Journal of Algorithms and Computation

journal homepage: http://jac.ut.ac.ir

An Accelerated Active-Set Algorithm for Convex Quadratic Programming

N. Bagherpour*[1], A. Shanaghi[1] and N. Minayi[1]

[1]Department of Engineering Sciences, University of Tehran, Tehran, Iran.

ABSTRACT

Convex quadratic programming (QP) arises in many applications such as control theory, economics, and robotics. In this paper, we propose two active-set schemes for solving convex QP problems by exploiting structural properties of the Karush–Kuhn–Tucker (KKT) system. The proposed methods are implemented and tested on randomly generated problems as well as benchmark instances from the CUTEst test collection. In total, 2000 numerical experiments are conducted to evaluate the performance of the algorithms. The numerical results indicate that the proposed approaches reduce computational time compared with the classical active-set method, with the first strategy performing better for problems with few constraints and the second showing improved efficiency when the number of independent constraints is large.

Keyword: Quadratic programming, Active Set, SVD, Spectral Decomposition.

ARTICLE INFO

Article history:
Research paper
Received 10, May 2026
Accepted 07, June 2026
Available online 13, June 2026

AMS subject Classification: 90C20, 65K05.

*Corresponding author: N. Bagherpour. Email: negin.bagherpour@ut.ac.ir

1 Introduction

Convex quadratic programming (CQP) is a fundamental class of optimization problems with numerous applications in areas such as control theory, financial modeling, and engineering design. Many practical problems can be formulated as CQPs, including those arising from mass–inertia modeling, portfolio optimization, and intelligent control systems. In addition, convex quadratic programs frequently appear as subproblems in more general optimization frameworks. For instance, sequential quadratic programming (SQP), one of the most widely used methods for nonlinear constrained optimization, generates a quadratic programming subproblem at each iteration by using a quadratic approximation of the objective function together with linear approximations of the constraints.
For convex quadratic programming problems with only equality constraints, the solution can be obtained by solving the linear system derived from the Karush–Kuhn–Tucker (KKT) optimality conditions. When inequality constraints are present, however, the problem becomes more challenging and specialized algorithms are required [1]. Various algorithmic strategies have been proposed in the literature for solving quadratic programming problems. Among the classical approaches, simplex-type methods for quadratic programming have also been studied extensively, including Wolfe's method [2], Swarup's simplex method [3], and the method proposed by Gupta and Sharma [4]. Among the most widely used approaches are active-set methods, which iteratively identify the set of constraints that are active at the solution and solve a sequence of equality-constrained quadratic subproblems [5, 6]. These methods are particularly attractive for medium-scale problems and for applications in which warm-start capabilities are important. In addition to active-set techniques, other classes of algorithms have been developed for quadratic programming, including interior-point methods [7] and more recent operator-splitting approaches designed for large-scale problems [8]. Despite these developments, improving the efficiency of solving the linear systems arising from the KKT conditions remains an important research direction in the design of quadratic programming algorithms.
In this paper, we develop two active-set strategies for solving convex quadratic programming problems by exploiting structural properties of the KKT system. The main idea is to combine the KKT equations in a way that leads to a reduced formulation with lower computational complexity. The resulting system can then be treated through a null-space computation that is equivalent to performing a singular value decomposition, which improves the efficiency of solving the linear systems arising within the active-set framework.
The contributions of this paper can be summarized as follows. First, we present a reformulation of the KKT system that enables a more efficient computation within the active-set method. Second, based on this reformulation, we propose two algorithmic strategies designed for different constraint structures. Finally, the proposed approaches are evaluated through numerical experiments on randomly generated problems and benchmark instances, demonstrating that the new strategies can solve the KKT systems more efficiently than the classical active-set approach.
The remainder of the paper is organized as follows. In Section 2, the classical active-set

algorithm for convex quadratic programming is briefly reviewed. Sections 3 and 4 present the proposed strategies for solving the KKT systems more efficiently. Numerical results are reported in Section 5 to demonstrate the performance of the proposed methods.

