STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **16**, December 2026, pp 5695–5709.
Published online in International Academic Press (www.IAPress.org)

Presolving a Convex Quadratic Minimization Problem

Samia Djemai [1,2,*], Mohand Ouamer Bibi [1,3], Abdelhek Laouar [1,3]

[1] *LaMOS Research Unit, Bejaia University, Targa Ouzemmour, 06000 Bejaïa, Algeria*
[2] *Department of Mathematics, Jijel University, Ouled Aissa BP 98, 18000 Jijel, Algeria*
[3] *Department of Operational Research, Bejaia University, Targa Ouzemmour, 06000 Bejaïa, Algeria*

Abstract In this work, we consider a convex quadratic minimization problem with one linear equality and box constraints. The goal is to find certain indices active at the optimum in order to fix the corresponding variables in advance. We also look for other variables whose corresponding components of the reduced gradient will be null at the optimum. We propose a presolving procedure that allows us to reduce the dimension of the problem and to begin the resolution algorithm with an appropriate support (basis) that ensures its convergence. For this, the technique of bounding the optimal reduced gradient will be used. Two numerical illustrative examples are presented, and comparative numerical results are stated on some test problems to prove the efficiency of our proposed approach.

Keywords Presolving, Variable reduction, Convex quadratic problem, Equality and box constraints

AMS 2010 subject classifications 90C20, 90C52, 90C25

DOI: 10.19139/soic-2310-5070-4658

Received: 29 March 2026; **Revised:** 12 September 2026; **Accepted:** 14 September 2026

1. Introduction

Many optimization problems, such as support vector machines [33, 12, 6], differential equations [30], and optimal control [4, 8, 22], can be modeled as convex quadratic programming problems [18, 7, 31]. Small and medium problems can be solved by many approaches, such as active set method [29], adaptive method [13], and interior point method [3]. But for large problems, these approaches will suffer from the complexity and convergence. To improve the performance for learning large-scale problems, the presolving technique is proposed. This process reduces the number of variables before applying a resolution method to the problem. It simplifies the initial problem by fixing some variables at one of their boundaries without affecting the optimality.

Many works on presolving have been studied over the years depending on the formulation of the problem. We find the preprocessing of binary programming problems, which has been proposed in several works [17, 19, 15, 21]. There are also works that presolve integer and mixed-integer programming problems such as in papers [1, 2, 5, 9, 11, 14, 24, 27, 28]. Additionally, linear bilevel optimization problems have been studied in [23]. Finally, convex quadratic programming problems with continuous and integer variables have been presolved in many works [16, 20, 25, 26, 34].

Convex quadratic programming problems (CQPP) with a linear equality and box constraints appear very often in the resolution of support vector machines and sequential quadratic programming methods. This kind of problem is usually large.

*Correspondence to: Samia Djemai (Email: samia.djemai@univ-jijel.dz). Research Unit LaMOS, Department of Mathematics, Faculty of the Exact Sciences, Jijel University, Ouled Aissa BP 98, 18000 Jijel, Algeria

ISSN 2310-5070 (online) ISSN 2311-004X (print)
Copyright © 2026 International Academic Press

In this paper, we propose a presolving procedure in order to reduce the number of variables of a convex quadratic problem. The aim is to find some active indices at the optimum in order to fix in advance the corresponding variables. In addition, our novelty compared with standard methods lies in identifying indices for which the corresponding components of the reduced gradient must be zero at the optimum. This makes it possible to include these indices in the support of the objective function I_S and retain them throughout the solution process of the general algorithm, namely the adaptive support method [12, 25, 6]. This allows for ensuring rapid convergence to the optimum.
Note that in the works [25, 26], the presolving technique is done only with box constraints, and here an equality constraint is added. For the matrix case, the problem is more difficult, since it requires establishing an upper bound for the norm of the inverse of an arbitrary matrix, which constitutes a real challenge [35].
We propose a pre-resolution algorithm [25, 26, 32] that allows reducing the dimension of the problem and starting the resolution algorithm with an appropriate support (basis), which ensures its convergence. For this, the optimal reduced gradient bounding technique will be used. In this work, only the non-degenerate case has been considered.

In order to test our algorithm and to confirm its effectiveness, a comparative study of the proposed method is performed with the Active Set and Interior Point Quadprog algorithms of the Matlab Optimization Toolbox. The obtained results show that the proposed presolving algorithm is competitive and significantly improves the computational efficiency.

This paper is organized as follows: in Section 2, we present the formulation of the problem and review some basic definitions. In Section 3, we give the proposed presolving procedure, allowing us to reduce the size of the problem. Two numerical examples for illustration purposes are presented in Section 4. In Section 5 we have provided a comparative experimental study of the proposed presolving procedure with the Active Set Method (ASM) and the Interior Point Method (IPM) implemented under Matlab. Finally, in the last section, we conclude the paper and give some perspectives.

2. Formulation of the problem and definitions

We consider the following quadratic minimization problem:

$$\begin{cases} \min\limits_{x} F(x) = \frac{1}{2}x'Dx \ - c'x, \\ \qquad A'x = b, \\ \qquad 0 \leq x \leq u, \end{cases} \tag{1}$$

where $x, c, u \in \mathcal{R}^n$; $b \in \mathcal{R}$; A is an n-column vector with $A = (a_i, i \in I)$, $a_i \in \mathcal{R}$, $a_i \neq 0$ for $i \in I = \{1, 2, ..., n\}$. The matrix $D = D(I, I) = (d_{ij}, i \in I, j \in I)$ is square of order n, symmetric, and positive semidefinite.

At an optimal solution $x^* = (x_i^*, i \in I)$, some variables x_i^* are zero or u_i, the others verify: $0 < x_i^* < u_i$.

In our contribution, we will construct subsets of the set I, noted I_0, I_u and I_S, corresponding respectively to some active variables equal to 0 or u_i and some inactive ones.

