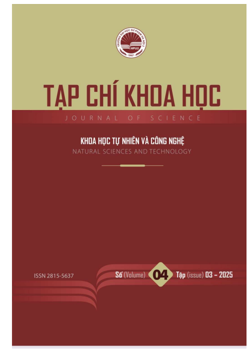
HPU2 Journal of Sciences:
Natural Sciences and Technology

Journal homepage: https://sj.hpu2.edu.vn



Article type: Research article

Quadratic programming and quadratically constrained quadratic programming: theory, algorithms, and applications

Kim-Thuy Dinh Thi*

Hanoi Pedagogical University 2, Phu Tho, Vietnam

Abstract

This survey provides a systematic review of quadratic programming (QP) and quadratically constrained quadratic programming (QCQP) problems. The paper reviews mathematical formulations and problem taxonomies based on convexity properties, surveys optimality conditions through Lagrangian theory and KKT conditions, and examines foundational work by Markowitz, Wolfe, and Frank-Wolfe and key algorithmic developments. Four major algorithmic paradigms are examined: (1) active-set methods with finite convergence properties, (2) polynomial-time interior-point methods, (3) modern operator-splitting approaches including OSQP and ADMM, and (4) semidefinite programming relaxations for non-convex cases, with review of their theoretical properties and convergence guarantees. The methodology is illustrated through case studies in facility location optimization and production planning that demonstrate the application of KKT conditions and Lagrange multiplier theory, while examples from portfolio optimization to model predictive control illustrate broader applicability. This work connects classical optimization theory with contemporary algorithmic approaches, providing insights for researchers and guidance for practitioners in operations research, engineering, and applied mathematics.

Keywords: Quadratic programming, QCQP, Lagrange multipliers, KKT conditions, constrained optimization, facility location, production planning

1. Introduction

Quadratic programming represents one of the most fundamental and well-studied classes of optimization problems, with roots tracing back to the pioneering work of Markowitz [1] in portfolio theory and the subsequent developments by Wolfe [2] and Frank and Wolfe [3] in algorithmic approaches. The extension to quadratically constrained quadratic programs emerged naturally as researchers recognized

* Corresponding author, E-mail: dinhthikimthuy@hpu2.edu.vn

https://doi.org/10.56764/hpu2.jos.2025.4.3.80-96

Received date: 11-7-2025 ; Revised date: 28-7-2025 ; Accepted date: 20-11-2025

the need to model more complex real-world phenomena involving nonlinear relationships in both objectives and constraints.

The significance of QP extends far beyond its mathematical elegance. These problems arise naturally in numerous applications including portfolio optimization, support vector machines [4], [5], model predictive control [6], [7], and facility location problems. The computational tractability of convex QP problems, combined with their modeling flexibility, has made them indispensable tools in modern optimization practice.

Main contributions. We provide a systematical review of quadratic programming (QP) and quadratically constrained quadratic programming (QCQP), outlining problem structures, key optimality conditions, and influential algorithmic paradigms including active-set, interior-point, operator-splitting (OSQP, ADMM), and semidefinite relaxations. Notably, highlights on practical relevance through case studies in facility location and production planning are presented. The paper effectively bridges foundational theory, modern algorithms, and real-world applications, offering guidance for researchers and practitioners alike.

1.1. Canonical Formulation of Quadratic Programming

Definition 1.1 (Quadratic Programming Problem). A *quadratic programming problem* is an optimization problem of the form:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) = \frac{1}{2}x^\top Qx + c^\top x + r \\ \text{subject to} \quad & Ax \leq b \\ & Ex = d \\ & x \geq 0 \end{aligned}$$

where $x \in \mathbb{R}^n$ is the vector of decision variables, $Q \in \mathbb{S}^n$ is a symmetric $n \times n$ matrix (e.g. an Hessian of an objective), $c \in \mathbb{R}^n$ is the linear coefficient vector, $r \in \mathbb{R}$ is a scalar constant, $A \in \mathbb{R}^{m\times n}$ defines inequality constraints with $b \in \mathbb{R}^m$, $E \in \mathbb{R}^{p\times n}$ defines equality constraints with $d \in \mathbb{R}^p$. When Q is positive semidefinite ($Q \succcurlyeq 0$), the problem is convex and possesses desirable computational properties.

1.2. Extension to Quadratically Constrained Quadratic Programming (QCQP)

A *quadratically constrained quadratic programming problem* (QCQP) is an optimization problem in which both the objective function and the constraints are quadratic functions.

Definition 1.2 (Quadratically Constrained Quadratic Programming). A QCQP extends QP by allowing quadratic constraints

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f_0(x) = \frac{1}{2}x^\top Qx + c^\top x + r \\ \text{subject to} \quad & f_i(x) = \frac{1}{2}x^\top P_i x + q_i^\top x + r_i \leq 0, \quad i = 1, \dots, m \\ & Ex = d \end{aligned}$$

where $P_i \in \mathbb{S}^n$ for $i = 0,1, \dots, m$ are symmetric matrices defining the quadratic terms in the objective and constraint functions.

If $P_1, \dots, P_m$ and Q are all positive semidefinite, then the problem is convex. When $P_1, \dots, P_m$ are all zero, then the problem is a quadratic program as all the constraints are linear.

1.3. History and Development of QCQP

Foundational Contributions. The theoretical foundations of quadratic programming were established through several seminal contributions in the mid-20th century. Markowitz [1] introduced quadratic programming in the context of portfolio optimization, formulating the mean-variance model that became a cornerstone of modern finance theory. This work demonstrated the practical importance of optimization problems with quadratic objectives and linear constraints. The algorithmic development of QP methods began with the work of Wolfe [2], who proposed the first systematic approach for solving QP problems using variants of the simplex method. Concurrently, Frank and Wolfe [3] developed gradient-based methods that could handle more general convex programming problems, including QP as a special case.

