A dual method for solving general convex quadratic programs

Belkacem BRAHMI a,b and Mohand Ouamer BIBI a,c

a LAMOS Laboratory, University of Bejaia, 06000 Bejaia, Algeria; b Department of Mathematics, University of Tizi-Ouzou, 01500 Tizi-Ouzou, Algeria; c Department of Operations Research, University of Bejaia, 06000 Bejaia, Algeria.

Abstract—In this paper, we present a new method for solving quadratic programming problems, not strictly convex. Constraints of the problem are linear equalities and inequalities, with bounded variables. The suggested method combines the active-set strategies and support methods. The algorithm of the method and numerical experiments are presented, while comparing our approach with the active set method on randomly generated problems.

Keywords—Convex quadratic programming, dual support methods, active set methods.

I. INTRODUCTION

Theory of quadratic programming (QP) handles a special class of problems of nonlinear mathematical programming. In practice, a lot of problems are represented naturally by quadratic models, as in the case of the finance, management of the production, statistics and the optimal control, etc. These models are also used in nonlinear optimization as subproblems in the sequential quadratic programming methods [9].

In the literature, one distinguishes three approaches for solving quadratic optimization problems, such as active-set strategies [7], [6], [9], developed in the beginning of the 70s years. The second approach is constituted from interior points methods, known for their good theoretical complexity that is polynomial and its efficiency for solving large-scale problems [8]. The third is formed by support methods [1], [2], [5] and they are intermediate between the two first approaches. The majority of these methods are primal or primal-dual, but a few are of the dual type. In [12], the dual simplex method has been extended for solving strictly convex problems. The dual active-set method for the strictly convex case has been presented in [11] and Boland [10] generalized it for the general convex case, i.e., the matrix associated to the quadratic form being positive semidefinite. The method presented here is constituted of two phases: the first consists in finding an initial active set by solving a linear program, while the second is iterative and permits to change the active set while improving the objective function until the optimum.

In this work, we generalize the dual support method [1], [5] for the general convex case and with linear general inequalities constraints. The new algorithm combines the active set strategy [6], [7] and support methods [2]. The algorithm of this dual method is iterative and is constituted of two phases: the first consists in constructing an initial coordinator support of the problem. The second phase permits to get the optimal solution for a bounded problem. In the numerical experiments we compare our method with active set method, implemented in Matlab 7 as Quadprog function, on randomly generated test problems.

II. STATEMENT OF THE PROBLEM AND DEFINITIONS

We consider the following quadratic model:

\[
\begin{align*}
\min \{ F(x) = & \frac{1}{2} x^T D x + c^T x \\
& A(I_1, J_1) x = b(I_1), \\
& A(I_2, J_2) x \leq b(I_2), \\
& d^- \leq x \leq d^+, 
\end{align*}
\]

where indices of constraints and decision variables are respectively noted by: \( I = I_1 \cup I_2 \), \( I_1 = \{1, 2, \ldots, m_1\} \), \( I_2 = \{m_1 + 1, m_1 + 2, \ldots, m\} \) and \( J = \{1, 2, \ldots, n\} \). So, \( c = (c_j, j \in J) \), \( d^- = (d^j_-, j \in J) \), \( d^+ = (d^j_+, j \in J) \) et \( x = (x_j, j \in J) \) are \( n \)-vectors; \( b = (b_i, i \in I) \) is an \( m \)-vector; \( A = (A(I, J)) \) is an \( (m \times n) \) matrix, such that \( m \leq \text{rang}(A) < n \). The square matrix \( D = (D(I,J)) \) is supposed symmetric and positive semidefinite. The symbol \( (\cdot)^T \) is the transposition operation, and \( A^T_i, i \in I \), represent the \( i \)-th line of \( A \).

Definition 1:

- Any vector \( x \) satisfying all the constraints of the problem (1) is called a feasible solution.
- A feasible solution \( x^* \) is called optimal if

\[
F(x^*) = \frac{1}{2} x^*^T D x^* + c^T x^* = \min F(x),
\]

where \( x \) is taking among all feasible solutions for the problem.

