Level-Set Geometry and the Performance of Restarted-PDHG for Conic LP

Zikai Xiong and Robert Freund

Zikai Xiong (MIT OR Center)

Robert Freund (MIT Sloan)

Paper available on arXiv

ALGOPT2024 UCLouvain Belgium August 2024

Israel (near Acre) in 2016

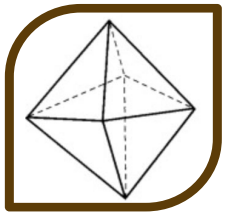
History of Linear Optimization ("LO" or "LP")

1947

Simplex Method

[George Dantzig, 1947]

75+ years ago

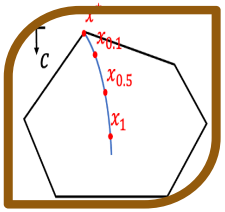


1984

Interior Point Method

[Narendra Karmarkar, 1984]

40 years ago



2024

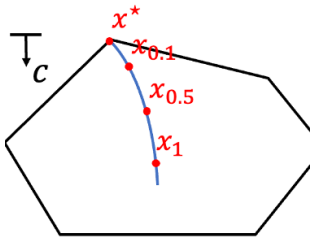
Huge-scale Method(s)?

The next method(s) designed to address huge-scale LP

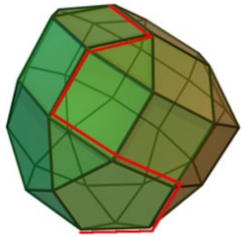
Recent Advances on Huge-Scale Commercial LP Solvers

Classic methods

Interior-point method



Simplex method



First-order methods

Primal-Dual Hybrid Gradient (aka PDHG, aka Chambolle-Pock method)

- Solves huge-scale problems
- Benefits from modern computational architecture (such as GPU)
- The method embedded in Google OR-Tools, COPT, and FICO Xpress

Google

COPT Cardinal Optimizer
State-of-art commercial optimization solver

NVIDIA®
The world's largest GPU company

FICO®
State-of-art commercial optimization solver

2021
Google OR-Tools proposed and implemented a distributed first-order LP method.
(2024 Beale-Orchard-Hays Prize)

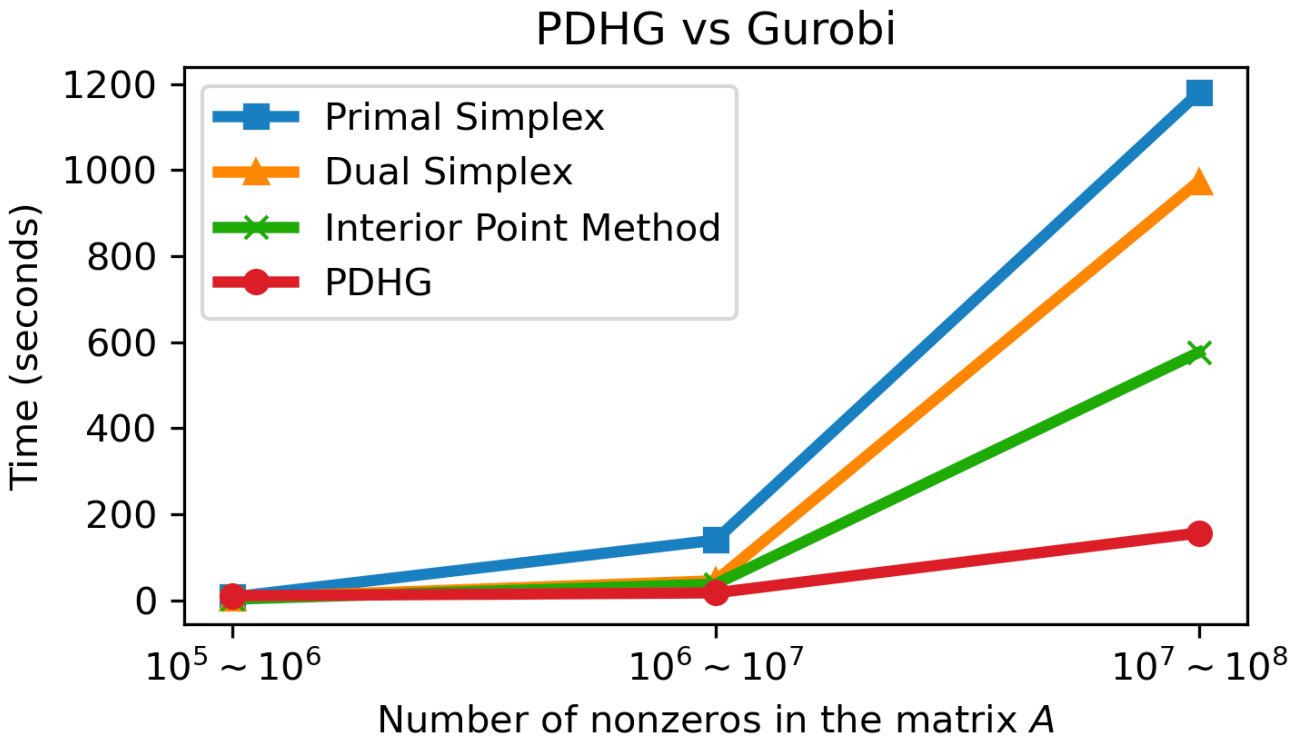
Feb 12, 2024
COPT embedded GPU-based LP method

Mar 20, 2024
NVIDIA announced (Likely cuOpt PDHG) a GPU-based LP solver

Apr 22, 2024
FICO Xpress embedded a matrix-factorization-free LP method

PDHG is very good at solving Large LP Instances

Geometric average runtime on problems of different scale
from LP relaxations of MIPLIB 2017



Termination tolerance is 10^{-4}

Data from:
Lu, H., & Yang, J. (2023). cuPDLP. jl: A GPU implementation of restarted primal-dual hybrid gradient for linear programming in Julia. arXiv preprint arXiv:2311.12180.

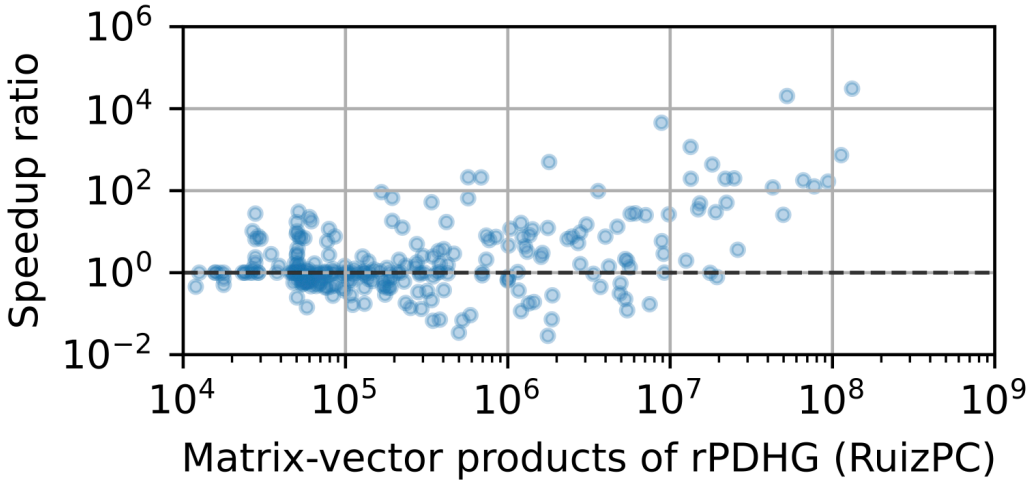
Huge-Scale LP Research

- SCS: Operator splitting/ADMM [O'Donoghue, Chu, Parikh, Boyd, 2016]
- ABIP+: ADMM-based interior-point method
  [Lin, Ma, Ye, Zhang, 2021] & [Deng, et al., 2022]
- Semi-smooth Newton augmented Lagrangian [Li, Sun, Toh, 2020]
- **Primal-Dual Hybrid Gradient (PDHG)** with restarts, applied directly to the primal-dual saddle point problem [Applegate, Hinder, Lu, Lubin, 2023] & [Applegate, et al., 2021] **(2024 Beale-Orchard-Hays Prize)**
- **GPU implementations** of PDHG in Julia and C
  [Lu and Yang, 2023] & [Lu, et al., 2023]
- **Guarantees for PDHG for LP** using "Limiting Error Ratios" and LP Sharpness [Xiong and F 2023]
- **Guarantees for PDHG for CLP** – using level-set geometry
  [Xiong and F 2024]

Sneak Preview:

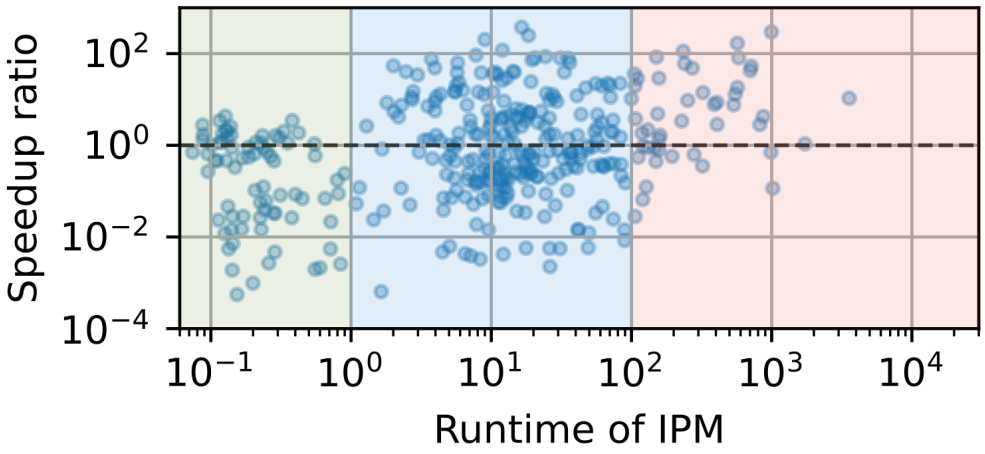
Distribution of Speedups of our method rPDHG-AHR

Speedups compared with restarted-PDHG (rPDHG) with PDLP's rescaling



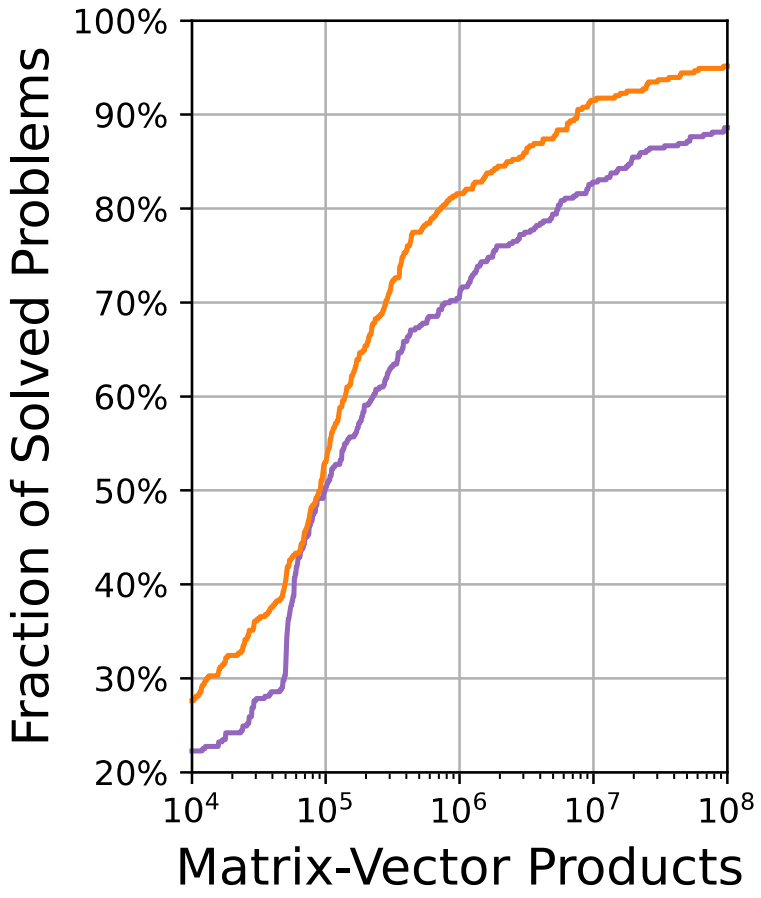
Speedups compared with a "home-grown" IPM

(Predictor-corrector path-following interior-point method in Nocedal and Wright *Numerical Optimization* (2006))



Sneak Preview:

Performance Comparison on MIPLIB 2017



rPDHG-AHR
rPDHG with our adaptive rescaling

rPDHG (RuizPC)
rPDHG with heuristic Ruiz/PC rescaling (same with "PDLP")

Conic Linear Optimization ("CLO" or "CLP")

CLP in standard form

(primal)

$$\begin{aligned} \min\ & c^\top x \\ \text{s.t. } & Ax = b \\ & x \in \mathcal{K} \end{aligned}$$

(dual)

$$\begin{aligned} \max\ & b^\top y \\ \text{s.t. } & c - A^\top y \in \mathcal{K}^* \end{aligned}$$

Decision variables

- $x \in R^n$ (for primal problem)
- $y \in R^m$ (for dual problem)

CLP saddlepoint formulation

$$\min_{x \in \mathcal{K}} \max_{y} \ c^\top x - y^\top A x + b^\top y$$

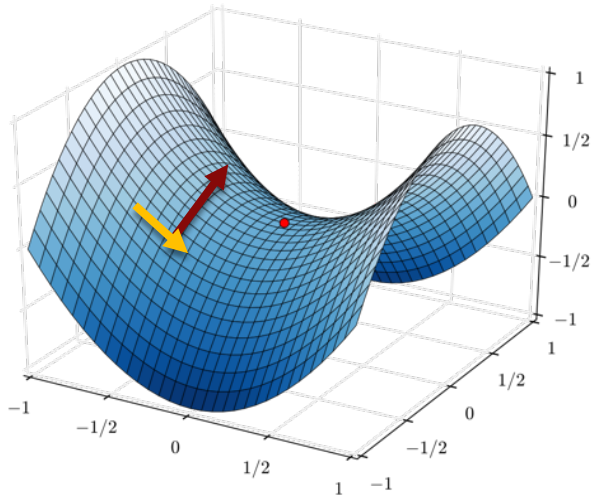
Yurii Nesterov research connection

- Conic formulation of convex optimization
- Strong focus on the cone variables
- Three cones (and their cross-products):
  - Nonnegative orthant
  - Second-order cone
  - PSD cone

Primal-Dual Hybrid Gradient Method (PDHG)

