

求解大规模线性规划问题的 HPR方法¹

孙德锋 讲座教授

应用数学系
香港理工大学



¹Kaihuang 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).



目录

1. 线性规划 (LP) 简介
2. 凸优化：加速的预条件ADMM算法 (pADMM)
3. 线性规划：Halpern Peaceman-Rachford算法
4. 数值结果
5. 总结和展望



1. 线性规划模型及其对偶模型

(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)

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

其中: $A_1 \in \mathbb{R}^{m_1 \times n}, A_2 \in \mathbb{R}^{m_2 \times n}, b_1 \in \mathbb{R}^{m_1}, b_2 \in \mathbb{R}^{m_2}, c \in \mathbb{R}^n$
 $C := \{x \in \mathbb{R}^n \mid l \leq x \leq u\}, l \in (\mathbb{R} \cup \{-\infty\})^n, u \in (\mathbb{R} \cup \{+\infty\})^n$
 $A := [A_1; A_2] \in \mathbb{R}^{m \times n}, m = m_1 + m_2, b := [b_1; b_2] \in \mathbb{R}^m$
 $D := \{y = (y_1, y_2) \in \mathbb{R}^{m_1} \times \mathbb{R}_+^{m_2}\}$

线性规划问题的应用



生产计划

通过合理安排生产计划，优化资源配置，提高生产效率



数据中心资源分配

合理分配有限资源，满足各种需求，提高资源利用效率



物流运输

优化运输路线和运输方式，降低运输成本



金融投资

在风险和受益的约束下，制定最优投资组合方案

Q：线性规划问题是否已经得到解决？

LP算法及求解器现状

01 LP算法:

- ✓ 基变换算法: 单纯形方法 (George Dantzig, 1947)
- ✓ 椭球算法 (Khachiyan, 1979)
- ✓ 内点法: Karmarkar 算法 (1984)
- ✓ 分裂算法: 将大规模问题分解为若干小规模问题, 分别求解再整合

02 LP求解器:

- ✓ 商业求解器: Gurobi, Cplex (IBM), Mosek, COPT (杉树), MindOpt (阿里云), OptVerse (华为天筹)
- ✓ 开源求解器: SCIP (ZIB), GLOP (Google), JuMP, CLP (COIN-OR)

相关研究进展

- 虽然上个世纪五十年代开始就用一阶方法求解线性规划的，但是在数值计算方面并没能很好的解决这一问题。
- 最近，由于大规模线性规划问题的涌现，这一问题又开始成为研究热点：

01

SCS

算子分裂/ADMM算法 [O' Donoghue, Chu, Parikh, Boyd, 2016]

02

ABIP

基于ADMM的内点方法 [Lin et al., 2021], [Deng et al., 2024]

03

SNIPAL

半光滑牛顿-增广拉格朗日法 [Li, Sun, Toh, 2020]

04

ECLIPSE

Nesterov's 加速梯度法 [Basu, Ghoting, Mazumder, Pan, 2020]

05

PDLP

原始-对偶混合梯度法 [Applegate et al., 2021]

大规模LP问题的挑战

- **计算复杂度**: LP问题规模越来越大, 求解时间过长, 甚至无法接受
- **存储限制**: 大规模LP问题需要大量内存来存储模型数据、参数和中间结果 (嵌入式系统或者移动设备内存有限)
- **求解效率**: 优化算法、并行计算与分布式技术还存在挑战



需要高效的优化算法, 克服内存限制的瓶颈, 降低计算复杂度, 提高求解效率

2. 凸优化问题：加速的预条件ADMM算法² (pADMM)

-- Preconditioned (Semi-Proximal) ADMM

$$\text{凸优化问题(COP)} \quad \Longrightarrow \quad \begin{array}{ll} \min_{y \in \mathbb{Y}, z \in \mathbb{Z}} & f_1(y) + f_2(z) \\ \text{s. t.} & B_1 y + B_2 z = c \end{array}$$

$$\text{对偶问题(COD)} \quad \Longrightarrow \quad \max_{x \in \mathbb{X}} \{-f_1^*(-B_1^* x) - f_2^*(-B_2^* x) - \langle c, x \rangle\}$$

其中： $\mathbb{X}, \mathbb{Y}, \mathbb{Z}$: 有限维实欧式空间

$f_1 : \mathbb{Y} \rightarrow (-\infty, +\infty]$, $f_2 : \mathbb{Z} \rightarrow (-\infty, +\infty]$: 正常闭凸函数

$B_1 : \mathbb{Y} \rightarrow \mathbb{X}$, $B_2 : \mathbb{Z} \rightarrow \mathbb{X}$: 线性算子, $c \in \mathbb{X}$

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

凸优化方法

- 增广拉格朗日函数: $\forall (y, z, x) \in \mathbb{Y} \times \mathbb{Z} \times \mathbb{X}$,

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

- 两个算子 Σ_{f_1} 和 Σ_{f_2} 满足:

- 对于任意 $y, \hat{y} \in \text{dom}(f_1)$, $\phi \in \partial f_1(y)$ 和 $\hat{\phi} \in \partial f_1(\hat{y})$

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

- 对于任意 $z, \hat{z} \in \text{dom}(f_2)$, $\varphi \in \partial f_2(z)$ 和 $\hat{\varphi} \in \partial f_2(\hat{z})$

