World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:3, No:10, 2009

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 few are of the dual type. In [12], the dual simplexe 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:

$$\begin{cases} \min F(x) = \frac{1}{2}x'Dx + c'x \\ A(I_1, J)x = b(I_1), \\ A(I_2, J)x \leq b(I_2), \\ d^- \leq x \leq d^+, \end{cases} \tag{1}$$

where indices of constraints and decision variables are respectively noted by: $I = I_1 \cup I_2$, $I_1 = \{1, 2, \cdots, m_1\}$, $I_2 = \{m_1 + 1, m_1 + 2, \cdots, m\}$ and $J = \{1, 2, \cdots, n\}$. 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 $m_1 \leq \text{rang}(A) < n$. The square matrix $D = D(J, J)$ is supposed symmetric and positive semidefinite. The symbol ($'$) is the transposition operation, and A_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
$$F(x^*) = \frac{1}{2}x^{*\prime}Dx^* + c'x^* = \min_x F(x),$$
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 $A_i'x = b_i$. The active set in x is noted $I_0 = I_1 \cup \{i \in I_2 : A_i'x = b_i\}$.
- The pair $S_B = \{I_a, J_B\}$, where $I_1 \subset I_a \subset I_0$ and $J_B \subset J$ such that $|I_a| = |J_B|$ is called a constraint support of the problem (1) if the matrix $A_B = A(I_a, J_B)$ is nonsingular. For the initial constraint support, we can choose $S_B = \{I_1, J_B\}$, 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
$$\begin{array}{ll} d_j^- < x_j < d_j^+, & \forall j \in J_B, \\ A_i'x < b_i, & \forall i \in I_{na} = I \backslash I_a. \end{array}$$

Set $J_N = J \backslash J_B$. Let's define the potential vector u and the reduced costs vector E:
$u' = u'(I) = (u'_a, u'_{na})$, such that $u_{na} = u(I_{na}) = 0$, $u'_a = u'(I_a) = g'_B A_B^{-1}$, $E' = E'(J) = g' - u'A = (E'_B, E'_N)$, such that $E'_N = E'(J_N) = g'_N(x) - u'A_N$, $E'_B = 0$, where $g = g(J) = g(x) = Dx + c$ represents the gradient of the function F in a point x, with $g' = (g'_B, g'_N)$ and $A_N = A(I_a, J_N)$.
The optimality criterion for the problem (1) is given by the following theorem.

Theorem 1: [3]. Let $\{x, S_B\}$ be a feasible support solution for the problem (1). Then the following relations:

$$\begin{cases} E_j \geq 0, & \text{for} \quad x_j = d_j^-, \\ E_j \leq 0, & \text{for} \quad x_j = d_j^+, \\ E_j = 0, & \text{for} \quad d_j^- < x_j < d_j^+, j \in J_N, \\ u_i \geq 0, & \text{for} \quad A'_i x = b_i, i \in I_2 \cap I_a \\ u_i = 0, & \text{for} \quad A'_i x < b_i, i \in I_{na}, \end{cases}$$

are sufficient for the optimality of the point x. The same relations are also necessary if $\{x, S_B\}$ is not degenerate.
For the case of bounded quadratic problems, the optimal solution can be an extreme point, a point on the face or in the interior of the convex polyhedron. Thus, it is important in the nonlinear optimization to define a new concept allowing to have some informations on the curvature of the objective function.

Definition 2:

- The indices set $J_S \subset J_N = J \backslash J_B$ is called an objective function support of the problem (1) if the submatrix $M_S = M(J_S, J_S)$ of M is nonsingular, where

$$M = M(J_N, J_N) = Z'DZ \text{ and } Z = \begin{pmatrix} -A_B^{-1} A_N \\ I_N \end{pmatrix}.$$

  Here I_N is the identity matrix of $(n - m_0)$-dimension and $m_0 = |I_a|$. Set $J_P = J_B \cup J_S$ and $J_{NN} = J \backslash J_P$.
- The indices set $S_P = \{S_B, J_S\} = \{I_a, J_P\}$, formed with the constraints support S_B and the functional support J_S is called a *support of the problem (1)*.

