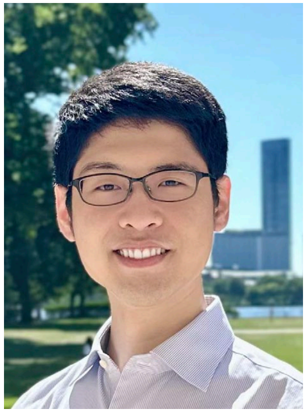
Level-Set Geometry and the Performance of Restarted-PDHG for Conic LP

Zikai Xiong (with Robert Freund)

Cargèse Workshop on Combinatorial Optimization



Zikai Xiong
(MIT, GaTech,
Northwestern)

Robert Freund
(MIT Sloan)

September 2025

Institut d'Etudes Scientifiques de Cargèse

Two slides on Interior-Point Methods (IPMs)

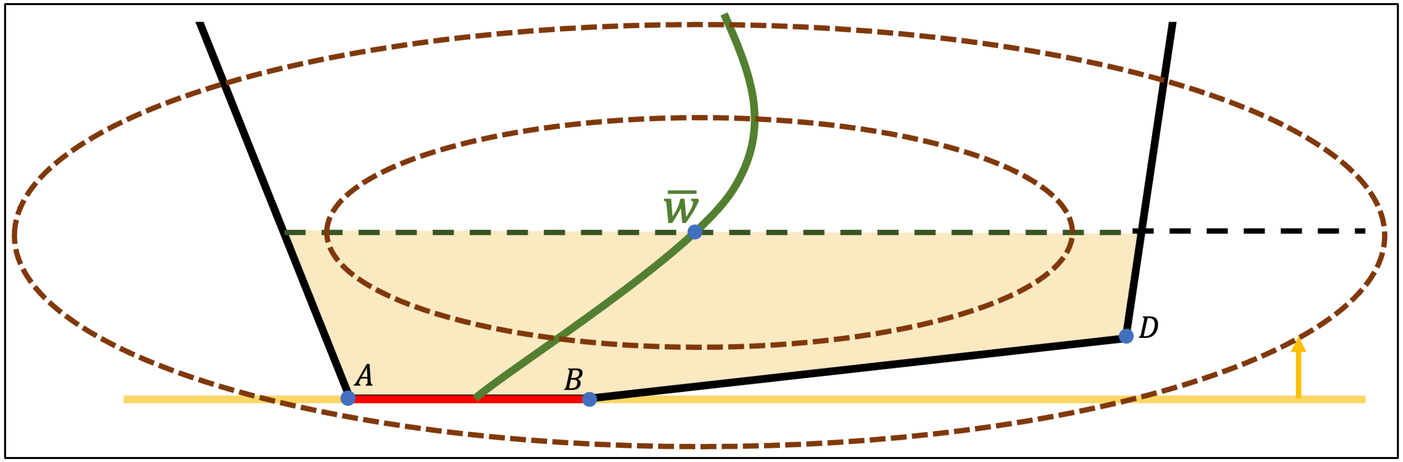
Central-Path ellipsoids have remarkable properties

Central-path solutions are:

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

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

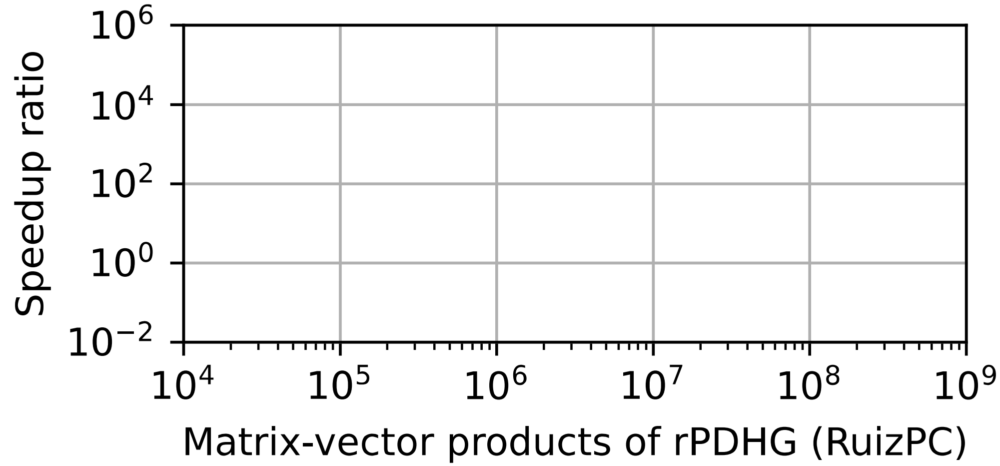
Example for LP: $F(x) := -\sum_{i=1}^{n} \ln(x_i)\,,\;\; \vartheta_F = n$



Sneak Preview:

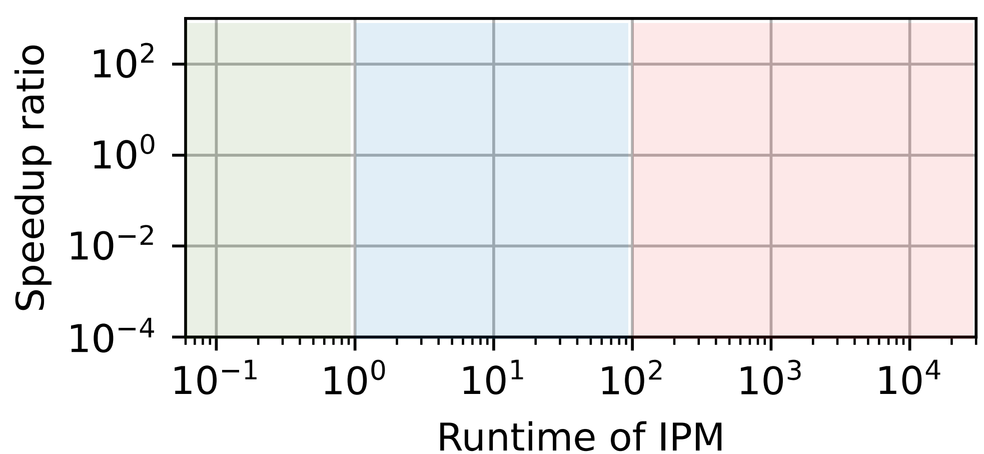
Distribution of Speedups of our method rPDHG-AHR

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))



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

CLP in standard form

(primal)

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

(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

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

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]

Challenge II: Loose/unworkable computational guarantees

Original computational guarantees:

Theorem [Applegate, Hinder, Lu, Lubin, 2023] PDHG computes an ε-optimal solution within

$$O\left((\|x^\star\| + \|y^\star\|) \cdot ||A|| \cdot \boldsymbol{H(K)} \cdot \log\left(\frac{\|x^\star\| + \|y^\star\|}{\varepsilon} \right) \right)$$

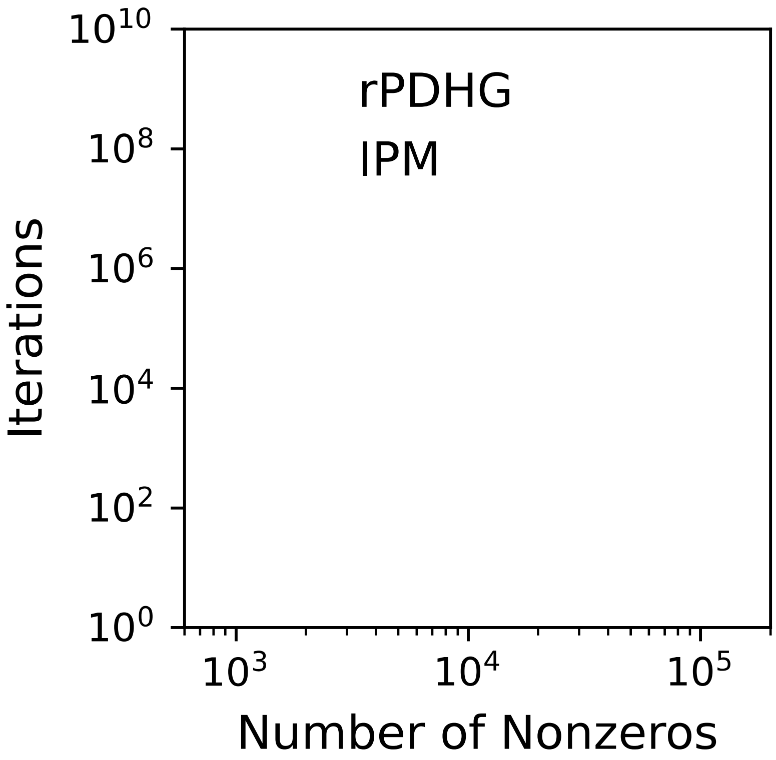
iterations.