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

算法1. 求解凸优化问题的pADMM³算法

输入：选择 $\mathcal{T}_1 (\geq 0), \mathcal{T}_2 (\geq 0)$ 满足

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

选择 $w^0 = (y^0, z^0, x^0)$. 令 $\sigma > 0, \rho \in (0, 2]$

$k = 0, 1, \dots,$

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

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

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

- 相关文献: ADMM^{4, 5}, 广义ADMM⁶, 半邻近ADMM⁷
- $\rho = 2$: 半邻近Peaceman-Rachford (PR)算法

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

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

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

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

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

pADMM算法求解LP

□ LP的增广拉格朗日函数:

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

算法2. pADMM-LP算法

输入: 选择 $\mathcal{T}_1 (\geq 0)$ 满足 $\mathcal{T}_1 + AA^* > 0$, $w^0 = (y^0, z^0, x^0) \in D \times \mathbb{R}^n \times \mathbb{R}^n$
 令 $\sigma > 0, \rho \in (0, 2]$

$k = 0, 1, \dots,$

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

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

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

表1 pADMM型算法复杂性现有结果

文献	M&S(2013) ⁸	D&Y(2016) ⁹	Cui(2016) ¹⁰
算法	ADMM	ADMM	maj. ADMM
Dual step	1	1	$(0, (1 + \sqrt{5})/2)$
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 -次微分残差	-	$O(1/\sqrt{k})$
Type	遍历的	非遍历的	非遍历的

Chambolle和Pock^{11,12}: PDHG (线性化 ADMM) 在某种原始对偶间隙意义下享有 $O(1/k)$ 遍历复杂性 (可由原始可行性和目标误差推导)。然而, 这可能无法推出 KKT 残差达到期望的 $O(1/k)$ 。

⁸Monteiro and Svaiter. SIAM J. Optim. (2013): 475-507. (First version: August 2010).

⁹Davis and Yin. Splitting methods in communication, imaging, science, and engineering (2016): 115-163.

¹⁰Cui, Li, Sun, and Toh. J. Optim. Theory Appl. (2016): 1013-1041.

¹¹Chambolle and Pock. J. Math. Imaging Vis. (2011): 120-145. (Online: May 2010).

¹²Chambolle and Pock. Math. Program. (2016): 253-287.

Q: pADMM的KKT残差至多为 $O(1/\sqrt{k})$ 。如何加速pADMM使其达到 $O(1/k)$ 或更好?

算法思维导图

半邻近交替方向乘子法

preconditioned (semi-proximal) ADMM (pADMM)

退化的邻近点法

degenerate proximal point method (dPPM)

等价



加速的半邻近交替方向乘子法

accelerated pADMM



等价

**Halpern
迭代法**



加速的退化邻近点算法

accelerated dPMM

Karush-Kuhn-Tucker (KKT)系统及其等价问题

01

(COP)问题的KKT系统

$$\begin{aligned} -B_1^*x^* &\in \partial f_1(y^*) \\ -B_2^*x^* &\in \partial f_2(z^*) \\ B_1y^* + B_2z^* - c &= 0 \end{aligned}$$

02

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

$$0 \in \mathcal{J}w = \begin{pmatrix} \partial f_1(y) + B_1^*x \\ \partial f_2(z) + B_2^*x \\ c - B_1y - B_2z \end{pmatrix} (\star)$$

假设 2.1

(KKT) 系统存在非空解集

退化邻近点算法

定义 2.1 [Bredies et al. (2022)¹³]

算子 $\mathcal{T}: \mathbb{W} \rightarrow 2^{\mathbb{W}}$ 的**可接受的预条件算子** $M: \mathbb{W} \rightarrow \mathbb{W}$ 是线性、有界、自共轭、**半正定**的，满足

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

是单值的，且定义域为全空间。

退化邻近点算法 (degenerate PPM, dPPM):

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

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

其中 $w^0 \in \mathbb{W}, \rho \in (0, 2]$

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

pADMM和dPPM的等价性

线性算子 $M: \mathbb{W} \rightarrow \mathbb{W}$:

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

命题 2.1

若从相同初始点 $w^0 \in \mathbb{W}$ 出发, 则pADMM(算法1)和dPPM算法生成的序列 $\{w^k\}$ 是一致的。此外, M 是**可接受的预条件算子**, 且 $(M + \mathcal{T})^{-1}$ 是Lipschitz连续的。

加速dPPM算法

- 定义 $\hat{F}_\rho := (1 - \rho)I + \rho\hat{\mathcal{T}}$, 这里 $\rho \in (0, 2]$, 算子 $\hat{F}_\rho$ 是 M -非扩张的, 即

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

算法3. 加速dPPM算法求解问题(★)

输入: 令 $w^0 \in \mathbb{W}, \rho \in (0, 2]$

$k = 0, 1, \dots,$

1. $\bar{w}^{k+1} = \hat{\mathcal{T}}w^k$

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

3. $w^{k+1} = \frac{1}{k+2}w^0 + \frac{k+1}{k+2}\hat{w}^{k+1}$ Halpern迭代^{14,15}

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

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

加速dPPM算法收敛性和复杂性

定理 2.1

若假设2.1 成立, M 是**可接受预条件算子**且 $(M + \mathcal{T})^{-1}$ 是连续的, 则由算法3产生的序列 $\{\bar{w}^k\}$ 收敛至KKT系统解集中一点 w^* 。

命题 2.2

若假设2.1 成立, M 是**可接受预条件算子**, 则由算法3产生的序列 $\{w^k\}$, $\{\hat{w}^k\}$ 满足

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

注¹⁶: **不加速**的dPPM具有 $O\left(\frac{1}{\sqrt{k}}\right)$ 迭代复杂性相对于 $\|w^k - w^{k-1}\|_M$

