Asynchronous Algorithms for Conic Programs, including Optimal, Infeasible, and Unbounded Ones

Wotao Yin
joint: Fei Feng, Robert Hannah, Yanli Liu, Ernest Ryu
(UCLA, Math)

DIMACS: Distributed Optimization, Information Processing, and Learning
August’17
Overview

- conic programming problem (P):

  \[
  \text{minimize } c^T x \quad \text{subject to } Ax = b, \ x \in K
  \]

  $K$ is a closed convex cone

- this talk: a first-order iteration
  - parallel: linear speedup, async
  - still working if problem is unsolvable
Approach overview

Douglas-Rachford\textsuperscript{1} fixed point iteration

\[ z^{k+1} = T z^k \]

\( T \) depends on \( A, b, c \) and has nice properties:

\textsuperscript{1}equivalent to standard ADMM, but the different form is important
Approach overview

Douglas-Rachford\textsuperscript{1} fixed point iteration

\[ z^{k+1} = T z^k \]

\( T \) depends on \( A, b, c \) and has nice properties:

- convergence guarantees and rates

\footnote{equivalent to standard ADMM, but the different form is important}
Douglas-Rachford\textsuperscript{1} fixed point iteration

\[ z^{k+1} = T z^k \]

\( T \) depends on \( A, b, c \) and has nice properties:

- convergence guarantees and rates
- coordinate friendly: break \( z \) into \( m \) blocks, cost\( (T_i) \sim \frac{1}{m} \) \( \text{cost}(T) \)

\textsuperscript{1} equivalent to standard ADMM, but the different form is important
Approach overview

Douglas-Rachford\(^1\) fixed point iteration

\[ z^{k+1} = Tz^k \]

\(T\) depends on \(A, b, c\) and has nice properties:

- convergence guarantees and rates
- coordinate friendly: break \(z\) into \(m\) blocks, \(\text{cost}(T_i) \sim \frac{1}{m}\text{cost}(T)\)
- divergent nicely:
  - \((P)\) has no primal-dual sol pair \(\Leftrightarrow \|z^k\| \to \infty\)
  - \(z^{k+1} - z^k\) tells a whole lot

\(^1\)equivalent to standard ADMM, but the different form is important
Douglas-Rachford splitting (Lions-Mercier’79)

- **proximal mapping** of a closed function $h$

$$\text{prox}_{\gamma h}(x) = \arg \min_z \{ h(z) + \frac{1}{2\gamma} \| z - x \|^2 \}$$
Douglas-Rachford splitting (Lions-Mercier’79)

- **proximal mapping** of a closed function $h$

  $$\text{prox}_{\gamma h}(x) = \arg \min_z \{ h(z) + \frac{1}{2\gamma} \| z - x \|^2 \}$$

- **Douglas-Rachford Splitting (DRS)** method solves

  $$\text{minimize } f(x) + g(x)$$

  by iterating

  $$z^{k+1} = T z^k$$
Douglas-Rachford splitting (Lions-Mercier’79)

- **proximal mapping** of a closed function $h$

  $\text{prox}_{\gamma h}(x) = \arg \min_z \{ h(z) + \frac{1}{2\gamma} \| z - x \|^2 \}$

- **Douglas-Rachford Splitting (DRS)** method solves

  minimize $f(x) + g(x)$

  by iterating

  $z^{k+1} = Tz^k$

  defined as:

  $x^{k+\frac{1}{2}} = \text{prox}_{\gamma g}(z^k)$

  $x^{k+1} = \text{prox}_{\gamma f}(2z^k - x^{k+\frac{1}{2}})$

  $z^{k+1} = z^k + (x^{k+1} - x^{k+\frac{1}{2}})$
Apply DRS to conic programming

\[ \text{minimize } c^T x \quad \text{subject to } Ax = b, \ x \in K \]

\[ \Leftrightarrow \text{minimize } \underbrace{(c^T x + \delta_{A \cdot b}(x))}_{f(x)} + \underbrace{\delta_K(x)}_{g(x)} \]

- cone \( K \) is nonempty closed convex
Apply DRS to conic programming

\[
\begin{align*}
\text{minimize} & \quad c^T x \quad \text{subject to} \quad Ax = b, \ x \in K \\
\Leftrightarrow \quad \text{minimize} & \quad \underbrace{(c^T x + \delta_{A \cdot = b}(x))}_f(x) + \underbrace{\delta_K(x)}_g(x)
\end{align*}
\]

- cone $K$ is nonempty closed convex
- each iteration: project onto $K$, then project onto $A \cdot = b$
Apply DRS to conic programming

\[
\begin{align*}
\text{minimize } & \ c^T x \quad \text{subject to } A x = b, \ x \in K \\
\Leftrightarrow & \quad \text{minimize } \left( c^T x + \delta_{A \cdot = b}(x) \right) + \delta_K(x) \\
\end{align*}
\]