LP in Saddlepoint Problem Form

$$\min_{x \geq 0} \max_{y} \; c^\top x - y^\top A x + b^\top y$$



Convex-Concave Saddlepoint Problem

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \; f(x) + \langle y, Mx \rangle - g(y)$$

Convex sets $\mathcal{X}, \mathcal{Y}$

Convex functions f, g

Can diverge

Naïve updating scheme

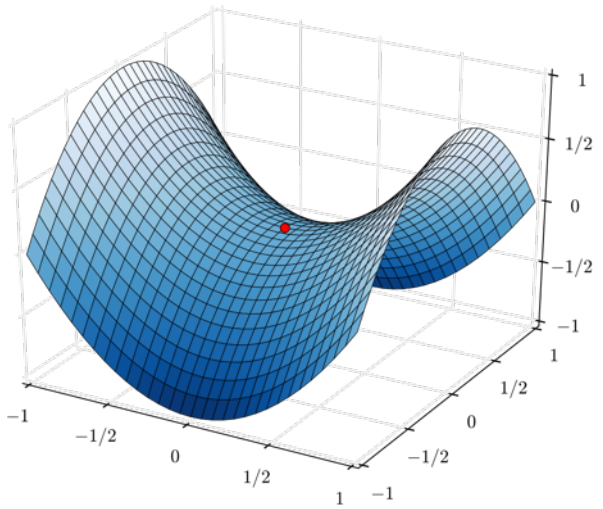
$$x^{t+1} \leftarrow \arg\min_{x \in \mathcal{X}} \; f(x) + \langle y^t, Mx \rangle + \frac{1}{2\tau} \|x - x^t\|^2$$

$$y^{t+1} \leftarrow \arg\min_{y \in \mathcal{Y}} \; -\langle y, Mx^{t+1} \rangle + g(y) + \frac{1}{2\sigma} \|y - y^t\|^2$$

Primal-Dual Hybrid Gradient Method (PDHG)

LP in Saddlepoint Problem Form

$$\min_{x \geq 0} \max_{y} \; c^\top x - y^\top A x + b^\top y$$



Convex-Concave Saddlepoint Problem

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \; f(x) + \langle y, Mx \rangle - g(y)$$

Convex sets $\mathcal{X}, \mathcal{Y}$

Convex functions f, g

PDHG

$$x^{t+1} \leftarrow \arg\min_{x \in \mathcal{X}} \; f(x) + \langle y^t, Mx \rangle + \frac{1}{2\tau} \|x - x^t\|^2$$

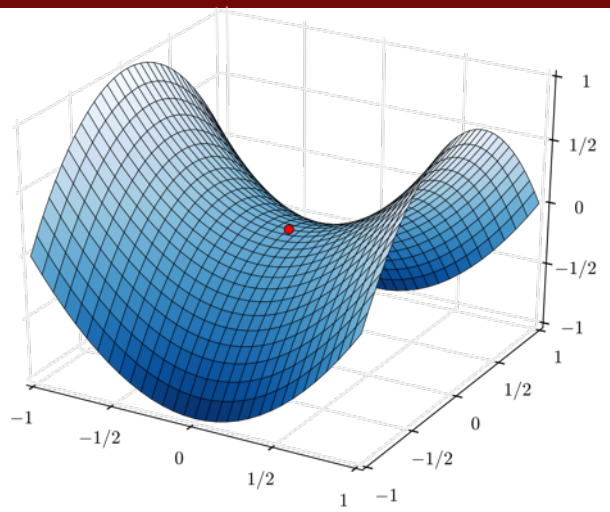
$$y^{t+1} \leftarrow \arg\min_{y \in \mathcal{Y}} \; -\langle y, M(2x^{t+1} - x^t) \rangle + g(y) + \frac{1}{2\sigma} \|y - y^t\|^2$$

Converges

Primal-Dual Hybrid Gradient for LP

PDHG

$$x^{k+1} \leftarrow \left(x^k + \tau A^\top y^k - \tau c\right)^+$$
$$y^{k+1} \leftarrow y^k - \sigma A\left(2x^{k+1} - x^k\right) + \sigma b$$



- Only requires matrix-vector multiplications
- "Simple" $O(1/t)$ convergence when τ and σ satisfy:

$$\tau \cdot \sigma \leq \frac{1}{\sigma_{max}(A)^2}$$

- For LP, restarts based on average iterates yield linear convergence [Applegate, Hinder, Lu, Lubin, 2023]

Motivation for Restarts for PDHG: "Visualization"

$$\min_x \max_y x \cdot y$$

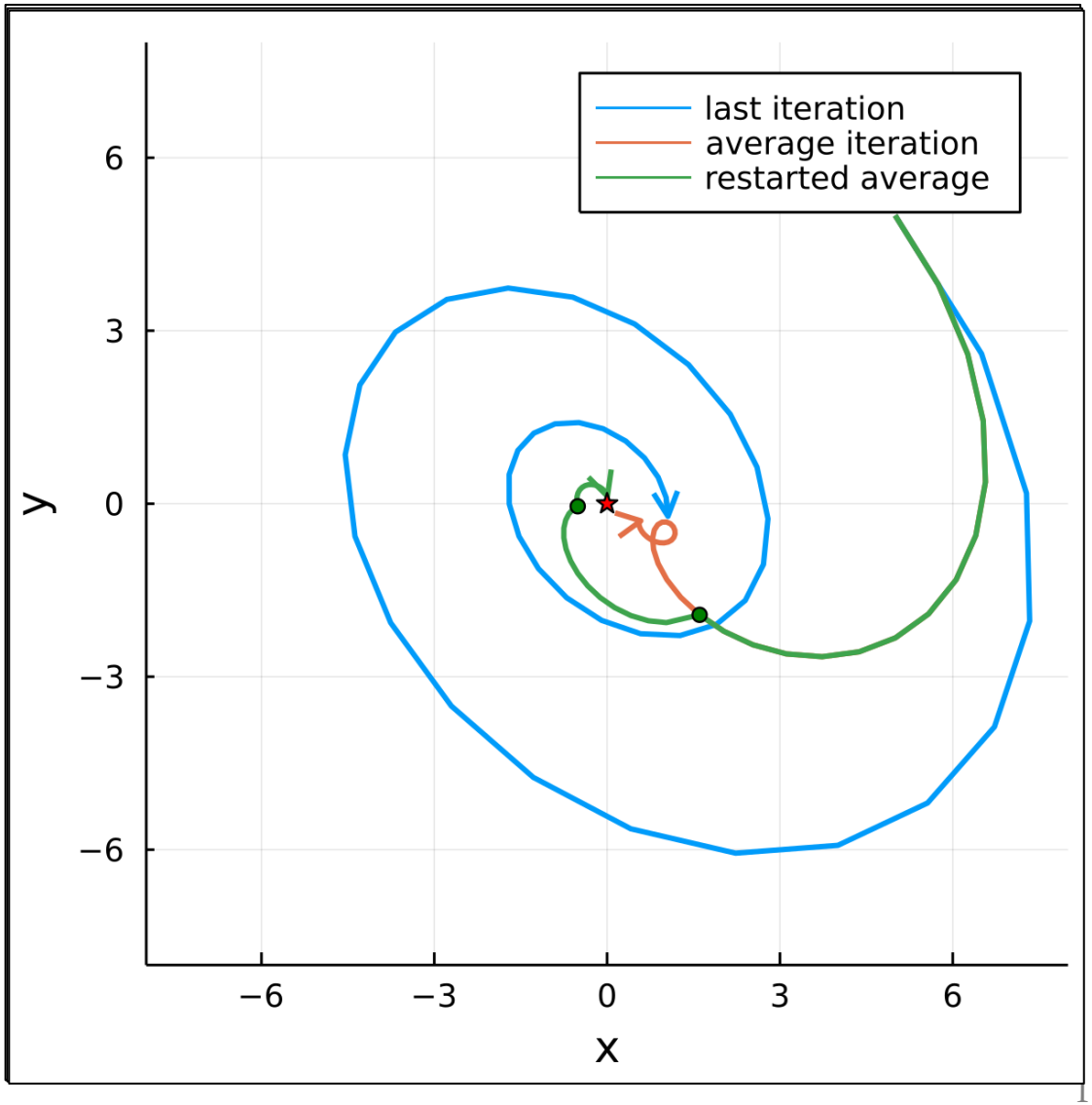


*figure courtesy Haihao Lu

Current theory for rPDHG for LP

Pros:

- rPDHG has global linear convergence for LP instances
- rPDHG is faster than standard PDHG in both theory and practice

Cons:

- Current theoretical complexity is quite loose and hard to compute/validate

  We seek an iteration bound that is both tighter and easier to analyze

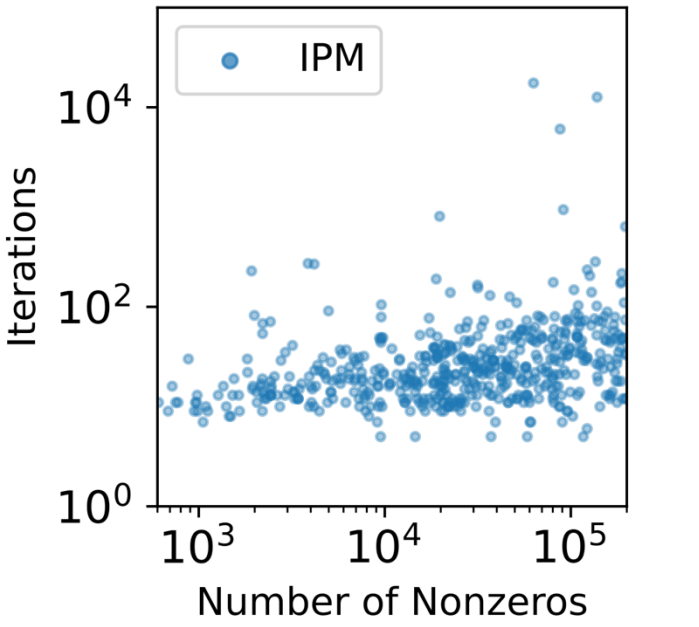
- Some seemingly "easy" problems are hard for rPDHG

  We seek to understand what properties of these "easy" problems make them hard

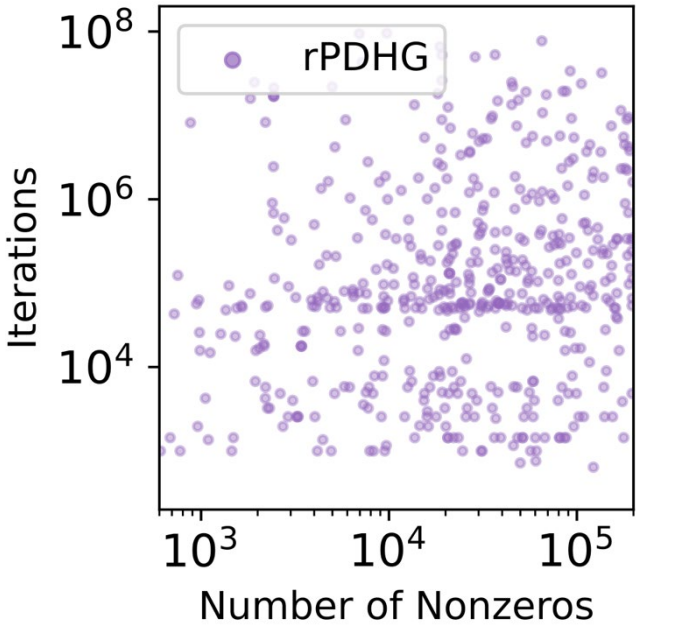
Performance of rPDHG

- rPDHG uses more iterations than IPM
  makes sense, as a first-order method …
- Some small instances require a very large number of iterations
  a real challenge for rPDHG

IPM Iterations needed for
LP relaxations from MIPLIB 2017



PDHG iterations
LP relaxations from MIPLIB 2017



A seemingly easy LP instance

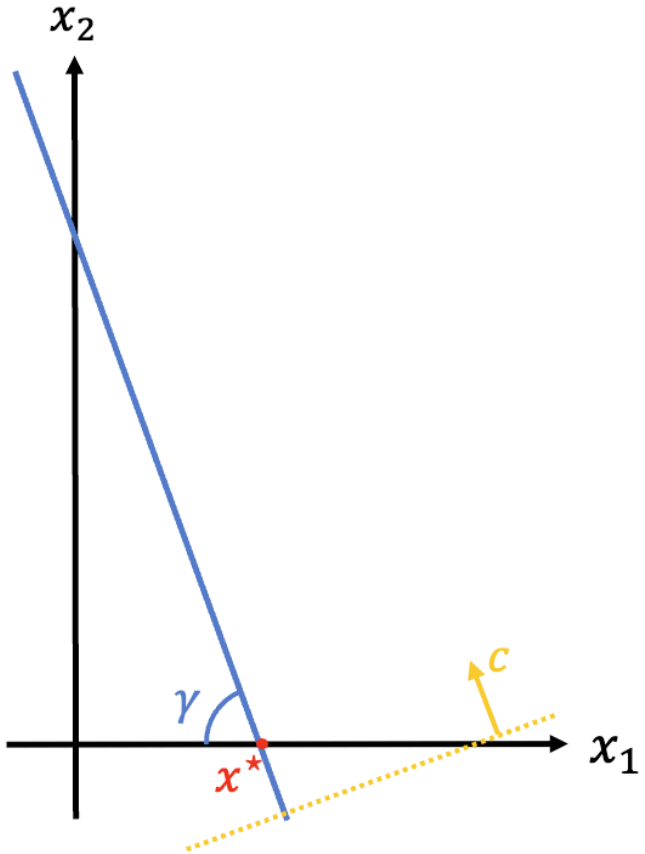
For $\gamma \in (0, \pi/2)$ define:

$$P(\gamma): \quad \min_{x_1, x_2} \; -\cos(\gamma)x_1 + \sin(\gamma)x_2$$

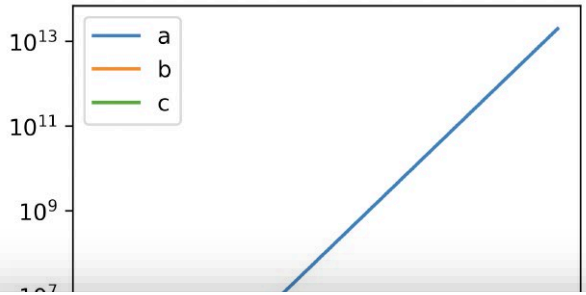
$$\text{s.t.} \; \sin(\gamma)x_1 + \cos(\gamma)x_2 = 1,$$

$$x_1 \geq 0, \; x_2 \geq 0$$

It is always easy for simplex method and interior-point method to solve $P(\gamma)$



P(γ) is hard to solve using rPDHG



Theoretical iteration complexity of [Applegate, Hinder, Lu, & Lubin, 2023]

What condition numbers drive the performance of rPDHG?

