HPR-LP: An implementation of an HPR method for solving linear programming[1]

Defeng Sun

The Hong Kong Polytechnic University

[1]Kaihuang Chen, Defeng Sun, Yancheng Yuan, Guojun Zhang, and Xinyuan Zhao. "HPR-LP: An implementation of an HPR method for solving linear programming." arXiv:2408.12179 (August 2024; Revised March 2025).

Outline

1. Introduction to Linear Programming

2. Convex Optimization: Accelerated Preconditioned ADMM

3. Linear Programming: Halpern Peaceman-Rachford Method

4. Numerical Results

5. Conclusion

1. LP Model and Its Dual Model

(P)

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & \langle c, x\rangle \\ \text{s.t.} \quad & A_1 x = b_1 \\ & A_2 x \geq b_2 \\ & x \in C \end{aligned}$$

(D)

$$\begin{aligned} \min_{y\in\mathbb{R}^m,\, z\in\mathbb{R}^n} \quad & -\langle b, y\rangle + \delta_D(y) + \delta_C^*(-z) \\ \text{s.t.} \quad & A^* y + z = c \end{aligned}$$

where: $A_1 \in \mathbb{R}^{m_1\times n}, A_2 \in \mathbb{R}^{m_2\times n}, b_1 \in \mathbb{R}^{m_1}, b_2 \in \mathbb{R}^{m_2}, c \in \mathbb{R}^n$

$C := \{x \in \mathbb{R}^n \mid l \leq x \leq u\}, l \in (\mathbb{R} \cup \{-\infty\})^n, u \in (\mathbb{R} \cup \{+\infty\})^n$

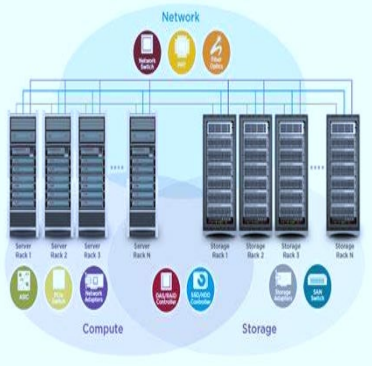
$A := [A_1; A_2] \in \mathbb{R}^{m\times n}, m = m_1 + m_2, b := [b_1; b_2] \in \mathbb{R}^m$

$D := \{y = (y_1, y_2) \in \mathbb{R}^{m_1} \times \mathbb{R}_+^{m_2}\}$

Applications of LP

Production Planning

By reasonably arranging production plans, optimizing resource allocation, and improving production efficiency.



Data Center Resource Allocation

Reasonably allocate limited resources to meet various demands and improve resource utilization efficiency.

Logistics and Transportation

Optimize transportation routes and methods to reduce transportation costs.

Financial Investment

Under risk and return constraints, formulate the optimal investment portfolio plan.

Q: Has the LP problem already been solved?

LP Algorithms and Solvers

01 LP Algorithms:

- ✓ Simplex Method (George Dantzig, 1947)
- ✓ Ellipsoid Algorithm (Khachiyan, 1979)
- ✓ Interior-Point Method: Karmarkar's Algorithm (1984)
- ✓ Splitting Algorithm: Divides a large LP problem into smaller subproblems by partitioning constraints or variables

02 LP Solvers:

- ✓ Commercial Solvers: Gurobi, Cplex, Mosek, COPT, MindOpt, OptVerse
- ✓ Open-Source Solvers: SCIP (ZIB), GLOP (Google), JuMP, CLP (COIN-OR)

Relevant Research Progress

- First-order methods have been used for linear programming since the **1950s** but remained inefficient in numerical computation.
- With the rise of **large-scale** LP problems, this direction has regained research interest.

01	SCS	Operator splitting /ADMM algorithm [O'Donoghue, Chu, Parikh, Boyd, 2016]
02	ABIP	ADMM-based interior point method [Lin et al., 2021], [Deng et al., 2024]
03	SNIPAL	Semismooth Newton based inexact proximal augmented Lagrangian method [Li, Sun, Toh, 2020]
04	ECLIPSE	Nesterov's accelerated gradient method [Basu, Ghoting, Mazumder, Pan, 2020]
05	PDLP	Primal-dual hybrid gradient method [Applegate et al., 2021] Nvidia: cuOPT

Challenges of Large-Scale LP Problems

- **Computational Complexity:** As LP problems grow larger, solving them takes excessive time, sometimes becoming impractical
- **Storage Limitations:** Large-scale LP problems demand high memory, posing challenges for embedded and mobile devices
- **Solution Efficiency:** Optimization, parallel, and distributed methods face challenges

Efficient optimization algorithms are needed to overcome memory constraints, reduce computational complexity, and improve solution efficiency.

2. Convex Optimization: Accelerated Preconditioned (Semi-Proximal) ADMM[2] (pADMM)

Convex Optimization problem (COP) $\Longrightarrow$

$$\min_{y\in\mathbb{Y},\, z\in\mathbb{Z}} \quad f_1(y) + f_2(z)$$
$$\text{s.t.} \quad \mathrm{B}_1\mathrm{y} + \mathrm{B}_2\mathrm{z} = \mathrm{c}$$

Dual problem (COD) $\Longrightarrow$

$$\max_{x\in\mathbb{X}} \{-f_1^*(-B_1^* x) - f_2^*(-B_2^* x) - \langle c, x\rangle\}$$

where: $\mathbb{X}, \mathbb{Y}, \mathbb{Z} : \textit{Finite} - \textit{dimensional real Euclidean spaces}$

$f_1 : \mathbb{Y} \to (-\infty, +\infty],\ f_2 : \mathbb{Z} \to (-\infty, +\infty] : \textit{proper closed convex functions}$

$B_1 : \mathbb{Y} \to \mathbb{X},\ B_2 : \mathbb{Z} \to \mathbb{X} : \textit{linear operators}, \quad c \in \mathbb{X}$

[2] Defeng Sun, Yancheng Yuan, Guojun Zhang, and Xinyuan Zhao. "Accelerating preconditioned ADMM via degenerate proximal point mappings." SIAM J. Optimization 35:2 (2025) 1165–1193.

Convex Optimization Method

- Augmented Lagrangian function: $\forall (y, z, x) \in \mathbb{Y} \times \mathbb{Z} \times \mathbb{X}$,

$$L_\sigma(y, z; x) := f_1(y) + f_2(z) + \langle x, B_1 y + B_2 z - c\rangle + \frac{\sigma}{2}||B_1 y + B_2 z - c||^2$$

- Operators Σ_{f_1} and Σ_{f_2} satisfy:
  - For any $y, \hat{y} \in dom(f_1)$, $\phi \in \partial f_1(y)$ and $\hat{\phi} \in \partial f_1(\hat{y})$

$$f_1(y) \geq f_1(\hat{y}) + \langle \hat{\phi}, y - \hat{y}\rangle + \frac{1}{2}\|y - \hat{y}\|^2_{\Sigma_{f_1}}, \ \langle \phi - \hat{\phi}, y - \hat{y}\rangle \geq \|y - \hat{y}\|^2_{\Sigma_{f_1}}$$

  - For any $z, \hat{z} \in dom(f_2)$, $\varphi \in \partial f_2(z)$ and $\hat{\varphi} \in \partial f_z(\hat{z})$

$$f_2(z) \geq f_2(\hat{z}) + \langle \hat{\varphi}, z - \hat{z}\rangle + \frac{1}{2}\|z - \hat{z}\|^2_{\Sigma_{f_2}}, \ \langle \varphi - \hat{\varphi}, z - \hat{z}\rangle \geq \|z - \hat{z}\|^2_{\Sigma_{f_2}}$$

Alg1. A pADMM[3] for COP

Input: Choose $\mathcal{T}_1(\succcurlyeq 0)$ and $\mathcal{T}_2(\succcurlyeq 0)$ such that

$$\Sigma_{f_1} + B_1^* B_1 + \mathcal{T}_1 \succ 0,\ \Sigma_{f_2} + B_2^* B_2 + \mathcal{T}_2 \succ 0$$

Choose $w^0 = (y^0, z^0, x^0)$. Let $\sigma > 0$ and $\rho \in (0,2]$.

