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-Rachfordf[1] fixed point iteration

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

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

[1] equivalent to standard ADMM, but the different form is important

Approach overview

Douglas-Rachfordf[1] fixed point iteration

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

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

- convergence guarantees and rates

[1] equivalent to standard ADMM, but the different form is important

Approach overview

Douglas-Rachfordf[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, $\mathrm{cost}(T_i) \sim \frac{1}{m}\mathrm{cost}(T)$

[1] equivalent to standard ADMM, but the different form is important

Approach overview

Douglas-Rachfordf[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, $\mathrm{cost}(T_i) \sim \frac{1}{m}\mathrm{cost}(T)$
- **divergent nicely:**
  - (P) has no primal-dual sol pair $\Leftrightarrow$ $\|\mathbf{z}^k\| \to \infty$
  - $z^{k+1} - z^k$ 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

$$\mathbf{prox}_{\gamma h}(x) = \arg\min_{z}\{h(z) + \tfrac{1}{2\gamma}\|z - x\|^2\}$$

Douglas-Rachford splitting (Lions-Mercier'79)

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

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

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

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

by iterating

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

Douglas-Rachford splitting (Lions-Mercier'79)

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

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

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

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

by iterating

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

defined as:

$$\begin{aligned} x^{k+\frac{1}{2}} &= \mathbf{prox}_{\gamma g}(z^k) \\ x^{k+1} &= \mathbf{prox}_{\gamma f}(2z^k - x^{k+\frac{1}{2}}) \\ z^{k+1} &= z^k + (x^{k+1} - x^{k+\frac{1}{2}}) \end{aligned}$$

Apply DRS to conic programming

$$\begin{aligned} &\text{minimize } \ c^T x \quad \text{subject to } Ax = b, \ x \in K \\ \Leftrightarrow \quad &\text{minimize } \ \underbrace{\left(c^T x + \delta_{A\cdot=b}(x)\right)}_{f(x)} + \underbrace{\delta_K(x)}_{g(x)} \end{aligned}$$

- cone K is nonempty closed convex

Apply DRS to conic programming

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

$$\Leftrightarrow \quad \text{minimize } \underbrace{\left(c^T x + \delta_{A\cdot=b}(x)\right)}_{f(x)} + \underbrace{\delta_K(x)}_{g(x)}$$

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

Apply DRS to conic programming

$$\begin{aligned} &\text{minimize } \; c^T x \quad \text{subject to } Ax = b, \; x \in K \\ \Leftrightarrow \quad &\text{minimize } \; \underbrace{\left(c^T x + \delta_{A\cdot=b}(x)\right)}_{f(x)} + \underbrace{\delta_K(x)}_{g(x)} \end{aligned}$$

- 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

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

$$\Leftrightarrow \quad \text{minimize } \underbrace{\left(c^T x + \delta_{A\cdot=b}(x)\right)}_{f(x)} + \underbrace{\delta_K(x)}_{g(x)}$$

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

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

it holds that

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

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 *cheap*, *easy-to-maintain*, or *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

$$T = \text{linear} \circ \mathbf{proj}_{Q^{n_1} \times \cdots \times Q^{n_p}}$$

Example: DRS for SOCP

- **second-order cone:**

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

- DRS operator has the form

$$T = \text{linear} \circ \mathbf{proj}_{Q^{n_1} \times \cdots \times Q^{n_p}}$$

- 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

$$T = \text{linear} \circ \mathbf{proj}_{Q^{n_1} \times \cdots \times Q^{n_p}}$$

- CF is trivial if all cones are small

- now, consider a big cone; property:

$$\mathbf{proj}_{Q^n}(x) = (\alpha x_1, \beta x_2, \ldots, \beta x_n)$$

where α, β 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

$$T = \text{linear} \circ \mathbf{proj}_{Q^{n_1} \times \cdots \times Q^{n_p}}$$

- CF is trivial if all cones are small

- now, consider a big cone; property:

$$\mathbf{proj}_{Q^n}(x) = (\alpha x_1, \beta x_2, \ldots, \beta x_n)$$

  where α, β depend on x_1 and $\gamma := \|(x_2, \ldots, x_n)\|_2$

- given γ and updating x_i, refreshing γ 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

$$T = \text{linear} \circ \mathbf{proj}_{Q^{n_1} \times \cdots \times Q^{n_p}}$$

- CF is trivial if all cones are small

- now, consider a big cone; property:

$$\mathbf{proj}_{Q^n}(x) = (\alpha x_1, \beta x_2, \ldots, \beta x_n)$$

  where α, β depend on x_1 and $\gamma := \|(x_2, \ldots, x_n)\|_2$

- given γ and updating x_i, refreshing γ costs $O(1)$

- by maintaining γ, $\mathbf{proj}_{Q^n}$ 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 $i_k \in [m]$,