2 Active Set Method

Lets consider the convex quadratic programming problem

$$\begin{aligned} \min \quad & \frac{1}{2}X^TQX + q^TX \\ s.t. \quad & AX = b \\ & GX \leq h \end{aligned}$$

where $Q \in \mathrm{R}^{n\times n}$ is positive definite and $A \in \mathrm{R}^{m\times n}$, $G \in \mathrm{R}^{k\times n}$, $q \in \mathrm{R}^{n\times 1}$, $b \in \mathrm{R}^{m\times 1}$ and $h \in \mathrm{R}^{k\times 1}$ are arbitrary matrices and vectors. In the so-called active set algorithm, each iteration consists of the following main steps:

1) determining the current active set

2) computing the desired direction

3) computing the Lagrange multipliers and updating the active set

More precisely, with $X_0 \in \mathrm{R}^n$ being the starting point, the current active set would be a matrix A_0 consisting of A and $G(I)$ which refers to the rows I of G and satisfies $G(I)X_0 = h(I)$. Hence, we have

$$A_0 = \begin{bmatrix} A \\ G(I) \end{bmatrix},$$
$$A_0X_0 = \begin{bmatrix} b \\ h(I) \end{bmatrix}. \tag{1}$$

Now, a feasible point with enough reduction to $\Phi(X) = \frac{1}{2}X^TQX + q^TX$ is needed. To compute such a direction, we solve the quadratic programming problem

$$\begin{aligned} \min \quad & \frac{1}{2}P^TQP + X_0^TQP + q^TP. \\ s.t. \quad & A_0P = 0 \end{aligned} \tag{2}$$

The corresponding KKT system for (2) would be

$$QP + A_0^T\lambda = -(QX_0 + q) \tag{3}$$

$$A_0 P = 0 \tag{4}$$

Before starting the next iteration, we need to update both the solution and the active set. The indices for which the Lagrange multipliers are positive will remain in active set while the constraint corresponding to the most negative Lagrange multiplier would be substitute by a new active constraint. In addition, the solution approximation would become $X = X_0 + \alpha P$ where α is set to be the largest value for which X is feasible. These steps are summarized in Algorithm 1 [5].

Algorithm 1 Active Set Algorithm for Convex Quadratic programming

```
Compute the direction P by solving (2)
if P = 0 then
    Compute the Lagrange multipliers λ from (3)
    if λ ≥ 0 then
        X is the solution: STOP
    else
        remove the constraint corresponding to the smallest Lagrange multiplier
    end if
else
    compute the maximum possible α value for which X + αP is a feasible solution
    add the blocking constraint to the active set
end if
```

3 New Scheme

The KKT conditions for solving convex quadratic problems might be established in the matrix form [1] to optimize the computations based on special matrix structures. In this paper, we develop two ideas to benefit from these special structures.We note that solving the QP subproblem (2) in Algorithm 1 needs to be reasonably easier than the original QP problem. Here, a cost-effective algorithm is suggested to solve (2). Let rank of A_0 be equal to k and its SVD decomposittion be $U\Sigma V_k^T$ with $U \in \mathrm{R}^{s\times s}$ and $V \in \mathrm{R}^{n\times n}$ being orthonormal and $\Sigma \in \mathrm{R}^{k\times k}$ being diagonal. We note that (4) gives $P = V_{n-k}Z \in \mathrm{Null}(A_0)$, while (3) results in

$$P + r_0 = V_k S \in \mathrm{R}(A_0^T),$$

where, $r_0 = QX_0 + q$. Hence,

$$P = V_{n-k}Z = V_k S - r_0$$

and

$$(V_k \quad V_{n-k}) \begin{pmatrix} S \\ -Z \end{pmatrix} = r_0.$$

Since V is orthonormal, we now get

$$P = -V_{n-k}V_{n-k}^T r_0, \tag{5}$$

which is the solution of (2).

