ISE

Industrial and
Systems Engineering

Efficient Use of Quantum Linear System Algorithms in Interior Point Methods for Linear Optimization

Mohammadhossein Mohammadisiahroudi[1], Ramin Fakhimi[1], and Tamás Terlaky[1]

[1]Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA

ISE Technical Report 21T-005

Lehigh University

Efficient Use of Quantum Linear System Algorithms in Interior Point Methods for Linear Optimization

Mohammadhossein Mohammadisiahroudi, Ramin Fakhimi,
and Tamás Terlaky

May 2, 2022

Abstract Quantum computing has attracted significant interest in the optimization community because it potentially can solve classes of optimization problems faster than conventional supercomputers. Several researchers proposed quantum computing methods, especially Quantum Interior Point Methods (QIPMs), to solve convex optimization problems, such as Linear Optimization, Semidefinite Optimization, and Second-order Cone Optimization problems. Most of them have applied a Quantum Linear System Algorithm at each iteration to compute a Newton step. However, using quantum linear solvers in QIPMs comes with many challenges, such as having ill-conditioned systems and the considerable error of quantum solvers. This paper investigates how one can efficiently use quantum linear solvers in QIPMs. Accordingly, an Inexact Infeasible Quantum Interior Point Method is developed to solve linear optimization problems. We also discuss how can we get an exact solution by Iterative Refinement without excessive time of quantum solvers. Finally, computational results with QISKIT implementation of our QIPM using quantum simulators are analyzed.

Keywords Quantum Interior Point Method · Linear Optimization · Quantum Linear System Algorithm · Iterative Refinement

Mathematics Subject Classification (2000) 90C51 · 90C05

1 Introduction

Linear Optimization (LO) is defined as optimizing a linear function over a set of linear equality and inequality constraints. Several algorithms were developed to solve LO problems [12, 25].

Mohammadhossein Mohammadisiahroudi, Corresponding author
Lehigh University
Bethlehem, PA 18015, USA
mom219@Lehigh.edu

Ramin Fakhimi
Lehigh University
Bethlehem, PA 18015, USA
fakhimi@Lehigh.edu

Tamás Terlaky
Lehigh University
Bethlehem, PA 18015, USA
terlaky@Lehigh.edu

Karmarkar [21] developed the foundations of polynomial time and practically efficient Interior Point Methods (IPMs) for solving LO problems. Since Karmarkar's publication, a large class of theoretically and practically efficient IPMs were developed, see, e.g., [35, 32, 38]. Contrary to the Simplex method, a feasible IPM reaches an optimal solution by traversing through the interior of the feasible region [32].

Contemporary IPMs reach an optimal solution by starting from an interior point and following the central path [32]. The most efficient IPMs are primal-dual methods, meaning they strive to satisfy the optimality conditions while maintaining strict primal and dual feasibility. It should be noted that basic IPMs need an initial feasible interior point. Some current commercial solvers apply Feasible IPMs (F-IPMs) based on the self-dual embedding formulation of LO problems, e.g., MOSEK, while Infeasible Interior Point Methods (I-IPMs) can start with an infeasible but positive solution. Theoretical analysis shows the best iteration complexity of F-IPMs is $\mathcal{O}(\sqrt{n}L)$, where n is the number of variables, and L is the binary length of the input data. On the other hand, the best iteration complexity of I-IPMs is $\mathcal{O}(nL)$. In practice, the performance of both feasible and infeasible IPM are similar [38].

A linear equation system is solved at each iteration of IPMs to calculate a Newton direction. There are three choices for the linear equation system: (i) Full Newton System, (ii) Augmented System, and (iii) Normal Equation System (NES). In classical computers, a general approach is applying Cholesky Factorization to solve the NES, because it has a symmetric positive definite coefficient matrix. A partial update technique improved the complexity of solving the system in each iteration. This approach leads to the best total complexity of $\mathcal{O}(n^3L)$ arithmetic operations for solving LO problems. However, this is not always efficient in practice [32]. Since Cholesky Factorization requires $\mathcal{O}(n^3)$ arithmetic operations for large dense matrices, several researchers studied inexact solution methodologies for solving Newton systems. Bellavia [3] proved the convergence of Inexact I-IPM (II-IPM) for general convex optimization problems. Mizuno and his colleagues studied the convergence of II-IPMs [29, 16]. Korzak [27] and Baryamureeba and Steihaug [2] proved the convergence of II-IPM proposed by Kojima et al. [26]. Korzak [27] also showed that the total time complexity of his algorithm is polynomial. Al-Jeiroudi and Gondzio [20], and Monteiro and O'Neal [31] investigated the use of Preconditioned Conjugate Gradient (PCG) methods in II-IPMs. Zhou and Toh [41] also proved the convergence of II-IPM for SDO problems. The best iteration complexity of II-IPMs is $\mathcal{O}(n^2L)$ which is $\Theta(n^{1.5})$ weaker than the best iteration complexity of exact IPMs.

Quantum computers have recently emerged as a powerful alternative to classic computers [11]. Starting from Deutsch's Problem, a series of problems and algorithms have demonstrated theoretically exponential speedup compared to their classical counterparts [14, 15, 34]. One of the promising quantum algorithms is the HHL method [19] to solve linear system problems. The HHL method showed quantum advantage with respect to the dimension compared to classical linear equation systems. However, this method has unfavorable dependence on the condition number, error, and the sparsity of the matrix. Several researchers attempted to improve the performance of Quantum Linear System Algorithms (QLSAs), [37, 11]. This paper explores an efficient use of the QLSAs for solving the Newton system at each iteration of IPMs.

To investigate quantum speedup for continuous optimization, Brandão and his colleagues proposed a non-IPM quantum algorithm based on the multiplicative weight method to solve SDO problems [5, 6]. A few studies were proposing QIPMs. Kerenidis and his colleagues presented a series of papers on QIPMs for solving LO, Semidefinite Optimization (SDO) [22], and Second-order Cone Optimization (SOCO) problems [24]. Kerenidis and Prakash [22] claimed that their algorithm has $\mathcal{O}(\frac{n^2}{\epsilon^2}\kappa^3\log(1/\zeta))$ complexity for LO problems where ζ is the final optimality gap, ϵ is the precision of the QLSA. Further, κ is an upper bound for the condition number of the Newton systems at each iteration. This result indicates polynomial improvement with respect to the number of variables n over the best classical solvers, while it suffers from high dependence on the condition number and the precision of the QLSA. They used block encoding to construct the Newton system and Quantum Tomography Algorithm (QTA) to extract the classical solution. Casares and Martin-Delgado [8] also provided a hybrid predictor-corrector QIPM scheme to solve LO problems using the well-known predictor-corrector method proposed by Ye et al. [40]. In each step, it uses the QLSA proposed by Chakraborty et al. [9] to solve the Newton systems. The complexity of their method is $\mathcal{O}(L\sqrt{n}(n+m)\overline{\|M\|}_F\kappa^2\epsilon^{-2})$, where m is the number of constraints, and $\overline{\|M\|}_F$ is an upper bound to the Frobenius norm of the coefficient matrix in the Newton system. The authors claimed a quantum speedup with respect to the dimension n compared to classical algorithms. However, they overlooked some elements in their time complexity. Thus, the actual complexity has higher dependence on n and the required precision.

In both mentioned QIPMs, an exact F-IPM framework was used, regardless of the inherent inexactness of QLSAs. So, the convergence of these QIPMs is questionable, and the proposed time complexities are not attainable, since Inexact IPMs have higher iteration complexity. A major challenge is that QLSAs' time complexities depend on the condition number of the linear system and the required precision of QLSAs. In IPMs, the condition number of the Newton system typically goes to infinity as the algorithm approaches an optimal solution [32]. It is also worthy to mention that ϵ, the required precision for solution of the Newton system, needs to be significantly smaller than the final precision ζ. Consequently, these complexity bounds are not polynomial in the classical sense. This paper explores an efficient use of the QLSAs for solving the Newton system at each iteration of QIPMs. We propose an II-QIPM to find an exact solution. We also employ an iterative refinement scheme to avoid exponential complexity in finding an exact optimal solution.

This paper is structured as follows. Section 2 discusses the performance of existing QLSAs. Section 3 introduces the LO problem and its characteristics. In Section 4, we present an II-QIPM to solve LO problems, in which a QLSA is used for solving the NES. We also prove the convergence of the proposed II-QIPM. In Section 5, we employ an Iterative Refinement scheme to find an exact optimal solution of a LO problem without excessive time of QLSAs. Section 6 discusses how the results change using different Newton systems. Section 7 presents the first implementation of the proposed II-QIPM with Iterative Refinement (IR-II-QIPM) and evaluates the algorithms through computational experiments.

2 Quantum Linear Algebra

This section reviews the use of Quantum algorithms to solve Linear System Problems (LSP).

Definition 2.1 (LSP) The Linear System Problem: Find a vector $z \in \mathbb{R}^p$ such that it satisfies equation $Mz = \sigma$ with coefficient matrix $M \in \mathbb{R}^{p\times p}$ and right-hand side (RHS) vector $\sigma \in \mathbb{R}^p$.

In the rest of the paper, $\|M\| = \|M\|_2$ is the 2-norm of matrix M, and $\|M\|_F$ is the Frobenius norm of M. Also, we sometimes use $\tilde{\mathcal{O}}$ which suppresses the polylogaritimic factors in the "Big-O" notation. It should be mentioned that the complexity of classical algorithm means the number of arithmetic operations and the complexity of quantum algorithm is the number of quantum gates.

The LSP can have either one, many, or no solutions. A basic approach for solving an LSP is Gaussian elimination, or LU factorization, with $\mathcal{O}(p^3)$ arithmetic operations. If M is a square symmetric positive semidefinite (PSD) matrix, we can also apply Cholesky factorization with $\mathcal{O}(p^3)$ arithmetic operations. The best complexity for an iterative algorithm with respect to p is $\mathcal{O}(pd\kappa\log(1/\epsilon))$ arithmetic operations for the Conjugate Gradient method solving systems with

PSD matrices, where d is the maximum number of non-zero elements in any row or column of M, κ is the condition number of M, and ϵ is the error allowed.

Before discussing QLSAs, we should mention that the $|z\rangle$ notation represent the quantum state corresponding to the unit classical vector z. We denote the basis state $|i\rangle$, which is a column vector with dimension p, one in coordinate i and zero in other coordinates [11]. QLSAs have different approaches with different assignments, while all of them are solving Quantum Linear System Problems (QLSPs) defined as follows.

Definition 2.2 (QLSP) Let $M \in \mathbb{C}^{p\times p}$ be a Hermitian matrix with $\|M\| = 1$, and let $\sigma \in \mathbb{C}^p$. We seek to find a quantum state $|z\rangle$ such that it satisfies the linear system $M|z\rangle = |\sigma\rangle$, where

$$|\sigma\rangle = \frac{\sum_{i=1}^{p} \sigma_i |i\rangle}{\| \sum_{i=1}^{p} \sigma_i |i\rangle \|} \quad \text{and} \quad |z\rangle = \frac{\sum_{i=1}^{p} z_i |i\rangle}{\| \sum_{i=1}^{p} z_i |i\rangle \|}.$$

Based on Definition 2.1 and Definition 2.2, QLSP is a different form of the LSP. It should be noted that at each iteration of QIPMs, instead of an LSP, we need to use a QLSA to solve a QLSP. Thus, we need to translate LSP to QLSP, solve the QLSP by QLSA and extract the solution by a QTA. Some papers incorrectly used the result of QLSAs for solving LSP without considering the cost of QTA, encoding, and scaling. Here, we analyze the details and costs of the process of translating LSPs to QLSPs, encoding in quantum setting, solving them with a QLSA, and extracting classical solutions with a QTA as follows.

(i) Model of Computation: the first important step is determining how to encode the input data in quantum setting. There are two major input models. One is the sparse-access model which used in the HHL algorithm [19] and then in other QLSAs [11, 7]. This is a quantum version of classical sparse matrix computation, and we assume access to unitaries that calculate the index of the lth non-zero element of the kth row of a matrix M when given (k, l) as input. A different input model, now known as the quantum operator input model, is proposed in Low and Chuang [28], which is based on the idea of block-encoded matrices. In this input model, one has access to unitaries that store the coefficient matrix:

$$U = \begin{pmatrix} M/\alpha & \cdot \\ \cdot & \cdot \end{pmatrix},$$

where $\alpha \geq \|M\|$ is a normalization factor chosen to ensure that U has norm at most 1. Chakraborty et al. [9] showed that this quantum operator input model is more efficient than the sparse-access model and oracles to encode input data using block-encoding has favorable complexity compared to the sparse-access model. On the other hand, most of block-encoding approaches use Quantum Random Access Memory (QRAM). However, the sparse-access model can be implemented in the standard gate-based quantum circuit model. Despite efficient encoding procedures, the quantum operator input model can not be implemented with current quantum computers since there is no physical implementation of QRAM. In our analysis, we assume that the data is stored in QRAM, and we use the quantum operator model by Chakraborty et al. [9] for the QLSA, which enjoys the best complexity to date. Using the QRAM structure, Kerenidis and Prakash [23] showed that one can implement ϵ-approximate block-encoding of M with $\mathcal{O}(\text{polylog}(\frac{p}{\epsilon}))$ complexity. Further, Childs [10] proved that given M in the sparse-access input model, there is an ϵ-approximate block-encoding of M that can be implemented in complexity $\mathcal{O}(\text{polylog}(\frac{p}{\epsilon}))$. Thus, our results also apply to the sparse-access input model if we have the data in that form. Using these results and assuming access to QRAM, Chakraborty et al. [9] proposed a QLSA, in which they construct state $|\sigma\rangle$, build and implement a block-encoding of matrix M with $\mathcal{O}(\text{polylog}(\frac{p}{\epsilon}))$ complexity.