$k = 0,1, \dots,$

1. $\bar{w}^{k+1} = \textbf{UpdateStep}(w^k, \mathcal{T}_1, \mathcal{T}_2, \sigma)$:

$$\begin{cases} \bar{z}^{k+1} = \operatorname{argmin}_{z\in\mathbb{Z}} \left\{ L_\sigma(y^k, z; x^k) + \dfrac{\sigma}{2}\|z - z^k\|^2_{\mathcal{T}_2} \right\} \\ \bar{x}^{k+1} = x^k + \sigma\left(B_1 y^k + B_2 \bar{z}^{k+1} - c\right) \\ \bar{y}^{k+1} = \operatorname{argmin}_{y\in\mathbb{Y}} \left\{ L_\sigma(y, \bar{z}^{k+1}; \bar{x}^{k+1}) + \dfrac{\sigma}{2}\|y - y^k\|^2_{\mathcal{T}_1} \right\} \end{cases}$$

2. $w^{k+1} = (1-\rho) w^k + \rho \bar{w}^{k+1}$

- **Related works:** ADMM[4, 5], Generalized ADMM[6], semi-proximal ADMM[7]
- $\boldsymbol{\rho = 2}$: Semi-proximal Peaceman-Rachford (PR) algorithm

[3]Xiao, Chen, and Li. Math. Program. Comput. (2018): 533-555.

[4]Glowinski and Marroco. Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique (1975): 41-76.

[5]Gabay and Mercier. Comput. Math. Appl. (1976): 17-40.

[6]Eckstein and Bertsekas. Math. Program. (1992): 293-318.

[7]Fazel, Pong, Sun, and Tseng. SIAM J. Matrix Anal. Appl. (2013): 946-977.

A pADMM for Solving LP

- Augmented Lagrangian function for LP:

$$L_\sigma^{LP}(y, z; x) := -\langle b, y\rangle + \delta_D(y) + \delta_C^*(-z) + \langle x, A^*y + z - c\rangle + \frac{\sigma}{2}\|A^*y + z - c\|^2$$

Alg2. A pADMM for LP

Input: Choose $\mathcal{T}_1 (\succcurlyeq 0)$ such that $\mathcal{T}_1 + AA^* \succ 0$ and $w^0 = (y^0, z^0, x^0) \in D \times \mathbb{R}^n \times \mathbb{R}^n$.
Let $\sigma > 0$ and $\rho \in (0,2]$.

$k = 0,1, \dots,$

1. $\bar{w}^{k+1} = \mathbf{UpdateStep}(w^k, \mathcal{T}_1, \sigma)$:

$$\begin{cases} \bar{z}^{k+1} = \operatorname{argmin}_{z \in \mathbb{R}^n} \left\{ L_\sigma^{LP}(y^k, z; x^k) + \frac{\sigma}{2}\|z - z^k\|_{\mathcal{T}_2}^2 \right\} \\ \bar{x}^{k+1} = x^k + \sigma(A^*y^k + \bar{z}^{k+1} - c) \\ \bar{y}^{k+1} = \operatorname{argmin}_{y \in \mathbb{R}^n} \left\{ L_\sigma^{LP}(y, \bar{z}^{k+1}; \bar{x}^{k+1}) + \frac{\sigma}{2}\|y - y^k\|_{\mathcal{T}_1}^2 \right\} \end{cases}$$

2. $w^{k+1} = (1-\rho)w^k + \rho\bar{w}^{k+1}$

Table 1: Existing Complexity Results of pADMM-type Algorithms

Reference	M&S(2013)[9]	D&Y(2016)[10]	Cui(2016)[10]
Algorithm	ADMM	ADMM	maj. ADMM
Dual step	1	1	$(0,\left(1+\sqrt{5}\right)/2)$
Prim. feas.	$O(1/k)$	$o(1/\sqrt{k})$	$o(1/\sqrt{k})$
Obj. err.	-	$o(1/\sqrt{k})$	-
KKT res.	$O(1/k)$ ε-subdifferential residual	-	$O(1/\sqrt{k})$
Type	Ergodic	Non-ergodic	Non-ergodic

Chambolle and Pock[11,12] : $O(1/k)$ **ergodic** iteration complexity for the **PDHG** (linearized ADMM) in terms of a certain **primal-dual gap** (derived by primal feasibility and objective error), which may not imply the desired $O(1/k)$ complexity bound for the KKT residual.

[8] Monteiro and Svaiter. SIAM J. Optim. (2013): 475-507. (First version: August 2010).
[9] Davis and Yin. Splitting methods in communication, imaging, science, and engineering (2016): 115-163.
[10] Cui, Li, Sun, and Toh. J. Optim. Theory Appl. (2016): 1013-1041.
[11] Chambolle and Pock. J. Math. Imaging Vis. (2011): 120-145. (Online: May 2010).
[12] Chambolle and Pock. Math. Program. (2016): 253-287.

Q: The KKT residual of pADMM is at most $O(1/\sqrt{k})$. How can we accelerate pADMM to achieve $O(1/k)$ or better?

Roadmap

preconditioned (semi-proximal) ADMM (pADMM)

equivalent

degenerate proximal point method (dPPM)

Halpern's iteration

accelerated pADMM

equivalent

Accelerated dPPM

Karush-Kuhn-Tucker (KKT) System and Its Equivalent Problems

01

The KKT system for COP

$$
\begin{aligned}
-B_1^* x^* &\in \partial f_1(y^*) \\
-B_2^* x^* &\in \partial f_2(z^*) \\
B_1 y^* + B_2 z^* - c &= 0
\end{aligned}
$$

02

$\forall\, w = (y, z, x) \in \mathbb{W} := \mathbb{Y} \times \mathbb{Z} \times \mathbb{X}$,

$$
\mathbf{0} \in \mathcal{T}\boldsymbol{w} = \begin{pmatrix} \partial f_1(y) + B_1^* x \\ \partial f_2(z) + B_2^* x \\ c - B_1 y - B_2 z \end{pmatrix} \quad (\star)
$$

Assumption 2.1

(KKT) system has a non-empty solution set

Degenerate proximal point method

Definition 2.1 [Bredies et al. (2022) [13]]

An **admissible preconditioner** $M\colon \mathbb{W} \to \mathbb{W}$ for the operator $\mathcal{T}\colon \mathbb{W} \to 2^{\mathbb{W}}$ is a linear, bounded self-adjoint, and **positive semidefinite** operator such that

$$\hat{\mathcal{T}} = (M + \mathcal{T})^{-1} M$$

is single-valued and has full domain.

degenerate PPM (dPPM):

$$\overline{w}^{k+1} = \hat{\mathcal{T}} w^k \qquad \left[\Leftrightarrow 0 \in \mathcal{T}\left(\overline{w}^{k+1}\right) + M\left(\overline{w}^{k+1} - w^k\right)\right]$$

$$w^{k+1} = (1-\rho) w^k + \rho \overline{w}^{k+1},$$

where $w^0 \in \mathbb{W}$ and $\rho \in (0,2]$.

[13] Bredies, Chenchene, Lorenz, and Naldi. SIAM J. Optim. (2022): 2376-2401.

The Equivalence of pADMM and dPPM

Linear operator $M\colon \mathbb{W} \to \mathbb{W}$:

$$M = \begin{bmatrix} \sigma B_1^* B_1 + \sigma \mathcal{T}_1 & 0 & B_1^* \\ 0 & \sigma \mathcal{T}_2 & 0 \\ B_1 & 0 & \sigma^{-1} I \end{bmatrix}$$

Proposition 2.1

If starting from the same initial point $w^0 \in \mathbb{W}$, the sequences $\{w^k\}$ generated by **pADMM** (Algorithm 1) and **dPPM** are **identical**. Moreover, M is an **admissible preconditioner**, and $(M + \mathcal{T})^{-1}$ **Lipschitz continuous**.