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

3. Halpern-Peaceman-Rachford (HPR) 算法

算法4. 半邻近HPR方法求解LP

输入：选择 $\mathcal{T}_1 (\geq 0)$ 满足 $\mathcal{T}_1 + AA^* > 0$, $w^0 = (y^0, z^0, x^0) \in D \times \mathbb{R}^n \times \mathbb{R}^n$, $\sigma > 0$
 $k = 0, 1, \dots$,

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

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

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

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

定理 2.2 HPR方法求解LP问题复杂性

若假设2.1 成立, 令 $R_0 = \|w^0 - w^*\|_M$, 则由**算法4**产生的序列 $\{\bar{w}^k\}$ 满足

$$\begin{array}{l} \text{原始不可行性} \rightarrow \\ \text{互补松弛性} \rightarrow \\ \text{对偶不可行性} \rightarrow \end{array} \left\| \begin{pmatrix} \bar{y}^{k+1} - \Pi_D(\bar{y}^{k+1} - A\bar{x}^{k+1} + b) \\ \bar{x}^{k+1} - \Pi_C(\bar{x}^{k+1} - \bar{z}^{k+1}) \\ c - A^*\bar{y}^{k+1} - \bar{z}^{k+1} \end{pmatrix} \right\| \leq \left(\frac{\sigma(\|A\| + \|\sqrt{\mathcal{I}_1}\|) + 1}{\sqrt{\sigma}} \right) \frac{R_0}{k+1}$$

$$\begin{array}{l} \text{对偶目标} \\ \text{函数误差} \end{array} \rightarrow \left(\frac{-\|x^*\|}{\sqrt{\sigma}} \right) \frac{R_0}{k+1} \leq h^{LP}(\bar{y}^{k+1}, \bar{z}^{k+1}) \leq \left(3R_0 + \frac{\|x^*\|}{\sqrt{\sigma}} \right) \frac{R_0}{k+1}$$

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

算法5: HPR-LP算法求解线性规划问题(P)

1. 输入: 选择 $\mathcal{T}_1 (\geq 0)$ 满足 $\mathcal{T}_1 + AA^* \succ 0$, $w^{0,0} := (y^{0,0}, z^{0,0}, x^{0,0}) \in D \times \mathbb{R}^n \times \mathbb{R}^n$

2. 初始化: 参数 $\sigma_0 > 0$ 和 $r = 0$

3. 外循环:

3.1 内循环初始化 $t = 0$

3.2 内循环: $\bar{w}^{r,t+1} = \text{UpdateStep}(w^{r,t}, \mathcal{T}_1, \sigma_r)$

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

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

若满足**重启标准之一**或者**终止标准**, 输出: $w^{r+1,0} = \bar{w}^{r,t+1}$

否则, $t = t + 1$

4. 若满足**终止标准**, 输出: $w^{r+1,0}$; 否则,

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

5. $r = r + 1$



HPR-LP重启标准

命题 2.2

若假设2.1 成立, M 是**可接受预条件算子**, 则由算法3(或4)产生的序列 $\{w^k\}$, $\{\hat{w}^k\}$ 满足

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

基于KKT残差的 $O(1/k)$ 迭代复杂度, 定义评价函数:

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

用 $\tilde{R}_{r,t}$ 近似 $R_{r,t}$

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

HPR-LP重启标准

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

- 若 $\tilde{R}_{r,t}$ 充分下降:

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

- 若 $\tilde{R}_{r,t}$ 必要下降+ “不单调”

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

- 内循环迭代过多:

$$t \geq \alpha_3 k,$$

其中 $\alpha_1 \in (0, \alpha_2)$, $\alpha_2 \in (0, 1)$, $\alpha_3 \in (0, 1)$ 。

HPR-LP: 更新 σ

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

$$\begin{aligned} \sigma_{r+1} &= \arg \min_{\sigma} \|w^{r+1,0} - w^*\|_M^2 \\ &= \arg \min_{\sigma} \left(\sigma \|y^{r+1,0} - y^*\|_{\mathcal{T}_1}^2 + \sigma^{-1} \|x^{r+1,0} - x^* + \sigma A^*(y^{r+1,0} - y^*)\|^2 \right) \\ &= \sqrt{\frac{\|x^{r+1,0} - x^*\|^2}{\|y^{r+1,0} - y^*\|_{\mathcal{T}_1}^2 + \|A^*(y^{r+1,0} - y^*)\|^2}} \end{aligned}$$

在HPR-LP中如下近似更新 σ :

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

HPR-LP: 更新 σ -example

- 考虑特殊结构的LP, 如最优运输问题¹⁷, 令 $\mathcal{T}_1 = 0$

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

- 考虑一般的LP, 令 $\mathcal{T}_1 = \lambda I_m - AA^*$, $\lambda \geq \lambda_1(AA^*)$

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

- ¹⁷Zhang, Gu, Yuan, and Sun. "HOT: An Efficient Halpern Accelerating Algorithm for Optimal Transport Problems." IEEE Transactions on Pattern Analysis and Machine Intelligence (2025), in print.

4. 数值实验（机器配置）

- HPR-LP：用 Julia 语言实现的，称为 HPR-LP.jl；
- cuPDLP.jl¹⁸：荣获国际奖的PDLP¹⁹求解器 的 GPU 版本，用 Julia 语言实现；
- 所有实验均在 SuperServer SYS-420GP-TNR 上进行测试，
 - HPR-LP.jl 和 cuPDLP.jl：NVIDIA A100-SXM4-80GB GPU，CUDA 12.3。
 - Gurobi 11.0.3(教育版)：Intel Xeon Platinum8338C CPU 2.60GHz，内存 256GB。

¹⁸Lu and Yang. arXiv preprint arXiv:2311.12180 (2023).