- cone \( K \) is nonempty closed convex
- each iteration: project onto \( K \), then project onto \( A \cdot = b \)
- per-iteration cost: \( O(n^2) \) if \( x \in \mathbb{R}^n \) (by pre-factorizing \( AA^T \))
Apply DRS to conic programming

minimize $c^T x$  \quad subject to $Ax = b, \ x \in K$

$\Leftrightarrow$ minimize $\underbrace{(c^T x + \delta_{A \cdot = b}(x))}_{f(x)} + \underbrace{\delta_{K}(x)}_{g(x)}$

- cone $K$ is nonempty closed convex

- each iteration: project onto $K$, then project onto $A \cdot = b$

- per-iteration cost: $O(n^2)$ if $x \in \mathbb{R}^n$ (by pre-factorizing $AA^T$)

- prior work: ADMM for SDP (Wen-Goldfarb-Y.’09)
Other choices of splitting

- linearized ADMM and primal-dual splitting: avoid inverting full $A$
- variations of Frank-Wolfe: avoid expensive projections to SDP cone
- subgradient and bundle methods ...
Coordinate friendly$^2$ (CF)

- (Block) coordinate update is fast only if the subproblems are simple

- **definition:** $T : \mathcal{H} \rightarrow \mathcal{H}$ is CF if, for any $z$ and $i \in [m],$

  $$z^+ := (z_1, \ldots, (Tz)_i, \ldots, z_m)$$

  it holds that

  $$\text{cost}[\{z, \mathcal{M}(z)\} \mapsto \{z^+, \mathcal{M}(z^+)\}] = O\left(\frac{1}{m} \text{cost}[z \mapsto Tz]\right)$$

  where $\mathcal{M}(z)$ is some quantity maintained in the memory

---

$^2$Peng-Wu-Xu-Yan-Y. AMSA'16
Composed operators

- 9 rules\(^3\) for CF \(T_1 \circ T_2\) cover many examples

- general principles:
  - \(T_1 \circ T_2\) inherits the (weaker) separability property
  - if \(T_1\) is CF and \(T_2\) to be either \textit{cheap}, \textit{easy-to-maintain}, or \textit{directly CF}, then \(T_1 \circ T_2\) is CF
  - if \(T_1\) is separable or cheap, \(T_1 \circ T_2\) is easier to CF

\(^3\)Peng-Wu-Xu-Yan-Y. AMSA’16
Lists of CF $T_1 \circ T_2$

- many convex image processing models
- portfolio optimization
- most sparse optimization problems
- all LPs, all SOCPs, and SDPs without large cones
- most ERM problems
- ...

Example: DRS for SOCP

- second-order cone:

\[ Q^n = \{ x \in \mathbb{R}^n : x_1 \geq \| (x_2, \ldots, x_n) \|_2 \} \]
Example: DRS for SOCP

- second-order cone:
  \[ Q^n = \{ x \in \mathbb{R}^n : x_1 \geq \|(x_2, \ldots, x_n)\|_2 \} \]

- DRS operator has the form
  \[ T = \text{linear} \circ \text{proj}_{Q^{n_1} \times \ldots \times Q^{n_p}} \]
Example: DRS for SOCP

- second-order cone:

\[ Q^n = \{ x \in \mathbb{R}^n : x_1 \geq \| (x_2, \ldots, x_n) \|_2 \} \]

- DRS operator has the form

\[ T = \text{linear} \circ \text{proj}_{Q^n_1 \times \ldots \times Q^n_p} \]

- CF is trivial if all cones are small
Example: DRS for SOCP

- second-order cone:
  \[ Q^n = \{ x \in \mathbb{R}^n : x_1 \geq \| (x_2, \ldots, x_n) \|_2 \} \]

- DRS operator has the form
  \[ T = \text{linear} \circ \text{proj}_{Q^{n_1} \times \cdots \times Q^{n_p}} \]

- CF is trivial if all cones are small

- now, consider a big cone; property:
  \[ \text{proj}_{Q^n}(x) = (\alpha x_1, \beta x_2, \ldots, \beta x_n) \]
  where \( \alpha, \beta \) depend on \( x_1 \) and \( \gamma : = \| (x_2, \ldots, x_n) \|_2 \)
Example: DRS for SOCP

- second-order cone:
  \[ Q^n = \{ x \in \mathbb{R}^n : x_1 \geq \| (x_2, \ldots, x_n) \|_2 \} \]

- DRS operator has the form
  \[ T = \text{linear} \circ \text{proj}_{Q^n_1 \times \ldots \times Q^n_p} \]

- CF is trivial if all cones are small

- now, consider a big cone; property:
  \[ \text{proj}_{Q^n}(x) = (\alpha x_1, \beta x_2, \ldots, \beta x_n) \]
  where \( \alpha, \beta \) depend on \( x_1 \) and \( \gamma := \| (x_2, \ldots, x_n) \|_2 \)