$\boldsymbol{H(K)}$ is the global Hoffman constant of the matrix $\boldsymbol{K}$ of the KKT system

$$\boldsymbol{K} = \begin{pmatrix} A & \\ -A & \\ & A^\top \\ c^\top & -b^\top \end{pmatrix} \qquad \boldsymbol{H(K)} \approx \frac{1}{\min_{J \subseteq \{1,\dots,2m+n+1\}} \sigma^{+}_{\min}(K_J)}$$

Performance of rPDHG

Iterations Required for
LP relaxations from MIPLIB 2017



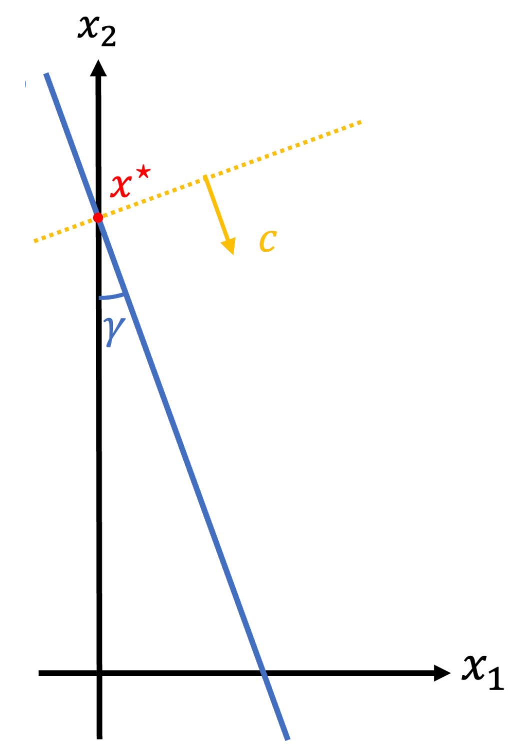
- rPDHG uses more iterations than IPM
  makes sense…

- Some small instances require a very large number of iterations
  a real challenge for rPDHG

A seemingly easy LP instance

For $\gamma \in \left(0, \frac{\pi}{2}\right)$ define:

$$P(\gamma): \quad \min_{x_1, x_2} \ \sin(\gamma)\, x_1 - \cos(\gamma)\, x_2 \quad \text{s.t.} \ \sin(\gamma)\, x_1 + \cos(\gamma)\, x_2 = 1, \quad x_1 \geq 0, x_2 \geq 0$$



$P(\gamma)$ is always easy for the simplex method and interior-point methods

However, when γ is small, PDHG requires at least 1,000,000 iterations. What **conditions** of $P(\gamma)$ make it so hard for PDHG?

Goals for improving theory for rPDHG for LP

Challenges:

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

We seek iteration bounds that are more natural, 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

Computational guarantees

Original computational guarantees:

Theorem [Applegate, Hinder, Lu, Lubin, 2023] PDHG computes an ε-optimal solution within

$$O\left((\|x^\star\| + \|y^\star\|) \cdot ||A|| \cdot \boldsymbol{H(K)} \cdot \log\left(\frac{\|x^\star\| + \|y^\star\|}{\varepsilon}\right)\right)$$

iterations.

Key questions:

- What conditions of the problem actually drive the performance of PDHG? Sublevel-set Geometry
- Can we improve these condition numbers and so improve computational performance in theory/practice?

  Yes, we will improve the geometry using Hessian rescaling

Condition numbers related to Level-set Geometry

Reformulation of Standard Form CLP

CLP in standard form

(Primal)

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

(Dual)

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

Assumptions (similar as in IPMs)

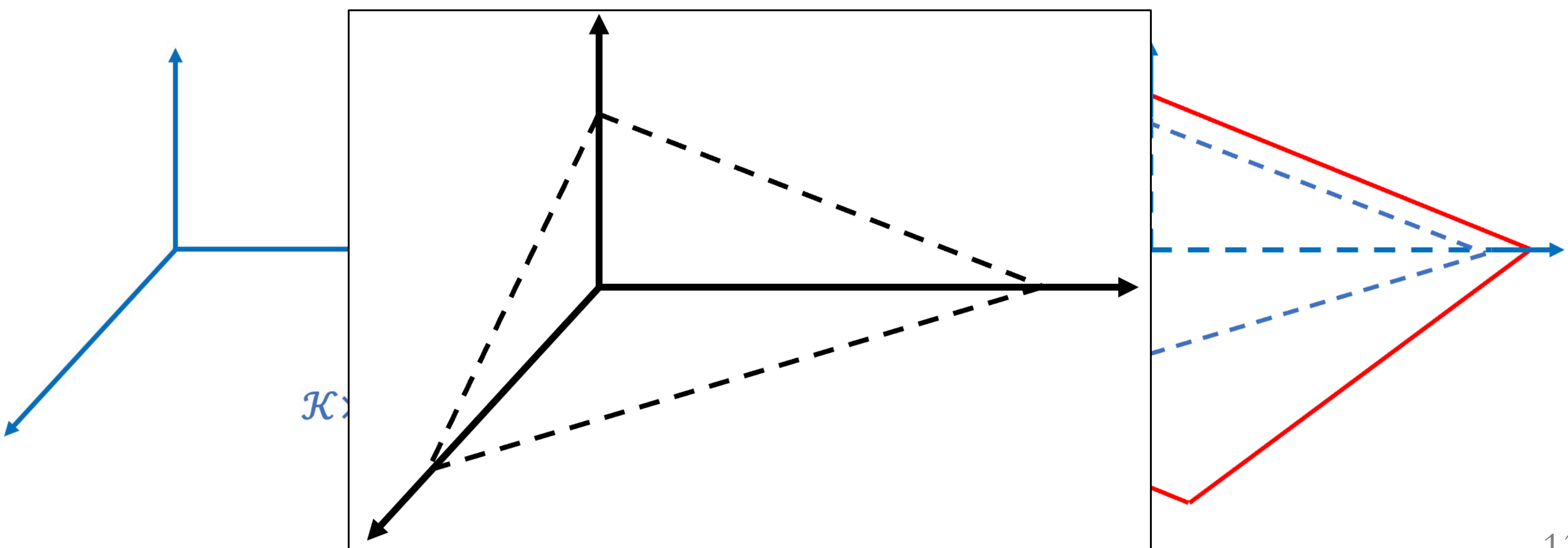
1. Strict feasibility of P and D (though not necessary in context of computation)
2. Rows of A are linearly independent
   - y and s correspond one-to-one
3. Objective vector $c \in \mathrm{Null}(A)$
   - $Ac = 0$
   - not essential, but keeps things simpler

