FIXED POINT METHODS AND OPTIMIZATION
Volume 2 (2025), No. 1, 38–55
https://doi.org/10.69829/fpmo-025-0201-ta03
Tulipa Opera Scholarum

TWO MODIFIED SELF-ADAPTIVE DUAL ASCENT METHODS WITH LOGARITHMIC-QUADRATIC PROXIMAL REGULARIZATION FOR LINEARLY CONSTRAINED QUADRATIC CONVEX OPTIMIZATION

YUAN SHEN[1], MUYUN XU[1], CHANG LIU[2,*], AND ZAIYUN PENG[3]

[1] *School of Applied Mathematics, Nanjing University of Finance & Economics, Nanjing, 210023, P.R.China*
[2] *Department of Accounting, Nanjing Vocational College of Finance and Economics, Nanjing, 210001, P.R.China*
[3] *School of Mathematics and Statistics, Chongqing JiaoTong University, Chongqing, 400074, P.R.China*

Abstract. Dual ascent method (DAM) is an effective algorithm to handle a class of convex optimization problems with linear constraint. For problems with non-negative orthant constraints, logarithmic quadratic proximal (LQP) method can solve well by transforming the sub-problems into nonlinear equations. The LQP term is applied to regularize the subproblems of DAM in this article, so a DAM-LQP method is developed for solving both linearly constrained and non-negative constrained optimization problems, and further extend the proposed method to solve separable convex optimization problem with two blocks. When the objective function is quadratic, the convergence of proposed methods can be guaranteed better; also, we can solve the subproblems of the convex optimization problem parallelly when parallel computation devices are available, thus the computation time in one iteration could be greatly reduced. For the sake of demonstrating the efficiency of proposed methods, numerical results are proposed to verify.

Keywords. Quadratic optimization with linear constraints, Logarithmic-quadratic proximal regularization, Dual ascent method, Global convergence.

1. Introduction

We consider a class of convex optimization problem with linearly constraints as follows:

$$\min\{f(w)|Aw = b, w \in \mathcal{X}\}, \tag{1.1}$$

in which $A \in \mathbb{R}^{m\times n}$, $f(w)$ is a quadratic function and the feasible domain $\mathcal{X} \subseteq \mathbb{R}^n$ is a non-empty closed convex set. Without loss of generality, we define $f(w) = \frac{1}{2}w^T Mw + q^T w$, then $\nabla f(w) = Mw + q$. Ω is denoted as $\Omega = \mathcal{X} \times \mathbb{R}^m$.

In this paper, we think about a kind of problem with a special background, that is the dedicated algorithm with convex quadratic objective function. We always assume that the quadratic term matrix of the function $f(w)$ is positive definite and thus is strongly convex. For some cases where the objective function is quadratic, there are some important applications such that sensitivity analysis of separable traffic equilibrium in [16], alternating direction method for multi-block convex optimization in [19] and so on. For the purpose of explaining the assumption of strong convexity on $f(w)$ rational, a good example is the application of the linearized Bregman scheme in [6] and it has wide range of applications in [9][10].

Let the Lagrange function of (1.1) be

$$L(w, z) = f(w) - z^T(Aw - b), \tag{1.2}$$

*Corresponding author.
E-mail address: ocsiban@126.com (Yuan Shen). 3013498724@qq.com (Muyun Xu), Moirai0613@163.com (Chang Liu), pengzaiyun@126.com (Zaiyun Peng).

2020 Mathematics Subject Classification: 65K10, 90C30.
Accepted: February 23, 2025.

where $z \in \mathbb{R}^m$ is the Lagrange multiplier, and corresponding dual function of (1.1) is as follows:

$$G(z) := \min_{w \in \mathcal{X}} L(w, z).$$

It is noticed that the minimizer of $G(z)$ exists owing to the strong convexity of $f(w)$, then the dual problem of (1.1) is:

$$\max_{z \in \mathbb{R}^m} G(z).$$

A point $(w^*, z^*) \in \Omega^*$ satisfying

$$w^* = \arg\min_{w \in \mathcal{X}} L(w, z^*) \quad \text{and} \quad z^* = \arg\max_{z \in \mathbb{R}^m} L(w^*, z), \text{ simutaneously}$$

or equivalently

$$\begin{pmatrix} w - w^* \\ z - z^* \end{pmatrix}^T \begin{pmatrix} \nabla f(w^*) - A^T z^* \\ Aw^* - b \end{pmatrix} \geq 0, \quad \forall w \in \mathcal{X}, z \in \mathbb{R}^m \tag{1.3}$$

is called a saddle point of $L(w, z)$. Ω^* is denoted as the set of all (w^*, z^*).

One standard way is the classical augmented Lagrange method (ALM) [12] which minimizes the following augmented Lagrange function:

$$L_A(w, z) = f(w) - z^T(Aw - b) + \frac{\beta}{2}||Aw - b||^2. \tag{1.4}$$

Then the ALM procedure for solving (1.1) can be described as follows:

$$\begin{cases} w^{k+1} = \arg\min_{w \in \mathcal{X}} L_A(w, z^k), \\ z^{k+1} = z^k - \gamma\beta(Aw^{k+1} - b), \end{cases} \tag{1.5}$$

with $\beta > 0$ being a penalty parameter and $\gamma \in (0, 2)$ being a relaxation factor. When the subproblems in (1.5) have a closed-form solution, the implementation of ALM could be of low cost. However, due to the presence of quadratic term $\frac{\beta}{2}||Aw - b||^2$, even for the separable $f(w)$ like that in LASSO [6], the augmented Lagrange function $L_A(w, z)$ is still inseparable, thus the subproblems in (1.5) may be more computationally expensive and the dimensional scalability of numerical performance could be unsatisfactory. Moreover, as the penalty parameter β is critical to the performance of ALM, it should be manually tuned and its setting must fall into a suitable interval, in the case, we can obtain the satisfactory speed performance of ALM. Otherwise, ALM may converge extremely slow. However, finding the optimal setting of β could not be easy and this is especially true for practical problems.

With the increasing dimension of subproblems, Uzawa method [1] has become a more populor method. Using Uzawa method to solve (1.1), we can get the following iterative scheme:

$$\begin{cases} z^{k+1} = \arg\max_{z \in \mathbb{R}^m} \{L(w^k, z) - \frac{1}{2\beta}||z - z^k||^2\}, \\ w^{k+1} = \arg\min_{w \in \mathcal{X}} L(w, z^{k+1}), \end{cases} \tag{1.6}$$

where $\beta > 0$ represents the iteration step size, $|| \cdot ||$ represents a general Euclidean norm. In order to ensure the convergence of the Uzawa method, the step size β should be chosen restrictively. In the literature, a lot of algorithms are related to the Uzawa method, e.g. [21], [11][13], [15] and so on. We Some can also refer to [7][8] for some applications.

A similar approach is the dual ascent method (DAM), which solves the dual problem of maximizing $G(z)$ by applying an iterative ascent method. The basic framework of DAM is as follows:

$$\begin{cases} z^{k+1} = z^k - \beta_k(Aw^k - b), \\ w^{k+1} = \arg\min_{w \in \mathcal{X}} L(w, z^{k+1}), \end{cases} \tag{1.7}$$

in which β_k is a self-adaptive step size. In the literature, many choices for the ascent methods include (single) coordinate ascent in [3], gradient ascent in [24] and gradient projection in [17] have been taken. The subproblems in (1.7) can be solved parallelly when parallel computation devices are available, so the computation time in one iteration could be greatly reduced, thus it is more suitable for solving large-dimension problems. Tseng .et al[20] have shown that a number of dual ascent methods, including dual coordinate ascent methods, certain dual gradient methods and a dual gradient projection algorithm, converge at least linearly. However, only if the stepsize is small enough, the iterative sequence generated by (1.7) is convergent, thus the convergence of DAM is usually slow.

In order to improve the efficiency of DAM in solving (1.1) and relax the step size condition, Zhang et al. proposed a new dual ascent method (NDAM) which apply gradient projection to solving the dual problem [27]. Based on the Uzawa method, Tao and Yuan proposed an inexact Uzawa method [23]. In a parallel work, a modified dual ascent method (MDAM) [22] has been proposed, which is an improvement of NDAM. Compared with NDAM or the Uzawa method, MDAM further relax the step size condition whereas the convergence result can still be guaranteed, thus potentially yields faster convergence speed. We can get the new iteration point by the following iteration scheme:

$$\begin{cases} z^{k+1} = z^k - \beta_k(Aw^k - b), \\ w^{k+1} = \arg\min_{w\in\mathcal{X}} L(w, z^{k+1}), \end{cases} \tag{1.8}$$

where β_k is an adaptive step size. However, MDAM can only solve the problem without additional constraints. When with non-negative orthant constraints, MDAM is unable to handle it properly. On this basis, it is suggested that the LQP terms is applied for regularizing the subproblems of DAM, then put forward a modified self-adaptive LQP-DAM method.

It is noticed that the LQP method was firstly proposed by Auslender [2], the non-negative orthant constraint $w \geq 0$ is penalized into the objective function, then the subproblems turn out to be unconstrained nonlinearly equations. A LQP-based decomposition method is developed by Yuan and Li [26], in which LQP terms are applied to the subproblems of ADMM. Later, a new inexact LQP method is presented by Bnouhachem et al. (see [5][4]). Recently, an prediction-correction method based on LQP is presented by Li [18]. As a result of the interior point property of LQP algorithm, we decompose the subproblems into a system of nonlinear equations, which can be handled by a bunch of algorithms efficiently, for example, Quasi-Newton algorithm. On these bases, we hope to reduce the computational costs.