Can we improve these condition numbers and so improve computational performance in theory/practice?

When γ is small, rPDHG requires at least 100,000 iterations.
What **conditions** of P(γ) makes it so hard for rPDHG?

Condition numbers describe the state/condition of the problem

Condition numbers related to Level-set Geometry

Reformulation of Standard Form CLP

CLP in standard form

(Primal)

$$\min\ c^\top x$$
$$\text{s.t. } Ax = b$$
$$x \in \mathcal{K}$$

(Dual)

$$\max\ b^\top y$$
$$\text{s.t. } A^\top y + s = c$$
$$s \in \mathcal{K}^\star$$

Assumptions (similar as in IPMs)

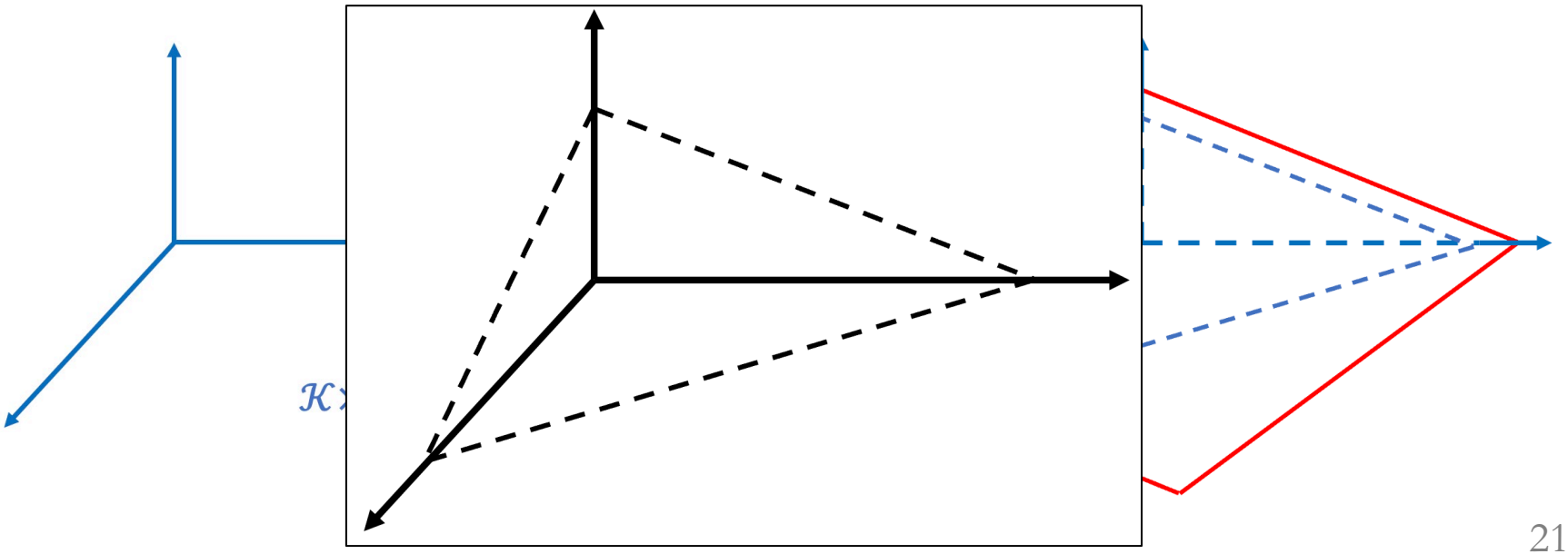
1. Strict feasibility of P and D (though not necessary in context of computation)
2. Rows of A are linearly independent
   - y and s correspond one-to-one
3. Objective vector $c \in \text{Null}(A)$
   - $Ac = 0$
   - not essential, but keeps things simpler

The feasible region in the space of (x, s)

$\min\ c^\top x$	$\max\ b^\top y$
s. t. $Ax = b$	s. t. $A^\top y + s = c$
$x \in \mathcal{K}$	$s \in \mathcal{K}^\star$

(x, s) lies in the cone $\mathcal{K} \times \mathcal{K}^*$
For example, the nonnegative orthant for LP

$Ax = b, \exists\ y$ s.t. $A^\top y + s = c$
(x, s) lies in an affine subspace



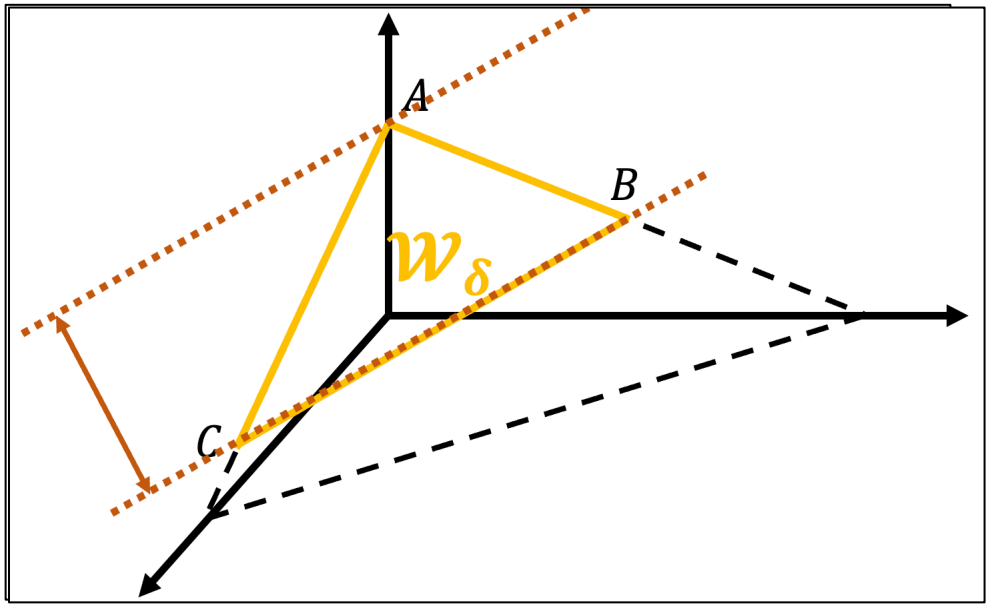
Primal-Dual Sublevel Sets

(x, s) is feasible

- The duality gap is a **linear function** of (x, s)
- The optimal solution set is $\mathcal{W}^\star \equiv$ set of optimal (x, s)
- The optimal solution is the point A in the figure

$$\mathcal{W}_\delta := \left\{ w := (x, s) \;\middle|\; \begin{array}{c} Ax = b, \exists\, y \text{ s.t. } A^T y + s = c \\ x \in \mathcal{K}, s \in \mathcal{K}^* \\ c^T x - b^T (AA^T)^{-1} A(c - s) \leq \delta \end{array} \right\}$$

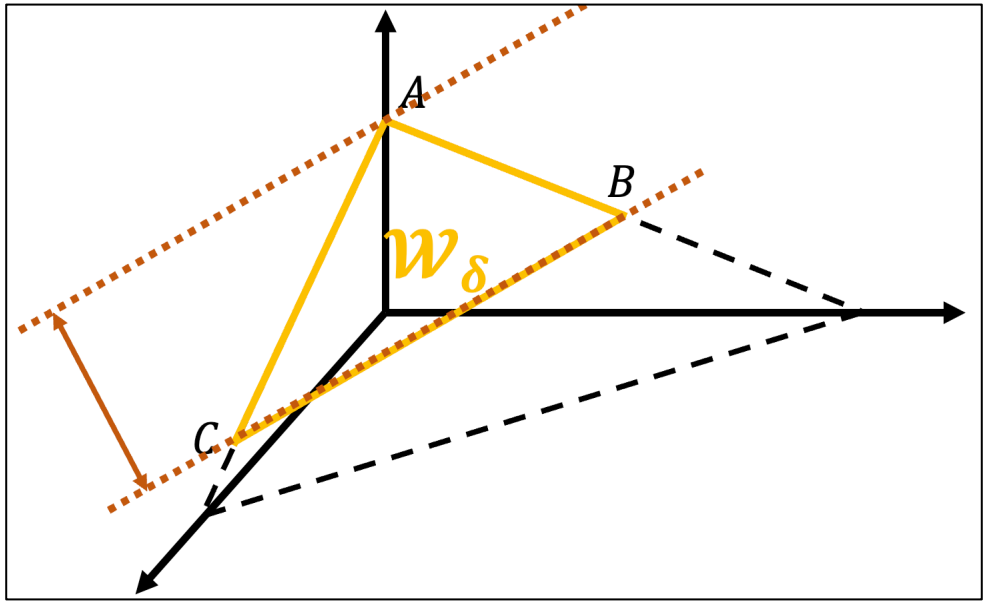
duality gap is at most δ



Note: $\mathcal{W}_0 = \mathcal{W}^\star$

Three Geometric Condition Numbers

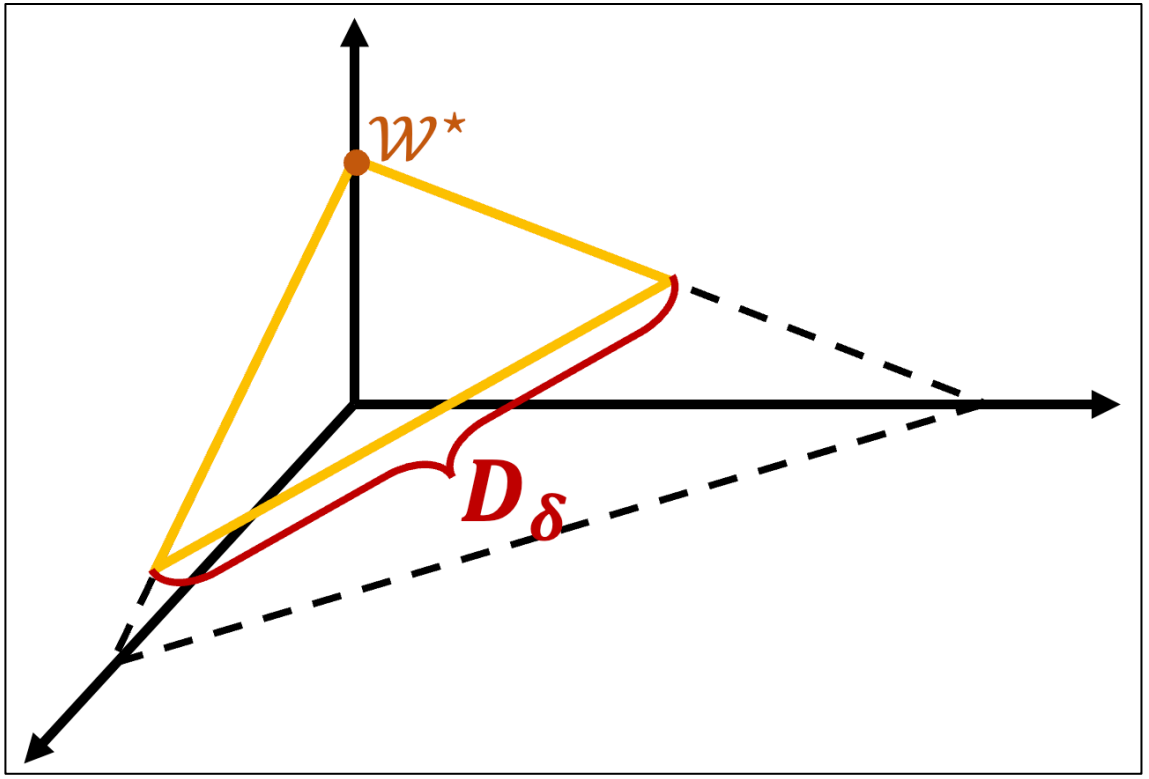
1. D_δ -- the diameter of w_δ
2. r_δ -- conic radius of w_δ
3. d_δ^H -- Haussdorf distance between w_δ and $w^\star$



D_δ: Diameter of δ-sublevel set $\mathcal{W}_\delta$

$$D_\delta := \max_{\bar{w}, \hat{w} \in \mathcal{W}_\delta} \|\bar{w} - \hat{w}\|$$

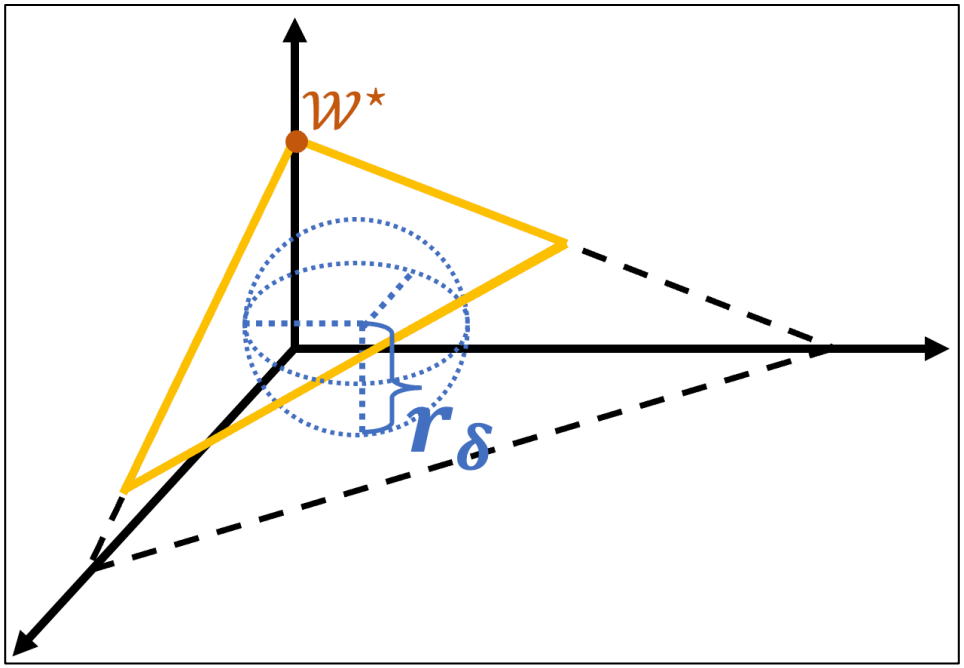
(D_δ is smaller when δ is small)



r_δ: "Conic Radius" of $\mathcal{W}_\delta$

$$r_\delta := \max_{r \geq 0,\, w \in \mathcal{W}_\delta} r$$
$$\text{s.t.} \quad B_w(r) \subset \mathcal{K} \times \mathcal{K}^*$$