III. Dual problem and optimality criterion

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

$$\begin{cases} \max \varphi(\lambda) = -\frac{1}{2}\kappa' D\kappa + b'y + v'd^- - w'd^+ \\ D\kappa + c - A'y - v + w = 0, \\ y(I_2) \geq 0, \quad v \geq 0, \quad w \geq 0, \end{cases} \tag{2}$$

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

Definition 3:

- The quadruplet $\lambda = (\kappa, y, v, w) \in \mathbb{R}^{n \times m \times n \times n}$ verifying all constraints of the problem (2) is called *a dual feasible solution*. The n-vector κ verifying $A(I_a, J)\kappa = b(I_a)$ is called a pseudosolution vector. Also, the n-vector $\delta = D\kappa + c - A'y$ is called a *reduced costs* vector, associated to the dual feasible solution λ.
- A dual feasible solution $\lambda^* = (\kappa^*, y^*, v^*, w^*)$ is said to be *optimal* if

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

  where $\lambda = (\kappa, y, v, w)$ is taking among all dual feasible solutions of the problem (2). The corresponding n-vector $\delta^* = D\kappa^* + c - A'y^*$ is then an optimal reduced costs one.
- Considering a support $S_P = (I_a, J_P)$ for the problem (1) and let $\lambda = (\kappa, y, v, w)$ a dual feasible solution accompanying S_P, constructed as follows:
  $d_j^- \leq \kappa_j \leq d_j^+, j \in J_{NN} = J \backslash J_P$,
  $\kappa_S = -M_S^{-1} Z'(J_S, J_P)[D(J_P, J_B) A_B^{-1} b + c(J_P)] - M_S^{-1} M(J_S, J_{NN})\kappa_{NN}$, $\kappa_B = A_B^{-1}(b_a - A_N \kappa_N)$;
  $y_a = [A'_B]^{-1}(D(J_B, J)\kappa + c_B), \quad y_{na} = 0$;
  $\delta = D\kappa + c - A'y$, with $\delta_P = \delta(J_P) = 0$ by construction of y_a and κ_S ; $v_j = \delta_j$, $w_j = 0$, if $\delta_j \geq 0$; $v_j = 0$, $w_j = -\delta_j$, if $\delta_j < 0$.
  The support S_P is said to be coordinator if the following relations are verified:

$$\begin{cases} \kappa_j = d_j^- & \text{if} \quad \delta_j \geq 0, \\ \kappa_j = d_j^+ & \text{if} \quad \delta_j \leq 0, \, j \in J_{NN} = J \backslash J_P. \\ A'_i \kappa = b_i & \text{if} \quad y_i \geq 0, \, i \in I_2 \cap I_a. \end{cases} \tag{3}$$

  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

$$\delta_j \neq 0, \, \forall j \in J_{NN}, \quad y_i > 0, \, i \in I_2 \cap I_a.$$

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

$$\begin{array}{ll} d_j^- \leq \kappa_j \leq d_j^+, & \forall j \in J_P = J_B \cup J_S, \\ A'_i \kappa \leq b_i, & \forall i \in I_{na} = I \backslash I_a, \end{array} \tag{4}$$

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 κ 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 δ associated to the initial support S_P:

- Construct the initial pseudosolution vector $\kappa = (\kappa_B, \kappa_N)'$:

$$\begin{cases} d_j^- \leq \kappa_j \leq d_j^+, \, j \in J_N, \\ \kappa_B = A_B^{-1}(b_a - A_N \kappa_N). \end{cases}$$

- Calculate the vector $\delta = (\delta_B, \delta_N)'$:
  $\delta_B = 0, \quad \delta_N = D(J_N, J)\kappa + c_N - A'_N y$, with
  $y = [A'_B]^{-1}(D(J_B, J)\kappa + c_B)$.

Step 2. Apply the coordination tests:

- If $\kappa_j = d_j^-$ for $\delta_j \geq 0$ and $\kappa_j = d_j^+$ 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 $\bar{\delta}$:

$$\bar{\kappa} = \kappa + \sigma l, \quad \bar{\delta} = \delta + \sigma t, \text{ with } t = Dl - A'q, \ \bar{y} = y + \sigma q,$$

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

$$\begin{cases} l_j = d_j^- - \kappa_j, & \text{if } \delta_j \geq 0, \\ l_j = d_j^+ - \kappa_j, & \text{if } \delta_j < 0, \ j \in J_{NN}, \\ l_S = -M_S^{-1} M(J_S, J_{NN}) l_{NN}, \\ l_B = -A_B^{-1} A_N l_N, \end{cases}$$
$$\begin{cases} t_B = 0, \ t_S = 0, \\ t_{NN} = D(J_{NN}, J) l - A'_{NN} q, \\ q = [A_B^{-1}]' D(J_B, J) l. \end{cases}$$