- A general constraint \( i \in I \) is said to be active in a point \( x \) if it satisfies the identity \( A^T_i x = b_i \). The active set in \( x \) is noted \( I_0 = I_1 \cup \{i \in I_2 : A^T_i x = b_i\} \).

- The pair \( S_B = \{I_0, J_B\}, \) where \( I_1 \subset I_0 \subset I \) and \( J_B \subset J \) such that \( |I_0| = |J_B| \) is called a constraint support of the problem (1) if the matrix \( A_B = A(I_0, J_B) \) is nonsingular. For the initial constraint support, we can choose \( S_B = \{I_0, J_B\}, \) with \( |J_B| = m_1 \).

- The couple \( (x, S_B) \) formed by the feasible solution \( x \) and the support \( S_B \) is called the support feasible solution. It is said to be nondegenerate if

\[
d^-_j < x_j < d^+_j, \quad \forall j \in J_B,
\]

\[
A^T_i x < b_i, \quad \forall i \in I_0 = I \setminus I_0.
\]
The optimality criterion for the problem (1) is given by the function.

Definition 2: Let \( \lambda, (\kappa, y, v, w) \) be a feasible solution vector of the primal problem (1). The quadruplet \( (\kappa, y, v, w) \) is called a reduced costs vector.

Definition 3: The indices set \( J_S \) is called a coordinate reduced costs vector if the constraints support \( J_S \) and \( (\kappa, y, v, w) \) is taking among all dual feasible solutions of the problem (2). The corresponding vector \( \delta^* = D\kappa^* + c - A^*y^* \) is then an optimal reduced costs one.

Theorem 1: Let \( \{x, S_B\} \) be a feasible support solution for the problem (1). Then the following relations:

\[
E_j \geq 0, \quad x_j = d_j^-, \quad E_j \leq 0, \quad x_j = d_j^+, \quad E_j = 0, \quad d_j^- < x_j < d_j^+, \quad j \in J_N.
\]

are sufficient for the optimality of the point \( x \). The same relations are also necessary if \( \{x, S_B\} \) is not degenerate.

III. Dual Problem and Optimality Criterion

The dual problem associated to the primal (1) is the following concave quadratic program:

\[
\begin{align*}
\max \varphi(\lambda) &= -\frac{1}{2}k'\kappa + b'y + v'd - w'd^+ \\
D\kappa + c - A'y + v + w &= 0, \\
y(I_2) &\geq 0, \quad v \geq 0, \quad w \geq 0,
\end{align*}
\]

where \( \lambda = (\kappa, y, v, w) \).

Definition 3: The quadruplet \( (\kappa, y, v, w) \) is called a dual feasible solution. The \( \nu^- \)-vector \( \kappa \) verifying \( A(I_2)\kappa = b(I_2) \) is called a pseudosolution vector. Also, the \( \nu^- \)-vector \( \delta = D\kappa + c - A'y \) is called a reduced costs vector, associated to the dual feasible solution \( \lambda \).

A dual feasible solution \( \lambda^* = (\kappa^*, y^*, v^*, w^*) \) is said to be optimal if

\[
\varphi(\lambda^*) = -\frac{1}{2}k'^*D\kappa^* + b'y^* + v'^*d - w'^*d^+ = \max_{\lambda} \varphi(\lambda),
\]

where \( \lambda = (\kappa, y, v, w) \) is taking among all dual feasible solutions of the problem (2). The corresponding \( \nu^- \)-vector \( \delta^* = D\kappa^* + c - A'y^* \) is then an optimal reduced costs one.