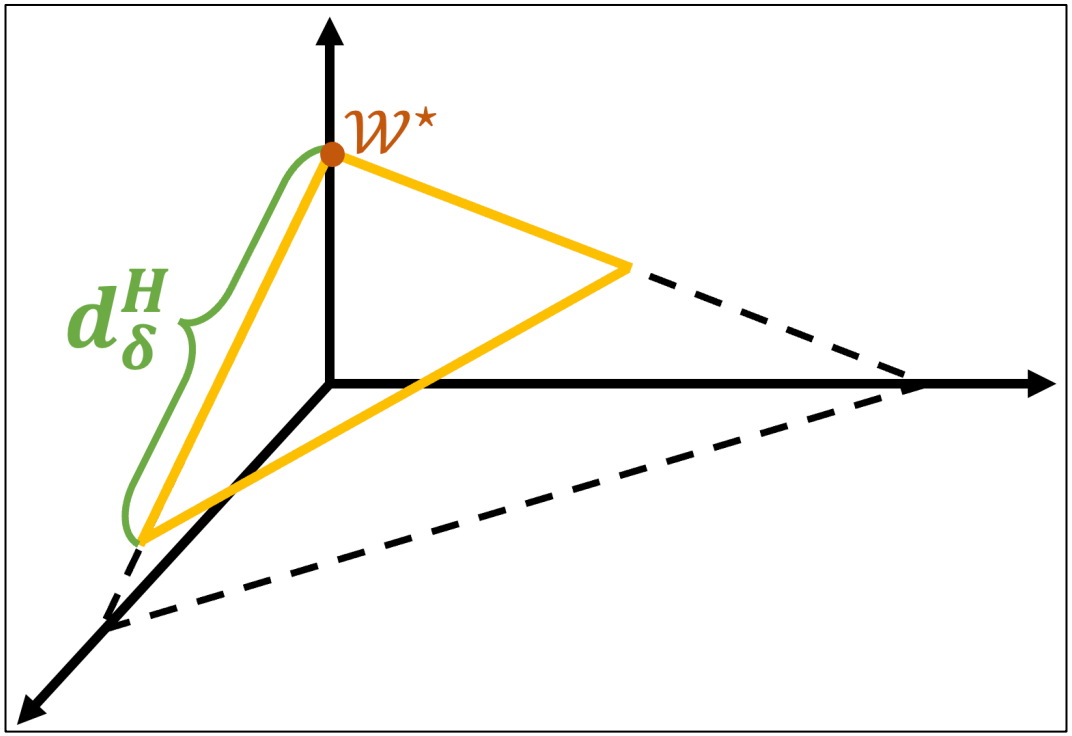
r_δ is the radius of the maximum ball inscribed in $\mathcal{K} \times \mathcal{K}^\star$ and centered at a point in $\mathcal{W}_\delta$



d_δ^H: Hausdorff distance from $\mathcal{W}_\delta$ to $\mathcal{W}^\star$

$$d_\delta^H := \max_{w \in \mathcal{W}_\delta} \mathrm{Dist}(w, \mathcal{W}^\star)$$

d_δ^H is small when δ is small



Convergence Guarantee for rPDHG

Target: ε-optimal solution

(x, s) is an $\boldsymbol{\varepsilon}$-optimal solution if:

- distance to each type of constraint is no larger than ε , and
- the duality gap is not larger than ε

(x, s) is an $\boldsymbol{\varepsilon}$-optimal solution if:

- $\mathrm{Dist}(x, \{x | Ax = b\}) \leq \varepsilon$
- $\mathrm{Dist}(x, \mathcal{K}) \leq \varepsilon$
- $\mathrm{Dist}(s, \{s | \exists\ y\ \ \mathrm{s.t.}\ A^\top y + s = c\}) \leq \varepsilon$
- $\mathrm{Dist}(s, \mathcal{K}^\star) \leq \varepsilon$
- $c^\top x - b^\top (AA^\top)^{-1} A(c - s) \leq \varepsilon$

A "Composite" Computational Guarantee for general **CLP**

Theorem: Starting from $(x^{0,0}, y^{0,0}) = (0,0)$, the number of PDHG iterations required to compute an $\boldsymbol{\varepsilon}$-optimal solution is upper bounded by:

$$\tilde{O}\left(\quad T_\delta := \boldsymbol{\kappa} \cdot \left[\frac{\boldsymbol{D_\delta}}{\boldsymbol{r_\delta}} \cdot \ln\left(\frac{1}{\boldsymbol{\varepsilon}}\right) + \boldsymbol{d_\delta^H} \cdot \frac{1 + \mathrm{Dist}(0, \mathcal{W}^\star)}{\boldsymbol{\varepsilon}} \right] \right)$$

for each $\delta > 0$.

Theorem: Starting from $(x^{0,0}, y^{0,0}) = (0,0)$, the number of PDHG iterations required to compute an $\boldsymbol{\varepsilon}$-optimal solution is upper bounded by:

$$\tilde{O}\left(\inf_{\delta > 0} \; T_\delta := \boldsymbol{\kappa} \cdot \left[\frac{\boldsymbol{D_\delta}}{\boldsymbol{r_\delta}} \cdot \ln\left(\frac{1}{\boldsymbol{\varepsilon}}\right) + \boldsymbol{d_\delta^H} \cdot \frac{1 + \mathrm{Dist}(0, \mathcal{W}^\star)}{\boldsymbol{\varepsilon}} \right] \right)$$

$\tilde{O}(\cdot)$ hides condition numbers and $\mathrm{Dist}(0, \mathcal{W}^\star)$ inside the logarithm term

A "Composite" Computational Guarantee for general **CLP**

Theorem: Starting from $(x^{0,0}, y^{0,0}) = (0,0)$, the number of PDHG iterations required to compute an $\boldsymbol{\varepsilon}$-optimal solution is upper bounded by:

$$\tilde{O}\left(\inf_{\delta>0} T_\delta := \boldsymbol{\kappa} \cdot \left[\frac{\boldsymbol{D_\delta}}{\boldsymbol{r_\delta}} \cdot \ln\left(\frac{1}{\boldsymbol{\varepsilon}}\right) + \boldsymbol{d_\delta^H} \cdot \frac{1 + \mathrm{Dist}(0, \mathcal{W}^\star)}{\boldsymbol{\varepsilon}}\right]\right)$$

Linear convergence part ($\boldsymbol{D_\delta}/\boldsymbol{r_\delta}$ might/not be large when δ is small)

Sublinear convergence part ($\boldsymbol{d_\delta^H}$ is small when δ is small)

$\boldsymbol{D_\delta}$: Diameter of $\mathcal{W}_\delta$

$\boldsymbol{r_\delta}$: Conic radius of $\mathcal{W}_\delta$

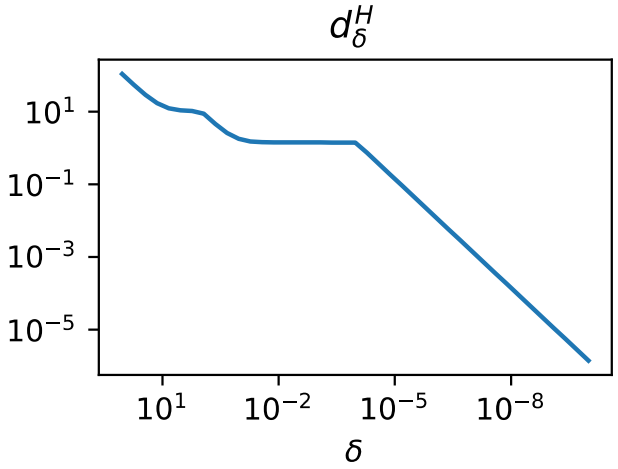
$\boldsymbol{d_\delta^H}$: Hausdorff distance from $\mathcal{W}_\delta$ to $\mathcal{W}^\star$

$\boldsymbol{\kappa} = \sigma_{\max}^+(A)/\sigma_{\min}^+(A)$ (the standard condition number of A)

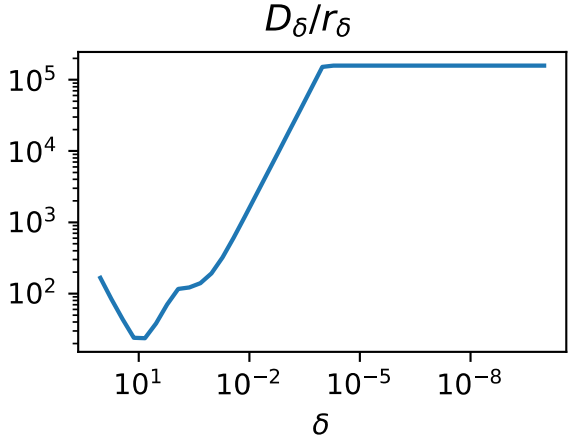
Key Observations, via a Small LP Example

$$\min_{x=(x_1,x_2,x_3)} \quad 0.20001 \cdot x_1 + x_2 + 1.0001 \cdot x_3$$
$$\text{s.t.} \quad -10x_1 + x_2 + x_3 = 1$$
$$x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0$$

- d_δ^H monotonically decreases as δ decreases
- $d_\delta^H = 0$ when $\delta = 0$



- The smallest D_δ / r_δ might still be large
- When the optimal solution is unique, $\lim_{\delta \searrow 0} D_\delta / r_\delta$ might/not be bounded (think LP under nondegeneracy)



A "Composite" Computational Guarantee for general **CLP**

Theorem: Starting from $(x^{0,0}, y^{0,0}) = (0,0)$, the number of PDHG iterations required to compute an $\boldsymbol{\varepsilon}$-optimal solution is upper bounded by:

$$\tilde{O}\left(\inf_{\delta>0} T_\delta := \boldsymbol{\kappa} \cdot \left[\frac{\boldsymbol{D_\delta}}{\boldsymbol{r_\delta}} \cdot \ln\left(\frac{1}{\boldsymbol{\varepsilon}}\right) + \boldsymbol{d_\delta^H} \cdot \frac{1 + \mathrm{Dist}(0, \mathcal{W}^\star)}{\boldsymbol{\varepsilon}}\right]\right)$$

- Linear convergence part
- Note that $\boldsymbol{D_\delta}/\boldsymbol{r_\delta}$ might/not be large when δ is small

- Sublinear convergence part
- Note $\boldsymbol{d_\delta^H}$ is small when δ is small

We can choose $\delta > 0$ that minimizes the bound ☺

$\boldsymbol{D_\delta}$: Diameter of $\mathcal{W}_\delta$

$\boldsymbol{r_\delta}$: Conic radius of $\mathcal{W}_\delta$

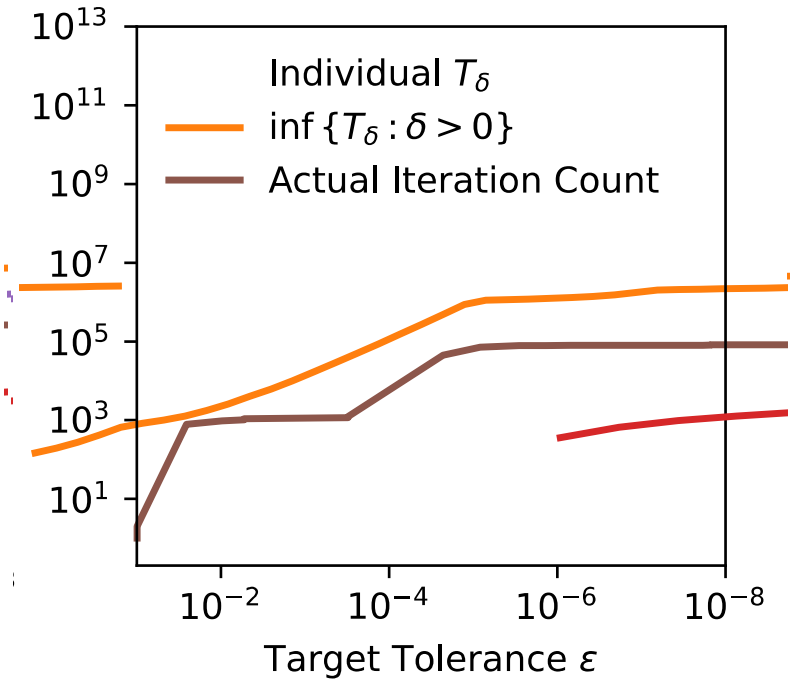
$\boldsymbol{d_\delta^H}$: Hausdorff distance between $\mathcal{W}_\delta$ and $\mathcal{W}^\star$

$\boldsymbol{\kappa} = \sigma^+_{\max}(A)/\sigma^+_{\min}(A)$

T_δ and $\inf\{T_\delta : \delta > 0\}$

$$\begin{aligned} \min_{x=(x_1,x_2,x_3)} \quad & 0.20001 \cdot x_1 + x_2 + 1.0001 \cdot x_3 \\ \text{s.t.} \quad & -10x_1 + x_2 + x_3 = 1 \\ & x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0 \end{aligned}$$

- Different δ yield different bounds T_δ
- For each tolerance ε, $\inf\{T_\delta : \delta > 0\}$ provides the best bound
- In the beginning, rPDHG converges sublinearly
- Since $\lim_{\delta \searrow 0} \frac{D_\delta}{r_\delta} < \infty$ and $\lim_{\delta \searrow 0} d_\delta^H = 0$, we eventually obtain linear convergence
- Validated by actual iteration count



A "Composite" Computational Guarantee for general **CLP**

Theorem: Starting from $(x^{0,0}, y^{0,0}) = (0,0)$, the number of PDHG iterations required to compute an $\boldsymbol{\varepsilon}$-optimal solution is bounded by:

$$\tilde{O}\left(\inf_{\delta>0} T_\delta := \boldsymbol{\kappa} \cdot \left[\frac{\boldsymbol{D_\delta}}{\boldsymbol{r_\delta}} \cdot \ln\left(\frac{1}{\boldsymbol{\varepsilon}}\right) + \boldsymbol{d_\delta^H} \cdot \frac{1 + \mathrm{Dist}(0, \mathcal{W}^\star)}{\boldsymbol{\varepsilon}}\right]\right)$$

Corollary: If there exists $\delta > 0$ whose δ-sublevel set satisfies:

- $\frac{D_\delta}{r_\delta}$ is small, and
- d_δ^H is small,

then rPDHG will converge faster. (If not, rPDHG might be slow…)

$\boldsymbol{D_\delta}$: Diameter of …
$\boldsymbol{r_\delta}$: Conic radius of …
$\boldsymbol{d_\delta^H}$: Hausdorff distance between…

Yurii Nesterov research connection

- Complexity bounds for algorithms that actually inform/explain the behavior of the method and the problem instance

"Improved" Linear Convergence of rPDHG (for **LP**)

Theorem: Starting from $(x^{0,0}, y^{0,0}) = (0,0)$, the number of PDHG iterations required to compute an $\boldsymbol{\varepsilon}$-optimal solution for **LP** is upper bounded by:

$$\tilde{O}\left(\boldsymbol{\kappa} \cdot \frac{\boldsymbol{D_{\bar{\gamma}}}}{\boldsymbol{r_{\bar{\gamma}}}} \cdot \ln\left(\frac{1}{\varepsilon}\right)\right)$$

$\bar{\gamma}$ is the optimality gap of the second-best extreme point solution:

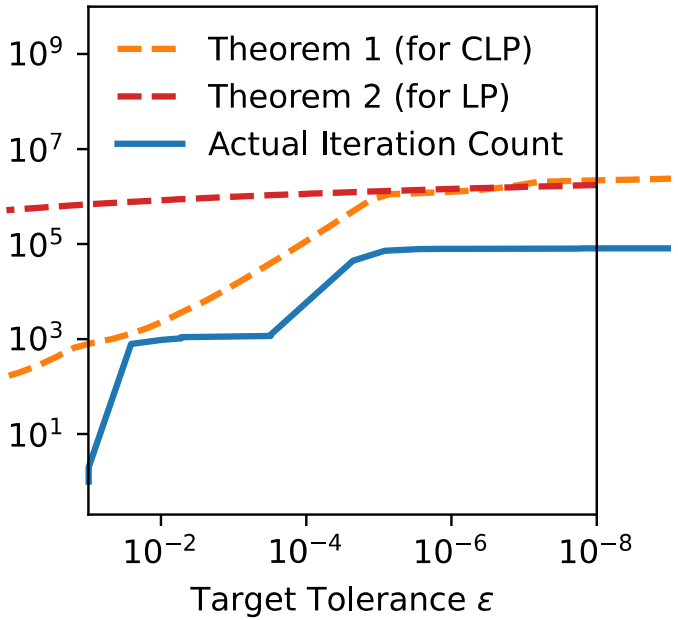
$\bar{\gamma}$: =min{Gap(w): w is a nonoptimal extreme point of the P/D feasible region}

Of course, $\bar{\gamma}$ might be exponentially small, in which case this bound will be inferior to the "composite" bound for CLP.