¹⁹Applegate, D'iaz, Hinder, Lu, Lubin, O'Donoghue, and Schudy were awarded the **Beale–Orchard-Hays Prize** for Excellence in ISMP2024, July 21-26, 2024, Montr'éal, Canada.

终止准则和评价指标

➤ **终止准则 (Optimality):** HPR-LP (PDLP) 当满足下述条件时终止, $\varepsilon \in (0, \infty)$

- ❑ 相对对偶间隙 $|\langle b, y \rangle - \delta_C^*(-z) - \langle c, x \rangle| \leq \varepsilon(1 + |\langle b, y \rangle - \delta_C^*(-z) + \langle c, x \rangle|)$
- ❑ 相对原始不可行性 $\|\Pi_D(b - Ax)\| \leq \varepsilon(1 + \|b\|)$
- ❑ 相对对偶不可行性 $\|c - A^*y - z\| \leq \varepsilon(1 + \|c\|)$

➤ **SGM10评价指标:** 求解时间的移位几何均值

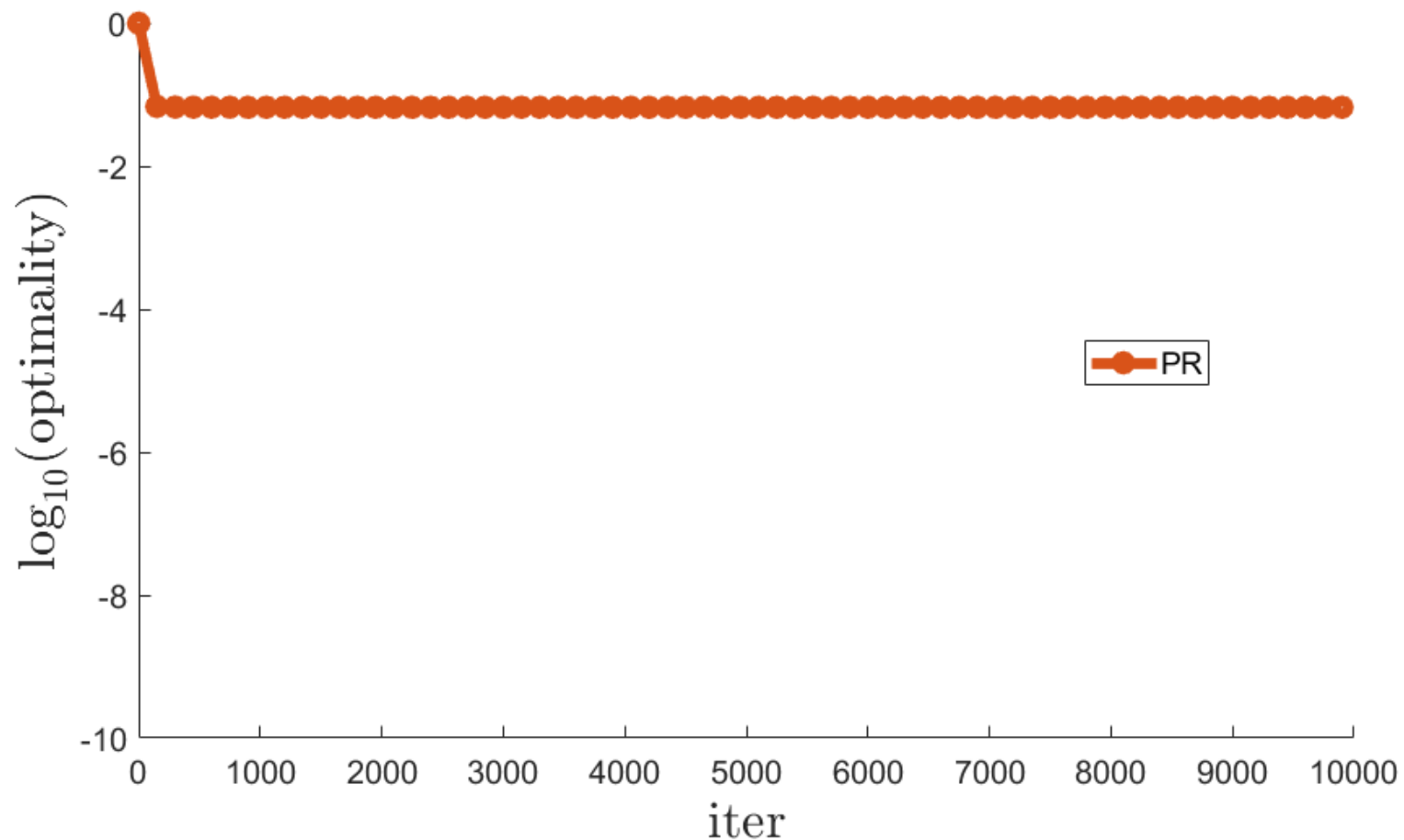
$$\left(\prod_{i=1}^n (t_i + 10) \right)^{1/n} - 10,$$

其中, t_i 是第 i 个实例的时间 (单位: 秒)

HPR重启策略效果

nug08-3rd问题 (预处理后) :