Algorithmic Evolution. The 1970s and 1980s witnessed significant advances in QP algorithms. Gill and Murray [8] developed numerically stable active-set methods that became the standard approach for medium-scale problems. This work was extended by Goldfarb and Idnani [9], who developed a dual active-set method that remains widely implemented in modern solvers. These methods systematically identify the optimal active constraint set by solving a sequence of equality-constrained quadratic subproblems. The introduction of interior-point methods by Karmarkar [10] for linear programming was quickly extended to QP by Megiddo [11] and others. Mehrotra's predictor-corrector method [12] further improved the practical performance of interior-point algorithms. These methods demonstrated polynomial-time complexity and excellent practical performance, particularly for large-scale problems with sparse structure.

Modern Developments. The 21st century has seen remarkable progress in both algorithmic sophistication and software implementation. The development of OSQP (Operator Splitting Quadratic Program) by Stellato et al. [13] represents a significant breakthrough in making high-performance QP solvers accessible through open-source software. OSQP’s operator-splitting approach enables efficient warm-starting and real-time applications. Other important modern QP solvers include qpOASES [14] for parametric problems arising in model predictive control and commercial solvers such as Gurobi [15], CPLEX [16], and MOSEK [17]. Research in quadratically constrained quadratic programming has focused primarily on relaxation techniques due to the general NP-hardness of non-convex QCQP. The seminal work of Shor [18] on semidefinite relaxations provided the theoretical foundation for many modern approaches.

2. Preliminaries, Taxonomy and Structural Properties

2.1. Fundamental Properties and Definitions

Definition 2.1 (Positive Definite and Semidefinite Matrices). Let $Q \in \mathbb{S}^n$.

- Q is *positive definite* (denoted $Q \succ 0$) if $x^\top Q x > 0$ for all $x \neq 0$
- Q is *positive semidefinite* (denoted $Q \succcurlyeq 0$) if $x^\top Q x \geq 0$ for all $x \in \mathbb{R}^n$.

Theorem 2.2 (Spectral Characterization). A symmetric matrix $Q \in \mathbb{S}^n$ is positive definite if and only if all its eigenvalues are strictly positive. Similarly, Q is positive semidefinite if and only if all its eigenvalues are nonnegative.

Proof. By the spectral theorem, any symmetric matrix Q can be diagonalized as $Q = U \Lambda U^\top$ where U is an orthogonal matrix whose columns are the eigenvectors of Q, and $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_n)$ contains the eigenvalues $\lambda_1, \dots, \lambda_n$.

For any $x \neq 0$, let $y = U^\top x$. Since U is orthogonal, $\| y \| = \| x \| > 0$, so $y \neq 0$. Then:

$$x^\top Q x = x^\top U \Lambda U^\top x = y^\top \Lambda y = \sum_{i=1}^{n} \lambda_i y_i^2$$

Since $y \neq 0$, at least one component $y_i \neq 0$, so $y_i^2 > 0$. Therefore $x^\top Q x > 0$ for all $x \neq 0$ if and only if $\lambda_i > 0$ for all i; and $x^\top Q x \geq 0$ for all x if and only if $\lambda_i \geq 0$ for all i.

This completes the proof. □

Theorem 2.3 (Global Optimality for Convex QP). Consider the QP problem (1.1) with $Q \succcurlyeq 0$. If the feasible region is nonempty and bounded, then: 1. The problem has a global minimum 2. Any local minimum is also a global minimum

3. If $Q \succ 0$, the global minimum is unique

Proof. (1) Since $Q \succcurlyeq 0$, the objective function $f(x) = \frac{1}{2} x^\top Q x + c^\top x + r$ is convex. The feasible region defined by linear constraints is a convex polyhedron. A continuous function on a compact convex set attains its minimum.

For convex functions on convex sets, any local minimum is necessarily global. To see this, suppose x^* is a local minimum but not global, so there exists a feasible $\bar{x}$ with $f(\bar{x}) < f(x^*)$. For any $\alpha \in (0,1)$, the point $x_\alpha = \alpha \bar{x} + (1-\alpha) x^*$ is feasible by convexity. By convexity of f:

$$f(x_\alpha) \leq \alpha f(\bar{x}) + (1-\alpha) f(x^*) < f(x^*)$$

For sufficiently small α, x_α is arbitrarily close to x^*, contradicting the local minimality of x^*.

When $Q \succ 0$, the objective function is strictly convex. Suppose x^* and $\bar{x}$ are both global minima with $x^* \neq \bar{x}$. Then for $\alpha \in (0,1)$:

$$f(\alpha x^* + (1-\alpha)\bar{x}) < \alpha f(x^*) + (1-\alpha) f(\bar{x}) = f(x^*)$$

This contradicts the global minimality of x^*. □

2.2. Problem Classification

The computational complexity and solution approaches for quadratic programming problems depend critically on the properties of the matrices involved. We present a comprehensive taxonomy based on these structural characteristics.

Table 1. Classification of QCQP Problems.

Problem Class	Mathematical Definition	Complexity	Representative Applications
Linear Programming (LP)	$Q = 0$ in (1.1)	P	Transportation, resource allocation
Convex QP	$Q \succcurlyeq 0$ in (1.1)	P	Portfolio optimization, SVM dual [4], [5]
Non-convex QP	Q indefinite in (1.1)	NP-hard	AC optimal power flow
Convex QCQP	$P_i \succcurlyeq 0, \forall i$ in (1.2)	P (with qualification)	Trust region subproblems [6], [7]
Non-convex QCQP	Some P_i indefinite in (1.2)	NP-hard	AC power flow, facility location
Mixed-Integer QP (MIQP)	Mixed integer and continuous variables	NP-hard	Production planning, scheduling

Definition 2.4 (Convex Quadratic Programming). A QP problem (1.1) is *convex* if $Q \succcurlyeq 0$. In this case, the problem belongs to the class P (polynomial-time solvable).

Definition 2.5 (Non-convex Quadratic Programming). A QP problem (1.1) is *non-convex* if Q has at least one negative eigenvalue. Such problems are generally NP-hard.

2.3. Special Structure and Tractable Cases

Theorem 2.6 (Exactness of SDP Relaxation). Consider a QCQP of the form (1.2). If all matrices P_i have at most one positive eigenvalue, then the semidefinite programming relaxation provides the exact optimal value.

