A dual-primal balanced augmented Lagrangian method for linearly constrained convex programming

Shengjie Xu¹,²

Received: 12 June 2022 / Revised: 28 July 2022 / Accepted: 2 August 2022 / Published online: 22 August 2022
© The Author(s) under exclusive licence to Korean Society for Informatics and Computational Applied Mathematics 2022

Abstract
Most recently, a balanced augmented Lagrangian method (ALM) has been proposed by He and Yuan for the canonical convex minimization problem with linear constraints, which advances the original ALM by balancing its subproblems, improving its implementation and enlarging its applicable range. In this paper, we propose a dual-primal version of the newly developed balanced ALM, which updates the new iterate via a conversely dual-primal iterative order formally. The new algorithm inherits all advantages of the prototype balanced ALM, and it can be extended to more general separable convex programming problems with both linear equality and inequality constraints. The convergence analysis of the proposed method can be well conducted in the context of variational inequalities. In particular, by some application problems, we numerically validate that these balanced ALM type methods can outperform existing algorithms of the same kind significantly.

Keywords Augmented Lagrangian method · Convex programming · Dual-primal · Proximal point algorithm · Variational inequality

Mathematics Subject Classification 90C25 · 90C06 · 65K05

1 Introduction

A basic optimization model is the canonical convex programming problem with linear equality constraints:

$$\min \left\{ \theta(x) \mid Ax = b, \ x \in \mathcal{X} \right\}, \quad (1)$$

¹ Shengjie Xu
xsjnsu@163.com

² Department of Mathematics, Harbin Institute of Technology, Harbin 150001, China

¹² Department of Mathematics, Southern University of Science and Technology, Shenzhen 518055, China
where $\theta : \mathbb{R}^n \to \mathbb{R}$ is a closed, proper and lower semicontinuous convex but not necessarily smooth function, $\mathcal{X} \subseteq \mathbb{R}^m$ is a closed convex set, $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. Among algorithms for solving (1), the augmented Lagrangian method (ALM) proposed in [1, 2] is classic and fundamental, and it plays a significant role in both theoretical study and algorithmic design for various convex programming problems. We refer to, e.g., [3–7], for some monographs about the ALM. In particular, it was shown in [8, 9] that the original ALM can be interpreted as an application of the proximal point algorithm (PPA) introduced in [10]. In practice, with given $\lambda^k \in \mathbb{R}^m$, the ALM generates the new iterate $(x^{k+1}, \lambda^{k+1})$ via the scheme

\[
\begin{align*}
  x^{k+1} &= \arg \min \{ L_\beta(x, \lambda^k) \mid x \in \mathcal{X} \}, \\
  \lambda^{k+1} &= \lambda^k - \beta(Ax^{k+1} - b),
\end{align*}
\]

where $\beta > 0$ is the penalty parameter for the linear constraints, $\lambda \in \mathbb{R}^m$ is the Lagrangian multiplier and

\[
L_\beta(x, \lambda) := \theta(x) - \lambda^T (Ax - b) + \frac{\beta}{2} \|Ax - b\|^2_2
\]

is the corresponding augmented Lagrangian function of (1). Throughout our discussion, the parameter $\beta$ is assumed to be fixed for simplification, and hereafter, we also call $x$ and $\lambda$ the primal and dual variables, respectively.

Ignoring some constant terms, it is easy to discern that the essential step for implementing the original ALM (2) equals to the minimization problem

\[
x^{k+1} = \arg \min \{ \theta(x) + \frac{r}{2} \|Ax - (b + \frac{1}{\beta} \lambda^k)\|^2_2 \mid x \in \mathcal{X} \}. \tag{3}
\]

Obviously, the solution set of (3) is essentially determined by the objective function $\theta$, the matrix $A$ and the domain $\mathcal{X}$ in (1). To improve the implementation of (2), the so-called linearized ALM has attracted a wide of attention in the literature, see, e.g., [11–13]. As discussed in [12], the linearized ALM for (1) can be stated as

\[
\begin{align*}
  x^{k+1} &= \arg \min \left\{ \theta(x) + \frac{r}{2} \|x - q^k\|^2_2 \mid x \in \mathcal{X} \right\}, \\
  \lambda^{k+1} &= \lambda^k - \beta(Ax^{k+1} - b), \tag{4a}
\end{align*}
\]

with

\[ q^k = x^k + \frac{1}{r} A^T [\lambda^k - \beta(Ax^k - b)] \cdot \]

Here, the parameters $r > 0$ and $\beta > 0$ need to satisfy the condition $r > \beta \rho (A^T A)$ to theoretically ensure the convergence of (4), where $\rho (\cdot)$ is the spectrum radius of a matrix. Note that the matrix $A$ is decoupled in (4a). The reshaped subproblem (4a) is thus easier to implement than (3). In particular, it reduces to the proximity operator of
\( \theta \) when \( \mathcal{X} = \mathbb{R}^n \), which generally has a closed-form solution or can be easily solved with a high precision for some special cases (e.g., \( \theta \) is a quadratic or norm function). We refer the readers to, e.g., [13–16], for these particular application scenarios arising in data science communities.

There is a structural restriction \( r > \beta \rho(A^T A) \) in the well-reshaped linearized ALM (4). For a fixed \( \beta > 0 \), it is clear that the quadratic term in (4a) will dominate the objective function of (4a) if \( \rho(A^T A) \) is too large. Accordingly, tiny step-size is inevitably generated and the convergence will be slowed down. To reduce such a restriction, it was proved in [11] that this restriction can be optimally improved to \( r > 0.75\beta \rho(A^T A) \) by using an indefinite proximal regularization technique, which allows a bigger step size and thus potentially accelerates the convergence. Most recently, a balanced version of the ALM has been presented in [12], which has no such a restriction and takes the following iterative scheme:

\[
\begin{align*}
    x^{k+1} &= \arg \min \left\{ \theta(x) + \frac{\beta}{2} \| x - (x^k + \frac{1}{\beta} A^T \lambda^k) \|_2^2 \mid x \in \mathcal{X} \right\}, \\
    \lambda^{k+1} &= \lambda^k - (\frac{1}{\beta} AA^T + \delta I_m)^{-1} [A(2x^{k+1} - x^k) - b],
\end{align*}
\]

where \( \beta > 0 \) and \( \delta > 0 \) are free parameters. Moreover, as discussed in [12], the parameter \( \delta > 0 \) is merely used to ensure the positive definiteness the underlying matrix theoretically, and it can be just fixed as a small value beforehand. It only needs to empirically and technically tune the parameter \( \beta \) when implementing (5). Clearly, compared with the original ALM (2), the balanced ALM (5) enjoys great advantages in mainly two fields: first, the primal subproblem (5a) is easier to implement; second, the free parameters \( \beta \) and \( \delta \) are irrelevant with \( \rho(A^T A) \). At the same time, we also need to discern that the dual subproblem (5b) becomes slightly complicated, because it requires to solve a linear system of equations. Fortunately, note that the coefficient matrix in (5b) is symmetric and positive definite. It can be computed easily, e.g., by the Cholesky decomposition.

