Accelerating preconditioned ADMM via degenerate proximal point mappings

Defeng Sun

Department of Applied Mathematics

THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

Joint works with: Kaihuang Chen (PolyU), Yancheng Yuan (PolyU), Guojun Zhang (PolyU), and Xinyuan Zhao (BJUT) [1,2].

[1]Sun, Yuan, Zhang, and Zhao. "Accelerating preconditioned ADMM via degenerate proximal point mappings." arXiv preprint arXiv:2403.18618 (2024).

[2]Chen, Sun, Yuan, Zhang, and Zhao. "HPR-LP: An implementation of an HPR method for solving linear programming". arXiv preprint arXiv:2408.12179 (2024).

Outline

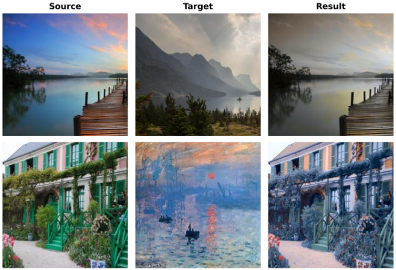

Figure 1: Optimal transport (color transfer).

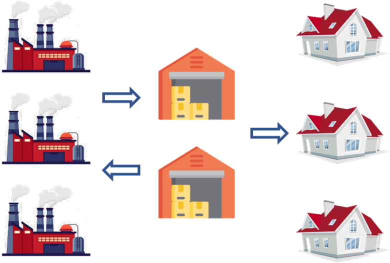

Figure 2: Production planning.

Many important applications require solving large-scale linear programming problems with **over 10 million** constraints and variables.

Convex optimization problem

Consider the following convex optimization problem (COP):

$$\begin{array}{ll} \min_{y \in \mathbb{Y}, z \in \mathbb{Z}} & f_1(y) + f_2(z) \\ \text{subject to} & B_1 y + B_2 z = c, \end{array} \qquad (1)$$

1. $\mathbb{X}$, $\mathbb{Y}$, and $\mathbb{Z}$ are three finite-dimensional real Euclidean spaces;
2. $f_1 : \mathbb{Y} \to (-\infty, +\infty]$ and $f_2 : \mathbb{Z} \to (-\infty, +\infty]$ are two proper closed convex functions;
3. $B_1 : \mathbb{Y} \to \mathbb{X}$ and $B_2 : \mathbb{Z} \to \mathbb{X}$ are two linear operators, $c \in \mathbb{X}$.

Purpose: accelerating an alternating direction method of multipliers (ADMM) with **semi-proximal terms** for solving COP.

The augmented Lagrangian function of problem (1) is defined by, for any $(y, z, x) \in \mathbb{Y} \times \mathbb{Z} \times \mathbb{X}$,

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

The dual of problem (1) is given by

$$\max_{x \in \mathbb{X}} \left\{ -f_1^*(-B_1^* x) - f_2^*(-B_2^* x) - \langle c, x \rangle \right\}, \tag{2}$$

where $B_1^* : \mathbb{X} \to \mathbb{Y}$ and $B_2^* : \mathbb{X} \to \mathbb{Z}$ are the adjoints of B_1 and B_2, respectively; f_1^* and f_2^* are Fenchel conjugate functions of f_1 and f_2, respectively.

Let $w := (y, z, x)$ and $\mathbb{W} := \mathbb{Y} \times \mathbb{Z} \times \mathbb{X}$.

Algorithm 1 A pADMM[3] for solving COP (1)

Input: Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two self-adjoint positive semidefinite linear operators. Choose $w^0 = (y^0, z^0, x^0)$. Set $\sigma > 0$ and $\rho_k \in (0, 2]$ for any $k \geq 0$. For $k = 0, 1, \ldots,$

Step 1. $\bar{z}^k = \arg\min_{z \in \mathbb{Z}} \left\{ L_\sigma \left(y^k, z; x^k \right) + \frac{1}{2} \|z - z^k\|^2_{\mathcal{T}_2} \right\}.$

Step 2. $\bar{x}^k = x^k + \sigma(B_1 y^k + B_2 \bar{z}^k - c).$

Step 3. $\bar{y}^k = \arg\min_{y \in \mathbb{Y}} \left\{ L_\sigma \left(y, \bar{z}^k; \bar{x}^k \right) + \frac{1}{2} \|y - y^k\|^2_{\mathcal{T}_1} \right\}.$

Step 4. $w^{k+1} = (1 - \rho_k) w^k + \rho_k \bar{w}^k.$

- **Connection:** ADMM[4,5], generalized ADMM[6], proximal ADMM[7], and semi-proximal ADMM[8].

[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] Eckstein. Optim. Methods Softw. (1994): 75-83.

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

Example: Convex quadratic programming

Consider the convex quadratic programming (CQP):

$$\begin{array}{ll} \min_{x\in\mathbb{X}} & \frac{1}{2}\langle x, Qx\rangle + \langle c, x\rangle \\ \text{subject to} & Ax - b \in K, \end{array} \tag{3}$$

where $Q : \mathbb{X} \to \mathbb{X}$ is a self-joint positive semidefinite matrix, $A : \mathbb{X} \to \mathbb{Y}$ is a linear operator, $K \subseteq \mathbb{X}$ is a closed convex (polyhedral $\Longrightarrow$ QP) set, and $c \in \mathbb{X}$ and $b \in \mathbb{Y}$.

The restricted-Wolfe dual[9] of (3) is

$$\min_{(y,z)\in\mathbb{Y}\times\mathbb{X}} \left\{ -\langle b, y\rangle + \delta_K^*(-y) + \frac{1}{2}\langle z, Qz\rangle \mid A^*y - Qz = c,\ z \in \mathcal{Z} \right\}, \tag{4}$$

where $\mathcal{Z}$ is any subspace of $\mathbb{X}$ containing $\mathrm{Range}(Q)$, e.g., $\mathcal{Z} = \mathrm{Range}(Q)$.

[9] Li, Sun, and Toh. Math. Program. Comput. (2018): 703-743.

Example: CQP

The augmented Lagrangian function of restricted-Wolfe dual (4) is defined by, for any $(y, z, x) \in \mathbb{Y} \times \mathcal{Z} \times \mathbb{X}$,

$$L_\sigma^{QP}(y, z; x) := -\langle b, y\rangle + \delta_K^*(-y) + \frac{1}{2}\langle z, Qz\rangle + \langle x, A^*y - Qz - c\rangle + \frac{\sigma}{2}\|A^*y - Qz - c\|^2.$$

Let

$$\mathcal{T}_1 = \sigma(\lambda_{\max}(AA^*)\mathcal{I} - AA^*), \quad \mathcal{T}_2 = \sigma Q(\lambda_{\max}(Q)\mathcal{I} - Q).$$

Algorithm 2 A linearized ADMM for solving restricted-Wolfe dual (4)

Input: Choose $w^0 = (y^0, z^0, x^0)$. Set $\sigma > 0$ and $\rho_k \in (0, 2]$ for any $k \geq 0$. For $k = 0, 1, \ldots,$

Step 1. $\bar{z}^k = \arg\min_{z\in\mathcal{Z}} \left\{ L_\sigma^{QP}\left(y^k, z; x^k\right) + \frac{1}{2}\|z - z^k\|^2_{\mathcal{T}_2} \right\}$ [$Q *$ vector]

Step 2. $\bar{x}^k = x^k + \sigma(A^*y^k - Q\bar{z}^k - c)$.

Step 3. $\bar{y}^k = \arg\min_{y\in\mathbb{Y}} \left\{ L_\sigma^{QP}\left(y, \bar{z}^k; \bar{x}^k\right) + \frac{1}{2}\|y - y^k\|^2_{\mathcal{T}_1} \right\}$ [needs $\Pi_K(\cdot)$]

Step 4. $w^{k+1} = (1 - \rho_k)w^k + \rho_k\bar{w}^k$.

Two main approaches of acceleration of pADMM

Table 1: Some existing convergence rate results of pADMM

Paper	Alg.	Dual step	Prim. feas.	Obj. err.	KKT res.	Type
M&S (2013)[10]	ADMM	1	$O(1/k)$	-	$O(1/k)$ ε-subdiff. res.	ergodic
D&Y (2016)[11]	ADMM	1	$o(1/\sqrt{k})$	$o(1/\sqrt{k})$	-	nonergodic
Cui. (2016)[12]	maj. ADMM	$(0, \frac{1+\sqrt{5}}{2})$	$O(1/\sqrt{k})$	-	$O(1/\sqrt{k})$	nonergodic

Two main approaches to accelerate the pADMM:

1. Integrate Nesterov's extrapolation directly into the pADMM to develop accelerated variants;
2. Reformulate pADMM as a fixed-point iterative method, if possible, and then accelerate the pADMM by accelerating the fixed-point iterative method.