Calculate the optimal stepsize $\sigma = \min\{1, \sigma_{j_0}\}$, where σ_{j_0} is determined such that δ_{NN} and $\bar{\delta}_{NN}$ don't change their signs:

$$\sigma_{j_0} = \min_{j \in J_{NN}} \sigma_j, \text{ with } \sigma_j = \begin{cases} -\delta_j / t_j, \text{ if } \delta_j t_j < 0, \\ 0, \text{ if } \delta_j = 0 \text{ and } t_j < 0, \\ +\infty, \text{ otherwise}, j \in J_{NN}. \end{cases}$$

- 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:

$$\bar{J}_S = J_S \cup j_0, \quad \bar{J}_B = J_B, \quad \bar{J}_P = \bar{J}_B \cup \bar{J}_S.$$

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 $J_S = \emptyset$.

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 κ its corresponding pseudosolution. The principle of the method consists to transform the pair $\{\delta, S_P\} \longrightarrow \{\bar{\delta}, \bar{S}_P\}$ in two steps; the first is the construction of the new reduced costs vector $\bar{\delta}$ such that $\varphi(\bar{\lambda}) \geq \varphi(\lambda)$, where λ and $\bar{\lambda}$ represent respectively the dual feasible solutions associated to the reduced costs vectors δ and $\bar{\delta}$. The second stage consists to transform $S_P \longrightarrow \bar{S}_P$ in such a way that $\bar{S}_P$ 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 κ is an optimal feasible solution of the primal problem (1). Stop the algorithm.
- Otherwise for all non optimal support indices $J_P^{\text{no}} \subset J_P$, and those of the violated general constraints at the courant point κ, noted by $I_{na}^{\text{no}} \subset I_{na} = I \backslash I_a$, calculate the following numbers:

$$\alpha_j = \begin{cases} \kappa_j - d_j^-, & \text{if } \kappa_j < d_j^-, \\ \kappa_j - d_j^+, & \text{if } \kappa_j > d_j^+, j \in J_P^{no}, \end{cases}$$
$$r_i = A'_i \kappa - b_i > 0, \ i \in I_{na}^{no}.$$

Choose an index $j_1 \in J_P^{no}$ and another $i_1 \in I_{na}^{no}$ verifying:

$$|\alpha_{j_1}| = \max\{|\alpha_j|, \, j \in J_P^{no}\}, \quad r_{i_1} = \max\{r_i, \, i \in I_{na}^{no}\}.$$

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

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

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

- If $|\alpha_{j_1}| \geq r_{i_1}$ then put:

$$\begin{cases} l_{NN} = 0, \\ l_S = t_{j_1} M_S^{-1} Z'(J_S, j_1), \\ l_B = -A_B^{-1} A(I_a, J_S) l(J_S), \end{cases}$$
$$\begin{cases} t_{j_1} = -\text{sign}\alpha_{j_1}, \ t(J_P \backslash j_1) = 0, \ q(I_{na}) = 0, \\ t_{NN} = D(J_{NN}, J_P) l_P - A'(J_{NN}, I_a) q_a, \\ q_a = q(I_a) = [A_B^{-1}]' [D(J_B, J_P) l_P - t_B]. \end{cases}$$

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

$$\begin{cases} l_{NN} = 0, \\ l_S = -M_S^{-1} Z(J_S, J)' A'(J, i_1), \\ l_B = -A_B^{-1} A(I_a, J_S) l(J_S), \end{cases}$$
$$\begin{cases} t(J_P) = 0, q_{i_1} = 1, \ q(I_{na} \backslash i_1) = 0, \\ t_{NN} = D(J_{NN}, J_P) l_P - A'(J_{NN}, I_a \cup i_1) q(I_a \cup i_1), \\ q(I_a) = [A_B^{-1}]' [D(J_B, J_P) l_P - A'(J_B, i_1)]. \end{cases}$$

- Calculate the optimal stepsize $\sigma = \min\{\sigma_{j_0}, \sigma_{j_1}, \tau_{i_0}, \tau_{i_1}\}$, where

$$\sigma_{j_0} = \min_{j \in J_{NN}} \sigma_j, \text{ with } \sigma_j = \begin{cases} -\delta_j / t_j, \text{ if } \delta_j t_j < 0, \\ 0, \text{ if } \delta_j = 0 \text{ and } t_j < 0, \\ \infty, \text{ otherwise }, \ j \in J_{NN}, \end{cases}$$

$$\text{and} \quad \sigma_{j_1} = \begin{cases} (d_{j_1}^- - \kappa_{j_1}) / l_{j_1}, \text{ if } l_{j_1} > 0, \\ (d_{j_1}^+ - \kappa_{j_1}) / l_{j_1}, \text{ if } l_{j_1} < 0, \\ \infty, \text{ if } l_{j_1} = 0. \end{cases}$$