The primary purpose of this paper is to present a dual-primal version of the balanced ALM (5) for the linearly constrained convex programming problem (1). More concretely, our new algorithm takes the following iterative scheme:

\[
\begin{align*}
    \lambda^{k+1} &= \lambda^k - (\frac{1}{\beta} AA^T + \delta I_m)^{-1} (Ax^k - b), \\
    \tilde{x}^k &= \arg \min \left\{ \theta(x) + \frac{\beta}{2} \| x - [x^k + \frac{1}{\beta} A^T (2\lambda^k - \lambda^k)] \|_2^2 \mid x \in \mathcal{X} \right\}, \\
    x^{k+1} &= x^k + \alpha(\tilde{x}^k - x^k), \\
    \lambda^{k+1} &= \lambda^k + \alpha(\tilde{\lambda}^k - \lambda^k),
\end{align*}
\]

where \( \beta > 0 \) and \( \delta > 0 \) are free parameters, and \( \alpha \in (0, 2) \) is the extrapolation parameter. As can be seen easily, the new algorithm (6) generates first the dual variable \( \lambda \), then the primal variable \( x \), and it maintains the same computational difficulty with the prototype balanced ALM (5). It is thus named the dual-primal balanced ALM.
throughout our discussion. In addition, the new algorithm (6) can be easily extended to
tackle more general separable convex programming problems with both linear equality
and inequality constraints. We will present a generalized dual-primal balanced ALM
for more general convex programming models in Sect. 4.

The rest of this paper is organized as follows. In Sect. 2, we summarize some
fundamental results for streamlining our analysis. In Sect. 3, we show the global
convergence of the proposed method upon its equivalent prediction–correction inter-
pretation, along with a worst-case $O(1/N)$ convergence rate. Moreover, we present
a generalized scheme for more general convex programming models in Sect. 4. The
numerical experiment is further conducted in Sect. 5, which is used to illustrate the
efficiency of the proposed method. Finally, some conclusions are made in Sect. 6.

2 Preliminaries

In this section, we summarize some preliminaries for further analysis. Let us first recall
a primary lemma, whose proof is elementary and can be found in, e.g., [17].

Lemma 2.1 Let $f : \mathbb{R}^l \to \mathbb{R}$ and $g : \mathbb{R}^l \to \mathbb{R}$ be convex functions, and $Z \subseteq \mathbb{R}^l$ be
a closed convex set. If $g$ is differentiable on an open set which contains $Z$ and the
solution set of the optimization problem $\min\{f(z) + g(z) \mid z \in Z\}$ is nonempty, then
we have

$$z^* \in \arg\min \{f(z) + g(z) \mid z \in Z\} \quad (7a)$$

if and only if

$$z^* \in Z, \quad f(z) - f(z^*) + (z - z^*)^T \nabla g(z^*) \geq 0, \quad \forall z \in Z. \quad (7b)$$

2.1 Variational inequality reformulation of (1)

Following the analogous techniques in, e.g., [12, 18–20], our analysis will be con-
ducted in the variational inequality (VI) context. Let us first write the VI reformulation
of the optimal condition for the studied model (1).

More specifically, let the Lagrangian function of (1) be defined as

$$L(x, \lambda) = \theta(x) - \lambda^T(Ax - b), \quad (8)$$

with $\lambda \in \mathbb{R}^m$ the Lagrangian multiplier. The pair $(x^*, \lambda^*) \in \mathcal{X} \times \mathbb{R}^m$ is called a saddle
point of (8) if it satisfies

$$L_{x \in \mathcal{X}}(x^*, \lambda) \leq L(x^*, \lambda^*) \leq L_{x \in \mathcal{X}}(x, \lambda^*). \quad (9)$$

It can be further characterized as

$$\begin{cases} x^* \in \arg\min\{L(x, \lambda^*) \mid x \in \mathcal{X}\}, \\ \lambda^* \in \arg\max\{L(x^*, \lambda) \mid \lambda \in \mathbb{R}^m\}. \end{cases}$$
According to Lemma 2.1, the above inequalities can be alternatively written as

\[
\begin{align*}
\begin{cases}
  x^* &\in \mathcal{X}, & \theta(x) - \theta(x^*) + (x - x^*)^T (-A^T \lambda^*) \geq 0, & \forall x \in \mathcal{X}, \\
  \lambda^* &\in \mathcal{R}^m, & (\lambda - \lambda^*)^T (Ax^* - b) \geq 0, & \forall \lambda \in \mathcal{R}^m,
\end{cases}
\end{align*}
\]

or more compactly,

\[
\text{VI}(F, \theta, \Omega) : \quad w^* \in \Omega, \quad \theta(x) - \theta(x^*) + (w - w^*)^T F(w^*) \geq 0, \quad \forall \ w \in \Omega, \quad (10a)
\]

where

\[
\begin{align*}
  w &= \begin{pmatrix} x \\ \lambda \end{pmatrix}, & \Omega &= \mathcal{X} \times \mathcal{R}^m \quad \text{and} \quad F(w) = \begin{pmatrix} -A^T \lambda \\ Ax - b \end{pmatrix}. \quad (10b)
\end{align*}
\]

Note that the operator \( F \) defined in (10b) is affine with a skew-symmetric matrix. It holds that

\[
(u - v)^T (F(u) - F(v)) = 0, \quad \forall \ u, \ v \in \mathcal{R}^{(n+m)}, \quad (11)
\]

which indicates that \( F \) is monotone. Throughout, we denote by \( \Omega^* \) the solution set of the VI (10), which is also the saddle point set of the Lagrangian function (8).

2.2 Prediction–correction interpretation of (6)

The meticulously-designed prediction–correction interpretation for a known algorithm is a powerful technique for streamlining its convergence analysis, and the related literature can be found in, e.g., [12, 18, 19]. To simplify the convergence analysis of the dual-primal balanced ALM (6), we also interpret it into a prediction–correction-type method as follows.

**Prediction step** With given \((x^k, \lambda^k)\), the dual-primal balanced ALM (6) begins with

\[
\begin{align*}
\tilde{x}^k &= \lambda^k - \left( \frac{1}{\beta} AA^T + \delta I_m \right)^{-1} (Ax^k - b), \quad (12a) \\
\tilde{x}^k &= \text{arg min} \left\{ \theta(x) + \frac{\beta}{2} \| x - \left[ x^k + \frac{1}{\beta} A^T (2\lambda^k - \lambda^k) \right] \| \right\}^2 \quad | \ x \in \mathcal{X} \right\}. \quad (12b)
\end{align*}
\]

**Correction step** Then, with \((\tilde{x}^k, \tilde{\lambda}^k)\) as a predictor, it further updates the new iterate \((x^{k+1}, \lambda^{k+1})\) via

\[
\begin{pmatrix} x^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} x^k \\ \lambda^k \end{pmatrix} - \alpha \begin{pmatrix} x^k - \tilde{x}^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}, \quad (13)
\]

where \(\alpha \in (0, 2)\) is the extrapolation parameter.
3 Convergence analysis

In this section, we establish the convergence analysis of the dual-primal balanced ALM (6) for the model (1), which is rooted in the prediction–correction interpretation (12)–(13). Let us first prove two pivotal lemmas.