considering a support \( S_P = (I_1, J_P) \) for the problem (1) and let \( \lambda = (\kappa, y, v, w) \) a dual feasible solution accompanying \( S_P \), constructed as follows:

\[
d_j^* \leq \kappa_j \leq d_j^+, \quad j \in J_N = J\setminus J_P, \\
\kappa_S = -M_S^{-1}Z(S_P, J_P)(D(J_P, J_B)A_B^{-1}b + c(J_P)) - M_S^{-1}M(S, J_N)k_{NN}, \quad \kappa_B = A_B^{-1}(b_B - A_Nk_{NN}); \\
y_S = [A_B^{-1}]^{-1}(D(J_B, J)\kappa + c_B), \quad y_{na} = 0; \\
\delta = D\kappa + c - A'y, \quad \delta_P = \delta(J_P) = 0 \text{ by construction}
\]

In this case, the couple \( \{\lambda, S_P\} \) is called a coordinated dual feasible solution, and \( \{\delta, S_P\} \) is also called a coordinated reduced costs vector. It is said to be nondegenerate if

\[
d_j \neq 0, \quad \forall j \in J_N, \\
y_0 > 0, \quad \forall i \in I_2 \cap I_a.
\]

Since \( \delta(J_P) = 0 \) and \( y(I_{na}) = 0 \), by construction, then the optimality criterion of the dual problem (2) is expressed as follows:

Theorem 2: The relations

\[
d_j^* \leq \kappa_j \leq d_j^+, \quad \forall j \in J_P = J_B \cup J_S, \\
\kappa_B \leq b_B, \quad \forall i \in I_{na} \cap I_a
\]

are sufficient for the optimality of the coordinated reduced costs vector \( \{\delta, S_P\} \). They are also necessary in the case when \( \{\delta, S_P\} \) is not degenerate. The corresponding pseudosolution \( \kappa \) of the optimal reduced costs vector is then an optimal solution to the primal problem (1).

IV. Construction of the Initial Coordinator Support

The phase 1 of the proposed method allows to construct an initial coordinator support that is necessary for the formulation of the optimality criterion, and to start the phase 2 of the dual support method. So, this procedure yields an initial coordinated reduced costs vector \( \{\delta, S_P\} \). As an initial support, we can choose \( S_P = (S_B, J_S = \emptyset) \), where \( S_B = (I_1, J_B) \) such that \( |J_B| = |I_1| = m_1 \) is the constraints support of the problem. The steps of the phase 1 are as follows:

**Step 1.** Construction of the reduced costs vector \( \delta \) associated to the initial support \( S_P \):

- Construct the initial pseudosolution vector \( \kappa = (\kappa_B, k_{NN})' \):

\[
\begin{align*}
d_j^* &\leq \kappa_j \leq d_j^+, \quad j \in J_N, \\
\kappa_B &= [A_B^{-1}]^{-1}(b_B - A_Nk_{NN}).
\end{align*}
\]

- Calculate the vector \( \delta = (\delta_B, \delta_N)' \):

\[
\delta_B = 0, \quad \delta_N = D(J_X, J)\kappa + c_N - A_Ny, \quad \text{with}
\]

\[
y = [A_B^{-1}]^{-1}(D(J_B, J) \kappa + c_B).
\]
Step 2. Apply the coordination tests:

- If $\kappa_j = d_{ij}^-$ for $\delta_j \geq 0$ and $\kappa_i = d_{ij}^+$ for $\delta_j < 0, \forall j \in J_{NN}$, then $\{\delta, S_p\}$ is a supporting coordinated reduced costs vector and consequently the procedure is stopped.
- Otherwise, construct a new pseudosolution vector $\bar{\kappa}$ and its corresponding reduced costs vector $\bar{\delta}$:

$$\bar{\kappa} = \kappa + \sigma l, \quad \bar{\delta} = \delta + \sigma t,$$

where the $n$-vectors $l$ and $t$, representing respectively the improvement directions of the pseudosolution and its reduced costs vector, are determined as follows:

$$\begin{align*}
    l_j &= d_{ij}^- - \kappa_i, \quad \text{if } \delta_j \geq 0, \\
    l_j &= d_{ij}^+ - \kappa_i, \quad \text{if } \delta_j < 0, \quad j \in J_{NN}, \\
    l_B &= -M_S^{-1}M(J_S, J_{NN})l_{NN}, \\
    l_B &= -A_B^{-1}A_Nl_N, \\
    t_B &= t_S = 0, \\
    t_{NN} &= D(J_{NN})Jl - A_Nq, \\
    q &= [A_B^{-1}]^T[D(J_B, J_P)].
\end{align*}$$