[10] Monteiro and Svaiter. SIAM J. Optim. (2013): 475-507. (First version: 2010).

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

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

1. Integrate Nesterov's extrapolation directly

Table 2: Applying Nesterov's extrapolation directly

Ref.	Alg.	Prim. feas.	Obj. err.
L&L (2019)[13]	acc-LADMM	$O(1/k)$	$O(1/k)$
S&T(2022)[14]	acc-pADMM ($\mathcal{T}_1 \succeq 0, \mathcal{T}_2 \succ 0$)	$O(1/k)$	$O(1/k)$

- In Li and Lin (2019), the convergence rate is $O(\frac{1}{1+k(1-\tau)})$ with the dual step length $\tau \in (0.5, 1)$. Furthermore, at the k-th iteration, for $i = 1, 2$,

$$\mathcal{T}_i^k = \sigma(\lambda_{\max}(B_i^* B_i)\mathcal{I} - B_i^* B_i)/\theta_k, \quad \theta_k = \frac{1}{1 + k(1-\tau)},$$

implying the primal step length approaches zero as $k \to \infty$.

[13] Li and Lin. J. Sci. Comput. (2019): 671-699.

[14] Sabach and Teboulle. SIAM J. Optim. (2022): 204-227.

1. Integrate Nesterov's extrapolation directly

Table 3: Applying Nesterov's extrapolation directly

Ref.	Alg.	Prim. feas.	Obj. err.
L&L (2019)[15]	acc-LADMM	$O(1/k)$	$O(1/k)$
S&T(2022)[16]	acc-pADMM $(\mathcal{T}_1 \succeq 0, \mathcal{T}_2 \succ 0)$	$O(1/k)$	$O(1/k)$

- In Sabach and Teboulle (2022), the convergence rate is $O(\frac{1}{k}) + O(\frac{1}{\mu k})$ with the dual step length μ satisfying

$$\mu \in (0, \delta], \quad \delta = 1 - \frac{\sigma \lambda_{\max}(B_2^* B_2)}{\sigma \lambda_{\max}(B_2^* B_2) + \lambda_{\min}(\mathcal{T}_2)} < 1.$$

- **Gap:** cannot handle the case where both $\mathcal{T}_1$ and $\mathcal{T}_2$ are positive semidefinite and/or with large step lengths.

[15] Li and Lin. J. Sci. Comput. (2019): 671-699.
[16] Sabach and Teboulle. SIAM J. Optim. (2022): 204-227.

Example: Halpern Peaceman-Rachford (HPR)[17]

The PR splitting method for solving problem (1):

Algorithm 3 A PR algorithm for solving COP (1)

1: Input: $y^0 \in \operatorname{dom}(f_1)$, $x^0 \in \mathbb{X}$, $\rho \in (0,2)$, and $\sigma > 0$. For $k = 0, 1, \ldots$
2: Step 1. $z^{k+1} = \underset{z \in \mathbb{Z}}{\arg\min} \left\{ L_\sigma(y^k, z; x^k) \right\}$.
3: Step 2. $x^{k+\frac{1}{2}} = x^k + \sigma(B_1 y^k + B_2 z^{k+1} - c)$.
4: Step 3. $y^{k+1} = \underset{y \in \mathbb{Y}}{\arg\min} \left\{ L_\sigma(y, z^{k+1}; x^{k+\frac{1}{2}}) \right\}$.
5: Step 4. $x^{k+1} = x^{k+\frac{1}{2}} + \sigma(B_1 y^{k+1} + B_2 z^{k+1} - c)$.

Rewrite: Given $\sigma > 0$ and $\eta^0 = x^0 + \sigma(B_1 y^0 - c)$,

$$\eta^{k+1} = \mathbf{T}_\sigma^{\mathrm{PR}}(\eta^k) := \boldsymbol{R}_{\sigma \boldsymbol{M}_1} \circ \boldsymbol{R}_{\sigma \boldsymbol{M}_2}(\eta^k), \quad \forall k \geq 0, \tag{5}$$

where

- $\boldsymbol{M}_1 = \partial(f_1^* \circ (-B_1^*)) + c, \quad \boldsymbol{M}_2 = \partial(f_2^* \circ (-B_2^*))$;
- $\boldsymbol{R}_{\mathcal{M}_i} := 2\boldsymbol{J}_{\boldsymbol{M}_i} - \mathcal{I}, \quad \boldsymbol{J}_{\boldsymbol{M}_i} := (\mathcal{I} + \boldsymbol{M}_i)^{-1}$, for $i = 1, 2$.

If $\eta^* \in \operatorname{Fix}\left(\mathbf{T}_\sigma^{\mathrm{PR}}\right)$, then $x^* = \boldsymbol{J}_{\sigma M_2}(\eta^*)$ is a solution to problem (2).

[17] Zhang, Yuan, and Sun. arXiv preprint arXiv:2211.14881 (2022).

2. Accelerate the fixed-point iterative method

Example: Halpern Peaceman-Rachford

Note that $\mathbf{T}_\sigma^{\mathrm{PR}} : \mathbb{X} \rightrightarrows \mathbb{X}$ is nonexpansive. Do not know when the PR splitting method converges.

The Halpern iteration[18] applying to the PR splitting method:

$$\eta^{k+1} := \lambda_k \eta^0 + (1-\lambda_k)\, \mathbf{T}_\sigma^{\mathrm{PR}}(\eta^k), \forall k \geq 0, \tag{6}$$

where $\eta^0 \in \mathbb{X}$ is any given initial point and $\lambda_k \in [0,1]$ is a specified parameter.

Theorem 1.1 (Wittmann (1992)[19])

Let D be a nonempty closed convex subset of $\mathbb{X}$, and let $\mathbf{T} : D \to D$ be a nonexpansive operator such that $\mathrm{Fix}(\mathbf{T}) \neq \varnothing$. Let $\{\lambda_k\}_{k=0}^{\infty}$ be a sequence in $[0,1]$ such that the following hold:

$$\lambda_k \to 0, \quad \sum_{k=0}^{\infty} \lambda_k = +\infty, \quad \sum_{k=0}^{\infty} |\lambda_{k+1} - \lambda_k| < +\infty.$$

Let $\eta^0 \in D$ and set

$$\eta^{k+1} := \lambda_k \eta^0 + (1-\lambda_k)\, \mathbf{T}(\eta^k), \forall k \geq 0.$$

Then $\eta^k \to \Pi_{\mathrm{Fix}(\mathbf{T})}(\eta^0)$.

[18] Halpern. Bull. Amer. Math. Soc. (1967): 957-961.
[19] Wittmann. Arch. Math. (1992): 486-491.

2. Accelerate the fixed-point iterative method

Example: Halpern Peaceman-Rachford

Lieder (2021)[20] showed that when $\lambda_k = 1/(k+2)$ for $k \geq 0$, the Halpern iteration will give the following best possible convergence rate regarding the residual:

$$\|\eta^k - \mathbf{T}_\sigma^{\mathrm{PR}}(\eta^k)\| \leq \frac{2\|\eta^0 - \bar{\eta}\|}{k+1}, \quad \forall k \geq 0 \; and \; \bar{\eta} \in \mathrm{Fix}(\mathbf{T}_\sigma^{\mathrm{PR}}).$$

Take $\lambda_k = 1/(k+2)$. An HPR algorithm is presented as follows:

Algorithm 4 An HPR algorithm for solving COP (1)

1: Input: $y^0 \in \mathrm{dom}(f_1)$, $x^0 \in \mathbb{X}$, and $\sigma > 0$.
2: Initialization: $\hat{x}^0 := x^0$.
3: For $k = 0, 1, \ldots$
4: Step 1. $z^{k+1} = \arg\min_{z \in \mathbb{Z}} \left\{ L_\sigma(y^k, z; \hat{x}^k) \right\}$.
5: Step 2. $x^{k+\frac{1}{2}} = \hat{x}^k + \sigma(B_1 y^k + B_2 z^{k+1} - c)$.
6: Step 3. $y^{k+1} = \arg\min_{y \in \mathbb{Y}} \left\{ L_\sigma(y, z^{k+1}; x^{k+\frac{1}{2}}) \right\}$.
7: Step 4. $x^{k+1} = x^{k+\frac{1}{2}} + \sigma(B_1 y^{k+1} + B_2 z^{k+1} - c)$.
8: Step 5. $\hat{x}^{k+1} = \left(\frac{1}{k+2} x^0 + \frac{k+1}{k+2} x^{k+1} \right) + \frac{\sigma}{k+2} \left[(B_1 y^0 - c) - (B_1 y^{k+1} - c) \right]$.

[20] Lieder. Optim. Lett. (2021): 405-418.