An Accelerated dPPM

- Define $\hat{F}_\rho := (1-\rho)I + \rho\hat{\mathcal{T}}$ with $\rho \in (0,2]$. $\hat{F}_\rho$ is M-nonexpansive, i.e.

$$\left\|\hat{F}_\rho w - \hat{F}_\rho w'\right\|_M \leq \|w - w'\|_M, \qquad \forall\, w, w' \in \mathbb{W}$$

Alg3. An accelerated dPPM for the inclusion problem $(\star)$

Input: Let $w^0 \in \mathbb{W}$ and $\rho \in (0,2]$

$k = 0,1,\ldots,$

1. $\overline{w}^{k+1} = \hat{\mathcal{T}} w^k$
2. $\widehat{w}^{k+1} = \hat{F}_\rho w^k = (1-\rho)w^k + \rho\overline{w}^{k+1}$
3. $w^{k+1} = \frac{1}{k+2} w^0 + \frac{k+1}{k+2}\widehat{w}^{k+1}$ Halpern's iteration[14,15]

[14]Halpern, B. Bull. Am. Math. Soc. 73(6), 957–961 (1967)

[15]Lieder, F. Optim. Lett. 15(2), 405–418 (2021)

Convergence Properties of the Accelerated dPPM

Theorem 2.1

If Assumption 2.1 holds and M is an **admissible preconditioner** such that $(M+\mathcal{T})^{-1}$ is continuous, then the sequence $\{\overline{w}^k\}$ generated by Algorithm 3 converges to a point $w^* \in \mathcal{T}^{-1}(0)$.

Proposition 2.2

If Assumption 2.1 holds and M is an **admissible preconditioner**, then the sequences $\{w^k\}$ *and* $\{\widehat{w}^k\}$ generated by Algorithm 3 satisfy

$$\left\|w^k - \widehat{w}^{k+1}\right\|_M \leq \frac{2\|w^0 - w^*\|_M}{k+1}, \quad \forall k \geq 0, w^* \in \mathcal{T}^{-1}(0).$$

- Without acceleration, dPPM[16] with ρ = 1 has $O\left(\frac{1}{\sqrt{k}}\right)$ iteration complexity for $\left\|w^k - w^{k-1}\right\|_M$

[16] Brézis, Haim, and Pierre Louis Lions. Israel J. Math. (1978): 329-345.

3. Halpern-Peaceman-Rachford (HPR) Method

Alg4. An HPR method with semi-proximal terms for LP

Input: Choose $\mathcal{T}_1 (\succeq 0)$ such that $\mathcal{T}_1 + AA^* \succ 0$ and $w^0 = (y^0, z^0, x^0) \in D \times \mathbb{R}^n \times \mathbb{R}^n$, $\sigma > 0$

$k = 0,1, \dots,$

1. $\bar{w}^{k+1} = \textbf{UpdateStep}(w^k, \mathcal{T}_1, \sigma)$:

$$\begin{cases} \bar{z}^{k+1} = \operatorname{argmin}_{z \in \mathbb{R}^n} \left\{ L_\sigma^{LP}(y^k, z; x^k) + \frac{\sigma}{2} \|z - z^k\|_{\mathcal{T}_2}^2 \right\} \\ \bar{x}^{k+1} = x^k + \sigma(A^* y^k + \bar{z}^{k+1} - c) \\ \bar{y}^{k+1} = \operatorname{argmin}_{y \in \mathbb{R}^n} \left\{ L_\sigma^{LP}(y, \bar{z}^{k+1}; \bar{x}^{k+1}) + \frac{\sigma}{2} \|y - y^k\|_{\mathcal{T}_1}^2 \right\} \end{cases}$$

2. $\hat{w}^{k+1} = 2\bar{w}^{k+1} - w^k$

3. $w^{k+1} = \frac{1}{k+2} w^0 + \frac{k+1}{k+2} \hat{w}^{k+1}$

Theorem 2.2 The complexity results of the HPR method for LP

Let $R_0 = \|w^0 - w^*\|_M$. If assumption 2.1 holds，then the sequence $\{\bar{w}^k\}$ generated by Algorithm 4 satisfies

Primal infeasibility →
Complementarity →
Dual infeasibility →

$$\left\|\begin{pmatrix} \bar{y}^{k+1} - \Pi_D\left(\bar{y}^{k+1} - A\bar{x}^{k+1} + b\right) \\ \bar{x}^{k+1} - \Pi_C\left(\bar{x}^{k+1} - \bar{z}^{k+1}\right) \\ c - A^*\bar{y}^{k+1} - \bar{z}^{k+1} \end{pmatrix}\right\| \leq \left(\frac{\sigma\left(\|A\|+\|\sqrt{\mathcal{T}_1}\|\right)+1}{\sqrt{\sigma}}\right)\frac{R_0}{k+1},$$

Dual objective function error →

$$\left(\frac{-\|x^*\|}{\sqrt{\sigma}}\right)\frac{R_0}{k+1} \leq h^{LP}\left(\bar{y}^{k+1}, \bar{z}^{k+1}\right) \leq \left(3R_0 + \frac{\|x^*\|}{\sqrt{\sigma}}\right)\frac{R_0}{k+1},$$

where $h^{LP}\left(\bar{y}^{k+1}, \bar{z}^{k+1}\right) := -\langle b, \bar{y}^{k+1}\rangle + \delta_C^*\left(-\bar{z}^{k+1}\right) - \left(-\langle b, y^*\rangle + \delta_C^*(-z^*)\right)$

Alg5：HPR-LP

1. Input: Choose $\mathcal{T}_1 (\succcurlyeq 0)$ such that $\mathcal{T}_1 + AA^* \succ 0$, and $w^{0,0} := (y^{0,0}, z^{0,0}, x^{0,0}) \in D \times \mathbb{R}^n \times \mathbb{R}^n$
2. Initialization: Set parameter $\sigma_0 > 0$ and $r = 0$
3. Outer loop:

3.1 Inner loop initialization: $\mathrm{t} = 0$

3.2 Inner loop: $\bar{w}^{r,t+1} = \mathbf{UpdateStep}(w^{r,t}, \mathcal{T}_1, \sigma_r)$

$$\hat{w}^{r,t+1} = \mathbf{2}\bar{w}^{r,t+1} - w^{r,t}$$

$$w^{r,t+1} = \frac{1}{t+2} w^{r,0} + \frac{t+1}{t+2} \hat{w}^{r,t+1}$$

If **one of the restart criteria** or **all termination criteria** are met, then output:

$$w^{r+1,0} = \bar{w}^{r,t+1}$$

Otherwise, $\mathrm{t} = t + 1$

4. If **all termination criteria** are met, output: $w^{r+1,0}$; otherwise,

$$\sigma_{r+1} = \mathbf{SigmaUpdate}(w^{r+1,0}, w^{r,0}, \mathcal{T}_1, A)$$

5. $r = r + 1$

Restart criteria for HPR-LP

Proposition 2.2

If Assumption 2.1 holds and M is an admissible preconditioner, then the sequences $\{w^k\}$ and $\{\widehat{w}^k\}$ generated by Algorithm 3 satisfy

$$\left\|w^k - \widehat{w}^{k+1}\right\|_M \leq \frac{2\left\|w^0 - w^*\right\|_M}{k+1}, \quad \forall k \geq 0, w^* \in \mathcal{T}^{-1}(0).$$

Based on $O(1/k)$ iteration complexity for the KKT residual, define the merit function:

$$R_{r,t} := \|w^{r,t} - w^*\|_M, \qquad \forall r \geq 0, t \geq 0$$

Approximate $R_{r,t}$ by

$$\tilde{R}_{r,t} = \|w^{r,t} - \widehat{w}^{r,t+1}\|_M$$

Restart criteria for HPR-LP

$\tilde{R}_{r,t} = \|w^{r,t} - \hat{w}^{r,t+1}\|_M$

- Sufficient decay of $\tilde{R}_{r,t}$:

$$\tilde{R}_{r,t} \leq \alpha_1 \tilde{R}_{r,0};$$

- Necessary decay + no local progress of $\tilde{R}_{r,t}$:

$$\tilde{R}_{r,t} \leq \alpha_2 \tilde{R}_{r,0}, \qquad \tilde{R}_{r,t+1} > \tilde{R}_{r,t};$$

- Long inner loop:

$$t \geq \alpha_3 k,$$

where $\alpha_1 \in (0, \alpha_2)$, $\alpha_2 \in (0,1)$, $\alpha_3 \in (0,1)$.