Lemma 3.1 Let \( \tilde{w}^k = (\tilde{x}^k; \tilde{\lambda}^k) \) be the predictor generated by the prediction step (12) with given \( w^k = (x^k; \lambda^k) \). Then, we have

\[
\tilde{w}^k \in \Omega, \quad \theta(x) - \theta(\tilde{x}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \geq (w - \tilde{w}^k)^T H(w^k - \tilde{w}^k), \quad \forall w \in \Omega,
\]

with

\[
H = \begin{pmatrix}
\beta I_n & -A^T \\
-A & \frac{1}{\beta} A A^T + \delta I_m
\end{pmatrix}.
\]

Proof To begin with, for the dual subproblem (12a), we have

\[
Ax^k - b + \left( \frac{1}{\beta} A A^T + \delta I_m \right) (\tilde{x}^k - \lambda^k) = 0,
\]

which is also equivalent to

\[
(\lambda - \tilde{\lambda})^T \{ A\tilde{x}^k - b - A(\tilde{x}^k - x^k) + \left( \frac{1}{\beta} A A^T + \delta I_m \right) (\tilde{x}^k - \lambda^k) \} \geq 0, \quad \forall \lambda \in \mathbb{R}^m.
\]

For the primal subproblem (12b), it follows from Lemma 2.1 that

\[
\tilde{x}^k \in \mathcal{X}, \quad \theta(x) - \theta(\tilde{x}^k) + (x - \tilde{x}^k)^T A\tilde{x}^k + \beta(\tilde{x}^k - x^k) + \beta(\tilde{x}^k - x^k) \geq 0, \quad \forall x \in \mathcal{X},
\]

which can be further rewritten as

\[
\tilde{x}^k \in \mathcal{X}, \quad \theta(x) - \theta(\tilde{x}^k) + (x - \tilde{x}^k)^T - A^T (\tilde{x}^k - \lambda^k) + \beta(\tilde{x}^k - x^k) \geq 0, \quad \forall x \in \mathcal{X}.
\]

Adding (16) and (17), it implies that

\[
(\tilde{x}^k, \tilde{\lambda}^k) \in \mathcal{X} \times \mathbb{R}^m, \quad \theta(x) - \theta(\tilde{x}^k) + \left( \begin{pmatrix}
x - \tilde{x}^k \\
\lambda - \tilde{\lambda}^k
\end{pmatrix}
\right)^T \left\{ \begin{pmatrix}
-A^T \tilde{x}^k \\
A\tilde{x} - b
\end{pmatrix}
\right\} + \left( \begin{pmatrix}
\beta(\tilde{x}^k - x^k) - A^T (\tilde{\lambda}^k - \lambda^k) \\
-A(\tilde{x}^k - x^k) + \left( \frac{1}{\beta} A A^T + \delta I_m \right) (\tilde{\lambda}^k - \lambda^k)
\end{pmatrix}
\right) \geq 0, \quad \forall (x, \lambda) \in \mathcal{X} \times \mathbb{R}^m.
\]

Using the notations in (10b) and the matrix \( H \) defined in (15), the assertion of this lemma follows immediately. \( \square \)

Moreover, the positive definiteness of the induced matrix \( H \) defined in (15) can be ensured by the following proposition.

Proposition 1 The matrix \( H \) defined in (15) is positive definite for any \( \beta > 0 \) and \( \delta > 0 \).
Proof First of all, it is trivial to verify that
\[
H = \left( \begin{array}{cc}
\beta I_n & -A^T \\
-A & \frac{1}{\beta} AA^T + \delta I_m
\end{array} \right) = \left( \begin{array}{cc}
-\sqrt{\beta} I_n & \sqrt{\beta} A^T \\
\sqrt{\beta} A & \frac{1}{\beta} A^2 + \delta I_m
\end{array} \right) + \left( \begin{array}{cc}
0 & 0 \\
0 & \delta I_m
\end{array} \right).
\]
Then, for any \( w = (x; \lambda) \neq 0 \), we have
\[
w^T H w = \| \sqrt{\beta} A^T \lambda - \sqrt{\beta} x \|^2 + \delta \| \lambda \|^2 > 0,
\]
and the proof is complete accordingly.

The following lemma further refines the right-hand side of (14), and it is used to quantify the difference of \( \bar{w}^k \) from a solution point of the VI (10) by recursively quadratic terms.

Lemma 3.2 Let \( \{w^k\} \) and \( \{\bar{w}^k\} \) be the sequences generated by the prediction–correction scheme (12)–(13) with arbitrary \( \beta > 0 \) and \( \delta > 0 \). Then, for any \( \alpha \in (0, 2) \), we have
\[
\theta(x) - \theta(\bar{x}^k) + (w - \bar{w}^k)^T F(w) \\
\geq \frac{1}{2\alpha} \left\{ \| w - w^{k+1} \|_H^2 - \| w - w^k \|_H^2 + \alpha(2 - \alpha) \| w^k - \bar{w}^k \|_H^2 \right\}, \forall w \in \Omega. \tag{18}
\]
where \( H \) is the matrix given by (15).

Proof It follows from (14) and \( w^{k+1} = w^k - \alpha(w^k - \bar{w}^k) \) (see (13)) that
\[
\alpha \left\{ \theta(x) - \theta(\bar{x}^k) + (w - \bar{w}^k)^T F(\bar{w}^k) \right\} \geq (w - \bar{w}^k)^T H(w^k - w^{k+1}), \forall w \in \Omega. \tag{19}
\]
Applying the identity
\[
(a - b)^T H (c - d) = \frac{1}{2} \left\{ \| a - d \|_H^2 - \| a - c \|_H^2 \right\} + \frac{1}{2} \left\{ \| c - b \|_H^2 - \| d - b \|_H^2 \right\}
\]
to the right-hand side of (19) with \( a = w, b = \bar{w}^k, c = w^k \) and \( d = w^{k+1} \), it further implies that
\[
(w - \bar{w}^k)^T H(w^k - w^{k+1}) = \frac{1}{2} \left\{ \| w - w^{k+1} \|_H^2 - \| w - w^k \|_H^2 \right\} \\
+ \frac{1}{2} \left\{ \| w^k - \bar{w}^k \|_H^2 - \| w^{k+1} - \bar{w}^k \|_H^2 \right\}. \tag{20}
\]
For the second term of right-hand side of (20), it follows from (13) that
\[
\frac{1}{2} \left\{ \| w^k - \bar{w}^k \|_H^2 - \| w^{k+1} - \bar{w}^k \|_H^2 \right\}
= \frac{1}{2} \left\{ \| w^k - \bar{w}^k \|_H^2 - \| w^k - \alpha (w^k - \bar{w}^k) - \bar{w}^k \|_H^2 \right\}
= \frac{1}{2} \alpha (2 - \alpha) \| w^k - \bar{w}^k \|_H^2.
\]
(21)

Then, combining with (20) and (21), the inequality (19) equals to
\[
\alpha \left\{ \theta(x) - \theta(\bar{x}^k) + (w - \bar{w}^k)^T F(\bar{w}^k) \right\}
\geq \frac{1}{2} \left\{ \| w - w^{k+1} \|_H^2 - \| w - w^k \|_H^2 + \alpha (2 - \alpha) \| w^k - \bar{w}^k \|_H^2 \right\}, \quad \forall w \in \Omega.
\]

Note that \((w - \bar{w}^k)^T F(\bar{w}^k) \equiv (w - \bar{w}^k)^T F(w)\) (see (11)). The assertion of this lemma follows immediately.