Table 4: Some nonergodic accelerated results by accelerating the fixed-point iterative method

Ref.	Acc.	Prim. feas.	Obj. err.	KKT res.
Kim (2021)[21]	acc-PPM $\rightarrow$ acc-ADMM	$O(1/k)$	-	-
T-D&L (2021)[22]	Halpern + DR $\rightarrow$ acc-ADMM	$O(1/k)$ f_2 strongly conv.	-	-
Zhang. (2022)	Halpern + PR $\rightarrow$ HPR	$O(1/k)$	$O(1/k)$	$O(1/k)$
Yang. (2023)[23]	Halpern+ precond. PPM $\rightarrow$ acc-pADMM ($\mathcal{T}_1 \succ 0$ and $\mathcal{T}_2 \succ 0$)	-	-	$O(1/k)$

- **Gap:** cannot handle the case where both $\mathcal{T}_1$ and $\mathcal{T}_2$ are positive semidefinite;
- **Advantages:** No restrictive requirements on the step lengths.

[21] Kim. Math. Program. (2021): 57-87.
[22] Tran-Dinh and Luo. arXiv preprint arXiv:2110.08150 (2021).
[23] Yang, Zhao, Li, and Sun. arXiv preprint arXiv:2304.11037 (2023).

The Chambolle-Pock scheme

Let $B_2 = -I$ and $c = 0$. The Chambolle-Pock scheme[24]: Given $w^0 := (y^0, x^0) \in \mathbb{Y} \times \mathbb{X}$, $\tau, \sigma > 0$,

$$\left\{ \begin{array}{l} y^{k+1} = J_{\tau \partial f_1}\left(y^k - \tau B_1^* x^k\right), \\ x^{k+1} = J_{\sigma \partial f_2^*}\left(x^k + \sigma B_1\left(2y^{k+1} - y^k\right)\right). \end{array} \right. \tag{7}$$

Define

$$\mathcal{T} := \begin{bmatrix} \partial f_1 & B_1^* \\ -B_1 & \partial f_2^* \end{bmatrix}, \quad \mathcal{M} := \begin{bmatrix} \frac{1}{\tau} I & -B_1^* \\ -B_1 & \frac{1}{\sigma} I \end{bmatrix}.$$

Recently, Bredies et al. (2022)[25] regarded the scheme (7) as a degenerate PPM to discuss its convergence[26]:

$$w^{k+1} = (\mathcal{M} + \mathcal{T})^{-1} \mathcal{M} w^k,$$

where $\mathcal{M}$ is positive semidefinite under the condition of $\tau\sigma\|B_1\|^2 = 1$.

[24] Chambolle and Pock. J. Math. Imaging Vision. (2011): 120-145.

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

[26] The Chambolle-Pock scheme under the condition of $\tau\sigma\|B_1\|^2 \leq 1$ is actually equivalent to LADMM and the convergence properties of LADMM, even with larger dual step lengths in the interval $(0, (1+\sqrt{5})/2)$, have already been covered in the work of Fazel et al. (2013).

Technology roadmap

Insight from the work of Bredies et al. (2022): Motivate us to reformulate the pADMM as a dPPM and accelerate the pADMM by accelerating the dPPM.

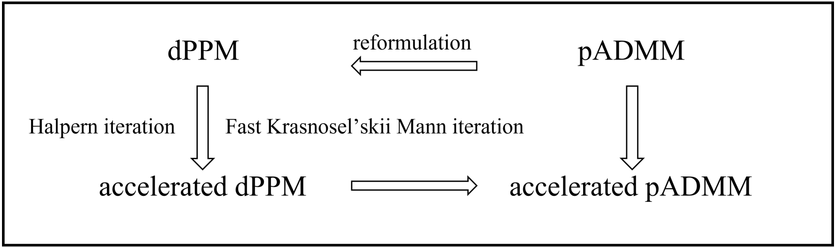

Figure 3: Technology Roadmap

The degenerate proximal point method

Let $\mathcal{H}$ be a real Hilbert space with inner product $\langle\cdot,\cdot\rangle$. Consider the monotone inclusion problem:

$$\text{find } w \in \mathcal{H} \text{ such that } 0 \in \mathcal{T}w, \tag{8}$$

where $\mathcal{T}$ is a maximal monotone operator from $\mathcal{H}$ into itself.

The degenerate proximal point method

Definition 2.1 (admissible preconditioner, Bredies et al. (2022))

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

$$\widehat{\mathcal{T}} = (\mathcal{M} + \mathcal{T})^{-1}\mathcal{M} \tag{9}$$

is single-valued and has full domain.

Let $\mathcal{M}$ be an admissible preconditioner for the maximal monotone operator $\mathcal{T}$. The degenerate PPM (dPPM) in the work of Bredies et al. (2022) for solving the inclusion problem (8) is expressed as follows:

$$w^0 \in \mathcal{H},\ w^{k+1} = (1-\rho_k)w^k + \rho_k \bar{w}^k,\ \bar{w}^k = \widehat{\mathcal{T}} w^k = (\mathcal{M} + \mathcal{T})^{-1}\mathcal{M}w^k, \tag{10}$$

where $\{\rho_k\}$ is a sequence in $[0, 2]$.

The degenerate proximal point method

Define

$$\widehat{\mathcal{Q}} := \mathcal{I} - \widehat{\mathcal{T}} \quad \text{and} \quad \widehat{\mathcal{F}}_\rho := (1-\rho)\mathcal{I} + \rho\widehat{\mathcal{T}}, \quad \rho \in [0,2], \tag{11}$$

where $\mathcal{I}$ is an identity operator on $\mathcal{H}$.

Proposition 2.1

The following things hold:

(a) $\widehat{\mathcal{T}}$ *is $\mathcal{M}$-firmly nonexpansive, i.e.,*

$$\|\widehat{\mathcal{T}}w - \widehat{\mathcal{T}}w'\|_{\mathcal{M}}^2 + \|\widehat{\mathcal{Q}}w - \widehat{\mathcal{Q}}w'\|_{\mathcal{M}}^2 \le \|w - w'\|_{\mathcal{M}}^2, \quad \text{for all } w, w' \in \mathcal{H};$$

(b) $\widehat{\mathcal{F}}_\rho$ *is $\mathcal{M}$-nonexpansive for $\rho \in (0,2]$, i.e.,*

$$\|\widehat{\mathcal{F}}_\rho w - \widehat{\mathcal{F}}_\rho w'\|_{\mathcal{M}} \le \|w - w'\|_{\mathcal{M}}, \quad \text{for all } w, w' \in \mathcal{H}.$$

Theorem 2.1 (Bredies et al. (2022))

Let $\mathcal{T} : \mathcal{H} \to 2^{\mathcal{H}}$ with $\mathcal{T}^{-1}(0) \neq \emptyset$ be a maximal monotone operator and let $\mathcal{M}$ be an admissible preconditioner such that $(\mathcal{M} + \mathcal{T})^{-1}$ is L-Lipschitz. Let $\{w^k\}$ be any sequence generated by the dPPM in (10). *If $0 < \inf_k \rho_k \le \sup_k \rho_k < 2$, then $\{w^k\}$ converges weakly to a point in $\mathcal{T}^{-1}(0)$.*

The connection between the dPPM and the PPM

Bredies et al. (2022) provided a connection between the PPM and the dPPM based on the following decomposition.

Proposition 2.2 (Bredies et al. (2022))

Let $\mathcal{M} : \mathcal{H} \to \mathcal{H}$ be a linear, bounded, self-adjoint, and positive semidefinite operator. Then, there exists a bounded and injective operator $\mathcal{C} : \mathcal{U} \to \mathcal{H}$ for some real Hilbert space $\mathcal{U}$, such that $\mathcal{M} = \mathcal{C}\mathcal{C}^*$, where $\mathcal{C}^* : \mathcal{H} \to \mathcal{U}$ is the adjoint of $\mathcal{C}$. Moreover, if $\mathcal{M}$ has closed range, then $\mathcal{C}^*$ is onto.