- given \( \gamma \) and updating \( x_i \), refreshing \( \gamma \) costs \( O(1) \)
Example: DRS for SOCP

- second-order cone:
  \[ Q^n = \{ x \in \mathbb{R}^n : x_1 \geq \|(x_2, \ldots, x_n)\|_2 \} \]

- DRS operator has the form
  \[ T = \text{linear} \circ \text{proj}_{Q_1 \times \ldots \times Q_p} \]

- CF is trivial if all cones are small

- now, consider a big cone; property:
  \[ \text{proj}_{Q^n}(x) = (\alpha x_1, \beta x_2, \ldots, \beta x_n) \]
  where \( \alpha, \beta \) depend on \( x_1 \) and \( \gamma := \|(x_2, \ldots, x_n)\|_2 \)

- given \( \gamma \) and updating \( x_i \), refreshing \( \gamma \) costs \( O(1) \)

- by maintaining \( \gamma \), \( \text{proj}_{Q^n} \) is cheap, and \( T = \text{linear} \circ \text{cheap} \) is CF
Fixed-point iterations

- full update

\[ z^{k+1} = Tz^k \]
Fixed-point iterations

- **full update**
  \[ z^{k+1} = Tz^k \]

- **(block) coordinate update (CU):** choose \( i_k \in [m] \),
  \[
  z_{i_k}^{k+1} = \begin{cases} 
  z_{i_k}^{k} + \eta((Tz^k)_i - z_{i_k}^k), & \text{if } i = i_k \\
  z_{i_k}^{k}, & \text{otherwise.}
  \end{cases}
  \]
Fixed-point iterations

- full update
  \[ z^{k+1} = Tz^k \]

- (block) coordinate update (CU): choose \( i_k \in [m] \),
  \[ z_i^{k+1} = \begin{cases} 
  z_i^k + \eta((Tz^k)_i - z_i^k), & \text{if } i = i_k \\
  z_i^k, & \text{otherwise.} 
  \end{cases} \]

- parallel CU: \( p \) agents choose \( I_k \subset [m] \)
  \[ z_i^{k+1} = \begin{cases} 
  z_i^k + \eta((Tz^k)_i - z_i^k), & \text{if } i \in I_k \\
  z_i^k, & \text{otherwise.} 
  \end{cases} \]

- \( \eta \) depends on properties of \( T, i_k, \) and \( I_k \)
Sync-parallel versus async-parallel

**Synchronous**
(faster agents must wait)

**Asynchronous**
(all agents are non-stop)
ARock: async-parallel CU

- $p$ agents
- every agent continuously does: pick $i_k \subseteq [m],

$$z_{i}^{k+1} = \begin{cases} 
    z_{i}^{k} + \eta((Tz^{k-d_k})_i - z_{i}^{k-d_k}), & \text{if } i = i_k \\
    z_{i}^{k}, & \text{otherwise. }
\end{cases}$$

new notation:
- $k$: increases after any agent completes an update
- $z^{k-d_k} = (z_{1}^{k-d_k,1}, \ldots, z_{m}^{k-d_k,m})$ may be stale
- allow inconsistent atomic read/write
Various theories and meanings

- 1969 – 90s: $T$ is contractive in $\| \cdot \|_{w, \infty}$, partially/totally async
Various theories and meanings

- 1969 – 90s: $T$ is contractive in $\| \cdot \|_{w, \infty}$, partially/totally async

- recent in ML community: async SG and async BCD
  - early works: random $i_k$, bounded delays, $\mathbb{E}f$ has sufficient descent, treat delays as noise, delays independent of $i_k$

- ARock: $T$ is non-expansive in $\| \cdot \|_2$
  - unbounded noise ($t^{-\frac{4}{3}}$ or faster decay), Lyapunov analysis, delays as overdue progress, delays independent of $i_k$, provable running time async:sync $= 1 : \log(p)$ in a poisson system, prox is async

- Combettes-Eckstein: async projective splitting, free of parameter

- in distributed comp, also refer to: random activations, may not delay
Various theories and meanings

- 1969 – 90s: $T$ is contractive in $\| \cdot \|_{w,\infty}$, partially/totally async

- recent in ML community: **async SG and async BCD**
  - early works: random $i_k$, bounded delays, $\mathbb{E}f$ has sufficient descent, treat delays as noise, delays independent of $i_k$
  - state-of-the-art: allow essential cyclic $i_k$, unbounded noise ($t^{-4}$ or faster decay), Lyapunov analysis, delays as overdue progress, delays can depend on $i_k$
Various theories and meanings

- 1969 – 90s: $T$ is contractive in $\| \cdot \|_{\infty}$, partially/totally async