Then, with the help of the above lemmas, the strict contraction of the sequence \(\{w^k\}\) generated by the new algorithm can be summarized in the following theorem.

**Theorem 2** Let \(\{w^k\}\) and \(\{\bar{w}^k\}\) be the sequences generated by the prediction–correction scheme (12)–(13) with arbitrary \(\beta > 0\) and \(\delta > 0\). Then, for any \(\alpha \in (0, 2)\), it holds that
\[
\| w^{k+1} - w^* \|_H^2 \leq \| w^k - w^* \|_H^2 - \alpha (2 - \alpha) \| w^k - \bar{w}^k \|_H^2, \quad \forall w^* \in \Omega^*.
\]
(22)

where \(H\) is the matrix defined in (15).

**Proof** Setting \(w\) in (18) as arbitrary \(w^* \in \Omega^*\), we have
\[
\| w^k - w^* \|_H^2 \leq \| w^k + 1 - w^* \|_H^2 - \alpha (2 - \alpha) \| w^k - \bar{w}^k \|_H^2
\geq 2 \alpha \left\{ \theta(\bar{x}^k) - \theta(x^k) + (\bar{w}^k - w^*)^T F(w^*) \right\}, \quad \forall w^* \in \Omega^*.
\]
(23)

Note that \(w^* \in \Omega^*\) and \(\bar{w}^k \in \Omega\). It follows from (10a) that the right-hand side of (23) is non-negative. This leads to the assertion of the theorem immediately. \(\square\)

Now we are ready to show the global convergence of the dual-primal balanced ALM (6), and it is based on the essential contraction property (22).

**Theorem 3** The sequence \(\{w^k\}\) generated by the dual-primal balanced ALM (6) converges to some \(w^\infty \in \Omega^*\) for any \(\beta > 0\), \(\delta > 0\) and \(\alpha \in (0, 2)\).

**Proof** To begin with, it follows from the inequality (22) that the sequence \(\{w^k\}\) is bounded. Summing (22) over \(k = 0, 1, \ldots, \infty\), we have
\[
\sum_{k=0}^{\infty} \alpha (2 - \alpha) \| w^k - \bar{w}^k \|_H^2 \leq \| w^0 - w^* \|_H^2.
\]
Therefore, we get

$$\lim_{k \to \infty} \|w^k - \tilde{w}^k\|^2_H = 0,$$

which means that the sequence \(\{\tilde{w}^k\}\) is also bounded. Let \(w^\infty\) be a cluster point of \(\{\tilde{w}^k\}\) and \(\{\tilde{w}^{kj}\}\) be a subsequence converging to \(w^\infty\). Then, according to \((14)\), we have \(\tilde{w}^{kj} \in \Omega\) such that

$$\theta(x) - \theta(\tilde{x}^{kj}) + (w - \tilde{w}^{kj})^T F(\tilde{w}^{kj}) \geq (w - \tilde{w}^{kj})^T H(w^{kj} - \tilde{w}^{kj}), \quad \forall \ w \in \Omega.$$ 

Note that the matrix \(H\) defined in \((15)\) is non-singular. It follows from \((24)\) and the lower semicontinuity of \(\theta\) that

$$w^\infty \in \Omega, \quad \theta(x) - \theta(x^\infty) + (w - w^\infty)^T F(w^\infty) \geq 0, \quad \forall \ w \in \Omega.$$ 

This VI characterization indicates that \(w^\infty\) is a solution point of VI \((10)\). On the one hand, according to \((22)\), the sequence \(\{\|w^k - w^\infty\|^2_H\}_{k \geq 0}\) is nonincreasing and it is bounded away below from zero. On the other hand, it follows from \((24)\) and \(\lim_{k \to \infty} \tilde{w}^{kj} = w^\infty\) that the subsequence \(\{w^{kj}\}\) also converges to \(w^\infty\). Therefore, the sequence \(\{w^k\}\) converges to \(w^\infty\) and the proof is complete. \(\square\)

**Remark 1** Following the similar analysis routine in, e.g., \([12, 19–21]\), it is easy to prove that the dual-primal balanced ALM \((6)\) also enjoys a worst-case \(O(1/N)\) convergence rate in both ergodic and non-ergodic senses, where \(N\) is the iteration counter. We opt to skip these trivial and meticulous proofs for succinctness.

### 4 Extensions to more general models

In this section, we extend the dual-primal balanced ALM \((6)\) to solve the following more general separable convex programming problem with linear equality or inequality constraints:

$$\min \sum_{i=1}^{p} \theta_i(x_i)$$

s.t. \(\sum_{i=1}^{p} A_i x_i = b\) (or \(\geq b\)),

$$x_i \in X_i, \quad i = 1, \ldots, p,$$

where \(\theta_i : R^{n_i} \to R\) \((i = 1, \ldots, p)\) are closed, proper and lower semicountinous convex but not necessarily smooth functions, \(X_i \subseteq R^{n_i}\) \((i = 1, \ldots, p)\) are closed convex sets, \(A_i \in R^{m \times n_i}\) \((i = 1, \ldots, p)\) and \(b \in R^m\). We refer to, e.g., \([22–26]\), for a number of applications that can be formed as the more general convex programming model \((25)\). To unify the notation, let us first define

$$\Lambda = \begin{cases} R^m, & \text{if } \sum_{i=1}^{p} A_i x_i = b, \\ R_+^m, & \text{if } \sum_{i=1}^{p} A_i x_i \geq b. \end{cases}$$

\((26)\)

It is clear that the basic model \((1)\) coincides with the case of \((25)\) where \(p = 1\) and \(\Lambda = R^m\).
4.1 Algorithm

With arbitrary constants $\beta_i > 0$ ($i = 1, \ldots, p$) and $\delta > 0$, let us first define

$$ M_p = \sum_{i=1}^{p} \frac{1}{\beta_i} A_i A_i^T + \delta I_m, \quad (27) $$

and

$$ q_i^k = x_i^k + \frac{1}{\beta_i} A_i^T (2\bar{\lambda}_i^k - \lambda_i^k), \quad i = 1, \ldots, p. \quad (28) $$

Then, a generalized version of the dual-primal balanced ALM (6) for the more general convex programming problem (25) is presented as follows.

**Algorithm: a generalized dual-primal balanced ALM for (25)**

(Prediction step) With given $(x_1^k, \ldots, x_p^k, \lambda^k)$, it first generates a predictor $(\bar{x}_1^k, \ldots, \bar{x}_p^k, \bar{\lambda}_k)$ via

$$ \bar{\lambda}_k = \arg \min_{\lambda \in \Lambda} \left\{ \frac{1}{2} (\lambda - \lambda^k)^T M_p (\lambda - \lambda^k) + \lambda^T \left( \sum_{i=1}^{p} A_i x_i^k - b \right) \right\}, \quad (29a) $$

$$ \bar{x}_i^k = \arg \min_{x_i \in X_i} \left\{ \theta_i(x_i) + \frac{\beta_i}{2} \| x_i - q_i^k \|_2^2 \right\}, \quad i = 1, \ldots, p, \quad (29b) $$

where $M_p$ and $q_i^k$ are defined in (27) and (28), respectively.