This result follows from the S-lemma [19] and the structure of the optimal solution to the SDP relaxation. The key insight is that under these conditions, the rank-one constraint $X = xx^\top$ is automatically satisfied at the SDP optimum. For a complete proof, see Luo et al. [20].

2.4. Geometric Interpretation

The feasible region of a QCQP is characterized by the intersection of ellipsoids (when $P_i \succ 0$), hyperplanes (linear constraints), and possibly non-convex quadratic surfaces. This geometric perspective provides important insights:

- **Convex Case**: The feasible region is convex, and any local optimum is global
- **Non-convex Case**: Multiple local optima may exist, requiring global optimization techniques
- **Degenerate Cases**: When constraint matrices are singular, the feasible region may be unbounded or empty.

3. Optimality Conditions and Lagrangian Theory

The method of Lagrange multipliers provides the fundamental theoretical framework for characterizing optimal solutions to constrained optimization problems. We develop this theory systematically for both QP and QCQP problems.

Definition 3.1 (Lagrangian Function). For the QCQP problem (1.2), the *Lagrangian function* is defined as

$$\mathcal{L}(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \nu^\top (Ex - d)$$

where $x \in \mathbb{R}^n$ is the primal variable vector, $\lambda = (\lambda_1, \dots, \lambda_m) \in \mathbb{R}^m$ are the inequality constraint multipliers, and $\nu \in \mathbb{R}^p$ are the equality constraint multipliers.

3.1. First-Order Necessary Conditions (KKT Conditions)

Definition 3.2 (Regular Point). A feasible point x^* is called *regular* if the gradients of all active constraints are linearly independent.

Theorem 3.3 (Karush-Kuhn-Tucker Necessary Conditions). Let x^* be a local minimum of the QCQP problem (1.2), and assume that x^* is a regular point. Then there exist multipliers $\lambda^* \in \mathbb{R}^m$ and $\nu^* \in \mathbb{R}^p$ such that:

$$\begin{aligned}
\nabla f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla f_i(x^*) + E^\top \nu^* &= 0 && \text{(Stationarity)} \\
f_i(x^*) &\leq 0, \quad i = 1, \dots, m && \text{(Primal feasibility)} \\
Ex^* &= d && \text{(Equality feasibility)} \\
\lambda_i^* &\geq 0, \quad i = 1, \dots, m && \text{(Dual feasibility)} \\
\lambda_i^* f_i(x^*) &= 0, \quad i = 1, \dots, m && \text{(Complementary slackness)}
\end{aligned}$$

Proof. We provide a complete proof using the method of Lagrange multipliers.

Let $\mathcal{A}(x^*) = \{i: f_i(x^*) = 0\}$ denote the active inequality constraints at x^*. Since x^* is a local minimum, there exists a neighborhood $N(x^*)$ such that $f_0(x) \geq f_0(x^*)$ for all feasible $x \in N(x^*)$.

Step 1: Define the cone of feasible directions at x^*:

$$\mathcal{F}(x^*) = \{d \in \mathbb{R}^n: \nabla f_i(x^*)^\top d \leq 0, \, i \in \mathcal{A}(x^*), \, Ed = 0\}$$

Step 2: For any $d \in \mathcal{F}(x^*)$, there exists $\alpha_0 > 0$ such that $x^* + \alpha d$ is feasible for all $\alpha \in (0, \alpha_0)$. Since x^* is a local minimum:

$$f_0(x^* + \alpha d) \geq f_0(x^*)$$

Step 3: Taking the directional derivative:

$$\lim_{\alpha \to 0^+} \frac{f_0(x^* + \alpha d) - f_0(x^*)}{\alpha} = \nabla f_0(x^*)^\top d \geq 0$$

Therefore, $\nabla f_0(x^*)^\top d \geq 0$ for all $d \in \mathcal{F}(x^*)$.

Step 4: By the Fundamental Theorem of Linear Programming (Farkas' Lemma), there exist $\lambda_i^* \geq 0$ for $i \in \mathcal{A}(x^*)$ and $\nu^* \in \mathbb{R}^p$ such that:

$$\nabla f_0(x^*) + \sum_{i \in \mathcal{A}(x^*)} \lambda_i^* \nabla f_i(x^*) + E^\top \nu^* = 0$$

Step 5: Set $\lambda_i^* = 0$ for $i \notin \mathcal{A}(x^*)$. Then $\lambda_i^* f_i(x^*) = 0$ for all i, establishing complementary slackness.

The remaining conditions follow directly from the problem definition and the construction.

3.2. Second-Order Conditions

Definition 3.4 (Active Set). At a feasible point x, the *active set* is defined as:

$$\mathcal{A}(x) = \{i: f_i(x) = 0\} \cup \{i: Ex = d\}$$

Theorem 3.5 (Second-Order Sufficient Conditions). Let (x^*, λ^*, ν^*) satisfy the KKT conditions (3.3). Define the *Lagrangian Hessian*:

$$\nabla^2 \mathcal{L}(x^*, \lambda^*, \nu^*) = \nabla^2 f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla^2 f_i(x^*)$$

If this Hessian is positive definite on the subspace:

$$\mathcal{T}(x^*) = \{d \in \mathbb{R}^n: \nabla f_i(x^*)^\top d = 0, \, i \in \mathcal{A}(x^*), \, Ed = 0\}$$

then x^* is a strict local minimum.

The proof follows from a second-order Taylor expansion analysis. For details, see Nocedal and Wright Chapter 12 [21].

3.3. Specialized Results for Quadratic Programming

For the standard QP problem (1.1), the KKT conditions simplify considerably due to the quadratic structure.

Theorem 3.6 (QP Optimality Conditions). For the QP problem (1.1), the KKT conditions become:

$$\begin{aligned} Qx^* + c + A^\top\lambda^* + E^\top\nu^* &= 0 \\ Ax^* &\leq b, \quad \lambda^* \geq 0, \quad \lambda^{*\top}(Ax^* - b) = 0 \\ Ex^* &= d \end{aligned}$$

Proof. For the QP problem, $\nabla f_0(x) = Qx + c$ and the constraints are linear, so $\nabla f_i(x) = A_i$ (the i-th row of A). Substituting into the general KKT conditions (Theorem 3.3) yields the result directly. □

Corollary 3.7 (Convex QP Global Optimality). If $Q \succcurlyeq 0$ in problem (1.1), then any point satisfying the KKT conditions (4.2) is a global optimum. If additionally $Q \succ 0$, the global optimum is unique.