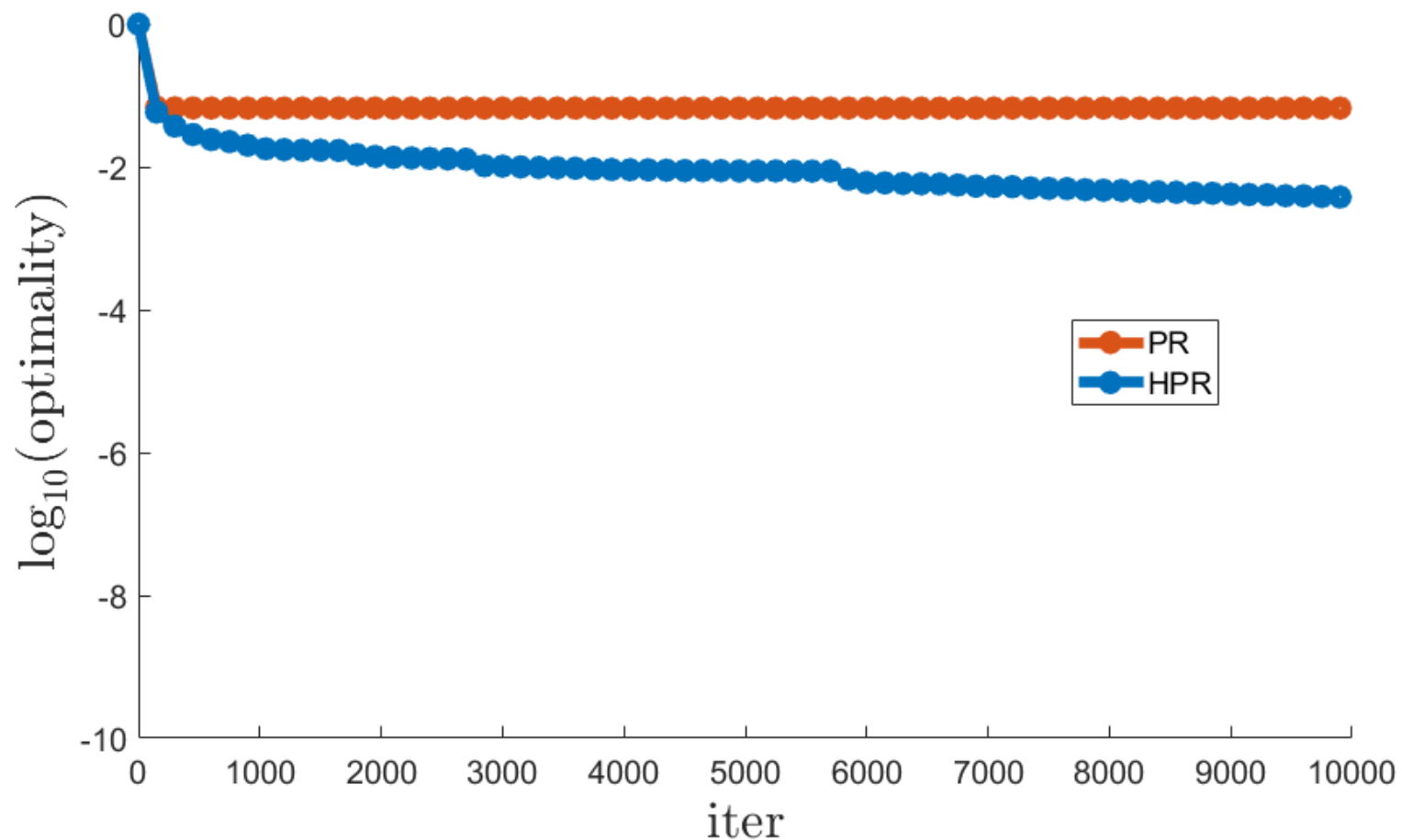
- 20,400变量
- 19,700约束
- 139,000,000非零元



HPR重启策略效果

nug08-3rd问题 (预处理后) :