3. Presolving for the quadratic convex problem

For the problem (1), we consider the following partitions:

$$I = I_B \cup I_N, \ I_B \cap I_N = \emptyset, \ I_N = I_S \cup I_{NN}, \ I_S \cap I_{NN} = \emptyset.$$

3.1. Definitions

- The singleton $I_B \subset I$ such as $I_B = \{i_B\}, i_B \in I$, is called a *constraint support* and $A_B = A(I_B) = a_{i_B}, (a_{i_B} \neq 0)$, is called the basis, with $A_B^{-1} = \frac{1}{a_{i_B}}$.
- The subset $I_S \subset I_N$ is called a *support of the objective function* of the problem (1) if the submatrix $M_S = M(I_S, I_S)$ of M is non-singular, where $M = M(I_N, I_N) = Z'DZ$, $Z = \begin{pmatrix} -A_B^{-1}A_N \\ J_N \end{pmatrix}$, J_N is the identity matrix of dimension $(n-1)$ and $A' = (A_B, A_N)$, with $A_N = A(I_N) = (a_i, i \in I_N) \in \mathcal{R}^{1\times(n-1)}$.
- The pair $I_P = \{I_B, I_S\}$ is called a *support of problem* (1). The couple $\{x, I_P\}$ formed from a feasible solution x and a problem support I_P, is called a *Support Feasible Solution* (SFS) of the problem (1). The constraint support feasible solution $\{x, I_B\}$ is said to be *non-degenerate* if $0 < x_i < u_i, i \in I_B$.
- The vector $g = g(x) = Dx - c$ is the gradient of the objective function at x, with $g = (g_B, g_N), g_B = g(I_B), g_N = g(I_N)$.
- The scalar $\pi = g_B A_B^{-1} = \frac{g_B}{a_{i_B}}$ is called a potential; the vector $E = g - \pi A$ is the reduced gradient vector or the vector of estimations, with $E_{i_B} = 0$.
- The SFS $\{x, I_P\}$ is called *consistent* if $E_S = E(I_S) = (E_i, i \in I_S) = 0$.

3.2. Optimality criterion

Let $\{x, I_B\}$ be a constraint SFS of problem (1). Then the relations

$$\begin{cases} E_i \geq 0, & if \quad x_i = 0 \quad ; \\ E_i \leq 0, & if \quad x_i = u_i \quad ; \\ E_i = 0, & if \quad 0 < x_i < u_i, \quad i \in I_N \quad , \end{cases} \tag{2}$$

are sufficient, and in the non-degeneracy case of $\{x, I_B\}$, also necessary for the optimality of the feasible solution x [13].

3.3. The Presolving Procedure

Before proceeding to construct an algorithm for solving the problem (1), we will do a preprocessing step that allows us to find some active indices at the optimum in order to fix in advance the corresponding variables. We also look for other variables, whose corresponding components of the reduced gradient will be null at the optimum. The indices of the latters will be incorporated in I_S.
Then, the presolving technique is based on the following majoration and minoration:

$$g^- \leq g \leq g^+, \quad \pi^- \leq \pi \leq \pi^+, \quad E^- \leq E \leq E^+.$$

From the matrix D, we define the following two matrices:

$$D^+ := (d_{ij}^+, i, j \in I), \quad D^- := (d_{ij}^-, i, j \in I),$$

with

$$d_{ij}^+ = \begin{cases} d_{ij}, & if \quad d_{ij} \geq 0 \quad , \\ 0, & if \quad d_{ij} < 0 \quad ; \end{cases} \tag{3}$$

$$d_{ij}^- = \begin{cases} d_{ij}, & if \quad d_{ij} \leq 0 \quad , \\ 0, & if \quad d_{ij} > 0 \quad . \end{cases} \tag{4}$$

Then, we calculate the vectors:

$$v^+(I) = (v_i^+, i \in I) := D^+u, \quad v^-(I) = (v_i^-, i \in I) := D^-u.$$

For any feasible solution x of the problem (1), we have the double inequality:

$$g^- = v^- - c \leq g(x) \leq v^+ - c = g^+, \tag{5}$$

i.e,

$$g_i^- = v_i^- - c_i \leq g_i \leq v_i^+ - c_i = g_i^+, \quad i \in I.$$

Since $\pi = g_B A_B^{-1}$, we can deduce:

$$\pi = g_B A_B^{-1} = \frac{g_{i_B}}{a_{i_B}} \Rightarrow E_i = g_i - \pi a_i = g_i - \frac{g_{i_B}}{a_{i_B}} a_i, \quad i \in I.$$

So, we have:

$$\begin{cases} E_i = g_i - \frac{g_{i_B}}{a_{i_B}} a_i \leq g_i^+ + \frac{g_{i_B}}{a_{i_B}}(-a_i), \\ E_i = g_i - \frac{g_{i_B}}{a_{i_B}} a_i \geq g_i^- + \frac{g_{i_B}}{a_{i_B}}(-a_i). \end{cases} \tag{6}$$

Also, we can write:

$$g_{i_B}^- \leq g_{i_B} \leq g_{i_B}^+ \implies \begin{cases} \frac{g_{i_B}^-}{a_{i_B}} \leq \frac{g_{i_B}}{a_{i_B}} \leq \frac{g_{i_B}^+}{a_{i_B}}, & \text{if } a_{i_B} > 0, \\ \frac{g_{i_B}^+}{a_{i_B}} \leq \frac{g_{i_B}}{a_{i_B}} \leq \frac{g_{i_B}^-}{a_{i_B}}, & \text{if } a_{i_B} < 0, \end{cases}$$

where i_B is an index of I such as $a_{i_B} \neq 0$. Therefore, we will pose

$$\pi^- = \min\left\{\frac{g_{i_B}^-}{a_{i_B}}, \frac{g_{i_B}^+}{a_{i_B}}\right\}, \quad \pi^+ = \max\left\{\frac{g_{i_B}^+}{a_{i_B}}, \frac{g_{i_B}^-}{a_{i_B}}\right\},$$

then the inequality holds:

$$\pi^- \leq \pi = \frac{g_{i_B}}{a_{i_B}} \leq \pi^+.$$

So, the relations (6) give:

$$E_i \leq g_i^+ + \frac{g_{i_B}}{a_{i_B}}(-a_i) \leq \begin{cases} g_i^+ - \pi^- a_i & \text{if } a_i > 0, \\ g_i^+ - \pi^+ a_i & \text{if } a_i < 0, \end{cases}$$

$$E_i \geq g_i^- + \frac{g_{iB}}{a_{iB}}(-a_i) \geq \begin{cases} g_i^- - \pi^+ a_i & \text{if } a_i > 0, \\ g_i^- - \pi^- a_i & \text{if } a_i < 0. \end{cases}$$

Hence, we set

$$E^+ = (E_i^+, i \in I), \text{with } E_i^+ = \begin{cases} g_i^+ - \pi^- a_i & \text{if } a_i > 0, \\ g_i^+ - \pi^+ a_i & \text{if } a_i < 0, \end{cases}$$

$$E^- = (E_i^-, i \in I), \text{with } E_i^- = \begin{cases} g_i^- - \pi^+ a_i & \text{if } a_i > 0, \\ g_i^- - \pi^- a_i & \text{if } a_i < 0. \end{cases}$$

In particular, for an optimal solution x^*, we get:

$$E^- \leq E^* \leq E^+. \tag{7}$$

The following lemma allows to fix some components of an optimal solution x^* at their bounds:

Lemma 1
Let $i \in I_N = I \backslash I_B = \{1, 2, ..., n\} \backslash I_B$. So we have:

(i) If $E_i^+ < 0$, then $E_i^* < 0 \Rightarrow x_i^* = u_i$,
(ii) if $E_i^- > 0$, then $E_i^* > 0 \Rightarrow x_i^* = 0$.