Denote by $\widetilde{\mathcal{T}}$ the resolvent of $\mathcal{C}^* \triangleright \mathcal{T} := \left(\mathcal{C}^* \mathcal{T}^{-1} \mathcal{C}\right)^{-1}$, i.e.,

$$\widetilde{\mathcal{T}} := \left(I + \mathcal{C}^* \triangleright \mathcal{T}\right)^{-1}.$$

The connection between the dPPM and the PPM

Proposition 2.3 (Bredies et al. (2022))

Let $\mathcal{T} : \mathcal{H} \to 2^{\mathcal{H}}$ be a maximal monotone operator and let $\mathcal{M}$ be an admissible preconditioner with closed range. Suppose that $\mathcal{M} = \mathcal{C}\mathcal{C}^*$ is a decomposition of $\mathcal{M}$ according to Proposition 2.2 with $\mathcal{C} : \mathcal{U} \to \mathcal{H}$. The parallel composition $\mathcal{C}^* \triangleright \mathcal{T}$ is a maximal monotone operator. Furthermore, $\widetilde{\mathcal{T}}$ has the following identity

$$\widetilde{\mathcal{T}} = \mathcal{C}^*(\mathcal{M} + \mathcal{T})^{-1}\mathcal{C}. \tag{12}$$

In particular, $\widetilde{\mathcal{T}} : \mathcal{U} \to \mathcal{U}$ is everywhere well-defined and firmly nonexpansive. Moreover, for any $\rho \in (0, 2]$, $\widetilde{\mathcal{F}}_\rho = (1 - \rho)\mathcal{I} + \rho\widetilde{\mathcal{T}}$ is nonexpansive and

$$\mathcal{C}^*\mathcal{T}^{-1}(0) = \mathcal{C}^* \operatorname{Fix} \widehat{\mathcal{T}} = \operatorname{Fix} \widetilde{\mathcal{T}} = \operatorname{Fix} \widetilde{\mathcal{F}}_\rho,$$

where we denote the set of fixed-points of an operator $\widehat{\mathcal{T}}$ by $\operatorname{Fix} \widehat{\mathcal{T}}$.

The proximal point method:

$$u^0 = \mathcal{C}^* w^0 \in \mathcal{U}, \quad u^{k+1} = (1 - \rho_k)u^k + \rho_k \bar{u}^k, \quad \bar{u}^k = \widetilde{\mathcal{T}} u^k, \tag{13}$$

with $\{\rho_k\}$ in $[0, 2]$ is equivalent to the dPPM in (10), in the sense that $u^k = \mathcal{C}^* w^k$ for all $k \geq 0$.

The accelerated dPPM

The dPPM in (10) can be reformulated as

$$w^0 \in \mathcal{H}, \quad w^{k+1} = \widehat{\mathcal{F}}_\rho w^k, \tag{14}$$

where $\widehat{\mathcal{F}}_\rho = (1-\rho)\mathcal{I} + \rho\widehat{\mathcal{T}}$ is $\mathcal{M}$-nonexpansive for $\rho \in (0, 2]$.

- Based on the $\mathcal{M}$-nonexpansiveness of $\widehat{\mathcal{F}}_\rho$, one can consider applying the Halpern iteration to (14) to accelerate the dPPM.
- Contreras and Cominetti (2023)[27] demonstrated that the best possible convergence rate for the Halpern iteration is lower bounded by $O(1/k)$ in normed spaces.

[27] Contreras and Cominetti. Math. Program. (2023): 343-374.

The accelerated dPPM

In contrast, given a nonexpansive operator $\widetilde{\mathcal{F}} : \mathcal{H} \to \mathcal{H}$, Bot and Nguyen (2023)[28] proposed the following fast Krasnosel'skiĭ-Mann (KM) iteration: given $\alpha > 2$ and $w^0, w^1 \in \mathcal{H}$,

$$w^{k+1} = w^k + \frac{\alpha}{2(k+\alpha)}(\widetilde{\mathcal{F}}w^k - w^k) + \frac{k}{k+\alpha}\left(\widetilde{\mathcal{F}}w^k - \widetilde{\mathcal{F}}w^{k-1}\right), k \geq 1. \quad (15)$$

Theorem 2.2 (Bot and Nguyen (2023))

Suppose $\mathrm{Fix}(\widetilde{\mathcal{F}}) \neq \varnothing$. *Let* $\{w_k\}$ *be the sequence generated by* (15). *Then* $\{w_k\}$ *converges weakly to an element in* $\mathrm{Fix}\,\widetilde{\mathcal{F}}$ *as* $k \to +\infty$. *Moreover,*

$$\|w^k - w^{k-1}\| = o\left(\frac{1}{k}\right) \quad \text{and} \quad \|w^{k-1} - \widetilde{\mathcal{F}}w^{k-1}\| = o\left(\frac{1}{k}\right) \text{ as } k \to +\infty.$$

It appears to offer better convergence rates than $O(1/k)$ in certain applications.

[28] Boț and Nguyen. SIAM J. Numer. Anal. (2023): 2813-2843.

The accelerated dPPM

When $\alpha = 2$, the fast KM in (15) reduces to the Halpern iteration:

$$w^{k+1} = \frac{1}{k+2} w^0 + \frac{k+1}{k+2} \widetilde{\mathcal{F}} w^k, k \geq 1. \tag{16}$$

Combining the Halpern iteration and fast KM iteration, we propose the following accelerated dPPM:

Algorithm 5 An accelerated dPPM for solving the inclusion problem (8)

Input: Let $\hat{w}^0 = w^0 \in \mathcal{H}$, $\alpha \geq 2$ and $\rho \in (0, 2]$. For $k = 0, 1, \ldots,$
Step 1. $\bar{w}^k = \widehat{\mathcal{T}} w^k$.
Step 2. $\hat{w}^{k+1} = \widehat{\mathcal{F}}_\rho w^k = (1-\rho) w^k + \rho \bar{w}^k$.
Step 3. $w^{k+1} = w^k + \frac{\alpha}{2(k+\alpha)} (\hat{w}^{k+1} - w^k) + \frac{k}{k+\alpha} \left(\hat{w}^{k+1} - \hat{w}^k\right)$.

The shadow sequences

Similar to the connection between the dPPM in (10) and the PPM in (13), we define two shadow sequences $\{u^k\}$ and $\{\bar{u}^k\}$ as follows:

$$u^k := \mathcal{C}^* w^k \text{ and } \bar{u}^k := \mathcal{C}^* \bar{w}^k, \quad \forall k \geq 0, \tag{17}$$

where the sequences $\{w^k\}$ and $\{\bar{w}^k\}$ are generated by Algorithm 5. This leads to the following identity:

$$u^{k+1} = u^k + \frac{\alpha}{2(k+\alpha)}(\widetilde{\mathcal{F}}_\rho u^k - u^k) + \frac{k}{k+\alpha}\left(\widetilde{\mathcal{F}}_\rho u^k - \widetilde{\mathcal{F}}_\rho u^{k-1}\right), \quad \forall k \geq 1, \tag{18}$$

where $\widetilde{\mathcal{F}}_\rho = (1-\rho)\mathcal{I} + \rho\widetilde{\mathcal{T}}$ for $\rho \in (0, 2]$ is nonexpansive by Proposition 2.3.

Global convergence of the accelerated dPPM

With the help of the shadow sequences $\{u^k\}$ and $\{\bar{u}^k\}$, we can obtain the global convergence of the accelerated dPPM.

Theorem 2.3

Let $\mathcal{T} : \mathcal{H} \to 2^{\mathcal{H}}$ with $\mathcal{T}^{-1}(0) \neq \emptyset$ be a maximal monotone operator, and let $\mathcal{M}$ be an admissible preconditioner with a closed range such that $(\mathcal{M} + \mathcal{T})^{-1}$ is continuous. Suppose that $\mathcal{M} = \mathcal{C}\mathcal{C}^*$ is a decomposition of $\mathcal{M}$ according to Proposition 2.2 with $\mathcal{C} : \mathcal{U} \to \mathcal{H}$. The following conclusions hold for the sequences $\{\bar{w}^k\}$, $\{\hat{w}^k\}$, and $\{w^k\}$ generated by the accelerated dPPM in Algorithm 5:

(a) *If $\alpha = 2$, then the sequence $\{\bar{w}^k\}$ converges strongly to a fixed point $w^* = (\mathcal{M} + \mathcal{T})^{-1}\mathcal{C}\Pi_{\mathcal{C}^*\mathcal{T}^{-1}(0)}(\mathcal{C}^* w^0)$ in $\mathcal{T}^{-1}(0)$, where $\Pi_{\mathcal{C}^*\mathcal{T}^{-1}(0)}(\cdot)$ is the projection operator onto the closed convex set $\mathcal{C}^*\mathcal{T}^{-1}(0)$; moreover, if $\rho \in (0, 2)$, then the sequences $\{w^k\}$ and $\{\hat{w}^k\}$ also converge strongly to w^*;*

(b) *If $\alpha > 2$, then the sequence $\{\bar{w}^k\}$ converges weakly to a fixed-point in $\mathcal{T}^{-1}(0)$.*

Convergence rate of the accelerated dPPM

Proposition 2.4

Let $\mathcal{T} : \mathcal{H} \to 2^{\mathcal{H}}$ with $\mathcal{T}^{-1}(0) \neq \emptyset$ be a maximal monotone operator and let $\mathcal{M}$ be an admissible preconditioner with closed range. The sequences $\{w^k\}$ and $\{\hat{w}^k\}$ generated by Algorithm 5 satisfy the following:

(a) *if $\alpha = 2$, then*

$$\|w^k - \hat{w}^{k+1}\|_{\mathcal{M}} \leq \frac{2\,\|w^0 - w^*\|_{\mathcal{M}}}{k+1}, \quad \forall k \geq 0 \text{ and } w^* \in \mathcal{T}^{-1}(0); \tag{19}$$

(b) *if $\alpha > 2$, then*

$$\|w^{k+1} - w^k\|_{\mathcal{M}} = o\left(\frac{1}{k+1}\right) \text{ and } \|w^k - \hat{w}^{k+1}\|_{\mathcal{M}} = o\left(\frac{1}{k+1}\right) \text{ as } k \to +\infty. \tag{20}$$

1. The rates are inherited from the results of the Halpern iteration and the fast KM iteration applied to the nonexpansive operator $\widetilde{\mathcal{F}}_\rho$.
2. Without acceleration, similar to Proposition 8 in the work of Brézis and Lions (1978)[29], the dPPM with $\rho = 1$ has an $O(1/\sqrt{k})$ convergence rate with respect to $\|w^k - w^{k-1}\|_{\mathcal{M}},\ k \geq 1$.