In this paper, we put forward a modified self-adaptive DAM with LQP regularization (DAMLQP) for solving (1.1). Compared with DAM or the Uzawa method, the step size is further relaxed, the convergence is also guaranteed. Moreover, the subproblems can be solved in parallel when parallel computation devices are available, thus the computation time in one iteration could be greatly reduced. Furthermore, we extend the DAMLQP to two blocks and the convergence could be also proved. In addition, we show that proposed algorithms have satisfactory numerical behaviors via the numerical experiment.

The remaining part of this paper is presented as follows. In Section 2, some preliminaries will be given. In Section 3, two proposed algorithms are put forward to solving problems (1.1) and (3.29), respectively. The convergence qualities are established in Section 3 to prove the convergence of our algorithms. In Section 4, we will present some experimental results compared with some state-of-the-art algorithms. Some conclusions will be drawn in the final section.

2. Preliminaries

Some basic knowledge about the LQP regularization is put forward, which will be useful in the following discussion in this section. More details also can be found in [2], which is necessary for the analysis of convergence in Section 3.

2.1. **The logarithmic-quadratic proximal regularization.** For any $s \in \mathbb{R}^n_+$, the LQP regularization term is defined as follows:

$$d(k,s) := \begin{cases} \sum_{i=1}^n [\frac{1}{2}(k_i - s_i)^2 + \mu(s_i^2 \log \frac{s_i}{k_i} + k_i s_i - s_i^2)], & \text{if} \quad k \in \mathbb{R}^n_+, \\ \infty, & \text{otherwise}, \end{cases} \tag{2.1}$$

where $\mu \geq 0$ is a weighting parameter. Based on this definition, we get

$$\nabla_k d(k,s) = (k - s) + \mu(s - S^2 k^{-1}),$$

where S := diag $(s_1, s_2, \ldots, s_n) \in \mathbb{R}^{n \times n}$, $k^{-1} \in \mathbb{R}^n$ is a vector and the i-th element of $\frac{1}{k}$ is $\frac{1}{k_i}$.

For the convergence analysis in further parts, we summarize the following lemma:

Lemma 1. [25] A positive definite diagonal matrix is defined as $Q := diag(q_1, q_2, \ldots, q_n) \in \mathbb{R}^{n \times n}$, p is a monotone mapping and $\vartheta : \mathbb{R}^n \to \mathbb{R}$. We define μ is a given positive constant. For given $\overline{s}, s$, there are $\overline{S} := diag(\overline{s}_1, \overline{s}_2, \ldots, \overline{s}_n),\ s^{-1} := (\frac{1}{s_1}, \ldots, \frac{1}{s_n})^T$ and

$$\Phi'(\overline{s}, s) := (s - \overline{s}) + \mu(\overline{s} - \overline{S}^2 s^{-1}).$$

Therefore, the variational inequality

$$\vartheta(k) - \vartheta(s) + (k - s)^T [p(s) + Q\Phi'(\overline{s}, s)] \geq 0,\ \forall k \in \mathbb{R}^n_+,$$

has an unique positive solution s. Moreover, we have

$$\vartheta(k) - \vartheta(s) + (k - s)^T p(s) \geq (1 + \mu)(\overline{s} - s)^T Q(k - s) - \mu||\overline{s} - s||_p^2$$

for this positive solution s $\in \mathbb{R}^n_+$ and any $k \in \mathbb{R}^n_+$.

As discussed in [14], solving (1.1) is equivalent to finding $t^* = (w^*, z^*) \in \Omega := \mathbb{R}^n_+ \times \mathbb{R}^m_+$ such that

$$VI(\Omega, F) : (t - t^*)^T F(t^*) \geq 0,\ \forall t \in \Omega, \tag{2.2}$$

in which

$$t = \begin{pmatrix} w \\ z \end{pmatrix} \text{ and } F(t) = \begin{pmatrix} M & -A^T \\ A & 0 \end{pmatrix}^T \begin{pmatrix} w \\ z \end{pmatrix} + \begin{pmatrix} p \\ -b \end{pmatrix}. \tag{2.3}$$

The mapping $F(t)$ defined in (2.3)is monotone. Ω^* is denoted as the solution set of $VI(\Omega, F, \theta)$, which is nonempty.

3. Two Modified Self-Adaptive Dual Ascent Methods with LQP Regularization

In this section, our proposed methods are described firstly, after that, we present the convergence analysis, the first algorithm is described as in Algorithm 1.

Remark 3.1. (1) In ALM, we need to set a proper fixed β to optimize its performance. In customized proximal point algorithm (CPPA for short), we need to manually tune the setting of r and s. In contrary, the step size β_k is self-adaptive and can be initialized with arbitrary value in our algorithm. Thus, compared with ALM and CPPA, our algorithm is easier to implement.
(2) Compared with CPPA, $||A^T A||$ could be too expensive to compute. Moreover, in some practical applications, the matrix A is not available so that the computation time could be long while in some applications, $||A^T A||$ may be easy to compute. However, our algorithm does not need to divide these cases so that its computing time is competitive compared to CPPA.

3.1. **The modified self-adaptive dual ascent method with LQP regularization.**

Algorithm 1: A modified self-adaptive dual ascent method with LQP regularization for (1.1).
Step 0. Initial $w^0 = \arg\min_{w\in\mathcal{X}} L(w, z^0)$, initial step size β_0, lower bound of step size $\beta_{\min}$, parameters $\mu, v \in (0,2)$ and $\mu < v$, $k = 0$, $\eta \in (0,1)$, tolerance ϵ.
Step 1. Find $(\tilde{z}^k, \tilde{w}^k, r_k)$ such that

$$\begin{cases} \tilde{z}^k = z^k - \beta_k(Aw^k - b), \\ \tilde{w}^k = \arg\min_{w\in\mathcal{X}}\{L(w, \tilde{z}^k) + rd(w, w^k)\}, \end{cases} \tag{3.1}$$

$$r_k = \frac{\beta_k(z^k - \tilde{z}^k)^T(w^k - \tilde{w}^k)}{||z^k - \tilde{z}^k||^2}, \tag{3.2}$$

in which $d(w, w^k)$ is defined in (2.1), $r > 0$ is a proximal parameter.
Step 2.
(a) If $r_k > v$, step size β_k is too large. $\beta_k := \eta * \beta_k * \min\{1, \frac{1}{r_k}\}$ and go to Step 1.
(b) If $r_k < v$, step size β_k is too small. β_{k+1}:=max$\{\beta_{\min}, \eta * \beta_k * \frac{1}{r_k}\}$ for the next iteration and go to Step 3.
(c) If $\mu < r_k < v$, step size β_k is proper. β_{k+1}:=max$\{\beta_{\min}, \beta_k\}$ and go to Step 3.
Step 3. Let $\{w^{k+1}, z^{k+1}\} = \{\tilde{w}^k, \tilde{z}^k\}$. If the stopping criterion $\frac{||w^{k+1}-w^k||}{||w^k||} \le \epsilon$ is met or a maximal iteration number is attained, stop the algorithm, or else, $k = k+1$, go to Step 1.

Lemma 2. If $\mathcal{X} = \mathbb{R}^n$, let the iterative sequence $\{t^k\} = \{(w^k, z^k)\} \in \Omega$ generated by Algorithm 1, then we have

$$(z^k - z^{k+1})^T A(w^k - w^{k+1}) \le \frac{v}{\beta}||z^k - z^{k+1}||^2, \; v \in (0,2). \tag{3.3}$$

Proof. It is easy to derive the above equation from Step 1 and Step 2(c) of Algorithm 1. □

Lemma 3. If $\mathcal{X} = \mathbb{R}^n$, let the iterative sequence $\{t^k\} = \{(w^k, z^k)\} \in \Omega$, $\{t^*\} = \{(w^*, z^*)\} \in \Omega^*$ generated by Algorithm 1,, then we have

$$(t^* - t^k)^T G(t^{k+1} - t^k) \ge ||t^{k+1} - t^k||_G^2 + \beta(w^k - w^*)^T M(w^{k+1} - w^*), \tag{3.4}$$

where $G = \begin{pmatrix} \beta R(1+\mu) & 0 \\ 0 & I \end{pmatrix}$, $R = rI_n$ with $r > 0$ is a proximal parameter.