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

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

(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



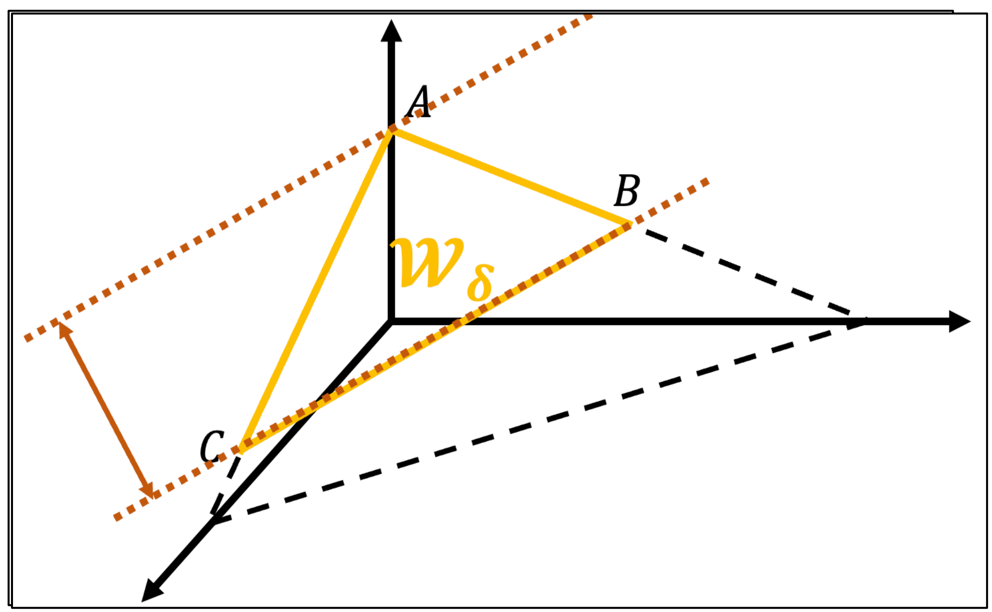
Primal-Dual Sublevel Sets

(x, s) is feasible

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

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

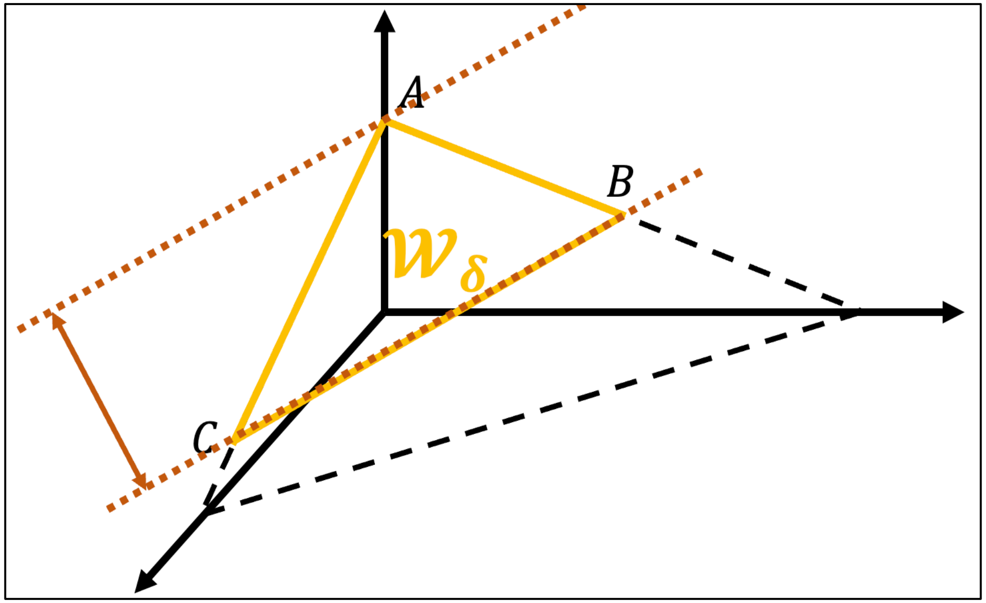
duality gap is at most δ



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

Three Geometric Condition Numbers

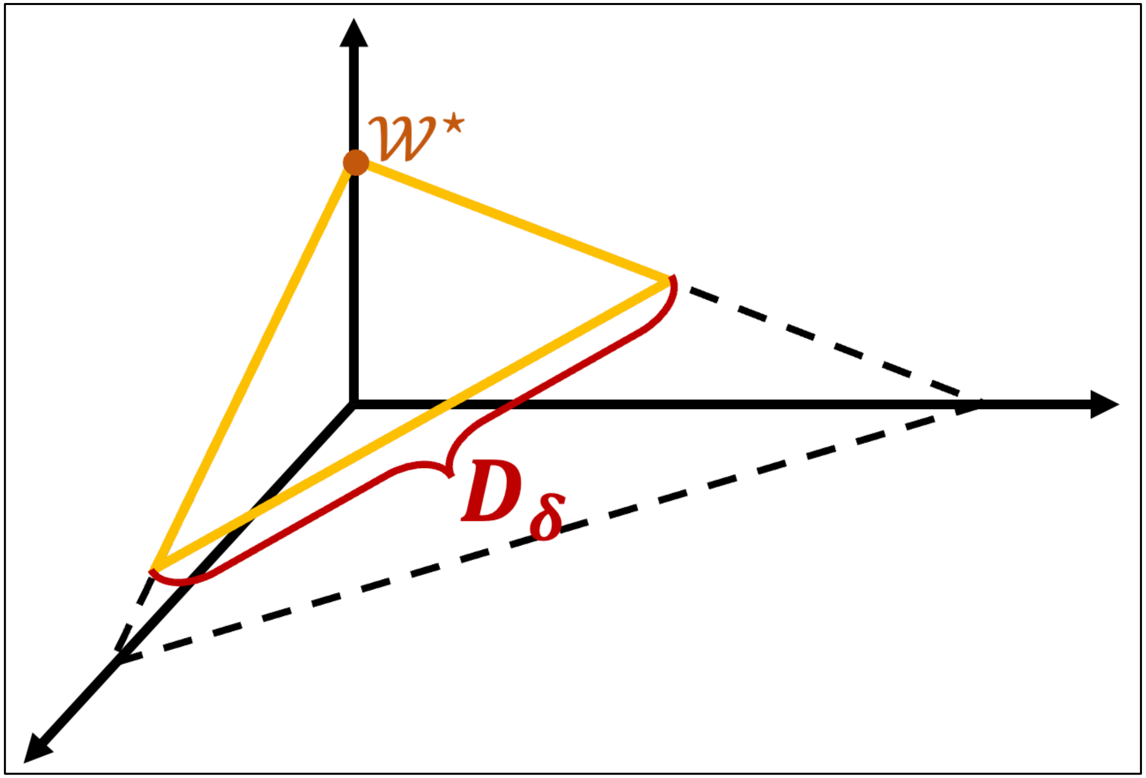
1. D_δ -- the diameter of $\mathcal{W}_\delta$
2. r_δ -- conic radius of $\mathcal{W}_\delta$
3. d_δ^H -- Haussdorf distance between $\mathcal{W}_\delta$ and $\mathcal{W}^\star$



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

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

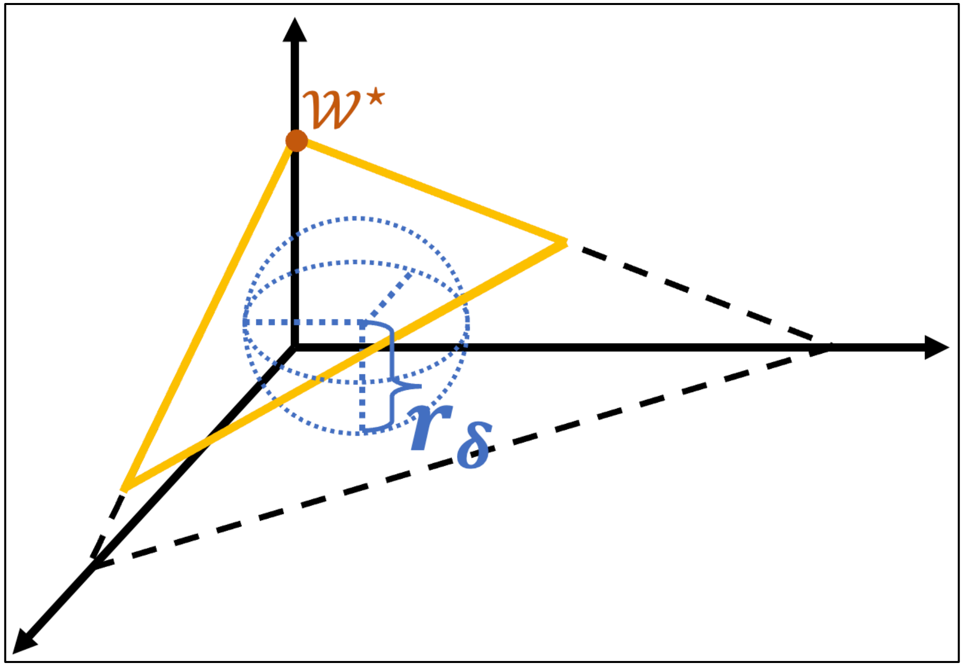
(D_δ is smaller when δ is small)