HPR-LP: Update rule for σ

$$M = \begin{bmatrix} \sigma A^* A + \sigma \mathcal{T}_1 & 0 & A^* \\ 0 & 0 & 0 \\ A^* & 0 & \sigma^{-1} I \end{bmatrix}$$

$$\sigma_{r+1} = \arg\min_{\sigma} \|w^{r+1,0} - w^*\|_M^2$$

$$= \arg\min_{\sigma} \left(\sigma \|y^{r+1,0} - y^*\|_{\mathcal{T}_1}^2 + \sigma^{-1} \|x^{r+1,0} - x^* + \sigma A^*(y^{r+1,0} - y^*)\|^2 \right)$$

$$= \sqrt{\frac{\|x^{r+1,0} - x^*\|^2}{\|y^{r+1,0} - y^*\|_{\mathcal{T}_1}^2 + \|A^*(y^{r+1,0} - y^*)\|^2}}$$

In **HPR-LP, update σ** with the approximation:

$$\sigma_{r+1} \approx \sqrt{\frac{\|x^{r+1,0} - x^{r,0}\|^2}{\|y^{r+1,0} - y^{r,0}\|_{\mathcal{T}_1}^2 + \|A^*(y^{r+1,0} - y^{r,0})\|^2}}$$

HPR-LP: Update rule for σ-example

- For LP with a special structure, such as optimal transport[17], set $\mathcal{T}_1 = 0$

$$\sigma_{r+1} \approx \frac{\|x^{r+1,0} - x^{r,0}\|}{\|A^*(y^{r+1,0} - y^{r,0})\|}$$

- For general LP, set $\mathcal{T}_1 = \lambda I_m - AA^*, \ \lambda \geq \lambda_1(AA^*)$

$$\sigma_{r+1} \approx \frac{1}{\sqrt{\lambda}} \frac{\|x^{r+1,0} - x^{r,0}\|}{\|A^*(y^{r+1,0} - y^{r,0})\|}$$

[17] Zhang, Gu, Yuan, and Sun. ”HOT: An Efficient Halpern Accelerating Algorithm for Optimal Transport Problems.” arXiv preprint arXiv:2408.00598 (2024). IEEE Transactions on Pattern Analysis and Machine Intelligence (2025).

4. Numerical experiment (computing environment)

- HPR-LP: implemented in Julia, referred to as HPR-LP.jl;
- cuPDLP.jl[18] : the GPU version of the award-winning solver PDLP[19] , also implemented in Julia;
- All experiments are conducted on a SuperServer SYS-420GP-TNR.
  - HPR-LP.jl and cuPDLP.jl : NVIDIA A100-SXM4-80GB GPU, CUDA 12.3
  - Gurobi 11.0.3 (academic license): Intel Xeon Platinum8338C CPU 2.60GHz with 256GB RAM

[18] Lu and Yang. arXiv preprint arXiv:2311.12180 (2023).
[19] Applegate, D'ıaz, Hinder, Lu, Lubin, O'Donoghue, and Schudy were awarded the **Beale–Orchard-Hays Prize** for Excellence in ISMP2024, July 21-26, 2024, Montr'eal, Canada.

Termination criteria and SGM10

- **Termination criteria (Optimality):** HPR-LP (PDLP) terminates when the stopping criteria are met for tolerance $\varepsilon \in (0, \infty)$
  - Relative duality gap $|\langle b, y\rangle - \delta_C^*(-z) - \langle c, x\rangle| \leq \varepsilon(1 + |\langle b, y\rangle - \delta_C^*(-z) + \langle c, x\rangle|)$
  - Relative primal infeasibility $\|\prod_D(b - Ax)\| \leq \varepsilon(1 + \|b\|)$
  - Relative dual infeasibility $\|c - A^*y - z\| \leq \varepsilon(1 + \|c\|)$

- **SGM10:** the shifted geometric mean of solving time

$$\left(\prod_{i=1}^{n}(t_i + 10)\right)^{1/n} - 10,$$

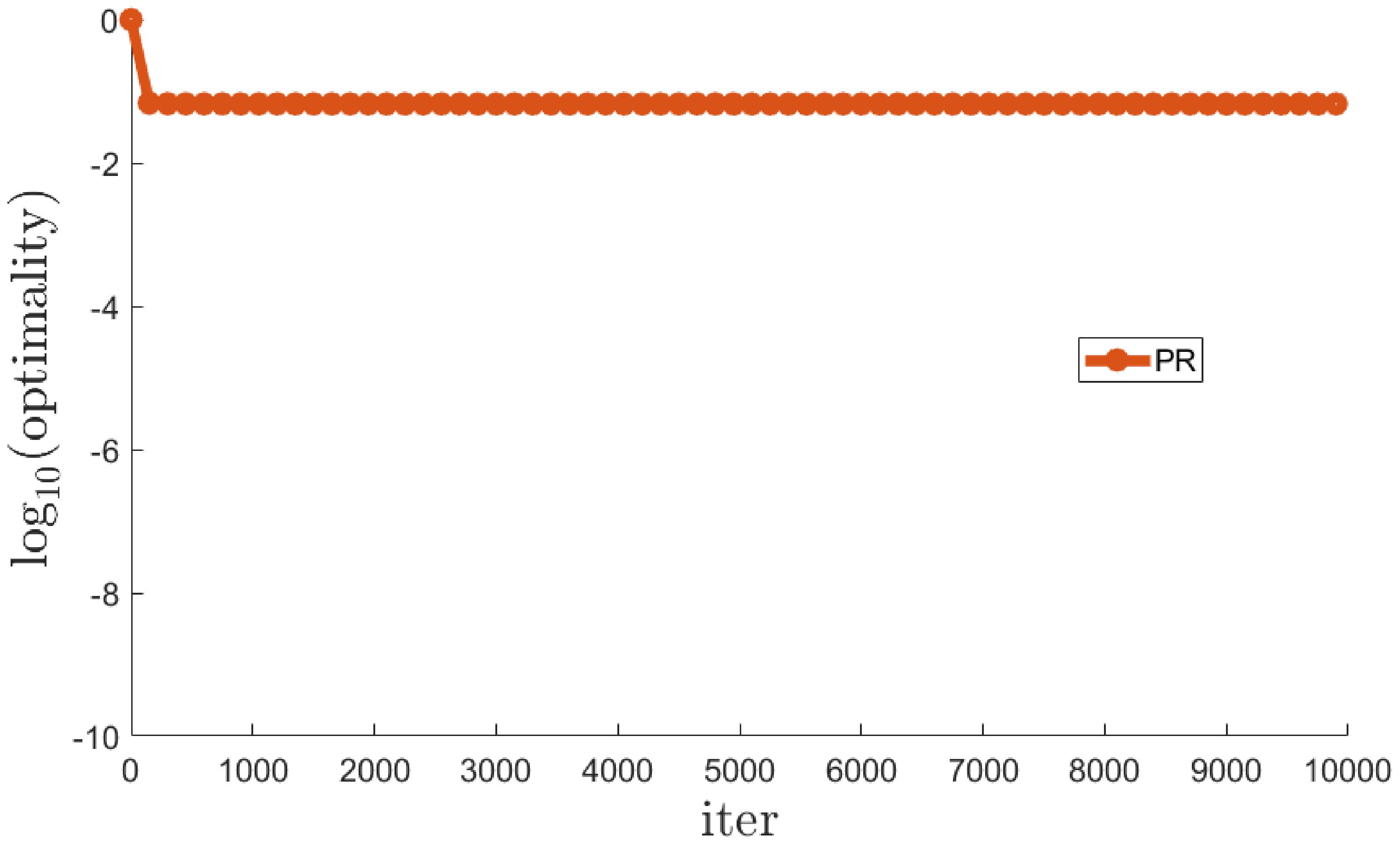
where t_i is the time in seconds for i–th instance