- recent in ML community: async SG and async BCD
  - early works: random $i_k$, bounded delays, $\mathbb{E} f$ has sufficient descent, treat delays as noise, delays independent of $i_k$
  - state-of-the-art: allow essential cyclic $i_k$, unbounded noise ($t^{-4}$ or faster decay), Lyapunov analysis, delays as overdue progress, delays can depend on $i_k$

- ARock: $T$ is non-expansive in $\| \cdot \|_2$
  - unbounded noise ($t^{-4}$ or faster decay), Lyapunov analysis, delays as overdue progress, delays independent of $i_k$, provable running time async:sync = $1 : \log(p)$ in a poisson system, prox is async

- Combettes-Eckstein: async projective splitting, free of parameter
Various theories and meanings

- 1969 – 90s: \( T \) is contractive in \( \| \cdot \|_{w, \infty} \), partially/totally async

- recent in ML community: async SG and async BCD
  - early works: random \( i_k \), bounded delays, \( \mathbb{E} f \) has sufficient descent, treat delays as noise, delays independent of \( i_k \)
  - state-of-the-art: allow essential cyclic \( i_k \), unbounded noise (\( t^{-4} \) or faster decay), Lyapunov analysis, delays as overdue progress, delays can depend on \( i_k \)

- ARock: \( T \) is non-expansive in \( \| \cdot \|_2 \)
  - unbounded noise (\( t^{-4} \) or faster decay), Lyapunov analysis, delays as overdue progress, delays independent of \( i_k \), provable running time async:sync = 1 : \log(p) in a poisson system, prox is async

- Combettes-Eckstein: async projective splitting, free of parameter

- in distributed comp, also refer to: random activations, may not delay
ARock convergence

notation:
- \( m = \# \) blocks
- \( \tau = \max \) async delay
- uniform random selection (non-uniform is okay)

Theorem (known max delay)

Assume: \( T \) is nonexpansive and has a fixed point, and delays do not depend on \( i_k \). Use step size \( \eta_k \in \left[ \epsilon, \frac{1}{2m^{1/2}\tau+1} \right) \). Then, \( x^k \to x^* \in \text{Fix}T \) almost surely.

\(^4\)Peng-Xu-Yan-Y. SISC’16
ARock convergence

notation:

- $m = \# \text{ blocks}$
- $\tau = \text{max async delay}$
- uniform random selection (non-uniform is okay)

Theorem (known max delay)

Assume: $T$ is nonexpansive and has a fixed point, and delays do not depend on $i_k$. Use step size $\eta_k \in \left[\epsilon, \frac{1}{2m-1/2\tau+1}\right)$. Then, $x^k \rightharpoonup x^* \in \text{Fix}T$ almost surely.

consequence:

- no sync at least until using $O(\sqrt{m})$ agents
- sharp when $\tau \ll m$
## Optimization and fixed-point examples

<table>
<thead>
<tr>
<th>Optimization problem</th>
<th>Algorithm</th>
<th>Nonexpansive fixed-point operator $T$</th>
<th>Assumptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\min f(x)$</td>
<td>Gradient descent</td>
<td>$I - \gamma \nabla f$</td>
<td>$\gamma \in (0, \frac{2}{L_{\nabla f}}]$</td>
</tr>
<tr>
<td>$\min f(x)$</td>
<td>Proximal point</td>
<td>$J_{\gamma \partial f}$</td>
<td>$\gamma &gt; 0$</td>
</tr>
<tr>
<td>$\min f(x) + g(x)$</td>
<td>Forward backward</td>
<td>$J_{\gamma \partial f} \circ (I - \gamma \nabla g)$</td>
<td>$\gamma \in (0, \frac{2}{L_{\nabla g}}]$</td>
</tr>
<tr>
<td>$\min {g(x) : x \in C}$</td>
<td>Projected gradient</td>
<td>$\text{Proj}_C \circ (I - \gamma \nabla g)$</td>
<td>$\gamma \in (0, \frac{2}{L_{\nabla g}}]$</td>
</tr>
<tr>
<td>$\min f(x) + g(x)$</td>
<td>Peaceman-Rachford</td>
<td>$R_{\gamma \partial f} \circ R_{\gamma \partial g}$</td>
<td>$\gamma &gt; 0$</td>
</tr>
<tr>
<td>$\min \sum_{i=1}^d f_i(x)$</td>
<td>Parallel Peaceman-Rachford</td>
<td>$(\frac{2}{d}1^T \mathbf{1} - I) \circ R_{\gamma \partial f}$ where $f = [f_1; \ldots; f_d] : \mathbb{H}^d \rightarrow \mathbb{R}^d$</td>
<td>$\gamma &gt; 0$</td>
</tr>
<tr>
<td>$\min f(x) + g(x)$</td>
<td>Douglas-Rachford</td>
<td>$\frac{1}{2}I + \frac{1}{2}R_{\gamma \partial f} \circ R_{\gamma \partial g}$</td>
<td>$\gamma &gt; 0$</td>
</tr>
<tr>
<td>$\min f(x) + g(x) + h(x)$</td>
<td>Davis-Yin</td>
<td>$I - J_{\gamma \partial g} \circ J_{\gamma \partial f} \circ (2J_{\gamma \partial g} - I - \gamma \nabla h \circ J_{\gamma \partial g})$</td>
<td>$\gamma \in (0, \frac{2}{L_{\nabla h}}]$</td>
</tr>
<tr>
<td>$\min {f(x) + g(z) : Ax + Bz = b}$</td>
<td>ADMM</td>
<td>$\frac{1}{2}I + \frac{1}{2}R_{\gamma \partial F} \circ R_{\gamma \partial G}$, where $F(y) := f^<em>(A^Ty)$, $G(y) := g^</em>(B^Ty) - b^Ty$</td>
<td>$\gamma &gt; 0$</td>
</tr>
</tbody>
</table>
## Applications