[29] Brézis and Lions. Israel J. Math. (1978): 329-345.

Convergence rate of the accelerated dPPM

Even when the proximal term $\mathcal{M}$ is positive semidefinite, the accelerated dPPM can achieve an $O(1/k)$ convergence rate in terms of the operator residual under the Euclidean norm.

Corollary 2.1

Let $\mathcal{T} : \mathcal{H} \to 2^{\mathcal{H}}$ with $\mathcal{T}^{-1}(0) \neq \emptyset$ be a maximal monotone operator and let $\mathcal{M}$ be an admissible preconditioner with closed range such that $(\mathcal{M} + \mathcal{T})^{-1}$ is L-Lipschitz. Suppose that $\mathcal{M} = \mathcal{C}\mathcal{C}^*$ is a decomposition of $\mathcal{M}$ according to Proposition 2.2 with $\mathcal{C} : \mathcal{U} \to \mathcal{H}$. Let $\|\mathcal{C}\| := \sup_{\|w\| \leq 1} \|\mathcal{C}w\|$ represent the spectral norm of the linear operator $\mathcal{C}$. Choose $\alpha = 2$ and $\rho = 1$. Then the sequences $\{w^k\}$ and $\{\bar{w}^k\}$ generated by Algorithm 5 satisfiy

$$\|w^k - \bar{w}^k\| \leq \frac{1}{k+1}\|w^0 - w^*\| + \frac{(5k+1)L\|\mathcal{C}\|}{(k+1)^2}\|w^0 - w^*\|_{\mathcal{M}}, \ \forall k \geq 0 \text{ and } w^* \in \mathcal{T}^{-1}(0).$$

The equivalence between the pADMM and the dPPM

To reformulate the pADMM in (1) as a dPPM in (10), we first introduce $\mathcal{T}$ and $\mathcal{M}$ for further analysis:

- Define the maximal monotone operator $\mathcal{T}: \mathbb{W} \to \mathbb{W}$ as follows:

$$\mathcal{T}w = \begin{pmatrix} \partial f_1(y) + B_1^* x \\ \partial f_2(z) + B_2^* x \\ c - B_1 y - B_2 z \end{pmatrix}, \quad \forall w = (y, z, x) \in \mathbb{W}. \tag{21}$$

- Define the self-adjoint linear operator $\mathcal{M}: \mathbb{W} \to \mathbb{W}$ as follows:

$$\mathcal{M} = \begin{bmatrix} \sigma B_1^* B_1 + \mathcal{T}_1 & 0 & B_1^* \\ 0 & \mathcal{T}_2 & 0 \\ B_1 & 0 & \sigma^{-1}\mathcal{I} \end{bmatrix}. \tag{22}$$

Assumption

- The KKT system of problem (1):

$$-B_1^* x^* \in \partial f_1(y^*), \quad -B_2^* x^* \in \partial f_2(z^*), \quad B_1 y^* + B_2 z^* - c = 0. \tag{23}$$

Assumption 3.1

The KKT system (23) *has a nonempty solution set.*

- Since f_1 and f_2 are proper closed convex functions, there exist two self-adjoint and positive semidefinite operators Σ_{f_1} and Σ_{f_2} such that for all $y, \hat{y} \in \mathrm{dom}(f_1), \phi \in \partial f_1(y)$, and $\hat{\phi} \in \partial f_1(\hat{y})$,

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

and for all $z, \hat{z} \in \mathrm{dom}(f_2), \varphi \in \partial f_2(z)$, and $\hat{\varphi} \in \partial f_2(\hat{z})$,

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

Assumption 3.2

Both $\Sigma_{f_1} + B_1^* B_1 + \mathcal{T}_1$ *and* $\Sigma_{f_2} + B_2^* B_2 + \mathcal{T}_2$ *are positive definite.*

The equivalence between the pADMM and the dPPM

Proposition 3.1

Suppose that Assumption 3.2 holds. Consider the operators $\mathcal{T}$ defined in (21) and $\mathcal{M}$ defined in (22). Then the sequence $\{w^k\}$ generated by the pADMM in Algorithm 1 coincides with the sequence $\{w^k\}$ generated by the dPPM in (10) with the same initial point $w^0 \in \mathbb{W}$. Additionally, $\mathcal{M}$ is an admissible preconditioner such that $(\mathcal{M} + \mathcal{T})^{-1}$ is Lipschitz continuous.

Based on the equivalence between the dPPM and the pADMM, we can

- deduce the convergence of pADMM in Algorithm 1 with varying relaxation factors $\rho_k \in (0, 2)$ for $k \geq 0$ by applying the convergence results of the dPPM;
- employ the accelerated dPPM introduced in Algorithm 5 to derive an accelerated pADMM.

An accelerated pADMM

Algorithm 6 An accelerated pADMM for solving COP (1)

Input: Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two self-adjoint positive semidefinite linear operators. Choose $w^0 = (y^0, z^0, x^0)$. Let $\hat{w}^0 := w^0$. Set $\sigma > 0$, $\alpha \geq 2$ and $\rho \in (0, 2]$. For $k = 0, 1, \dots,$

Step 1. $\bar{z}^k = \underset{z \in \mathbb{Z}}{\arg\min} \left\{ L_\sigma \left(y^k, z; x^k \right) + \frac{1}{2} \|z - z^k\|^2_{\mathcal{T}_2} \right\}$.

Step 2. $\bar{x}^k = x^k + \sigma(B_1 y^k + B_2 \bar{z}^k - c)$.

Step 3. $\bar{y}^k = \underset{y \in \mathbb{Y}}{\arg\min} \left\{ L_\sigma \left(y, \bar{z}^k; \bar{x}^k \right) + \frac{1}{2} \|y - y^k\|^2_{\mathcal{T}_1} \right\}$.

Step 4. $\hat{w}^{k+1} = (1 - \rho) w^k + \rho \bar{w}^k$.

Step 5. $w^{k+1} = w^k + \frac{\alpha}{2(k+\alpha)} (\hat{w}^{k+1} - w^k) + \frac{k}{k+\alpha} \left(\hat{w}^{k+1} - \hat{w}^k \right)$.

Corollary 3.1

Suppose that Assumptions 3.1 and 3.2 hold. The sequence $\{\bar{w}^k\} = \{(\bar{y}^k, \bar{z}^k, \bar{x}^k)\}$ generated by Algorithm 6 converges to the point $w^* = (y^*, z^*, x^*)$, where (y^*, z^*) is a solution to problem (1) and x^* is a solution to problem (2).

Connection to existing algorithms

1. When $\mathcal{T}_i = 0$ for $i = 1, 2$ in Algorithm 6, we can obtain an accelerated ADMM. In addition, if $\alpha = 2$, this algorithm is equivalent to the HPR in terms of the sequence $\{(\bar{y}^k, \bar{z}^k, \bar{x}^k)\}$.
2. Setting $\mathcal{T}_i = \sigma(\lambda_{\max}(B_i^* B_i)\mathcal{I} - B_i^* B_i)$ for $i = 1, 2$ in Algorithm 6, we can obtain an accelerated LADMM. Compared to the algorithm in Li and Lin (2019), the $\mathcal{T}_i$ for $i = 1, 2$ in Algorithm 6 will not tend to infinity as k increases, which implies that this accelerated LADMM has a larger primal step length.
3. Both $\mathcal{T}_1$ and $\mathcal{T}_2$ in Algorithm 6 can be positive semidefinite under Assumption 3.2, which is a significant difference compared to work of Sabach and Teboulle (2022) (Only $\mathcal{T}_1$ can be positive semidefinite).
4. The accelerated pADMM introduced in Yang et al. (2023), where both $\mathcal{T}_1$ and $\mathcal{T}_2$ are positive definite, is a special case of Algorithm 6 with $\alpha = 2$.