4 Another Approach

We note that by use of a change of variable $\tilde{X} = Q^{\frac{1}{2}}X$, the coefficient matrix Q can be eliminated in (1). So, without loss of generality, here we assume $Q = I$. The KKT condition for (2) will be

$$P + A_0^T\lambda = -(X_0 + q) \tag{6}$$
$$A_0 P = 0 \tag{7}$$

Substituting (7) in (6), we have $P^T P = -r_0^T P$ in which $r_0 = X_0 + q$. We then let $P = V_{n-k}Z$ and $\tilde{r_0} = V_{n-k}r_0$ to get

$$Z^T Z = -\tilde{r_0}^T Z. \tag{8}$$

We note that the most important benefit of this strategy is reducing the dimension of the problem to $n-k$. So, this is of more interest in problems with greater number of independent constraints. The solution set of (8) clearly consists of the points on the sphere

$$\|Z + C\| = \|C\| \tag{9}$$

where $C = \frac{\tilde{r_0}}{2}$. When $n-k$ is small enough, this approach is of significant interest. In case of $n-k=1$ and $n-k=2$ the results are given in Table 1.

Table 1: Simple and common cases

	$n-k=1$	$n-k=2$
Z	$Z = -2C$	$Z = -C \pm \|C\| \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix}$
P	$P = V_{n-k}Z$	$P = V_{n-k}Z$

Let describe the process in details. First, the argument θ needs to be computed so that

$$V_{n-k}^T r_0 = -Z = C \mp \|C\| \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix}.$$

Then, P is calculated from Z. So far, we presented two schemes for solving the subproblem presented in Algorithm 1 for computing the direction P. Theses are summarized in Table 2 below.
Next, in Section 5 numerical test results are reported to confirm the efficiency of Algorithm 1 by both schemes for computing P.

Table 2: Strategies for computing P

	Schme 1 (Section 3)	Scheme 2 (Section 4)
usage	otherwise	low $n-k$
P	$P=-V_{n-k}V_{n-k}^T r_0$	$P=V_{n-k}Z$ with Z on a sphere

5 Numerical Results

We first used Scheme 1 to solve 1000 randomly generated test problems with up to 600 unknown variables. The random tests are generated with a good variety with different number of equality and inequality constraints, n_e and n_i. The computing times for the original active set algorithm is compared with Scheme 1 in Figure 1.

For larger values of n and small number of constraints, Scheme 1 compute the solution faster. According to last row of figure 1, Scheme 1 fails to compute the solution in a competitive time. Now, we compare the computing times of the original active set and Scheme 2 in Figure 2. In this experiment, tests are conducted with high number of independent constraints.