<table>
<thead>
<tr>
<th>Convex Optimization Problem</th>
<th>Setup</th>
<th>ARock Iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smooth minimization: ( \min f(x) )  ( \nabla f ) is ( L )-Lipschitz, ( \nabla f = \begin{pmatrix} \nabla f_1 \ \vdots \ \nabla f_m \end{pmatrix} )</td>
<td>( x_{ik}^{k+1} \leftarrow x_{ik}^k - \frac{\eta^k}{L} \nabla f_{ik}(\hat{x}^k) )</td>
<td></td>
</tr>
<tr>
<td>Constrained minimization: ( \min f(x) ) subject to ( l \leq x \leq u )  ( \nabla f ) is ( L )-Lipschitz</td>
<td>( x_{ik}^{k+1} \leftarrow x_{ik}^k - \eta^k \left( \hat{x}<em>{ik}^k - \text{Proj}</em>{[l_{ik}, u_{ik}]}(\hat{x}<em>{ik}^k - \frac{2}{L} \nabla f</em>{ik}(\hat{x}^k)) \right) )</td>
<td></td>
</tr>
<tr>
<td>Composite minimization (ERM model): ( \min f(x) + g(x) )  ( g(x) = \sum_{i=1}^m g_i(x_i) )</td>
<td>( x_{ik}^{k+1} \leftarrow x_{ik}^k - \eta^k \left( \hat{x}<em>{ik}^k - \text{prox}</em>{\frac{2}{L} g_i}(\hat{x}<em>{ik}^k - \frac{2}{L} \nabla f</em>{ik}(\hat{x}^k)) \right) )</td>
<td></td>
</tr>
<tr>
<td>Kernel SVM: ( \min \frac{1}{2} s^T Q s - e^T s ) subject to ( \sum_i y_i s_i = 0 ), ( 0 \leq s_i \leq C )</td>
<td>training set ( {x_i, y_i} ), ( y_i \in {\pm 1} ), kernel ( k(\cdot, \cdot) ), ( Q_{ij} = y_i y_j k(x_i, x_j) ), applies Davis-Yin</td>
<td>See the last equation in [20, Section 5.2.1], and apply it with damping ( \eta^k )</td>
</tr>
<tr>
<td>Linear System: Solve ( Ax = b ) ( A ) is symmetric positive definite, ( \begin{pmatrix} -A_1 &amp; - &amp; \cdot \ \cdot &amp; \ddots &amp; \cdot \ - &amp; \cdot &amp; -A_m \end{pmatrix} \begin{pmatrix} x_1 \ \vdots \ x_m \end{pmatrix} = \begin{pmatrix} b_1 \ \vdots \ b_m \end{pmatrix} )</td>
<td>( x_{ik}^{k+1} \leftarrow x_{ik}^k - \left( \frac{2\eta^k}{M} \right)(A_{ik} \hat{x}^k + b_{ik}) )</td>
<td></td>
</tr>
<tr>
<td>Linear System: Solve ( Ax = b ) ( A = D + R ) where ( D ) is diagonal, ( M ) off-diagonal, ( \rho(D^{-1} R) \leq 1 )</td>
<td>( x_{ik}^{k+1} \leftarrow x_{ik}^k - \eta^k \left( (I + D^{-1} M) \hat{x}^k - D^{-1} b \right)_{ik} )</td>
<td></td>
</tr>
</tbody>
</table>
More complicated applications

- LP, QP, SOCP, some SDP
- Image reconstruction minimization
- Nonnegative matrix factorization
- Decentralized optimization (no global coordination anymore!)
QCQP test: ARock versus SCS\textsuperscript{5}

\textsuperscript{5}O’Donoghue, Chu, Parikh, Boyd’15
Practice

coding:

- OpenMP, C++11, MPI
- easier than you think

performance:

- if done “correctly”, async speed $\gg$ sync speed
- much faster when systems get bigger and/or unbalanced
An ideal solver

- find a solution if there is one
An ideal solver

- find a solution if there is one
- when there is no solution,
  - reliably report “no solution”
An ideal solver

- find a solution if there is one

- when there is no solution,
  - reliably report “no solution”
  - provide a “certificate”
An ideal solver

- find a solution if there is one

- **when there is no solution,**
  - reliably report “no solution”
  - provide a “certificate”
  - suggest the cheapest fix
An ideal solver

- find a solution if there is one

- **when there is no solution,**
  - reliably report “no solution”
  - provide a “certificate”
  - suggest the cheapest fix

- **status:** achievable for LP, not for SOCPs yet
Conic programming

\[ p^* = \min c^T x \quad \text{subject to} \quad Ax = b, \quad x \in K \]

\( K \) is a closed convex cone
Conic programming

\[ p^* = \min c^T x \quad \text{subject to} \quad Ax = b, \quad x \in K \]

\( K \) is a closed convex cone

- every problem falls in exactly one of the seven cases:
  1) \( p^* \) finite: 1a) has PD sol pair, 1b) only P sol, 1c) no P sol
Conic programming

\[ p^* = \min c^T x \quad \text{subject to} \quad Ax = b, \quad x \in K \]

\( K \) is a closed convex cone

- every problem falls in exactly one of the seven cases:
  1) \( p^* \) finite: 1a) has PD sol pair, 1b) only P sol, 1c) no P sol
  2) \( p^* = -\infty \): 2a) has improving dir, 2b) no improving dir
Conic programming

\[ p^* = \min c^T x \quad \text{subject to } A x = b, \quad x \in K \]

\(K\) is a closed convex cone

- every problem falls in exactly one of the **seven cases**:
  1) \(p^*\) finite: 1a) has PD sol pair, 1b) only P sol, 1c) no P sol
  2) \(p^* = -\infty\): 2a) has improving dir, 2b) no improving dir
  3) \(p^* = +\infty\): 3a) \(\text{dist}(L, K) > 0 \iff \text{has separating hyperplane}\)
    3b) \(\text{dist}(L, K) = 0 \iff \text{no strict separating hyperplane}\)

- (nearly) pathological cases fail existing solvers
Example 1

- 3-variable problem:

  minimize \( x_1 \)  subject to \( x_2 = 1, \ 2x_2x_3 \geq x_1^2 \).

- since \( x_2, x_3 \geq 0 \), the problem is equivalent to

  minimize \( x_1 \)  subject to \( x_2 = 1, \ (x_1, x_2, x_3) \in \text{rotated second-order cone} \).

---

\(^6\) \( p^* = -\infty \), by letting \( x_3 \to \infty \) and \( x_1 \to -\infty \)

\(^7\) reason: any improving direction \( u \) has form \((u_1, 0, u_3)\), but by the cone constraint \( 2u_2u_3 = 0 \geq u_1^2 \), so \( u_1 = 0 \), which implies \( c^T u_1 = 0 \) (not improving).
Example 1

- **3-variable problem:**

  \[
  \begin{align*}
  & \text{minimize } x_1 \quad \text{subject to } x_2 = 1, \ 2x_2x_3 \geq x_1^2.
  \end{align*}
  \]

- since \(x_2, x_3 \geq 0\), the problem is equivalent to

  \[
  \begin{align*}
  & \text{minimize } x_1 \quad \text{subject to } x_2 = 1, \ (x_1, x_2, x_3) \in \text{rotated second-order cone}.
  \end{align*}
  \]

- **classification:** (2b)
  - feasible
  - unbounded\(^6\)
  - no improving direction\(^7\)

\(^6\)\text{\(p^* = -\infty\), by letting \(x_3 \rightarrow \infty\) and \(x_1 \rightarrow -\infty\)}

\(^7\)\text{\(\text{reason: any improving direction } u \text{ has form } (u_1, 0, u_3), \text{ but by the cone constraint } 2u_2u_3 = 0 \geq u_1^2, \text{ so } u_1 = 0, \text{ which implies } c^Tu_1 = 0 \text{ (not improving).}\)}
Example 1

- **3-variable problem:**

  minimize \( x_1 \) subject to \( x_2 = 1, \ 2x_2x_3 \geq x_1^2 \).

- since \( x_2, x_3 \geq 0 \), the problem is equivalent to

  minimize \( x_1 \) subject to \( x_2 = 1, (x_1, x_2, x_3) \in \) rotated second-order cone.