Proof. This follows immediately from Theorem 1.2 and the equivalence of KKT conditions with global optimality for convex problems. □

3.4. Constraint Qualifications

The regularity condition in Theorem 4.1 is one of several *constraint qualifications* that ensure the KKT conditions are necessary for optimality.

Definition 3.8 (Linear Independence Constraint Qualification - LICQ). The LICQ holds at x^* if the gradients $\{\nabla f_i(x^*): i \in \mathcal{A}(x^*)\}$ are linearly independent.

Definition 3.9 (Mangasarian-Fromovitz Constraint Qualification - MFCQ). The MFCQ holds at x^* if:

1. The gradients of equality constraints are linearly independent

2. There exists $d \in \mathbb{R}^n$ such that $\nabla f_i(x^*)^\top d < 0$ for all $i \in \mathcal{A}(x^*)$ corresponding to inequality constraints

These constraint qualifications are progressively weaker (LICQ ⇒ MFCQ), with MFCQ being sufficient for the KKT conditions to hold at local optima.

4 Algorithmic Approaches

4.1. Active-Set Methods

Active-set methods represent one of the most fundamental and well-established approaches for solving quadratic programming problems. These methods work by systematically identifying the optimal active constraint set through a sequence of equality-constrained quadratic subproblems.

Algorithm 4.1 (Primal Active-Set Method). Given a QP problem (1.1):

Initialization: Choose a feasible starting point $x^{(0)}$ and an initial working set $\mathcal{W}^{(0)}$

Iteration k: Given $x^{(k)}$ and $\mathcal{W}^{(k)}$, solve the equality-constrained quadratic program:

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \frac{1}{2} d^\top Q d + \left(Qx^{(k)} + c\right)^\top d \\ \text{subject to} \quad & A_{\mathcal{W}^{(k)}} d = 0 \\ & Ed = 0 \end{aligned}$$

where $A_{\mathcal{W}^{(k)}}$ represents the rows of A corresponding to the working set.

Optimality Test: If $d^{(k)} = 0$, check the multiplier signs. If all multipliers are non-negative, terminate with optimal solution $x^{(k)}$.

Working Set Update: If $d^{(k)} \neq 0$, perform a line search and update the working set by adding or removing constraints.

Theorem 4.2 (Finite Convergence). For non-degenerate QP problems with $Q \succ 0$, the active-set method converges in a finite number of iterations, bounded by $\binom{m+p}{n}$ where m is the number of inequality constraints.

The proof follows from the fact that each iteration either decreases the objective value or changes the working set. Since there are only finitely many possible working sets and no working set can be repeated (due to non-degeneracy), the algorithm must terminate. For numerical stability considerations in implementing active-set methods, see Goldfarb and Idnani [9] and the matrix computation techniques in Golub and Van Loan [22], or see Nocedal and Wright [21], Chapter 16, for complete details.

4.2. Interior-Point Methods

Interior-point methods have revolutionized large-scale quadratic programming by achieving polynomial-time complexity and excellent practical performance on sparse problems.

Algorithm 4.3 (Primal-Dual Interior-Point Method). For the QP problem (1.1), introduce slack variables $s \geq 0$ and consider the barrier subproblem:

$$\begin{aligned} \min_{x,s} \quad & \frac{1}{2}x^\top Qx + c^\top x - \mu\sum_{i=1}^{m} \ln s_i \\ \text{subject to} \quad & Ax + s = b \\ & Ex = d \end{aligned}$$

The perturbed KKT conditions are:

$$\begin{aligned} Qx + c - A^\top\lambda - E^\top\nu &= 0 \\ Ax + s - b &= 0 \\ Ex - d &= 0 \\ S\Lambda e - \mu e &= 0 \\ s, \lambda &\geq 0 \end{aligned}$$

where $S = \mathrm{diag}(s)$, $\Lambda = \mathrm{diag}(\lambda)$, and e is the vector of ones.

Theorem 4.4 (Polynomial Complexity). The primal-dual interior-point method for QP requires at most $O\left(n^{1.5}\ln(\epsilon^{-1})\right)$ iterations to achieve ϵ-optimality, where each iteration costs $O(n^3)$ for dense problems.

This follows from the general theory of interior-point methods for convex optimization. The predictor-corrector variant [12] typically requires fewer iterations in practice. See Wright [23] for a comprehensive treatment.

4.3. Operator-Splitting Methods (ADMM Framework)

The Alternating Direction Method of Multipliers (ADMM) has gained significant attention for its ability to decompose large-scale problems and enable warm-starting in real-time applications.

Algorithm 4.5 (OSQP: Operator-Splitting QP). Consider the QP problem in the form:

$$\begin{aligned} \min_{x} \quad & \frac{1}{2}x^\top Qx + c^\top x \\ \text{subject to} \quad & l \leq Ax \leq u \end{aligned}$$

The ADMM iterations are:

$$\begin{aligned} x^{k+1} &= \operatorname*{argmin}_x \left\{ \frac{1}{2} x^\top Q x + c^\top x + \frac{\rho}{2} \| Ax - z^k + u^k \|^2 \right\} \\ z^{k+1} &= \Pi_{[l,u]}\left(Ax^{k+1} + u^k\right) \\ u^{k+1} &= u^k + Ax^{k+1} - z^{k+1} \end{aligned}$$

where $\Pi_{[l,u]}$ denotes projection onto the box $[l, u]$ and $\rho > 0$ is a penalty parameter.

Theorem 4.6 (ADMM Convergence). Under standard assumptions, the ADMM iterations (4.5) converge to the optimal solution with $O(1/k)$ convergence rate in objective value.

See Boyd et al. [24] for the complete convergence analysis. For problems with parametric variations, specialized solvers like qpOASES [14] can exploit warm-starting more effectively than general-purpose methods. The theory of warm-starting for interior-point methods is developed in [25].