Calculate the optimal stepsize $\sigma = \min \{1, \sigma_m\}$, where $\sigma_m$ is determined such that $\delta_{NN}$ and $\delta_{NN}$ don’t change their signs:

$$\sigma_m = \min_{j \in J_{NN}} \sigma_j,$$

where $\sigma_j = \begin{cases} -\delta_j/l_j, & \text{if } \delta_j > 0, \\
0, & \text{if } \delta_j = 0 \text{ and } l_j > 0, \\
+\infty, & \text{otherwise, } j \in J_{NN}.
\end{cases}$$

- If $\sigma = 1$ then $S_P = \{S_B, J_S\}$ is an initial coordinated support with $\bar{\kappa} = \kappa + l$ its associated pseudosolution vector.
- Otherwise, do the following appropriate index change:

$$J_S = J_S \cup J_0, \quad J_B = J_B, \quad J_P = J_B \cup J_S.$$ 

Go to step (2).

**Remark 1**: The proposed procedure is finite, since it allows to construct in the worst case a coordinator support in $(n - m_1)$ iterations, when the method is started with $J_S = 0$.

V. ALGORITHM OF THE DUAL SUPPORT METHOD

This new dual method combines the support method [1] and the active set strategy [6], [7], [9]. Let $\{\delta, S_P\}$ be an initial coordinated reduced costs vector, obtained by the phase 1 of the method and $\kappa$ its corresponding pseudosolution. The principle of the method consists to transform the pair $\{\delta, S_P\} \rightarrow \{\delta, S_P\}$ in two steps; the first is the construction of the new reduced costs vector $\delta$ such that $\varphi(\delta) \geq \varphi(\lambda)$, where $\lambda$ and $\lambda$ represent respectively the dual feasible solutions associated to the reduced costs vectors $\delta$ and $\delta$. The second stage consists to transform $S_P \rightarrow S_P$ in such a way that $S_P$ must be a coordinator support.

The algorithm of the method presents the following stages:

**Step 1.** Verification of the optimality conditions of the couple $\{\delta, S_P\}$:

- If optimality relations (4) are verified, then $\kappa$ is an optimal feasible solution of the primal problem (1). Stop the algorithm.
- Otherwise for all non optimal support indices $J_{PP}^o \subset J_P$, and those of the violated general constraints at the constraint point $\kappa$, noted $I_{PP}^o \subset I_{Na} = I \setminus I_a$, calculate the following:

$$\alpha_j = \begin{cases} \kappa_j - d_{ij}^- & \text{if } \kappa_j < d_{ij}^-, \\
\kappa_j - d_{ij}^+ & \text{if } \kappa_j > d_{ij}^-, \quad j \in J_{PP}^o, \\
\kappa_j - b_i & \text{if } \kappa_j > b_i, \quad i \in I_{PP}^o.
\end{cases}$$

Choose an index $j_1 \in J_{PP}^o$ and another $i_1 \in I_{PP}^o$ verifying:

$$|\alpha_{j_1}| = \max \{|\alpha_j|, j \in J_{PP}^o\}, \quad r_{i_1} = \max \{r_i, \quad i \in I_{PP}^o\}.$$ 

**Step 2.** Construction of the new pseudosolution and its associated reduced costs vector as follows:

$$\bar{\kappa} = \kappa + \sigma l, \quad \bar{\gamma} = y + \sigma q, \quad \delta = \delta + \sigma t, \quad \sigma \geq 0.$$ 