Convergence rate of the accelerated pADMM

To analyze the convergence rate of Algorithm 6, we define the following residual mapping associated with the KKT system (23):

$$\mathcal{R}(w) = \begin{pmatrix} y - \mathrm{Prox}_{f_1}(y - B_1^* x) \\ z - \mathrm{Prox}_{f_2}(z - B_2^* x) \\ c - B_1 y - B_2 z \end{pmatrix}, \quad \forall w = (y, z, x) \in \mathbb{W}, \tag{24}$$

and the objective error:

$$h(\bar{y}^k, \bar{z}^k) := f_1(\bar{y}^k) + f_2(\bar{z}^k) - f_1(y^*) - f_2(z^*), \quad \forall k \geq 0,$$

where (y^*, z^*) is the limit point of the sequence $\{(\bar{y}^k, \bar{z}^k)\}$.

Convergence rate of the accelerated pADMM

Theorem 3.1

Suppose that Assumptions 3.1 and 3.2 hold. Let $\{(\bar{y}^k, \bar{z}^k, \bar{x}^k)\}$ *be the sequence generated by Algorithm 6, and let* $w^* = (y^*, z^*, x^*)$ *be the limit point of the sequence* $\{(\bar{y}^k, \bar{z}^k, \bar{x}^k)\}$ *and* $R_0 = \|w^0 - w^*\|_{\mathcal{M}}$.

(a) *If* $\alpha = 2$, *then for all* $k \geq 0$, *we have the following bounds:*

$$\|\mathcal{R}(\bar{w}^k)\| \leq \left(\frac{\sigma\|B_1^*\| + 1}{\sqrt{\sigma}} + \|\sqrt{\mathcal{T}_2}\| + \|\sqrt{\mathcal{T}_1}\| \right) \frac{2R_0}{\rho(k+1)} \tag{25}$$

and

$$\left(\frac{-1}{\sqrt{\sigma}}\|x^*\| \right) \frac{2R_0}{\rho(k+1)} \leq h(\bar{y}^k, \bar{z}^k) \leq \left(3R_0 + \frac{1}{\sqrt{\sigma}}\|x^*\| \right) \frac{2R_0}{\rho(k+1)}. \tag{26}$$

(b) *If* $\alpha > 2$, *then we have the following bounds:*

$$\|\mathcal{R}(\bar{w}^k)\| = \left(\frac{\sigma\|B_1^*\|+1}{\sqrt{\sigma}} + \|\sqrt{\mathcal{T}_2}\| + \|\sqrt{\mathcal{T}_1}\| \right) o\left(\frac{1}{k+1} \right) \text{ as } k \to +\infty \tag{27}$$

and

$$|h(\bar{y}^k, \bar{z}^k)| = o\left(\frac{1}{k+1} \right) \text{ as } k \to +\infty. \tag{28}$$

Convergence rate of the accelerated pADMM

Table 5: Comparison of the convergence rates of accelerated pADMM variants

Ref.	Prox. oper.	Prim. feas.	Obj. err.	KKT res.
L&L (2019)	$\mathcal{T}_i^k = \sigma(\lambda_{\max}(B_i^* B_i)\mathcal{I} - B_i^* B_i)/\theta_k$, $\theta_k \to 0, i = 1, 2$	$O(1/k)$	$O(1/k)$	-
S&T (2022)	$\mathcal{T}_1 \succeq 0, \mathcal{T}_2 \succ 0$	$O(1/k)$	$O(1/k)$	-
Kim (2021)	$\mathcal{T}_1 = 0, \mathcal{T}_2 = 0$	$O(1/k)$	-	-
T-D&L (2021)	$\mathcal{T}_1 = 0, \mathcal{T}_2 = 0$	$O(1/k)$ f_2 strongly conv.	-	-
Zhang. (2022)	$\mathcal{T}_1 = 0, \mathcal{T}_2 = 0$	$O(1/k)$	$O(1/k)$	$O(1/k)$
Yang. (2023)	$\mathcal{T}_1 \succ 0, \mathcal{T}_2 \succ 0$	-	-	$O(1/k)$
Ours	$\mathcal{T}_1 \succeq 0, \mathcal{T}_2 \succeq 0$	$O(1/k)$ or $o(1/k)$	$O(1/k)$ or $o(1/k)$	$O(1/k)$ or $o(1/k)$

Numerical experiments: Linear programming

Consider the following linear programming (LP) problem:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & \langle c, x\rangle \\ \text{s.t.} \quad & A_1 x = b_1 \\ & A_2 x \geq b_2 \\ & x \in C, \end{aligned} \tag{29}$$

where $A_1 \in \mathbb{R}^{m_1\times n}$, $A_2 \in \mathbb{R}^{m_2\times n}$, $b_1 \in \mathbb{R}^{m_1}$, $b_2 \in \mathbb{R}^{m_2}$, $c \in \mathbb{R}^n$, and $C := \{x \in \mathbb{R}^n \mid l \leq x \leq u\}$ with $l \in (\mathbb{R}\cup\{-\infty\})^n$ and $u \in (\mathbb{R}\cup\{+\infty\})^n$. Let $A = [A_1; A_2] \in \mathbb{R}^{m\times n}$ with $m = m_1 + m_2$, and $b = [b_1; b_2] \in \mathbb{R}^m$. Then, the dual of problem (29) is given by

$$\begin{aligned} \min_{y\in\mathbb{R}^m, z\in\mathbb{R}^n} \quad & -\langle b, y\rangle + \delta_D(y) + \delta_C^*(-z) \\ \text{s.t.} \quad & A^* y + z = c, \end{aligned} \tag{30}$$

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

Experimental design:

1. Evaluate the performance of various algorithms by solving the dual problem (30):
   - LADMM with $\rho = 1.8$ as described in Algorithm 1;
   - LADMM with $\tau = 1.618$ as proposed by Fazel et al. (2013);
   - Accelerated LADMM by Li and Lin (2019);
   - Accelerated LADMM by Sabach and Teboulle (2022);
   - Accelerated LADMM as in Algorithm 6 with $\alpha = 2$ (e_2), $\alpha = 5$ (e_5), and $\alpha = 15$ (e_{15}).

2. Evaluate the performance of the LP solver: HPR-LP[30], an implementation of an HPR method with semi-proximal terms (an accelerated pADMM with $\alpha = 2$ in Algorithm 6) for solving LP.

[30] Chen, Sun, Yuan, Zhang, and Zhao. "HPR-LP: An implementation of an HPR method for solving linear programming". arXiv preprint arXiv:2408.12179 (2024)

Numerical experiments: Part 1

- The relative KKT residual is defined as:

$$\mathrm{KKT}_{\mathrm{res}} = \max\left\{ \frac{\|b - Ax\|}{1 + \|b\|}, \frac{\left\|A^T y + z - c\right\|}{1 + \|c\|}, \frac{\|x - \Pi_K(x - z)\|}{1 + \|x\| + \|z\|} \right\}.$$

- All tested algorithms are terminated when $\mathrm{KKT}_{\mathrm{res}} \leq 10^{-4}$. The maximum iteration count is set to 10^6, and the maximum runtime is 3600 seconds.
- The algorithm is restarted for the r-th time if either of the following conditions[31] is met:

$$\begin{cases} \mathrm{KKT}_{\mathrm{res}}(w^{r-1,t}) \leq 0.3 \times \mathrm{KKT}_{\mathrm{res}}(w^{r-1,0}), \\ t/k \geq 0.1. \end{cases} \tag{31}$$

  Upon restarting, set

$$\sigma_r = \sqrt{\frac{\|x^{r,0} - x^{r-1,0}\|}{\|y^{r,0} - y^{r-1,0}\|}} \cdot \sigma_{r-1}, \quad r \geq 1, \text{ with } \sigma_0 = 1.$$

[31] Applegate, Díaz, Hinder, Lu, Lubin, O'Donoghue, and Schudy. Adv Neural Inf Process Syst. (2021): 20243-20257.

Numerical experiments: Part 1

The data collection is sourced from Mittelmann[32], and we preprocess it using Gurobi's presolve feature.