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

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

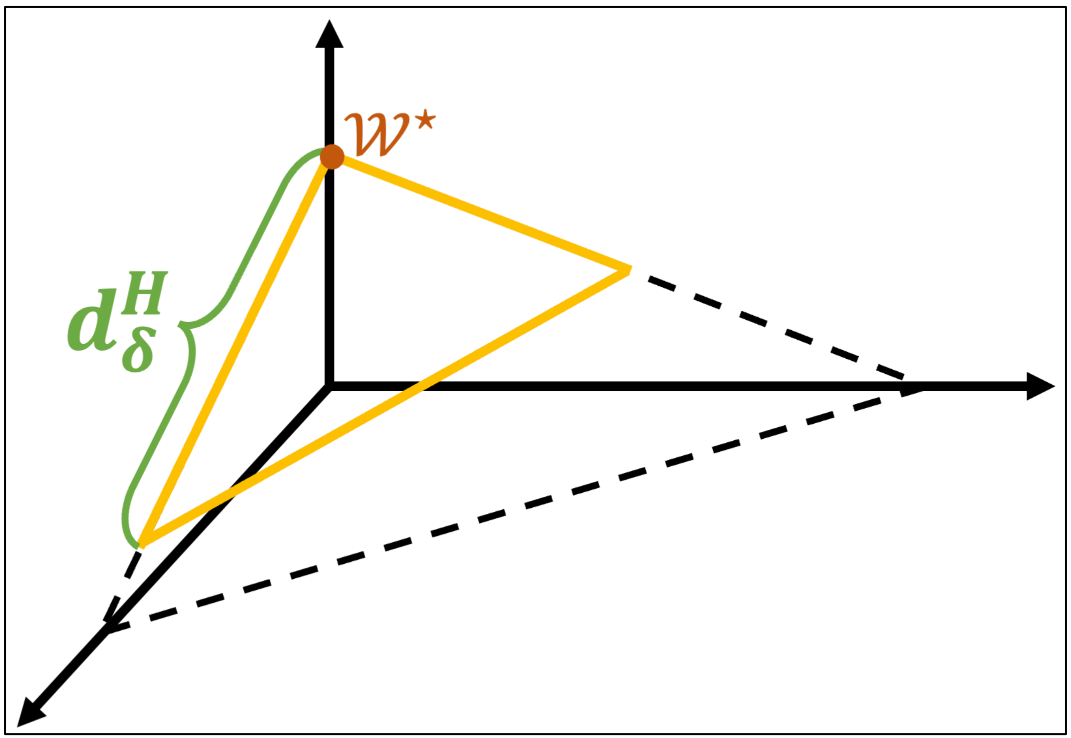
r_δ is the radius of the maximum ball inscribed in $\mathcal{K} \times \mathcal{K}^\star$ and centered at a point in $\mathcal{W}_\delta$



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

Worst-case complexity of PDHG

To get some intuition first, let's assume for the moment that the instance has unique optimum. Then:

Theorem [Xiong and F 2024]: Suppose $w^\star$ is unique. PDHG computes an $\boldsymbol{\varepsilon}$-optimal solution within

$$\tilde{O}\left(\kappa \cdot \lim_{\delta\to 0} \frac{\boldsymbol{D_\delta}}{\boldsymbol{r_\delta}} \cdot \ln\left(\frac{1}{\varepsilon}\right)\right)$$

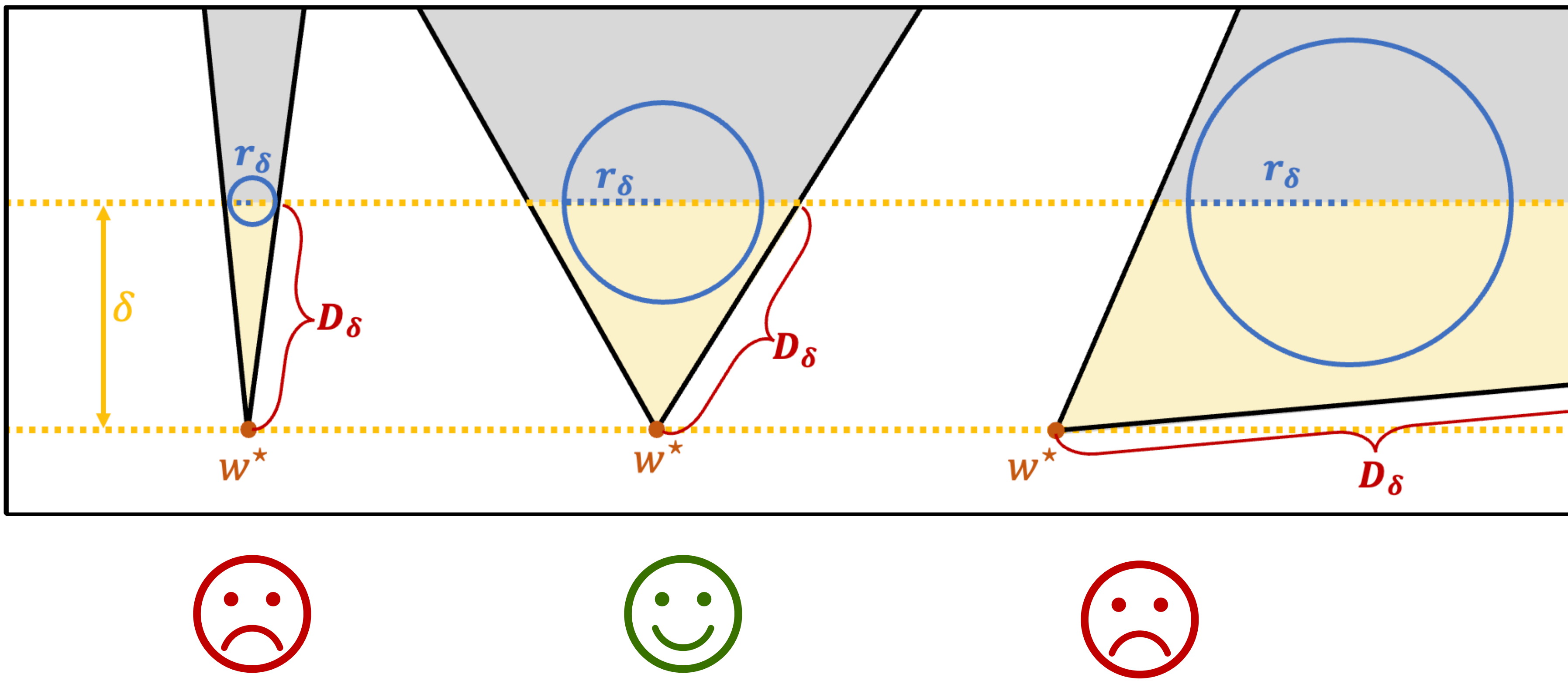
iterations.

Matrix condition number of A:
$\kappa := \sigma_{\max}^{+}(A)/\sigma_{\min}^{+}(A)$

"Sublevel-set geometry"

Local Geometry of and $\lim_{\delta \to 0} \frac{D_\delta}{r_\delta}$ in the case of LP

When $w^\star$ is unique and δ is sufficiently small, $\mathcal{W}_\delta$ is a slice of a pointed cone at $w^\star$.