- Determine the improvement directions $t = (l_B, l_S, J_{NN})'$ of the pseudosolution and those of its reduced costs vector $t = (t(J_P), t(J_{NN}))'$ in the following way:

$$\begin{align*}
    l_{NN} &= 0, \\
    t_{NN} &= -M_S^{-1}Z'(J_S, j_1), \\
    l_B &= -A_B^{-1}A_Nl_N, \\
    l_P &= l_B, \\
    t_B &= t_S = 0, \\
    t_{NN} &= D(J_{NN})Jl - A_Nq, \\
    q &= [A_B^{-1}]^T[D(J_B, J_P)].
\end{align*}$$

- Else $(|\alpha_{j_1}| < r_{i_1})$, set:

$$\begin{align*}
    l_{NN} &= 0, \\
    t_{NN} &= -M_S^{-1}Z'(J_S, J)'A'(J, i_1), \\
    l_B &= -A_B^{-1}A_Nl_N, \\
    t_B &= t_S = 0, \\
    t_{NN} &= D(J_{NN})Jl - A'(J_{NN}, I_a)q, \\
    q &= [A_B^{-1}]^T[D(J_B, J_P)].
\end{align*}$$

Calculate the optimal stepsize $\sigma = \min \{\sigma_{j_1}, \sigma_{i_1}, \tau_{i_1}, \tau_{j_1}\}$, where

$$\sigma_{j_1} = \min_{j \in J_{NN}} \sigma_j,$$

and $\sigma_{i_1} = \begin{cases} -d_{ij}^-/l_j, & \text{if } l_j < 0, \\
0, & \text{if } \delta_j = 0 \text{ and } l_j > 0, \\
+\infty, & \text{otherwise, } j \in J_{NN},
\end{cases}$

and $\sigma_{j_1} = \begin{cases} -d_{ij}^+/(d_{ij}^- - \kappa_j), & \text{if } l_j > 0, \\
0, & \text{if } \delta_j = 0 \text{ and } l_j < 0, \\
+\infty, & \text{otherwise, } j \in J_{NN}.
\end{cases}$

The step $\sigma_{j_1}$ represent the maximal value for which $\delta_{NN}$ and $\delta_{NN}$ keep the same sign, and $\sigma_{j_1}$ is the value for which the index $j_1$ becomes optimal, i.e., the component $\bar{\kappa}_j = d_{ij}^- \vee d_{ij}^+$. The number $\tau_{i_1}$ is the value of stepsize for which active components of multipliers vectors $y_a = y(I_a)$ and $\bar{y}_a = y(I_a)$ don’t change a sign:

$$\tau_{i_1} = \min_{i \in I_{2} \cap I_{Na}} \tau_{i_1},$$

with $\tau_{i_1} = \begin{cases} -y_{qi}, & \text{if } y_{qi} < 0, \\
0, & \text{if } y_{qi} = 0, \quad q_i < 0, \quad \text{or otherwise, } i \in I_2 \cap I_{Na}.
\end{cases}$

Finally, $\tau_{i_1}$ is the optimal value for which the component $y_{qi}$ becomes critical:

$$\tau_{i_1} = \begin{cases} -r_{i_1}/A_{i_1}'l_i, & \text{if } A_{i_1}'l_i < 0, \\
+\infty, & \text{otherwise}.
\end{cases}$$
If $\sigma = \infty$, then the dual problem (2) is not bounded below and consequently its primal (1) is infeasible. Stop the resolution process.

Otherwise $\sigma$ is finite, and calculate $\delta = \delta + \sigma \epsilon$, $\bar{\epsilon} = \epsilon + \sigma g$.