Pros and Cons of focusing on the linear convergence term

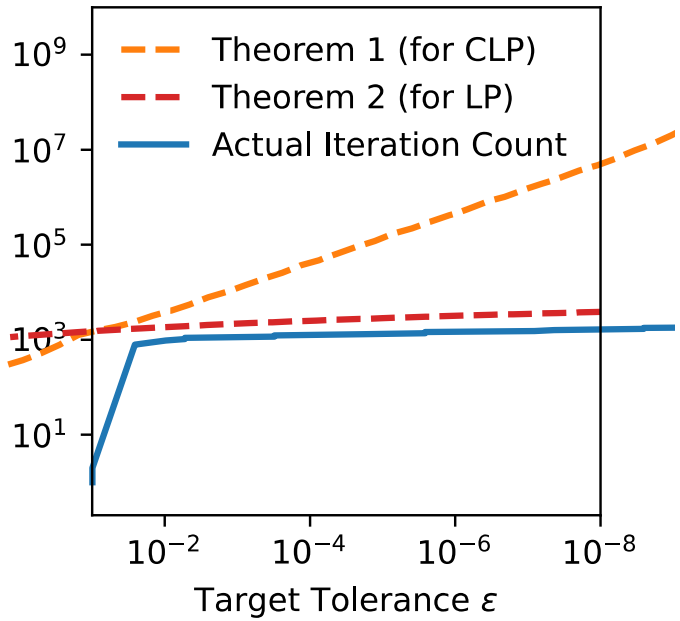
1. Linear convergence term might not reflect the early stage performance of PDHG

$$\min_{x=(x_1,x_2,x_3)} \quad 0.20001 \cdot x_1 + x_2 + 1.0001 \cdot x_3$$
$$\text{s.t.} \quad -10x_1 + x_2 + x_3 = 1$$
$$x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0$$



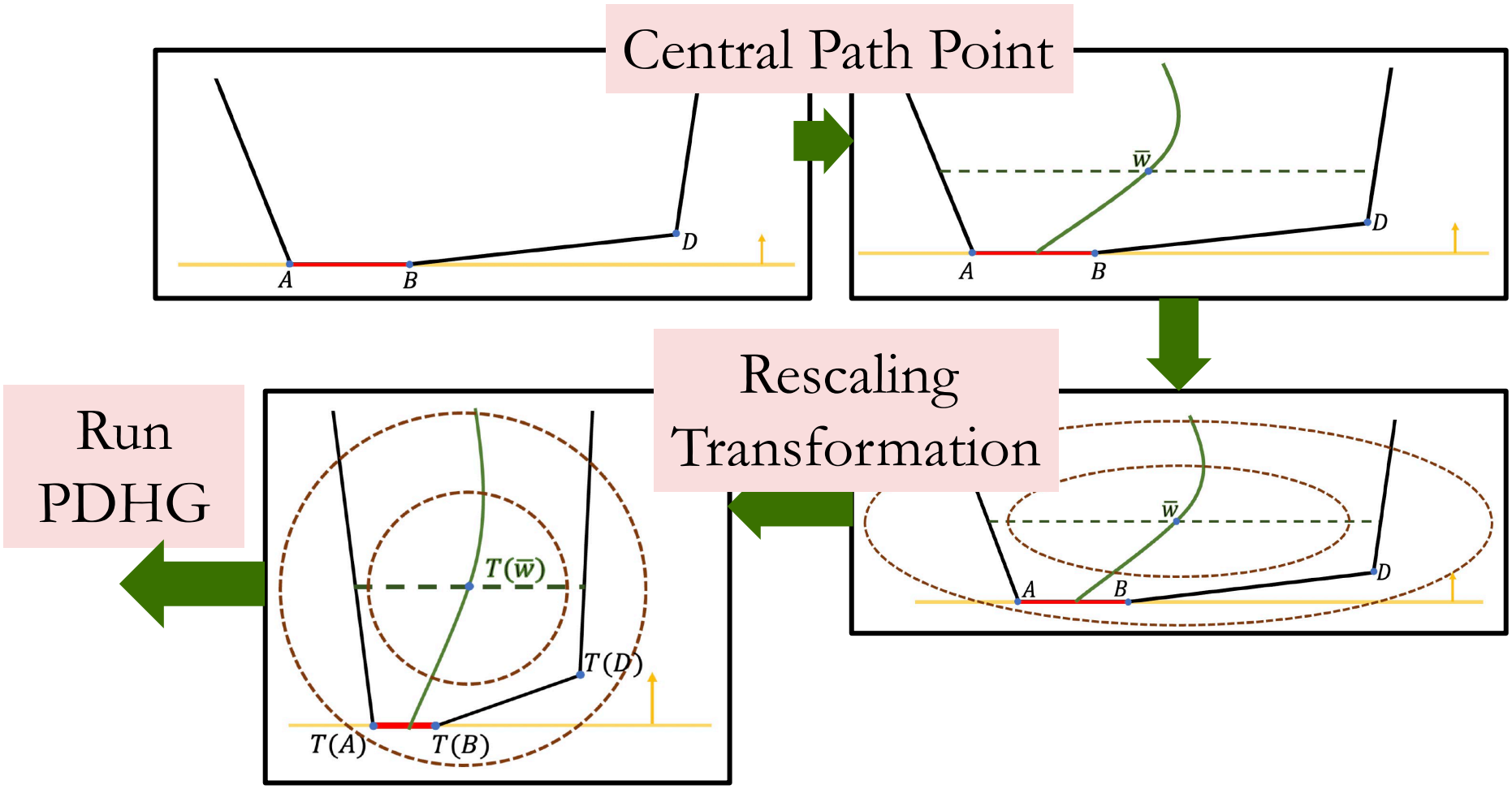
2. But linear convergence does exist even if the instance has multiple optimal solutions

$$\min_{x=(x_1,x_2,x_3)} \quad 0.2 \cdot x_1 + x_2 + x_3$$
$$\text{s.t.} \quad -10x_1 + x_2 + x_3 = 1$$
$$x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0$$



Using theory to develop practical computational speed-ups of PDHG

Brief idea


Central Path Point
Rescaling
Transformation
Run
PDHG
A
B
D
$\bar{w}$
$T(\bar{w})$
$T(A)$
$T(B)$
$T(D)$

Central-Path solutions are good interior-points

Let $F(\cdot)$ be a logarithmically homogeneous self-concordant barrier for $\mathcal{K}$ with complexity value ϑ_F

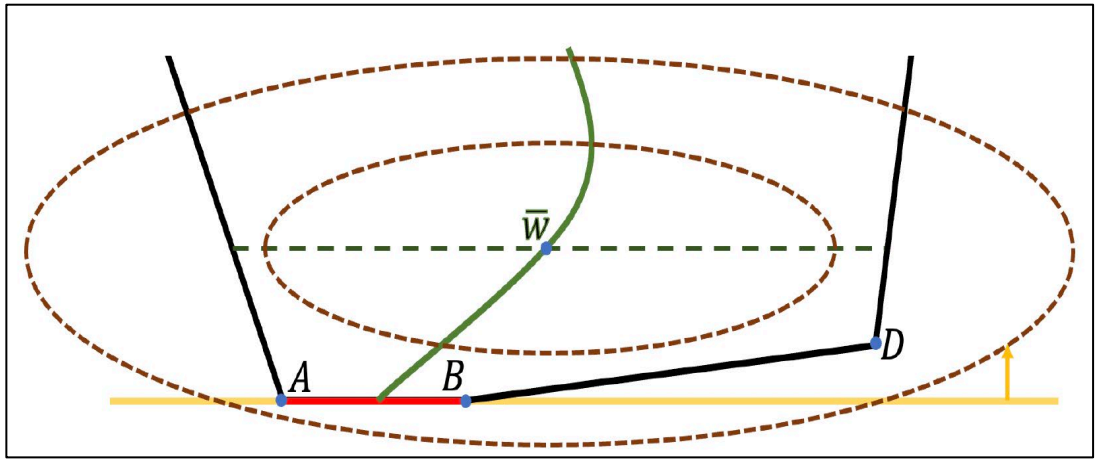
Central-path solutions are:

$$x_\mu := \arg\min_x \; c^\top x + \mu \cdot F(x) \qquad \text{s.t. } Ax = b,\ x \in \mathcal{K}$$

$$y_\mu, s_\mu := \arg\max_{y,s} \; b^\top y - \mu \cdot F^*(s) \qquad \text{s.t. } A^\top y + s = c,\ s \in \mathcal{K}^*$$

Example for LP: $F(x) := -\sum_{i=1}^n \ln(x_i)$, $\vartheta_F = n$

At $\bar{w} = (x_\mu, s_\mu)$, the local-norm ball nicely approximates the shape of sublevel sets:



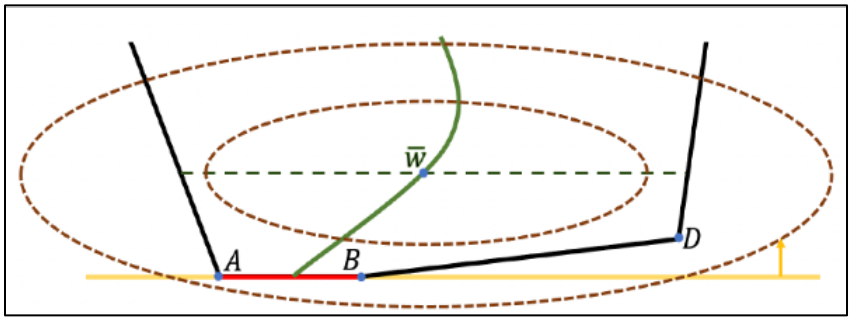
Yurii Nesterov research connection

- Interior Point Methods (IPMs)
- Self-concordance
- Self-scaled cones
- Amazing theory, and equally amazing usefulness in practice

Transformation based on a central-path solution

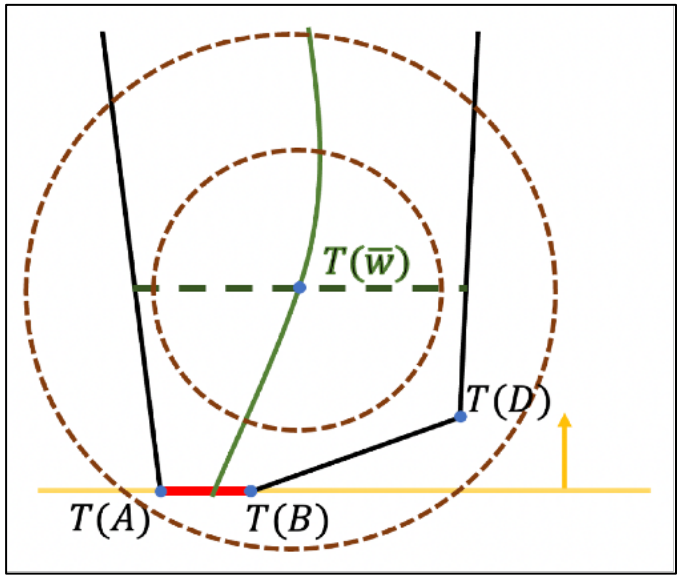
Let $H := \mu \cdot \nabla^2 F(x_\mu)$, then for $\overline{\boldsymbol{\delta}} := \boldsymbol{\vartheta_F} \cdot \boldsymbol{\mu}$ we have:

- $\boldsymbol{D_{\overline{\delta}} \leq 2\vartheta_F \cdot \sigma^+_{\min}(H)^{-1/2}}$
- $\boldsymbol{r_{\overline{\delta}} \geq \sigma^+_{\max}(H)^{-1/2}}$
- $\boldsymbol{d^H_{\overline{\delta}} \leq 2\vartheta_F \cdot \sigma^+_{\min}(H)^{-1/2}}$



After a rescaling transformation (turns the local-norm ball into a Euclidean norm ball) we have:

- $\boldsymbol{D_{\overline{\delta}} \leq 2\sqrt{\vartheta_F} \cdot \sqrt{\overline{\delta}}}$
- $\boldsymbol{r_{\overline{\delta}} \geq \sqrt{\frac{1}{\vartheta_F}} \cdot \sqrt{\overline{\delta}}}$
- $\boldsymbol{d^H_{\overline{\delta}} \leq 2\sqrt{\vartheta_F} \cdot \sqrt{\overline{\delta}}}$
- $\boldsymbol{\frac{D_{\overline{\delta}}}{r_{\overline{\delta}}} \leq 2\vartheta_F}$
- $d^H_{\overline{\delta}}$ is small if $\overline{\boldsymbol{\delta}}$ is small
- "Very nice theory"



Complexity under good linear transformation

Suppose we do the following:

1. rescaling transformation using a central-path solution **with duality gap** $\bar{\delta}$
2. row transformation to make $\kappa = 1$