The step σ_{j_0} represent the maximal value for which δ_{NN} and $\bar{\delta}_{NN}$ keep the same sign, and σ_{j_1} is the value for which the index j_1 becomes optimal, i.e., the component $\bar{\kappa}_{j_1} = d_{j_1}^- \vee d_{j_1}^+$. The number τ_{i_0} is the value of stepsize for which active components of multipliers vectors $y_a = y(I_a)$ and $\bar{y}_a = \bar{y}(I_a)$ don't change a sign:

$$\tau_{i_0} = \min_{i \in I_2 \cap I_a} \tau_i, \text{ with } \tau_i = \begin{cases} -y_i / q_i, \text{ if } y_i q_i < 0, \\ 0, \text{ if } y_i = 0, \, q_i < 0, \\ \infty, \text{ otherwise}, \ i \in I_2 \cap I_a. \end{cases}$$

Finally, τ_{i_1} is the optimal value for which the component y_{i_1} becomes critical:

$$\tau_{i_1} = \begin{cases} -r_{i_1} / A'_{i_1} l, & \text{if } A'_{i_1} l < 0, \\ \infty, & \text{otherwise }. \end{cases}$$

- If $\sigma = \infty$, then the dual problem (2) is not bounded below and consequently its primal (1) is infeasible. Stop the resolution process.
- Otherwise σ is finite, and calculate $\bar{\delta} = \delta + \sigma t$, $\bar{\kappa} = \kappa + \sigma l$, $\bar{y} = y + \sigma q$.

Step 3. Construction of a new coordinator support $\bar{S}_P = \{\bar{S}_B, \bar{J}_S\}$, where $= \bar{S}_B = (\bar{I}_a, \bar{J}_B)$:
This change is operated according to the following cases:
Step 3.1. If $\sigma = \sigma_{j_1}$, $j_1 \in J_P = J_B \cup J_S$, the component κ_{j_1} reaches the left or the right bound:

- If $j_1 \in J_S$, then put
$$\bar{S}_B = S_B, \quad \bar{J}_S = J_S \backslash j_1.$$
- 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
$$\bar{I}_a = I_a, \quad \bar{J}_B = (J_B \backslash j_1) \cup j_*, \quad \bar{J}_S = J_S \backslash j_*.$$

Go to step **(1)**.
Step 3.2. If $\sigma = \sigma_{j_0}$, then component κ_{j_1} is not critical and consequently the index j_1 cannot be transferred to $\bar{J}_{NN}$.

- If $\eta_0 = M(j_0, j_0) - M(j_0, J_S) M_S^{-1} M(J_S, j_0) \neq 0$, then the index j_0 will be transferred to $\bar{J}_S$ such that the functional support matrix $M_S = M(\bar{J}_S, \bar{J}_S)$ remains nonsingular:
$$\bar{S}_B = S_B, \quad \bar{J}_S = J_S \cup j_0.$$
Go to step **(2)**.
- If $\eta_0 = 0$, then the index j_0 cannot be added to the functional support:
 - If $j_1 \in J_S$ then do the appropriate following support change.
$$\bar{S}_B = S_B, \quad \bar{J}_S = (J_S \backslash j_1) \cup j_0.$$
 - Otherwise, change the constraints support while entering the index j_0 and take out the index j_1:
$$\bar{I}_a = I_a \quad \bar{J}_B = (J_B \backslash j_1) \cup j_0, \quad \bar{J}_S = J_S.$$
 - Correct the pseudosolution vector $\tilde{\kappa} = \bar{\kappa} + \tilde{l}$, where the improvement direction $\tilde{l}$ is calculated in such a way that the component $\tilde{\kappa}_{j_1}$, $j_1 \in \bar{J}_{NN} = J \backslash (\bar{J}_B \cup \bar{J}_S)$ becomes coordinated with the support $\bar{S}_P$. Consequently, it is determined as follows:
$$\tilde{l}_{j_1} = d_{j_1}^- - \bar{\kappa}_{j_1}, \text{ if } t_{j_1} = 1; \tilde{l}_{j_1} = d_{j_1}^+ - \bar{\kappa}_{j_1}, \text{ if } t_{j_1} = -1;$$
$$\tilde{l}_j = 0, j \neq j_1, j \in \bar{J}_{NN}; \tilde{l}_S = -M_S^{-1} M(\bar{J}_S, \bar{J}_{NN}) \tilde{l}_{NN};$$
$$\tilde{l}_B = Z(\bar{J}_B, \bar{J}_N) \tilde{l}_N,$$
 where M and Z are the updating matrices defined by the relation (2). Go to step **(1)**.