(ii) Translating LSP to QLSP: Based on the definition of QLSP, the coefficient matrix of the system must be Hermitian. If M is not Hermitian, one can construct $\bar{M}\bar{z} = \bar{\sigma}$, where

$$\bar{M} = \begin{bmatrix} 0 & M \\ M^\dagger & 0 \end{bmatrix}, \quad \bar{\sigma} = \begin{pmatrix} \sigma \\ 0 \end{pmatrix},$$

and find the vector $\bar{z} = \begin{pmatrix} 0 \\ z \end{pmatrix}$, where $M^\dagger$ denotes the conjugate transpose of M. The size of the problem increases from p to $2p$. QLSP assumes $\|M\| = 1$. In the structure of block encoding, we address this normalization but for sparse encoding we need to normalize the system $\bar{M}\bar{z} = \bar{\sigma}$ where

$$\bar{M} = \frac{M}{\|M\|}, \quad \bar{\sigma} = \frac{\sigma}{\|M\|}, \quad \text{and} \quad \bar{z} = z.$$

We also have a similar scaling in the definitions of states $|z\rangle$ and $|\sigma\rangle$. This scaling affects the target precision. Let $|\tilde{z}\rangle$ be an inexact solution of the QLSP such that $\| \, |\tilde{z}\rangle - |z\rangle \, \| \leq \epsilon_{QLSP}$. To extract the solution of the LSP ($\tilde{z}$), we have $\tilde{z} = \|\sigma\| \, |\tilde{z}\rangle$ and the error for the solution of the LSP will be

$$\|\tilde{z} - z\| = \|\sigma\| \| \, |\tilde{z}\rangle - |z\rangle \, \| \leq \|\sigma\| \epsilon_{QLSP}.$$

Thus, scaling changes the error of a solution. If we want to have error bound ϵ_{LSP} for the solution of the LSP, then we must set the target error of the QLSP as $\epsilon_{QLSP} = \epsilon_{LSP}/\|\sigma\|$.

(iii) QLSA: After preprocessing and encoding, we can apply the QLSA to solve the QLSP. The HHL algorithm [19] solves QLSPs with $\tilde{\mathcal{O}}(\log(p)\frac{d^2\kappa^2}{\epsilon_{QLSP}})$ complexity. Several researchers attempted to improve the performance of the HHL algorithm. As the first attempt, Amplitude Amplification decreases the dependence on κ^2 to κ [11]. Wossnig et al. [37] proposed a QLSA with $\tilde{\mathcal{O}}(\text{polylog}(p)\|M\|_F \frac{\kappa}{\epsilon_{QLSP}})$ complexity by using the Quantum Singular Value Estimation. In another direction, Childs et al. [11] developed two QLSAs with exponentially better dependence on error with $\tilde{\mathcal{O}}(\text{polylog}(\frac{p\kappa}{\epsilon_{QLSP}})d\kappa)$ complexity. They proposed two approaches using Fourier and Chebyshev series representations. The best QLSA with respect to complexity uses block encoding and QRAM with $\tilde{\mathcal{O}}(\text{polylog}(\frac{p}{\epsilon_{QLSP}})\|M\|_F\kappa)$ complexity [9]. The details of these methods are out of the scope of this paper. For further details, see [13].

(iv) QTA: QLSAs provide a quantum state proportional to the solution. We cannot extract the classical solution by a single measurement. We need Quantum Tomography Algorithms (QTAs) to extract the classical solution. There are several papers improving QTAs, see e.g., [24]. We used the best QTA by [36], with complexity $\mathcal{O}(\frac{p}{\epsilon_{QTA}})$. Since the error is additive, we may choose

$$\epsilon_{QTA} = \epsilon_{QLSA} = \frac{\epsilon_{LSP}}{2\|\sigma\|}.$$

Table 1 presents the complexity of different classical and quantum algorithms for solving an LSP. Here, the complexity of a QTA is considered in the complexity of QLSAs. As shown in Table 1, although QLSAs have better dependence on dimension p compared to classical solvers, the complexity of solving LSP using QLSA+QTA will have similar dependence on p, but better dependence on sparsity d, compared to Conjugate Gradient (CG). QLSAs have better dependence on dimension p compared to factorization and elimination techniques. Generally, QLSA has worse dependence on κ, $\frac{1}{\epsilon_{LSP}}$, $\|M\|$, and $\|\sigma\|$. The following section will discuss how we can deal with error and condition number when we use QLSA in IPMs.

Table 1: Complexity of solving an LSP.

Algorithm	Complexity
Factorization methods (e.g. Cholesky)	$\mathcal{O}(p^3)$
Conjugate Gradient	$\mathcal{O}\big(pd\sqrt{\kappa}\log(\frac{1}{\epsilon})\big)$
HHL [19] + QTA [36]	$\mathcal{O}\Big(p\frac{d^2\kappa^2\\|\sigma\\|^2}{\\|M\\|^2\epsilon^2}\text{polylog}(\frac{p\kappa}{\epsilon})\Big)$
QLSA [11] + QTA [36]	$\mathcal{O}\Big(p\frac{d\kappa\\|\sigma\\|}{\\|M\\|\epsilon}\text{polylog}(\frac{p\kappa}{\epsilon})\Big)$
QLSA [9] + QTA [36]	$\mathcal{O}\Big(p\frac{\kappa\\|\sigma\\|\\|M\\|_F}{\epsilon}\text{polylog}(\frac{p}{\epsilon})\Big)$

3 The Linear Optimization Problem

Here we consider the standard form of Linear Optimization (LO) problem as follows:

$$\begin{array}{llll} & \min\ c^Tx & & \max\ b^Ty \\ \text{(P)} & \text{s.t. } Ax = b, & \text{(D)} & \text{s.t. } A^Ty+s = c, \\ & \qquad x \geq 0, & & \qquad\qquad s \geq 0, \end{array} \tag{1}$$

where $A : m \times n$ matrix with $\text{rank}(A) = m$, vectors $y, b \in \mathbb{R}^m$, and $x, s, c \in \mathbb{R}^n$. Problem (P) is called the primal problem and (D) called the dual problem. Due to the Strong Duality Theorem [32], all optimal solutions, if exist, belong to the set $\mathcal{PD}^*$, which is defined as

$$\mathcal{PD}^* = \Big\{(x, y, s) \in \mathbb{R}^{n+m+n} :\ Ax = b,\ A^Ty + s = c,\ x^Ts = 0,\ (x, s) \geq 0\Big\}.$$

Now, we can define the optimal partition of the LO problem as

$$\begin{aligned} B &= \{j \in \{1, \ldots, n\} : x^*_j > 0 \text{ for some } (x^*, y^*, s^*) \in \mathcal{PD}^*\}, \\ N &= \{j \in \{1, \ldots, n\} : s^*_j > 0 \text{ for some } (x^*, y^*, s^*) \in \mathcal{PD}^*\}, \end{aligned}$$

where $B \cup N = \{1, \ldots, n\}$, and $B \cap N = \emptyset$ [32].

Assumption 1. *From now on, without loss of generality [32], we assume that the Interior Point Condition (IPC) holds, i.e., there exist a solution (x, y, s) such that*

$$Ax = b,\ x > 0,\ A^Ty + s = c,\ \textit{and}\ s > 0.$$

The IPC warranties [32] that the optimal set $\mathcal{PD}^*$ is bounded, so there exists $\omega \geq 1$ such that

$$\omega \geq \max\{\|(x^*, s^*)\|_\infty : (x^*, s^*) \in \mathcal{PD}^*\}.$$

The central path is the curve defined by

$$\mathcal{CP} = \Big\{(x, y, s) \in \mathbb{R}^{n+m+n} :\ Ax = b,\ A^Ty + s = c,\ x_is_i = \mu \text{ for } i \in \{1, \ldots, n\},\ x, s, \mu > 0\Big\}.$$

By the IPC, the central path is well-defined, and an interior feasible solution (x, y, s) exists for all $\mu > 0$. Now, for any $0 < \gamma_1 < 1$, we define an infeasible neighborhood of the central path as the following definition in [38].

$$\mathcal{N}(\gamma_1, \gamma_2) = \{(x, y, s) \in \mathbb{R}^{n+m+n} : (x, s) \geq 0,\ x_is_i \geq \gamma_1\frac{x^Ts}{n} \text{ for } i \in \{1, \ldots, n\}, \|(R_P, R_D)\| \leq \gamma_2\mu\},$$

where $R_P = b - Ax$, and $R_D = c - A^T y - s$. Assuming that the input data is integer, we denote the binary length of the input data by

$$L = mn + m + n + \sum_{i,j} \lceil \log(|a_{ij}| + 1) \rceil + \sum_i \lceil \log(|c_i| + 1) \rceil + \sum_j \lceil \log(|b_j| + 1) \rceil.$$

The following lemma is a classical result first proved by Khachiyan [25].

Lemma 3.1 *Let $(x^*, y^*, s^*) \in \mathcal{PD}^*$ be a basic solution. If $x_i^* > 0$, then we have $x_i^* \geq 2^{-L}$. If $s_i^* > 0$, then we have $s_i^* \geq 2^{-L}$.*

Theorem 3.1 (Chapter 3 in [38]) *An exact optimal solution can be attained by a strongly polynomial rounding procedure when $(x, y, s) \in \mathcal{N}(\gamma_1, \gamma_2)$ and $\mu \leq 2^{-2L}$.*

If the IPC holds, then the optimal set $\mathcal{PD}^*$ is bounded, and we can find the upper bound for all the coordinates of all optimal solutions as described in the following lemma.

Lemma 3.2 (Chapter 5 in [39]) *Assuming the IPC, then for any $(x^*, y^*, s^*) \in \mathcal{PD}^*$, $\max_i\{x_i^*\} \leq 2^L$, and $\max_i\{s_i^*\} \leq 2^L$.*

For theoretical purpose, we can use $\omega = 2^L$ from Lemma 3.2, but in practice for concrete LO problems we may find a smaller bound.

4 An Inexact Infeasible Quantum IPM

To speed up IPMs, we use QLSAs to solve the Newton system at each iteration of IPMs. As discussed in Section 2, QLSAs inherently produce inexact solutions. Thus, one approach to use QLSA efficiently is to develop an Inexact Infeasible QIPM (II-QIPM). In this paper, we utilize the KMM method proposed by Kojima et al. [26] with the inexact Newton steps calculated by a QLSA. Given $(x^k, y^k, s^k) \in \mathcal{N}(\gamma_1, \gamma_2)$, let $\mu^k = \frac{(x^k)^T s^k}{n}$ and $0 < \beta_1 < 1$ be the centering parameter, then the Newton system is defined as

$$\begin{aligned} A\Delta x^k &= b - Ax^k, \\ A^T \Delta y^k + \Delta s^k &= c - A^T y^k - s^k, \\ X^k \Delta s^k + S^k \Delta x^k &= \beta_1 \mu^k e - X^k s^k, \end{aligned} \tag{2}$$

where e is all one vector with appropriate dimension, $X^k = \operatorname{diag}(x^k)$, and $S^k = \operatorname{diag}(s^k)$. Instead of solving the full Newton system, we may solve the Augmented system or the Normal Equation System (NES). In the following, we use the NES and discuss the effect of different systems on the results later in Section 6. From the Newton system (2), the NES is formulated as

$$M^k \Delta y^k = \sigma^k, \tag{NES}$$

where

$$\begin{aligned} D^k &= (X^k)^{1/2}(S^k)^{-1/2}, \\ M^k &= A(D^k)^2 A^T, \\ \sigma^k &= AX^k(S^k)^{-1}c - AX^k(S^k)^{-1}A^T y^k - \beta_1 \mu^k A(S^k)^{-1}e + b - Ax^k \\ &= b - \beta_1 \mu^k A(S^k)^{-1}e + AX^k(S^k)^{-1}(c - A^T y^k - s^k). \end{aligned}$$