Table 6: The problem size of tested data collection

No.	problem	nRow	nCol
1	'a2864_presolved'	20669	34493
2	'datt256_lp_presolved'	9863	196147
3	'ex10_presolved'	62934	78632
4	'karted_presolved'	46501	133114
5	'neos-3025225_lp_presolved'	81172	151017
6	'neos-5052403-cygnet_presolved'	19134	46727
7	'nug08-3rd_presolved'	19728	29856
8	'pds-100_presolved'	94994	441224
9	'qap15_presolved'	6330	22275
10	'rail4284_presolved'	4176	1094702
11	'scpm1_lp_presolved'	5000	67631
12	'set-cover_presolved'	10000	1112008
13	'stp3d_presolved'	95279	205516

[32] https://plato.asu.edu/ftp/lptestset/

Numerical experiments: Part 1

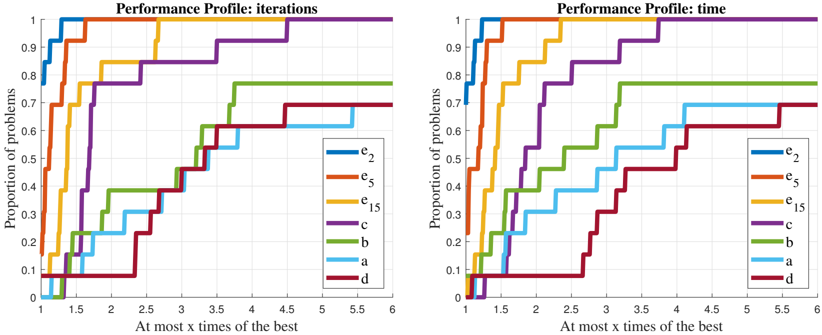

Figure 4: Performance profile of different methods. (a: LADMM with $\rho = 1.8$ in Algorithm 1; b: LADMM with $\tau = 1.618$ in Fazel et al. (2013); c: accelerated LADMM in Li and Lin (2019); d: accelerated LADMM in Sabach and Teboulle (2022); e_2, e_5, e_{15}: accelerated LADMM in Algorithm 6 with $\alpha = 2, 5, 15$, respectively.

Algorithm 7 HPR-LP

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

Initialization: Set $r = 0$, $k = 0$, and $\sigma_0 > 0$.

repeat

initialize the inner loop: set inner loop counter $t = 0$;

repeat

$$\bar{z}^{r,t+1} = \arg\min_{z \in \mathbb{R}^n} \left\{ L_{\sigma_r}\left(y^{r,t}, z; x^{r,t}\right) \right\};$$

$$\bar{x}^{r,t+1} = x^{r,t} + \sigma_r(A^* y^{r,t} + \bar{z}^{r,t+1} - c);$$

$$\bar{y}^{r,t+1} = \arg\min_{y \in \mathbb{R}^m} \left\{ L_{\sigma_r}\left(y, \bar{z}^{r,t+1}; \bar{x}^{r,t+1}\right) + \frac{\sigma_r}{2} \|y - y^{r,t}\|^2_{\mathcal{T}_1} \right\};$$

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

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

$t = t + 1$, $k = k + 1$;

until one of the restart criteria holds or termination criteria hold

restart the inner loop: $\tau_r = t, w^{r+1,0} = \bar{w}^{r,\tau_r}$,

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

until termination criteria hold

Output: $\{\bar{w}^{r,t}\}$.

HPR-LP: Restart criteria

Based on the iteration complexity of $O(1/k)$ in terms of the KKT residual (as derived from Proposition 2.4), we define the merit function:

$$\widetilde{R}_{r,l} = \|w^{r,t} - \hat{w}^{r,t+1}\|_{\mathcal{M}}.$$

The restart criteria in HPR-LP are as follows:

1. Sufficient decay of $\widetilde{R}_{r,t}$:
$$\widetilde{R}_{r,t} \leq \alpha_1 \widetilde{R}_{r,0}; \tag{32}$$
2. Necessary decay + no local progress of $\widetilde{R}_{r,t}$:
$$\widetilde{R}_{r,t} \leq \alpha_2 \widetilde{R}_{r,0} \quad \text{and} \quad \widetilde{R}_{r,t+1} > \widetilde{R}_{r,t}; \tag{33}$$
3. Long inner loop:
$$t \geq \alpha_3 k, \tag{34}$$

where $\alpha_1 \in (0, \alpha_2)$, $\alpha_2 \in (0,1)$, and $\alpha_3 \in (0,1)$. In HPR-LP, we set $\alpha_1 = 0.2$, $\alpha_2 = 0.6$, and $\alpha_3 = 0.2$.

HPR-LP: Update rule for σ

To minimize the upper bound of the complexity results $\left\|w^{r+1,0} - w^*\right\|_{\mathcal{M}}$ in Proposition 2.4 at the $(r+1)$-th outer loop, we update σ as follows

$$\begin{aligned}\sigma_{r+1} &= \arg\min_{\sigma} \left\|w^{r+1,0} - w^*\right\|_{\mathcal{M}}^2 \\ &= \arg\min_{\sigma} \left(\sigma\|y^{r+1,0} - y^*\|_{\mathcal{T}_1}^2 + \sigma^{-1}\|x^{r+1,0} - x^* + \sigma A^*(y^{r+1,0} - y^*)\|^2\right) \\ &= \sqrt{\frac{\|x^{r+1,0} - x^*\|^2}{\|y^{r+1,0} - y^*\|_{\mathcal{T}_1}^2 + \|A^*(y^{r+1,0} - y^*)\|^2}}.\end{aligned} \tag{35}$$

In HPR-LP, we update σ_{r+1} using the approximation:

$$\sigma_{r+1} = \frac{\Delta_x}{\Delta_y}, \tag{36}$$

where

$$\Delta_x := \|\bar{x}^{r,\tau_r} - x^{r,0}\| \text{ and } \Delta_y := \sqrt{\|\bar{y}^{r,\tau_r} - y^{r,0}\|_{\mathcal{T}_1}^2 + \|A^*(\bar{y}^{r,\tau_r} - y^{r,0})\|^2}. \tag{37}$$

Numerical experiments: Part 2

Benchmark datasets.

- 49 publicly available Mittelmann's LP benchmark instances;
- 380 instances of MIP relaxations from the MIPLIB 2017 collection[33];
- 20 LP instances generated from quadratic assignment problems (QAPs)[34] and the "zib03" instance[35].

Software and computing environment.

- HPR-LP is implemented in Julia, referred to as HPR-LP.jl;
- cuPDLP.jl[36], the GPU version of the award-winning solver PDLP[37], is also implemented in Julia;
- All tested solvers are run on an NVIDIA A100-SXM4-80GB GPU with CUDA 12.3.

---

[33] Gleixner, Hendel, Gamrath, Achterberg, Bastubbe, Berthold, Christophel, Jarck, Koch, Linderoth, Lübbecke. Math. Program. Comput. (2021): 443-490.

[34] Burkard, Karisch, Rendl. J. Global Optim. (1997) 391-403.

[35] Koch, Berthold, Pedersen, Vanaret. EURO J. Comput. Optim. (2022): 100031.

[36] Lu and Yang. arXiv preprint arXiv:2311.12180 (2023).

[37] Applegate, Díaz, Hinder, Lu, Lubin, O'Donoghue, and Schudy were awarded the Beale–Orchard-Hays Prize for Excellence in Computational Mathematical Programming at the 25th International Symposium on Mathematical Programming (https://ismp2024.gerad.ca/), July 21-26, 2024, Montréal, Canada.

Termination criteria. We terminate HPR-LP when the following stopping criteria used in PDLP are satisfied for the tolerance $\varepsilon \in (0, \infty)$:

$$
\begin{aligned}
&|\langle b, y\rangle - \delta_C^*(-z) - \langle c, x\rangle| \leq \varepsilon \left(1 + |\langle b, y\rangle - \delta_C^*(-z)| + |\langle c, x\rangle|\right), \\
&\|\Pi_D(b - Ax)\| \leq \varepsilon \left(1 + \|b\|\right), \\
&\|c - A^* y - z\| \leq \varepsilon \left(1 + \|c\|\right).
\end{aligned}
$$

Shifted geometric mean. We use the shifted geometric mean of solving time to measure the performance of solvers on a collection of problems:

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

where t_i is the solving time in seconds for the i-th instance. We shift by $\Delta = 10$ and denote this measure as SGM10.

Part 2: Mittelmann's LP benchmark with presolve

Table 7: Numerical performance of different solvers on 49 instances of Mittelmann's LP benchmark set 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 across all tolerance levels;
- In terms of SGM10, HPR-LP.jl achieves a **3.71x** speedup over cuPDLP.jl for 10^{-8} accuracy on the presolved dataset.

Part 2: Mittelmann's LP benchmark without presolve

Table 8: Numerical performance of different solvers on 49 instances of Mittelmann's LP benchmark set 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 consistently solves **3-5 more** problems than cuPDLP.jl does across all tolerance levels;
- In terms of SGM10, HPR-LP.jl achieves a **2.68x** speedup over cuPDLP.jl to obtain a solution with a 10^{-8} relative accuracy for the unpresolved dataset.