Step 3.3. If $\sigma = \tau_{i_1}$, $i_1 \in I_{na}$, then the i_1-th general constraint becomes active. Find an index $j_* \in J_S$ verifying $h_{j_*} \neq 0$, where $h'(J_S) = A(i_1, J_S) - A(i_1, J_B) A_B^{-1} A(I_a, J_S)$ and put:
$$\bar{I}_a = I_a \cup i_1, \quad \bar{J}_B = J_B \cup j_*, \quad \bar{J}_S = J_S \backslash j_*.$$
Go to step **(1)**.

Step 3.4. If $\sigma = \tau_{i_0}, i_0 \in I_2 \cap I_a$, then the i_0-th constraint becomes nonactive and consequently, it will be deleted in the active set by this change:

a) If $|\alpha_{j_1}| \leq r_{i_1}$, then the j_1-th simple constraint will be deleted in $\bar{J}_P$ and added to $\bar{J}_{NN}$. Set $\bar{I}_a = I_a \backslash i_0$ and $\bar{J}_S = J_S$.
- If $j_1 \in J_B$ then $\bar{J}_B = J_B \backslash j_1$.
- If $j_1 \notin J_B$ then search an index $j_* \in J_B$ such that $A_{i_0 j_*}^{-1} \neq 0$ and put $\bar{J}_B = J_B \backslash j_*$.
 Go to step **(2)**.

b) Otherwise the i_1-th general constraint becomes active and the i_0-th nonactive:
$$\bar{I}_a = (I_a \cup i_1) \backslash i_0, \quad \bar{J}_B = J_B, \quad \bar{J}_S = J_S.$$
Go to step **(1)**.

Remark 2: For the case where $I_2 = \emptyset$, 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, where respectively they represent the number of variables, the number of general constraints, the number of active variables at the optimum ($x_j^* = d_j^- \vee d_j^+$, $j = 1 \cdots n_0$) and the number of general active constraints ($A_i' x^* = b_i$, $i = 1 \cdots m_0$). Its stages are summarized as follows:

1) Generate randomly (or fixed) the optimal solution x^*.
2) Calculate the reduced costs vector δ^* 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^*:
$$y_i^* \neq 0,\ i = \overline{1, m_1}; \quad y_i^* > 0,\ i = \overline{m_1 + 1, m_0};$$
$$y_i^* = 0,\ i = \overline{m_0 + 1 : m}.$$
4) Construct the n-vectors d^- and d^+, such that $d_j^- < d_j^+$, for all $j = 1, \cdots, n$:
$$\begin{array}{lll} d_j^- = x_j^*, & d_j^+ > x_j^*, & \delta_j^* > 0, \\ d_j^- < x_j^*, & d_j^+ = x_j^*, & \delta_j^* < 0, \\ d_j^- < x_j^*, & d_j^+ > x_j^*, & \delta_j^* = 0,\ j \in J. \end{array}$$
5) Generate the constraints matrix A and the symmetric semidefinite matrix D in the form $D = G'G$ where G is an $r \times n$ matrix, with rank$(G) \leq r$.
6) Calculate the m-vector b:
$$b_i = A_i' x^*, \forall i = 1 \cdots m_0; \ A_i' x^* < b_i, \forall i = m_0 + 1 \cdots m.$$
7) For the n-vector c, set $c = -Dx^* + A'y^* + \delta^*$.

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 $I_2 = \emptyset$ and on two kinds of test problems: strictly convex and convex quadratic programs. Concerning the general case $I_2 \neq \emptyset$, numeric results are under realization.

Table 1 reports numerical results for strictly convex problems, where T and $Iters$ 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 that when $m \leq 30$ our method is very efficient compared to the ASM.