4.4. Semidefinite Programming Relaxations for QCQP

For non-convex QCQP problems, semidefinite programming provides a powerful relaxation framework that often yields tight bounds or exact solutions.

Algorithm 4.7 (SDP Relaxation). For the QCQP problem (1.2), introduce the matrix variable $X \in \mathbb{S}^{n+1}$ and consider:

$$\begin{aligned} \min_{X \succcurlyeq 0} \quad & \left\langle \begin{pmatrix} Q & \frac{1}{2}c \\ \frac{1}{2}c^\top & r \end{pmatrix}, X \right\rangle \\ \text{subject to} \quad & \left\langle \begin{pmatrix} P_i & \frac{1}{2}q_i \\ \frac{1}{2}q_i^\top & r_i \end{pmatrix}, X \right\rangle \leq 0, \quad i = 1, \dots, m \\ & X_{n+1,n+1} = 1 \end{aligned}$$

Theorem 4.8 (SDP Relaxation Quality). The SDP relaxation provides a lower bound on the optimal value of the QCQP. Under certain conditions (e.g., when all P_i have at most one positive eigenvalue), the relaxation is exact.

Kim and Kojima [26] provide conditions under which SDP relaxations are exact for specific classes of QCQP problems. The relaxation bound follows from the fact that any feasible solution to the original QCQP induces a rank-one feasible solution to the SDP. For exactness conditions, see Luo et al. [20].

4.5. Lagrange Multiplier Methods: Computational Implementation

The theoretical Lagrangian framework developed in Section 4 requires careful numerical implementation to ensure stability and efficiency.

Algorithm 4.9 (Newton-Lagrange Method). For equality-constrained QP:

$$\begin{aligned} \min_x \quad & \frac{1}{2} x^\top Q x + c^\top x + r \\ \text{subject to} \quad & Ex = d \end{aligned}$$

Form the KKT system:

$$\begin{pmatrix} Q & E^\top \\ E & 0 \end{pmatrix} \begin{pmatrix} x \\ \nu \end{pmatrix} = \begin{pmatrix} -c \\ d \end{pmatrix}$$

Theorem 4.10 (KKT System Solvability). If $Q \succ 0$ and E has full row rank, then the KKT matrix in (3.6) is nonsingular and the system has a unique solution.

Proof. The KKT matrix is a saddle-point matrix of the form $\begin{pmatrix} A & B^\top \\ B & 0 \end{pmatrix}$ where $A = Q \succ 0$ and $B = E$ has full row rank. To show nonsingularity, suppose $\begin{pmatrix} x \\ \nu \end{pmatrix}$ is in the null space:

$$Qx + E^\top \nu = 0 \quad \text{and} \quad Ex = 0$$

From the first equation: $x^\top Qx + x^\top E^\top \nu = 0$. From the second equation: $x^\top E^\top \nu = \nu^\top Ex = 0$. Therefore: $x^\top Qx = 0$. Since $Q \succ 0$, this implies $x = 0$. Substituting back: $E^\top \nu = 0$. Since E has full row rank, $E^\top$ has full column rank, so $\nu = 0$. Thus, the only solution to the homogeneous system is the trivial solution, proving nonsingularity. □

For sparse problems, specialized linear algebra techniques [27] can significantly improve the computational efficiency of solving these KKT systems. The numerical stability of these computations is thoroughly analyzed in [22].

4.6. Contemporary Algorithmic Comparison and Performance Analysis

The landscape of quadratic programming algorithms has evolved significantly with recent advances in computational hardware and specialized software implementations. This section provides a comprehensive comparison of the four major algorithmic paradigms based on recent empirical evidence and theoretical developments.

4.6.1. Comprehensive Algorithmic Comparison

Table 2 presents a systematic comparison of quadratic programming methods based on recent literature and standardized benchmarking studies. The analysis incorporates findings from the `qpbenchmark` project [31], which provides standardized evaluation across multiple solver implementations, and recent advances in GPU acceleration [32].

4.6.2. Empirical Performance Analysis

Recent standardized benchmarking studies provide quantitative evidence for algorithmic performance across different problem classes. The `qpbenchmark` project [31] has established comprehensive test suites, including the Maros-Meszaros collection, model predictive control problems, and community-contributed test cases.

GPU Acceleration Impact. The integration of graphics processing units has significantly impacted large-scale quadratic programming performance. Schubiger et al. [32] demonstrate substantial speedups for GPU-accelerated ADMM implementations, with particularly strong performance on problems with sparse structure and large numbers of variables. The authors report speedups of up to two orders of magnitude compared to CPU implementations on appropriately sized problems.

Interior-Point Method Advances. Recent developments in interior-point methods include proximal stabilization techniques that improve robustness and convergence properties [34]. These methods address numerical challenges in degenerate problems while maintaining polynomial-time complexity guarantees. The enhanced preconditioning strategies reduce the computational burden of Newton system solutions across multiple iterations.

4.6.3. Contemporary Software Ecosystem

The modern quadratic programming software landscape reflects both theoretical advances and practical implementation requirements. Open-source frameworks have gained significant adoption, with standardized interfaces enabling fair performance comparisons [31].

Benchmarking Infrastructure. The establishment of systematic benchmarking protocols [31], [35] has improved the ability to make objective algorithm comparisons. These benchmarks evaluate multiple metrics, including solution accuracy, computation time, and robustness across diverse problem instances. The Mittelmann benchmarks [35] provide additional comparative data, though recent changes in commercial solver participation have affected the benchmarking landscape.

Specialized Applications. Different algorithmic approaches show distinct advantages for specific application domains. Active-set methods with parametric capabilities [14] remain competitive for model predictive control with frequent problem updates. Interior-point methods continue to excel for large-scale optimization problems where sparse structure can be exploited effectively.

Table 2. Contemporary Comparison of Quadratic Programming Algorithms.