(Correction step) Then, it updates the new iterate $(x_1^{k+1}, \ldots, x_p^{k+1}, \lambda^{k+1})$ by the following correction step

$$ \begin{pmatrix} x_1^{k+1} \\ \vdots \\ x_p^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} x_1^k \\ \vdots \\ x_p^k \\ \lambda^k \end{pmatrix} - \alpha \begin{pmatrix} x_1^k - \bar{x}_1^k \\ \vdots \\ x_p^k - \bar{x}_p^k \\ \lambda^k - \bar{\lambda}_k \end{pmatrix}, \quad (29c) $$

where $\alpha \in (0, 2)$ is the extrapolation parameter.

**Remark 2** Note that the $\lambda$-subproblem in (6) is equivalent to the minimization problem

$$ \bar{\lambda}_k = \arg \min_{\lambda \in \Lambda} \left\{ \frac{1}{2} (\lambda - \lambda^k)^T \left[ \frac{1}{\beta} A A^T + \delta I_m \right] (\lambda - \lambda^k) + \lambda^T (A x^k - b) \mid \lambda \in \mathbb{R}^m \right\}. $$

The elementary dual-primal balanced ALM (6) is thus a special case of (29) with $p = 1$ and $\Lambda = \mathbb{R}^m$.

**Remark 3** When the inequality-constrained case of (25) is considered, the subproblem (29a) would reduce to a standard quadratic programming with non-negative sign
constraints:
\[
\min \left\{ \frac{1}{2} (\mathbf{\lambda} - \mathbf{\lambda}^k)^T M_p (\mathbf{\lambda} - \mathbf{\lambda}^k) + \mathbf{\lambda}^T \left( \sum_{i=1}^p A_i x_i^k - b \right) \mid \mathbf{\lambda} \in \mathbb{R}^m_+ \right\}.
\]

As discussed in [12], such a minimization problem can be efficiently solved by many well-known solvers such as conjugate gradient method and Lemke algorithm (see, e.g., [27, 28]).

4.2 VI reformulation of (25)

To simplify the analysis for the more general model (25), similar as Sect. 2, we also derive the optimal condition of (25) in the VI context. More specifically, let the Lagrangian function of (25) be defined as

\[
L(x_1, \ldots, x_p, \mathbf{\lambda}) = \sum_{i=1}^p \theta_i(x_i) - \mathbf{\lambda}^T (\sum_{i=1}^p A_i x_i - b),
\]

where \( \lambda \in \Lambda \) is the Lagrangian multiplier. It is trivial to verify that the optimal condition of (25) is equivalent to finding a saddle point \((x_1^*, \ldots, x_p^*, \lambda^*) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_p \times \Lambda\) of (30) such that

\[
\begin{align*}
\theta_1(x_1) - \theta_1(x_1^*) + (x_1 - x_1^*)^T (-A_1^T \lambda^*) & \geq 0, \quad \forall x_1 \in \mathcal{X}_1, \\
\vdots \\
\theta_p(x_p) - \theta_p(x_p^*) + (x_p - x_p^*)^T (-A_p^T \lambda^*) & \geq 0, \quad \forall x_p \in \mathcal{X}_p, \\
(\lambda - \lambda^*)^T (\sum_{i=1}^p A_i x_i^* - b) & \geq 0, \quad \forall \mathbf{\lambda} \in \Lambda,
\end{align*}
\]

which can be compactly rewritten as the following VI:

\[
\text{VI}(\Omega, F, \theta): \quad w^* \in \Omega, \quad \theta(x) - \theta(x^*) + (w - w^*)^T F(w^*) \geq 0, \quad \forall w \in \Omega, \quad (32a)
\]

where

\[
x = \begin{pmatrix} x_1 \\ \vdots \\ x_p \end{pmatrix}, \quad w = \begin{pmatrix} x_1 \\ \vdots \\ x_p \\ \mathbf{\lambda} \end{pmatrix}, \quad F(w) = \begin{pmatrix} -A_1^T \mathbf{\lambda} \\ \vdots \\ -A_p^T \mathbf{\lambda} \\ \sum_{i=1}^p A_i x_i^* - b \end{pmatrix}, \\
\theta(x) = \sum_{i=1}^p \theta_i(x_i) \quad \text{and} \quad \Omega = \mathcal{X}_1 \times \cdots \times \mathcal{X}_p \times \Lambda.
\]

Again, we denote by \( \Omega^* \) the solution set of the VI (32), which is also the saddle point set of the the Lagrangian function (30).
4.3 Convergence analysis for (29)

To establish the convergence analysis of the generalized dual-primal balanced ALM (29) for the more general model (25), reusing the same letters in Sect. 3, we only need to extend Lemma 3.1 to a more general case.

Lemma 4.1 Let $M_p$ be the matrix defined in (27) and $\tilde{w}^k = (\tilde{x}^k_1, \ldots, \tilde{x}^k_p, \tilde{\lambda}^k)$ be the predictor generated by the prediction step (29) with given $w^k = (x^k_1, \ldots, x^k_p, \lambda^k)$. Then, we have

$$\theta(x) - \theta(\tilde{x}^k) + (w - \tilde{w}^k)^T F(\tilde{w}^k) \geq (w - \tilde{w}^k)^T H(w^k - \tilde{w}^k), \quad \forall \ w \in \Omega,$$

where

$$H = \begin{pmatrix} \beta_1 I_{n_1} & 0 & \cdots & 0 & -A_1^T \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \beta_p I_{n_p} & -A_p^T \\ -A_1 & \cdots & \cdots & \cdots & -A_p \end{pmatrix}.$$

Proof First of all, for the subproblem (29a), it follows from Lemma 2.1 that

$$(\lambda - \tilde{\lambda}^k)^T \left\{ \sum_{i=1}^p A_i \tilde{x}^k_i - b + M_p(\tilde{\lambda}^k - \lambda^k) \right\} \geq 0, \quad \forall \ \lambda \in \Lambda,$$

which can be further rewritten as

$$(\lambda - \tilde{\lambda}^k)^T \left\{ \sum_{i=1}^p A_i \tilde{x}^k_i - b - \sum_{i=1}^p A_i (\tilde{x}^k_i - x^k_i) + M_p(\tilde{\lambda}^k - \lambda^k) \right\} \geq 0, \quad \forall \ \lambda \in \Lambda. \quad (35)$$

For each $x_i$-subproblem in (29b), it follows from Lemma 2.1 that $\tilde{x}^k_i \in X_i$ such that

$$\theta_i(x_i) - \theta_i(\tilde{x}^k_i) + (x_i - \tilde{x}^k_i)^T \left\{ -A_i^T (2\tilde{\lambda}^k - \lambda^k) + \beta (\tilde{x}^k_i - x^k_i) \right\} \geq 0, \quad \forall \ x_i \in X_i,$$

which also equals to

$$\tilde{x}^k_i \in X_i, \quad \theta_i(x_i) - \theta_i(\tilde{x}^k_i) + (x_i - \tilde{x}^k_i)^T \left\{ -A_i^T \tilde{\lambda}^k + \beta (\tilde{x}^k_i - x^k_i) - A_i^T (\tilde{\lambda}^k - \lambda^k) \right\} \geq 0, \quad \forall \ x_i \in X_i. \quad (36)$$

Adding (35) and (36), and using the notations defined in (32b) and the matrix $H$ defined in (34), the assertion of this lemma follows immediately.