TABLE I
NUMERICAL RESULTS FOR STRICTLY CONVEX QUADRATIC TEST PROBLEMS

			DSM with $J_S = \emptyset$		DSM with $J_S = J_N$		ASM	
n	m	n_0	T(s)	Iters	T(s)	Iters	T(s)	Iters
100	10	50	0.39	111.6	0.34	51.2	0.41	51
100	50	50	0.43	101.8	0.30	51.8	0.23	50
200	10	100	2.47	222.6	2.68	101.6	3.02	101
200	10	150	3.33	243.2	4.32	157.4	3.66	151
200	50	150	3.72	254.4	3.77	163.8	2.80	150
200	100	150	2.51	206.6	1.98	106.6	1.56	100
300	10	250	14.55	370.8	20.94	264.4	15	251
300	50	100	7.37	314.8	6.90	100.4	7.45	101
300	50	250	14.91	391.8	17.99	277.2	12.15	250
300	100	100	6.99	300.2	5.12	100.2	5.51	101
300	100	200	12.14	366.6	11.48	218	8.40	200
400	10	350	44.107	494.8	65.62	372.4	42.73	351
400	50	100	18.26	417	15.32	100.20	17.30	101
400	50	350	47.27	525.6	60.83	383.6	36.99	350
400	100	100	17.48	400	12.63	100	14.35	101
400	100	300	43.02	515.2	45.05	332	28.97	300
500	10	450	106.24	632.2	155.95	476.6	95.73	451
500	50	200	57.33	553.4	60.25	204.4	55.74	201
500	50	450	114.82	676.6	148.58	492.4	85.66	450
500	100	100	36.15	500	26.07	100	28.74	101
500	100	400	110.16	664.2	117.37	436.4	69.20	400

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].

TABLE II
NUMERICAL RESULTS FOR SEMIDEFINITE QUADRATIC TEST PROBLEMS

				DSM with $J_S = \emptyset$		ASM	
n	m	n_0	rank(D)	T(s)	Iters	T(s)	Iters
100	10	50	80	0.375	107.2	0.656	51
200	10	150	150	4.734	281.4	5.469	151
200	40	150	150	4.687	279.7	3.391	151
300	10	250	250	25.609	429	36.734	251
300	30	250	250	18.338	434.3	19.323	251
400	20	200	350	44.953	529.5	81.797	201
500	10	450	450	135.734	675.1	272.141	451
500	30	300	450	91.063	646	140.766	301
500	50	450	450	126.078	669.4	88.406	450
700	10	600	650	432.843	939.2	755.282	601
700	50	500	650	427.390	911.5	322.922	501

REFERENCES

[1] R. Gabasov, F. M. Kirillova and V. M. Raketsky. On methods for solving the general problem of convex quadratic programming. Soviet. Math. Dokl. **23** (1981) 653–657.
[2] R. Gabasov, F. M. Kirillova, V. M. Raketsky and O.I. Kostyukova. Constructive Methods of Optimization, Volume 4: Convex Problems. University Press, Minsk (1987).
[3] E. A. Kostina and O. I. Kostyukova. An algorithm for solving quadratic programming problems with linear equality and inequality constraints. Computational Mathematics and Mathematical Physics, **41(7)** (2001) 960–973.
[4] E. A. Kostina and O. I. Kostyukova. A primal-dual active-set method for convex quadratic programming. Preprints of the University of Karlsruhe, Germany (2003).
[5] B. Brahmi and M. O. Bibi. Dual support method for solving convex quadratic programs. To appear in Optimization (2009).
[6] R. Fletcher. *Practical Methods of Optimization*. J. Wiley and Sons, Chichester, England, second edition (1987).
[7] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press Inc, London (1981).
[8] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM Studies in Applied Mathematics: Vol 13, Philadelphia (1994).
[9] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, New York, second edition (2006).
[10] N. L. Boland. A dual-active-set algorithm for positive semi-definite quadratic programming. *Mathematical Programming*, **78(1)**(1997)1–27.
[11] D. Goldfarb and A. U. Idnani. A numerically stable dual method for solving strictly convex quadratic problems. *Mathematical Programming*, **27** (1983) 1–33.
[12] C. Van de Panne and A. Whinston. The simplex and dual method for quadratic programming. *Operational Research Quarterly Journal*, **15** (1964) 355–388.
[13] R. A. Bartlett and L. T. Biegler. QPSchur: A dual, active-set, Schur-complement method for large-scale and structured convex quadratic programming. *Optimization and Engineering*, **7** (2006) 5–32.