- **classification:** (2b)
  - feasible
  - unbounded\(^6\)
  - no improving direction\(^7\)

- **solver results:**
  - SDPT3: “Failed”, \( p^* \) no reported
  - SeDuMi: “Inaccurate/Solved”, \( p^* = -175514 \)
  - Mosek: “Inaccurate/Unbounded”, \( p^* = -\infty \)

---

\(^6\) \( p^* = -\infty \), by letting \( x_3 \to \infty \) and \( x_1 \to -\infty \)

\(^7\) reason: any improving direction \( u \) has form \((u_1, 0, u_3)\), but by the cone constraint \( 2u_2u_3 = 0 \geq u_1^2 \), so \( u_1 = 0 \), which implies \( c^T u_1 = 0 \) (not improving).
Example 2

- 3-variable problem:

\[
\begin{align*}
\text{minimize } & 0 \quad \text{subject to } \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad x_3 \geq \sqrt{x_1^2 + x_2^2}. \\
& x \in L, \quad x_3 \geq \sqrt{x_1^2 + x_2^2}.
\end{align*}
\]

\[\text{classifiers:} \quad (3b) \]

\[\text{infeasible: } \quad L \cap K = \emptyset \]

\[\text{dist}(L, K) = 0 \]

\[\text{no strict separating hyperplane} \]

\[\text{solver results:} \quad \begin{align*}
\text{SDPT3: } & \text{"Infeasible"}, \quad p^\star = \infty \\
\text{SeDuMi: } & \text{"Solved"}, \quad p^\star = 0 \\
\text{Mosek: } & \text{"Failed"}, \quad p^\star \text{not reported}
\end{align*} \]

\[8 \; x \in L \text{ imply } x = [1, -\alpha, \alpha]^T, \; \alpha \in \mathbb{R}, \text{ which always violates the second-order cone constraint.} \]

\[9 \; \text{dist}(L, K) \leq \|[1, -\alpha, \alpha] - [1, -\alpha, (\alpha^2 + 1)^{1/2}]\|_2 \to 0 \text{ as } \alpha \to \infty. \]
Example 2

- 3-variable problem:

\[
\text{minimize } 0 \quad \text{subject to } \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad x_3 \geq \sqrt{x_1^2 + x_2^2}.
\]

\[x \in L, \quad x_3 \geq \sqrt{x_1^2 + x_2^2} \quad x \in K\]

- classification: (3b)

  - infeasible\(^8\), \(L \cap K = \emptyset\)
  - \(\text{dist}(L, K) = 0\) \(^9\)
  - no strict separating hyperplane

\(^8\) \(x \in L\) imply \(x = [1, -\alpha, \alpha]^T, \alpha \in \mathbb{R}\), which always violates the second-order cone constraint.

\(^9\) \(\text{dist}(L, K) \leq \|[1, -\alpha, \alpha] - [1, -\alpha, (\alpha^2 + 1)^{1/2}]\|_2 \to 0\) as \(\alpha \to \infty\).
Example 2

- **3-variable problem:**
  
  minimize 0 \quad \text{subject to} \quad \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \begin{cases} x_3 \geq \sqrt{x_1^2 + x_2^2} \quad & x \in L \\ x \in K \end{cases}

- **classification:** (3b)
  - infeasible, \( L \cap K = \emptyset \)
  - \( \text{dist}(L, K) = 0 \)
  - no strict separating hyperplane

- **solver results:**
  - SDPT3: “Infeasible”, \( p^* = \infty \)
  - SeDuMi: “Solved”, \( p^* = 0 \)
  - Mosek: “Failed”, \( p^* \) not reported

---

\( x \in L \) imply \( x = [1, -\alpha, \alpha]^T, \alpha \in \mathbb{R}, \) which always violates the second-order cone constraint.

\( \text{dist}(L, K) \leq \|[1, -\alpha, \alpha] - [1, -\alpha, (\alpha^2 + 1)^{1/2}]\|_2 \to 0 \) as \( \alpha \to \infty. \)
Then, what happens to DRS?

In 1970s, Paty, Rockafellar

- assume $T$ is firmly nonexpansive
- run $z^{k+1} = T(z^k)$
- converges if has PD sol; otherwise, $\|z^k\| \to \infty$

In 1979, Bailion-Bruck-Reich nailed

$$z^k - z^{k+1} \to v = \text{Proj}_{\text{ran}(I-T)}(0)$$
Our analysis results (Liu-Ryu-Y.’17)

- **rate of convergence**: \[ \| z^k - z^{k+1} \| \leq \| v \| + \epsilon + O\left(\frac{1}{\sqrt{k+1}}\right) \]
Our analysis results (Liu-Ryu-Y.’17)

- **rate of convergence:** \[ \| z^k - z^{k+1} \| \leq \| v \| + \epsilon + O\left( \frac{1}{\sqrt{k+1}} \right) \]

- **deciphered** \( \text{Proj}_{\text{ran}(I-T)} \)
Our analysis results (Liu-Ryu-Y.’17)