Then the number of PDHG iterations required to compute an ε-optimal solution of the original CLP problem is upper bounded by:

$$\tilde{O}\left(\vartheta_F \cdot \left(\ln\left(\frac{1}{\varepsilon}\right) + \frac{D_{\bar{\delta}} + \bar{\delta}}{\varepsilon}\right)\right)$$

And for LP we obtain:

$$\tilde{O}\left(n \cdot \frac{\bar{\delta}}{\bar{\gamma}} \cdot \ln\left(\frac{1}{\varepsilon}\right)\right)$$

Here $\bar{\gamma}$ is the optimality gap of the second-best extreme point solution

Suppose we do the following:

1. rescaling transformation using a central-path solution **with duality gap** $\bar{\delta}$
2. row transformation to make $\kappa = 1$

Then the number of PDHG iterations required to compute an ε-optimal solution of the original CLP problem is upper bounded by:

$$\tilde{O}\left(\vartheta_F \cdot \left(\ln\left(\frac{1}{\varepsilon}\right) + \frac{D_{\bar{\delta}} + \bar{\delta}}{\varepsilon}\right)\right)$$

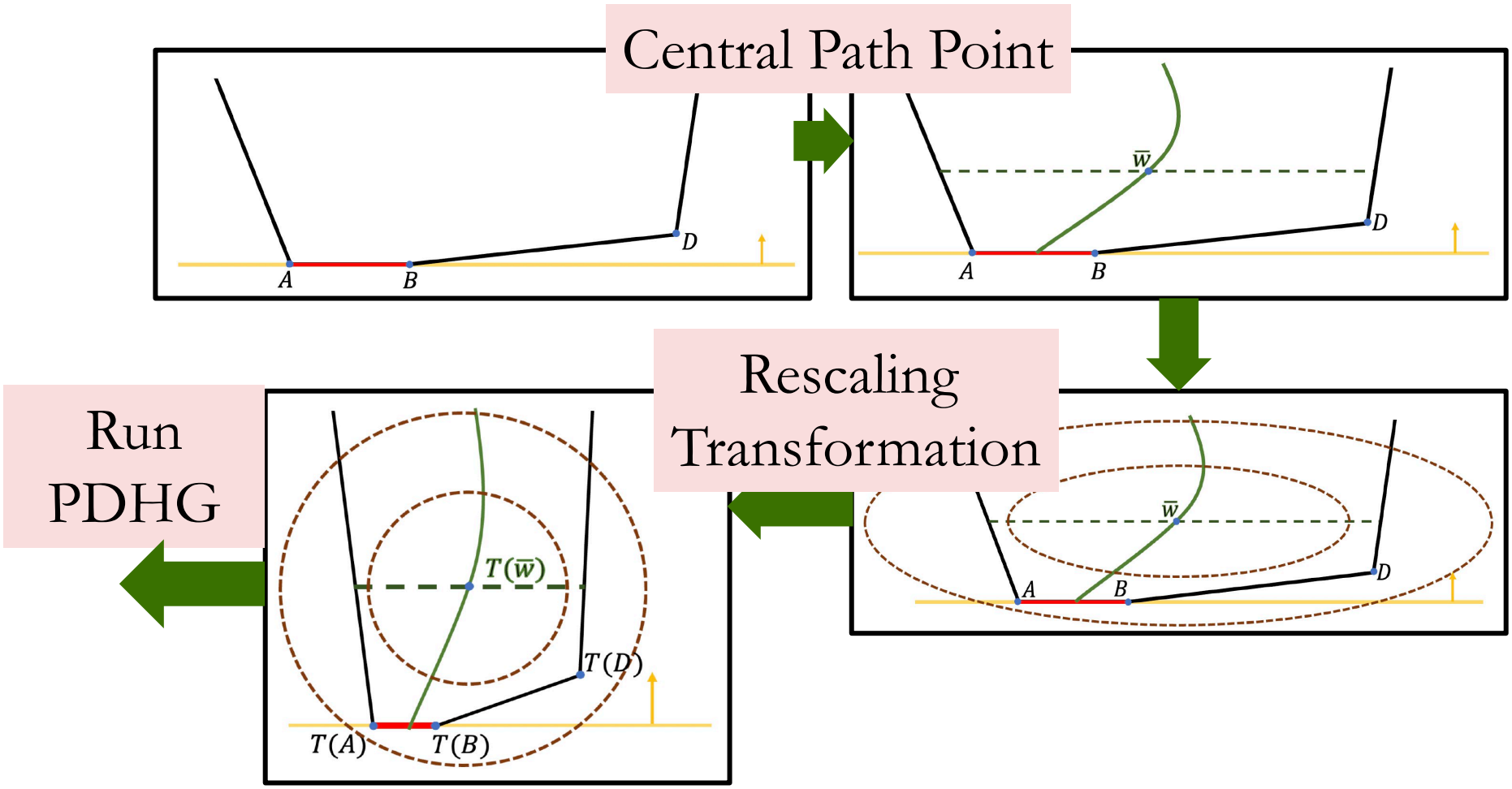
And for LP we obtain:

$$\tilde{O}\left(n \cdot \frac{\bar{\delta}}{\bar{\gamma}} \cdot \ln\left(\frac{1}{\varepsilon}\right)\right)$$

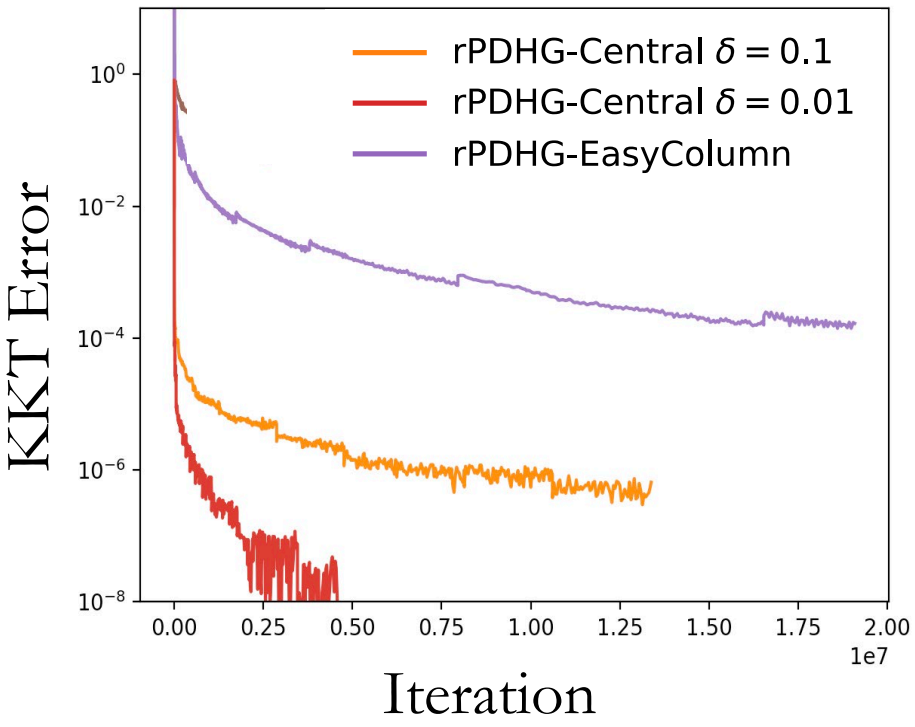
Here $\bar{\gamma}$ is the optimality gap of the second-best extreme point solution

- We have replaced $\frac{D_\delta}{r_\delta}$ by ϑ_F
- The smaller $\bar{\delta}$ is, the faster the convergence
- Of course we will need to "pay" to compute the point on the central path…

Summary


Central Path Point
Rescaling
Transformation
Run
PDHG
A
B
D
$\bar{w}$
$T(\bar{w})$
$T(A)$
$T(B)$
$T(D)$

"Proof of concept" applied to problem instance **bmoipr2**



Here we pre-multiply A by $(AA^{\mathsf{T}})^{-1/2}$ to yield $\kappa = 1$

rPDHG-EasyColumn
Column rescaling to **normalize L_∞** column norms

rPDHG-Central δ=0.1
Column rescaling using **central-path solution** with KKT error **0.1**

rPDHG-Central δ=0.01
Similar as above, with KKT error **0.01**

- Here the central path solutions were computed by MOSEK
- This is an admittedly "unfair" comparison, but it validates the potential of the overall approach…

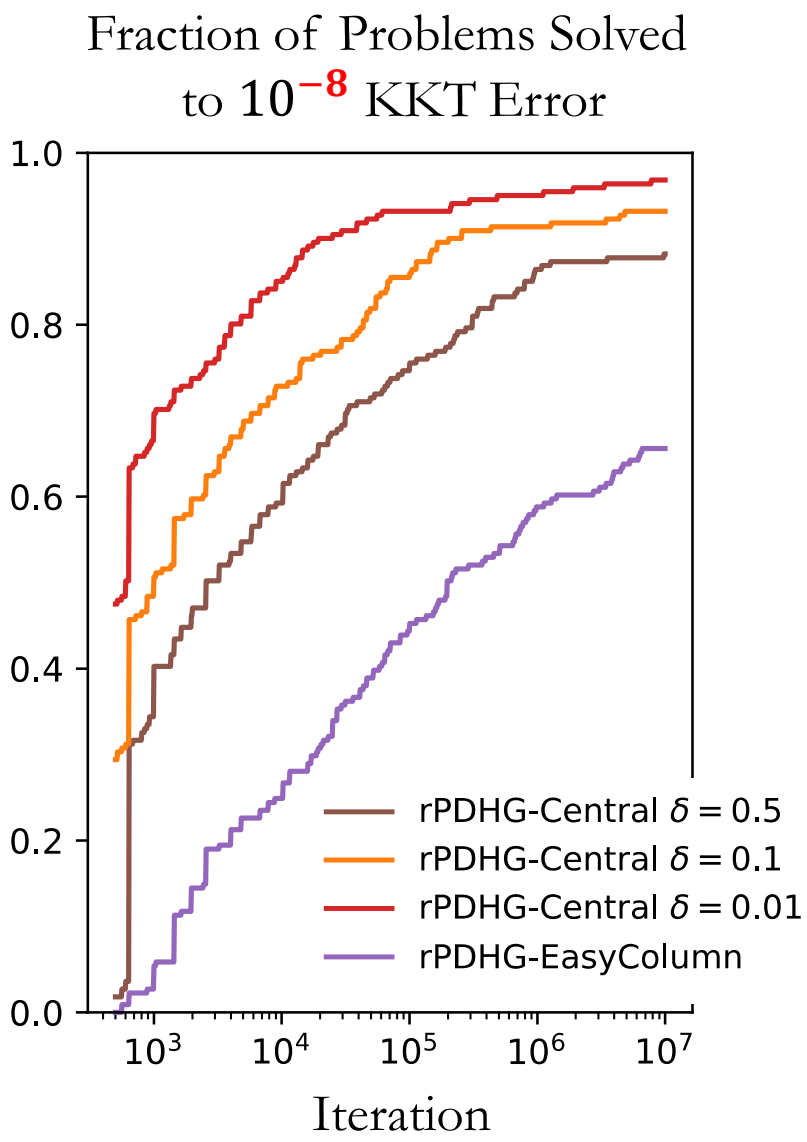
More proof of concept…

- Standard-form reformulations of the LP relaxations from the MIPLIB 2017 dataset. We selected all the problems (222 in total) for which
  - $m \times n$ is in the range from 10^7 to 10^9 (so MOSEK doesn't fail!), and
  - Central-path solutions exist
- We u

$$\varepsilon := \max \left\{ \frac{\|Ax - b\|}{1 + \|b\|}, \frac{\left\|(c - A^\top y)^-\right\|}{1 + \|c\|}, \frac{|c^\top x - b^\top y|}{1 + |c^\top x| + |b^\top y|} \right\}$$

- MOSEK was used to compute central-path solutions with relative duality gap δ in the range .01 to .50
- All PDHG methods are implemented with restarts.
- The initial iterates for all methods are 0

Proof of concept, continued



All methods pre-multiply row transformations to yield $\kappa = 1$

rPDHG-Central δ=0.5
Column rescaling using **central-path solution** with KKT error **0.5**

rPDHG-Central δ=0.1
Similar as above, with KKT error **0.1**

rPDHG-Central δ=0.01
Similar as above, with KKT error **0.01**

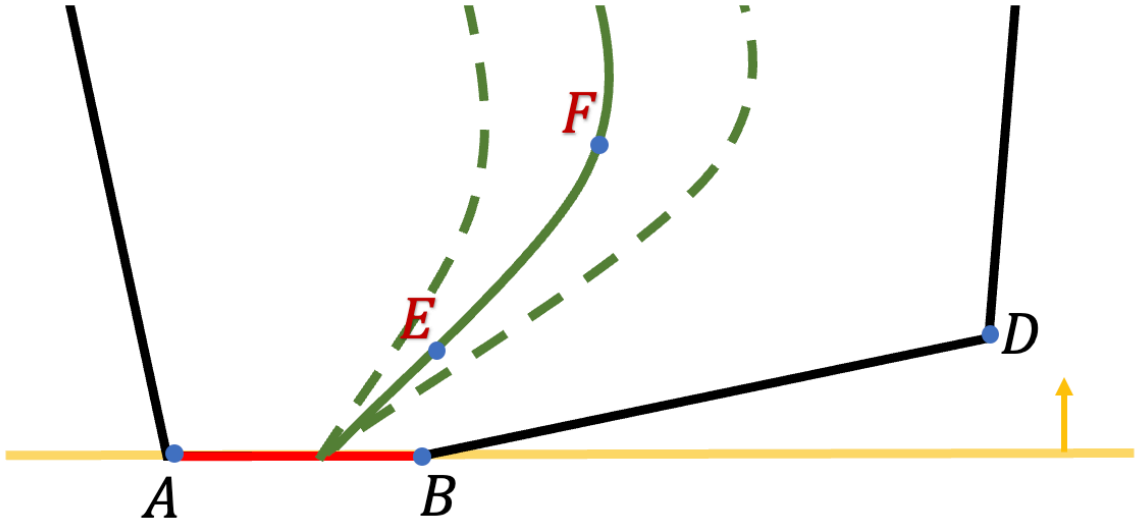
rPDHG-EasyColumn
Column rescaling to **normalize L_∞** column norms

The better the central-path solution is, the better the rescaling is …

rPDHG-AHR ("Adaptive Hessian Rescaling") and Computational Experiments

Main Strategy

Main strategy: use a "CG-IPM" to compute a low-accuracy central-path solution to obtain a good rescaling, and then use rPDHG



Main Strategy, continued

Main strategy: use a "CG-IPM" to compute a low-accuracy central-path solution to obtain a good rescaling, and then use rPDHG