As we can see, the NES has a smaller size, m, than the full Newton system. Further, the matrix of the NES is symmetric, and positive definite, thus Hermitian. Consequently, QLSAs can solve the NES efficiently. By its nature, a QLSA generates an inexact solution $\widetilde{\Delta y}^k$ with error bound $\|\Delta y^k - \widetilde{\Delta y}^k\| \leq \epsilon^k$. This error leads to residual r^k as

$$M^k \widetilde{\Delta y}^k = \sigma^k + r^k,$$

where $r^k = M^k(\widetilde{\Delta y}^k - \Delta y^k)$. After finding $\widetilde{\Delta y}^k$ inexactly by solving the NES using QLSA, we compute the inexact $\widetilde{\Delta x}^k$ and $\widetilde{\Delta s}^k$ classically as

$$\begin{aligned} \widetilde{\Delta s}^k &= c - A^T y^k - s^k - A^T \widetilde{\Delta y}^k, \\ \widetilde{\Delta x}^k &= \beta_1 \mu^k (S^k)^{-1} e - x^k - (S^k)^{-1} X^k \widetilde{\Delta s}^k. \end{aligned} \tag{3}$$

As $\widetilde{\Delta s}^k$ and $\widetilde{\Delta x}^k$ are directly calculated by equations (3), one can verify that $(\widetilde{\Delta x}^k, \widetilde{\Delta s}^k, \widetilde{\Delta y}^k)$ satisfies

$$\begin{aligned} A\widetilde{\Delta x}^k &= b - Ax^k + r^k, \\ A^T \widetilde{\Delta y}^k + \widetilde{\Delta s}^k &= c - A^T y^k - s^k, \\ X^k \widetilde{\Delta s}^k + S^k \widetilde{\Delta x}^k &= \beta_1 \mu^k e - X^k s^k. \end{aligned} \tag{4}$$

To have a II-IPM using (NES) with iteration complexity $\mathcal{O}(n^2 L)$, the residual norm must decrease at least $\mathcal{O}(\lambda_{\min}(A)\sqrt{n}\log n)$ time faster than $(x^k)^T s^k$ where $\lambda_{\min}(A)$ is the smallest singular value of A [41]. We can have wider residual bound but with higher iteration complexity of II-IPM [4]. In the literature of preconditioning the NES, some papers modified the equations (3) and the (NES) to transfer the residual from the first equation of (4) to its last equation. By these changes, we can get much better bounds [20, 31]. Since tight residual bound leads to high complexity of QLSA+QTA, in this paper, we use a modification of the NES which leads to $\mathcal{O}(n^2 L)$ iteration complexity of II-QIPM, where the residual is decreasing with the rate of $\mathcal{O}(\sqrt{\mu^k})$.

Since A has full row rank, one can choose an arbitrary basis $\hat{B}$, and calculate $A_{\hat{B}}^{-1}$, $\hat{A} = A_{\hat{B}}^{-1} A$, and $\hat{b} = A_{\hat{B}}^{-1} b$. This calculation needs $\mathcal{O}(m^2 n)$ arithmetic operations and happens just one time before the iterations of IPM. The cost of this preprocessing is dominated by the cost of II-QIPM, but it can be reduced by using the structure of A. For example, if the problem is in the canonical form, there is no need for this preprocessing. In the rest of this paper, all methodology is applied to the preprocessed problem with input data $(\hat{A}, \hat{b}, c)$. Now, we can modify the (NES) to

$$\hat{M}^k z^k = \hat{\sigma}^k \tag{MNES}$$

where

$$\begin{aligned} \hat{M}^k &= (D_{\hat{B}}^k)^{-1} A_{\hat{B}}^{-1} M^k ((D_{\hat{B}}^k)^{-1} A_{\hat{B}}^{-1})^T = (D_{\hat{B}}^k)^{-1} \hat{A} (D^k)^2 ((D_{\hat{B}}^k)^{-1} \hat{A})^T, \\ \hat{\sigma}^k &= (D_{\hat{B}}^k)^{-1} A_{\hat{B}}^{-1} \sigma^k = (D_{\hat{B}}^k)^{-1} \hat{b} - \beta_1 \mu^k (D_{\hat{B}}^k)^{-1} \hat{A} (S^k)^{-1} e + (D_{\hat{B}}^k)^{-1} \hat{A} (D^k)^2 (c - A^T y^k - s^k). \end{aligned}$$

We use the following procedure to find the Newton direction by solving (MNES) inexactly with QLSA+QTA.

Step 1. Find $\tilde{z}^k$ such that $\hat{M}^k \tilde{z}^k = \hat{\sigma}^k + \hat{r}^k$ and $\|\hat{r}^k\| \leq \eta \frac{\sqrt{\mu^k}}{\sqrt{n}}$.
Step 2. Calculate $\widetilde{\Delta y}^k = ((D_{\hat{B}}^k)^{-1} A_{\hat{B}}^{-1})^T \tilde{z}^k$.
Step 3. Calculate $v^k = (v_{\hat{B}}^k, v_{\hat{N}}^k) = (D_{\hat{B}}^k \hat{r}^k, 0)$.
Step 4. Calculate $\widetilde{\Delta s}^k = c - A^T y^k - s^k - A^T \widetilde{\Delta y}^k$.

Step 5. Calculate $\widetilde{\Delta x}^k = \beta_1 \mu^k (S^k)^{-1} e - x^k - (D^k)^2 \widetilde{\Delta s}^k - v^k$.

The following Lemma shows how the inexact solution of (MNES) lead to residual only in the last equation of the Newton system.

Lemma 4.1 *For the Newton direction* $(\widetilde{\Delta x}^k, \widetilde{\Delta y}^k, \widetilde{\Delta s}^k)$, *we have*

$$\begin{aligned} A\widetilde{\Delta x}^k &= b - Ax^k, \\ A^T\widetilde{\Delta y}^k + \widetilde{\Delta s}^k &= c - A^T y^k - s^k, \\ X^k\widetilde{\Delta s}^k + S^k\widetilde{\Delta x}^k &= \beta_1\mu^k e - X^k s^k - S^k v^k. \end{aligned} \tag{5}$$

Proof. For the Newton direction $(\widetilde{\Delta x}^k, \widetilde{\Delta y}^k, \widetilde{\Delta s}^k)$, one can verify that

$$\begin{aligned} \hat{M}^k \tilde{z}^k &= \hat{\sigma}^k + \hat{r}^k \\ M^k \widetilde{\Delta y}^k &= \sigma^k + A_{\hat{B}} D^k_{\hat{B}} \hat{r}^k. \end{aligned}$$

For the first equation of (5), we can write

$$\begin{aligned} A\widetilde{\Delta x}^k =& A(\beta_1\mu^k(S^k)^{-1}e - x^k - (S^k)^{-1}X^k\widetilde{\Delta s}^k - v^k) \\ =& A(\beta_1\mu^k(S^k)^{-1}e - x^k - (S^k)^{-1}X^k(c - s^k - A^T y^k - A^T\widetilde{\Delta y}^k) - v^k) \\ =& \beta_1\mu^k A(S^k)^{-1}e - Ax^k - A(S^k)^{-1}X^k c + A(S^k)^{-1}X^k s^k + A(S^k)^{-1}X^k A^T y^k \\ & + A(S^k)^{-1}X^k A^T\widetilde{\Delta y}^k - Av^k \\ =& \beta_1\mu^k A(S^k)^{-1}e - A(S^k)^{-1}X^k c + A(S^k)^{-1}X^k A^T y^k + \sigma^k + A_{\hat{B}} D^k_{\hat{B}} \hat{r}^k - A_{\hat{B}} D^k_{\hat{B}} \hat{r}^k \\ =& b - Ax^k. \end{aligned}$$

The second and third equations of (5) are obtained by the Steps 4 and 5. □

To have a convergent IPM, we need $\|S^k v^k\|_\infty \le \eta\mu^k$, where $0 \le \eta < 1$ is an enforcing parameter.

Lemma 4.2 *For the Newton direction* $(\widetilde{\Delta x}^k, \widetilde{\Delta y}^k, \widetilde{\Delta s}^k)$, *if the residual* $\|\hat{r}^k\| \le \eta\frac{\sqrt{\mu^k}}{\sqrt{n}}$, *then* $\|S^k v^k\|_\infty \le \eta\mu^k$.

Proof. We have

$$\|S^k v^k\|_\infty = \|S^k_{\hat{B}} v^k_{\hat{B}}\|_\infty = \|S^k_{\hat{B}} D^k_{\hat{B}} \hat{r}^k\|_\infty = \|(S^k_{\hat{B}})^{1/2}(X^k_{\hat{B}})^{1/2}\|_\infty \|\hat{r}^k\|_\infty \le \sqrt{n\mu^k}\|\hat{r}^k\| \le \eta\mu^k.$$

□

In the following, we show that by satisfying $\|\hat{r}^k\| \le \eta\frac{\sqrt{\mu^k}}{\sqrt{n}}$, then the iterations of the II-QIPM remain in the $\mathcal{N}(\gamma_1, \gamma_2)$ neighborhood of the central path. The following theorem presents the complexity of solving the (MNES) system by utilizing the QLSA of Chakraborty et al. [9]. We can also use other QLSAs discussed in Section 2, leading to different complexity bounds.

Lemma 4.3 *The QLSA by [9] and the QTA by [36] can build the* (MNES) *system, and produce a solution* $\tilde{z}^k$ *for the* (MNES) *system satisfying* $\|\hat{r}^k\| \le \eta\frac{\sqrt{\mu^k}}{\sqrt{n}}$ *with* $\tilde{\mathcal{O}}(mn^{1.5}\frac{\kappa^k_E\|\hat{\sigma}^k\|\|E^k\|_F}{\sqrt{\mu^k}})$ *complexity, where* $E^k = (D^k_{\hat{B}})^{-1}\hat{A}D^k$, *and* κ^k_E *is the condition number of* E^k.

Proof. Building the (MNES) system in classical computer needs some matrix multiplications, which costs $\mathcal{O}(m^2n)$ arithmetic operations. We can write (MNES) as $E^k(E^k)^T\tilde{z}^k = \hat{\sigma}^k$. As we can see, calculating E^k and $\hat{\sigma}^k$ needs just $\mathcal{O}(mn)$ arithmetic operations. Chakraborty et al. [9] proposed an efficient way to build and solve a linear system in the form $E^k(E^k)^T\tilde{z}^k = \hat{\sigma}^k$, with $\tilde{\mathcal{O}}(\text{polylog}(\frac{n}{\epsilon^k_{QLSA}})\kappa^k_E\|E^k\|_F)$ complexity. Also, we need to find the target precision for QLSA and QTA such that $\|\hat{r}^k\| \leq \eta\frac{\sqrt{\mu^k}}{\sqrt{n}}$ is satisfied. For LSP, we have

$$\|\hat{r}^k\| = \|\hat{M}^k z^k - \hat{M}^k\tilde{z}^k\| \leq \|\hat{M}^k\| \cdot \|z^k - \tilde{z}^k\| \leq \|\hat{M}^k\|\epsilon^k_{LSP}.$$

Thus, to have $\|\hat{r}^k\| \leq \eta\frac{\sqrt{\mu^k}}{\sqrt{n}}$, it is sufficient to require $\epsilon^k_{LSP} \leq \eta\frac{\sqrt{\mu^k}}{\sqrt{n}\|\hat{M}^k\|}$. Based on the discussion of Section 2, we need to have

$$\epsilon^k_{QLSP} \leq \eta\frac{\sqrt{\mu^k}}{\sqrt{n}\|\hat{\sigma}^k\|}, \quad \epsilon^k_{QLSA} \leq \eta\frac{\sqrt{\mu^k}}{2\sqrt{n}\|\hat{\sigma}^k\|}, \text{ and } \epsilon^k_{QTA} \leq \eta\frac{\sqrt{\mu^k}}{2\sqrt{n}\|\hat{\sigma}^k\|}.$$

With this target precision, the QLSA by [9] has $\mathcal{O}(\text{polylog}(\frac{n^{1.5}\|\hat{\sigma}^k\|}{\sqrt{\mu^k}})\kappa^k_E\|E^k\|_F)$ complexity, and the QTA by [36] has $\mathcal{O}(\frac{m\sqrt{n}\|\hat{\sigma}^k\|}{\sqrt{\mu^k}})$ complexity. Since calculating E^k and $\hat{\sigma}^k$ needs $\mathcal{O}(mn)$ arithmetic operations, the total cost of building and solving the (NES) system is

$$\tilde{\mathcal{O}}(m\sqrt{n}\frac{\kappa^k_E\|\hat{\sigma}^k\|\|E^k\|_F}{\sqrt{\mu^k}}).$$

The proof is complete. □

We present the II-QIPM as Algorithm 1 for solving LO problems. In this algorithm, we use QLSA and QTA to solve the NES.