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

Fixed-point iterations

- **full update**

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

- **(block) coordinate update (CU)**: choose $i_k \in [m]$,

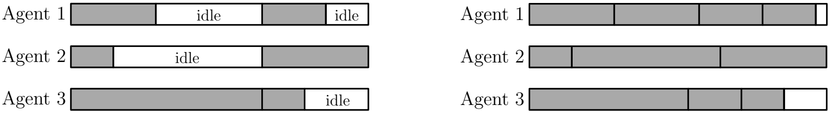
$$z_i^{k+1} = \begin{cases} z_i^k + \eta((Tz^k)_i - z_i^k), & \text{if } i = i_k \\ z_i^k, & \text{otherwise.} \end{cases}$$

- **parallel CU**: p agents choose $I_k \subset [m]$

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

- η 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 $i_k \subset [m]$,

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

  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

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** $\|\cdot\|_{w,\infty}$, 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

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[4]

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 $\eta_k \in [\epsilon, \frac{1}{2m^{-1/2}\tau+1})$. Then, $x^k \rightharpoonup x^* \in \mathrm{Fix}T$ almost surely.

[4] Peng-Xu-Yan-Y. SISC'16

ARock convergence[4]

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 $\eta_k \in [\epsilon, \frac{1}{2m^{-1/2}\tau+1})$. Then, $x^k \rightharpoonup x^* \in \mathrm{Fix}T$ almost surely.

consequence:

- **no sync at least until using $O(\sqrt{m})$ agents**
- sharp when $\tau \ll m$

[4] Peng-Xu-Yan-Y. SISC'16

Optimization and fixed-point examples

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

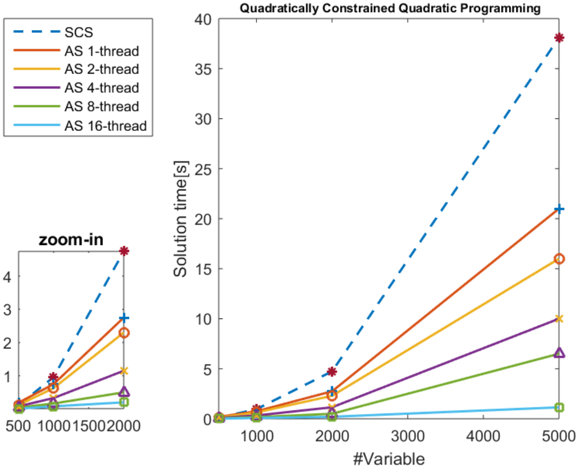
Applications

Convex Optimization Problem	Setup	ARock Iteration
Smooth minimization: min $f(x)$	∇f is L-Lipschitz, $\nabla f = \begin{pmatrix} \nabla f_1 \\ \vdots \\ \nabla f_m \end{pmatrix}$	$x_{i_k}^{k+1} \leftarrow x_{i_k}^k - \frac{2\eta^k}{L} \nabla f_{i_k}(\hat{x}^k)$
Constrained minimization: min $f(x)$ subject to $\ell \leq x \leq u$	same as above	$x_{i_k}^{k+1} \leftarrow x_{i_k}^k - \eta^k \Big(\hat{x}_{i_k}^k - \mathrm{Proj}_{[\ell_{i_k}, u_{i_k}]}\big(\hat{x}_{i_k}^k - \frac{2}{L}\nabla f_{i_k}(\hat{x}^k)\big) \Big)$
Composite minimization (ERM model): min $f(x) + g(x)$	same as above, plus $g(x) = \sum_{i=1}^m g_i(x_i)$	$x_{i_k}^{k+1} \leftarrow x_{i_k}^k - \eta^k \Big(\hat{x}_{i_k}^k - \mathrm{prox}_{\frac{2}{L} g_i}\big(\hat{x}_{i_k}^k - \frac{2}{L}\nabla f_{i_k}(\hat{x}^k)\big) \Big)$
Kernel SVM: $\min_s \frac{1}{2} s^T Q s - e^T s$ subject to $\sum_i y_i s_i = 0$, $0 \leq s_i \leq C,\ \forall i$	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	See the last equation in [20, Section 5.2.1], and apply it with damping η^k
Linear System: Solve $Ax = b$	A is symmetric positive definite, $\begin{pmatrix} - A_1 - \\ \vdots \\ - A_m - \end{pmatrix} x = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$	$x_{i_k}^{k+1} \leftarrow x_{i_k}^k - \left(\frac{2\eta^k}{M}\right)\left(A_{i_k}\hat{x}^k + b_{i_k}\right)$
Linear System: Solve $Ax = b$	$A = D + R$ where D is diagonal, M off-diagonal, $\rho\big(-D^{-1}R\big) \leq 1$	$x_{i_k}^{k+1} \leftarrow x_{i_k}^k - \eta^k \big((I + D^{-1}M)\hat{x}^k - D^{-1}b \big)_{i_k}$