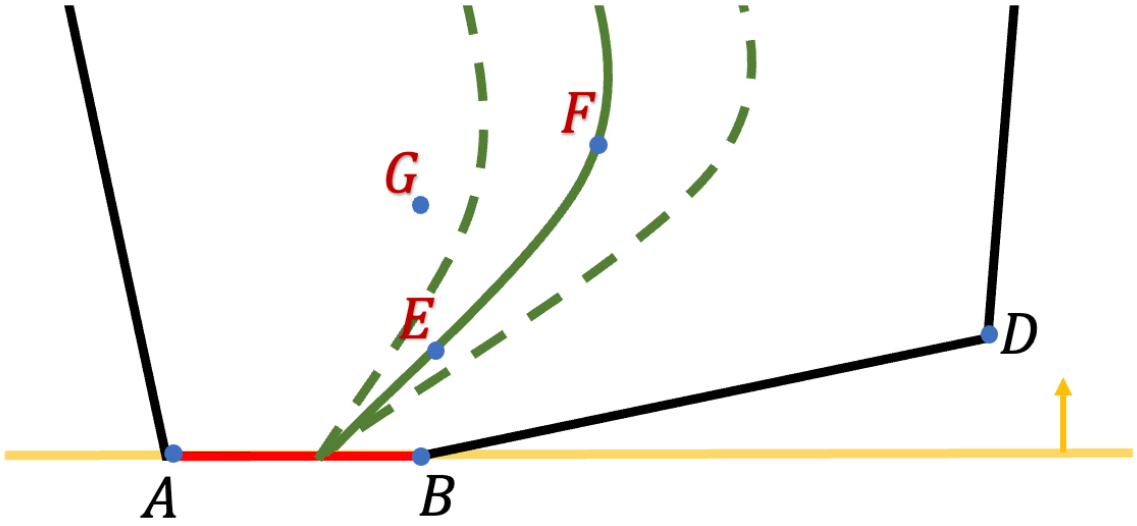
- Time to compute a central-path solution can be significant
  - we use a conjugate-gradient-method-based IPM (**CG-IPM**)
  - Implementation follows Nocedal and Wright *Numerical Optimization* (2006)
  - but we solve the normal equations using conjugate gradient method
- We employ only diagonal row rescaling to try to improve κ
- Column rescaling uses the central-path Hessian of w_{int} followed by PDLP's rescaling (Ruiz rescaling and Pock-Chambolle rescaling), which we call the "w_{int}-rescaled problem"

Computational Experiments

Main strategy: use a "CG-IPM" to compute a low-accuracy central-path solution to obtain a good rescaling, and then use rPDHG

- We selected all the LP relaxations from MIPLIB 2017 dataset that are:
  - large enough ($m \times n > 10^6$)
  - but not too large (number of non-zeros $< 10^5$)
  - This yielded 413 instances in total
- Some instances are feasible but do not have a central path. We used such instances regardless.

Using Adaptive Rescaling

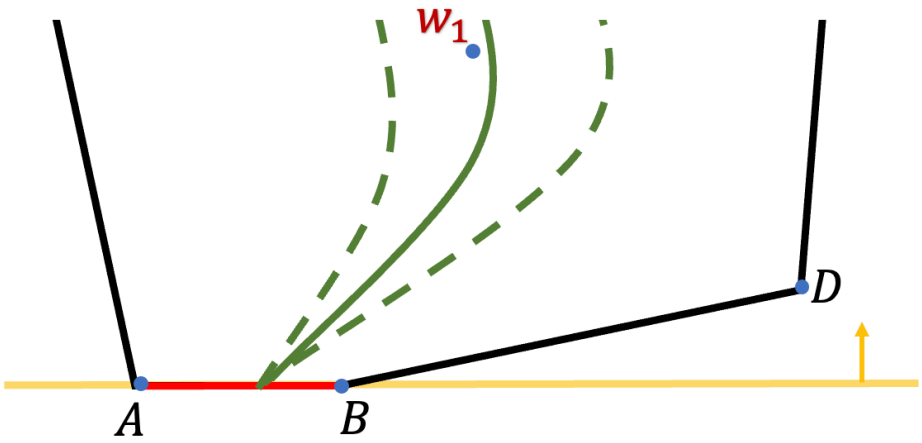


Motivating concepts of Adaptive Rescaling:

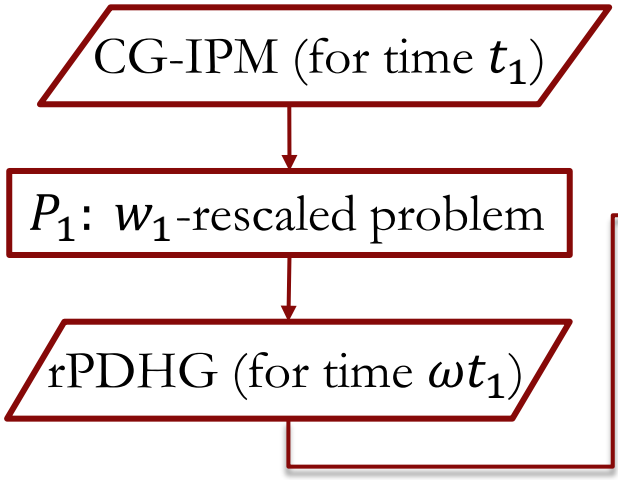
- Adaptively balance the cost of computing the rescaling (CG-IPM) with the savings from running rPDHG on the rescaled instance
- Try to identify a "good-enough" rescaling as early as possible and use the good-enough rescaling
- Seek to avoid unnecessary extra computation to get an "even better" rescaling when the improvement will not be cost-effective

An adaptive rescaling heuristic (1st round)

The initial interior solution is $w_0 = e$, and P_0 is the w_0-rescaled problem



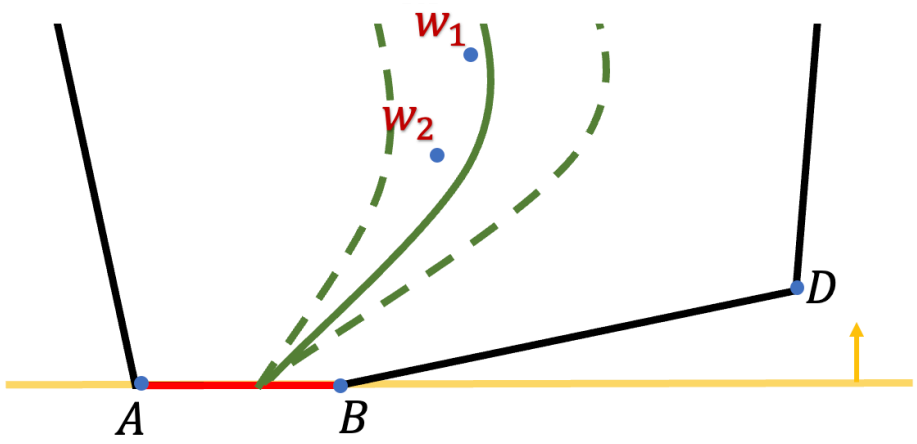
Parameters: $t_1 = 2$ seconds, $\omega = 3$ ...



1. If current solution is "reasonably close" to required tolerance, we continue using rPDHG on P_1 until the end

2. Otherwise, continue to the next round …

An adaptive rescaling heuristic (2nd round)



Increased time: $t_2 = 2 \cdot t_1$ ...

CG-IPM (for time t_2)

P_2: w_2-rescaled problem

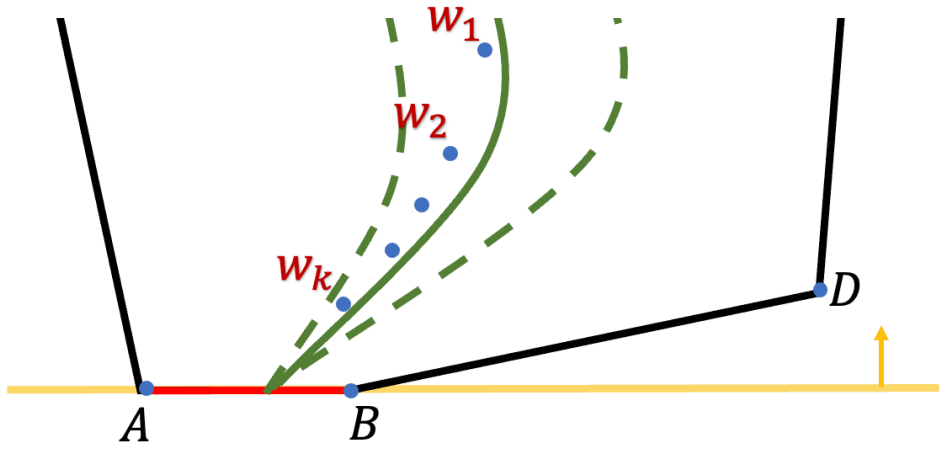
rPDHG (for time ωt_2)

1. If current solution is "reasonably close" to required tolerance, we continue using rPDHG on P_2 until the end

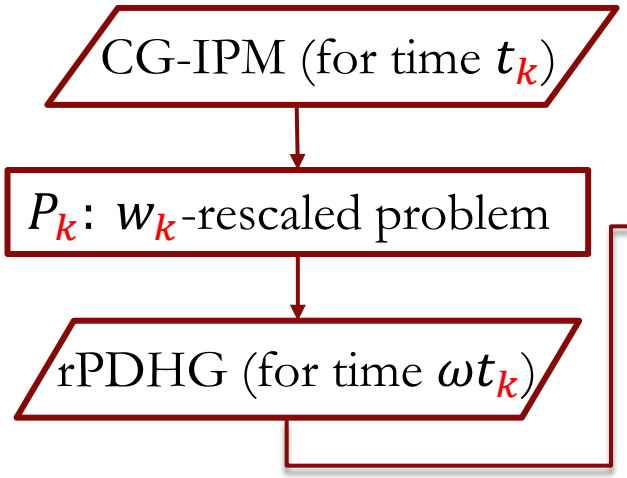
2. If current rPDHG solution is not as good as in the previous round, we revert to P_1 and continue only using rPDHG on P_1 until the end

3. Otherwise, continue to the next round …

An Adaptive rescaling heuristic (k^{th} round)



Increased time serials: $t_k = 2^k$ …



1. If current solution is "reasonably close" to required tolerance, we continue using rPDHG on P_k until the end

2. If current rPDHG solution is not as good as in the previous round, we revert to P_{k-1} and continue only using rPDHG on P_{k-1} until the end

3. Otherwise, continue to the next round …

Adaptive rescaling heuristic

rPDHG with Adaptive central-path Hessian Rescaling : **rPDHG−AHR**

Algorithm 4: Scheme for rPDHG with Adaptive Hessian Rescaling (rPDHG-AHR)

Input: Initial iterate $z^0 := (x^{0,0}, y^{0,0})$, initial point w^0 for CP-CGM, time parameter t, time multiplier ω, target relative error ε, and adaptivity error parameters $\hat{\varepsilon}$ and $\bar{\varepsilon}$. Define $k := 0$, $\varepsilon_0 := +\infty$;

repeat

  Run (or continue running) CP-CGM from w^k for t seconds. Output w^{k+1} ;

  Construct new rescaled problem: define $\eta^{k+1} := (\mathsf{s}^{k+1})^\top \mathsf{x}^{k+1}$, $\tilde{D}_1 := \sqrt{\eta^{k+1}} H^{-1/2}_{\mathsf{x}^{k+1}}$ and $\tilde{D}_2 = I$. Optionally do further rescaling by introducing $\bar{D}_1$ and $\bar{D}_2$, and setting $D_1 := \bar{D}_1\tilde{D}_1$ and $D_2 := \bar{D}_2\tilde{D}_2$. Then construct the new rescaled problem $(\tilde{\mathrm{P}})_{k+1}$ using D_1 and D_2 ;

  Run rPDHG on rescaled problem $(\tilde{\mathrm{P}})_{k+1}$ for ωt seconds. Output the transformed solution z^{k+1} and the relative error $\varepsilon_{k+1} := \mathrm{MErr}_\varepsilon(x^{k+1}, y^{k+1})$;

  $k \leftarrow k+1$ and $t \leftarrow 2t$;

until *either (a)* $\varepsilon_k \le \bar{\varepsilon}$, *or (b)* $\varepsilon_k > \varepsilon_{k-1}$ *and* $\varepsilon_{k-1} \le \hat{\varepsilon}$;

If (a) holds, then fix the new rescaling: run rPDHG on $(\tilde{\mathrm{P}})_k$ until a solution $z = (x, y)$ is computed for which $\mathrm{MErr}_\varepsilon(x, y) \le \varepsilon$;

If (b) holds, then revert to and fix the previous rescaling: run rPDHG on $(\tilde{\mathrm{P}})_{k-1}$ until a solution $z = (x, y)$ is computed for which $\mathrm{MErr}_\varepsilon(x, y) \le \varepsilon$;

Computational Experiments with rPDHG-AHR

Algorithm 4: Scheme for rPDHG with Adaptive Hessian Rescaling (rPDHG-AHR)

Input: Initial iterate $z^0 := (x^{0,0}, y^{0,0})$, initial point w^0 for CP-CGM, time parameter t, time multiplier ω, target relative error ε, and adaptivity error parameters $\hat{\varepsilon}$ and $\bar{\varepsilon}$. Define $k := 0$, $\varepsilon_0 := +\infty$;

repeat

  Run (or continue running) CP-CGM from w^k for t seconds. Output w^{k+1} ;

  Construct new rescaled problem: define $\eta^{k+1} := (\mathsf{s}^{k+1})^\top \mathsf{x}^{k+1}$, $\tilde{D}_1 := \sqrt{\eta^{k+1}} H_{\mathsf{x}^{k+1}}^{-1/2}$ and $\tilde{D}_2 = I$. Optionally do further rescaling by introducing $\bar{D}_1$ and $\bar{D}_2$, and setting $D_1 := \bar{D}_1 \tilde{D}_1$ and $D_2 := \bar{D}_2 \tilde{D}_2$. Then construct the new rescaled problem $(\tilde{\mathrm{P}})_{k+1}$ using D_1 and D_2 ;

  Run rPDHG on rescaled problem $(\tilde{\mathrm{P}})_{k+1}$ for ωt seconds. Output the transformed solution z^{k+1} and the relative error $\varepsilon_{k+1} := \mathrm{MErr}_\varepsilon(x^{k+1}, y^{k+1})$;

  $k \leftarrow k+1$ and $t \leftarrow 2t$;

until *either (a)* $\varepsilon_k \le \bar{\varepsilon}$, *or (b)* $\varepsilon_k > \varepsilon_{k-1}$ *and* $\varepsilon_{k-1} \le \hat{\varepsilon}$;