Effect of Restart Strategy for HPR-LP

nug08-3rd instance

(with Gourbi's presolve):

- 20,400 variables
- 19,700 constraints
- 139,000,000 non-zeros

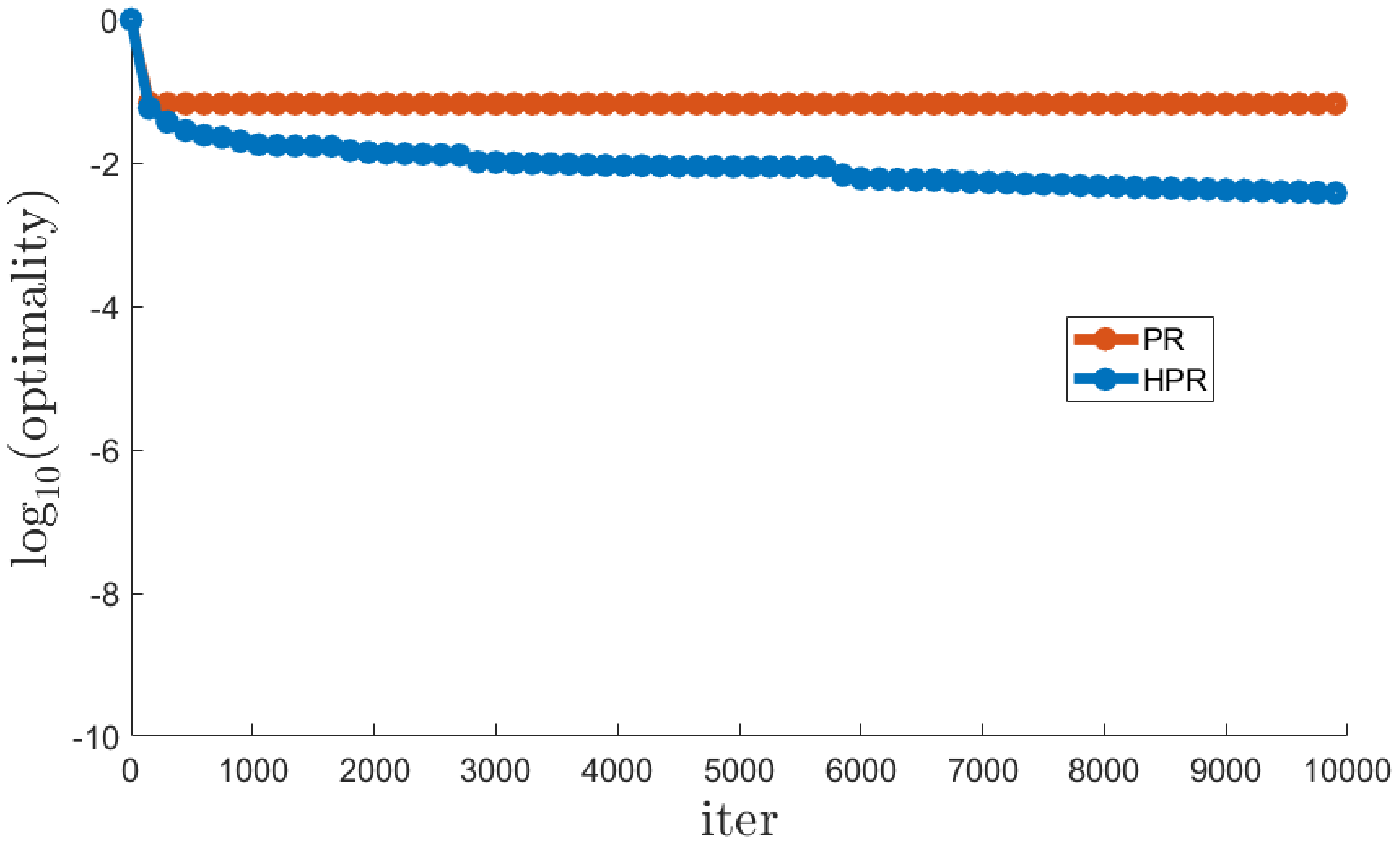


Effect of Restart Strategy for HPR-LP

nug08-3rd instance

(with Gourbi's presolve):