Proof. Plugging $(w, z) = (w^{k+1}, z^{k+1})$ into the optimal condition (1.3) and substituting $\nabla f(w^*) = Mw^* + q$, we get

$$\begin{pmatrix} w^{k+1} - w^* \\ z^{k+1} - z^* \end{pmatrix}^T \left[\begin{pmatrix} M & -A^T \\ A & 0 \end{pmatrix}\begin{pmatrix} w^* \\ z^* \end{pmatrix} + \begin{pmatrix} q \\ -b \end{pmatrix}\right] \ge 0. \tag{3.5}$$

From Step 1 and Step 3, we get $z^{k+1} = z^k - \beta(Aw^k - b)$, that is

$$(z^* - z^{k+1})^T[Aw^k - b + \frac{1}{\beta}(z^{k+1} - z^k)] \ge 0. \tag{3.6}$$

Similarly, we have $w^{k+1} = \arg\min_{w\in\mathcal{X}}\{L(w, z^{k+1}) + rd(w, w^k)\}$ and get the following inequality

$$(w^* - w^{k+1})^T\{Mw^{k+1} + q - A^T z^{k+1} + R[(w - w^k) + \mu(w^k - W_k^2 w^{-1})]\} \ge 0. \tag{3.7}$$

It is trivial that $\frac{{w_{(i)}^k}^2}{w_{(i)}} \ge 2w_{(i)}^k - w_{(i)}$ providing $i = 1, \cdots, n$, $w_{(i)} \ne 0$, we substitute the inequality into (3.7) and then we get

$$(w^* - w^{k+1})^T\{Mw^{k+1} + q - A^T z^{k+1} + R[(w - w^k) + \mu(w - w^k)]\} \ge 0. \tag{3.8}$$

Invoking (3.6) and (3.8), and noticing $(w^{k+1}, z^{k+1}) \in \Omega$, we have

$$\begin{pmatrix} w^* - w^{k+1} \\ z^* - z^{k+1} \end{pmatrix}^T \left[\begin{pmatrix} M & -A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} w^{k+1} \\ z^{k+1} \end{pmatrix} + \begin{pmatrix} q \\ -b \end{pmatrix} + \begin{pmatrix} R(1+\mu) & 0 \\ -A & \frac{1}{\beta} I \end{pmatrix} \begin{pmatrix} w^{k+1} - w^k \\ z^{k+1} - z^k \end{pmatrix} \right] \geq 0. \quad (3.9)$$

The following inequality can be obtained from (3.5) and (3.9):

$$\begin{pmatrix} w^{k+1} - w^* \\ z^{k+1} - z^* \end{pmatrix}^T \begin{pmatrix} -R(1+\mu) & 0 \\ A & -\frac{1}{\beta} \end{pmatrix} \begin{pmatrix} w^{k+1} - w^k \\ z^{k+1} - z^k \end{pmatrix} \geq ||w^{k+1} - w^*||_M^2. \quad (3.10)$$

Inequality (3.10) is equivalent to

$$\left[\begin{pmatrix} w^{k+1} - w^k \\ z^{k+1} - z^k \end{pmatrix} - \begin{pmatrix} w^* - w^k \\ z^* - z^k \end{pmatrix} \right]^T \begin{pmatrix} -R(1+\mu) & 0 \\ A & -\frac{1}{\beta} I \end{pmatrix} \begin{pmatrix} w^{k+1} - w^k \\ z^{k+1} - z^k \end{pmatrix} \geq ||w^{k+1} - w^*||_M^2,$$

and the above inequality can be rewritten as

$$\begin{aligned} &(z^* - z^k)^T (z^{k+1} - z^k) + \beta R(1+\mu)(w^* - w^k)^T (w^{k+1} - w^k) \\ &\geq -\beta (z^k - z^*)^T A (w^{k+1} - w^k) - \beta (z^{k+1} - z^k)^T A (w^{k+1} - w^k) \\ &\quad + ||z^{k+1} - z^k||^2 + \beta R(1+\mu) ||w^{k+1} - w^k||^2 + \beta ||w^{k+1} - w^*||_M^2. \end{aligned} \quad (3.11)$$

When $\mathcal{X} = \mathbb{R}^n$, according to the optimal condition (1.3), the following equation holds

$$\nabla f(w^*) = A^T z^* \quad i.e. \quad Mw^* + q = A^T z^*. \quad (3.12)$$

Similarly, when $\mathcal{X} = \mathbb{R}^n$, the following equation can be obtained from (3.9):

$$\nabla f(w^{k+1}) = A^T z^{k+1} \quad i.e. \quad Mw^{k+1} + q = A^T z^{k+1}. \quad (3.13)$$

Since $z^{k+1} = z^k - \beta(Aw^k - b)$ and $Aw^* - b = 0$, $(z^k - z^*)^T A(w^{k+1} - w^k)$ in inequality (3.11) could be rewritten as

$$\begin{aligned} &(z^k - z^*)^T A (w^{k+1} - w^k) \\ &= (A^T z^k - A^T z^*)^T (w^{k+1} - w^k) \\ &= [A^T (z^{k+1} + \beta(Aw^k - b)) - A^T z^*]^T (w^{k+1} - w^k) \\ &= (w^{k+1} - w^*)^T M (w^{k+1} - w^k) + (z^k - z^{k+1})^T A (w^{k+1} - w^k), \end{aligned} \quad (3.14)$$

where the last equality uses the relations (3.12) and (3.13). The following identity is true for arbitrary vectors a, b, c:

$$(a - b)^T (a - c) = \frac{1}{2}(||a - b||^2 + ||a - c||^2 - ||b - c||^2). \quad (3.15)$$

Taking $a = w^{k+1}, b = w^*, c = w^k$ in (3.15), we get

$$(w^{k+1} - w^*)^T M (w^{k+1} - w^k) = \frac{1}{2}(||w^{k+1} - w^*||_M^2 + ||w^{k+1} - w^k||_M^2 + ||w^k - w^*||_M^2). \quad (3.16)$$

Substituting (3.16) into equation (3.14), we get

$$\begin{aligned} (z^k - z^*)^T A (w^{k+1} - w^k) = &\frac{1}{2} ||w^{k+1} - w^*||_M^2 + \frac{1}{2} ||w^{k+1} - w^k||_M^2 - \frac{1}{2} ||w^k - w^*||_M^2 \\ &+ (z^k - z^{k+1})^T A (w^{k+1} - w^k), \end{aligned} \quad (3.17)$$

and then substituting (3.17) into the right-hand side of (3.11), we have

$$\begin{aligned} &(z^* - z^k)^T (z^{k+1} - z^k) + \beta R(1+\mu)(w^* - w^k)^T (w^{k+1} - w^k) \\ \geq &||z^{k+1} - z^k||^2 + \beta R(1+\mu) ||w^{k+1} - w^k||^2 + \beta (w^k - w^*)^T M (w^{k+1} - w^*). \end{aligned} \quad (3.18)$$

Then the assertion is proved. □

Lemma 4. If $\mathcal{X}=\mathbb{R}^n$, let the iterative sequence $\{t^k\}=\{(w^k,z^k)\}$ is generated by Algorithm 1, and let $H=\begin{pmatrix}\beta R(1+\mu) & 0\\ 0 & \frac{2-v}{2}I\end{pmatrix}$, we have

$$||t^{k+1}-t^*||_G^2 \le ||t^k-t^*||_G^2 - ||t^k-t^{k+1}||_H^2. \tag{3.19}$$

Proof. First, we have

$$\begin{aligned}
&\left\|\begin{matrix} w^{k+1}-w^* \\ z^{k+1}-z^*\end{matrix}\right\|_G^2\\
&=\left\|\begin{matrix} w^{k}-w^* \\ z^{k}-z^*\end{matrix}\right\|_G^2+\left\|\begin{matrix} w^{k}-w^{k+1} \\ z^{k}-z^{k+1}\end{matrix}\right\|_G^2-2\begin{pmatrix} w^{k}-w^* \\ z^{k}-z^*\end{pmatrix}^T\begin{pmatrix}\beta R(1+\mu) & 0\\ 0 & I\end{pmatrix}\begin{pmatrix} w^{k}-w^{k+1} \\ z^{k}-z^{k+1}\end{pmatrix}\\
&=\beta R(1+\mu)||w^k-w^*||^2+||z^k-z^*||^2+\beta R(1+\mu)||w^k-w^{k+1}||^2\\
&\qquad+||z^k-z^{k+1}||^2-2\beta R(1+\mu)(w^k-w^*)^T(w^k-w^{k+1})-2(z^k-z^*)^T(z^k-z^{k+1})\\
&\le \beta R(1+\mu)||w^k-w^*||^2+||z^k-z^*||^2+\beta R(1+\mu)||w^k-w^{k+1}||^2+||z^k-z^{k+1}||^2\\
&\qquad-2||z^k-z^{k+1}||^2-2\beta R(1+\mu)||w^k-w^{k+1}||^2-2\beta(w^k-w^*)^TM(w^{k+1}-w^*)\\
&=\beta R(1+\mu)||w^k-w^*||^2+||z^k-z^*||^2-\beta R(1+\mu)||w^k-w^{k+1}||^2-||z^k-z^{k+1}||^2\\
&\qquad-2\beta(w^k-w^*)^TM(w^{k+1}-w^*)\\
&=\beta R(1+\mu)||w^k-w^*||^2+||z^k-z^*||^2-\beta R(1+\mu)||w^k-w^{k+1}||^2-||z^k-z^{k+1}||^2\\
&\qquad-\beta||w^k-w^*||_M^2-\beta||w^{k+1}-w^*||_M^2+\beta||w^k-w^{k+1}||_M^2,
\end{aligned}\tag{3.20}$$

where the inequality uses (3.4). It holds that $||a||^2+||b||^2=\frac{1}{2}(||a+b||^2+||a-b||^2)$ for any vectors a,b. Taking $a=w^{k+1}-w^*$, $b=w^k-w^*$ in this identity, we obtain

$$||w^{k+1}-w^*||_M^2+||w^k-w^*||_M^2=\frac{1}{2}||w^{k+1}+w^k-2w^*||_M^2+\frac{1}{2}||w^k-w^{k+1}||_M^2\ge\frac{1}{2}||w^k-w^{k+1}||_M^2. \tag{3.21}$$