Again, the positive definiteness of the induced matrix $H$ defined in (34) can be guaranteed by the following proposition.
Proposition 4 The matrix $H$ defined in (34) is positive definite for any $\delta > 0$ and $\beta_i > 0$ ($i = 1, \ldots, p$).

Proof To begin with, it is trivial to check that

$$H = \begin{pmatrix} \beta_1 I_{n_1} & 0 & \cdots & 0 & -A_1^T \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & \beta_p I_{n_p} & -A_p^T \\ -A_1 & \cdots & \cdots & -A_p & \sum_{i=1}^p \frac{1}{\beta_i} A_i A_i^T + \delta I_m \end{pmatrix}$$

Then, for arbitrary $w = (x_1; \ldots; x_p; \lambda) \neq 0$, we have

$$w^T H w = \sum_{i=1}^p \left( -\sqrt{\beta_i} I_{n_i} \right) \left( \cdots -\sqrt{\beta_i} I_{n_i} \cdots \sqrt{\frac{1}{\beta_i} A_i^T} \right) + \left( \begin{array}{cc} 0 & 0 \\ 0 & \delta I_m \end{array} \right)$$

and the proof is complete.

Beginning with Lemma 4.1 and using the same letters, the remaining proofs can be seamlessly repeated by the analysis in Sect. 3. The convergence of the generalized dual-primal balanced ALM (29) for (25) is thus established.

5 Numerical experiments

In this section, we report the numerical results of the dual-primal balanced ALM for some application problems. The preliminary experimental results show that the new algorithm has a same high efficiency with the original balanced ALM, and it can outperform existing algorithms of the same kind significantly. Our algorithms were written in a Python 3.9 and implemented in a personal computer with 2.20 GHz Intel Core i7-8750H CPU and 16 GB memory.

5.1 Equality-constrained $l_1$ minimization problem

5.1.1 Model and implementation

Let us first consider the classic equality-constrained $l_1$ minimization problem:

$$\min \{ \|x\|_1 \mid Ax = b, \ x \in \mathbb{R}^n \},$$

(37)
where $\|x\|_1 = \sum_{i=1}^n |x_i|$, $A \in \mathbb{R}^{m \times n}$ ($m < n$) and $b \in \mathbb{R}^m$. The model (37) is also known as the basis pursuit problem, and it plays a significant role in various areas such as compressed sensing and statistical learning. We see, e.g., [15, 29] for some survey papers.

Applying the dual-primal balanced ALM (6) (denoted by “DP-BALM”) to (37), the resulting scheme then reads as

$$
\begin{align*}
\bar{\lambda}^k &= \lambda^k - (\frac{1}{\beta} AA^T + \delta I_m)^{-1}(Ax^k - b), \\
\tilde{x}^k &= \arg \min \left\{ \|x\|_1 + \frac{\beta}{2} \|x - [\lambda^k + \frac{1}{\beta} A^T (2\bar{\lambda}^k - \lambda^k)]\|_2^2 \mid x \in \mathbb{R}^n \right\}, \\
x^{k+1} &= x^k + \alpha (\tilde{x}^k - x^k), \\
\lambda^{k+1} &= \lambda^k + \alpha (\bar{\lambda}^k - \lambda^k).
\end{align*}
$$

We fix $\alpha = 1$ in (38) for simplification. Clearly, the $x$-subproblem in (38) has a closed-form solution, which can be represented explicitly by the shrinkage operator defined in, e.g., [15]. At the same time, as a contrast, we also report the numerical results of the primal-dual algorithm proposed in [30] (denoted by “PDA”), the linearized ALM (4) (denoted by “LALM”) and the balanced ALM (5) (denoted by “BALM”). Their associated iterative schemes are trivial and thus skipped for succinctness.

5.1.2 Settings

To simulate, we follow the standard way (see, e.g., [31]) to generate a $x^* \in \mathbb{R}^n$ randomly whose $s$ entries are drawn from the normal distribution $\mathcal{N}(0, 1)$ and the rest are zeros. Then, we generate a standard Gaussian matrix $A \in \mathbb{R}^{m \times n}$ whose entries satisfying the normal distribution, and further set $b = Ax^*$. In our experiments, we take various settings of $m$ and corresponding sparse parameters $s$, and use $(x^0, \lambda^0) = (0, 0)$ as the initial iterate. According to the settings in [31], the stopping criterion for (37) is defined as

$$
\text{ReE}(k) := \frac{\|x^k - \bar{x}\|}{\|\bar{x}\|} < 10^{-11},
$$

where $\bar{x}$ is an approximate optimal solution numerically computed by the PDA with 10000 iterations, and “ReE” is short for the relative error. To implement the aforementioned algorithms efficiently, we take the specific parameter settings as following:

- **PDA**: $r = \sqrt{\rho(A^T A) + 0.001}$ and $s = \sqrt{\rho(A^T A) + 0.001}$;
- **LALM (4)**: $\beta = 0.001$ and $r = \beta \rho(A^T A) + 0.001$;
- **BALM (5)**: $\beta = 10$ and $\delta = 0.001$;
- **DP-BALM (6)**: $\beta = 10$, $\delta = 0.001$.

They are almost optimal for all the tested algorithms, selected out of a number of various values.
5.1.3 Numerical results

In Table 1, spectrum of the matrix $A^T A$ ("$\rho(A^T A)$"), required iteration numbers ("Iter") and totally computing time in seconds ("CPU(s)") are reported for various values of $n$, $m$ and $s$. It can be seen easily from Table 1 that the proposed method performs competitively with the prototype balanced ALM, and it has a significant acceleration compared with the PDA and the linearized ALM. To further visualize the numerical results, for the case $n = 10,000$, we plot the convergence curves versus both iteration numbers and CPU time for three various settings of $m$ and $s$ in Fig. 1, which can be further used to demonstrate the numerical efficiency of these BALM-type methods.