Very small r_δ
Intermediate D_δ

Intermediate r_δ
Intermediate D_δ

Intermediate r_δ
Very large D_δ

Worst-case complexity of PDHG (under unique optima)

Matrix condition number of A: $\kappa = \sigma^+_{\max}(A)/\sigma^+_{\min}(A)$

Theorem [Xiong and F 2024]: Suppose $w^\star$ is unique. PDHG computes an $\boldsymbol{\varepsilon}$-optimal solution within

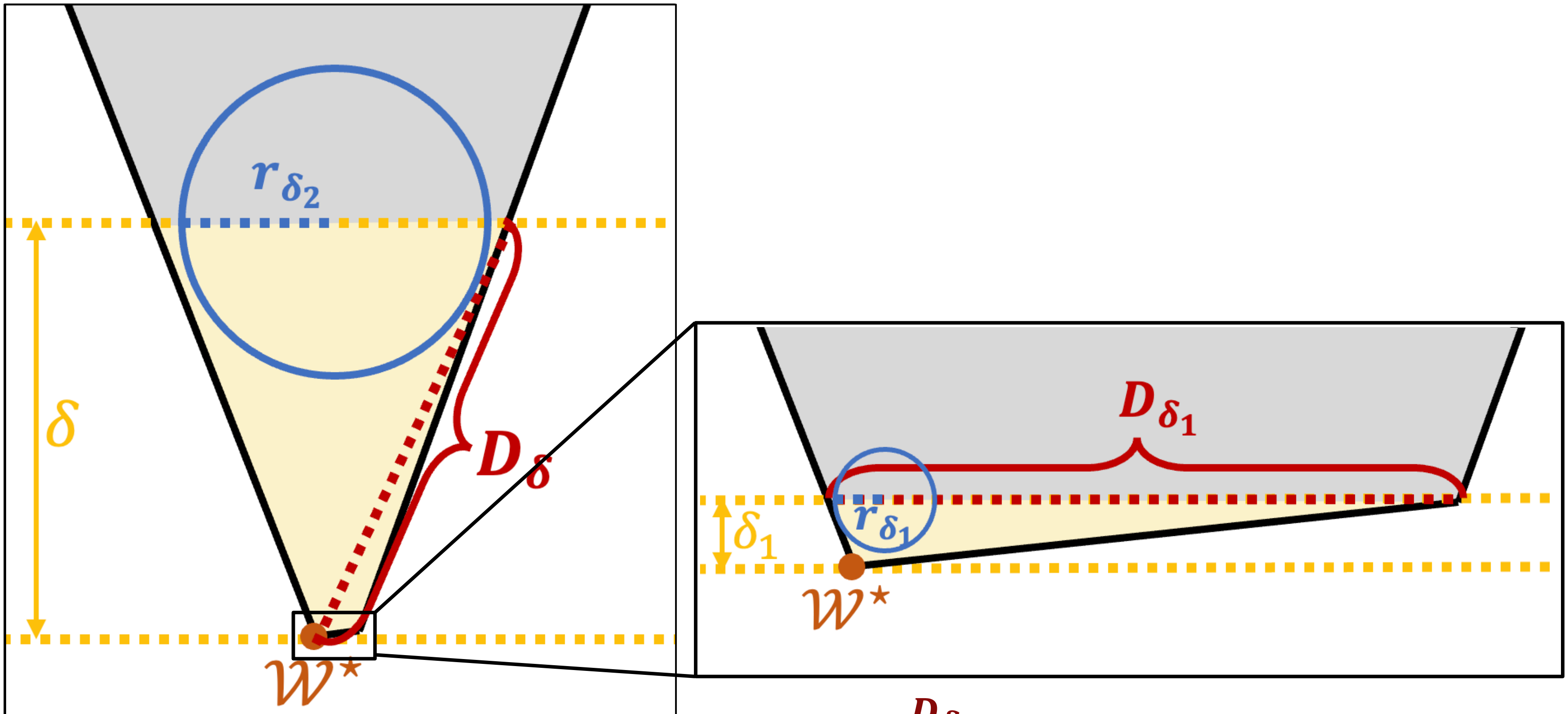
$$\tilde{O}\left(\kappa \cdot \lim_{\delta \to 0} \frac{\boldsymbol{D_\delta}}{\boldsymbol{r_\delta}} \cdot \ln\left(\frac{1}{\varepsilon}\right)\right)$$

iterations.

Matrix condition number	Local geometric condition

- This iteration bound is "superior" to the Hoffman constant
- For LP, this bound is $\tilde{O}\left(n^{2.5} \cdot \ln\left(\frac{1}{\varepsilon}\right)\right)$ with high probability [Xiong, 2024]
- For LP, this bound has a closed-form expression [Xiong, 2024]

Another example

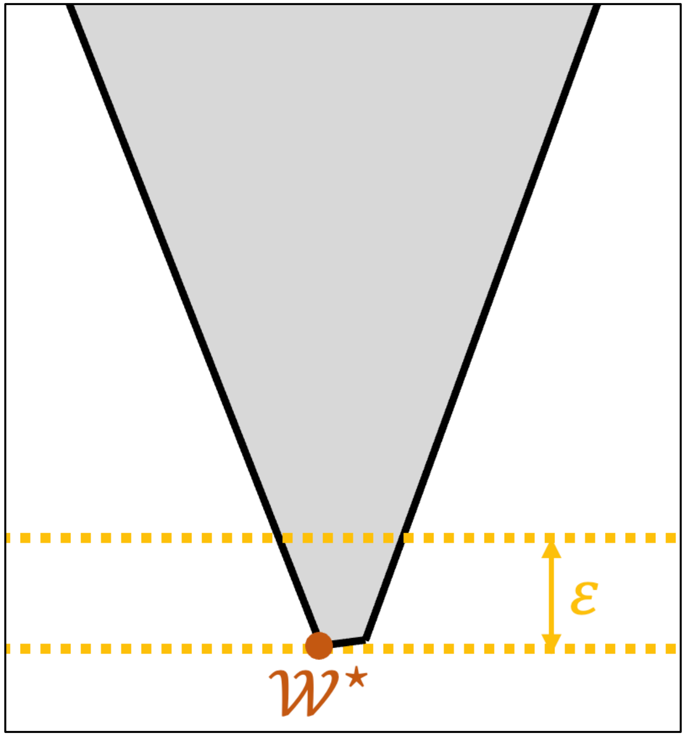


For $\delta_2 > \delta_1$, $\frac{D_{\delta_2}}{r_{\delta_2}}$ becomes smaller/better

$\frac{D_{\delta_1}}{r_{\delta_1}}$ is very large/bad (due to the small r_{δ_1})

Is $\lim_{\delta \to 0} \frac{D_\delta}{r_\delta}$ the only geometric condition?

Suppose we want an ε-optimal solution:



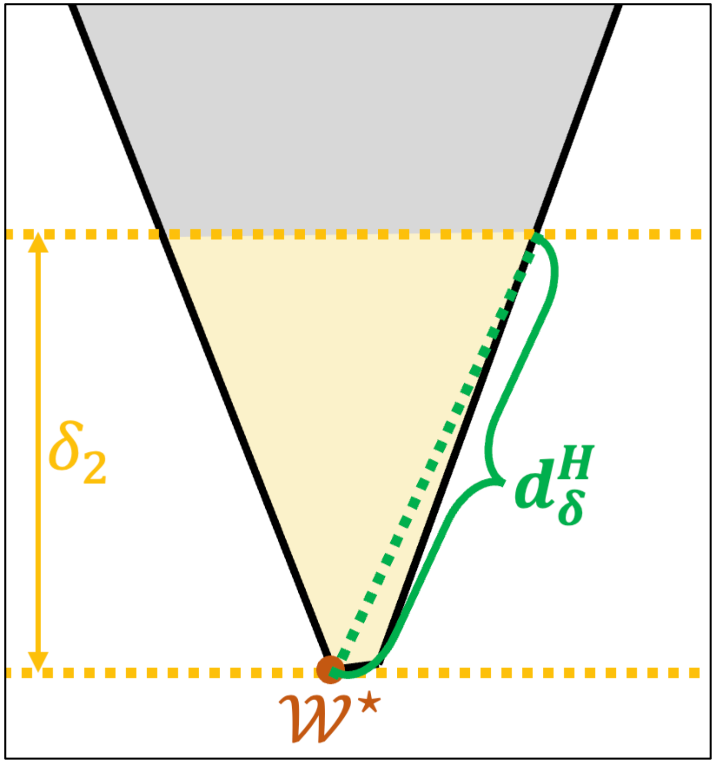
Intuition: The very-local bad geometry should not have a significant impact when the iterates of the algorithm have not yet reached the local neighborhood.