Proof

This comes from the optimality relations (2) and the inequality (7). □

The second lemma allows to exclude a component x_i^* to be at a certain bound. For this, we define for all $i \in I$, the following formulas:

$$q_i^- = \begin{cases} d_{ii}u_i + v_i^- - \pi^+ a_i, & \text{if} \quad a_i > 0, \\ d_{ii}u_i + v_i^- - \pi^- a_i, & \text{if} \quad a_i < 0. \end{cases}$$

$$q_i^+ = \begin{cases} -d_{ii}u_i + v_i^+ - \pi^- a_i, & \text{if} \quad a_i > 0, \\ -d_{ii}u_i + v_i^+ - \pi^+ a_i, & \text{if} \quad a_i < 0. \end{cases}$$

Lemma 2
Let $i \in I_N$, then:

(i) $c_i < q_i^- \Rightarrow x_i^* < u_i$;
(ii) $c_i > q_i^+ \Rightarrow x_i^* > 0$.

Proof
(i) Suppose that $x_i^* = u_i$. So $E_i^* \leq 0$ according to the optimality criterion (2). We define for all $i \in I$, the sets:

$$I_i^+ = \{j \in I : j \neq i \quad \text{and} \quad d_{ij} \geq 0\}, \quad I_i^- = \{j \in I : j \neq i \quad \text{and} \quad d_{ij} < 0\}.$$

So we have

$$v_i^+ = d_{ii}u_i + \sum_{j \in I_i^+} d_{ij}u_j, \quad v_i^- = \sum_{j \in I_i^-} d_{ij}u_j.$$

So, we will have:

$$E_i^* = g_i - \pi a_i = d_{ii}u_i + \sum_{j \in I_i^+} d_{ij}x_j^* + \sum_{j \in I_i^-} d_{ij}x_j^* - c_i - \pi a_i,$$

and

$$E_i^* \geq \begin{cases} d_{ii}u_i + v_i^- - \pi^+ a_i - c_i, & \text{if } a_i > 0, \\ d_{ii}u_i + v_i^- - \pi^- a_i - c_i, & \text{if } a_i < 0, \end{cases}$$

which implies

$$E_i^* \geq q_i^- - c_i.$$

According to the hypothesis (i), we obtain $E_i^* > 0$, which brings to a contradiction. Therefore, we must have $x_i^* < u_i$.
(ii) Suppose that $x_i^* = 0$. According to the optimality criterion (2), we have $E_i^* \geq 0$. On the other hand, we can write:

$$E_i^* = \sum_{j \in I_i^+} d_{ij}x_j^* + \sum_{j \in I_i^-} d_{ij}x_j^* - c_i - \pi a_i,$$

$$E_i^* \leq \begin{cases} -d_{ii}u_i + v_i^+ - \pi^- a_i - c_i, & \text{if} \quad a_i > 0, \\ -d_{ii}u_i + v_i^+ - \pi^+ a_i - c_i, & \text{if} \quad a_i < 0, \end{cases}$$

which implies

$$E_i^* \leq q_i^+ - c_i.$$

The hypothesis (ii) implies that $E_i^* < 0$, which contradicts the inequality $E_i^* \geq 0$. Therefore, we must have $x_i^* > 0$. □

The third lemma predicts that for some index $i \in I_N$, the component E_i^* of the reduced gradient of an optimal solution x^* must be null.

Lemma 3
Let $i \in I_N$. If $q_i^+ < c_i < q_i^-$, then $E_i^* = 0$.

Proof
If $q_i^+ < c_i < q_i^-$, the application of the Lemma 2 will imply that $x_i^* > 0$ and $x_i^* < u_i$. So $0 < x_i^* < u_i$, and therefore $E_i^* = 0$. □

Remark 1
The condition $q_i^+ < q_i^-$ occurs if and only if:

$$d_{ii}u_i - \sum_{j \neq i} |d_{ij}|u_j > S_i, \quad \text{with} \quad S_i = |a_i|(\pi^+ - \pi^-).$$

Indeed, we have for $a_i < 0$

$$q_i^+ = v_i^+ - u_i d_{ii} + \pi^+|a_i| = \sum_{i \in I_i^+} d_{ij}u_j + \pi^+|a_i|,$$

$$q_i^- = v_i^- + u_i d_{ii} + \pi^-|a_i| = u_i d_{ii} + \sum_{i \in I_i^-} d_{ij}u_j + \pi^-|a_i|,$$

so

$$q_i^+ - q_i^- < 0 \Leftrightarrow$$

$$\sum_{i \in I_i^+} d_{ij}u_j - \sum_{i \in I_i^-} d_{ij}u_j - u_i d_{ii} + \pi^+|a_i| - \pi^-|a_i| < 0.$$

Thus, we get

$$d_{ii}u_i - \sum_{i \in I_i^+} d_{ij}u_j + \sum_{i \in I_i^-} d_{ij}u_j > |a_i|(\pi^+ - \pi^-),$$

$$\Leftrightarrow d_{ii}u_i - \sum_{j \neq i} |d_{ij}|u_j > |a_i|(\pi^+ - \pi^-) = S_i.$$

so,

$$q_i^+ < q_i^- \Leftrightarrow d_{ii}u_i - \sum_{j \neq i} |d_{ij}|u_j > S_i,$$

and we obtain the same inequality if $a_i > 0$. □
Thus, we deduce

$$q_i^+ < c_i < q_i^- \Rightarrow E_i^* = 0 \Rightarrow i \in I_S^*.$$

Thanks to the Lemmas 1,2 and 3, we define the following subsets of I:

$$\begin{gathered} I_0 := \{i \in I_N : E_i^- > 0\}; \\ I_u := \{i \in I_N : E_i^+ < 0\}; \\ I_S := \{i \in I_N : q_i^+ < c_i < q_i^-\}; \\ \bar{I} := I_B \cup \bar{I}_N; \bar{I}_N = I_N \backslash (I_0 \cup I_u); \bar{I}_N = I_S \cup \bar{I}_{NN}. \end{gathered} \tag{8}$$