Criterion	Active-Set Methods	Interior-Point Methods	Operator-Splitting (ADMM/OSQP)	SDP Relaxations
Representative Work	Goldfarb & Idnani [9]	Mehrotra [12]	Stellato et al. [13]	Luo et al. [20]
Latest Developments	qpOASES parametric extensions [14]	Proximal stabilized methods [34]	GPU acceleration [32]	Differentiable SDP layers [30]
Theoretical Complexity	Finite: $\leq \binom{m+p}{n}$ iterations	$O\left(n^{1.5}\ln(\epsilon^{-1})\right)$ iterations	$O(1/k)$ convergence rate	Polynomial (SDP solver dependent)
Memory Scaling	$O(n^2)$ (working set dependent)	$O(n^2)$ to $O(n^3)$ (factorizations)	$O(n)$ (matrix-free iterations)	$O(n^2)$ (PSD matrices)
Convergence Properties	Finite (non-degenerate)	Enhanced with regularization [34]	Operator splitting advantages [13]	Problem-dependent
Warm-Starting	Excellent (working set preservation)	Improved with stabilization [34]	Natural ADMM structure [13]	Not commonly used
Current Software	qpOASES [14], commercial integration	Gurobi [15], CPLEX [16], MOSEK [17]	OSQP ecosystem [13], GPU [32]	MOSEK [17], CVXPY [28]
Standardized Benchmarking	Competitive on medium-scale problems [31]	Strong on sparse problems [35]	Excellent for MPC applications [31]	Provides global bounds
Hardware Optimization	Single-core	Multi-core sparse solvers	GPU [32]	Hardware support [30]
GPU Acceleration	Limited parallel structure	Moderate (sparse linear algebra)	Significant speedup [32]	Moderate (matrix operations)

5. Application Snapshots with Detailed Case Studies

5.1. Facility Location and Logistics Optimization

5.1.1. Problem Formulation: Multi-Facility Warehouse Location

Consider a logistics company that needs to optimally position warehouses to minimize transportation costs while satisfying operational constraints. This class of problems exemplifies the practical importance of constrained quadratic optimization in supply chain management.

Case Study 5.1 (Central Warehouse Location Problem)

A manufacturing company operates three production facilities and needs to establish a central warehouse to minimize total transportation costs. The problem incorporates both distance minimization and operational constraints.

Problem Data:

Factory A: Located at coordinates $(0,0)$

Factory B: Located at coordinates $(4,0)$

Factory C: Located at coordinates $(2,3)$

Constraint: Warehouse must be located on the main transportation route $y = 1$

Safety requirement: Minimum distance of 2 units from Factory B

Mathematical Formulation:

The objective function represents the sum of squared Euclidean distances (proportional to transportation costs):

$$f(x,y) = \sum_{i\in\{A,B,C\}} \| (x,y) - (x_i, y_i) \|^2$$

Expanding this expression:

$$\begin{aligned} f(x,y) &= (x-0)^2 + (y-0)^2 + (x-4)^2 + (y-0)^2 + (x-2)^2 + (y-3)^2 \\ &= x^2 + y^2 + x^2 - 8x + 16 + y^2 + x^2 - 4x + 4 + y^2 - 6y + 9 \\ &= 3x^2 - 12x + 3y^2 - 6y + 29 \end{aligned}$$

Constrained Optimization Problem:

$$\begin{aligned} \min_{x,y} \quad & f(x,y) = 3x^2 - 12x + 3y^2 - 6y + 29 \\ \text{subject to} \quad & g(x,y) = y - 1 = 0 \\ & h(x,y) = (x-4)^2 + (y-0)^2 - 4 \geq 0 \quad \text{(safety constraint)} \end{aligned}$$

Solution Using Lagrange Multipliers:

For the equality constraint $y = 1$, we form the Lagrangian:

$$\mathcal{L}(x,y,\lambda) = 3x^2 - 12x + 3y^2 - 6y + 29 + \lambda(y-1)$$

Step 1: Compute First-Order Conditions

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial x} &= 6x - 12 = 0 \quad \Rightarrow \quad x^* = 2 \\ \frac{\partial \mathcal{L}}{\partial y} &= 6y - 6 + \lambda = 0 \\ \frac{\partial \mathcal{L}}{\partial \lambda} &= y - 1 = 0 \quad \Rightarrow \quad y^* = 1 \end{aligned}$$

From the constraint $y^* = 1$ and the second equation:

$$\lambda^* = 6 - 6y^* = 6 - 6(1) = 0$$

Step 2: Verify Optimality

The Hessian of the objective function is:

$$H = \begin{pmatrix} 6 & 0 \\ 0 & 6 \end{pmatrix} \succ 0$$

Since the Hessian is positive definite, $(x^*, y^*) = (2,1)$ is indeed a minimum.

Step 3: Verify Constraints

Equality constraint: $y^* - 1 = 1 - 1 = 0$

Safety constraint: $(x^* - 4)^2 + (y^* - 0)^2 = (2-4)^2 + (1-0)^2 = 4 + 1 = 5 \geq 4$

Optimal Solution:

Warehouse location: $(2,1)$

Minimum transportation cost: $f(2,1) = 3(4) - 12(2) + 3(1) - 6(1) + 29 = 14$

Distance to Factory B: $\sqrt{5} \approx 2.236 > 2$ (safety satisfied)

5.1.2. Economic Interpretation and Sensitivity Analysis

The optimal location $(2,1)$ represents a compromise between minimizing total transportation costs and satisfying operational constraints. The warehouse is positioned to balance access to all three factories while maintaining required safety distances.

Sensitivity Analysis: - If the safety constraint were tightened to distance ≥ 3, the constraint would become active. The transportation route constraint $(y = 1)$ significantly influences the solution; without it, the unconstrained optimum would be at the centroid of the factory locations

5.2 Production Planning and Resource Allocation

5.2.1 Multi-Product Manufacturing Optimization

Production planning problems frequently involve quadratic costs due to economies and diseconomies of scale, making them natural applications for quadratic programming.

Case Study 5.2 (Dual-Product Manufacturing with Resource Constraints)

A manufacturing facility produces two products (A and B) with production costs that exhibit increasing marginal costs due to capacity constraints and resource limitations.