5.2 Exchange problem

5.2.1 Model and implementation

Let us also consider the exchange problem arising in economics, which aims at minimizing a function with a common objective among different agents. Moreover, as discussed in [22, 31], its mathematical form can be stated as

$$\min \left\{ \sum_{i=1}^{p} \theta_i(x_i) \mid \sum_{i=1}^{p} x_i = 0_n, x_i \in \mathbb{R}^n, i = 1, \ldots, p \right\},$$

(39)

where $\theta_i : \mathbb{R}^n \to \mathbb{R}$ ($i = 1, \ldots, p$) are cost functions corresponding to the agents. Clearly, the model (39) corresponds to the more general problem (25) with $A_i = I_n$ ($i = 1, \ldots, p$) and $b = 0_n$. Next, let us focus on a specific example of (39) as following:

$$\min \left\{ \frac{1}{2} \sum_{i=1}^{p} \|B_ix_i - c_i\|^2_{2} \mid \sum_{i=1}^{p} x_i = 0_n, x_i \in \mathbb{R}^n, i = 1, \ldots, p \right\},$$

(40)

where $B_i \in \mathbb{R}^{l \times n}$ ($l < n$) and $c_i \in \mathbb{R}^l$ are given matrices and vectors. We refer to [31] for more discussions on the model (40).

When the generalized dual-primal balanced ALM (29) (denoted by "G-DP-BALM") is applied to (40), it is easy to verify that the resulting scheme is

$$\begin{align*}
\bar{\lambda}_k &= \lambda_k - \frac{1}{\sum_{i=1}^{p} 1/\eta_i + \delta} \sum_{i=1}^{p} \lambda_i^k, \\
\bar{x}_i^k &= [\beta I_n + B_i^T B_i]^{-1}(B_i^T c_i + \beta_i x_i^k + 2\bar{x}_i^k - \lambda_i^k), \quad i = 1, \ldots, p, \\
x_i^{k+1} &= x_i^k + \alpha(x_i^k - x_i^k), \quad i = 1, \ldots, p, \\
\lambda^{k+1} &= \lambda^k + \alpha(\bar{\lambda}_i - \lambda_i^k).
\end{align*}$$

(41)

As can be seen easily, the $x_i$-subproblems in (41) can be treated in parallel. Hence, as a contrast, we also report the numerical results of some parallel splitting methods.
Table 1 Numerical results for (37) solved by the above mentioned algorithms

| n   | \(\rho(A^T A)\) | PDA     | LALM   | BALM   | DP-BALM |
|-----|------------------|---------|--------|--------|---------|
|     | Iter | CPU(s) | Iter | CPU(s) | Iter | CPU(s) | Iter | CPU(s) |
| 1000| 2875 | 654 | 0.19 | 1623 | 0.49 | 233 | 0.09 | 232 | 0.06 |
| 2000| 5784 | 736 | 1.14 | 1337 | 3.20 | 256 | 0.54 | 254 | 0.54 |
| 3000| 8593 | 776 | 3.17 | 759 | 4.80 | 202 | 1.06 | 201 | 1.06 |
| 5000| 14,464 | 811 | 9.90 | 729 | 13.09 | 194 | 2.97 | 193 | 3.06 |
| 8000| 23,319 | 838 | 26.26 | 737 | 34.32 | 212 | 8.25 | 203 | 8.23 |
| 10,000| 29,150 | 845 | 40.94 | 745 | 54.25 | 260 | 15.89 | 259 | 15.82 |
| 20,000| 58,161 | 931 | 177.69 | 768 | 224.11 | 432 | 104.00 | 432 | 103.78 |
| 30,000| 87,271 | 975 | 406.77 | 774 | 482.81 | 488 | 252.85 | 488 | 252.66 |

The associated convergence curves on some examples are plotted in Fig. 1 of the same kind including the modified Jacobian splitting ALM proposed in [32] (denoted by “JSALM”), the proximal Jacobian splitting ALM introduced in [31, 33] (denoted by “PJALM”) and the generalized balanced ALM proposed in [12] (denoted by “G-BALM”). Again, we fix \(\alpha = 1\) in (41) for simplification.
Fig. 1 Convergence curves for the basis pursuit problem (37) with $n = 10,000$, solved by the linearized ALM, the PDA, the BALM and the DP-BALM.

5.2.2 Settings

To simulate, we generate $x^*_i \in \mathbb{R}^n \ (i = 1, \ldots, p - 1)$ and $\lambda^* \in \mathbb{R}^n$ randomly whose entries satisfy the normal distribution $\mathcal{N}(0, 1)$, and $B_i \ (i = 1, \ldots, p)$ are random Gaussian matrices. To satisfy the Karush-Kuhn-Tucker (KKT) condition, we set $x^*_p = -\sum_{i=1}^{p-1} x^*_i$ and $c_i = (B_i B_i^T)^{-1} (B_i B_i^T B_i x^*_i - B_i \lambda^*) = B_i x^*_i - (B_i B_i^T)^{-1} B_i \lambda^* \ (i = 1, \ldots, p)$.
1, \ldots, p). Accordingly, the stopping criterion for (40) can be specified as the following KKT error (denoted by “KKTE”):

$$\text{KKTE}(k) := \max \left\{ \| B_i^T (B_i x_i^k - c_i) - \lambda^k \|, i = 1, \ldots, p, \sum_{i=1}^{p} x_i^k \right\} < 10^{-3}.$$  \hspace{1cm} (42)

Moreover, we choose \( n = 80 \) and \( l = 50 \) in (40), and take \( x_i^0 = 0 \) (\( i = 1, \ldots, p \)) and \( \lambda^0 = 0 \) as the initial point for all the algorithms tested. To implement the aforementioned algorithms efficiently, we take the specific parameter settings as following:

- JSALM: \( \beta = 1.0 \) and \( \alpha = 2(1 - \sqrt{p}/(p + 1)) \);
- PJALM: \( \beta = 0.1 \) and \( \tau = p - 1 \);
- G-BALM: \( \beta = 10 \) and \( \delta = 0.001 \);
- G-DP-BALM (41): \( \beta = 10 \) and \( \delta = 0.001 \).

5.2.3 Numerical results

In Table 2, iteration numbers (“Iter”), computing time in seconds (“CPU(s)”) are reported for different values of \( p \). As can be seen easily, the new algorithm (41) and the G-BALM outperform the other two parallel splitting algorithms very significantly when \( p \) is large; the proposed G-DP-BALM (41) has the same high efficiency with the prototypical G-BALM for the multiple-block separable exchange problem (40). Moreover, it is interesting to discern that the performance of the G-DP-BALM (41) and G-BALM is very stable with respect to various values of \( p \), which makes these BALM-type methods very attractive to the case (25) with large \( p \). Again, to further visualize the numerical results, in Fig. 2, we plot the convergence curves with first 300 iterates for the cases \( p = 500 \) and \( p = 1000 \), which further validate the numerical efficiency of these BALM-type algorithms.

| \( p \) | JSALM \( \text{Iter} \) \( \text{CPU(s)} \) | PJALM \( \text{Iter} \) \( \text{CPU(s)} \) | G-BALM \( \text{Iter} \) \( \text{CPU(s)} \) | G-DP-BALM \( \text{Iter} \) \( \text{CPU(s)} \) |
|-------|-------------------------|-------------------------|-------------------------|-------------------------|
| 100   | 2603 30.77               | 38 0.48                 | 38 0.45                 | 39 0.49                 |
| 200   | 5482 140.44              | 72 1.91                 | 40 1.05                 | 39 1.04                 |
| 300   | 8466 359.32              | 112 4.81                | 38 1.75                 | 42 1.81                 |
| 400   | 11,518 706.25            | 162 10.03               | 42 2.89                 | 45 2.85                 |
| 500   | 14,621 1197.74           | 190 16.11               | 44 3.92                 | 40 3.51                 |
| 600   | 17,764 1764.08           | 269 26.94               | 49 5.21                 | 48 4.96                 |
| 700   | 20,940 2443.07           | 316 37.17               | 51 6.44                 | 46 5.65                 |
| 800   | 24,144 3236.73           | 327 44.16               | 43 5.79                 | 46 6.62                 |
| 900   | 27,375 4106.01           | 366 54.94               | 42 7.00                 | 47 7.26                 |
| 1000  | 30,627 5127.66           | 361 60.23               | 41 7.55                 | 41 7.20                 |
6 Conclusions