- 20,400 variables
- 19,700 constraints
- 139,000,000 non-zeros

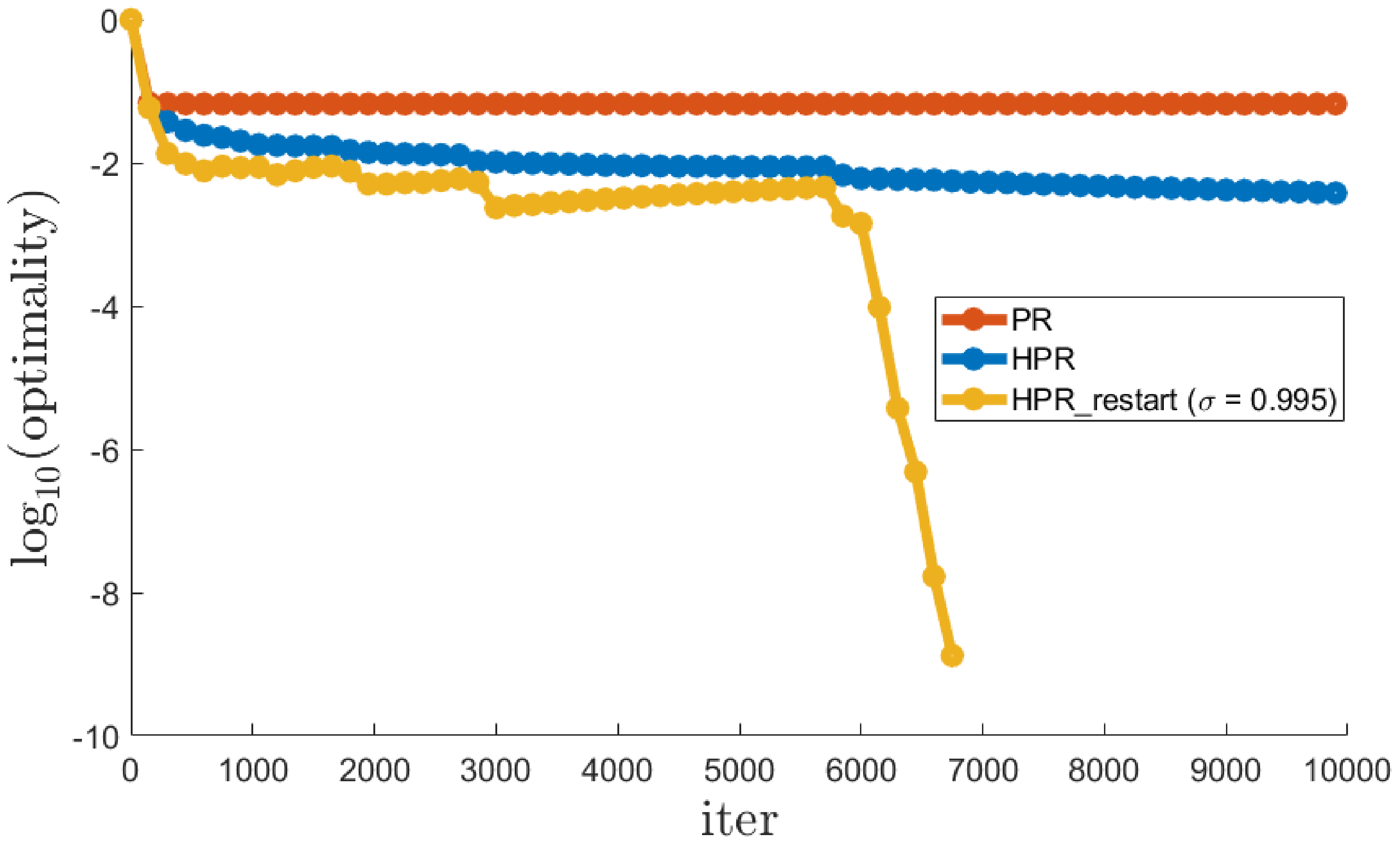


Effect of Restart Strategy for HPR-LP

nug08-3rd instance

(with Gourbi's presolve):

- 20,400 variables
- 19,700 constraints
- 139,000,000 non-zeros

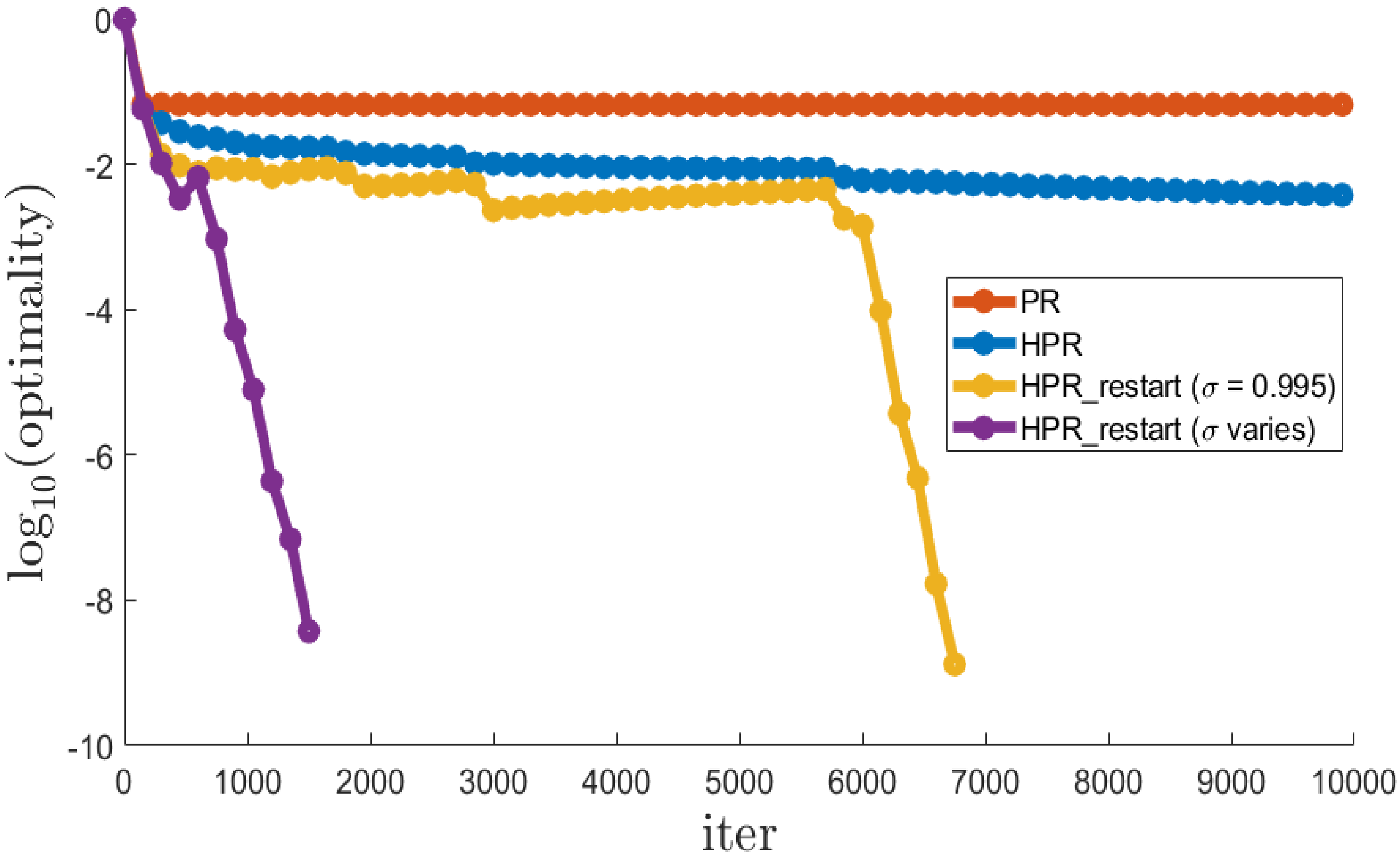


Effect of Restart Strategy for HPR-LP

nug08-3rd instance

(with Gourbi's presolve):

- 20,400 variables
- 19,700 constraints
- 139,000,000 non-zeros



Mittelmann's LP benchmark with presolve

Table 2: Numerical performance on 49 instances with Gurobi's presolve

Tolerance	10^{-4}		10^{-6}		10^{-8}	
Solvers	SGM10	Solved	SGM10	Solved	SGM10	Solved
cuPDLP.jl	60.0	46	118.6	45	220.6	43
HPR-LP.jl	17.4	49	31.8	49	59.4	48

- HPR-LP.jl solves 3-5 more problems than cuPDLP.jl does;
- Speed up in SGM10: 3.45x (10^{-4}), 3.73x (10^{-6}), and 3.71x (10^{-8}).

Mittelmann's LP benchmark with presolve

Table 3： Numerical performance on 49 instances without presolve

Tolerance	10^{-4}		10^{-6}		10^{-8}	
Solvers	SGM10	solved	SGM10	solved	SGM10	solved
cuPDLP.jl	76.9	42	156.2	41	277.9	40
HPR-LP.jl	30.2	47	69.1	44	103.8	43

- HPR-LP.jl solves 3-5 more problems than cuPDLP.jl does；
- Speed up in SGM10: 2.55x (10^{-4}), 2.26x (10^{-6}), and 2.68x (10^{-8}).

MIP relaxations with presolve

Table 4: Numerical performance on 380 instances with presolve

Tolerance	10^{-4}		10^{-6}		10^{-8}	
Solvers	SGM10	Solved	SGM10	Solved	SGM10	Solved
cuPDLP.jl	9.6	373	18.6	370	28.4	363
HPR-LP.jl	5.1	373	8.3	370	11.9	370

- For 10^{-8} , HPR-LP.jl solves 7 more problems than cuPDLP.jl does;
- Speed up in SGM10: 1.88x (10^{-4}), 2.24x (10^{-6}), and 2.39x (10^{-8}).

MIP relaxations without presolve

Table 5: Numerical performance on 380 instances without presolve

Tolerance	10^{-4}		10^{-6}		10^{-8}	
Solver	SGM10	Solved	SGM10	Solved	SGM10	Solved
cuPDLP.jl	14.3	372	25.0	366	36.3	359
HPR-LP.jl	6.9	376	11.6	371	17.9	363

- For 10^{-8} , HPR-LP.jl solves 4 more problems than cuPDLP.jl does;
- Speed up in SGM10: 2.07x (10^{-4}), 2.16x (10^{-6}), and 2.03x (10^{-8}).