Substituting (3.21) into (3.20), we obtain:

$$\begin{aligned}
\beta R(1+\mu)||w^{k+1}-w^*||^2+||z^{k+1}-z^*||^2 \le&\beta R(1+\mu)||w^k-w^*||^2+||z^k-z^*||^2\\
&-\beta R(1+\mu)||w^k-w^{k+1}||^2\\
&-||z^k-z^{k+1}||^2+\frac{1}{2}\beta||w^k-w^{k+1}||_M^2.
\end{aligned}\tag{3.22}$$

Combining (3.13)and (3.3), the last term of (3.22) can be reformulated as

$$\begin{aligned}
||w^k-w^{k+1}||_M^2&=(w^k-w^{k+1})^TM(w^k-w^{k+1})\\
&=(Mw^k-Mw^{k+1})^T(w^k-w^{k+1})\\
&=(z^k-z^{k+1})^T(w^k-w^{k+1})\\
&\le\frac{v}{\beta}||z^k-z^{k+1}||^2,\quad v\in(0,2).
\end{aligned}\tag{3.23}$$

Substituting (3.23) into (3.22), we get

$$\begin{aligned}
&\beta R(1+\mu)||w^{k+1}-w^*||^2+||z^{k+1}-z^*||^2\\
&\le\beta R(1+\mu)||w^k-w^*||^2+||z^k-z^*||^2-[\beta R(1+\mu)||w^k-w^{k+1}||^2+\frac{2-v}{2}||z^k-z^{k+1}||^2].
\end{aligned}\tag{3.24}$$

Let $t = \begin{pmatrix} w \\ z \end{pmatrix}$, $G = \begin{pmatrix} \beta R(1+\mu) & 0 \\ 0 & I \end{pmatrix}$, $H = \begin{pmatrix} \beta R(1+\mu) & 0 \\ 0 & \frac{2-v}{2} I \end{pmatrix}$, we get

$$||t^{k+1} - t^*||_G^2 \leq ||t^k - t^*||_G^2 - ||t^k - t^{k+1}||_H^2.$$

The assertion is proved. □

Theorem 3.1. If $\mathcal{X} = \mathbb{R}^n$, let the iterative sequence $\{t^k\} = \{(w^k, z^k)\} \in \Omega$ is generated by Algorithm 1. Therefore, the iterative sequence $\{t^k\} = \{(w^k, z^k)\}$ converges to a solution of (1.1).

Proof. Let (w^*, z^*) be the optimal solution of problem (1.1) and $z^k \neq z^{k+1}$ holds for any $k = 1, 2, \ldots n$, then the sequence $\{||t^k - t^*||_G^2\}$ is strictly monotonic decreasing by (3.19), namely

$$||t^k - t^*||_G^2 \leq ||t^{k-1} - t^*||_G^2 \leq \ldots \leq ||t^0 - t^*||_G^2. \tag{3.25}$$

From the above inequality, it follows that:

$$||t^k||_G \leq ||t^k - t^*||_G + ||t^*||_G \leq ||t^0 - t^*||_G + ||t^*||_G,$$

therefore, $\{t^k\}$ is proved to be bounded. Thus there is a cluster point $\bar{t}$ and a subsequence $\{t^{k_j}\}$ such that

$$||t^{k_j} - \bar{t}||_G \to 0 \quad (j \to \infty), \tag{3.26}$$

that is $||w^{k_j} - \overline{w}||_G \to 0,\ ||z^{k_j} - \overline{z}||_G \to 0 \quad (j \to \infty)$. We then turn to prove $\bar{t} = t^*$.

Summing (3.19) over $k = 0, 1, 2, \ldots$, we get

$$\sum_{k=0}^{\infty} ||t^k - t^{k+1}||_H^2 \leq ||t^0 - t^*||_G^2,$$

which implies

$$||t^k - t^{k+1}||_H \to 0 (k \to 0). \tag{3.27}$$

For any $(w, z) \in \Omega$, the optimality condition for the $(k+1)-$th subproblem is as follows

$$\begin{pmatrix} w - w^{k+1} \\ z - z^{k+1} \end{pmatrix}^T \left[\begin{pmatrix} \nabla f(w^{k+1}) - A^T z^{k+1} \\ Aw^{k+1} - b \end{pmatrix} + \begin{pmatrix} R(1+\mu) & 0 \\ -A & \frac{1}{\beta} I \end{pmatrix} \begin{pmatrix} w^{k+1} - w^k \\ z^{k+1} - z^k \end{pmatrix} \right] \geq 0.$$

Let $k = k_j - 1$ in the above inequality, we get:

$$\begin{pmatrix} w - w^{k_j} \\ z - z^{k_j} \end{pmatrix}^T \left[\begin{pmatrix} \nabla f(w^{k_j}) - A^T z^{k_j} \\ Aw^{k_j} - b \end{pmatrix} + \begin{pmatrix} R(1+\mu) & 0 \\ -A & \frac{1}{\beta} I \end{pmatrix} \begin{pmatrix} w^{k_j} - w^{k_j - 1} \\ z^{k_j} - z^{k_j - 1} \end{pmatrix} \right] \geq 0. \tag{3.28}$$

Taking the limit $(j \to 0)$ on the inequality (3.28) and noticing (3.27), we obtain

$$\begin{pmatrix} w - \overline{w} \\ z - \overline{z} \end{pmatrix}^T \begin{pmatrix} \nabla f(\overline{w}) - A^T \overline{z} \\ A\overline{w} - b \end{pmatrix} \geq 0,$$

which means $\overline{w} = w^*$, $\overline{z} = z^*$. Together with (3.25), the original sequence $\{w^k, z^k)\}$ also converges to $\{w^*, z^*\}$. We prove the assertion. □

3.2. **The extended modified dual ascent method with LQP regularization.** We consider the two blocks separable convex optimization problem as follows

$$\min\{f(w)+g(u)|Aw+Bu=b, w\in\mathcal{X}, u\in\mathcal{Y}\}, \tag{3.29}$$

in which $w, u \in \mathbb{R}^{n_1}, \mathbb{R}^{n_2}$, respectively; b is defined as before, $f(w)$ and $g(u)$ are quadratic functions and the feasible domains $\mathcal{X}$ and $\mathcal{Y}$ are non-empty closed convex sets. Without loss of generality, we define $f(w)$ as before, $g(u)=\frac{1}{2}u^TNu+p^Tu$ and then $\nabla g(u)=Nu+p$. We assume that the solution set of (3.29) is nonempty.

Let the Lagrange function $L(w,u,z)$ of (3.29) be

$$L(w,u,z)=f(w)+g(u)-z^T(Aw+Bu-b). \tag{3.30}$$

Similarly to (1.3), we have the following equation, which is the first-order optimality conditions of (3.29).

$$\left(l-l^*\right)^T F(l^*)\geq 0, \quad \forall w\in\mathcal{X}, u\in\mathcal{Y}, z\in\mathbb{R}^m, \tag{3.31}$$

where

$$l:=\begin{pmatrix} w\\ u\\ z\end{pmatrix}, F(l):=\begin{pmatrix}\nabla f(w)-A^Tz\\ \nabla g(u)-B^Tz\\ Ax+By-b\end{pmatrix}=\begin{pmatrix}Mw+q-A^Tz\\ Nu+p-B^Tz\\ Aw+Bu-b\end{pmatrix}.$$

We propose the extended Modified Self-adaptive DAM with LQP regularization (DAMLQP) with two blocks cases (3.29):

Algorithm 2: An extented modified self-adaptive dual ascent method with LQP regularization for (3.29).

Step 0. Initial $(w^0,u^0)=\arg\min_{w\in\mathcal{X},u\in\mathcal{Y}}L(w,u,z^0)$, initial step size β_0, lower bound of step size $\beta_{\min}$, parameters $\mu, v\in(0,1)$ and $\mu<v$, $k=0$, $\eta\in(0,1)$, tolerance ϵ.

Step 1. Find $(\tilde{z}^k,\tilde{w}^k,\tilde{u}^k,r_k,s_k)$ such that

$$\begin{cases}\tilde{z}^k=z^k-\beta_k(Aw^k+Bu^k-b),\\ \tilde{w}^k=\arg\min_{w\in\mathcal{X}}\{L(w,u^k,\tilde{z}^k)+rd(w,w^k)\},\\ \tilde{u}^k=\arg\min_{u\in\mathcal{Y}}\{L(\tilde{w}^k,u,\tilde{z}^k)+sd(u,u^k)\}\end{cases} \tag{3.32}$$

$$r_k=\frac{\beta(z^k-z^{k+1})^TA(w^k-w^{k+1})}{||z^k-z^{k+1}||^2}, \quad s_k=\frac{\beta(z^k-z^{k+1})^TB(u^k-u^{k+1})}{||z^k-z^{k+1}||^2}, \tag{3.33}$$

where $d(w,w^k)$ and $d(u,u^k)$ are defined in (2.1), $r,s>0$ are parameters.

Step 2.

(a) If $r_k>v$, $s_k>v$, step size β_k is too large. $\beta_k:=\eta*\beta_k*\min\{1,\frac{1}{r_k}\}$ and go to Step 1.