Algorithm 1 II-QIPM

```
Choose ζ > 0, γ_1 ∈ (0,1), 0 < η < β_1 < β_2 < 1,
Choose ω ≥ max{1, ‖x*, s*‖_∞}.
k ← 0, (x^0, y^0, s^0) ← (ωe, 0e, ωe), and γ_2 ← ‖(R_p^0, R_D^0)‖ / μ^0
while (x^k, y^k, s^k) ∉ PD_ζ do
    μ^k ← (x^k)^T s^k / n
    E^k ← (D^k_B̂)^{-1} Â D^k and σ̂^k ← (D^k_B̂)^{-1} b̂ − β_1 μ^k (D^k_B̂)^{-1} Â (S^k)^{-1} e + (D^k_B̂)^{-1} Â (D^k)^2 (c − A^T y^k − s^k)
    ε^k_QLSA ← η √(μ^k) / (2√n ‖σ^k‖) and ε^k_QTA ← η √(μ^k) / (2√n ‖σ^k‖)
    (Δx^k, Δy^k, Δs^k) ← solve (MNES) by QLSA+QTA with precision ε^k_QLSA and ε^k_QTA
    α̂^k ← max { ᾱ ∈ [0,1] | for all α ∈ [0, ᾱ] we have
        ((x^k, y^k, s^k) + α(Δx^k, Δy^k, Δs^k)) ∈ N(γ_1, γ_2) and
        (x^k + αΔx^k)^T (s^k + αΔs^k) ≤ (1 − α(1 − β_2))(x^k)^T s^k }
    (x^{k+1}, y^{k+1}, s^{k+1}) ← (x^k, y^k, s^k) + α̂^k (Δx^k, Δy^k, Δs^k)
    if ‖x_{k+1}, s_{k+1}‖_∞ > ω then
        return Primal or dual is infeasible.
    k ← k + 1
return (x^k, y^k, s^k)
```

It can be easily verified that e.g., $\beta_1 = 0.5$, $\beta_2 = 0.9995$, $\eta = 0.4$ and $\gamma_1 = 0.5$ yield a valid choice in Algorithm 1. In the following, we prove the polynomial complexity of Algorithm 1.

4.1 Convergence of the II-QIPM

First, we study basic properties of the proposed II-QIPM as Algorithm 1 in Lemma 4.4. Then, Lemma 4.7 shows that the sequence $\{\hat{\alpha}^k\}$ is strictly positive for all k. The iteration complexity of Algorithm 1 is proved in Theorem 4.1. In the proof, we need to analyze values of α^k such that at iteration k of Algorithm 1, the Newton step satisfies all the conditions in line 9. To ease notation, we use

$$x^k(\alpha) = x^k + \alpha\Delta x^k,\ y^k(\alpha) = y^k + \alpha\Delta y^k,\ s^k(\alpha) = s^k + \alpha\Delta s^k,\ \mu^k(\alpha) = \frac{x^k(\alpha)^T s^k(\alpha)}{n},$$
$$R_P^k(\alpha) = b - Ax^k(\alpha),\ \text{and } R_D^k(\alpha) = c - A^T y^k(\alpha) - s^k(\alpha).$$

The following lemma shows some properties of the proposed II-QIPM.

Lemma 4.4 *At iteration k of Algorithm 1, for any $\alpha \in [0,1]$, with $\mu^k = \frac{(x^k)^T s^k}{n}$, we have*

$$R_P^k(\alpha) = (1-\alpha)R_P^k, \tag{6a}$$
$$R_D^k(\alpha) = (1-\alpha)R_D^k, \tag{6b}$$
$$(x^k(\alpha))^T s^k(\alpha) \geq (1 + \alpha(\beta_1 - \eta - 1))n\mu^k + \alpha^2(\Delta x^k)^T\Delta s^k, \tag{6c}$$
$$x_i^k(\alpha)s_i^k(\alpha) \geq (1-\alpha)x_i^k s_i^k + \alpha(\beta_1 - \eta)\mu^k + \alpha^2\Delta x_i^k\Delta s_i^k \ \textit{for } i \in \{1,2,\dots,n\}. \tag{6d}$$

Proof. To prove (6a) and (6b), for any $\alpha \in [0,1]$, by (2) we have

$$R_P^k(\alpha) = b - A(x^k + \alpha\Delta x^k) = b - Ax^k - \alpha A\Delta x^k = b - Ax^k - \alpha(b - Ax^k) = (1-\alpha)R_P^k,$$
$$R_D^k(\alpha) = c - A^T(y^k + \alpha\Delta y^k) - s^k - \alpha\Delta s^k = c - A^T y^k - s^k - \alpha(A^T\Delta y + \Delta s) = (1-\alpha)R_D^k.$$

To prove (6c), based on (5), we have

$$\begin{aligned}(x^k + \alpha\Delta x^k)^T(s^k + \alpha\Delta s^k) &= (x^k)^T s^k + \alpha[(x^k)^T\Delta s^k + (s^k)^T\Delta x^k] + \alpha^2(\Delta x^k)^T\Delta s^k\\ &\geq (x^k)^T s^k + \alpha[n\beta_1\mu^k - (x^k)^T s^k - n\eta\mu^k] + \alpha^2(\Delta x^k)^T\Delta s^k\\ &= [1 + \alpha(\beta_1 - \eta - 1)](x^k)^T s^k + \alpha^2(\Delta x^k)^T\Delta s^k.\end{aligned}$$

Based on (5), we can similarly prove (6d) for all $i \in \{1,2,\dots,n\}$ as follows:

$$\begin{aligned}(x_i^k + \alpha\Delta x_i^k)(s^k + \alpha\Delta s_i^k) &= x_i^k s_i^k + \alpha(x_i^k\Delta s_i^k + s_i^k\Delta x_i^k) + \alpha^2\Delta x_i^k\Delta s_i^k\\ &\geq x_i^k s_i^k + \alpha(\beta_1\mu^k - x_i^k s_i^k - \eta\mu^k) + \alpha^2\Delta x_i^k\Delta s_i^k\\ &= (1-\alpha)x_i^k s_i^k + \alpha(\beta_1 - \eta)\mu^k + \alpha^2\Delta x_i^k\Delta s_i^k.\end{aligned}$$

Thus, the proof is complete. □

Let us define the following functions

$$\begin{aligned}G_i^k(\alpha) &= x_i^k(\alpha)s_i^k(\alpha) - \gamma_1\mu^k(\alpha) \text{ for } i \in \{1,\dots,n\},\\ g^k(\alpha) &= x^k(\alpha)^T s^k(\alpha) - (1-\alpha)(x^k)^T s^k,\\ h^k(\alpha) &= \big(1 - \alpha(1-\beta_2)\big)(x^k)^T s^k - x^k(\alpha)^T s^k(\alpha).\end{aligned}$$

Using the defined function, the following lemma providing sufficient conditions for a step length which sufficiently reduce the complementarity gap and keeps the next iterate in the neighborhood $\mathcal{N}(\gamma_1,\gamma_2)$.

Lemma 4.5 *For step length $0 < \alpha \leq 1$, if $G_i^k(\alpha) \geq 0$, $g^k(\alpha) \geq 0$, and $h^k(\alpha) \geq 0$ holds, then $\big(x^k(\alpha), y^k(\alpha), s^k(\alpha)\big) \in \mathcal{N}(\gamma_1, \gamma_2)$ and the Armijo condition*

$$(x^k + \alpha \Delta x^k)^T (s^k + \alpha \Delta s^k) \leq \big(1 - \alpha(1-\beta_2)\big)(x^k)^T s^k$$

holds.

Proof. It is easy to verify that that conditions $G_i^k(\alpha) \geq 0$ and $g^k(\alpha) \geq 0$ lead to

$$\begin{aligned} x_i^k(\alpha) s_i^k(\alpha) &\geq \gamma_1 \mu^k(\alpha) \text{ for } i \in \{1, \dots, n\}, \\ (x^k + \alpha \Delta x^k)^T (s^k + \alpha \Delta s^k) &\leq \big(1 - \alpha(1-\beta_2)\big)(x^k)^T s^k, \end{aligned}$$

respectively. Since $g^k(\alpha) \geq 0$, i.e. $x^k(\alpha)^T s^k(\alpha) \geq (1-\alpha)(x^k)^T s^k$, we have

$$\frac{\big\|\big(R_P^k(\alpha), R_D^k(\alpha)\big)\big\|}{\|(R_P^0, R_D^0)\|} = \frac{(1-\alpha)\|(R_P^k, R_D^k)\|}{\|(R_P^0, R_D^0)\|} \leq \frac{(1-\alpha)\mu^k}{\mu^0} \leq \frac{\mu^k(\alpha)}{\mu^0}.$$

Further, as $\gamma_2 = \frac{\|(R_P^0, R_D^0)\|}{\mu^0}$, we can conclude that $\big(x^k(\alpha), y^k(\alpha), s^k(\alpha)\big) \in \mathcal{N}(\gamma_1, \gamma_2)$. ☐

In order to prove polynomial complexity of II-QIPM we need to find a positive lower bound for the step length. The following lemma is bounding some remaining elements to get the step length bound.

Lemma 4.6 *There exist $0 \leq \nu^k = \mathcal{O}(n^2 \mu^k)$ such that $\left| \Delta x_i^k \Delta s_i^k - \gamma_1 \frac{(\Delta x^k)^T \Delta s^k}{n} \right| \leq \nu^k$ for $i \in \{1, 2, \dots, n\}$ and $|(\Delta x^k)^T \Delta s^k| \leq \nu^k$.*

Proof. To prove this lemma, we need to do the following steps

Step 1. finding bound $\mathcal{C}_1 = \mathcal{O}(n\mu^k)$ such that $\omega \theta^{k-1} \|x^k, s^k\|_1 \leq C_1$
Step 2. finding bound $\mathcal{C}_2 = \mathcal{O}(n\sqrt{\mu^k})$ such that $\|D^{-1} \Delta x^k\| \leq \mathcal{C}_2$ and $\|D \Delta s^k\| \leq \mathcal{C}_2$
Step 3. finding bound $0 \leq \nu^k = \mathcal{O}(n^2 \mu^k)$ such that

$$|(\Delta x^k)^T \Delta s^k| \leq \nu^k \text{ and } \left| \Delta x_i^k \Delta s_i^k - \gamma_1 \frac{(\Delta x^k)^T \Delta s^k}{n} \right| \leq \nu^k \text{ for } i \in \{1, 2, \dots, n\}.$$

Step 1. Let us define $\theta^k = \prod_{i=0}^k (1 - \hat{\alpha}^i)$. One can verify that

$$R_P^k = \theta^{k-1} R_p^0, \text{ and } R_D^k = \theta^{k-1} R_D^0. \tag{7}$$

Based on the definition of $\mathcal{N}(\gamma_1, \gamma_2)$ and the choice of γ_2, we have

$$\frac{\|(R_P^k, R_D^k)\|}{\mu^k} \leq \frac{\|(R_P^0, R_D^0)\|}{\mu^0}, \tag{8}$$

implying $\mu^k \geq \theta^{k-1} \mu^0$. We also define

$$(\bar{x}^k, \bar{y}^k, \bar{s}^k) = \theta^{k-1}(x^0, y^0, s^0) + (1 - \theta^{k-1})(x^*, y^*, s^*) - (x^k, y^k, s^k),$$

where, $(x^*, y^*, s^*) \in \mathcal{PD}^*$. One can verify that

$$\begin{aligned} A^T \bar{y}^k + \bar{s}^k &= 0, \\ A \bar{x}^k &= 0. \end{aligned}$$

Since $\bar{s}^k$ is in the row space of A and $\bar{x}^k$ is in the null space of A, we have $(\bar{x}^k)^T\bar{s}^k = 0$, or equivalently,

$$[\theta^{k-1}x^0 + (1-\theta^{k-1})x^* - x^k]^T[\theta^{k-1}s^0 + (1-\theta^{k-1})s^* - s^k] = 0.$$

Since $(x^*, s^*, x^k, s^k) \geq 0$, $x^*s^* = 0$, and $(x^0, s^0) = \omega e$, we can write

$$[\theta^{k-1}x^0 + (1-\theta^{k-1})x^* - x^k]^T[\theta^{k-1}s^0 + (1-\theta^{k-1})s^* - s^k] = 0, \tag{9}$$

$$(\theta^{k-1})^2(s^0)^Tx^0 + \theta^{k-1}(1-\theta^{k-1})[(s^0)^Tx^* + (s^*)^Tx^0] + (s^k)^Tx^k \geq \theta^{k-1}[(s^0)^Tx^k + (x^0)^Ts^k], \tag{10}$$

$$(\theta^{k-1})^2n\mu^0 + 2\theta^{k-1}(1-\theta^{k-1})n\mu^0 + (s^k)^Tx^k \geq \theta^{k-1}\omega[(e)^Tx^k + (e)^Ts^k], \tag{11}$$

$$\theta^{k-1}n\mu^k + 2(1-\theta^{k-1})\mu^k + n\mu^k \geq \theta^{k-1}\omega\|x^k, s^k\|_1. \tag{12}$$

Inequality (11) is obtained by using $(x^0, s^0) = \omega e$ and $\|(x^*, s^*)\|_\infty \leq \omega$, and the last inequality is obtained by using (7). Thus, $\mathcal{C}_1 = \mathcal{O}(n\mu^k)$.