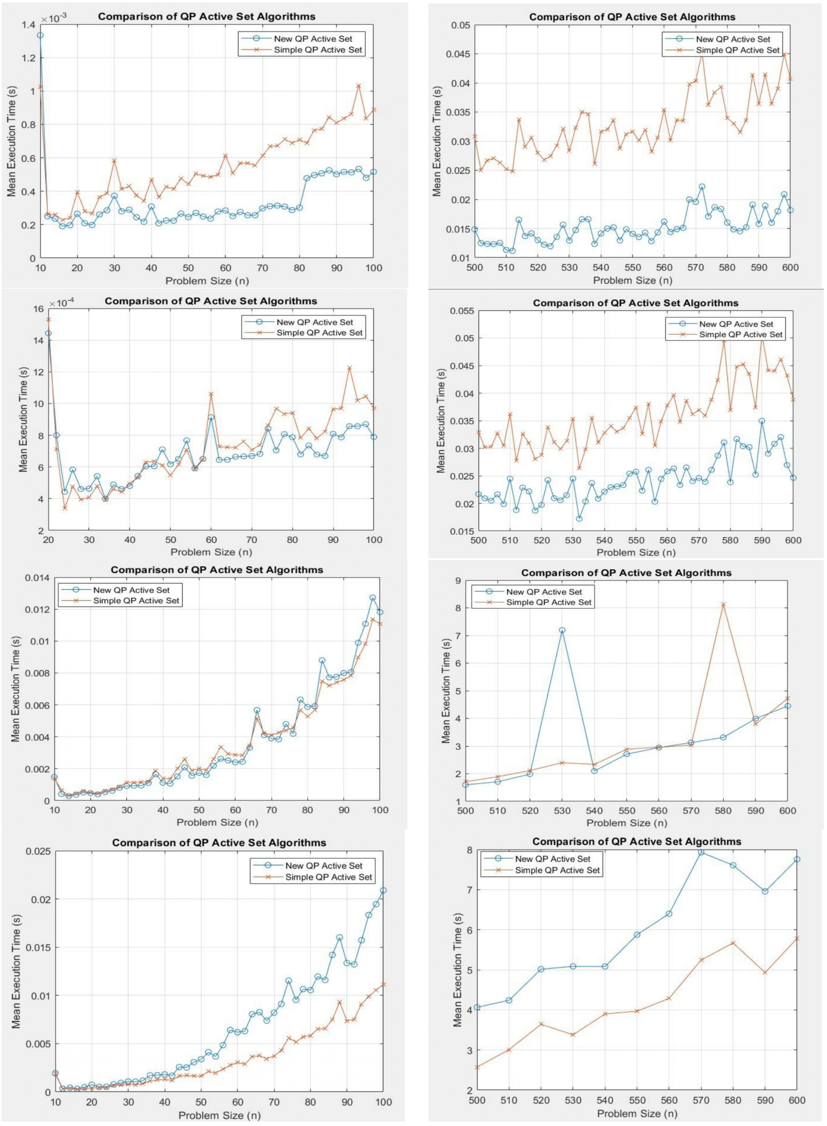


Figure 1: Time Comparison of the Original Active Set and Scheme 1

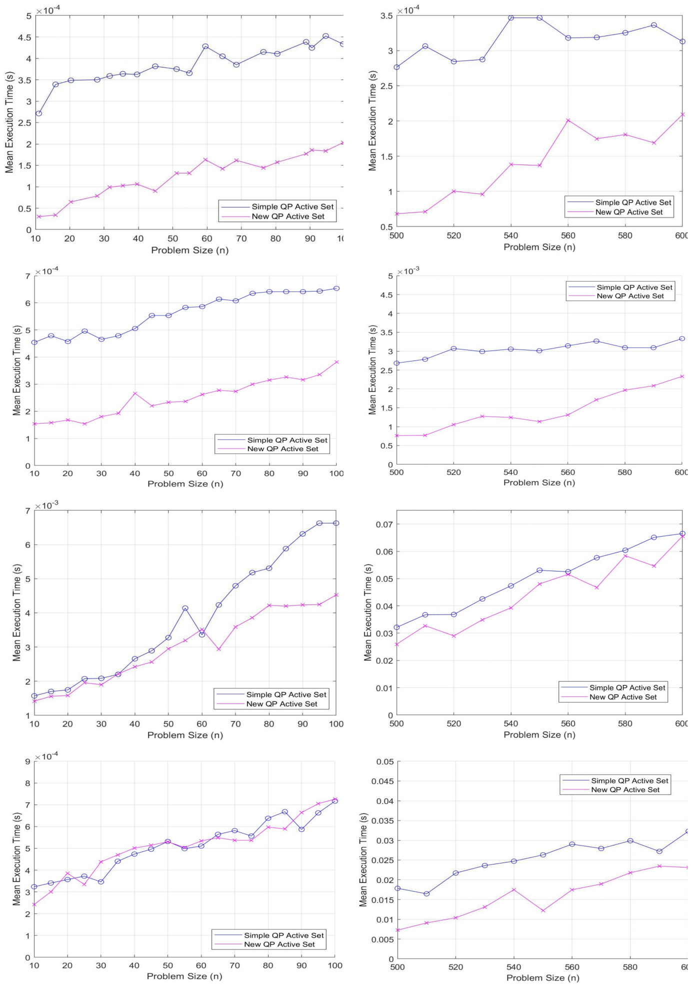


Figure 2: Time Comparison of the Original Active Set and Scheme 2

Since, Scheme 1 failed to solve some of the test problems in the expected time, we need to confirm its convergence regardless of computing times. For the proposed scheme, we present the average error norm in Figure 3 to confirm that although the computing time is not competitive, the convergence has been achieved. The error norm here is defined to be the norm of difference between the computed solution and the exact solution computed by quadprog command of MATLAB.