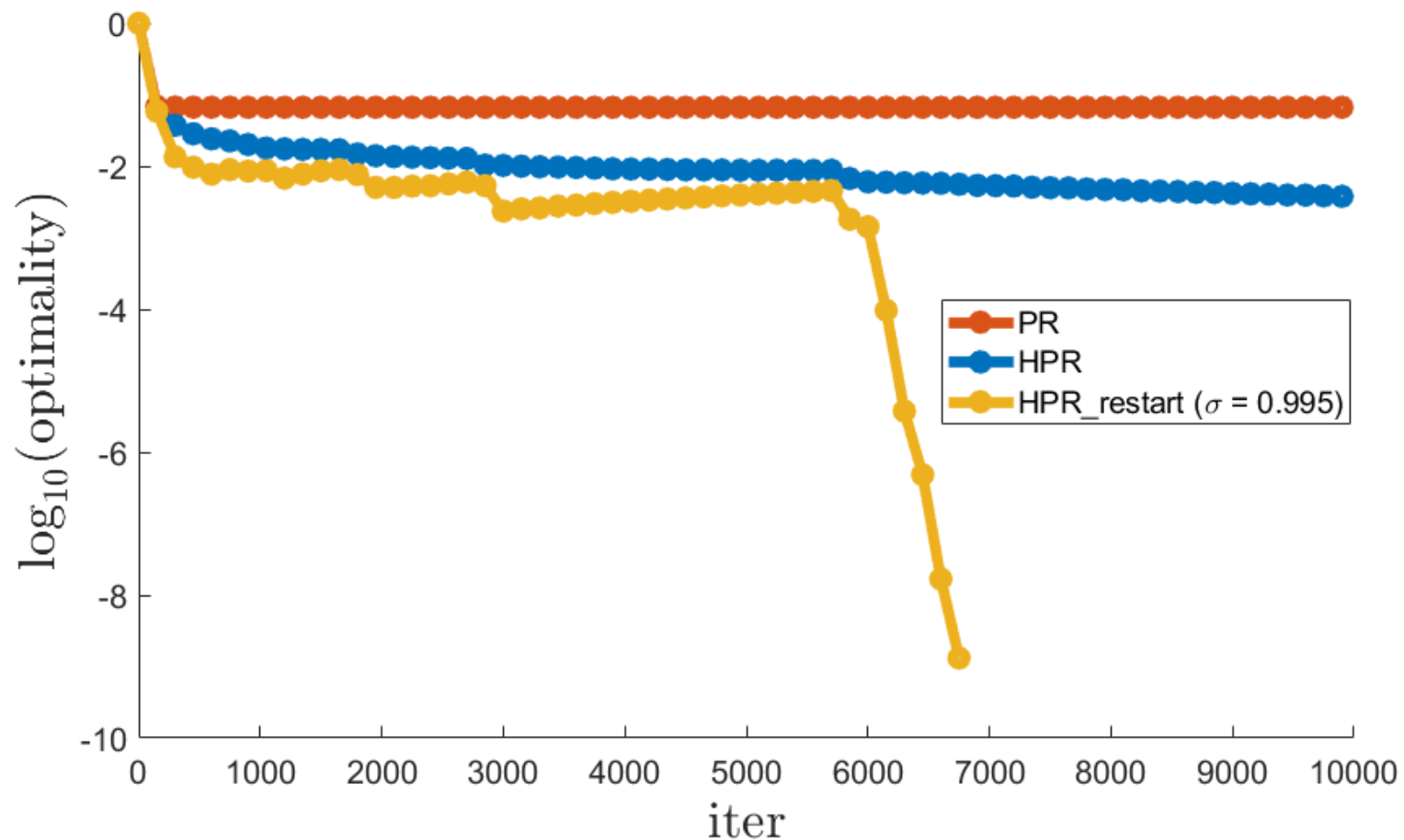
- 20,400变量
- 19,700约束
- 139,000,000非零元



HPR重启策略效果

nug08-3rd问题 (预处理后) :

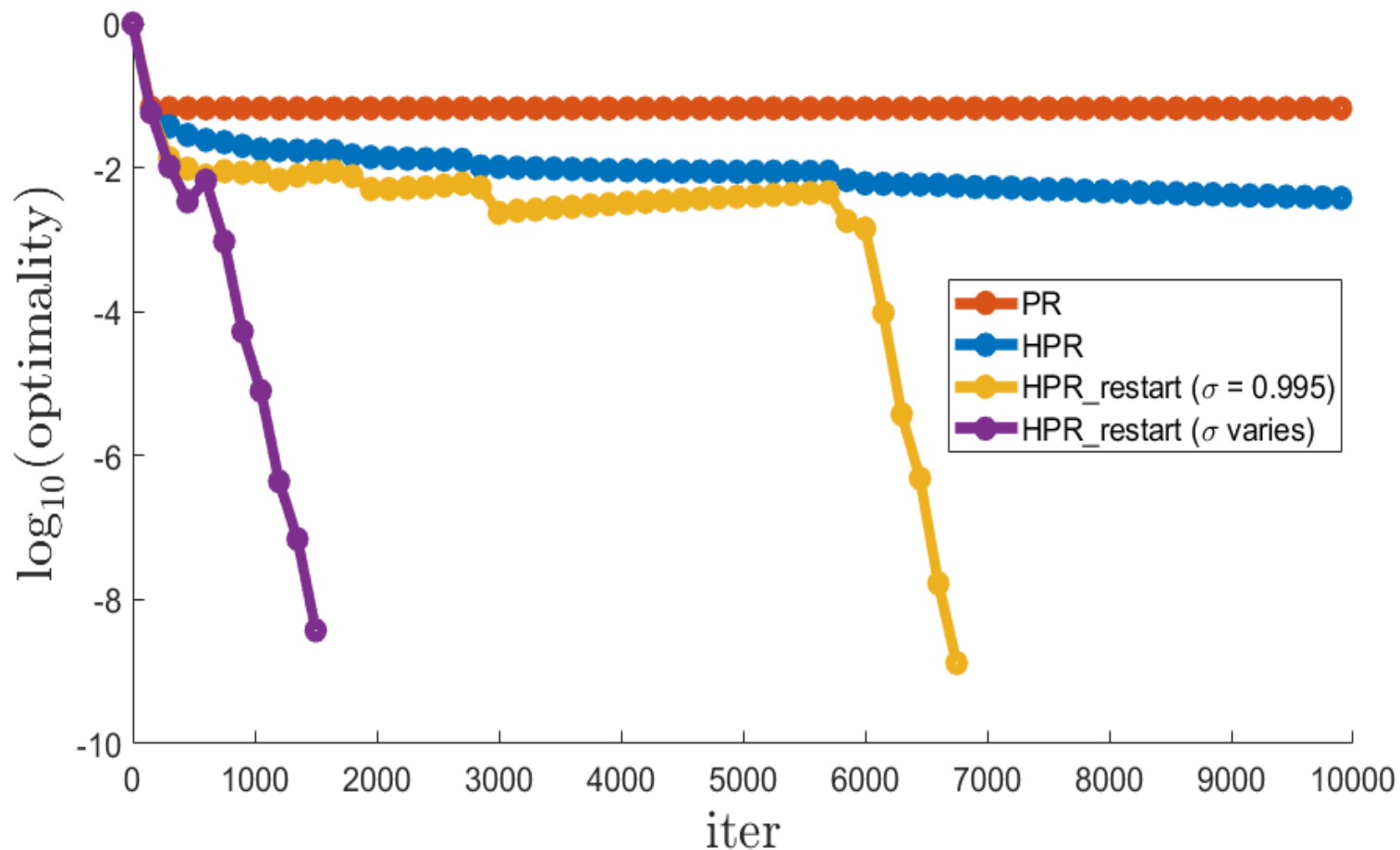
- 20,400变量
- 19,700约束
- 139,000,000非零元



HPR重启策略效果

nug08-3rd问题 (预处理后) :