In this paper, we present a dual-primal variant of the newly developed balanced ALM for the canonical convex programming problem with linear equality constraints. The new algorithm uses a conversely dual-primal iterative order compared with the prototypical balanced ALM; it can be also generalized to tackle more general convex programming problems with both linear equality and inequality constraints. By some numerical experiments, we show that the new algorithm enjoys an almost same high efficiency with the original balanced ALM, and that these balanced ALM type methods can outperform existing algorithms of the same kind numerically. This work can significantly enhance the rich literature for the original ALM and particularly the most recent balanced ALM.

Acknowledgements The author is greatly indebted to Professor Bingsheng He, Nanjing University, China, for numerous enlightening discussions and his helpful suggestions.

Data availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.
Declarations

Conflict of interest The author declares no competing interests.

References

1. Hestenes, M.R.: Multiplier and gradient methods. J. Optim. Theory Appli. 4(5), 303–320 (1969). https://doi.org/10.1007/BF0027673
2. Powell, M.J.: A method for nonlinear constraints in minimization problems. In: Fletcher, R. (ed.) Optimization, pp. 283–298. Academic Press, New York (1969)
3. Bertsekas, D.P.: Constrained Optimization and Lagrange Multiplier Methods. Academic Press, New York (1982)
4. Birgin, E.G., Martínez, J.M.: Practical Augmented Lagrangian Methods for Constrained Optimization. SIAM, Philadelphia (2014)
5. Fortin, M., Glowinski, R.: Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems. North-Holland, Amsterdam (1983)
6. Glowinski, R., Le Tallec, P.: Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics. SIAM, Philadelphia (1989)
7. Ito, K., Kunisch, K.: Lagrange Multiplier Approach to Variational Problems and Applications. SIAM, Philadelphia (2008)
8. Rockafellar, R.T.: Monotone operators and the proximal point algorithm. SIAM J. Control Optim. 14(5), 877–898 (1976). https://doi.org/10.1137/0314056
9. Rockafellar, R.T.: Augmented Lagrangians and applications of the proximal point algorithm in convex programming. Math. Oper. Res. 1(2), 97–116 (1976). https://doi.org/10.1287/moor.1.2.97
10. Martinet, B.: Régularisation d’inéquations variationnelles par approximations successives. Rev. Française Informat. Recherche Opérationnelle. 4, 154–158 (1970)
11. He, B., Ma, F., Yuan, X.: Optimal proximal augmented Lagrangian method and its application to full Jacobian splitting for multi-block separable convex minimization problems. IMA J. Num. Anal. 40(2), 1188–1216 (2020)
12. He, B., Yuan, X.: Balanced augmented Lagrangian method for convex programming. arXiv preprint. arXiv:2108.08554 (2021)
13. Yang, J., Yuan, X.: Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization. Math. Comp. 82(281), 301–329 (2013). https://doi.org/10.1090/S0025-5718-2012-02598-1
14. Candes, E.J., Recht, B.: Exact matrix completion via convex optimization. Found. Comput. Math. 9(6), 717–772 (2009). https://doi.org/10.1007/s10208-009-9045-5
15. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM Rev. 43(1), 129–159 (2001). https://doi.org/10.1137/S003614450037906X
16. Parikh, N., Boyd, S.: Proximal algorithms. Found. Trends Optim. 1(3), 127–239 (2014). https://doi.org/10.1561/2400000003
17. Beck, A.: First-Order Methods in Optimization. SIAM, Philadelphia (2017)
18. Gu, G., He, B., Yuan, X.: Customized proximal point algorithms for linearly constrained convex minimization and saddle-point problems: a unified approach. Comput. Optim. Appl. 59(1), 135–161 (2014). https://doi.org/10.1007/s10589-013-9616-x
19. He, B., Yuan, X.: Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective. SIAM J. Imaging Sci. 5(1), 119–149 (2012). https://doi.org/10.1137/100814494
20. He, B., Yuan, X.: On the $O(1/n)$ convergence rate of the Douglas–Rachford alternating direction method. SIAM J. Numer. Anal. 50(2), 700–709 (2012). https://doi.org/10.1137/110836936
21. He, B., Yuan, X.: On non-ergodic convergence rate of Douglas–Rachford alternating direction method of multipliers. Numer. Math. 130(3), 567–577 (2015). https://doi.org/10.1007/s00211-014-0673-6
22. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. 3(1), 1–122 (2010). https://doi.org/10.1561/2200000016
23. Chandrasekaran, V., Parrilo, P.A., Willsky, A.S.: Latent variable graphical model selection via convex optimization. Ann. Stat. 40(4), 1935–1967 (2012). https://doi.org/10.1214/11-AOS949
24. McLachlan, G.J.: Discriminant Analysis and Statistical Pattern Recognition. Wiley Interscience, New York (2004)
25. Sun, H., Tai, X.-C., Yuan, J.: Efficient and convergent preconditioned ADMM for the Potts models. SIAM J. Sci. Comput. 43(2), 455–478 (2021). https://doi.org/10.1137/20M1343956
26. Yuan, J., Bae, E., Tai, X.-C., Boykov, Y.: A continuous max-flow approach to Potts model. In: Computer Vision-ECCV 2010, pp. 379–392. Springer, New York (2010)
27. Golub, G.H., Van Loan, C.F.: Matrix Computations. Johns Hopkins University Press, Baltimore (1996)
28. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, New York (2006)
29. Bruckstein, A.M., Donoho, D.L., Elad, M.: From sparse solutions of systems of equations to sparse modeling of signals and images. SIAM Rev. 51(1), 34–81 (2009). https://doi.org/10.1137/060657704
30. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imaging Vis. 40(1), 120–145 (2011). https://doi.org/10.1007/s10851-010-0251-1
31. Deng, W., Lai, M.-J., Peng, Z., Yin, W.: Parallel multi-block ADMM with o(1/k) convergence. J. Sci. Comput. 71(2), 712–736 (2017). https://doi.org/10.1007/s10915-016-0318-2
32. He, B., Hou, L., Yuan, X.: On full Jacobian decomposition of the augmented Lagrangian method for separable convex programming. SIAM J. Optim. 25(4), 2274–2312 (2015). https://doi.org/10.1137/130922793
33. He, B., Xu, H.-K., Yuan, X.: On the proximal Jacobian decomposition of ALM for multiple-block separable convex minimization problems and its relationship to ADMM. J. Sci. Comput. 66(3), 1204–1217 (2016). https://doi.org/10.1007/s10915-015-0060-1

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.