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[5]



[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

$$p^\star = \min \ c^T x \quad \text{subject to } \underbrace{Ax = b}_{x \in L}, \ x \in K$$

K is a closed convex cone

Conic programming

$$p^\star = \min \ c^T x \quad \text{subject to } \underbrace{Ax = b}_{x \in L}, \ x \in K$$

K is a closed convex cone

- every problem falls in exactly one of the **seven cases**:

  1) $p^\star$ finite: 1a) has PD sol pair, 1b) only P sol, 1c) no P sol

Conic programming

$$p^\star = \min\ c^T x \quad \text{subject to } \underbrace{Ax = b}_{x \in L},\ x \in K$$

K is a closed convex cone

- every problem falls in exactly one of the **seven cases**:

  1) $p^\star$ finite: 1a) has PD sol pair, 1b) only P sol, 1c) no P sol

  2) $p^\star = -\infty$: 2a) has improving dir, 2b) no improving dir

Conic programming

$$p^\star = \min\ c^T x \quad \text{subject to } \underbrace{Ax = b}_{x \in L},\ x \in K$$

K is a closed convex cone

- every problem falls in exactly one of the **seven cases**:

  1) $p^\star$ finite: 1a) has PD sol pair, 1b) only P sol, 1c) no P sol

  2) $p^\star = -\infty$: 2a) has improving dir, 2b) no improving dir

  3) $p^\star = +\infty$: 3a) $\mathrm{dist}(L, K) > 0 \Leftrightarrow$ has separating hyperplane
     3b) $\mathrm{dist}(L, K) = 0 \Leftrightarrow$ no strict separating hyperplane

- (nearly) pathological cases fail existing solvers

Example 1

- **3-variable problem:**

$$\text{minimize } x_1 \quad \text{subject to } x_2 = 1,\ 2x_2x_3 \geq x_1^2.$$

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

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

[6] $p^\star = -\infty$, 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:**

$$\text{minimize } \; x_1 \quad \text{subject to } x_2 = 1, \; 2x_2x_3 \geq x_1^2.$$

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

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

- **classification:** (2b)
  - feasible
  - unbounded[6]
  - no improving direction[7]

---

[6] $p^\star = -\infty$, 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:**

$$\text{minimize } x_1 \quad \text{subject to } x_2 = 1,\ 2x_2x_3 \geq x_1^2.$$

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

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

- **classification:** (2b)
  - feasible
  - unbounded[6]
  - no improving direction[7]

- **solver results:**
  - SDPT3: "Failed", $p^\star$ no reported
  - SeDuMi: "Inaccurate/Solved", $p^\star = -175514$
  - Mosek: "Inaccurate/Unbounded", $p^\star = -\infty$

[6] $p^\star = -\infty$, 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:**

$$\text{minimize } 0 \quad \text{subject to } \underbrace{\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{x \in L}, \ \underbrace{x_3 \geq \sqrt{x_1^2 + x_2^2}}_{x \in K}.$$

[8] $x \in L$ imply $x = [1, -\alpha, \alpha]^T$, $\alpha \in \mathbb{R}$, which always violates the second-order cone constraint.
[9] $\mathrm{dist}(L, K) \leq \|[1, -\alpha, \alpha] - [1, -\alpha, (\alpha^2 + 1)^{1/2}]\|_2 \to 0$ as $\alpha \to \infty$.

Example 2

- **3-variable problem:**

$$\text{minimize } 0 \quad \text{subject to } \underbrace{\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{x \in L}, \; \underbrace{x_3 \geq \sqrt{x_1^2 + x_2^2}}_{x \in K}.$$

- **classification:** (3b)
  - infeasible[8], $L \cap K = \emptyset$
  - $\mathrm{dist}(L, K) = 0$ [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] $\mathrm{dist}(L, K) \leq \|[1, -\alpha, \alpha] - [1, -\alpha, (\alpha^2 + 1)^{1/2}]\|_2 \to 0$ as $\alpha \to \infty$.

Example 2

- **3-variable problem:**

$$\text{minimize } 0 \quad \text{subject to } \underbrace{\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{x \in L}, \ \underbrace{x_3 \geq \sqrt{x_1^2 + x_2^2}}_{x \in K}.$$

- **classification:** (3b)
  - infeasible[8], $L \cap K = \emptyset$
  - $\operatorname{dist}(L, K) = 0$ [9]
  - no strict separating hyperplane

- **solver results:**
  - SDPT3: "Infeasible", $p^\star = \infty$
  - SeDuMi: "Solved", $p^\star = 0$
  - Mosek: "Failed", $p^\star$ not reported