(b) If $r_k<\mu$, $s_k<\mu$, step size β_k is too small. $\beta_{k+1}:=\max\{\beta_{\min},\eta*\beta_k*\frac{1}{r_k}\}$ for the next iteration and go to Step 3.

(c) If $\mu<r_k<v$, $\mu<s_k<v$, step size β_k is proper. $\beta_{k+1}:=\max\{\beta_{\min},\beta_k\}$ and go to Step 3.

Step 3. Let $(w^{k+1},u^{k+1},z^{k+1})=(\tilde{w}^k,\tilde{u}^k,\tilde{z}^k)$. If the stopping criterion $\max\{\frac{||w^{k+1}-w^k||}{||w^k||},\frac{||u^{k+1}-u^k||}{||u^k||}\}\leq\epsilon$ is met or maximal iteration numbers are attained, stop the algorithm, or else, $k=k+1$, go to Step 1.

Lemma 5. Let the iterative sequence $\{l^k\}=\{(w^k,u^k,z^k)\}\in\Omega$ is generated by Algorithm 2, then we have:

$$(l^*-l^k)^TG(l^{k+1}-l^k)\geq||l^{k+1}-l^k||_G^2+\beta(w^k-w^*)^TM(w^{k+1}-w^*)+\beta(u^k-u^*)^TN(u^{k+1}-u^*), \tag{3.34}$$

where $G := \begin{pmatrix} \beta R(1+\mu) & 0 & 0 \\ 0 & \beta S(1+\mu) & 0 \\ 0 & 0 & I \end{pmatrix}$.

Proof. Plugging $(w,u,z) = (w^{k+1}, u^{k+1}, z^{k+1})$ into the optimal condition (3.31), and noticing $\nabla f(w^*) = Mw^* + q$, $\nabla g(u^*) = Nu^* + p$, we get

$$\begin{pmatrix} w^{k+1} - w^* \\ u^{k+1} - u^* \\ z^{k+1} - z^* \end{pmatrix}^T \left[\begin{pmatrix} M & 0 & -A^T \\ 0 & N & -B^T \\ A & B & 0 \end{pmatrix} \begin{pmatrix} w^* \\ u^* \\ z^* \end{pmatrix} + \begin{pmatrix} q \\ p \\ -b \end{pmatrix} \right] \geq 0. \tag{3.35}$$

From Step 1 and Step 3, we have $z^{k+1} = z^k - \beta(Aw^k + Bu^k - b)$, and then we get:

$$(z^* - z^{k+1})^T [Aw^k + Bu^k - b + \frac{1}{\beta}(z^{k+1} - z^k)] \geq 0.$$

Similarly to (3.9), we have:

$$\begin{pmatrix} w^* - w^{k+1} \\ u^* - u^{k+1} \\ z^* - z^{k+1} \end{pmatrix}^T \left[\begin{array}{c} \begin{pmatrix} M & 0 & -A^T \\ 0 & N & -B^T \\ A & B & 0 \end{pmatrix} \begin{pmatrix} w^{k+1} \\ u^{k+1} \\ z^{k+1} \end{pmatrix} + \begin{pmatrix} q \\ p \\ -b \end{pmatrix} \\ + \begin{pmatrix} R(1+\mu) & 0 & 0 \\ 0 & S(1+\mu) & 0 \\ -A & -B & \frac{1}{\beta} I \end{pmatrix} \begin{pmatrix} w^{k+1} - w^k \\ u^{k+1} - u^k \\ z^{k+1} - z^k \end{pmatrix} \end{array} \right] \geq 0, \tag{3.36}$$

where $R = rI_{n_1}$ with $r > 0$ and $S = sI_{n_2}$ with $s > 0$ are proximal parameters.

The following inequality can be obtained from (3.35) and (3.36):

$$\begin{pmatrix} w^{k+1} - w^* \\ u^{k+1} - u^* \\ z^{k+1} - z^* \end{pmatrix}^T \begin{pmatrix} -R(1+\mu) & 0 & 0 \\ 0 & -S(1+\mu) & 0 \\ A & B & -\frac{1}{\beta} I \end{pmatrix} \begin{pmatrix} w^{k+1} - w^k \\ u^{k+1} - u^k \\ z^{k+1} - z^k \end{pmatrix} \geq ||w^{k+1} - w^*||_M^2 + ||u^{k+1} - u^*||_N^2. \tag{3.37}$$

Inequality (3.37) is equivalent to

$$\left[\begin{pmatrix} w^{k+1} - w^k \\ u^{k+1} - u^k \\ z^{k+1} - z^k \end{pmatrix} - \begin{pmatrix} w^* - w^k \\ u^* - u^k \\ z^* - z^k \end{pmatrix} \right]^T \begin{pmatrix} -R(1+\mu) & 0 & 0 \\ 0 & -S(1+\mu) & 0 \\ A & B & \frac{1}{\beta} I \end{pmatrix} \begin{pmatrix} w^{k+1} - w^k \\ u^{k+1} - u^k \\ z^{k+1} - z^k \end{pmatrix}$$
$$\geq ||w^{k+1} - w^k||_M^2 + ||u^{k+1} - u^k||_N^2.$$

We can reformulate the above inequality as follows:

$$\begin{aligned} &\beta R(1+\mu)(w^* - w^k)^T (w^{k+1} - w^k) + \beta S(1+\mu)(u^* - u^k)^T (u^{k+1} - u^k) + (z^* - z^k)^T (z^{k+1} - z^k) \\ &\geq -\beta (z^k - z^*)^T A(w^{k+1} - w^k) - \beta (z^k - z^*)^T B(u^{k+1} - u^k) - \beta (z^{k+1} - z^k)^T A(w^{k+1} - w^k) \\ &\quad - \beta (z^{k+1} - z^k)^T B(u^{k+1} - u^k) + \beta R(1+\mu)||w^{k+1} - w^k||^2 + \beta S(1+\mu)||u^{k+1} - u^k||^2 \\ &\quad + ||z^{k+1} - z^k||^2 + \beta(||w^{k+1} - w^*||_M^2 + ||u^{k+1} - u^*||_N^2). \end{aligned} \tag{3.38}$$

Similarly to (3.12), (3.13) and invoking $z^{k+1} = z^k - \beta(Aw^k + Bu^k - b)$, $Aw^* + Bu^* - b = 0$, $(z^k - z^*)^T A(w^{k+1} - w^k) + (z^k - z^*)^T B(u^{k+1} - u^k)$ in inequality (3.38) are written as:

$$\begin{aligned}
&(z^k - z^*)^T A(w^{k+1} - w^k) + (z^k - z^*)^T B(u^{k+1} - u^k)\\
&= [A^T(z^{k+1} + \beta(Aw^k + Bu^k - b)) - A^T z^*]^T (w^{k+1} - w^k)\\
&\quad + [B^T(z^{k+1} + \beta(Aw^k + Bu^k - b) - B^T z^*]^T (u^{k+1} - u^k)\\
&= (w^{k+1} - w^*)^T M(w^{k+1} - w^k) + (u^{k+1} - u^*)^T N(u^{k+1} - u^k)\\
&\quad + (z^k - z^{k+1})^T A(w^{k+1} - w^k) + (z^k - z^{k+1})^T B(u^{k+1} - u^k).
\end{aligned} \tag{3.39}$$

Similarly, we get

$$\begin{aligned}
&(w^{k+1} - w^*)^T M(w^{k+1} - w^k) = \frac{1}{2}(||w^{k+1} - w^*||_M^2 + ||w^{k+1} - w^k||_M^2 - ||w^k - w^*||_M^2),\\
&(u^{k+1} - u^*)^T N(u^{k+1} - u^k) = \frac{1}{2}(||u^{k+1} - u^*||_N^2 + ||u^{k+1} - u^k||_N^2 - ||u^k - u^*||_N^2).
\end{aligned} \tag{3.40}$$

The right-hand side of equation (3.39) is substituted into (3.40), then taking (3.39) into (3.38), we can get

$$\begin{aligned}
&\beta R(1+\mu)(w^* - w^k)^T (w^{k+1} - w^k) + \beta S(1+\mu)(u^* - u^k)^T (u^{k+1} - u^k) + (z^* - z^k)^T (z^{k+1} - z^k)\\
&\geq \beta R(1+\mu)||w^{k+1} - w^k||^2 + \beta S(1+\mu)||u^{k+1} - u^k||^2 + ||z^{k+1} - z^k||^2\\
&\quad + \frac{1}{2}\beta(||w^{k+1} - w^*||_M^2 + ||w^k - w^*||_M^2 - ||w^{k+1} - w^k||_M^2)\\
&\quad + \frac{1}{2}\beta(||u^{k+1} - u^*||_N^2 + ||u^k - u^*||_N^2 - ||u^{k+1} - u^k||_N^2)\\
&= \beta R(1+\mu)||w^{k+1} - w^k||^2 + \beta S(1+\mu)||u^{k+1} - u^k||^2 + ||z^{k+1} - z^k||^2\\
&\quad + \beta(w^k - w^*)^T M(w^{k+1} - w^*) + \beta(u^k - u^*)^T N(u^{k+1} - u^*).
\end{aligned}$$