Step 3. Construction of a new coordinator support $S_P = \{S_B, J_S\}$, where $S_B = (I_a, J_B)$:

This change is operated according to the following cases:

Step 3.1. If $\sigma = \sigma_1$, then coordinate $j_1 \in J_S$ reaches the left or the right bound:

- If $j_1 \in J_S$, then put $S_B = S_B$, $J_S = J_S \cup j_1$.
- Otherwise, change the constraints support while choosing an index $j_* \in J_S$ such that $z_{j_1, j_*} = Z(j_1, j_*) \neq 0$ and put $I_a = I_a, \quad J_B = (J_B \setminus j_1) \cup j_*, \quad J_S = J_S \setminus j_*$.

Go to step (1).

Step 3.2. If $\sigma = \sigma_0$, then component $\kappa_0$ is not critical and consequently the index $I_1$ cannot be transferred to $J_S$.

- If $\eta_0 = M(j_0, J_0) - M(J_0, J_S) \eta_0 \neq 0$, then the index $J_0$ will be transferred to $J_S$ such that the functional support matrix $M_S = M(J, J_S)$ remains nonsingular:

$$S_B = S_B, \quad J_S = J_S \cup j_0.$$

Go to step (2).

- Otherwise, change the constraints support while entering the index $j_0$ and take out the index $j_1$ from $J_S$:

$$I_a = I_a, \quad J_B = (J_B \setminus j_1) \cup j_0, \quad J_S = J_S \setminus j_*.$$

- Correct the pseudosolution vector $\bar{\epsilon} = \epsilon + \bar{\epsilon}$, where the improvement direction $l$ is calculated in such a way that the component $\bar{\epsilon}_{j_1}$, $j_1 \in J_N = J(J_B \cup J_S)$ becomes coordinated with the support $S_P$. Consequently, it is determined as follows:

$$l_{j_1} = \frac{d_j^T}{\bar{\epsilon}_{j_1}} - \bar{\epsilon}_{j_1}, \quad \text{if } t_{j_1} = 1; \quad l_{j_1} = \frac{d_j^T}{\bar{\epsilon}_{j_1}} - \bar{\epsilon}_{j_1}, \quad \text{if } t_{j_1} = -1; \quad l_j = 0, j \neq j_1, j \in J_{N_N}; \quad l_S = -M_S^{-1}M(J, J_{N_N})l_{N_N}; \quad l_B = Z(J_B, J_{N_N})l_{N_N},$$

where $M$ and $Z$ are the updating matrices defined by the relation (2). Go to step (1).

Step 3.3. If $\sigma = \sigma_1$, $i_1 \in I_m$, then the $i_1$-th general constraint becomes active. Find an index $j_* \in J_S$ verifying $h_{i_1} \neq 0$, where $h_j(J_S) = A(i_1, J_S) - A(i_1, J_B)A_B^{-1}A(I_m, J_S)$ and put:

$$I_a = I_a \cup i_1, \quad J_B = J_B \cup j_*, \quad J_S = J_S \setminus j_*.$$

Go to step (1).

Step 3.4. If $\sigma = \sigma_0$, $i_0 \in I_m$, then the $i_0$-th constraint becomes nonactive and consequently, it will be deleted in the active set by this change:

- If $\sigma_0 \leq \tau_1$, then the $j_1$-th simple constraint will be deleted in $J_P$ and added to $J_{N_N}$. Set $I_a = I_a \setminus i_0$ and $J_S = J_S$.
- If $j_1 \in J_B$ then $J_B = J_B \setminus j_1$.
- If $j_1 \notin J_B$ then search an index $j_* \in J_B$ such that $A_{j_1j_2} \neq 0$ and put $J_B = J_B \setminus j_*$.

Go to step (2).

b) Otherwise the $i_1$-th general constraint becomes active and the $i_2$-th nonactive:

$$I_a = (I_a \cup i_1) \setminus i_0, \quad J_B = J_B, \quad J_S = J_S.$$

Go to step (1).

Remark 2: For the case where $I_4 = \emptyset$, the algorithm presented before is the same suggested in [5].