If (a) holds, then fix the new rescaling: run rPDHG on $(\tilde{\mathrm{P}})_k$ until a solution $z = (x, y)$ is computed for which $\mathrm{MErr}_\varepsilon(x, y) \le \varepsilon$;

If (b) holds, then revert to and fix the previous rescaling: run rPDHG on $(\tilde{\mathrm{P}})_{k-1}$ until a solution $z = (x, y)$ is computed for which $\mathrm{MErr}_\varepsilon(x, y) \le \varepsilon$;

We compare:

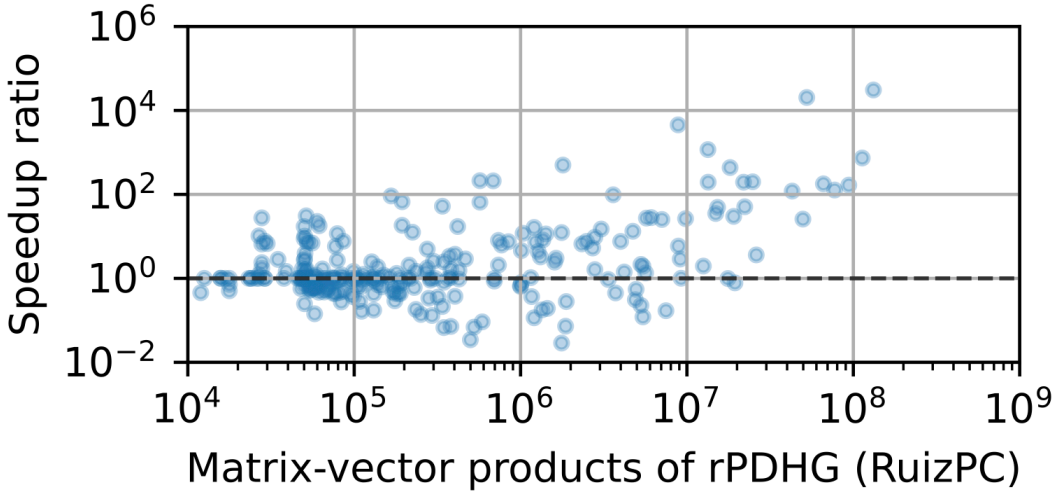
1. **rPDHG-AHR** : rPDHG with Adaptive central-path Hessian Rescaling (using CG-IPM for the central-path computations)
2. **rPDHG-RuizPC** : use heuristic Ruiz rescaling on ***A***, followed by Pock-Chambolle rescaling. (This is the same as the rescaling used in PDLP.)
3. **IPM** : a home-grown standard primal-dual predictor-corrector interior point method, straight from Nocedal and Wright *Numerical Optimization* (2006).

Computational Comparison: rPDHG-AHR and rPDHG(RuizPC)

		Fraction solved by **rPDHG-AHR**	
		S	**N**
Fraction solved by **rPDHG (RuizPC)**	**S**	**87.7%**	**1.7%**
	N	**7.5%**	**3.1%**

Speedup Ratio:

$$\frac{\text{Matrix-vector products } \textbf{rPDHG(RuizPC)}}{\text{Matrix-vector products } \textbf{rPDHG-AHR}}$$



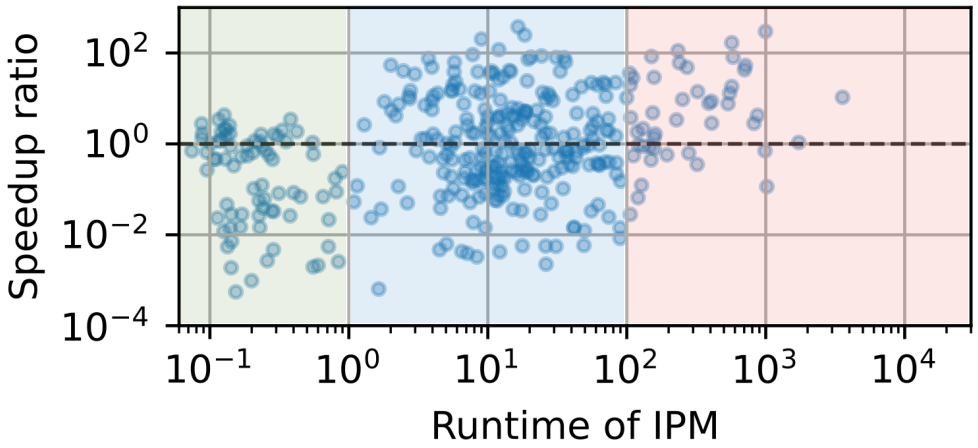
- In general, the harder the problem is for rPDHG(RuizPC), the larger the speedups from using rPDHG-AHR
- rPDHG-AHR solves about 95.2% of problems and is more reliable than rPDHG(RuizPC) (solves about 89.4% of problems)

Computational Comparison: rPDHG-AHR and IPM

		Fraction solved by **rPDHG−AHR**	
		S	**N**
Fraction solved by **IPM**	**S**	**93.0%**	**1.5%**
	N	**2.2%**	**3.4%**

Speedup Ratio:

$$\frac{\text{Runtime } \textbf{IPM}}{\text{Runtime } \textbf{rPDHG−AHR}}$$



- Generally speaking, the harder the problem is for IPM, the larger the speedups from using rPDHG-AHR
- rPDHG-AHR is slightly more reliable than IPM

Yurii Nesterov research connection

- Theory that leads to new methods that lead to computational improvements in practice

Extra Experiment: comparing with the "best possible" rescaling

Extra Experiment: what if we were clairvoyant and can use the "best possible" rescaling?

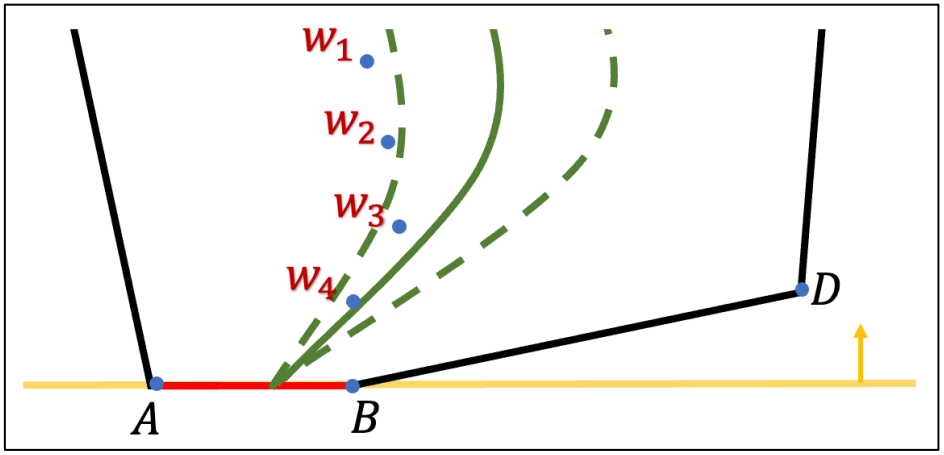
rPDHG-B-t Run CG-IPM for t seconds and get solution w_{int}. And then run rPDHG on the w_{int}-rescaled problem.

rPDHG-Ideal

We report the best potential result of **rPDHG-B-t**, which is rPDHG-B-t with t^* optimized over $\{0.5, 1, 2, 4, 8, \dots\}$ to achieve the best runtime.

Extra Experiment: comparing with the "best possible" rescaling

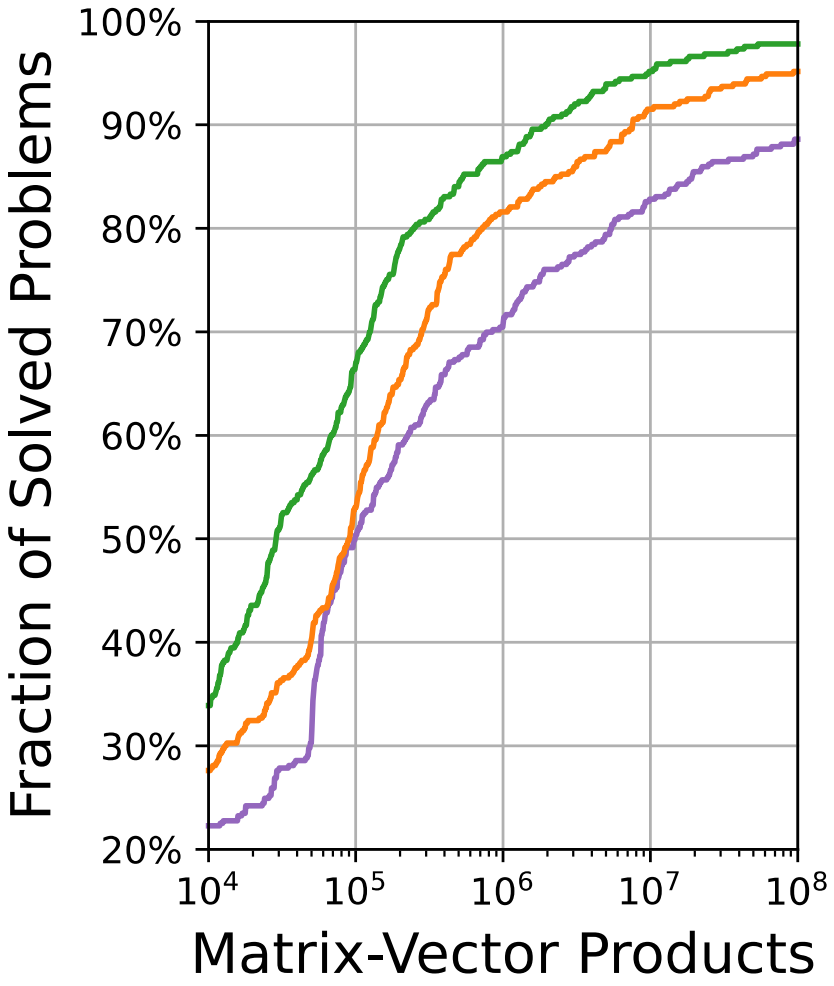
What if we were clairvoyant and can use the "best possible" rescaling?



	CG-IPM time (sec.)	rPDHG time (sec.)	Total time (sec.)
w_1	2	16	18
w_2	4	8	12
w_3	8	6	14
w_4	16	5	21

"best possible" rescaling

Performance of Adaptive interior-point rescaling



rPDHG-Ideal
The rPDHG with our best rescaling
rPDHG-AHR
The rPDHG with our adaptive rescaling
rPDHG (RuizPC)
The rPDHG with heuristic Ruiz/PC rescaling (same with PDLP)

Adaptive rescaling **rPDHG-AHR** closely mimics **rPDHG-Ideal.**

Recap, Takeaways, and Remarks

Recap and takeaways:

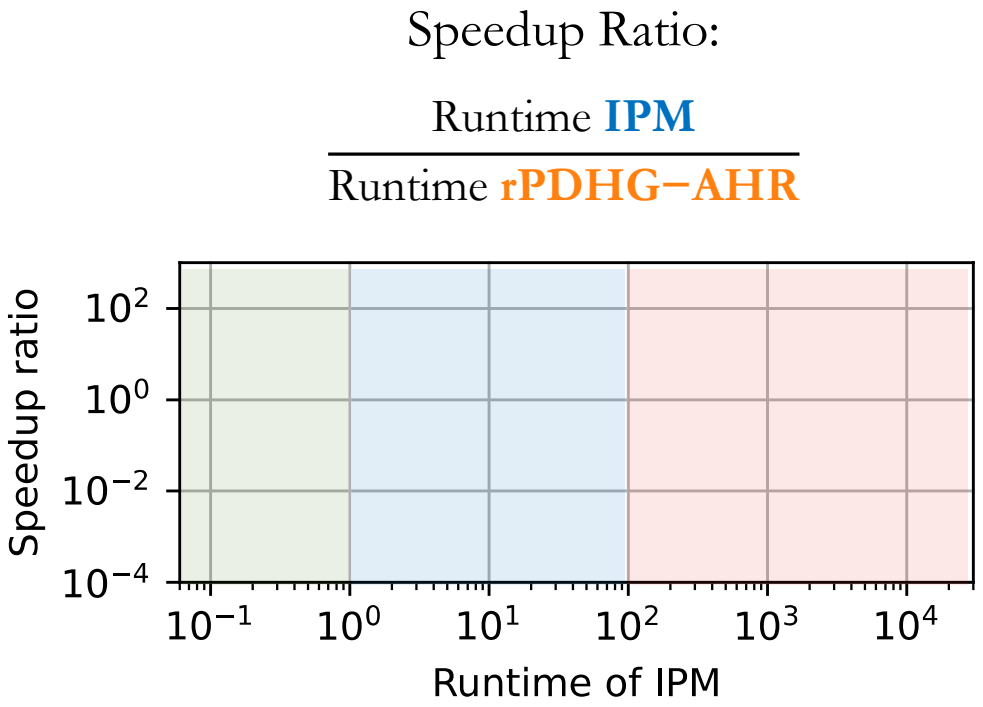
- The convergence rate of PDHG on CLP is related to the geometry of primal-dual sublevel sets measured with $\boldsymbol{D_\delta}$, $\boldsymbol{r_\delta}$, $\boldsymbol{d_\delta^H}$
- Rescaling using a central-path solution can improve the geometry of the primal-dual sublevel sets
- Our strategy: use a "CG-IPM" to compute a low-accuracy central-path solution to obtain a good rescaling, and then use rPDHG
- FOMs can competitively compute solutions of the same high accuracy as IPMs

Remarks:

- Our results relied only on PDHG's average iterate convergence and non-expansiveness properties. Similar results might also hold for other FOMs, in particular ADMM, EGM, PPM …

Computational Comparison: rPDHG-AHR and IPM

		Fraction solved by **rPDHG−AHR**	
		S	**N**
Fraction solved by **IPM**	**S**	**93.0%**	**1.5%**
	N	**2.2%**	**3.4%**



- Generally speaking, the harder the problem is for IPM, the larger the speedups from using rPDHG-AHR.
- rPDHG-AHR is also slightly more reliable than IPM.

On the Community of Modern Continuous Optimization

Imagine you could mold the culture in your field. What characteristics would you want the culture to have?

- High standards of intellectual excellence
- Passion for research process
- Integrity in authorship, publishing, intellectual property
- Judgement to know which ideas are better than others
- Generosity in crediting others with their contributions
- Empathy for colleagues in their professional experiences
- Supporting and encouraging others to do their best
- Collaborative spirit to work together
- Trusting the best in our colleagues

Who could ask for a better example than Yurii Nesterov?

How fortunate are we that Yurii Nesterov has been in our field?

How fortunate are we that Yurii Nesterov has been in our field?

VERY!

Thank you!