- **rate of convergence:** \[ \| z^k - z^{k+1} \| \leq \| v \| + \epsilon + O\left(\frac{1}{\sqrt{k+1}}\right) \]

- **deciphered** \( \text{Proj}_{\text{ran}(I-T)} \)

- **a workflow** running three similar DRS, differ by only constants:
  1) DRS
  2) feasibility DRS with \( c = 0 \)
  3) boundedness DRS with \( b = 0 \)

- most pathological cases are **identified**
Our analysis results (Liu-Ryu-Y.’17)

- **rate of convergence**: \( \| z^k - z^{k+1} \| \leq \| v \| + \epsilon + O\left( \frac{1}{\sqrt{k+1}} \right) \)

- **deciphered** \( \text{Proj}_{\text{ran}(I-T)} \)

- **a workflow** running three similar DRS, differ by only constants:
  1) DRS
  2) feasibility DRS with \( c = 0 \)
  3) boundedness DRS with \( b = 0 \)

- most pathological cases are **identified**

- **compute an improving direction** if one exists
Our analysis results (Liu-Ryu-Y.’17)

- **rate of convergence:** \[ |z^k - z^{k+1}| \leq |v| + \epsilon + O\left(\frac{1}{\sqrt{k+1}}\right) \]

- **deciphered** \[ \text{Proj}_{\text{ran}(I-T)} \]

- **a workflow** running three similar DRS, differ by only constants:
  1) DRS
  2) feasibility DRS with \( c = 0 \)
  3) boundedness DRS with \( b = 0 \)

- most pathological cases are **identified**

- **compute an improving direction** if one exists

- **compute a separating hyperplane** if one exists
Our analysis results (Liu-Ryu-Y.’17)

- rate of convergence: \( \|z^k - z^{k+1}\| \leq \|v\| + \epsilon + O\left(\frac{1}{\sqrt{k+1}}\right) \)

- deciphered \( \operatorname{Proj}_{\operatorname{ran}(I-T)} \)

- a workflow running three similar DRS, differ by only constants:
  1) DRS
  2) feasibility DRS with \( c = 0 \)
  3) boundedness DRS with \( b = 0 \)

- most pathological cases are identified

- compute an improving direction if one exists

- compute a separating hyperplane if one exists

- for all infeasible problems, minimally change to restore strong feasibility
Decision flow

Start

Thm 6
Alg 2

Thm 2
Alg 1

Thm 7
Alg 2

(f) Strongly infeasible

(g) Weakly infeasible

Infeasible

Feasible

Thm 13
Alg 1

Thm 11,12
Alg 3

(a) There is a primal-dual solution pair with $d^* = p^*$

(b) There is a primal solution but no dual solution or $d^* < p^*$

(c) $p^*$ is finite but there is no solution

(d) Unbounded ($p^* = -\infty$) with an improving direction

(e) Unbounded ($p^* = -\infty$) without an improving direction
Infeasible SDP test set (Liu-Pataki’17)

<table>
<thead>
<tr>
<th></th>
<th>Clean</th>
<th>Messy</th>
<th>Clean</th>
<th>Messy</th>
</tr>
</thead>
<tbody>
<tr>
<td>SeDuMi</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>SDPT3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mosek</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>PP$^{10}$ + SeDuMi</td>
<td>100</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
</tbody>
</table>

percentage of success detection on clean and messy examples in Liu-Pataki’17

$^{10}$ PreProcessing by Permenter-Parilo’14
Identify weakly infeasible SDPs

<table>
<thead>
<tr>
<th></th>
<th>$m = 10$</th>
<th></th>
<th>$m = 20$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean</td>
<td>100</td>
<td>Clean</td>
<td>100</td>
</tr>
<tr>
<td>Messy</td>
<td>21</td>
<td>Messy</td>
<td>99</td>
</tr>
</tbody>
</table>

(stopping: $\|z^{1e7}\|_2 \geq 800$)

our percentage is way much better!
Identify strongly infeasible SDPs

<table>
<thead>
<tr>
<th></th>
<th>$m = 10$</th>
<th></th>
<th>$m = 20$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean</td>
<td>100</td>
<td>Clean</td>
<td>100</td>
</tr>
<tr>
<td>Messy</td>
<td>100</td>
<td>Messy</td>
<td>100</td>
</tr>
</tbody>
</table>

Proposed

(stopping: $\|z^{5e4} - z^{5e4+1}\|_2 \leq 10^{-3}$)

our percentage is way much better!
Thank you!

References: UCLA CAM reports

- 15-37: ARock
- 16-13: Coordinate friendly, SOCP applications
- 17-30: Unbounded and realistic-delay async BCD
- 17-31: DRS for unsolvable conic programs
- arXiv:1708.05136: provably async-to-sync speedup