Then we have

$$x_i^* = \begin{cases} 0, \quad if \quad i \in I_0 \quad ; \\ u_i, \quad if \quad i \in I_u \quad ; \\ 0 < x_i^* < u_i \ (E_i^* = 0), \quad if \quad i \in I_S \quad . \end{cases} \tag{9}$$

Consequently, we have obtained the following reduced quadratic problem:

$$I := \bar{I}, n := |\bar{I}|, D := D(\bar{I}, \bar{I}), x(I) := x(\bar{I}), \bar{c}(\bar{I}) = c(\bar{I}) - D(\bar{I}, I_0 \cup I_u)x^*(I_0 \cup I_u).$$

The equality constraint still needs to be updated:

$$A'x = b \Leftrightarrow A'(\bar{I})x(\bar{I}) + A'(I_0 \cup I_u)x^*(I_0 \cup I_u) = b,$$

We set $\bar{\gamma} = A'(I_0 \cup I_u)x^*(I_0 \cup I_u)$. So

$$A'(\bar{I})x(\bar{I}) = \bar{b} = b - \bar{\gamma}.$$

After that, this reduced problem can be solved by applying the adaptive method [12, 10] that starts with an appropriate initial support of the objective function I_S defined by (8).
Let x_r^* be its solution. So, the solution of the original problem is:

$$x^* = (x_r^*, x^*(I_0 \cup I_u)).$$

4. Numerical examples

4.1. Numerical example 1

We consider the following convex quadratic minimization problem:

$$\begin{cases} \min\limits_x \quad F(x) = \frac{1}{2}x'Dx \ - c'x, \\ \\ \qquad A'x = b, \\ \\ \qquad 0 \le x \le u, \end{cases} \tag{10}$$

with

$$D = \begin{pmatrix} 8 & 1 & 0 & 0 & 0 \\ 1 & 6 & 1 & 0 & 0 \\ 0 & 1 & 5 & 1 & 0 \\ 0 & 0 & 1 & 10 & 1 \\ 0 & 0 & 0 & 1 & 20 \end{pmatrix}, \quad c = \begin{pmatrix} 44 \\ 26 \\ -18 \\ 5 \\ 4 \end{pmatrix}, \quad u = e = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad x^* = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 96/199 \\ 35/199 \end{pmatrix},$$

$$A' = (-2, -3, 1, 10, 1), \quad b = 0.$$

We set

$$I_B = \{4\}, \quad I_N = \{1, 2, 3, 5\},$$

then we demonstrate that the SRS $\{x^*, I_B\}$ is an optimal solution of (10).
Indeed, we will have:

$$A_B = A(I_B) = a_{i_B} = a_4 = 10 \Rightarrow A_B^{-1} = \frac{1}{a_{i_B}} = \frac{1}{10},$$

and

$$g(x^*) = Dx^* - c = \begin{pmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \end{pmatrix} = \begin{pmatrix} -35 \\ -19 \\ 3877/199 \\ 0 \\ 0 \end{pmatrix} \Rightarrow g_B = g_4 = 0, g_N = \begin{pmatrix} g_1 \\ g_2 \\ g_3 \\ g_5 \end{pmatrix} = \begin{pmatrix} -35 \\ -19 \\ 3877/199 \\ 0 \end{pmatrix}.$$

So, $\pi = g_B A_B^{-1} = 0 \implies E = g - \pi A = g$.
Therfore, we have a support feasible solution $\{x^*, I_B\}$ such as:

- $x_1^* = 1 \implies E_1^* = g_1 - \pi a_1 = -35 < 0$ (optimal);
- $x_2^* = 1 \implies E_2^* = g_2 - \pi a_2 = -19 < 0$ (optimal);
- $x_3^* = 0 \implies E_3^* = g_3 - \pi a_3 = 19.4824$ (optimal);
- $x_5^* = 35/199 \implies E_5^* = g_5 - \pi a_5 = 0$ (optimal).

By the definition of the reduced gradient, we have

$$E^*(I_B) = E_4^* = 0 \quad \text{for} \quad x_4^* = \frac{96}{199},$$

and the optimality criterion is therefore verified, with $F(x^*) = \frac{-12648}{199} \simeq -63.5578$.

To execute the presolving procedure, we take

$$I_B = \{4\}, I_N = I \setminus I_B = \{1, 2, 3, 5\},$$

then we apply the different lemmas to set certain variables at the optimum, or conclude that the corresponding component of the reduced gradient is zero at the optimum.
Therefore, we determine:

$$D^+ = D, \quad D^- = 0,$$

$$v^+(I) = (v_i^+, i \in I) = D^+ u = \begin{pmatrix} 9 \\ 8 \\ 7 \\ 12 \\ 21 \end{pmatrix}, \quad v^- = (v_i^-, i \in I) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

$$g^- = v^- - c = \begin{pmatrix} -44 \\ -26 \\ 18 \\ -5 \\ -4 \end{pmatrix}, \quad g^+ = v^+ - c = \begin{pmatrix} -35 \\ -18 \\ 25 \\ 7 \\ 17 \end{pmatrix}.$$

So, we calculate E^+ and E^- for $I_B = i_B = 4$. Then

$$\pi^- = \frac{g_4^-}{a_4} = \frac{-1}{2}, \quad \pi^+ = \frac{g_4^+}{a_4} = \frac{7}{10} \implies -\frac{1}{2} \leq \pi = \frac{g_{i_B}}{a_{i_B}} \leq \frac{7}{10}.$$

Consequently, we will have

$$\begin{cases} a_1 = -2 \implies E_1^- = -45 < 0, & \text{and} \quad E_1^+ = -33.6 < 0, \\ a_2 = -3 \implies E_2^- = -27.5 < 0, & \text{and} \quad E_2^+ = -15.9 < 0, \\ a_3 = 1 \implies E_3^- = 17.3 > 0, & \text{and} \quad E_3^+ = 25.5 > 0, \\ a_4 = 10 \implies E_4^- = -12 < 0, & \text{and} \quad E_4^+ = 12 > 0, \\ a_5 = 1 \implies E_5^- = -4.5 < 0, & \text{and} \quad E_5^+ = 17.5 > 0. \end{cases} \tag{11}$$