QAP problem

Table 6: SGM10 on 20 QAP instances with presolve

Tolerance	10^{-4}		10^{-6}		10^{-8}	
Solver	HPR-LP.jl	cuPDLP.jl	HPR-LP.jl	cuPDLP.jl	HPR-LP.jl	cuPDLP.jl
SGM10	2.9	12.7	8.8	60.0	60.2	343.1

- Speed up in SGM10: 4.38x (10^{-4}), 6.82x (10^{-6}), and 5.70x (10^{-8}).

Table 7: SGM10 on 20 QAP instances without presolve

Tolerance	10^{-4}		10^{-6}		10^{-8}	
Solver	HPR-LP.jl	cuPDLP.jl	HPR-LP.jl	cuPDLP.jl	HPR-LP.jl	cuPDLP.jl
SGM10	18.9	43.9	150.7	342.4	1246.4	3202.5

- Speed up in SGM10: 2.32x (10^{-4}), 2.27x (10^{-6}), and 2.57x (10^{-8}).

ZIB problem (zib03)

Dimensions of A	Rows	Columns	Non-zeros
After presolve	19,701,908	29,069,187	104,300,584
Without presolve	19,731,970	29,128,799	104,422,573

✓ The LP solver COPT used **16.5** hours to solve this instance on an AMD Ryzen 9 5900X[20]

Table 8: Solving time in seconds for the "zib03"

Tolerance	10^{-4}		10^{-6}		10^{-8}		
Solver	HPR-LP.jl	cuPDLP.jl	HPR-LP.jl	cuPDLP.jl	HPR-LP.jl	cuPDLP.jl	Speed up in SGM10:
With presolve	273.8	351.9	1317.2	1634.6	3685.8	16462.2	← 4.47x
Without presolve	154.2	237.7	1063.6	1963.9	4865.3	19746.4	← 4.06x

[20] Lu, Yang, Hu, Huangfu, Liu, Liu, Ye, Zhang, Ge. arXiv preprint arXiv:2312.14832 (2023).

The numerical results of Gurobi

Table 9: Numerical results of Gurobi (32 threads) on several Mittelmann's LP benchmark instances with presolve (Tolerance: 10^{-8}, Time-limit: 15000s)

Instance	Rows	Columns	Non-zeros	HPR-LP.jl	Barrier	Primal Simplex	Dual Simplex
square41	1,754	23,828	4,336,554	93.09	1.55	2.89	2.22
physiciansched3-3	68,149	16,275	415,175	77.53	5.52	40.57	34.98
degma	185,501	659,415	8,127,528	49.32	60.15	15000	15000
dlr2	2,565,754	2,441,978	8,904,144	226.1	570.79	15000	15000
thk_48	4,229,611	6,339,391	22,790,807	362.62	9366.02	15000	15000
Dual2_5000	30,000,600	33,050,602	93,001,800	61.49	out of memory	15000	7291.77

Comparison with Ergodic PR (EPR)[21]

Alg5. An EPR for LP

Input: Choose $\mathcal{T}_1 (\succcurlyeq 0)$ such that $\mathcal{T}_1 + AA^* \succ 0$ and $w^0 = (y^0, z^0, x^0) \in D \times \mathbb{R}^n \times \mathbb{R}^n$.
Let $\sigma > 0$.

$k = 0,1, \dots,$

1. $\overline{w}^k = \mathbf{UpdateStep}\big(w^k, \mathcal{T}_1, \sigma\big)$:

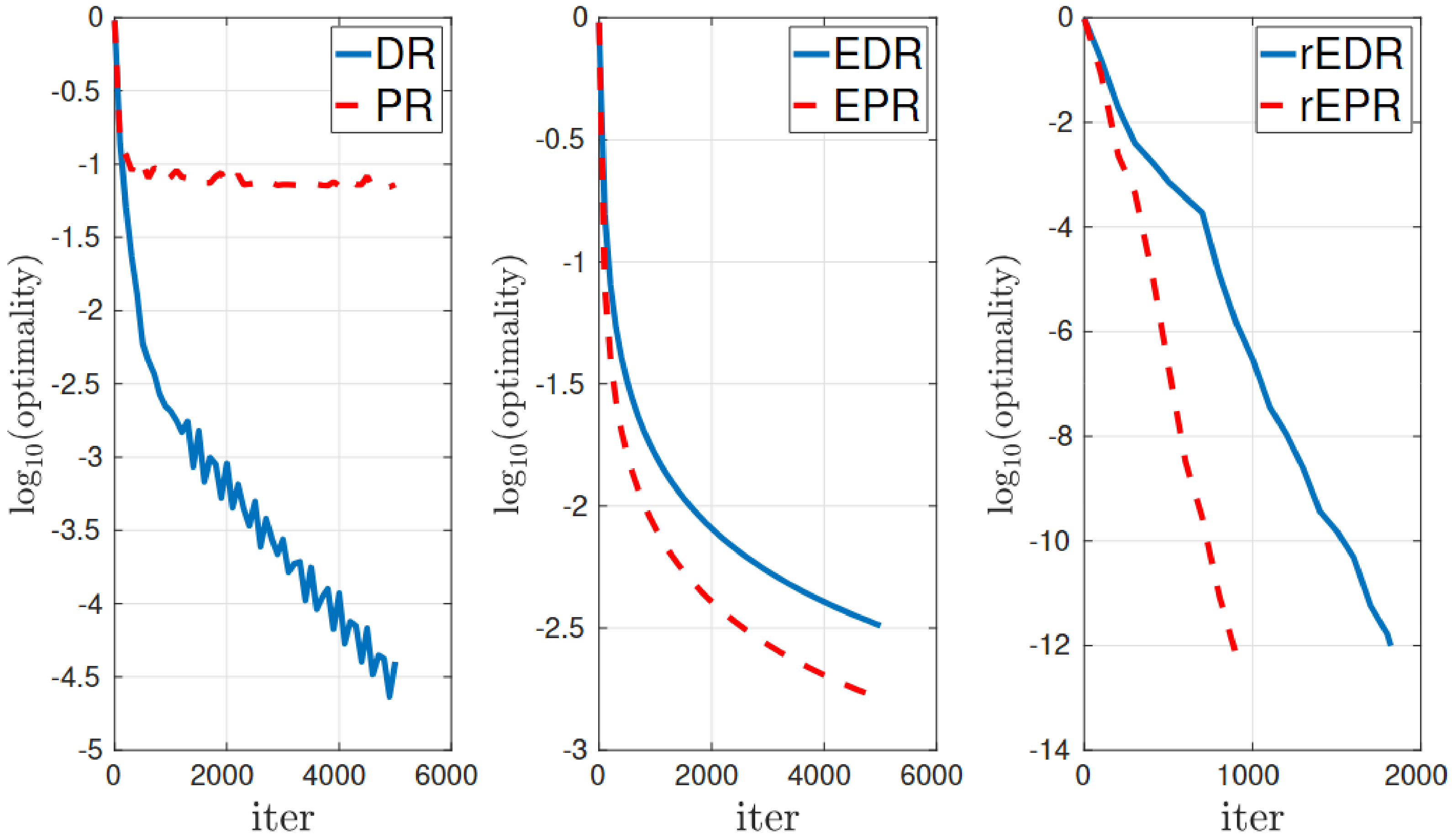
$$\begin{cases} \bar{z}^k = \mathrm{argmin}_{z \in \mathbb{R}^n} \left\{ L_\sigma^{LP}\left(y^k, z; x^k\right) + \frac{\sigma}{2} \left\| z - z^k \right\|_{\mathcal{T}_2}^2 \right\} \\ \bar{x}^k = x^k + \sigma\left(A^* y^k + \bar{z}^k - c\right) \\ \bar{y}^k = \mathrm{argmin}_{y \in \mathbb{R}^n} \left\{ L_\sigma^{LP}\left(y, \bar{z}^k; \bar{x}^k\right) + \frac{\sigma}{2} \left\| y - y^k \right\|_{\mathcal{T}_1}^2 \right\} \end{cases}$$

2. $w^{k+1} = 2\overline{w}^k - w^k$ $\quad [w^{k+1} = \overline{w}^k$: Douglas–Rachford (DR) $(\rho = 1)]$

Output: $\overline{w}_a{}^k = \frac{1}{k+1} \sum_{t=0}^k \overline{w}^t$

[21] Chen, Sun, Yuan, Zhang, and Zhao. "Peaceman-Rachford Splitting Method Converges Ergodically for Solving Convex Optimization Problems." *arXiv preprint arXiv:2501.07807* (2025).