Step 2. In this step we define

$$(\bar{x}^k, \bar{y}^k, \bar{s}^k) = (\Delta x^k, \Delta y^k, \Delta s^k) + \theta^{k-1}(x^0, y^0, s^0) - \theta^{k-1}(x^*, y^*, s^*),$$

where, $(x^*, y^*, s^*) \in \mathcal{PD}^*$. Similar to Step 1, one can verify that

$$\begin{aligned} A^T\bar{y}^k + \bar{s}^k &= 0,\\ A\bar{x}^k &= 0,\\ (\bar{x}^k)^T\bar{s}^k &= 0. \end{aligned}$$

Consequently, one can verify

$$\begin{aligned} &\|D^{-1}(\Delta x^k + \theta^{k-1}(x^0 - x^*)) + D(\Delta s^k + \theta^{k-1}(s^0 - s^*))\|\\ &\quad= \|D^{-1}(\Delta x^k + \theta^{k-1}(x^0 - x^*))\| + \|D(\Delta s^k + \theta^{k-1}(s^0 - s^*))\|\\ &\quad\leq \|(XS)^{-1/2}\|(\|XSe - \beta_1\mu^ke\| + \eta\mu^k) + \theta^{k-1}\|D^{-1}(x^0 - x^*))\| + \theta^{k-1}\|D(s^0 - s^*)\|. \end{aligned}$$

Now, we have

$$\|D^{-1}\Delta x^k\| \leq \|(XS)^{-1/2}\|(\|XSe - \beta_1\mu^ke\| + \eta\mu^k) + 2\theta^{k-1}\|D^{-1}(x^0 - x^*))\| + 2\theta^{k-1}\|D(s^0 - s^*)\|.$$

According to pages 116-118 of [38], we can derive the following inequalities:

$$\begin{aligned} \|(XS)^{-1/2}\| &\leq \frac{1}{\sqrt{\gamma_1}\sqrt{\mu^k}},\\ \|XSe - \beta_1\mu^ke\| &\leq n\mu^k,\\ \theta^{k-1}\|D^{-1}(x^0 - x^*))\| + \theta^{k-1}\|D(s^0 - s^*)\| &\leq \theta^{k-1}\|x^k, s^k\|_1\|(XS)^{-1/2}\|\max(\|x^0 - x^*\|, \|s^0 - s^*\|)\\ &\leq \frac{\omega\theta^{k-1}\|x^k, s^k\|_1}{\sqrt{\gamma_1}\sqrt{\mu^k}} \leq \frac{C_1}{\sqrt{\gamma_1}\sqrt{\mu^k}}. \end{aligned}$$

Thus, $\|D^{-1}\Delta x^k\| \leq \frac{n\mu^k + \eta\mu^k + C_1}{\sqrt{\gamma_1}\sqrt{\mu^k}} = C_2$, where $C_2 = \mathcal{O}(n\sqrt{\mu^k})$. We can similarly show that $\|D\Delta s^k\| \leq C_2$.

Step 3. Based on Steps 1 and 2, we have

$$
\begin{aligned}
(\Delta x^k)^T \Delta s^k &\leq \|D^{-1}\Delta x^k\| \|D\Delta s^k\| \leq \mathcal{C}_2^2, \\
|\Delta x_i^k \Delta s_i^k| &\leq \|D^{-1}\Delta x^k\| \|D\Delta s^k\| \leq \mathcal{C}_2^2, \\
\left| \Delta x_i^k \Delta s_i^k - \gamma_1 \frac{(\Delta x^k)^T \Delta s^k}{n} \right| &\leq (1 + \frac{\gamma_1}{n}) C_2^2 \leq 2C_2^2.
\end{aligned}
$$

Thus, $\nu^k = 2C_2^2 = \mathcal{O}(n^2 \mu^k)$. □

In the next lemma, we present a strictly positive lower bound for $\hat{\alpha}^k$. In the proof, we use parameters δ_1, δ_2, and δ_3 defined as follows:

$$
\delta_1 = \frac{(1-\gamma_1)(\beta_1 - \eta)}{n} > 0, \quad \delta_2 = \beta_1 - \eta > 0, \quad \delta_3 = \beta_2 - \beta_1 + \eta > 0. \tag{13}
$$

Lemma 4.7 *At line 9 of Algorithm 1, at iteration k, we have*

$$
\hat{\alpha}^k \geq \tilde{\alpha}^k := \min \left\{ 1, \quad \min\{\delta_1, \delta_2, \delta_3\} \frac{(x^k)^T s^k}{\nu^k} \right\} > 0.
$$

Proof. It is enough to show that the conditions of Lemma 4.5 hold for all $\alpha \in [0, \tilde{\alpha}^k]$. Based on Lemma 4.4, for any $\alpha \in [0, \tilde{\alpha}^k]$, we have

$$
\begin{aligned}
G_i^k(\alpha) &= (x_i^k + \alpha \Delta x_i^k)(s_i^k + \alpha \Delta s_i^k) - \gamma_1 \frac{(x^k + \alpha \Delta x^k)^T (s^k + \alpha \Delta s^k)}{n} && \text{(14a)} \\
&= (1-\alpha) x_i^k s_i^k + \alpha(\beta_1 - \eta)\mu^k + \alpha^2 \Delta x_i^k \Delta s_i^k - \gamma_1 \frac{(1 + \alpha(\beta_1 - \eta - 1))(x^k)^T s^k + \alpha^2 (\Delta x^k)^T \Delta s^k}{n} && \text{(14b)} \\
&\geq \alpha^2 \big(\Delta x_i^k \Delta s_i^k - \frac{\gamma_1}{n} (\Delta x^k)^T \Delta s^k\big) + (1-\alpha)\big(x_i^k s_i^k - \frac{\gamma_1}{n}(x^k)^T s^k\big) + \alpha(\beta_1 - \eta)(1-\gamma_1)\mu^k && \text{(14c)} \\
&\geq -\alpha^2 \nu^k + \alpha \delta_1 (x^k)^T s^k \geq \alpha\big(\delta_1 (x^k)^T s^k - \nu^k \tilde{\alpha}^k\big) \geq 0. && \text{(14d)}
\end{aligned}
$$

Equality (14b) follows from equation (6a) of Lemma 4.4, and inequality (14d) is due to the definition of $\tilde{\alpha}^k$ and the neighborhood $\mathcal{N}(\gamma_1, \gamma_2)$. Similarly, we show that $g_i^k(\alpha) \geq 0$ as

$$
\begin{aligned}
g_i^k(\alpha) &= x^k(\alpha)^T s^k(\alpha) - (1-\alpha)(x^k)^T s^k \\
&\geq \big(1 + \alpha(\beta_1 - \eta - 1)\big)(x^k)^T s^k + \alpha^2 (\Delta x^k)^T \Delta s^k - (1-\alpha)(x^k)^T s^k \\
&\geq \alpha(\beta_1 - \eta)(x^k)^T s^k - \alpha^2 \nu^k \\
&\geq \alpha \delta_2 (x^k)^T s^k - \alpha^2 \nu^k \\
&\geq \alpha\big(\delta_2 (x^k)^T s^k - \nu^k \tilde{\alpha}^k\big) \geq 0.
\end{aligned}
$$

Again, by (6c) of Lemma 4.4, we have for all $\alpha \in [0, \tilde{\alpha}^k]$

$$
\begin{aligned}
h^k(\alpha) &= \big(1 - \alpha(1-\beta_2)\big)(x^k)^T s^k - (x^k + \alpha \Delta x^k)^T (s^k + \alpha \Delta s^k) \\
&= \big(1 - \alpha(1-\beta_2)\big)(x^k)^T s^k - \big(1 + \alpha(\beta_1 - \eta - 1)\big)(x^k)^T s^k - \alpha^2 (\Delta x^k)^T \Delta s^k \\
&\geq \alpha(\beta_2 - \beta_1 + \eta)(x^k)^T s^k - \alpha^2 \nu^k \\
&\geq \alpha \delta_3 (x^k)^T s^k - \alpha^2 \nu^k \\
&\geq \alpha\big(\delta_3 (x^k)^T s^k - \nu^k \tilde{\alpha}^k\big) \geq 0.
\end{aligned}
$$

We showed that for all $\alpha \in [0, \tilde{\alpha}^k]$, all the conditions of Lemma 4.5 hold. Thus, we can conclude that $\tilde{\alpha}^k \leq \hat{\alpha}^k$ and the proof is complete. □

By Lemma 4.7, we have a strictly positive lower bound for step length $\tilde{\alpha}^k$ to remain in the neighborhood of the central path while we decrease the optimality gap. In what follows, using the results of the previous lemmas, we establish the complexity of the Algorithm 1.

Theorem 4.1 *If Algorithm 1 does not terminate in line 11, then it reaches a ζ-optimal solution in at most $\mathcal{O}(n^2 \log \frac{1}{\zeta})$ iterations.*

Proof. Based on Lemma 4.7, we have

$$\hat{\alpha}^k \geq \alpha^k \geq \alpha^L := \min\left\{1, \min\{\delta_1, \delta_2, \delta_3\} \frac{n\mu^k}{\nu^k}\right\} \in (0, 1].$$

Hence, by the definition of the neighborhood, we have

$$(x^k)^T s^k \leq \left(1 - \alpha^L(1 - \beta_2)\right)^k (x^0)^T s^0.$$

This implies that $\lim_{k\to\infty}(x^k)^T s^k = 0$. Thus, the algorithm terminates in finite number of steps. The algorithm stops when

$$\mu^k \leq \left(1 - \alpha^L(1 - \beta_2)\right)^k \mu^0 \leq \zeta\mu^0.$$

By the definition of α^L and Lemma 4.6, we have $\frac{1}{\alpha^L} = \mathcal{O}(n^2)$. We can conclude that $k = \mathcal{O}(n^2 \log \frac{1}{\zeta})$. The proof is complete. □

Remark 4.1 The sequences of $\{\mu^k\}$, primal infeasibility $\{\|Ax^k - b\|\}$, and dual infeasibility $\{\|A^T y^k + s^k - c\|\}$, generated by II-QIPM, converge linearly to zero.

Remark 4.2 Based on Theorem 3.1, we can calculate an exact solution by a rounding procedure if

$$\mu_k \leq \zeta\mu^0 \leq 2^{-2L}, \text{ and } \zeta \leq 2^{-4L}.$$

Consequently, the iteration complexity of II-QIPM for finding an exact solution is $\mathcal{O}(n^2 L)$.

In the next section, we provide the total time complexity of II-QIPM.

4.2 Total time complexity of the II-QIPM

As discussed in Lemma 4.3, we need to solve the NES at each iteration of the II-QIPM by subsequent application of QLSA and QTA, which requires $\tilde{\mathcal{O}}(mn^{1.5} \frac{\kappa_E^k \|\hat{\sigma}^k\| \|E^k\|_F}{\sqrt{\mu^k}})$ computational cost. We can calculate the total time complexity of the II-QIPM as the product of the complexity of the QLSA and the number of iterations of the II-QIPM. The computational cost of the QLSA depends on κ_E^k, $\|\hat{\sigma}^k\|$, $\|E^k\|_F$, and μ^k which change through the algorithm. In the following theorem, we bound them properly and obtain the detailed total time complexity of the proposed II-QIPM algorithm.

Theorem 4.2 *The total time complexity of the proposed II-QIPM with QLSA by Chakraborty et al. [9] and QTA by Van Apeldoorn et al. [36] is*

$$\tilde{\mathcal{O}}\left(n^2 \log \frac{1}{\zeta}\left[n^2 + \left(\frac{n\kappa_{\hat{A}}\|\hat{A}\|_F}{\zeta^3}\right)\frac{m\sqrt{n}(\|\hat{A}\| + \|\hat{b}\|)}{\zeta^2}\right]\right).$$

Proof. To establish the total time complexity of II-QIPM, we need to analyze how the matrices M^k and E^k evolve through the iterations. As in [32], considering the optimal partition B and N, we have

$$\frac{x_i^k}{s_i^k} = \mathcal{O}(\frac{1}{\mu^k}) \to \infty \text{ for } i \in B \qquad \text{and} \qquad \frac{x_i^k}{s_i^k} = \mathcal{O}(\mu^k) \to 0 \text{ for } i \in N. \tag{15}$$

Appropriate bounds are provided in the following results.

(i) Based on Theorem 4.1, we have at the termination $\frac{1}{\mu^k} = \mathcal{O}(\zeta^{-1})$ and $\mu^k \leq \mathcal{O}(\mu^0) = \mathcal{O}(\omega^2)$.