From the first lemma, we conclude that

$$\begin{cases} E_1^+ < 0 \implies x_1^* = 1, \\ E_2^+ < 0 \implies x_2^* = 1, \\ E_3^- > 0 \implies x_3^* = 0. \end{cases} \tag{12}$$

The second and third lemmas allow us to predict that the component E_5 must be zero.
We calculate

$$a_5 = 1 \implies \begin{cases} q_5^- = d_{55}u_5 + v_5^- - \pi^+ a_5 = 19.3, \\ q_5^+ = -d_{55}u_5 + v_5^+ - \pi^- a_5 = 1.5. \end{cases}$$

So, we have

$$q_5^+ = 1.5 < c_5 = 4 < 19.3 = q_5^- \quad \implies E_5^* = 0.$$

It remains to determine the values of x_4^* and x_5^*.
Therefore, we fix: $x_1^* = x_2^* = 1$, $x_3^* = 0$, and solve the following system of equations, relatively to the variables π^* and x_5^*.

$$\begin{cases} E_4^* = g_4^* - \pi^* a_4 = 0 \\ E_5^* = g_5^* - \pi^* a_5 = 0 \end{cases} \Rightarrow \begin{cases} \pi^* = \frac{1}{a_4} g_4^* \\ E_5^* = g_5^* - \frac{1}{10} g_4^* a_5 = 0. \end{cases}$$

where

$$\begin{cases} g_4^* = x_3^* + 10x_4^* + x_5^* - 5 = 10x_4^* + x_5^* - 5, \\ g_5^* = x_4^* + 20x_5^* - 4. \end{cases}$$

So

$$E_5^* = \cancel{x_4^*} + 20x_5^* - 4 - \cancel{x_4^*} - \frac{1}{10}x_5^* + \frac{1}{2} = 0 \Rightarrow \frac{199}{10}x_5^* = \frac{7}{2} \Rightarrow x_5^* = \frac{35}{199}.$$

The equation $E_4^* = 0$ is always satisfied because it defines the potential $\pi^* = \frac{1}{a_4} g_4^*$. Therefore, it is redundant with respect to the variable x_4^*. Consequently, the latter will be deduced from the equation $A'x^* = b$, i.e.,

$$-2x_1^* - 3x_2^* + x_3^* + 10x_4^* + x_5^* = 0 \Rightarrow -2 - 3 + 0 + 10x_4^* + \frac{35}{199} = 0 \Rightarrow x_4^* = \frac{96}{199}.$$

Therefore, the optimal solution of the initial problem is $x^* = \left(1, 1, 0, \frac{96}{199}, \frac{35}{199}\right)$ with $F(x^*) = \frac{-12648}{199} \simeq -63.5578$.

Remark 2
This procedure can be repeated until the sets I_0, I_u and I_S become constant.

4.2. Numerical example 2

Let us consider the following quadratic program:

$$D = \begin{pmatrix} 4 & 0 & 1 & -2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 1 & 1 & 4 & -2 & 1 & 0 & 0 & -2 & 1 & -1 \\ -2 & 0 & -2 & 4 & 0 & 0 & -1 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 4 & 1 & -1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 3 & 0 & 2 & 0 & -1 \\ 0 & 0 & 0 & -1 & -1 & 0 & 5 & 1 & 1 & 2 \\ 0 & 0 & -2 & 2 & 0 & 2 & 1 & 4 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & -1 & -1 & 0 & 0 & -1 & 2 & 0 & 1 & 6 \end{pmatrix}, \quad c = \begin{pmatrix} 32 \\ -30 \\ -19 \\ 76 \\ -35 \\ 113 \\ -1 \\ 48 \\ 23 \\ 174 \end{pmatrix}, \quad u = \begin{pmatrix} 9 \\ 5 \\ 5 \\ 4 \\ 5 \\ 14 \\ 4 \\ 8 \\ 6 \\ 19 \end{pmatrix}.$$

$$A = \begin{pmatrix} 6, & -5, & 3, & 1, & -6, & 7, & -6, & 1, & -4, & 3 \end{pmatrix}, \quad b = 197,$$

Algorithm 1: Presolving Procedure

Inputs : D, c, A, u;
Outputs : The reduced problem $D_r, \widetilde{c}_r, A_r, \widetilde{b}_r, u_r, \overline{I}$;
I_0, I_u and I_S.
Begin
❶ **Initialization** : $I_0 = I_u = I_S = \emptyset; n = length(u); \overline{I} = I = \{1, \cdots, n\}, k = 1;$

❷ Apply Lemma 1 to determine $I_0^{(k)}$ and $I_u^{(k)}$ of the current problem.

If $I_0^{(k)} \cup I_u^{(k)} \neq \emptyset$, then:

Update the subsets I_0, I_u and $\overline{I}$ as follows:

$$I_0 = I_0 \cup I_0^{(k)}, I_u = I_u \cup I_u^{(k)}, \overline{I} = I \setminus \{I_0 \cup I_u\}.$$

Deduce the reduced problem $D_r, \widetilde{c}_r, A_r, \widetilde{b}_r, u_r$ and go to ❷ with $k := k + 1$.

Otherwise, go to ❸.

❸ Apply Lemma 3 to determine I_S for the final reduced problem obtained in step ❷.

End.

where the optimal solution is :

$$X^* = \begin{pmatrix} 6 \\ 0 \\ 2 \\ 4 \\ 0 \\ 14 \\ 0 \\ 4 \\ 2 \\ 19 \end{pmatrix}, \quad F^* = -4163, \text{ and } E^* = \begin{pmatrix} 0 \\ 27 \\ 0 \\ -68 \\ 51 \\ -76 \\ 41 \\ 0 \\ 0 \\ -74 \end{pmatrix}.$$

$$I_0^* = \{2, 5, 7\}; \quad I_u^* = \{4, 6, 10\} \text{ and } I_S^* = \{1, 3, 8, 9\}.$$

For all iterations, we set $I_B = \{1\}$.