- 20,400变量
- 19,700约束
- 139,000,000非零元



Mittelman LP基准测试集 (预处理)

表2: 49 个Mittelman LP基准测试集实例的数值性能表现 (使用 Gurobi 预处理)

精度	10^{-4}		10^{-6}		10^{-8}	
求解器	SGM10	可解	SGM10	可解	SGM10	可解
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 比 cuPDLP.jl 多解决了 3 - 5 个问题;
- SGM10加速比分别为: 3.45 倍 (10^{-4})、3.73 倍 (10^{-6})和 3.71 倍 (10^{-8})。

Mittelman LP基准测试集 (未预处理)

表3: 49 个Mittelman LP基准测试集实例的数值性能表现 (未预处理)

精度	10^{-4}		10^{-6}		10^{-8}	
求解器	SGM10	可解	SGM10	可解	SGM10	可解
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 比 cuPDLP.jl 多解决了 3 - 5 个问题;
- SGM10加速比分别为: 2.55 倍 (10^{-4})、2.26 倍 (10^{-6})和 2.68 倍 (10^{-8})。

MIP松弛 (预处理)

表4: 380个 MIP 松弛实例的数值性能表现 (预处理)

精度	10^{-4}		10^{-6}		10^{-8}	
求解器	SGM10	可解	SGM10	可解	SGM10	可解
cuPDLP.jl	9.6	373	18.6	370	28.4	363
HPR-LP.jl	5.1	373	8.3	370	11.9	370

- 在 10^{-8} 精度下, HPR-LP.jl 比 cuPDLP.jl 多解决了 7 个问题;
- SGM10加速比分别为: 1.88 倍 (10^{-4})、2.24 倍 (10^{-6})和 2.39 倍(10^{-8})。

MIP松弛 (未预处理)

表5: 380个 MIP 松弛实例的数值性能表现 (未预处理)

精度	10^{-4}		10^{-6}		10^{-8}	
求解器	SGM10	可解	SGM10	可解	SGM10	可解
cuPDLP.jl	14.3	372	25.0	366	36.3	359
HPR-LP.jl	6.9	376	11.6	371	17.9	363

- 在 10^{-8} 精度下, HPR-LP.jl 比 cuPDLP.jl 多解决了 4 个问题;
- SGM10加速比分别为: 2.07 倍 (10^{-4})、2.16 倍 (10^{-6})和 2.03倍 (10^{-8})。

QAP问题

表6: 20个 QAP 实例的数值性能表现 (预处理)

精度	10^{-4}		10^{-6}		10^{-8}	
求解器	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

- SGM10加速比分别为: 4.38 倍 (10^{-4})、6.82倍 (10^{-6})和 5.70 倍 (10^{-8})。

表7: 20个 QAP 实例的数值性能表现 (未预处理)

精度	10^{-4}		10^{-6}		10^{-8}	
求解器	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

- SGM10加速比分别为: 2.32 倍 (10^{-4})、2.27 倍 (10^{-6})和 2.57 倍 (10^{-8})。

ZIB问题 (zib03)

矩阵A的维数	行	列	非零元个数
预处理后	19,701,908	29,069,187	104,300,584
未预处理	19,731,970	29,128,799	104,422,573

✓ 商业线性规划求解器
COPT 在 AMD Ryzen 9
 5900X 上求解这个实例
 用了**16.5h**。²⁰

表8: “zib03” 实例的求解时间 (秒)

精度	10^{-4}		10^{-6}		10^{-8}	
	求解器		求解器		求解器	
	HPR-LP.jl	cuPDLP.jl	HPR-LP.jl	cuPDLP.jl	HPR-LP.jl	cuPDLP.jl
预处理	273.8	351.9	1317.2	1634.6	3685.8	16462.2
未预处理	154.2	237.7	1063.6	1963.9	4865.3	19746.4

SGM10加速比:
 ← 4.47 倍
 ← 4.06 倍

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

Gurobi数值结果

表9: 6个Mittelman LP基准测试实例的数值性能表现
(Gurobi、32线程、预处理、精度: 10^{-8} 、时间限制: 15K秒)

实例	行数	列数	非零元	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

与遍历PR (Ergodic PR, EPR) 对比²¹

算法5. 求解LP的EPR算法

输入: 选择 $\mathcal{T}_1 (\geq 0)$ 使得 $\mathcal{T}_1 + AA^* > 0$ 以及 $w^0 = (y^0, z^0, x^0) \in D \times \mathbb{R}^n \times \mathbb{R}^n$.
令 $\sigma > 0$.