Is $\lim_{\delta \to 0} \frac{D_\delta}{r_\delta}$ the only geometric condition?

We need a third geometric measure to define "being close to $\mathcal{W}^\star$"

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

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



Our General Conic Optimization Computational Guarantee

Theorem [Xiong and F 2024]: The number of PDHG iterations required to compute an $\boldsymbol{\varepsilon}$-optimal solution is upper bounded by:

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

for each $\delta > 0$.

How good the geometry of $\mathcal{W}_\delta$ is

How close $\mathcal{W}_\delta$ is to $\mathcal{W}^\star$

Remark: This result holds under multiple optima and general conic optimization.

$\boldsymbol{D_\delta}$: 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)

Our General Conic Optimization Computational Guarantee

Theorem [Xiong and F 2024]: The number of PDHG iterations required to compute an $\boldsymbol{\varepsilon}$-optimal solution is upper bounded by:

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

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

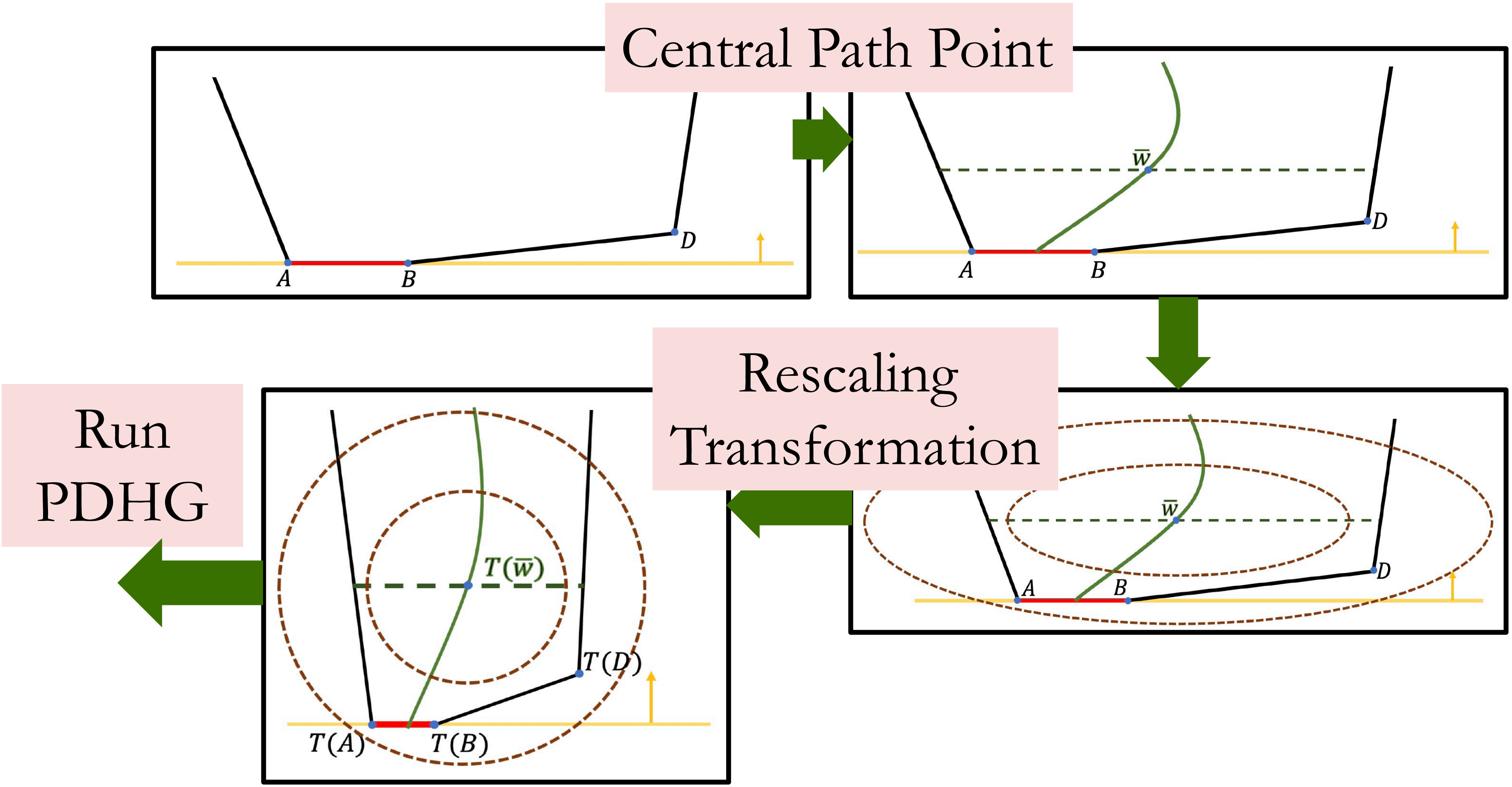
- Sublinear convergence part
- Note $\boldsymbol{d_\delta^H}$ is small when δ is small

The tightest bound is given by the T_δ 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 $\mathcal{W}^\star$
$\boldsymbol{\kappa} = \sigma_{\max}^+(A)/\sigma_{\min}^+(A)$

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

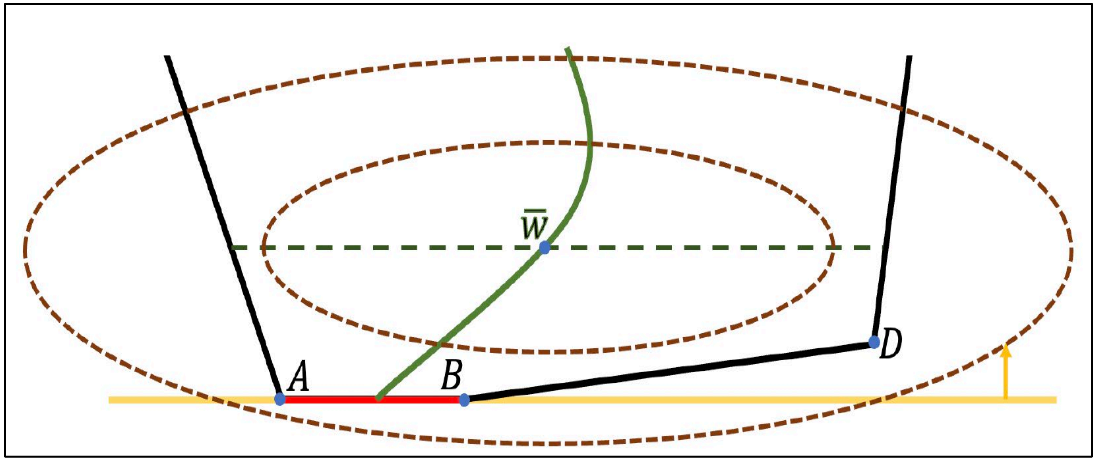
Central-path solutions are:

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

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

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

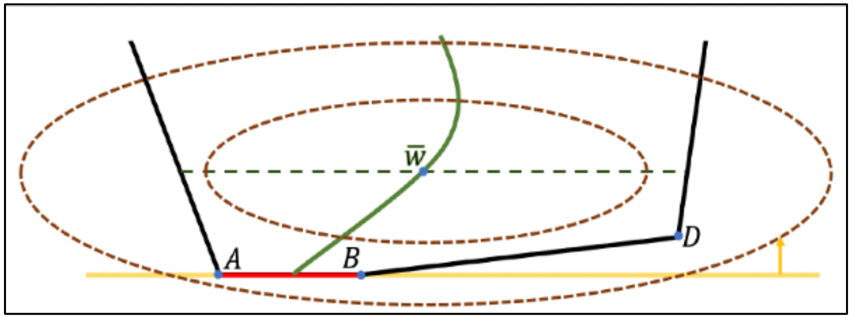
At $\overline{w} = (x_\mu, s_\mu)$, the local-norm ball nicely approximates the shape of sublevel sets:



Transformation based on a central-path solution

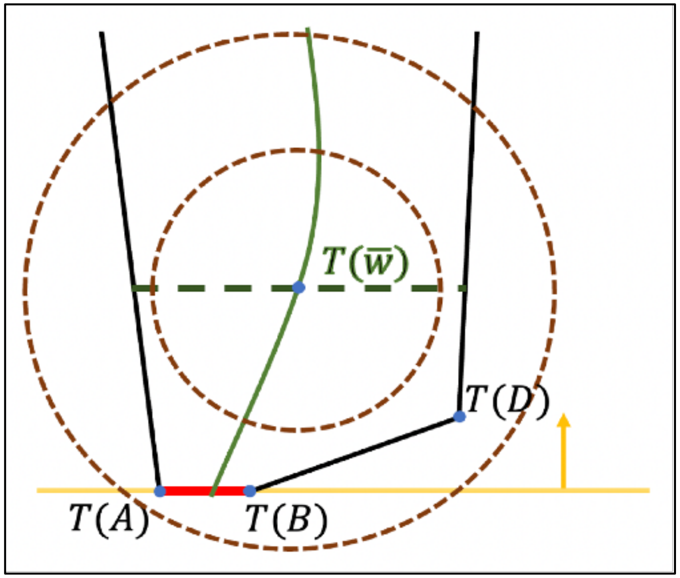
Let $H := \mu \cdot \nabla^2 F(x_\mu)$, then for $\bar{\boldsymbol{\delta}} := \boldsymbol{\vartheta_F} \cdot \boldsymbol{\mu}$ we have:

- $D_{\bar{\delta}} \leq 2\vartheta_F \cdot \sigma_{\min}^{+}(H)^{-1/2}$
- $r_{\bar{\delta}} \geq \sigma_{\max}^{+}(H)^{-1/2}$
- $d_{\bar{\delta}}^{H} \leq 2\vartheta_F \cdot \sigma_{\min}^{+}(H)^{-1/2}$



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

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



Proposed strategy

Suppose we do the following:

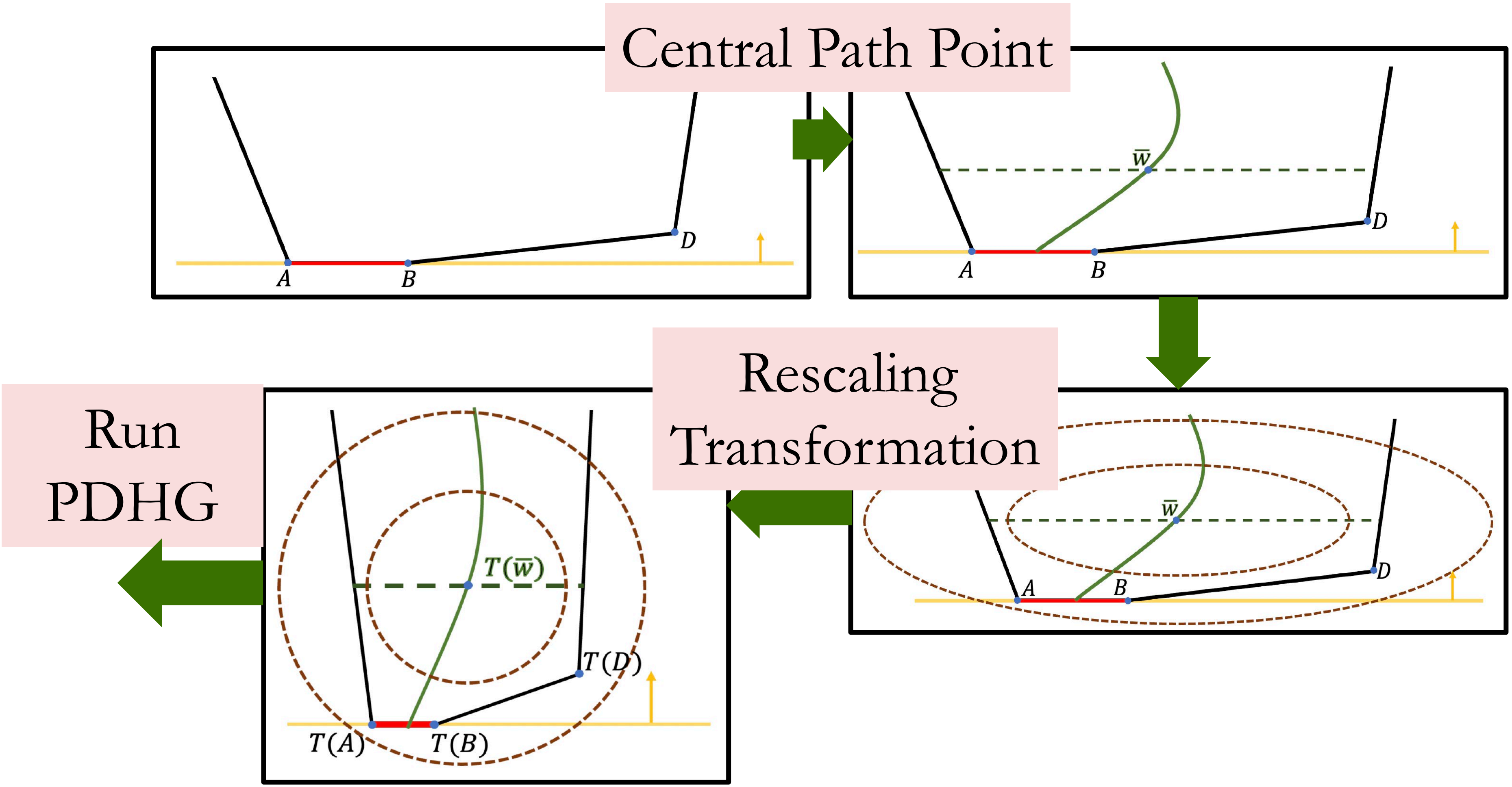
1. rescaling transformation using a central-path solution **with duality gap $\bar{\boldsymbol{\delta}}$**
2. row transformation to try to decrease closer to $\kappa = 1$

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

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

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

Summary



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

rPDHG-AHR (“Adaptive Hessian Rescaling”) and Computational Experiments

Proof of concept

PDHG-EasyColumn

Column rescaling to **normalize L_∞** column norms

PDHG-Central δ=0.1

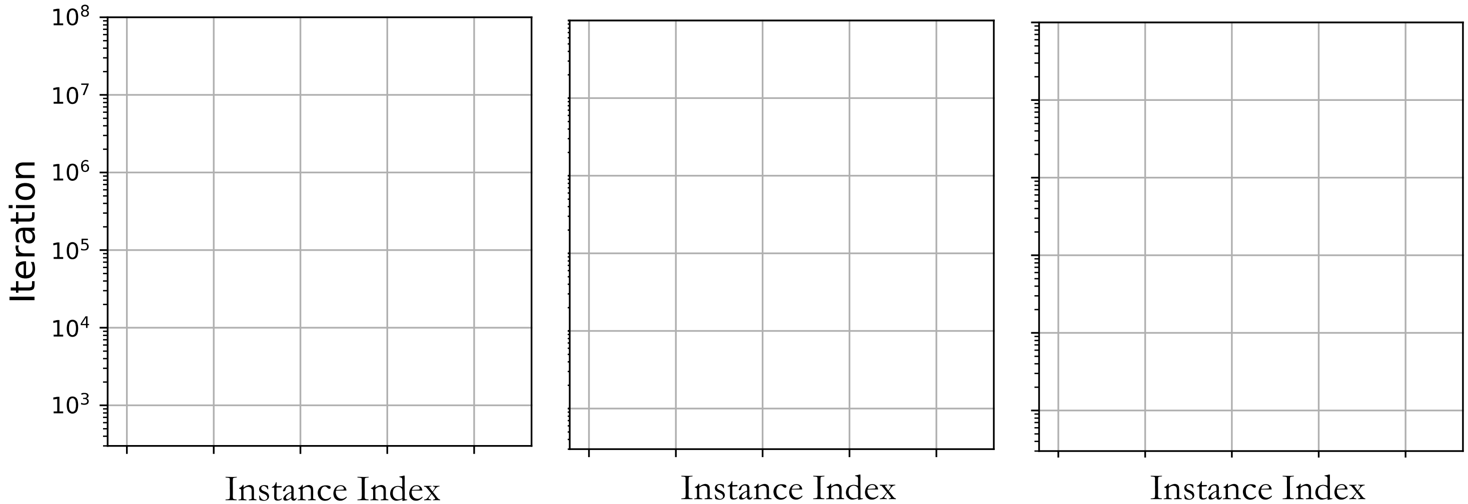
Column rescaling using **central-path solution** with KKT error **0.1**