(ii) Since $\|E^k\|_F \leq \|(D^k)^{-1}\|_F \|\hat{A}\|_F \|D^k\|_F$, and $\|D^k\|_F = \mathcal{O}(\sqrt{\frac{n}{\mu^k}}) = \mathcal{O}(\sqrt{\frac{n}{\zeta}})$ by (15). Similarly, $\|(D^k)^{-1}\|_F = \mathcal{O}(\sqrt{\frac{n}{\zeta}})$, and we have $\|E^k\|_F = \mathcal{O}\big(\frac{n}{\zeta}\|\hat{A}\|_F\big)$.

(iii) Let $\kappa_{\hat{A}}$ be the condition number of $\hat{A}$. Using (15), we have $\kappa_E^k = \mathcal{O}\big(\zeta^{-2}\kappa_{\hat{A}}\big)$.

(iv) In time complexity of II-QIPM, we also have $\frac{\|\sigma^k\|}{\mu^k}$ coming from precision of QLSA and QTA. We can easily verify that

$$\begin{aligned}\frac{\|\hat{\sigma}^k\|}{\mu^k} &\leq \frac{\|(D^k)^{-1}\|}{\mu^k}(\|\hat{b}\| + \|\hat{A}X^k(S^k)^{-1}\|\|c - A^T y^k - s^k\| + \beta_1\|\mu^k \hat{A}(S^k)^{-1}e\|) \\ &\leq \frac{\|(D^k)^{-1}\|}{\mu^k}(\|\hat{b}\| + \|\hat{A}X^k(S^k)^{-1}\|\frac{\mu^k}{\gamma_1\mu^0}\|R_D^0\| + \|\hat{r}^k\| + \beta_1\|\hat{A}(S^k)^{-1}e\|\mu^k).\end{aligned}$$

Since $\frac{\|R_D^0\|}{\mu^0} = \mathcal{O}(\frac{\|c\|+\sqrt{n}\omega}{\omega^2})$ is small number due to $\omega = 2^L$, we can get bounds $\frac{1}{\epsilon_{QLSP}} = \mathcal{O}(\frac{\|\sigma^k\|}{\mu^k}) = \mathcal{O}(\frac{\|\hat{A}\|+\|\hat{b}\|}{\zeta^2})$, and $\frac{1}{\epsilon_{QLSA}} = \frac{1}{\epsilon_{QTA}} = \mathcal{O}(\frac{\|\hat{A}\|+\|\hat{b}\|}{\zeta^2})$.

(v) Based on Lemma 4.3, the complexity of QLSA by Chakraborty et al. [9] for building and solving the NES is

$$\tilde{\mathcal{O}}(\frac{n\kappa_{\hat{A}}\|\hat{A}\|_F}{\zeta^3}).$$

In addition, The complexity of QTA by Van Apeldoorn et al. [36] for solving the NES is

$$\mathcal{O}(\frac{m\sqrt{n}(\|\hat{A}\| + \|\hat{b}\|)}{\zeta^2}).$$

Thus, the detailed time complexity of the proposed II-QIPM with QLSA by Chakraborty et al. [9] and QTA by Van Apeldoorn et al. [36] is

$$\tilde{\mathcal{O}}\bigg(n^2 \log\frac{1}{\zeta}\Big[n^2 + \Big(\frac{n\kappa_{Ahat}\|\hat{A}\|_F}{\zeta^3}\Big)\frac{m\sqrt{n}(\|\hat{A}\| + \|\hat{b}\|)}{\zeta^2}\Big]\bigg). \tag{16}$$

The time complexity can be achieved by multiplying the number of iteration of II-QIPM and the total cost of each iteration, including building and solving it by QLSA+QTA. Thus, the proof is complete. □

By assuming that $\|\hat{A}\|_F \geq \|\hat{A}\| \geq \|\hat{b}\|$, the complexity of II-QIPM can be simplified as

$$\tilde{\mathcal{O}}\Big(n^{4.5}\zeta^{-5}\|\hat{A}\|_F\kappa_{\hat{A}}\Big). \tag{17}$$

In the complexity of II-QIPM, the ζ^{-1} factors come from bounding the condition number and from QTA. Based on Theorem 3.1, $\zeta^{-1} = 2^{\mathcal{O}(L)}$ leading to exponential complexity. We discuss how we can solve this and improve the complexity of the algorithm by stopping II-QIPM early, e.g., $\zeta = 10^{-2}$, and using the Iterative Refinement scheme discussed in the Section 5 to improve the precision. In Section 6, we investigate pros and cons of different systems.

Remark 4.3 Some QLSAs, such as Harrow et al. [19] and Childs et al. [11], take advantage of the sparsity of the NES. Consider (MNES), the sparsity of $\hat{M}^k = \hat{A}X^k(S^k)^{-1}\hat{A}^T$ is independent of X^k and S^k. So, let d be the maximum number of nonzero elements in any row or column of the matrix $\hat{A}\hat{A}^T$. Since matrix $\hat{A}$ has two blocks $\left[I \; A_{\hat{B}}^{-1}A_{\hat{N}}\right]$, we have $d \leq \min\{m, n-m+1\}$. As matrix $\hat{A}$ determines the sparsity of matrix M^k, so we can take advantage of the sparsity structure of matrix $\hat{A}$. In case, $\hat{A}$ is mostly sparse, e.g., $n-m \ll m$, but has a few dense columns, then this structure can be exploited. As described in [1], the sparse part can be separated to solve a sparse linear system by QLSA, and then the use of the Sherman-Morrison-Woodbury [33] formula allows calculating the solution of the original linear system efficiently.

5 Iterative Refinement Method

Based on Remark 4.2, we need $\zeta \leq 2^{-4L}$ to have an exact solution for the LO problem with integer data. The proposed II-QIPM has exponential complexity to find an exact solution. An Iterative Refinement (IR) scheme can be employed to achieve polynomial complexity. By this scheme, a series of LO problems are solved by the II-QIPM with low precision $\hat{\zeta}$, e.g., $\hat{\zeta} = 10^{-2}$, and an IR method improves the precision to reach an exact solution. In the classical IPM literature, the IR method by Gleixner et al. [18] and the Rational Reconstruction method by Gleixner and Steffy [17] employed low-precision methods to generate high-precision solution. Here, we adopt the IR method to generate high precision solution that allows the identification of an exact optimal solution. While using only low-precision II-QIPM, Theorem 5.1 is the foundation of IR method.

Theorem 5.1 (Gleixner et al. [18]) *Let the primal problem (P) be given as (1). For $\tilde{x} \in \mathbb{R}^n$ and $\tilde{y} \in \mathbb{R}^m$ and scaling factor $\nabla > 0$ consider the refining problem $(\bar{P})$*

$$\max\left\{\nabla \bar{c}^T x | Ax = \nabla \bar{b} \text{ and } x \geq 0\right\},$$

where $\bar{c} = c - A^T\tilde{y}$ and $\bar{b} = b - A\tilde{x}$. Then $\bar{x}$ and $\bar{y}$ are $\hat{\zeta}$-optimal solution for problem $(\bar{P})$ if and only if $\tilde{x} + \frac{1}{\nabla}\bar{x}$ and $\tilde{y} + \frac{1}{\nabla}\bar{y}$ are the $\frac{\hat{\zeta}}{\nabla}$-optimal solution for problem (P).

As presented in Algorithm 2, we use the II-QIPM of Algorithm 1 to solve the refining model and update the solution with an intelligent scaling procedure. Theorem 5.2 shows that a polynomial number of iterations are sufficient to reach an exact optimal solution.

Algorithm 2 Iterative Refinement (IR-II-QIPM)

Require: $\left(A \in \mathbb{Z}^{m\times n}, b \in \mathbb{Z}^m, c \in \mathbb{Z}^n, \hat{\zeta} = 10^{-2}, \zeta = 2^{-2L}\right)$
Choose scaling multiplier $\rho \in \mathbb{N}$ such that $\rho > 1$
$k \leftarrow 0$
$(x^{*0}, y^{*0}, s^{*0}) \leftarrow$ **solve** (A, b, c) using Algorithm 1 with $\hat{\zeta}$ precision
while $r^k > \zeta$ **do**
  $\pi^k \leftarrow \max\left\{r^k, \frac{1}{\rho\nabla^k}\right\}$
  $\nabla^k \leftarrow 2^{\lceil \log(1/\pi_k)\rceil}$
  $(\hat{x}^k, \hat{y}^k, \hat{s}^k) \leftarrow$ **solve** $(A, \nabla^k\bar{b}^k, \nabla^k\bar{c}^k)$ using the Algorithm 1 with $\hat{\zeta}$ precision
  $x^{*k+1} \leftarrow x^{*k} + \frac{1}{\nabla^k}\hat{x}^k$ and $y^{*k+1} \leftarrow y^{*k} + \frac{1}{\nabla^k}\hat{y}^k$
  $\bar{b}^{k+1} \leftarrow b - Ax^{*k+1}$ and $\bar{c}^{k+!} \leftarrow c - A^Ty^{*k+1}$
  $r^{k+1} \leftarrow \max\left\{\max_j |\bar{b}_j^{k+1}|, \max_i(-\bar{c}_i^{k+1}), \sum_i |\bar{c}_i^{k+1}x_i^{*k+1}|\right\}$
  $k \leftarrow k+1$

Theorem 5.2 (Corollary 3.6 in [18]) *Number of iterations of Algorithm 2 is at most*

$$\left\lceil \frac{\log(\zeta)}{\log(\hat{\zeta})} \right\rceil = \left\lceil \frac{-4L\log(2)}{-2} \right\rceil = \mathcal{O}(L).$$

One can observe that, except ω, all parameters in the complexity of II-QIPM as specified in (17), are constant in all iterations of the IR method . Lemma 5.1 provides an upper bound for ω^k at iteration k of the IR method.

Lemma 5.1 *At the k^{th} iteration of the IR method, let $\omega^k \geq \max\{1, \|\hat{x}^{*k}, \hat{s}^{*k}\|_\infty\}$ where $(\hat{x}^{*k}, \hat{y}^{*k}, \hat{s}^{*k})$ is the exact optimal solution of the refining problem $(A, \nabla^k \bar{b}_{k-1}, \nabla^k \bar{c}_{k-1})$. Then, $\omega^k = \mathcal{O}\big((2\rho)^L\big)$.*

Proof. From Theorem 5.2 and last line of Algorithm 2, we have

$$x^* = x_0^* + \sum_{k=1}^{\mathcal{O}(L)} \frac{1}{\nabla^k}\hat{x}^k, \text{ and } s^* = s_0^* + \sum_{k=1}^{\mathcal{O}(L)} \frac{1}{\nabla^k}\hat{s}^k.$$

Based on Lemma 3.1, we know that $\|x^*, s^*\|_\infty \leq 2^L$, then we have $\|\hat{x}^{*k}, \hat{s}^{*k}\|_\infty \leq \nabla^k 2^L$.

Based on the procedure of updating the scaling factor in Algorithm 2, we can drive $\nabla^k = \mathcal{O}(\rho^k)$. We can conclude that $\omega^k = \mathcal{O}\big((2\rho)^L\big)$. □

In Theorem 5.3, we have the total time complexity of the IR method using the proposed II-QIPM to find an exact optimal solution for LO problems.

Theorem 5.3 *Let $\hat{\zeta} = 10^{-2}$, then the total time complexity of finding an exact optimal solution using the IR-II-QIPM Algorithm 1 for solving the LO problem (1) is polynomial with*

$$\tilde{\mathcal{O}}\bigg(n^2 L\Big[n^2 + (n\kappa_{\hat{A}}\|\hat{A}\|_F)n^{1.5}(\|\hat{A}\| + \|\hat{b}\|)\Big]\bigg),$$

the arithmetic operations, where $\hat{A}$ and $\hat{b}$ are preprocessed A and b.

Proof. The proof comes from combining the result of Theorem 5.2, the total time complexity of the proposed II-QIPM in (16), and Lemma 5.1. □

Corollary 5.1 *The simplified complexity of the proposed IR-II-QIPM using the (MNES) is*

$$\mathcal{O}(n^{4.5}L\|\hat{A}\|_F\kappa_{\hat{A}}).$$

The following section discusses different Newton systems to compare their adaptability in the II-QIPM.

6 Different Newton Systems

This section discusses different Newton systems. In Section 4, we proposed an II-QIPM based on solving the NES with $m \times m$ symmetric positive definite, i.e., with a Hermitian, coefficient matrix which is appropriate for QLSAs. The full Newton system (2) when solved by a QLSA, delivers a solution for

$$\begin{aligned} A\Delta x^k &= b - Ax^k + r_P, \\ A^T\Delta y^k + \Delta s^k &= c - A^T y^k - s^k + r_D, \\ X^k\Delta s^k + S^k\Delta x^k &= \beta_1\mu^k e - X^k s^k + r_C, \end{aligned} \tag{18}$$

where (r_P, r_D, r_C) are residuals generated by an inexact QLSA coupled with QTA. As discussed earlier, when solving equation (4) classically through the NES, then we have $r_C = r_D = 0$. One can easily adapt the proofs of the proposed II-QIPM for the II-QIPM when the full Newton system is solved.