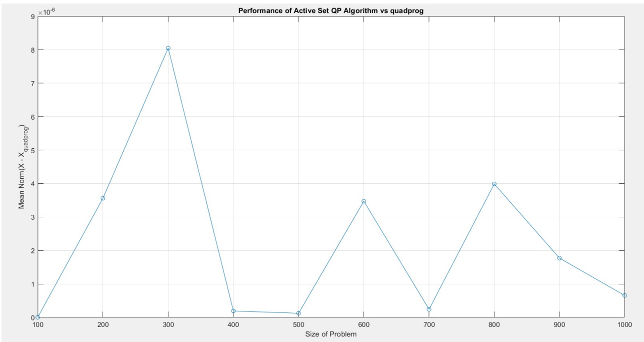


Figure 3: Error Norm for Scheme 1

Finally, in Figure 4, the Dolan More [9] time profiles are presented to compare Algorithm 1 with scheme 1 and 2 in solving the QP standard tests introduced in CUTEst [10]. Considering the presented profile, it can be seen that Sceme 2 outperforms other in converging to the solution faster. Moreover, it is the only method (between the three) which is able to solve all the CUTEst test problems in at most four time the best.

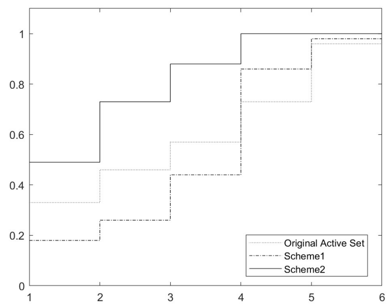
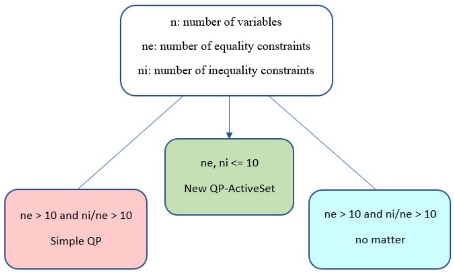


Figure 4: Dolan More Time Comparison

6 Concluding Remarks

Quadratic programming is one of the most important parts in optimization mainly because of variety of applications. The performance of different techniques for solving such problems considerably depends on size. In this paper we present a new scheme which



is originally based on active set algorithm while using optimal matrix decompositions to manage the subproblems. Based on the presented numerical results, the best performance of the suggested algorithm is when the number of constraints, both equality and inequality, are relatively low. We suggest making use of the proposed active set scheme for nonlinear optimization problems in SQP method. There, solving a quadratic programming is needed in each iteration and the lower computing time by use of the new active set scheme would be more important. On the other hand, we note that special matrix structures in which the decompositions are easier, can be categorized in future research. The following flowchart restate our results:

References

[1] Z. Iqbal and S. Nooshabadi, *Exploiting block structures of kkt matrices for efficient solution of convex optimization problems*, IEEE Access, 9:116604–116611, 2021.

[2] Ph. Wolfe, *The simplex method for quadratic programming*, Econometrica, 27:382–398, 1959.

[3] N. Suleiman and M. Nawkhass, *A new modified simplex method to solve quadratic fractional programming problem and compared it to a traditional simplex method by using pseudoaffinity of quadratic fractional functions*, Appl. Math. Sci., 7:3749–3764, 2013.

[4] A. Gupta and J. Sharma., *A generalized simplex technique for solving quadratic-programming problem*, Indian J. Technol., 21:198–201, 1983.

[5] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, New York, 2006.

[6] D. Goldfarb and A. Idnani, A numerically stable dual method for solving strictly convex quadratic programs, *Mathematical Programming*, 27 (1983), 1–33.

[7] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, 1997.

[8] B. Stellato, G. Banjac, P. Goulart, A. Bemporad and S. Boyd, OSQP: An operator splitting solver for quadratic programs, *Mathematical Programming Computation*, 12 (2020), 637–672.

[9] E. Dolan and J. B. More, *Benchmarking optimization software with performance profiles*, Math. Prog., 91:201–213, 2002.

[10] N. I. M. Gould, D. Orban, and Ph. Toint, *Cutest: A constrained and unconstrained testing environment, latest*, visited December 2024.