PDHG-Central δ=0.01

Column rescaling using **central-path solution** with KKT error **0.01**

Note: we pre-multiply A by $(AA^\top)^{-1/2}$ to yield $\kappa = 1$ for all rescalings

PDHG iterations needed to compute a solution with KKT error **10^{-8}**



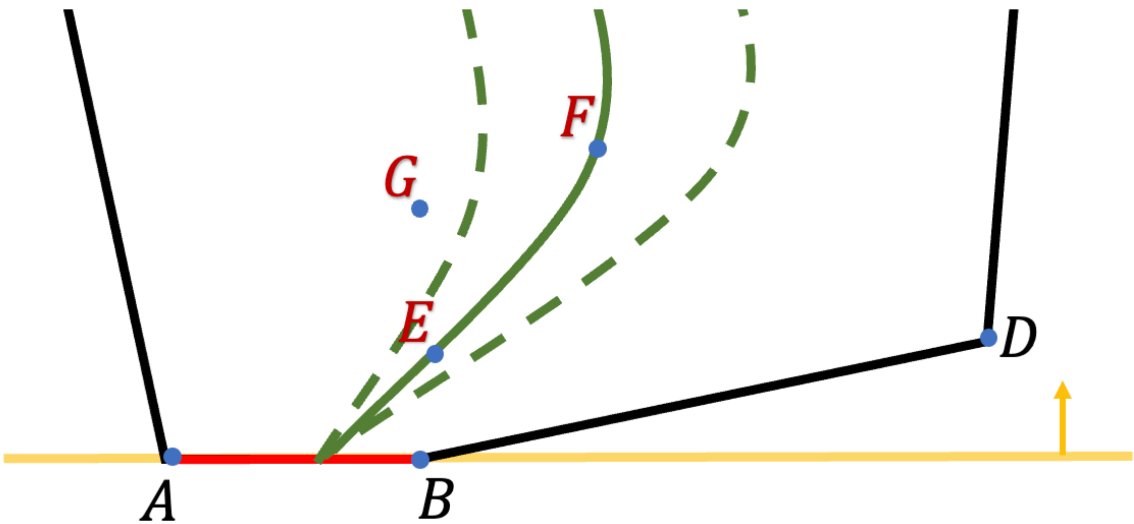
222 LP relaxation instances from the MIPLIB 2017 dataset

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 PDHG

- **We use a first-order method to compute the Newton steps of a "central path" solution**
  - we use a conjugate-gradient-method-based IPM (**CG-IPM**)
  - otherwise, implementation exactly 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 $\boldsymbol{\kappa}$**
  - 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 "$\boldsymbol{w_{int}}$-rescaled problem"

Using Adaptive Hessian Rescaling



Motivating concepts of Adaptive Hessian Rescaling:

- Adaptively balance the cost of computing the rescaling (CG-IPM) with the savings from running PDHG on the rescaled instance
- Try to identify a "good-enough" rescaling <u>as early as possible</u> and use the good-enough rescaling

Computational Experiments with **PDHG-AHR**

We compare:

1. **PDHG-AHR** : PDHG with Adaptive central-path Hessian Rescaling (using CG-IPM for the central-path computations)
2. **PDHG(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).

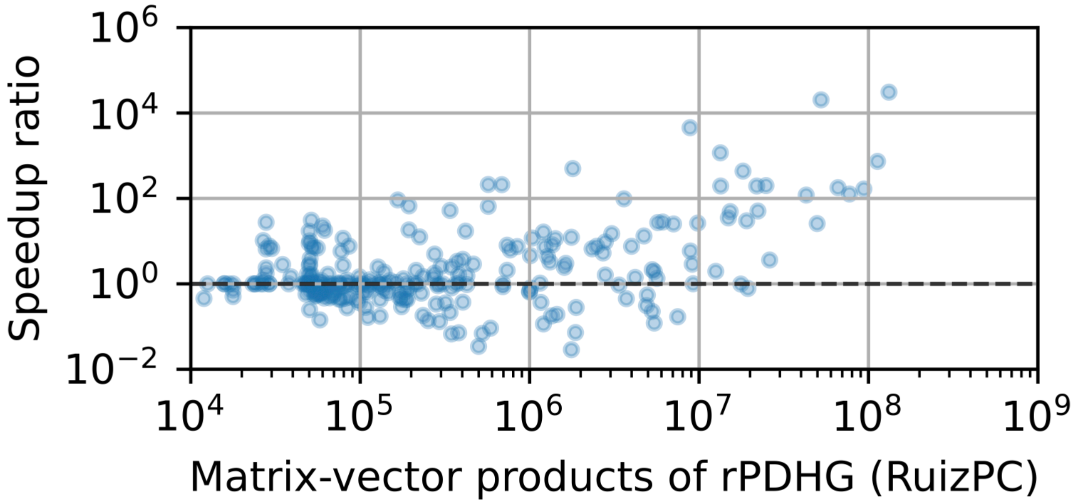
We performed tests on all the LP relaxations from MIPLIB 2017 dataset that are:

- large enough ($m \times n > 10^6$)
- but not too large for the IPM (number of non-zeros $< 10^5$)
- This yielded 413 instances in total

Computational Comparison: PDHG-AHR and PDHG(RuizPC)

Speedup Ratio:

$$\frac{\text{Matrix−vector products PDHG(RuizPC)}}{\text{Matrix−vector products PDHG−AHR}}$$



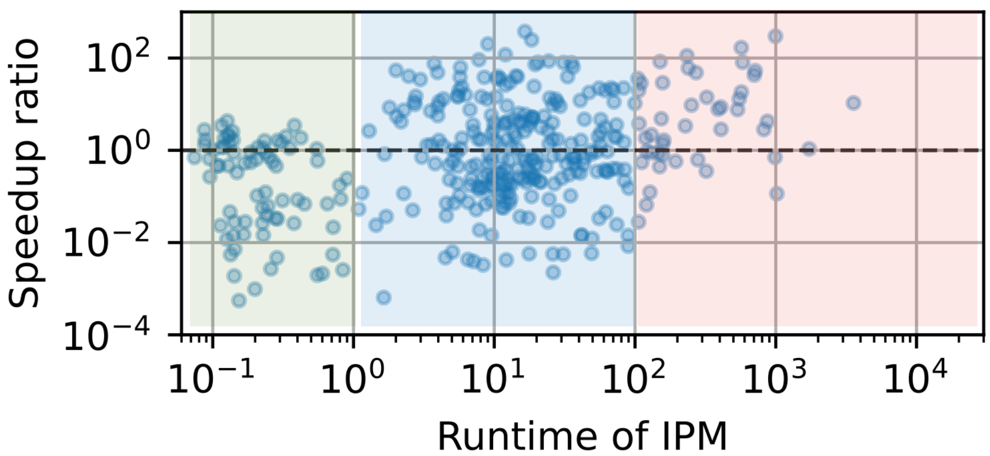
		Fraction solved by PDHG-AHR	
		S	**N**
Fraction solved by **PDHG (RuizPC)**	**S**	**87.7%**	**1.7%**
	N	**7.5%**	**3.1%**

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

Computational Comparison: PDHG-AHR and IPM

Speedup Ratio:

$$\frac{\text{Runtime IPM}}{\text{Runtime PDHG-AHR}}$$



		Fraction solved by PDHG−AHR: S	Fraction solved by PDHG−AHR: 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 **PDHG-AHR**.
- **PDHG-AHR** is also (slightly) more reliable than **IPM**

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: **PDHG-AHR** uses a "CG-IPM" to compute a low-accuracy central-path solution to obtain a good rescaling, and then use PDHG
- FOMs can compete and outperform 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, …

Thank you!