Iteration 01 : $k = 1$

$$E^- = \begin{pmatrix} -63.00 \\ -22.33 \\ -35.50 \\ -111.83 \\ -9.00 \\ -158.83 \\ -48.00 \\ -61.83 \\ -49.66 \\ -209.50 \end{pmatrix}, \qquad E^+ = \begin{pmatrix} 63.00 \\ 78.16 \\ 84.00 \\ -37.33 \\ 97.00 \\ 10.66 \\ 96.00 \\ 30.66 \\ 26.33 \\ -26.00 \end{pmatrix}.$$

The application of the pre-solving procedure allows us to fix two components of the optimal solution x_4^* and x_{10}^* at their upper bounds as follows :

$$I_0 = \emptyset; I_u = \{4, 10\}; \overline{I} = \{1, 2, 3, 5, 6, 7, 8, 9\}.$$

The reduced problem :

$$D_r = \begin{pmatrix} 4 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 4 & 1 & 0 & 0 & -2 & 1 \\ 0 & 0 & 1 & 4 & 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 3 & 0 & 2 & 0 \\ 0 & 0 & 0 & -1 & 0 & 5 & 1 & 1 \\ 0 & 0 & -2 & 0 & 2 & 1 & 4 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}, \quad \widetilde{c}_r = \begin{pmatrix} 40 \\ -11 \\ 8 \\ -35 \\ 132 \\ -35 \\ 40 \\ 4 \end{pmatrix}, \quad u_r = \begin{pmatrix} 9 \\ 5 \\ 5 \\ 5 \\ 14 \\ 4 \\ 8 \\ 6 \end{pmatrix},$$

$$A_r = \begin{pmatrix} 6, & -5, & 3, & -6, & 7, & -6, & 1, & -4 \end{pmatrix}, \quad \widetilde{b}_r = 136$$

Iteration 02 : $k = 2$

$$E^- = \begin{pmatrix} -55.00 \\ -22.33 \\ -31.50 \\ -9.00 \\ -149.50 \\ -10.00 \\ -52.50 \\ -30.66 \end{pmatrix}, \qquad E^+ = \begin{pmatrix} 55.00 \\ 52.50 \\ 57.00 \\ 89.00 \\ -8.33 \\ 84.00 \\ 30.66 \\ 21.00 \end{pmatrix}.$$

The application of the presolving procedure to the reduced problem allows us to fix one other component x_6^* at his upper bound:

$$I_0 = I_0 = \emptyset; I_u = I_u \cup \{6\} = \{4, 6, 10\}; \overline{I} = \{1, 2, 3, 5, 7, 8, 9\}.$$

The reduced problem:

$$D_r = \begin{pmatrix} 4 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 4 & 1 & 0 & -2 & 1 \\ 0 & 0 & 1 & 4 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 5 & 1 & 1 \\ 0 & 0 & -2 & 0 & 1 & 4 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}, \quad \widetilde{c}_r = \begin{pmatrix} 26 \\ -25 \\ 8 \\ -49 \\ -35 \\ 12 \\ 4 \end{pmatrix}, \quad u_r = \begin{pmatrix} 9 \\ 5 \\ 5 \\ 5 \\ 4 \\ 8 \\ 6 \end{pmatrix},$$

$$A_r = \begin{pmatrix} 6, & -5, & 3, & -6, & -6, & 1, & -4 \end{pmatrix}, \quad \widetilde{b}_r = 38$$

Iteration 03 : $k = 3$

$$E^- = \begin{pmatrix} -41.00 \\ 3.33 \\ -31.50 \\ 19.00 \\ 4.00 \\ -24.50 \\ -21.33 \end{pmatrix}, \qquad E^+ = \begin{pmatrix} 41.00 \\ 52.50 \\ 50.00 \\ 89.00 \\ 84.00 \\ 28.33 \\ 21.00 \end{pmatrix}.$$

The application of the presolving procedure to the reduced problem allows us to fix three other components at their lower bound (0):

$$I_0 = I_0 \cup \{2, 5, 7\} = \{2, 5, 7\}; I_u = I_u = \{4, 6, 10\}; \overline{I} = \{1, 3, 8, 9\}.$$

We observe that the presolving procedure detects all binding variables at the optimum. The final reduced problem obtained by applying the preprocessing is:

$$D_r = \begin{pmatrix} 4 & 1 & 0 & 0 \\ 1 & 4 & -2 & 1 \\ 0 & -2 & 4 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}, \quad \widetilde{c}_r = \begin{pmatrix} 26 \\ 8 \\ 12 \\ 4 \end{pmatrix}, \quad u_r = \begin{pmatrix} 9 \\ 5 \\ 8 \\ 6 \end{pmatrix},$$

$$A_r = \begin{pmatrix} 6, & 3, & 1, & -4 \end{pmatrix}, \quad \widetilde{b}_r = 0. \text{ where } \overline{I} = \{1, 3, 8, 9\}$$

The third lemma allows us to predict that the component E_8^* (third position in the reduced problem) must be zero as follows:

$$q^- = \begin{pmatrix} 21.00 \\ -3.50 \\ 19.50 \\ -11.33 \end{pmatrix}, \qquad q^+ = \begin{pmatrix} 31.00 \\ 28.00 \\ 4.33 \\ 15.00 \end{pmatrix}, \text{ where } \overline{I} = \{1, 3, 8, 9\}.$$

We have $q_8^+ = 4.33 < \widetilde{c}_8 = 12 < q_8^+ = 19.5 \Rightarrow E_8^* = 0 \Rightarrow I_S = \{8\}$.

5. Experimental results and discussions

This section reports the experimental results carried out on a laptop with an Intel(R) Core(TM) i3-2377M CPU at 1.5 GHz and 4 GB of RAM, running MATLAB version 9.10.0 R2021a. To assess the effectiveness of our presolving technique, we compared the iteration count and execution time of the original and reduced problems, using both the Active-Set and Interior-Point convex algorithms available in the MATLAB Optimization Toolbox's quadprog function. Additionally, we developed a dedicated test example generator that enables a direct comparison between the solution produced by our method and the known exact solution. So we set:

- ASM : Active Set Method.
- IPM : Interior Point Method.
- CPU : Average execution time (CPU) of test problems (in seconds).
- Iter : Average number of iterations required to solve test problems.