The assertion is proved. □

Lemma 6. Let the iterative sequence $\{l^k\} = \{(w^k, u^k, z^k)\} \in \Omega$ is generated by Algorithm 2, and let $H := \begin{pmatrix} \beta R(1+\mu) & 0 & 0 \\ 0 & \beta S(1+\mu) & 0 \\ 0 & 0 & (1-v)I \end{pmatrix}$, $v \in (0, 1)$, then we have

$$||l^{k+1} - l^*||_G^2 \leq ||l^k - l^*||_G^2 - ||l^k - l^{k+1}||_H^2. \tag{3.41}$$

Proof. First, we have

$$\begin{aligned}
&\left\| \begin{matrix} w^{k+1} - w^* \\ u^{k+1} - u^* \\ z^{k+1} - z^* \end{matrix} \right\|^2_{\begin{pmatrix} \beta R(1+\mu) & 0 & 0 \\ 0 & \beta S(1+\mu) & 0 \\ 0 & 0 & I \end{pmatrix}}\\
&= \left\| \begin{matrix} w^k - w^* \\ u^k - u^* \\ z^k - z^* \end{matrix} \right\|^2_{\begin{pmatrix} \beta R(1+\mu) & 0 & 0 \\ 0 & \beta S(1+\mu) & 0 \\ 0 & 0 & I \end{pmatrix}} + \left\| \begin{matrix} w^k - w^{k+1} \\ u^k - u^{k+1} \\ z^k - z^{k+1} \end{matrix} \right\|^2_{\begin{pmatrix} \beta R(1+\mu) & 0 & 0 \\ 0 & \beta S(1+\mu) & 0 \\ 0 & 0 & I \end{pmatrix}}
\end{aligned}$$

$$
\begin{aligned}
&- 2 \begin{pmatrix} w^k - w^* \\ u^k - u^* \\ z^k - z^* \end{pmatrix}^T \begin{pmatrix} \beta R(1+\mu) & 0 & 0 \\ 0 & \beta S(1+\mu) & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} w^k - w^{k+1} \\ u^k - u^{k+1} \\ z^k - z^{k+1} \end{pmatrix} \\
&= \beta R(1+\mu)||w^k - w^*||^2 + \beta S(1+\mu)||u^k - u^*||^2 + ||z^k - z^*||^2 + \beta R(1+\mu)||w^k - w^{k+1}||^2 \\
&\quad + \beta S(1+\mu)||u^k - u^{k+1}||^2 + ||z^k - z^{k+1}||^2 - 2\beta R(1+\mu)(w^k - w^*)^T(w^k - w^{k+1}) \\
&\quad - 2\beta S(1+\mu)(u^k - u^*)^T(u^k - u^{k+1}) - 2(z^k - z^*)^T(z^k - z^{k+1}) \\
&\le \beta R(1+\mu)||w^k - w^*||^2 + \beta S(1+\mu)||u^k - u^*||^2 + ||z^k - z^*||^2 + \beta R(1+\mu)||w^k - w^{k+1}||^2 \\
&\quad + \beta S(1+\mu)||u^k - u^{k+1}||^2 + ||z^k - z^{k+1}||^2 - 2\beta R(1+\mu)||w^{k+1} - w^k||^2 \\
&\quad - 2\beta S(1+\mu)||u^{k+1} - u^k||^2 - 2||z^{k+1} - z^k||^2 - 2\beta(w^k - w^*)^T M(w^{k+1} - w^*) \\
&\quad - 2\beta(u^k - u^*)^T N(u^{k+1} - u^*) \\
&= \beta R(1+\mu)||w^k - w^*||^2 + \beta S(1+\mu)||u^k - u^*||^2 + ||z^k - z^*||^2 - \beta R(1+\mu)||w^k - w^{k+1}||^2 \\
&\quad - \beta S(1+\mu)||u^k - u^{k+1}||^2 - ||z^k - z^{k+1}||^2 - \beta(||w^k - w^*||_M^2 + ||w^{k+1} - w^*||_M^2 \\
&\quad - ||w^{k+1} - w^k||_M^2) - \beta(||u^k - u^*||_N^2 + ||u^{k+1} - u^*||_N^2 - ||u^{k+1} - u^k||_N^2),
\end{aligned} \tag{3.42}
$$

where the inequality uses (3.34).

Similarly to (3.21), we obtain

$$
\begin{aligned}
&||w^{k+1} - w^*||_M^2 + ||w^k - w^*||_M^2 \ge \frac{1}{2}||w^k - w^{k+1}||_M^2, \\
&||u^{k+1} - u^*||_N^2 + ||u^k - u^*||_N^2 \ge \frac{1}{2}||u^k - u^{k+1}||_N^2.
\end{aligned} \tag{3.43}
$$

Substituting (3.43) into (3.42), we obtain

$$
\begin{aligned}
&\beta R(1+\mu)||w^{k+1} - w^*||^2 + \beta S(1+\mu)||u^{k+1} - u^*||^2 + ||z^{k+1} - z^*||^2 \\
&\le \beta R(1+\mu)||w^k - w^*||^2 + \beta S(1+\mu)||u^k - u^*||^2 + ||z^k - z^*||^2 - \beta R(1+\mu)||w^k - w^{k+1}||^2 \\
&\quad - \beta S(1+\mu)||u^k - u^{k+1}||^2 - ||z^k - z^{k+1}||^2 + \frac{1}{2}\beta||w^k - w^{k+1}||_M^2 + \frac{1}{2}\beta||u^k - u^{k+1}||_N^2.
\end{aligned} \tag{3.44}
$$

Similarly to (3.23), we have

$$
\begin{aligned}
||w^k - w^{k+1}||_M^2 &= (Mw^k - Mw^{k+1})^T(w^k - w^{k+1}) = (z^k - z^{k+1})^T A(w^k - w^{k+1}) \\
&\le \frac{v}{\beta}||z^k - z^{k+1}||^2, \\
||u^k - u^{k+1}||_N^2 &= (Nu^k - Nu^{k+1})^T(u^k - u^{k+1}) = (z^k - z^{k+1})^T B(u^k - u^{k+1}) \\
&\le \frac{v}{\beta}||z^k - z^{k+1}||^2.
\end{aligned} \tag{3.45}
$$

Substituting (3.45) into (3.44), we get

$$
\begin{aligned}
&\beta R(1+\mu)||w^{k+1} - w^*||^2 + \beta S(1+\mu)||u^{k+1} - u^*||^2 + ||z^{k+1} - z^*||^2 \\
&\le \beta R(1+\mu)||w^k - w^*||^2 + \beta S(1+\mu)||u^k - u^*||^2 + ||z^k - z^*||^2 \\
&\quad - [\beta R(1+\mu)||w^k - w^{k+1}||^2 + \beta S(1+\mu)||u^k - u^{k+1}||^2 + (1-v)||z^k - z^{k+1}||^2]. \quad v \in (0,1)
\end{aligned}
$$

Recalling the definitions of G and H, the assertion is proved. □

Theorem 3.2. Let the iterative sequence $\{l^k\} = \{(w^k, u^k, z^k)\} \in \Omega$ is generated by Algorithm 2, then it converges to a solution of (3.29).

This proof is similar to that of Theorem 3.1, thus it is omitted.

4. Numerical Experiments

We will investigate the performance of our proposed algorithm in this section, thus we do some experiments on solving a synthetic problem. The experiments were written on a laptop computer with Intel Core i5-8250U CPU at 1.60GHz, 8GB memory and Windows 10 operating system. All the codes were written on MATLAB R2017b.

4.1. **Setup of experiments.** We take the following optimization problem into account:

$$\begin{aligned} \min\ & \frac{1}{2}||x-c||^2 \\ \text{s.t}\ & Ax=b, \\ & x\geq 0, \end{aligned} \tag{4.1}$$

where $A\in R^{m\times n}$, $b\in R^m$, $c\in R^n$. For solving problem (4.1), there have been quite a few efficient algorithms/solvers such as ALM, CPPA, CVX , etc. For simplicity, we only include CPPA and CVX in our experiments. Let $z\in\mathbb{R}^m$ be the Lagrange multiplier. For solving (4.1), the subproblem of CPPA is as follows:

$$\tilde{x}^k=\arg\min\{\frac{1}{2}||x-c||^2-{z^k}^T(Ax-b)|x\geq 0\},$$

which can be solved by $\tilde{x}^k=\max\{0,\frac{c+A^Tz^k+rx^k}{1+r}\}$.

The subproblem of our algorithm (DAMLQP) is as follows:

$$\tilde{x}^k=\arg\min\{\frac{1}{2}||x-c||^2-{z^k}^T(Ax-b)+rd(x,x^k)\}, \tag{4.2}$$

where $d(x,x^k)$ is defined as (2.1). The optimality condition of (4.2) is equivalent to the following equation

$$x-c-A^Tz^{k+1}+r[x-x^k+\mu(x^k-x^k.*x^k./x)]=0, \tag{4.3}$$

where $.*$ and $./$ denote the elementwise production and division, respectively. Multiplying $\frac{1}{r}x$ elementwisely on both sides of (4.3), we get

$$(1+\frac{1}{r})x.*x-[\frac{c+A^Tz^{k+1}}{r}+(1-\mu)x^k].*x-\mu x^k.*x^k=0. \tag{4.4}$$

Noticing (4.4) is in fact a separable equation, letting $u=1+\frac{1}{r}$, $v=\frac{c+A^Tz^{k+1}}{r}+(1-\mu)x^k$, $w=-\mu x^k.*x^k$, (4.4) can be solved by $x=\frac{v+\sqrt{v.*v-4uw}}{2u}$, where $\sqrt{\ }$ denotes elementwise square root operation.