VI. NUMERICAL EXPERIMENTS

In this section, we present numerical results of comparison between dual support method (DSM) and active set method (ASM), implemented in Matlab 7.0 R14, under Windows Vista, at a CPU 3Ghz and 512 MO of RAM. The comparison is based on quadratic programming problems randomly generated. The characteristic of these problems is that the optimal solution $x^*$, and the value of the objective function $f^*$ at the optimum are known previously.

Input parameters of the procedure that generate randomly test quadratic problems are $n, m, n_0, m_0$, respectively they represent the number of variables, the number of general constraints, the number of active variables at the optimum $(x_j^* = d_j^T \vee d_j^T, j = 1 \cdots m_0)$ and the number of general active constraints $(A^T x^* = b_i, i = 1 \cdots m_0)$. Its stages are summarized as follows:

1) Generate randomly (or fixed) the optimal solution $x^*$.
2) Calculate the reduced costs vector $\delta^*$ by the following rules:

$$\delta_j^* \neq 0, \forall j = 1 \cdots n_0; \quad \delta_j^* = 0, \forall j = n_0 + 1 \cdots n.$$

3) Generate the multiplier vector $y^*$:

$$y_i^* \neq 0, i = 1 \cdots m; \quad y_i^* > 0, i = m_1 + m_0; \quad y_i^* = 0, i = m_0 + m_1: m.$$

4) Construct the $n$-vectors $d^-$ and $d^+$, such that $d_j^- < d_j^+$, for all $j = 1, \cdots, n$:

$$d_j^- = x_j^*, \quad d_j^- > x_j^*, \quad d_j^+ > 0; \quad d_j^- < x_j^*, \quad d_j^+ = x_j^*, \quad d_j^- < 0; \quad d_j^- < x_j^*, \quad d_j^+ > x_j^*, \quad d_j^* = 0, j \in J.$$

5) Generate the constraints matrix $A$ and the symmetric semidefinite matrix $D$ in the form $D = G^T G$ where $G$ is an $r \times n$ matrix, with $\det(G) \leq r$.

6) Calculate the $m$-vector $b$:

$$b_i = A_i^T x^*, \forall i = 1 \cdots m_0; \quad A_i^T x^* < b_i, \forall i = m_0 + 1 \cdots m.$$

7) For the $n$-vector $c$, set $c = -D x^* + A^T y^* + \delta^*$. 
Random values of different vectors and matrices are uniformly distributed in the range [-1,1]. For computational experiments we compare our dual support method (DSM) and the primal active-set method (ASM) implemented in Matlab 7 as function quadprog. This comparison is done for the standard case \( J_2 = \emptyset \) and on two kinds of test problems: strictly convex and convex quadratic programs. Concerning the general case \( J_2 \neq \emptyset \), numeric results are under realization.

Table 1 reports numerical results for strictly convex problems, where \( T \) and \( \text{Iters} \) represent respectively the average CPU time in seconds and the average number of iterations required for solving 10 QPs with the same entries. For DSM, we adapt two strategies: the first when we start the method with the empty functional support (\( J_S = \emptyset \)) and the second with the full functional support (\( J_S = J_N \)). For DSM, we note that the first strategy requires more iterations compared to the second strategy for all test problems that have been solved. When the number of active constraints at the optimum is small, DSM with the second strategy is required to DSM with empty support and also to ASM. But when the number of active constraints is large we conclude that Active-set method is relatively efficient to our approach for solving strictly convex quadratic programs. For general convex quadratic programs, the results are drawn up in table 2. We constat when \( m \leq 30 \) our method is very efficient compared to the ASM.

VII. CONCLUSION

In this paper, we have developed a new method for solving general convex quadratic problems that combines active set strategy [7] and support methods [1], [2]. Its advantage is that it handles constraints such as initially presented, without any preliminary transformation. This permits to have a gain in memory space and CPU time. It also allows, following the example that of Boland [10], to deal with semidefinite quadratic problems, contrary to other methods which consider only the strictly convex case [12], [11], [13].

REFERENCES