The obtained results are presented in the following tables and figures:

Table 1. Comparative Results for Active-set Method (ASM)

	ASM		ASM with Presolving	
n	Iter	CPU	Iter	CPU
10	11.20	0.0623	0.50	0.0320
20	22.10	0.0837	0.90	0.0565
30	35.10	0.1002	1.80	0.0850
40	46.25	0.1785	3.60	0.1003
50	58.30	0.3283	5.60	0.1930
80	94.50	0.8972	10.10	0.3554
100	118.40	1.2832	14.50	0.7928
200	235.80	7.8853	92.20	1.1934
300	372.50	23.5524	161.20	3.5572
400	496.00	73.6007	226.10	18.8830
500	601.80	164.7736	375.00	92.2779

Figure 1. Comparative Results for Active-set Method (ASM)

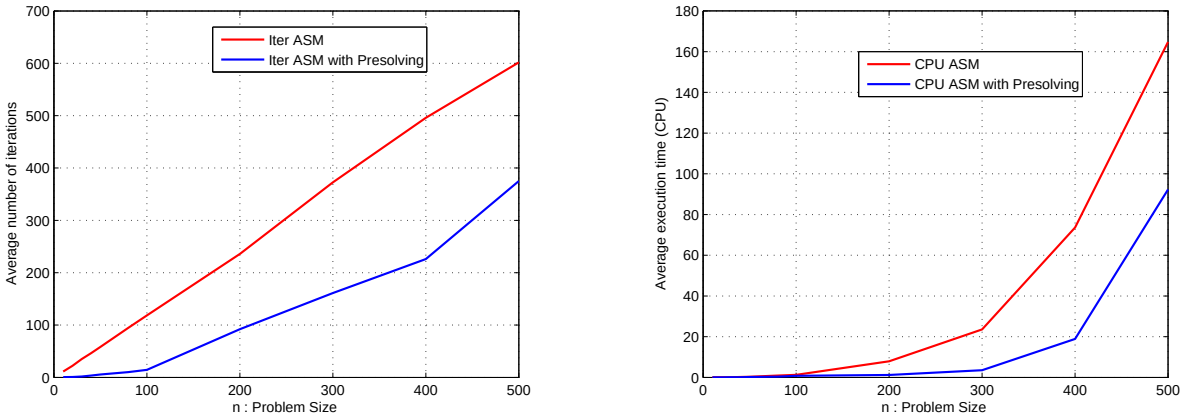
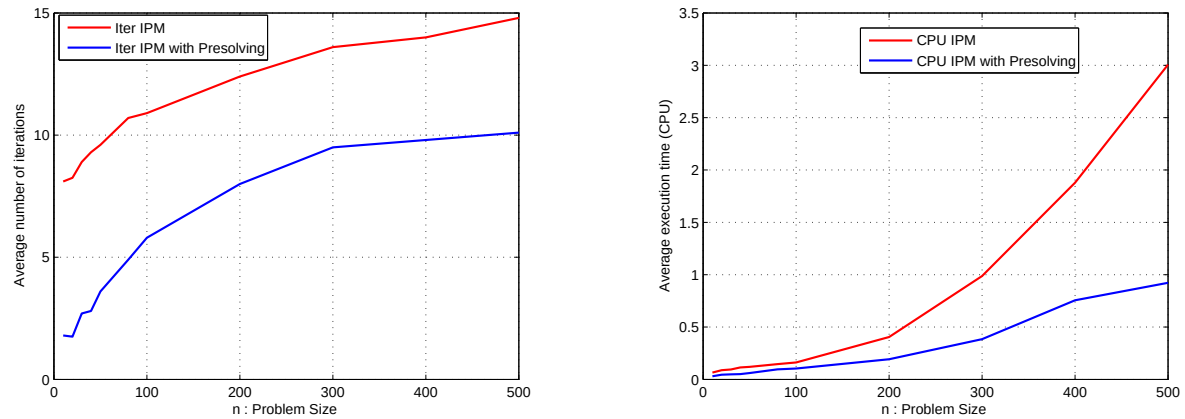


Table 2. Comparative Results for Interior Points Method (IPM)

	IPM		IPM with Presolving	
n	Iter	CPU	Iter	CPU
10	8.10	0.0653	1.80	0.0298
20	8.25	0.0885	1.75	0.0445
30	8.90	0.0945	2.70	0.0486
40	9.30	0.1145	2.80	0.0508
50	9.60	0.1200	3.60	0.0612
80	10.70	0.1454	4.90	0.0956
100	10.90	0.1620	5.80	0.1035
200	12.40	0.4043	8.00	0.1920
300	13.60	0.9875	9.50	0.3846
400	14.00	1.8784	9.80	0.7554
500	14.80	3.0065	10.10	0.9225

Figure 2. Comparative Results for Interior Points Method (IPM)



The numerical results consistently demonstrate that, by incorporating our presolving procedure before the resolution with both the Active Set Method and the Interior Point Method of the Matlab Optimization Toolbox, we have significantly reduced the overall number of iterations while maintaining competitive CPU time.
These results confirm that the proposed technique is an efficient and robust alternative for reducing the problem size and calculation effort by bounding multiple variables and also speeding the convergence of the resolution.

6. Conclusion

In this paper, we have proposed a preprocessing procedure in order to reduce the complexity of the resolution algorithm, to speed up convergence, and to improve its performance for learning large problems. This preprocessing is based on bounding the optimal reduced gradient. The indices corresponding to zero components of the reduced gradient at the optimum are introduced for the first time [25]. This presolving procedure is simple to implement and does not require extensive computational effort. Its success rate simply depends on the nature of the problems being considered. The numerical results indicate the practical performance of the presolving technique.
In a future work, by incorporating this presolving, the resulting reduced problem will be solved by applying the adaptive method [12, 10, 6] to construct an optimal solution of the original SVM problem.

Acknowledgments The authors are grateful to the editor and reviewers for useful suggestions, which improved the contents of this paper.