Based on the optimality condition of (4.1)

$$\begin{cases} (x-c-A^Tz,x)=0, \\ x\geq 0, \\ x-c-A^Tz\geq 0, \\ Ax=b, \end{cases} \tag{4.5}$$

the random problem can be generated by the following procedure such that the accurate solution of (4.1) is known:

CVX: http://cvxr.com/cvx/download/

```
m = 50; n =2000;
A = rand(m,n); xop = max(0,rand(n,1)*2-1); b = A*xop;
p = setdiff(1:n,find(xop))';
q = zeros(n,1);q(p)=rand(size(p));
lamop = randn(m,1);
c = xop - q - (lamop'*A)';
```

We compare the speed performance of all test algorithms of both iteration number and computation time. When either a maximal iteration number (denoted by "maxit") is attained or the relative error between the current iterate x^k and the accurate solution x^* defined as

$$err(x^k, x^*) := \frac{||x^k - x^*||}{||x^*||}$$

becomes less than some tolerance (denoted by "tol"), i.e., $err(x^k, x^*) < tol$, algorithms are stopped. The default setting of maxit is 3000 unless otherwise specified.

To ensure a fair comparison, the parameters in all algorithms were manually tuned to maximize their performance. In CPPA, we set $r = 30$, $\gamma = 1.5$ when m=100, 150 and 200; $r = 10$, γ=1 when m=10 and 50. In DAMLQP, the step size β is self-adaptive and can be initialized with arbitrary value. The parameter v usually takes $v = 1.7$. The parameters r and μ take $r = 3 \times 10^{-5}$ and $\mu = 3 \times 10^{-3}$ when m=100, 150 and 200, while they take $r = 10^{-4}$ and $\mu = 3 \times 10^{-4}$ when m=10 and 50.

4.2. **Numerical results on synthetic problems.** First of all, the convergence behavior of our algorithms is investigated and we test three cases: $(m, n) = (50, 5000)$, (100,5000) and (150,10000). The iteration progress of relative error are plotted in Figure 1. Since CVX usually uses interior-point method whose per-iteration cost is not comparable to that of DAMLQP and CPPA, so CVX is not included here. From Figure 1, it can be seen that the convergence speed of DAMLQP is overall satisfactory compared with that of CPPA, implying that our algorithm can rapidly obtain an accurate solution of problem (4.1). Also, we can observe that the iteration number of our algorithm is about 50% less than that of CPPA in all three cases.

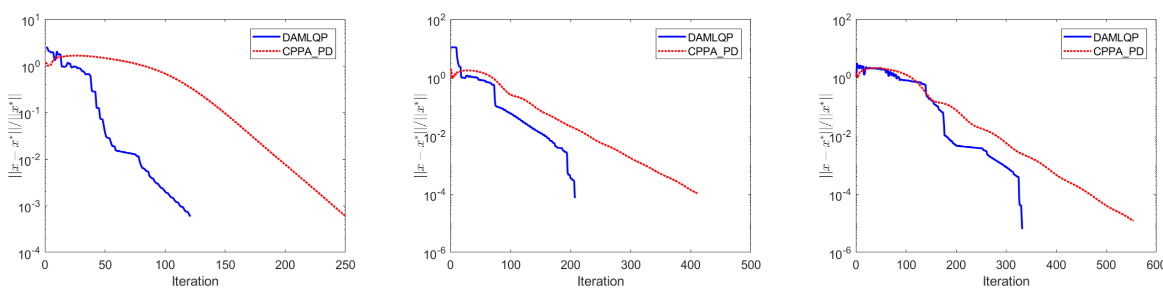

FIGURE 1. Numerical results with (m, n)=(50,5000), (100,5000) and (150,10000), respectively

Then, we turn to investigate the speed performance of test algorithms under several different settings of (m, n). "Iter" represents the iteration number and "Time" denote the computation time (in seconds). We run CVX first, denote its output as x_{CVX} and set $tol := err(x_{\text{CVX}}, x^*)$, then run DAMLQP and CPPA until $err(x^k, x^*) < tol$ is met. The mean values of iteration numbers and computation time over 10 random instances are presented in Table 1. Moreover, CVX is excluded when $(m, n) = (200, 18000)$ owning to its excessively long computation time. As the per-iteration costs of different algorithms are different, the iteration number may not be a good measure of numerical efficiency. It is more informative to show the computation time rather than the iteration number. $\text{Time}_{\text{CPPA}}$ denotes the computation time of CPPA, $\text{Time}_{||\text{A}^\text{T}\text{A}||}$ denotes the computation time of $||A^\text{T}A||$, Time_{sum} denotes the sum of $\text{Time}_{||A^\text{T}A||}$ and $\text{Time}_{\text{CPPA}}$. On one hand, $||A^\text{T}A||$ is necessary in order to implement CPPA; on the other hand, $\text{Time}_{||A^\text{T}A||}$ could be much longer than $\text{Time}_{\text{CPPA}}$, so we list it separately. In contrary, $||A^\text{T}A||$ is not required in the implementation of DAMLQP as illustrated in Remark 3.1. From Table 1, we can see that DAMLQP outperforms CVX in terms of computation time in all cases. In 8 cases among all 11 cases, DAMLQP outperforms CPPA in terms of computation time when $\text{Time}_{||A^\text{T}A||}$ is not included. Take $(m, n) = (100, 10000)$ as an example, the computation time of DAMLQP is 58.19% that of CPPA. If $\text{Time}_{||A^\text{T}A||}$ is included, the advantages of our algorithm will be more obvious. In this case, the computation time of DAMLQP is 0.56% that of Time_{sum}. Furthermore, the performance advantage becomes more pronounced when the setting of ratio $\frac{n}{m}$ is larger. For instance, when $(m, n) = (100, 2000)(\frac{n}{m} = 20)$, the computation time of DAMLQP is 11.4% less than that of CPPA, when $(m, n) = (200, 5000)(\frac{n}{m} = 25)$, the computation time of DAMLQP is 41.8% less than that of CPPA.

Table 1. Iteration number and computation time with different settings of (m, n)

(m, n)	CVX	CPPA				DAMLQP	
	Time	Iter.	Time_{sum}	$\text{Time}_{\|\|A^T A\|\|}$	$\text{Time}_{\text{CPPA}}$	Iter.	Time
(50,2000)	6.594	263	0.449	0.423	0.028	191.3	0.026
(50,5000)	9.469	243	3.662	3.627	0.035	120	0.03
(100,2000)	6.813	374	0.423	0.388	0.035	259	0.031
(100,5000)	16.938	351	3.774	3.681	0.093	150	0.073
(100,10000)	33.094	315	24.17	23.938	0.232	105	0.135
(100,20000)	54.359	441.6	191.537	190.605	0.932	235.2	0.578
(150,2000)	71.703	584.4	0.418	0.366	0.052	492	0.089
(150,5000)	500.125	574.4	4.126	3.876	0.25	448.1	0.303
(200,2000)	90.875	693.6	0.441	0.359	0.082	562.7	0.183
(200,5000)	564.5	626.2	4.924	4.313	0.611	575.7	0.546
(200,18000)	-	712.6	168.77	165.77	2.998	480.1	2.917

To justify the above observation, with fixed $m = 10$, 50 and 100, we let n increases from 1000 to 20000 with interval 1000, and plot the computation time $v.s.$ dimension n in Figure 2. It can be seen that when $m = 10$ and 100, our algorithm is usually faster than CVX and CPPA in terms of computation time. For example, when m =10, as the dimension n increases from 5000 to 18000, $\text{Time}_{\text{DAMLQP}}$ increases from 0.029 to 0.125 (approximately 3 times), whereas $\text{Time}_{\text{CPPA}}$ increases from 0.048 to 0.246 (approximately 40 times), $\text{Time}_{||A^T A||}$ increases from 10.276 to 191.422 (approximately 19 times) and Time_{CVX} increases from 5.247 to 27.738 (approximately 5 times). This may indicate that our algorithm has both higher efficiency and better dimensional scalability. When $m = 50$, it is shown that CPPA may be preferable for smaller n ($n \leq 11000$). For example, CPPA is approximately 23.9% faster than our algorithm when $n = 10000$, but its performance advantage diminishes as n grows. For example, our algorithm is approximately 29.4% faster than CPPA when $n = 17000$.

The above experimental results were obtained under a mild tolerance setting around 10^{-4}. We observe from Figure 1 that our algorithm converges faster at the final stage of the iteration progress, so we would like to study the numerical behavior of the tested algorithms under more challenging tolerance settings. Under different tolerance settings, we tested over 10 instances randomly, and the average computation time and iteration number of the test algorithms are shown in Table 2.