The Augmented system is

$$\begin{aligned} A\Delta x^k &= b - Ax^k + r'_P, \\ A^T\Delta y^k + {X^k}^{-1}S^k\Delta x^k &= c - A^Ty^k - \beta_1\mu^k {X^k}^{-1}e + r'_D. \end{aligned} \tag{19}$$

After solving the Augmented system, Δs^k can be calculated by the last line of system (18). For the Augmented system approach, we have $r_C = 0$. Again, one can redo the convergence proof for the II-QIPM with an Augmented system. The full Newton system and the Augmented system are neither symmetric nor Hermitian. Thus, as an intermediate step, one need the procedure mentioned in Section 2 to make the matrix of the system Hermitian. This increases the size of the linear system. This increased size is not impacting the theoretical complexity of QIPMs significantly.

Table 2 shows that the Augmented system and the full Newton system have better dependence on n which comes from better bound for the norm of the right-hand side vectors, thus the required scaling has smaller impact on complexity. For the full Newton system and the Augmented system, we build the coefficient matrix in a classical computer with $\mathcal{O}(n^2)$ complexity. We can not improve it by quantum computing, since we need at least $\mathcal{O}(n^2)$ to read data from classical computer to encode in quantum. In this respect, the full Newton system has the best complexity. Although the Augmented system has better dependence on n, its complexity is not polynomial, since the norm of the RHS vector for this system depends on ω. The following section uses the II-QIPM with the NES because it has smaller dimension than the other systems and more tractable in current quantum computers and simulators. In the presence of access to large scale reliable quantum computers with large number of Qubits may make the II-QIPMs with the full Newton system more attractive due to its better complexity and better sparsity properties.

	Full Newton System	Augmented System	NES
Size	$2(2n+m)$	$2(n+m)$	m
$\frac{\\|\sigma\\|}{\mu}$	$\mathcal{O}(\sqrt{n})$	$\mathcal{O}\left(\frac{\sqrt{n}\omega}{\zeta}\right)$	$\mathcal{O}\left(\frac{\\|\hat{A}\\|+\\|\hat{b}\\|}{\zeta}\right)$
QLSA	$\tilde{\mathcal{O}}\left(\frac{\sqrt{n}\kappa_A^2\\|A\\|_F}{\zeta^3}\right)$	$\tilde{\mathcal{O}}\left(\frac{\sqrt{n}\kappa_A^2\\|A\\|_F}{\zeta^3}\right)$	$\tilde{\mathcal{O}}\left(\frac{n\kappa_{\hat{A}}\\|\hat{A}\\|_F}{\zeta^3}\right)$
QTA	$\mathcal{O}((2n+m)\sqrt{n})$	$\mathcal{O}\left(\frac{(m+n)\sqrt{n}\omega}{\zeta}\right)$	$\mathcal{O}\left(\frac{n^{1.5}(\\|\hat{A}\\|+\\|\hat{b}\\|)}{\zeta^2}\right)$
II-QIPM	$\tilde{\mathcal{O}}\left(n^4\zeta^{-3}\\|A\\|_F\kappa_A^2\log(\omega)\right)$	$\tilde{\mathcal{O}}\left(n^4\zeta^{-4}\\|A\\|_F\kappa_A^2\omega\right)$	$\tilde{\mathcal{O}}\left(n^{4.5}\zeta^{-5}\\|\hat{A}\\|_F\kappa_{\hat{A}}\right)$
IR-II-QIPM	$\tilde{\mathcal{O}}\left(n^4L\\|A\\|_F\kappa_A^2\right)$	$\tilde{\mathcal{O}}\left(n^4L2^{2L}\\|A\\|_F\kappa_A^2\right)$	$\tilde{\mathcal{O}}\left(n^{4.5}L\\|\hat{A}\\|_F\kappa_{\hat{A}}\right)$

Table 2: Size and complexity implications of different Newton systems.

7 Numerical Experiments

This section provides numerical results for the proposed II-QIPM using the QISKIT AQUA quantum simulator. Due to the limitation on available Qubits on current quantum computers and simulators, we prefer to use the NES, which has smaller dimension. The numerical results are run on a workstation with Dual Intel Xeon® CPU E5-2630 @ 2.20 GHz (20 cores) and 64 GB of RAM. For the computational experiments, we have developed a Python `qipm` package available for public use [1].

IBM has implemented a QLSA, which is similar to the HHL method, without block-encoding and QRAM. With the current technology, the number of available qubits in gate-based quantum computers is limited to less than one hundred. One of the main issues with quantum computers is that they are not scalable compared to classical computers. Currently, larger NISQ[2] devices suffer more from the lack of precision. On the other hand, quantum simulator algorithms are computationally expensive. The maximum number of qubits in a quantum simulator is roughly similar to that in an actual quantum computer. The main advantage of using a quantum simulator is that we do not need to handle the noise of NISQ devices. However, we still need to handle a high error level because of insufficient qubits and high cost of QLSA+QTA for finding high quality solutions.

We used two post-processing procedures to improve the performance of the QLSA.

(i) We first scale the linear system solution such that it satisfies the equality $\|Mz\| = \|\sigma\|$.
(ii) We also check the sign of the linear system solution by comparing it with its negate.

The condition number increases the solution time. Notably, the dimension of the linear system increases the solution time as a step function since for building the quantum circuit, the dimension of the system must be a power of two, and the simulator expands the system size to the smallest possible power of two. Table 3 presents the number of qubits in quantum circuits for achieving the same precision. The Qiskit simulator error oscillates between zero and the norm of an actual solution. In some cases, the IBM Qiskit simulator fails, and it reports zero vector as a solution. The simulator has a parameter for tuning precision, but the simulator does not obey the predefined precision, and we could not find meaningful trend in precision of the simulator w.r.t condition number of the coefficient matrix and the dimension of the system. In the following sections, we discuss the implementation of our II-QIPM and IR-II-QIPM using the QISKIT simulator of QLSA and evaluate their performance.

κ	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}
#qubits	6	6	7	8	9	10	11	12	12	13	15

Table 3: Size of the circuit for linear systems with different condition numbers.

7.1 Evaluation of the II-QIPM

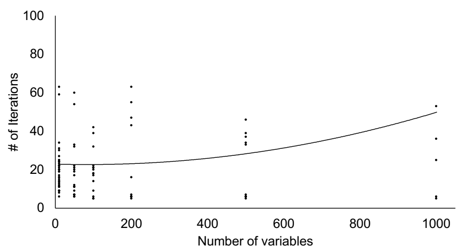
As discussed in Section 2, QLSAs have better dependence on the size of the linear system than classical algorithms. However, insufficiently precise qubits cause a significant error in the solution of a linear system. Here, we use the IBM Qiskit simulator to solve linear systems arising in the proposed II-QIPM. Here, for a fair comparison, we considered those experiments that reached the desired precision. The running time of the quantum computers and quantum simulators is not comparable. Thus, instead of running time, we use the number of iterations as a performance measure.

We use the random instance generator of [30], where the norm of primal and dual solutions is set to two. The norm of the coefficient matrix and the RHS vector are set to one and two,

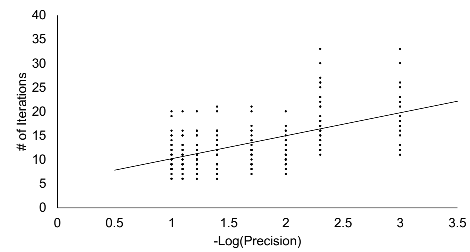
[1] https://github.com/qcol-lu/qipm
[2] NISQ: Noisy Intermediate Scale Quantum

respectively. The condition number of the coefficient matrix is set to two. The desired precision of the II-QIPM is equal to 0.1. It is worth noting that we set ω to 10.

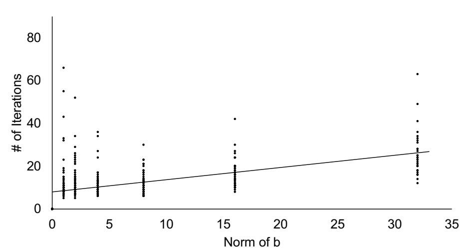
As illustrated in Figure 1a, the number of iterations quadratically increases by increasing the number of variables. As shown in Figure 1b, it also linearly increases by the negative of the logarithm of the desired precision. It almost linearly increases by the norm of the RHS vector, and ω (see Figures 1). As discussed in Section 4.2, the iterations are affected by the error of QLSAs by increasing the norm of the RHS vector, and ω as the error of QLSA increases by scaling. In addition, the larger ω is, the further away the starting point is from the optimal solution.



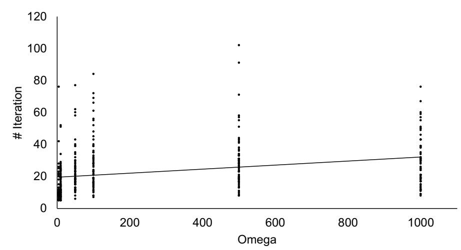
a) Effect of number of variables.



b) Effect of the desired precision.



c) Effect of norm of RHS vector.



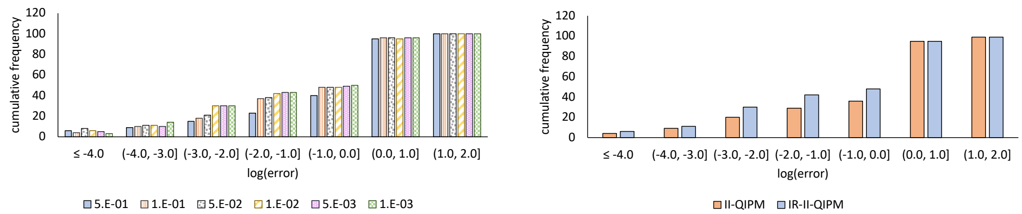
d) Effect of ω.

Fig. 1: Effect of different characteristics of a LO problem on the number of iterations.

7.2 Evaluation of the IR-II-QIPM

Even with smart parameter tuning and prefect implementation, the HHL simulator has limited precision. Here, we analyze the performance of the IR method where combined with the II-QIPM. One important parameter that highly affects the norm of the RHS vector and, consequently, the HHL simulator's error is ω. As shown in Figure 1d, as ω increases, the number of iterations of the II-QIPM increases as well. On the other hand, as the IR progresses, the norm of a solution of the LO subproblem increases. Thus, we need to set ω based on the desired precision. Here, we set ω to 1000 to avoid infeasibility report by the proposed II-QIPM.

Figure 2a shows the logarithmic error of IR-II-QIPM for different desired precisions. We set the LO precision to 0.01 to evaluate the effect of the IR on the proposed II-QIPM. Figure 2b illustrates that by using IR, we can reach higher precision. The final precision in both the pure II-QIPM and the IR-II-QIPM is set to 10^{-4}. However, we can still not reach the desired precision in half of the instances because of the QLSA's error.



a) Effect of the LO precision on the IR-II-QIPM.

b) Effect of iterative refinement on the II-QIPM.

Fig. 2: Obtained precision verses the desired precision on the IR-II-QIPM.

A part of speed up caused by Iterative Refinement is mitigating the condition number effect, since by early stopping QIPMs the condition number of Newton systems are bounded by constant $\mathcal{O}(\kappa_{\hat{A}})$. Figure 3 shows how condition number of the solved linear systems in IR-QIPM is bounded, although the condition number of solved linear systems in QIPM without IR goes to infinity.

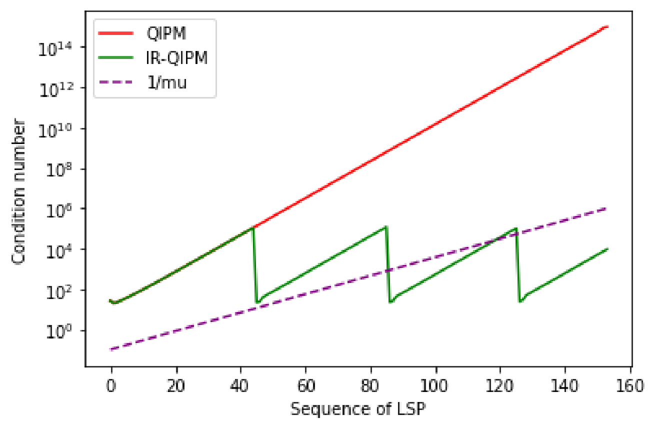


Fig. 3: The condition number of linear systems in QIPM and IR-QIPM to get 10^{-6}-precision solution for a primal-degenerate LO with 10 variables and 5 constraints

8 Conclusion

This paper analyzes in details the benefits and challenges when developing Quantum Interior Point Methods. Specifically, we analyze the use of QLSAs inside IPMs and present a convergent II-QIPM. Previous papers overlooked that when one use QLSA with QTA, the solution of Newton system is inexact and the Newton system's condition number, goes to infinity as IPMs approach the optimal set. Here, we also adapt an IR method to find an exact solution in polynomial time. After addressing issues in earlier QIPMs, we proved the correctness and convergence of the proposed II-QIPM and analyzed its performance, both theoretically and empirically.