REFERENCES

1. T. Achterberg, R.E. Bixby, Z. Gu, E. Rothberg, and D. Weninger, *Presolve reductions in mixed integer programming*, INFORMS Journal on Computing, vol. 32, no. 2, pp. 473–506, 2020.
2. E.D. Andersen , and K.D. Andersen, *Presolving in linear programming*, Mathematical Programming, vol. 71, pp. 221–245, 1995.
3. A. Arbel, *Exploring interior-point linear programming: algorithms and software*, MIT Press, Cambridge, MA , 1993.
4. I. Alioua,and M. O. Bibi, *Method for solving the norm in an optimal control problem*, Numerical Algebra, Control and Optimization, vol. 15, no. 4, pp. 1066–1091, 2025.
5. D.A. Babayev, and S.S. Mardanov, *Reducing the number of variables in integer and linear programming problems*, Computational Optimization and Applications, vol. 3, no. 2, pp. 99–109, 1994.
6. L. Belkhiri-Brahmi, B. Brahmi, and M. O. Bibi, *An Adaptive Support Method for Training SVM Problems*, Statistics, Optimization and Information Computing, 16(2), 1447–1465, 2026.
7. M. Bentobache and M. O. Bibi, *Numerical Methods of Linear and Quadratic Programming*, French Academic Press, Germany, 2016 (in French).
8. M. O. Bibi, *Methods for solving linear-quadratic problems of optimal control*, PhD thesis, Minsk University, 1985.
9. M.O. Bibi, H. Boussouira, and A. Laouar, *Solution of an integer linear programming problem via a primal dual method combined with a heuristic*, International Journal of Mathematical Modelling and Numerical Optimisation, vol. 12, no. 1, pp. 69–87, 2022.
10. A. Bounceur, S. Djemai, B. Brahmi, M.O. Bibi, and R. Euler, *A classification approach for an accurate analog/RF BIST evaluation based on the process parameters*, J. Electron. Test. , vol. 34, no. 3, pp. 321-–335, 2018.
11. S.S. Brito, and H.G. Santos, *Preprocessing and cutting planes with conflict graphs*, Computers & Operations Research, vol. 128, pp. 105–176, 2021.
12. S. Djemai, B. Brahmi, and M.O. Bibi, *A primal-dual method for SVM training*, Neurocomputing, vol. 211, pp. 34-–40, 2016.
13. R. Gabasov, F.M. Kirillova, and A.I. Tyatyushkin, *Constructive Methods of Optimization*, University Press, Minsk , 1984. (In Russian)
14. G. Gamrath, T. Koch, A. Martin, M. Miltenberger, and D. Weninger, *Progress in presolving for mixed integer programming*, Mathematical Programming Computation, vol. 7, pp. 367–398,2015.
15. A. Gleixner, L. Gottwald, and A. Hoen, *PaPILO: A Parallel Presolving Library for Integer and Linear Optimization with Multiprecision Support*, INFORMS Journal on Computing , vol. 35, no. 6, pp. 1329–1341, 2023.
16. N. Gould, and P.L. Toint, *Preprocessing for quadratic programming*, Mathematical Programming, vol. 100, pp. 95–132, 2004.
17. M. Guignard, and K. Spielberg, *Logical reduction methods in zero-one programming—minimal preferred variables*, Operations Research, vol. 29, no. 1, pp. 49–74, 1981.
18. K. Hassaini, and M. O. Bibi, *An Algorithm for Solving Quadratic Programming Problems with an M-matrix*, Statistics, Optimization and Information Computing, vol. 12, no. 2, pp. 405–417, 2024.
19. K.L. Hoffman, and M. Padberg, *Improving LP-representations of zero-one linear programs for branch-and-cut*, ORSA Journal on Computing, vol. 3, no. 2, pp. 121–134, 1991.
20. Z. Hua, B. Zhang, and X. Xu, *A new variable reduction technique for convex integer quadratic programs*, Applied mathematical modelling, vol. 32, no. 2, pp. 224–231, 2008.
21. E.L. Johnson, and U.H. Suhl, *Experiments in integer programming*, Discrete Applied Mathematics, vol. 2, no. 1, pp. 39–55, 1980.

22. N. Khimoum, and M. O. Bibi, *Primal-dual method for solving a linear-quadratic multi-input optimal control problem*, Optimization Letters, vol. 14, pp. 653-–669, 2020.
23. T. Kleinert, J. Manns, M. Schmidt, and D. Weninger, *Presolving linear bilevel optimization problems*, EURO Journal on Computational Optimization, vol. 9, pp. 100–120, 2021.
24. T. Koch, T. Achterberg, E. Andersen, O. Bastert, T. Berthold, R.E. Bixby, E. Danna, G. Gamrath, A.M. Gleixner, S. Heinz, A. Lodi, H. Mittelmann, T. Ralphs, D. Salvagnin, D.E. Steffy, and K. Wolter, *MIPLIB 2010: mixed integer programming library version 5*, Mathematical Programming Computation, vol. 3, pp. 103–163, 2011.
25. A. Laouar, and M.O. Bibi, *An adaptive hybrid direction algorithm for convex box-QP problems with enhanced presolving*, IJMOR, vol. 32, no. 2, pp. 251–275, 2025.
26. A. Laouar, and M. O. Bibi, *A New Presolving Technique for Convex Box-Constrained Quadratic Programming Problems*, Advances in Nonlinear Variational Inequalities, vol. 28, pp. 106–118, 2025.
27. F. Matter, and M.E. Pfetsch, *Presolving for Mixed-Integer Semidefinite Optimization*, INFORMS Journal on Optimization, vol. 5, no. 2, pp. 131–154, 2022.
28. C.S. Mészáros, and U.H. Suhl, *Advanced preprocessing techniques for linear and quadratic programming*, OR Spectrum, vol. 4, no. 25, pp. 575–595, 2003.
29. J. Nocedal, and S.J. Wright, *Numerical Optimization*. Springer, 1999.
30. S. Pougkakiotis, J. W. Pearson, S. Leveque, and J. Gondzio, *Fast Solution Methods for Convex Quadratic Optimization of Fractional Differential Equations*, SIAM Journal on Matrix Analysis and Applications, vol. 41, no. 3, pp. 1443-–1476, 2020.
31. S. Radjef, and M. O. Bibi, *An Effective Generalization of the Direct Support Method in Quadratic Convex Programming*, Applied Mathematical Programming, vol. 6, no. 31, pp. 1525–1540, 2012.
32. A. Rezzag, M. O. Bibi, and A. Laouar, *Reduction of bounded variables in integer linear programming problems*, IJMOR, vol. 30, no. 2, pp. 251–265, 2025.
33. V.N.Vapnik, *The nature of statistical learning theory*, Information Science and Statistics, Springer-verlag, New York, 1995.
34. M. Zhou, and W. Chen, *Reducing the number of variables in integer quadratic programming problem*, Applied mathematical modelling, vol. 34, no. 2, pp. 424–436, 2010.
35. X. Wang, F. Wang, *Infinity norm upper bounds for the inverse of SDD_k matrices*, AIMS Mathematics, vol. 8, no. 10, pp. 24999-25016, 2023.