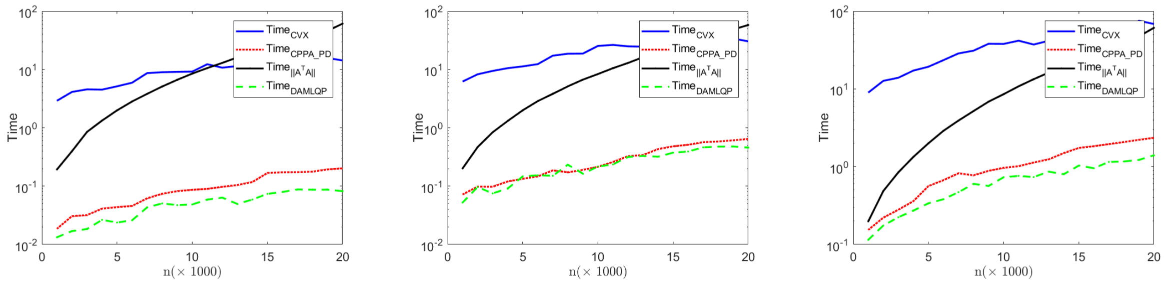

FIGURE 2. the computation time $v.s.$ dimension n: when $n \in [1000, 20000]$, $m = 10, 50$ and 100, respectively

Table 2 indicates that, with these more challenging tolerance settings, in 20 cases among all 24 cases, DAMLQP is faster than CPPA by 17.6% in average. Specially, when $(m, n) = (50, 3000)$ and $tol = 10^{-8}$, the computation time of DAMLQP is 9.3% less than that of CPPA. In contrary, when the tolerance is decreased to 10^{-12}, the computation time of DAMLQP is 19.4% less than that of CPPA. Furthermore, when the ratio $\frac{n}{m}$ increases, the advantage of our algorithm is more conspicuous. Take the case $tol = 10^{-13}$ as an example, when $(m, n) = (100, 3000)(\frac{n}{m} = 30)$, the computation time of DAMLQP is 20.3% less than that of CPPA. In contrary, when $(m, n) = (100, 10000)(\frac{n}{m} = 100)$, the computation time of DAMLQP is 31.6% less than that of CPPA, which suggests that the speed advantage of our algorithm is more obvious with more challenging settings of tolerance.

TABLE 2. Iteration number with different tolerance and (m, n)

tolerance	(m,n)=(50,3000)				(m,n)=(100,3000)			
	DAMLQP		CPPA		DAMLQP		CPPA	
	Iter.	Time	Iter.	Time	Iter.	Time	Iter.	Time
10^{-8}	277.7	0.039	599.1	0.043	469.8	0.082	790.8	0.071
10^{-9}	297.3	0.042	664.4	0.048	516.5	0.090	885.5	0.080
10^{-10}	321.2	0.045	730.1	0.052	559.2	0.097	977.6	0.088
10^{-11}	341	0.048	796	0.057	592.6	0.103	1072.5	0.103
10^{-12}	360.4	0.050	862.1	0.062	654.9	0.113	1164.7	0.128
10^{-13}	382.4	0.053	928.9	0.067	734.5	0.126	1259.6	0.158
tolerance	(m,n)=(100,10000)				(m,n)=(200,18000)			
	DAMLQP		CPPA		DAMLQP		CPPA	
	Iter.	Time	Iter.	Time	Iter.	Time	Iter.	Time
10^{-8}	337.1	0.364	782	0.490	610.3	3.116	897.4	3.312
10^{-9}	361.9	0.390	874.4	0.543	646	3.288	992.4	3.648
10^{-10}	388.1	0.415	968.3	0.598	698.2	3.541	1088.1	4.016
10^{-11}	421.8	0.451	1059.5	0.658	732.9	3.703	1184.1	4.394
10^{-12}	454.9	0.486	1152.9	0.734	814.9	4.099	1279.9	4.773
10^{-13}	526.8	0.557	1247.5	0.814	990.1	4.905	1381.4	5.176

We may conclude that for solving the synthetic problem (4.1), our algorithm outperforms CVX markedly; its performance is competitve compared with CPPA and its performance advantage is more conspicuous when $\text{Time}_{||A^TA||}$ is included. Moreover, our algorithm is preferable with larger dimension n and smaller tolerance tol.

5. Conclusion

In this paper, by combining DAM algorithm and LQP regularization, we propose a new algorithm DAMLQP for solving the linearly constrained quadratic convex problems. When the objective function $f(x)$ is quadratic, we derived the relaxed step size condition such that its convergence speed is potentially faster. Moreover, the step size β can be initialized with arbitrary value in DAMLQP, thus it could be more suitable for practical problems. Preliminary numerical results demonstrated the effectiveness of our proposed algorithm in terms of both computation time and dimension scalability.

Statements and Declarations

The authors declare that they have no conflict of interest, and the manuscript has no associated data.

Acknowledgements

Research supported by Qinglan Project of Jiangsu Province (2024), Key Project of Sichuan Provincial Administration of Traditional Chinese Medicine(2024zd029)), Research Project on the Reform of Postgraduate Education and Teaching in Chongqing (yjg223094), Chongqing Research Project of Higher Education Teaching Reform (232077), National Natural Science Foundation of China (12271067), the Chongqing Natural Science Foundation (CSTB2024NSCQ-MSX0973), the Science and Technology Research Key Program of Chongqing Municipal Education Commission (KJZD-K202200704), and the Group Building Scientific Innovation Project for Universities in Chongqing (CXQT21021).

References

[1] K. J. Arrow, L. Hurwicz, and H. Uzawa. Studies in linear and non-linear programming. *Stanford University Press*, II, 1958.

[2] A. Auslender, M. Teboulle, and S. Ben-tiba. A logarithmic-quadratic proximal method for variational inequalities. *Computational Optimization and Applications*, 12:31–40, 1999.

[3] D. P. Bertsekas and P. Tseng. Relaxation methods for minimum cost ordinary and generalized network flow problems. *Operations Research*, 36(1):93–114, 1988.

[4] A. Bnouhachem. On LQP alternating direction method for solving variational inequality problems with separable structure. *Journal of Inequalities and Applications*, 246(1):519–532, 2014.

[5] A. Bnouhachem, H. Benazza, and M. Khalfaoui. An inexact alternating direction method for solving a class of structured variational inequalities. *Applied Mathematics and Computation*, 219(14):7837–7846, 2013.

[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[7] J. H. Bramble. The Lagrange multiplier method for Dirichlet's problem. *Mathematic of Computation*, 37(155):1–11, 1981.

[8] J. H. Bramble, J. E. Pasciak, and A. T. Vassilev. Analysis of the inexact Uzawa algorithm for saddle point problems. *SIAM Journal on Numerical Analysis*, 34(3):1072–1092, 1997.

[9] J. F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[10] J. F. Cai, S. Osher, and Z. W. Shen. Linearized Bregman iterations for compressed sensing. *Mathematics of Computation*, 78(267):1515–1536, 2009.

[11] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[12] Y. H. Dai, D. Han, X. M. Yuan, and W. Zhang. A sequential updating scheme of the lagrange multiplier for separable convex programming. *Mathematics of Computation*, 86(303):315–343, 2017.

[13] E. Esser, X. Q. Zhang, and T. F. Chan. A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM Journal on Imaging Sciences*, 3(4):1015–1046, 2010.

[14] B. S. He, F. Ma, and X. M. Yuan. Convergence study on the symmetric version of ADMM with larger step sizes. *SIAM Journal on Imaging Sciences*, 9(3):1467–1501, 2016.

[15] B. S. He and X. M. Yuan. Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective. *SIAM Journal on Imaging Sciences*, 5(1):119–149, 2012.
[16] M. Josefsson and M. Patriksson. Sensitivity analysis of separable traffic equilibrium equilibria with application to bilevel optimization in network design. *Transportation Research Part B: Methodological*, 41(1):4–31, 2007.
[17] E. S. Levitin and B. T. Polyak. Constrained minimization methods. *USSR Computational Mathematics and Mathematical Physics*, 6(5):1–50, 1966.
[18] M. Li. A hybrid LQP-based method for structured variational inequalities. *International Journal of Computer Mathematics*, 89(10):1412–1425, 2012.
[19] F. X. Liu, L. L. Xu, Y. H. Sun, and D. R. Han. A proximal alternating direction method for multi-block coupled convex optimization. *Journal of Industrial and Management Optimization*, 15(2):723–737, 2019.
[20] Z. Q. Luo and P. Tseng. On the convergence rate of dual ascent methods for linearly constrained convex minimization. *Mathematics of Operations Research*, 18(4):846–867, 1993.
[21] L. D. Popov. A modification of the Arrow-Hurwitz method for search of saddle points. *Mathematical notes of the Academy of Science of the USSR*, 28(5):845–848, 1980.
[22] Y. Shen, C. Liu, Y. Zuo, and X. Zhang. A modified self-adaptive dual ascent method with relaxed stepsize condition for linearly constrained quadratic convex optimization. *Journal of Industrial and Management Optimization*, 19(5):3637–3649, 2023.
[23] M. Tao and X. M. Yuan. Accelerated Uzawa methods for convex optimization. *Mathematics of Computation*, 86(306):1821–1845, 2017.
[24] J. A. Ventura. Computational development of a lagrangian dual approach for quadratic networks. *Networks*, 21(4):469–485, 1991.
[25] Z. M. Wu and M. Li. An LQP-based symmetric alternating direction method of multipliers with larger step sizes. *Journal of the Operations Research Society of China*, 7(2):365–383, 2019.
[26] X. M. Yuan and M. Li. An LQP-based decomposition method for solving a class of variational inequalities. *SIAM Journal on Optimization*, 21(4):1309–1318, 2011.
[27] X. Zhang. Dual ascent method for some convex optimization problems. Master's thesis, Nanjing University, 2012.