Part 2: MIP relaxations with presolve

Table 9: Numerical performance of different solvers on 380 instances of MIP relaxations with presolve.

Tolerance	10^{-4}		10^{-6}		10^{-8}	
Solver	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

- With a 10^{-8} accuracy, HPR-LP.jl solves **7 more** problems than cuPDLP.jl does across the 380 presolved MIP relaxation instances;
- In terms of SGM10, HPR-LP.jl achieves a **2.39x** speedup over cuPDLP.jl to obtain a solution with a 10^{-8} accuracy for the presolved dataset.

Table 10: Numerical performance of different solvers on 380 instances of MIP relaxations 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

- With a 10^{-8} accuracy, HPR-LP.jl solves **4 more** problems than cuPDLP.jl does across the 380 unpresolved MIP relaxation instances;
- In terms of SGM10, HPR-LP.jl achieves a **2.03x** speedup over cuPDLP.jl to obtain a solution with a 10^{-8} accuracy for the unpresolved dataset.

Part 2: QAP problem instances

Table 11: SGM10 for different solvers 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

- HPR-LP.jl achieves a **5.70x** speedup over cuPDLP.jl for 10^{-8} accuracy on the presolved dataset.

Table 12: SGM10 for different solvers 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

- On the unpresolved dataset, HPR-LP.jl achieves a **2.57x** speedup over cuPDLP.jl for the 10^{-8} accuracy.

Part 2: QAP problem instances

Table 11: SGM10 for different solvers 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

- HPR-LP.jl achieves a **5.70x** speedup over cuPDLP.jl for 10^{-8} accuracy on the presolved dataset.

Table 12: SGM10 for different solvers 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

- On the unpresolved dataset, HPR-LP.jl achieves a **2.57x** speedup over cuPDLP.jl for the 10^{-8} accuracy.

Part 2: ZIB problem instance

Dimensions of matrix A in "zib03":

- After presolve: 19,701,908 rows, 29,069,187 columns, 104,300,584 non-zeros;
- Without presolve: 19,731,970 rows, 29,128,799 columns, 104,422,573 non-zeros.

Table 13: Solving time in seconds for the "zib03" instance.

Tolerance	10^{-4}		10^{-6}		10^{-8}	
Solver	HPR-LP.jl	cuPDLP.jl	HPR-LP.jl	cuPDLP.jl	HPR-LP.jl	cuPDLP.jl
With presolve	273.8	351.9	1317.2	1634.6	3685.8	16462.2
Without presolve	154.2	237.7	1063.6	1963.9	4865.3	19746.4

The commercial LP solver COPT used 16.5 hours to solve this instance on an AMD Ryzen 9 5900X.[38]

- HPR-LP.jl achieves a **4.47x** speedup over cuPDLP.jl on the presolved dataset and a **4.06x** speedup on the unpresolved dataset, both in terms of SGM10, to return a solution with a 10^{-8} relative accuracy.

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

Conclusion

- We proposed an accelerated dPPM with both asymptotic $o(1/k)$ and non-asymptotic $O(1/k)$ convergence rates by unifying the Halpern iteration and the fast Krasnosel'skiĭ-Mann iteration.
- Leveraging the equivalence between the pADMM and the dPPM, we derived an accelerated pADMM, which exhibited both asymptotic $o(1/k)$ and non-asymptotic $O(1/k)$ convergence rates with respect to the KKT residual and the objective error.
- The Julia version of HPR-LP achieves a **2.39x** to **5.70x** speedup measured by SGM10 on benchmark datasets with presolve (**2.03x** to **4.06x** without presolve) over the award-winning solver PDLP with the tolerance of 10^{-8}.

Thank you for your attention!

References I

David Applegate, 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, 34:20243–20257, 2021.

David Applegate, Oliver Hinder, Haihao Lu, and Miles Lubin.
Faster first-order primal-dual methods for linear programming using restarts and sharpness.
Math. Program., 201(1):133–184, 2023.

Rainer E Burkard, Stefan E Karisch, and Franz Rendl.
QAPLIB–a quadratic assignment problem library.
J. Glob. Optim., 10:391–403, 1997.

Haim Brézis and Pierre Louis Lions.
Produits infinis de résolvantes.
Israel J. Math., 29:329–345, 1978.

References II

Radu Ioan Boț and Dang-Khoa Nguyen.
Fast Krasnosel'skiĭ-Mann algorithm with a convergence rate of the fixed point iteration of $o(1/k)$.
SIAM J. Numer. Anal., 61(6):2813–2843, 2023.

Kristian Bredies and Hongpeng Sun.
Preconditioned Douglas–Rachford splitting methods for convex-concave saddle-point problems.
SIAM J. Numer. Anal., 53(1):421–444, 2015.

Juan Pablo Contreras and Roberto Cominetti.
Optimal error bounds for non-expansive fixed-point iterations in normed spaces.
Math. Program., 199(1-2):343–374, 2023.

Ying Cui, 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.
J. Optim. Theory Appl., 169:1013–1041, 2016.

Antonin Chambolle and Thomas Pock.
A first-order primal-dual algorithm for convex problems with applications to imaging.
J. Math. Imaging Vision, 40:120–145, 2011.

Damek Davis and Wotao Yin.
Convergence rate analysis of several splitting schemes.
In Roland Glowinski, Stanley J Osher, and Wotao Yin, editors, *Splitting Methods in Communication, Imaging, Science, and Engineering*, pages 115–163. Springer, 2016.

Jonathan Eckstein and Dimitri P Bertsekas.
On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators.
Math. Program., 55(1):293–318, 1992.

Jonathan Eckstein.
Some saddle-function splitting methods for convex programming.
Optim. Methods Softw., 4(1):75–83, 1994.

References IV

Maryam Fazel, Ting Kei Pong, Defeng Sun, and Paul Tseng.
Hankel matrix rank minimization with applications to system identification and realization.
SIAM J. Matrix Anal. Appl., 34(3):946–977, 2013.

Roland Glowinski 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(R2):41–76, 1975.

Daniel Gabay and Bertrand Mercier.
A dual algorithm for the solution of nonlinear variational problems via finite element approximation.
Comput. Math. Appl., 2(1):17–40, 1976.

Benjamin Halpern.
Fixed points of nonexpanding maps.
Bull. Amer. Math. Soc., 73(6):957–961, 1967.

References V

Thorsten Koch, Timo Berthold, Jaap Pedersen, and Charlie Vanaret.
Progress in mathematical programming solvers from 2001 to 2020.
EURO J. Comput. Optim., 10:100031, 2022.

Donghwan Kim.
Accelerated proximal point method for maximally monotone operators.
Math. Program., 190(1-2):57–87, 2021.

Felix Lieder.
On the convergence rate of the Halpern-iteration.
Optim. Lett., 15(2):405–418, 2021.

Huan Li and Zhouchen Lin.
Accelerated alternating direction method of multipliers: An optimal $O(1/K)$ nonergodic analysis.
J. Sci. Comput., 79:671–699, 2019.

References VI

Xudong Li, Defeng Sun, and Kim-Chuan Toh.
QSDPNAL: A two-phase augmented Lagrangian method for convex quadratic semidefinite programming.
Math. Program. Comput., 10:703–743, 2018.

Haihao Lu 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.

Renato DC Monteiro and Benar F Svaiter.
Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers.
SIAM J. Optim., 23(1):475–507, 2013.

Shoham Sabach and Marc Teboulle.
Faster Lagrangian-based methods in convex optimization.
SIAM J. Optim., 32(1):204–227, 2022.

References VII

Quoc Tran-Dinh and Yang Luo.
Halpern-type accelerated and splitting algorithms for monotone inclusions.
arXiv preprint arXiv:2110.08150, 2021.

Rainer Wittmann.
Approximation of fixed points of nonexpansive mappings.
Arch. Math., 58(5):486–491, 1992.

Yunhai Xiao, Liang Chen, and Donghui Li.
A generalized alternating direction method of multipliers with semi-proximal terms for convex composite conic programming.
Math. Program. Comput., 10:533–555, 2018.

Bo Yang, Xinyuan Zhao, Xudong Li, and Defeng Sun.
An accelerated proximal alternating direction method of multipliers for optimal decentralized control of uncertain systems.
arXiv preprint arXiv:2304.11037, 2023.

References VIII

Mingqiang Zhu and Tony Chan.
An efficient primal-dual hybrid gradient algorithm for total variation image restoration.
UCLA Cam Report, 34(2), 2008.

Guojun Zhang, Yancheng Yuan, and Defeng Sun.
An efficient HPR algorithm for the Wasserstein barycenter problem with $O(\text{Dim(P)}/\varepsilon)$ computational complexity.
arXiv preprint arXiv:2211.14881, 2022.