambda_j v_j v_j^T \) and let
  \[ X^{k+1} = \text{Proj}_K(\hat{X}^{k+1}) = \sum_{j: \lambda_j > 0} \lambda_j v_j v_j^T. \]

  (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.
Consider the conic nonlinear optimization problem: \( \min f(x) \) s.t. \( Ax = b \). that is 
\[ K = \{ x : Ax = b \} . \]

The projection method becomes, starting from a feasible solution \( x^0 \) and let direction
\[
d^k = -(I - A^T(AA^T)^{-1}A) \nabla f(x^k)
\]

\[ x^{k+1} = x^k + \alpha^k d^k ; \] \hspace{2cm} (12)

where the stepsizze can be chosen from line-search or again simply let
\[
\alpha^k = \frac{1}{\beta}
\]

and \( \beta \) is the (global) Lipschitz constant.

Does the method converge? What is the convergence speed? See more details in HW3.
SDM Followed by the Feasible-Region-Projection III

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

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

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

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

Does the method converge? What is the convergence speed? What if $f(\cdot)$ is not a convex function?
Multiplicative-Update I: “Mirror” SDM for CCO

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

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

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

The classical Projected SDM update can be viewed as

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

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

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

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

12
At the $k$th iterate with $x^k > 0$, let $D^k$ be a diagonal matrix such that

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

and

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

or

$$x^{k+1} = x^k - \alpha_k (D^k)^2 \nabla f(x^k) = x^k \cdot (e - \alpha_k \nabla f(x^k) \cdot x^k)$$

where variable step-sizes can be

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

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

Geometric Interpretation: inscribed ball vs inscribed ellipsoid.
At the $k$th iterate with $X^k \succ 0$, 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.
Reduced Gradient Method – the Simplex Algorithm for LP

\[ \text{LP: } \min c^T x \quad \text{s.t. } Ax = b, \; x \geq 0, \]

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

**Theorem 3** (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.
LP theorems depicted in two variable space

Figure 1: The LP Simplex Method
When a Basic Feasible Solution is Optimal

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

$$A_B^{-1}A x = A_B^{-1}b,$$

so that $x_B = A_B^{-1}b - A_B^{-1}A_Nx_N$.

That is, we express $x_B$ in terms of $x_N$, the non-basic variables are are active for constraints $x \geq 0$.

Then the objective function equivalently becomes

$$c^T x = c_B^T x_B + c_N^T x_N = c_B^T A_B^{-1}b - c_B^T A_B^{-1}A_Nx_N + c_N^T x_N$$

$$= c_B^T A_B^{-1}b + (c_N - c_B^T A_B^{-1}A_N)x_N.$$

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

**Theorem 4** If Reduced Gradient Vector $r^T = c^T - c_B^T A_B^{-1}A \geq 0$, then the BFS is optimal.

**Proof**: Let $y^T = c_B^T A_B^{-1}$ (called Shadow Price Vector), then $y$ is a dual feasible solution $(r = c - A^Ty \geq 0)$ and $c^T x = c_B^T x_B = c_B^T A_B^{-1}b = y^T b$, 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 $r$ at the current BFS and let

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

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

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

   $$ x_B = A_B^{-1} b - A_B^{-1} A_e x_e (\geq 0). $$

   If $x_e$ can be increased to $\infty$, 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

\[
\begin{align*}
\text{minimize} & \quad -x_1 - 2x_2 \\
\text{subject to} & \quad x_1 + x_3 = 1 \\
& \quad x_2 + x_4 = 1 \\
& \quad x_1 + x_2 + x_5 = 1.5.
\end{align*}
\]

\[
A = \begin{pmatrix}
1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 1
\end{pmatrix}, \quad b = \begin{pmatrix}
1 \\
1 \\
1.5
\end{pmatrix}, \quad c^T = (-1 \quad -2 \quad 0 \quad 0 \quad 0).
\]

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

**Iteration 1:**

1. \( A_B = I, \ A_B^{-1} = I, \ y^T = (0 \ 0 \ 0) \) and \( r_N = (-1 \quad -2) \) – it’s NOT optimal. Let \( e = 2 \).
2. Increase \( x_2 \) while

\[
x_B = A_B^{-1} b - A_B^{-1} A_2 x_2 = \begin{pmatrix} 1 \\ 1 \\ 1.5 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} x_2.
\]

We see \( x_4 \) becomes 0 first.

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

**Iteration 2:**

1. \( A_B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \), \( A_B^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \),

\( y^T = (0 \ -2 \ 0) \) and \( r_N = (-1 \ 2) \) – it’s NOT optimal. Let \( e = 1 \).
2. Increase $x_1$ while

$$x_B = A_B^{-1}b - A_B^{-1}A_1x_1 = \begin{pmatrix} 1 \\ 1 \\ 0.5 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}x_1.$$ 

We see $x_5$ becomes 0 first.

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

**Iteration 3:**

1. 

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

$$y^T = (0 \ -1 \ -1) \text{ and } r_N = (1 \ 1) - \text{ it's Optimal}.$$ 

Is the Simplex Method always convergent to a minimizer? Which condition of the Global Convergence Theorem failed?
The Frank-Wolf Algorithm

P: \[ \min f(x) \quad \text{s.t.} \quad Ax = b, \ x \geq 0, \]

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

Start with a feasible solution \( x^0 \), and at the \( k \)th iterate do:

- Solve the LP problem
  \[ \min \nabla f(x^k)^T x \quad \text{s.t.} \quad Ax = b, \ x \geq 0 \]
  and let \( \tilde{x}^{k+1} \) be an optimal solution.

- Choose a step-size \( 0 < \alpha^k \leq 1 \) and let
  \[ x^{k+1} = x^k + \alpha^k (\tilde{x}^{k+1} - x^k). \]

This is also called sequential linear programming (SLP) method.
Let \( y \in \mathbb{R}^m \) represent the cost-to-go values of the \( m \) states, \( i \)th entry for \( i \)th state, of a given policy. The MDP problem entails choosing the optimal value vector \( y^* \) which is a fixed-point of:

\[
y_i^* = \min_{j \in A_i} \{ c_j + \gamma p_j^T y^* \}, \quad \forall i,
\]

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

\[
y_i^{k+1} = A(y^k)_j = \min_{j \in A_i} \{ c_j + \gamma p_j^T y^k \}, \quad \forall i.
\]

If the initial \( y^0 \) is strictly feasible for state \( i \), that is, \( y_i^0 < c_j + \gamma p_j^T y^0 \), \( \forall j \in A_i \), 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

\[
\min_{j \in A_i} \{ c_j + \gamma p_j^T y^0 \}
\].
Theorem 5 Let the VI algorithm mapping be \( A(v)_i = \min_{j \in A_i} \{c_j + \gamma p_j^T v, \forall i\} \). Then, for any two value vectors \( u \in \mathbb{R}^m \) and \( v \in \mathbb{R}^m \) and every state \( i \):

\[
|A(u)_i - A(v)_i| \leq \gamma \|u - v\|_{\infty}, \text{ which implies } \|A(u)_i - A(v)_i\|_{\infty} \leq \gamma \|u - v\|_{\infty}
\]

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

\[
0 \leq A(u)_i - A(v)_i = (c_{j_u} + \gamma p_{j_u}^T u) - (c_{j_v} + \gamma p_{j_v}^T v) \\
\leq (c_{j_v} + \gamma p_{j_v}^T u) - (c_{j_v} + \gamma p_{j_v}^T v) \\
= \gamma p_{j_v}^T (u - v) \leq \gamma \|u - v\|_{\infty}.
\]

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

Many research issues in suggested Project III.