Problem Data:

Cost Structure: Quadratic production costs with interaction terms

Resource Constraints: Limited production capacity and raw materials

Production Requirements: Minimum production targets and quality standards

Mathematical Model:

The production cost function incorporates both individual product costs and interaction effects:

$$f(x_1, x_2) = \frac{1}{2}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}^{\top}\begin{pmatrix} 6 & 2 \\ 2 & 4 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \begin{pmatrix} 60 & 50 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

Expanding the matrix form:

$$f(x_1, x_2) = 3x_1^2 + 2x_1x_2 + 2x_2^2 - 60x_1 - 50x_2$$

Constraint Structure:

Production capacity: $x_1 + x_2 = 30$ (total daily output limit)

Raw material: $2x_1 + x_2 = 50$ (material consumption constraint)

Complete Optimization Problem:

$$\begin{aligned} \min_{x_1,x_2} \quad & f(x_1,x_2) = 3x_1^2 + 2x_1x_2 + 2x_2^2 - 60x_1 - 50x_2 \\ \text{subject to} \quad & x_1 + x_2 = 30 \\ & 2x_1 + x_2 = 50 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Solution Method: KKT Conditions

Step 1: Lagrangian Formulation

$$\mathcal{L}(x_1,x_2,\lambda_1,\lambda_2) = f(x_1,x_2) + \lambda_1(x_1 + x_2 - 30) + \lambda_2(2x_1 + x_2 - 50)$$

Step 2: First-Order Conditions

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial x_1} &= 6x_1 + 2x_2 - 60 + \lambda_1 + 2\lambda_2 = 0 \\ \frac{\partial \mathcal{L}}{\partial x_2} &= 2x_1 + 4x_2 - 50 + \lambda_1 + \lambda_2 = 0 \\ \frac{\partial \mathcal{L}}{\partial \lambda_1} &= x_1 + x_2 - 30 = 0 \\ \frac{\partial \mathcal{L}}{\partial \lambda_2} &= 2x_1 + x_2 - 50 = 0 \end{aligned}$$

Step 3: Solve the Linear System

From constraints (7.3) and (7.4):

$$\begin{aligned} x_1 + x_2 &= 30 \\ 2x_1 + x_2 &= 50 \end{aligned}$$

Subtracting the first from the second: $x_1 = 20$ Substituting back: $x_2 = 10$

Step 4: Determine Lagrange Multipliers

Substituting $x_1^* = 20, x_2^* = 10$ into equations (7.1) and (7.2):

$$\begin{aligned} 6(20) + 2(10) - 60 + \lambda_1 + 2\lambda_2 &= 0 \\ 2(20) + 4(10) - 50 + \lambda_1 + \lambda_2 &= 0 \end{aligned}$$

Simplifying:

$$\begin{aligned} 80 + \lambda_1 + 2\lambda_2 &= 0 \quad \Rightarrow \quad \lambda_1 + 2\lambda_2 = -80 \\ 30 + \lambda_1 + \lambda_2 &= 0 \quad \Rightarrow \quad \lambda_1 + \lambda_2 = -30 \end{aligned}$$

Solving: $\lambda_2 = -50, \lambda_1 = 20$

Step 5: Verification and Interpretation

Optimality Check: The Hessian matrix is

$$H = \begin{pmatrix} 6 & 2 \\ 2 & 4 \end{pmatrix}$$

Eigenvalues: $\lambda_{1,2} = 5 \pm \sqrt{5}$, both positive, confirming $H \succ 0$.

Economic Interpretation:

Optimal Production: 20 units of Product A, 10 units of Product B

Resource Utilization: Both constraints are active (binding)

Lagrange Multipliers:

$\lambda_1 = 20$: Shadow price of production capacity

$\lambda_2 = -50$: Shadow price of raw materials (negative indicates constraint relaxation would increase costs)

Step 6: Cost Analysis

Minimum total cost:

$$\begin{aligned} f(20,10) &= 3(20)^2 + 2(20)(10) + 2(10)^2 - 60(20) - 50(10) \\ &= 1200 + 400 + 200 - 1200 - 500 \\ &= 100 \text{ monetary units} \end{aligned}$$

Modern software packages provide high-level modeling interfaces for formulating these problems. CVXPY [28] and similar tools allow rapid prototyping and solution of QP problems without requiring detailed knowledge of the underlying algorithms.

5.3. Advanced Applications in Modern Contexts

5.3.1. Portfolio Optimization with Environmental, Social, and Governance (ESG) Constraints

Mathematical Formulation:

$$\begin{aligned} \min_{w} \quad & \frac{1}{2} w^\top \Sigma w - \gamma \mu^\top w + \beta \parallel Ew \parallel^2 \\ \text{subject to} \quad & \mathbf{1}^\top w = 1, \quad w \geq 0 \\ & \text{ESG}_{\min} \leq S^\top w \leq \text{ESG}_{\max} \end{aligned}$$

where Σ is the covariance matrix, μ represents expected returns, E captures ESG factors, and S represents ESG scores. For robust formulations that handle uncertainty in these parameters, see [29].

5.3.2. Model Predictive Control in Autonomous Systems

Formulation:

$$\begin{aligned} \min_{u_0,\dots,u_{N-1}} \quad & \sum_{k=0}^{N-1} \left(\parallel x_k - x_{\text{ref}} \parallel_Q^2 + \parallel u_k \parallel_R^2 \right) + \parallel x_N - x_{\text{ref}} \parallel_P^2 \\ \text{subject to} \quad & x_{k+1} = Ax_k + Bu_k, \quad k = 0,\dots,N-1 \\ & u_{\min} \leq u_k \leq u_{\max}, \quad k = 0,\dots,N-1 \end{aligned}$$

This formulation is fundamental in MPC applications [6]. Fast solution methods for real-time implementation are discussed in [7], while [14] provides specialized algorithms for the parametric QP problems that arise when the initial state x_0 varies.

6. Conclusion

This comprehensive survey has presented the mathematical foundations, algorithmic approaches, and diverse applications of quadratic programming and quadratically constrained quadratic programming. From the classical Lagrangian theory to modern operator-splitting methods, and from facility location to production planning applications, QP continues to serve as a cornerstone of optimization science.