Performance of EPR and restarted EPR (rEPR)



Performance comparison of algorithms (σ = 1) on the "ex10" instance from Mittelmann's LP benchmark

Comparison of cuPDLP, rEPR and HPR-LP

Table 10: Performance comparison of ADMM-type LP solvers on 49 instances of Mittelmann's LP benchmark without presolve (Tolerance: 10^{-8}, Time-limit: 3600s)

Algorithm	SGM10	Scaled SGM10	Solved
cuPDLP.jl	205.49	2.48	38
rEPR.jl	94.04	1.14	42
HPR-LP.jl	82.89	1.00	42

5. Conclusion

- We proposed an accelerated dPPM with $O(1/k)$ iteration complexity using the Halpern iteration
- Based on the equivalence between pADMM and dPPM, we derived an accelerated pADMM with $O(1/k)$ iteration complexity for the KKT residual and objective error
- We introduced HPR-LP: an implementation of an HPR method for solving LP
- In SGM10, HPR-LP.jl achieved a 2.39x to 5.70x speedup with presolve (2.03x to 4.06x without presolve) over cuPDLP.jl at 10^{-8} tolerance

References

- Kaihuang Chen, Defeng S., Yancheng Yuan, Guojun Zhang, and Xinyuan Zhao. "HPR-LP: An implementation of an HPR method for solving linear programming." arXiv:2408.12179 (August 2024; Revised March 2025).
- O'donoghue, Brendan, Eric Chu, Neal Parikh, and Stephen Boyd. "Conic optimization via operator splitting and homogeneous self-dual embedding." Journal of Optimization Theory and Applications 169 (2016): 1042-1068.
- Lin, Tianyi, Shiqian Ma, Yinyu Ye, and Shuzhong Zhang. "An ADMM-based interior-point method for large-scale linear programming." *Optimization Methods and Software* 36, no. 2-3 (2021): 389-424.
- Deng, Qi, Qing Feng, Wenzhi Gao, Dongdong Ge, Bo Jiang, Yuntian Jiang, Jingsong Liu et al. "An enhanced alternating direction method of multipliers-based interior point method for linear and conic optimization." *INFORMS Journal on Computing* (2024).

References

- Basu, Kinjal, Amol Ghoting, Rahul Mazumder, and Yao Pan. "ECLIPSE: An extreme-scale linear program solver for web-applications." In International Conference on Machine Learning, pp. 704-714. PMLR, 2020.

- Applegate, David, et al. "Practical large-scale linear programming using primal-dual hybrid gradient." Advances in Neural Information Processing Systems 34 (2021): 20243-20257.

- Sun, Defeng, Yancheng Yuan, Guojun Zhang, and Xinyuan Zhao. “Accelerating preconditioned ADMM via degenerate proximal point mappings.” SIAM Journal on Optimization 35:2 (2025) 1165–1193.

- Xiao, Yunhai, Liang Chen, and Donghui Li. "A generalized alternating direction method of multipliers with semi-proximal terms for convex composite conic programming." Mathematical Programming Computation 10 (2018): 533-555.

References

- Glowinski, Roland, and Americo Marroco. "Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires." Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique 9, no. R2 (1975): 41-76.
- Gabay, Daniel, and Bertrand Mercier. "A dual algorithm for the solution of nonlinear variational problems via finite element approximation." Computers & mathematics with applications 2, no. 1 (1976): 17-40.
- Eckstein, Jonathan, and Dimitri P. Bertsekas. "On the Douglas—Rachford splitting method and the proximal point algorithm for maximal monotone operators." Mathematical programming 55 (1992): 293-318.
- Fazel, Maryam, Ting Kei Pong, Defeng Sun, and Paul Tseng. "Hankel matrix rank minimization with applications to system identification and realization." SIAM Journal on Matrix Analysis and Applications 34, no. 3 (2013): 946-977.

References

- Monteiro, Renato DC, and Benar F. Svaiter. "Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers." SIAM Journal on Optimization 23, no. 1 (2013): 475-507.
- Davis, Damek, and Wotao Yin. "Convergence rate analysis of several splitting schemes." Splitting methods in communication, imaging, science, and engineering (2016): 115-163.
- Cui, Ying, Xudong Li, Defeng Sun, and Kim-Chuan Toh. "On the convergence properties of a majorized alternating direction method of multipliers for linearly constrained convex optimization problems with coupled objective functions." Journal of Optimization Theory and Applications 169 (2016): 1013-1041.
- Chambolle, Antonin, and Thomas Pock. "On the ergodic convergence rates of a first-order primal–dual algorithm." Mathematical Programming 159, (2016): 253-287.
- Chambolle, Antonin, and Thomas Pock. “A first-order primal-dual algorithm for convex problems with applications to imaging.” Journal of mathematical imaging and vision 40 (2011): 120-145.

References

- Bredies, Kristian, Enis Chenchene, Dirk A. Lorenz, and Emanuele Naldi. "Degenerate preconditioned proximal point algorithms." SIAM Journal on Optimization 32, no. 3 (2022): 2376-2401.

- Halpern, Benjamin. "Fixed points of nonexpanding maps." (1967): 957-961.

- Lieder, Felix. "On the convergence rate of the Halpern-iteration." Optimization letters 15, no. 2 (2021): 405-418.
- Brézis, Haim, and Pierre Louis Lions. "Produits infinis de résolvantes." Israel Journal of Mathematics 29 (1978): 329-345.

- Zhang, Guojun, Zhexuan Gu, Yancheng Yuan, and Defeng Sun. "HOT: An efficient Halpern accelerating algorithm for optimal transport problems." arXiv preprint arXiv:2408.00598 (2024). IEEE Transactions on Pattern Analysis and Machine Intelligence (2025), in print.

References

- Lu, Haihao, and Jinwen Yang. "cuPDLP. jl: A GPU implementation of restarted primal-dual hybrid gradient for linear programming in Julia." arXiv preprint arXiv:2311.12180 (2023).
- Applegate, David, Mateo Díaz, Oliver Hinder, Haihao Lu, Miles Lubin, Brendan O'Donoghue, and Warren Schudy. "Practical large-scale linear programming using primal-dual hybrid gradient." Advances in Neural Information Processing Systems 34 (2021): 20243-20257.
- Lu, Haihao, Jinwen Yang, Haodong Hu, Qi Huangfu, Jinsong Liu, Tianhao Liu, Yinyu Ye, Chuwen Zhang, and Dongdong Ge. "cuPDLP-C: A strengthened implementation of cuPDLP for linear programming by C language." arXiv preprint arXiv:2312.14832 (2023).
- Chen, Kaihuang, Defeng Sun, Yancheng Yuan, Guojun Zhang, and Xinyuan Zhao. "Peaceman-Rachford Splitting Method Converges Ergodically for Solving Convex Optimization Problems." arXiv preprint arXiv:2501.07807 (2025).

THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

Thank you for your attention!

Q & A

DEPARTMENT OF APPLIED MATHEMATICS
應 用 數 學 系

Opening Minds • Shaping the Future
啟迪思維 • 成就未來