[8] $x \in L$ imply $x = [1, -\alpha, \alpha]^T$, $\alpha \in \mathbb{R}$, which always violates the second-order cone constraint.
[9] $\operatorname{dist}(L, K) \leq \|[1, -\alpha, \alpha] - [1, -\alpha, (\alpha^2 + 1)^{1/2}]\|_2 \to 0$ 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

$$z^k - z^{k+1} \to v = \mathrm{Proj}_{\overline{\mathrm{ran}(I-T)}}(\mathbf{0})$$

Our analysis results (Liu-Ryu-Y.'17)

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

Our analysis results (Liu-Ryu-Y.'17)

- **rate of convergence:** $\|z^k - z^{k+1}\| \leq \|v\| + \epsilon + O(\frac{1}{\sqrt{k+1}})$
- **deciphered** $\mathrm{Proj}_{\overline{\mathrm{ran}(I-T)}}$

Our analysis results (Liu-Ryu-Y.'17)

- **rate of convergence:** $\|z^k - z^{k+1}\| \le \|v\| + \epsilon + O(\frac{1}{\sqrt{k+1}})$
- **deciphered** $\mathrm{Proj}_{\overline{\mathrm{ran}(I-T)}}$
- **a workflow** running three similar DRS, differ by only constants:
  1) DRS
  2) feasibility DRS with $c = \mathbf{0}$
  3) boundedness DRS with $b = \mathbf{0}$
- most pathological cases are **identified**

Our analysis results (Liu-Ryu-Y.'17)

- **rate of convergence:** $\|z^k - z^{k+1}\| \leq \|v\| + \epsilon + O(\frac{1}{\sqrt{k+1}})$
- **deciphered** $\mathrm{Proj}_{\overline{\mathrm{ran}(I-T)}}$
- **a workflow** running three similar DRS, differ by only constants:
  1) DRS
  2) feasibility DRS with $c = \mathbf{0}$
  3) boundedness DRS with $b = \mathbf{0}$
- most pathological cases are **identified**
- **compute an improving direction** if one exists

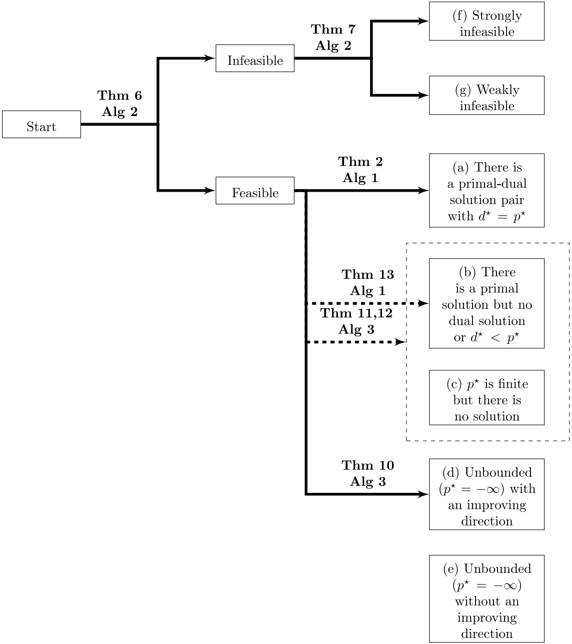
Our analysis results (Liu-Ryu-Y.'17)

- **rate of convergence:** $\|z^k - z^{k+1}\| \leq \|v\| + \epsilon + O(\frac{1}{\sqrt{k+1}})$
- **deciphered** $\mathrm{Proj}_{\overline{\mathrm{ran}(I-T)}}$
- **a workflow** running three similar DRS, differ by only constants:
  1) DRS
  2) feasibility DRS with $c = \mathbf{0}$
  3) boundedness DRS with $b = \mathbf{0}$
- 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:** $\|z^k - z^{k+1}\| \leq \|v\| + \epsilon + O(\frac{1}{\sqrt{k+1}})$
- **deciphered** $\mathrm{Proj}_{\overline{\mathrm{ran}(I-T)}}$
- **a workflow** running three similar DRS, differ by only constants:
  1) DRS
  2) feasibility DRS with $c = \mathbf{0}$
  3) boundedness DRS with $b = \mathbf{0}$
- 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



Infeasible SDP test set (Liu-Pataki'17)

	$m = 10$		$m = 20$	
	Clean	Messy	Clean	Messy
SeDuMi	0	0	1	0
SDPT3	0	0	0	0
Mosek	0	0	11	0
PP[10]+SeDuMi	100	0	100	0

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

[10] PreProcessing by Permenter-Parilo'14

Identify weakly infeasible SDPs

	$m = 10$		$m = 20$	
	Clean	Messy	Clean	Messy
Proposed	100	21	100	99

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

our percentage is way much better!

Identify strongly infeasible SDPs

	$m = 10$		$m = 20$	
	Clean	Messy	Clean	Messy
Proposed	100	100	100	100

(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