The detailed case studies in facility location and production planning demonstrate the practical power of these mathematical tools in solving real-world problems. The warehouse location problem illustrated the systematic application of Lagrange multipliers to equality-constrained optimization, while the production planning example showcased the complete KKT framework for problems with multiple constraints.

The mathematical rigor established over decades continues to provide the foundation for innovations in autonomous systems, intelligent infrastructure, and optimization-driven technologies. Recent developments include the integration of QP layers in deep learning architectures [30], enabling end-to-end differentiable optimization within neural networks. As computational capabilities expand and new application domains emerge, quadratic programming will undoubtedly remain central to the optimization toolkit, bridging theoretical elegance with practical impact across science, engineering, and society.

References

[1] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, Mar. 1952, doi: 10.1111/j.1540-6261.1952.tb01525.x

[2] P. Wolfe, "The simplex method for quadratic programming," *Econometrica*, vol. 27, no. 3, pp. 382–398, 1959, doi: 10.2307/1909468.

[3] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Research Logistics Quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956, doi: 10.1002/nav.3800030109.

[4] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola, Eds. MIT Press, pp. 185–208, 1999, doi: 10.7551/mitpress/1130.003.0016.

[5] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[6] D. Q. Mayne and J. B. Rawlings, "Correction to "Constrained model predictive control: stability and optimality,'" *Automatica*, vol. 37, no. 3, p. 483, Mar. 2001, doi: 10.1016/s0005-1098(00)00173-4.

[7] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 267–278, 2010, doi: 10.1109/TCST.2009.2017934.

[8] P. E. Gill and W. Murray, "Numerically stable methods for quadratic programming," *Mathematical Programming*, vol. 14, no. 1, pp. 349–372, Dec. 1978, doi: 10.1007/bf01588976.

[9] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Mathematical Programming*, vol. 27, no. 1, pp. 1–33, Sep. 1983, doi: 10.1007/bf02591962.

[10] N. Karmarkar, "A new polynomial-time algorithm for linear programming," *Combinatorica*, vol. 4, no. 4, pp. 373-395, 1984.

[11] N. Megiddo, "Pathways to the optimal set in linear programming," in *Progress in Mathematical Programming*, pp. 131-158, Springer, 1989, doi: 10.1007/978-1-4613-9617-8_8.

[12] S. Mehrotra, "On the implementation of a primal-dual interior point method," *SIAM Journal on Optimization*, vol. 2, no. 4, pp. 575–601, Nov. 1992, doi: 10.1137/0802028.

[13] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, Feb. 2020, doi: 10.1007/s12532-020-00179-2.

[14] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpOASES: a parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014, doi: 10.1007/s12532-014-0071-1.

[15] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2023. [Online]. Available: https://www.gurobi.com

[16] IBM ILOG CPLEX, "CPLEX Optimization Studio," IBM Corporation, 2023. [Online]. Available: https://www.ibm.com/products/ilog-cplex-optimization-studio

[17] MOSEK ApS, "The MOSEK optimization toolbox for MATLAB manual," Version 10.0, 2023. [Online]. Available: https://docs.mosek.com

[18] N. Z. Shor, "Class of global minimum bounds of polynomial functions," *Cybernetics*, vol. 23, no. 6, pp. 731–734, 1987, doi: 10.1007/BF01070233.

[19] V. A. Yakubovich, "S-procedure in nonlinear control theory," *Vestnik Leningrad University*, vol. 1, pp. 62–77, 1971.

[20] Z. Q. Luo, W. K. Ma, A. M. C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 20–34, 2010, doi: 10.1109/MSP.2010.936019.

[21] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York: Springer, 2006.

[22] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed. Baltimore: Johns Hopkins University Press, 2013.

[23] S. J. Wright, *Primal-Dual Interior-Point Methods*. Philadelphia: SIAM, 1997, doi: 10.1137/1.9781611971453.

[24] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011, doi: 10.1561/2200000016.

[25] E. A. Yildirim and S. J. Wright, "Warm-start strategies in interior-point methods for linear programming," *SIAM Journal on Optimization*, vol. 12, no. 3, pp. 782–810, Jan. 2002, doi: 10.1137/s1052623400369235.

[26] S. Kim and M. Kojima, "Exact solutions of some nonconvex quadratic optimization problems via SDP and SOCP relaxations," *Computational Optimization and Applications*, vol. 26, no. 2, pp. 143–154, 2003, 10.1023/A:1025794313696.

[27] T. A. Davis, *Direct Methods for Sparse Linear Systems*. Philadelphia: SIAM, 2006, 10.1137/1.9780898718881.

[28] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, paper 83, pp. 1–5, 2016.

[29] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust Optimization*. Princeton: Princeton University Press, 2009, doi: 10.1515/9781400831050.

[30] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter, "Differentiable convex optimization layers," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[31] S. Caron, A. Zaki, P. Otta, D. Arnström, J. Carpentier, F. Yang, and P.-A. Leziart, "qpbenchmark: Benchmark for quadratic programming solvers available in Python," version 2.5.0, 2025, https://github.com/qpsolvers/qpbenchmark

[32] M. Schubiger, G. Banjac, and J. Lygeros, "GPU acceleration of ADMM for large-scale quadratic programming," *Journal of Parallel and Distributed Computing*, vol. 144, pp. 55–67, 2020. doi: 10.1016/j.jpdc.2020.04.008

[33] S. Cipolla and J. Gondzio, "Proximal stabilized interior point methods and low-frequency-update preconditioning techniques,"" *Journal of Optimization Theory and Applications*, vol. 197, no. 3, pp. 1061–1103, Apr. 2023, doi: 10.1007/s10957-023-02194-4.

[34] H. D. Mittelmann, "Decision tree for optimization software," Arizona State University, 2024. [Online]. Available: https://plato.asu.edu/guide.html

[35] A. Leulmi, R. Ziadi, C. Souli, M. A. Saleh, and A. Z. Almaymuni, "An interior point algorithm for quadratic programming based on a new step-length," *International Journal of Analysis and Applications*, vol. 22, article 233, 2024, 10.28924/2291-8639-22-2024-233.