$k = 0, 1, \dots,$

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

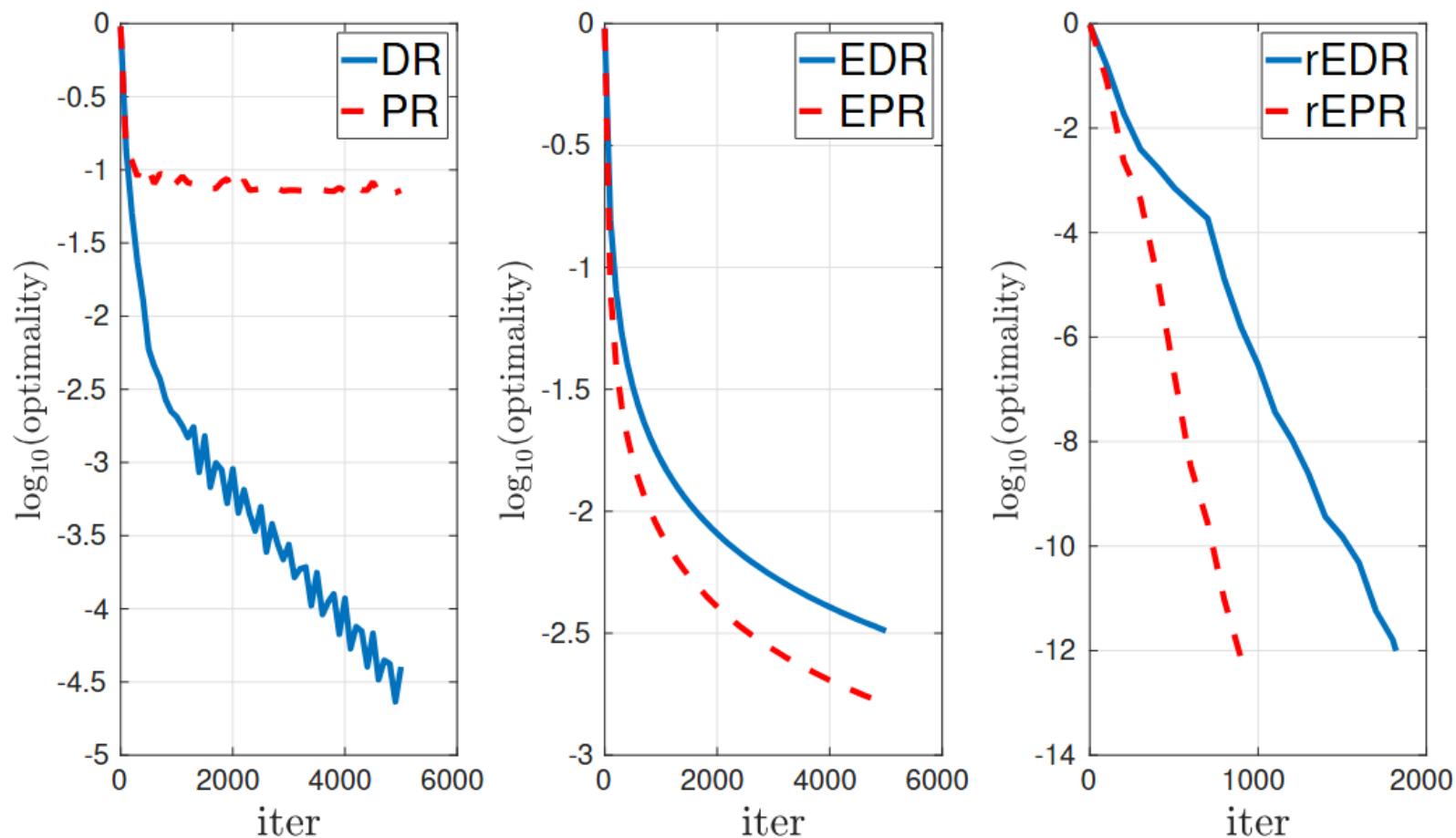
$$\begin{cases} \bar{z}^k = \operatorname{argmin}_{z \in \mathbb{R}^n} \left\{ L_{\sigma}^{LP}(y^k, z; x^k) + \frac{\sigma}{2} \|z - z^k\|_{\mathcal{T}_2}^2 \right\} \\ \bar{x}^k = x^k + \sigma(A^*y^k + \bar{z}^k - c) \\ \bar{y}^k = \operatorname{argmin}_{y \in \mathbb{R}^n} \left\{ L_{\sigma}^{LP}(y,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

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

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

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

EPR和重启EPR (restarted EPR, rEPR) 性能对比



不同算法 ($\sigma = 1$) 在 Mittelman LP 基准测试的 “ex10” 实例上性能比较

cuPDLP, rEPR和HPR-LP性能比较

表 10: ADMM 类型 LP 求解器在 49 个 Mittelmann LP 基准测试实例上的性能比较
(未预处理, 精度: 10^{-8} , 时间限制: 3600s)

算法	SGM10	Scaled SGM10	求解问题个数
cuPDLP.jl	205.49	2.48	38
rEPR.jl	94.04	1.14	42
HPR-LP.jl	82.89	1.00	42

5. 结论与展望

- 我们使用 Halpern 迭代提出了具有 $O(1/k)$ 迭代复杂度的加速 dPPM。
- 基于 pADMM 与 dPPM 之间的等价性，我们推导出了针对 KKT 残差和目标误差具有 $O(1/k)$ 迭代复杂度的加速 pADMM。
- 我们介绍了 HPR-LP：一种用于求解线性规划（LP）的 HPR 方法的实现。
- 在 SGM10 中，在 10^{-8} 的容差下，HPR-LP.jl 与 cuPDLP.jl 相比，在使用预处理的情况下实现了 2.39 倍到 5.70 倍的加速（在不使用预处理的情况下为 2.03 倍到 4.06 倍）。

参考文献

- 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).



参考文献

- 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.

参考文献

- 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.

参考文献

- 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.

参考文献

- 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." *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025), in print. arXiv:2408.00598 (August 2024).

参考文献

- 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).

感谢聆听

Q & A



DEPARTMENT OF APPLIED MATHEMATICS

應 用 數 學 系



THE HONG KONG
POLYTECHNIC UNIVERSITY
香港理工大學

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