Table 4 compares the best complexity results for IR-II-QIPM with two analogous classical II-IPMs and two QIPMs. The proposed IR-II-QIPM has polynomial complexity, while the other QIPMs cannot find an exact optimal solution in polynomial time. The exponential complexity

Algorithm	Simplified Complexity
II-IPM with Cholesky	$\mathcal{O}\Big(n^5 L\Big)$
II-IPM with CG	$\mathcal{O}\Big(n^4 L\kappa_A^2 2^{4L}\Big)$
QIPM of [22]	$\mathcal{O}\Big(n^2 L\kappa^3 2^{4L}\Big)$
QIPM of [8]	$\mathcal{O}\Big(n^2 L\kappa^2 2^{2L}\Big)$
Proposed IR-II-QIPM	$\tilde{\mathcal{O}}\Big(n^4 L\kappa_A^2 \|A\|_F\Big)$

Table 4: Time complexity of finding an exact solution

of those QIPMS are caused by QLSA's error and the increasing condition number of the Newton system. Classical II-IPMs with CG method face exponential complexity as well, and some classical papers used preconditioning techniques to reduce the condition number [e.g., 20]. For the first sight, one can get the impression that the other two QIPMs have better complexity, but these time complexities cannot be attained since they only contain the iteration complexity of exact IPMs while these QIPMs solve the Newton system inexactly. They also need appropriate bound for condition number κ and precision ϵ based on their setting of QIPMs. To correct the complexity of the QIPMs, at least $\mathcal{O}(n^{1.5})$ must be added for inexact Newton steps and an appropriate upper bound for QLSAs' error.

The complexity of the proposed IR-II-QIPM has better dependence on n than its classical counterparts and realistic complexity of its quantum counterparts. Still, the complexity of the proposed method depends on constants $\kappa_{\hat{A}}$, and $\|\hat{A}\|_F$. One may apply scaling and preconditioning techniques for LO problems with large $\kappa_{\hat{A}}$ to decrease $\kappa_{\hat{A}}$. In the literature of preconditioning the NES, [31] used the speculated optimal partition instead of predefined basis $(\hat{B}, \hat{N})$. The major problem of this approach is that the cost of calculating the precondition $A_{\hat{B}}^{-1}$ in each iteration will destroy the quantum speed up in QIPMs. A useful research direction is to explore how to mitigate the condition number and norm of Newton systems with a quantum-friendly approach.

The performance of QIPMs will be improved if faster QTAs and QLSAs are proposed. We can also investigate Inexact-Feasible IPMs that are more adaptable with QLSAs since Feasible IPMs have better complexity than infeasible IPMs. Another direction can be developing pure QIPMs in which all calculations happen in the quantum setting. However, there are some limitations of current NISQ devices that prevent to have a pure QIPM. Such a method would not need QTA inside and so could take advantage of fast QLSAs.

Our computational experiments show that the proposed II-QIPM embedded in the IR scheme and using the QLSA simulator of QISKIT AQUA can solve problems with hundreds of variables to a user-defined precision. However, there is a limitation in the number of constraints, since we can simulate only a limited number of Qubits by a classical computer. Although there are Feasible IPMs with better complexity than the proposed IR-II-QIPM, this paper is a significant step towards using quantum solvers in classical methods correctly and efficiently. We also demonstrated for the first time that LO problems could practically be solved using quantum solvers.

Acknowledgements This work is supported by Defense Advanced Research Projects Agency as part of the project W911NF2010022: *The Quantum Computing Revolution and Optimization: Challenges and Opportunities*.

References

[1] Erling D. Andersen, Cornelis Roos, Tamás Terlaky, Theodore Trafalis, and Joost P. Warners. "The use of low-rank updates in interior-point methods". *Numerical Linear Algebra and Optimization*. Science Press, Beijing, China, 2004, pp. 3–14.

[2] Venansius Baryamureeba and Trond Steihaug. "On the convergence of an inexact primal-dual interior point method for linear programming". *The Large-Scale Scientific Computing*. Ed. by Ivan Lirkov, Svetozar Margenov, and Jerzy Waśniewski. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 629–637. DOI: 10.1007/11666806_72.

[3] Stefania Bellavia. "Inexact interior-point method". *Journal of Optimization Theory and Applications* 96.1 (1998), pp. 109–121. DOI: 10.1023/A:1022663100715.

[4] Stefania Bellavia and Sandra Pieraccini. "Convergence analysis of an inexact infeasible interior point method for semidefinite programming". *Computational Optimization and Applications* 29.3 (2004), pp. 289–313.

[5] Fernando G.S.L. Brandao and Krysta M Svore. "Quantum speed-ups for solving semidefinite programs". *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2017, pp. 415–426.

[6] Fernando G. S. L. Brandão, Amir Kalev, Tongyang Li, Cedric Yen-Yu Lin, Krysta M. Svore, and Xiaodi Wu. "Quantum SDP solvers: Large speed-ups, optimality, and applications to quantum learning". *arXiv e-prints* (2017). URL: https://bit.ly/3qbCClo.

[7] Almudena Carrera Vazquez, Ralf Hiptmair, and Stefan Woerner. "Enhancing the quantum linear systems algorithm using Richardson extrapolation". *arXiv e-prints* (2020), arXiv–2009. URL: https://arxiv.org/abs/2009.04484.

[8] Pablo AM Casares and Miguel Angel Martin-Delgado. "A quantum interior-point predictor–corrector algorithm for linear programming". *Journal of physics A: Mathematical and Theoretical* 53.44 (2020), p. 445305.

[9] Shantanav Chakraborty, András Gilyén, and Stacey Jeffery. "The power of block-encoded matrix powers: Improved regression techniques via faster Hamiltonian simulation". *arXiv preprint* (2018). URL: https://arxiv.org/abs/1804.01973.

[10] Andrew M Childs. "On the relationship between continuous-and discrete-time quantum walk". *Communications in Mathematical Physics* 294.2 (2010), pp. 581–603.

[11] Andrew M. Childs, Robin Kothari, and Rolando D. Somma. "Quantum algorithm for systems of linear equations with exponentially improved dependence on precision". *SIAM Journal on Computing* 46.6 (Jan. 2017), pp. 1920–1950. DOI: 10.1137/16m1087072.

[12] George B. Dantzig. *Linear Programming and Extensions*. Princeton, NJ, USA: Princeton University Press, 1963.

[13] Danial Dervovic, Mark Herbster, Peter Mountney, Simone Severini, Naïri Usher, and Leonard Wossnig. "Quantum linear systems algorithms: a primer". *arXiv preprint* (2018). URL: https://arxiv.org/abs/1802.08227.

[14] David Deutsch. "Quantum theory, the Church-Turing principle and the universal quantum computer". *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 400.1818 (1985), pp. 97–117. DOI: 10.1098/rspa.1985.0070.

[15] David Deutsch and Richard Jozsa. "Rapid solution of problems by quantum computation". *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 439.1907 (1992), pp. 553–558. DOI: 10.1098/rspa.1992.0167.

[16] Roland W. Freund, Florian Jarre, and Shinji Mizuno. "Convergence of a class of inexact interior-point algorithms for linear programs". *Mathematics of Operations Research* 24.1 (1999), pp. 50–71. DOI: 10.1287/moor.24.1.50.

[17] Ambros Gleixner and Daniel E. Steffy. "Linear programming using limited-precision oracles". *Mathematical Programming* 183.1 (2020), pp. 525–554. DOI: 10.1007/s10107-019-01444-6.

[18] Ambros M. Gleixner, Daniel E. Steffy, and Kati Wolter. "Iterative refinement for linear programming". *INFORMS Journal on Computing* 28.3 (2016), pp. 449–464. DOI: 10.1287/ijoc.2016.0692.

[19] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. "Quantum algorithm for linear systems of equations". *Physical Review Letters* 103.15 (Oct. 2009). DOI: 10.1103/physrevlett.103.150502.

[20] Ghussoun Al-Jeiroudi and Jacek Gondzio. "Convergence analysis of the inexact infeasible interior-point method for linear optimization". *Journal of Optimization Theory and Applications* 141.2 (2009), pp. 231–247. DOI: 10.1007/s10957-008-9500-5.

[21] Narendra Karmarkar. "A new polynomial-time algorithm for linear programming". *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*. STOC '84. New York, NY, USA: Association for Computing Machinery, 1984, pp. 302–311. DOI: 10.1145/800057.808695.

[22] Iordanis Kerenidis and Anupam Prakash. "A quantum interior point method for LPs and SDPs". *ACM Transactions on Quantum Computing* 1.1 (2020), pp. 1–32. DOI: 10.1145/3406306.

[23] Iordanis Kerenidis and Anupam Prakash. "Quantum Recommendation Systems". *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*. Ed. by Christos H. Papadimitriou. Vol. 67. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017, 49:1–49:21. ISBN: 978-3-95977-029-3. DOI: 10.4230/LIPIcs.ITCS.2017.49. URL: http://drops.dagstuhl.de/opus/volltexte/2017/8154.

[24] Iordanis Kerenidis, Anupam Prakash, and Dániel Szilágyi. "Quantum algorithms for second-order cone programming and support vector machines". *Quantum* 5 (2021), p. 427. DOI: 10.22331/q-2021-04-08-427.

[25] Leonid Genrikhovich Khachiyan. "A polynomial algorithm in linear programming". *Doklady Akademii Nauk*. Vol. 244. 5. Russian Academy of Sciences. 1979, pp. 1093–1096. URL: http://mi.mathnet.ru/eng/dan42319.

[26] Masakazu Kojima, Nimrod Megiddo, and Shinji Mizuno. "A primal—dual infeasible-interior-point algorithm for linear programming". *Mathematical Programming* 61.1 (1993), pp. 263–280. DOI: 10.1007/BF01582151.

[27] János Korzak. "Convergence analysis of inexact infeasible-interior-point algorithms for solving linear programming problems". *SIAM Journal on Optimization* 11.1 (2000), pp. 133–148. DOI: 10.1137/S1052623497329993.

[28] Guang Hao Low and Isaac L Chuang. "Hamiltonian simulation by qubitization". *Quantum* 3 (2019), p. 163.

[29] Shinji Mizuno and Florian Jarre. "Global and polynomial-time convergence of an infeasible-interior-point algorithm using inexact computation". *Mathematical Programming* 84.1 (1999), pp. 105–122. DOI: 10.1007/s10107980020a.

[30] Mohammadhossein Mohammadisiahroudi, Ramin Fakhimi, Brandon Augustino, and Tamás Terlaky. "Generating Appropriate Linear, Semidefinite, and Second-order Cone Optimization Problems for Numerical Experiments". *Tech Rep ISE* (2021).

[31] Renato Monteiro and Jerome W. O'Neal. "Convergence analysis of a long-step primal-dual infeasible interior-point LP algorithm based on iterative linear solvers". *Georgia Institute of Technology* (2003). URL: https://bit.ly/3ss0kg1.

[32] Cornelis Roos, Tamás Terlaky, and Jean-Philippe Vial. *Interior Point Methods for Linear Optimization*. Springer Science & Business Media, 2005. DOI: 10.1007/b100325.

[33] Jack Sherman and Winifred J Morrison. "Adjustment of an inverse matrix corresponding to a change in one element of a given matrix". *The Annals of Mathematical Statistics* 21.1 (1950), pp. 124–127.

[34] Daniel R. Simon. "On the power of quantum computation". *SIAM Journal on Computing* 26.5 (1997), pp. 1474–1483. DOI: 10.1137/s0097539796298637.

[35] Tamás Terlaky. *Interior Point Methods of Mathematical Programming*. Vol. 5. Applied Optimization Kluwer Academic Publisher, 1996.

[36] Joran Van Apeldoorn, Arjan Cornelissen, and Giacomo Nannicini. *Optimal algorithms for quantum state tomography*. Tech. rep. In preparation. 2022.

[37] Leonard Wossnig, Zhikuan Zhao, and Anupam Prakash. "Quantum linear system algorithm for dense matrices". *Physical Review Letters* 120.5 (Jan. 2018). DOI: 10.1103/physrevlett.120.050502.

[38] Stephen J. Wright. *Primal-Dual Interior-Point Methods*. SIAM, 1997. DOI: 10.1137/1.9781611971453.

[39] Yinyu Ye. *Interior Point Algorithms: Theory and Analysis*. Vol. 44. John Wiley & Sons, 2011.

[40] Yinyu Ye, Michael J. Todd, and Shinji Mizuno. "An $O(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm". *Mathematics of Operations Research* 19.1 (1994), pp. 53–67. DOI: 10.1287/moor.19.1.53.

[41] Guanglu Zhou and Kim-Chuan Toh. "Polynomiality of an inexact infeasible interior point algorithm for semidefinite programming". *Mathematical Programming* 99.2 (2004), pp. 261–282. DOI: 10.1007/s10107-003-0431-5.