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

MOHAMMADHOSSEIN MOHAMMADISIAHROUDI¹, RMIN FAKHIMI¹, AND TAMÁS TERLAKY¹

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

ISE Technical Report 21T-005
Efficient Use of Quantum Linear System Algorithms in Interior Point Methods for Linear Optimization

Mohammadhossein Mohammadisiahroudi, Ramin Fakhimi and Tamás Terlaky

May 3, 2021

Abstract

Quantum computing has attracted significant interest due to its potential to solve some crucial mathematical problems much faster than conventional supercomputers. Since mathematical optimization problems yield one of the most important classes of problems with widespread applications in practice and complicated computational challenges, some researchers proposed quantum methods, especially Quantum Interior Point Methods (QIPMs), to solve some types of optimization problems such as Semi-definite Optimization, Linear Optimization, and Second-order Cone Optimization problems. Most of them have applied a Quantum Linear System Algorithm (QLSA) at each iteration of their algorithm. However, the use of QLSAs in QIPMs comes with many challenges, such as having ill-conditioned systems and the considerable error of QLSAs. This paper investigates how we can use QLSAs in QIPMs efficiently. Accordingly, an Inexact Infeasible QIPM is developed to solve linear optimization problems. We also discuss how we can get an exact solution by Iterative Refinement without excessive time of QLSAs. Finally, the results of implementing our quantum method using quantum simulators are analyzed.

1 Introduction

Linear Optimization (LO) is defined as optimizing a linear function over a set of linear constraints. Several algorithms were developed to solve LO problems. In the first decades, the primary solution methodology was Dantzing’s Simplex method (Dantzig, 1990). It has been shown by Klee and Minty, 1972 that Simplex algorithms have an exponential time complexity. However, Simplex algorithms solve LO problems efficiently in practice.

Khachiyan, 1979 proposed the Ellipsoid method, the first polynomial-time method to solve LO problems with integer data. However, the Ellipsoid method was not as efficient as its counterparts. Karmarkar, 1984 developed a practically efficient Interior Point Method (IPM) for solving LO problems with polynomial time complexity. Since Karmarkar’s publication, a large class of theoretically and practically efficient IPMs were developed, see e.g., (Wright, 1997; Roos, Terlaky, and Vial, 2005; Terlaky, 2013). Contrary to the Simplex method, a feasible IPM reaches an optimal solution by traversing the interior of the feasible region (Roos, Terlaky, and Vial, 2005).
Contemporary IPMs reach an optimal solution by starting from an interior point and following the central path (Roos, Terlaky, and Vial, 2005). Most of IPMs are primal-dual methods meaning they strive to satisfy the optimality conditions maintaining primal and dual feasibility. It should be noted that basic IPM needs an initial feasible interior point. Some current commercial solvers apply Feasible IPMs (F-IPMs) on the self-dual embedding formulation of LO problems where we know that all-one vector is a feasible interior solution for this formulation (Roos, Terlaky, and Vial, 2005). On the other hand, Infeasible Interior Point Methods (I-IPMs) can start with an infeasible but positive solution. Theoretical analysis shows the best time complexity of F-IPMs is $O(\sqrt{n}L)$ where $n$ is the number of variables and $L$ is the binary length of the input data. On the other hand, the best time complexity of I-IPMs is $O(nL)$. In practice, the performance of both feasible and infeasible IPM are similar and I-IPM has better performance on some problems contrary to theoretical analysis (Wright, 1997).

At each iteration of IPMs, one solves a Linear Equation System to calculate Newton direction. There are three choices: the full Newton System, Augmented System, and Normal Equation System (NES) which will be discussed in Section 7. In classical computers, a prevailing approach is applying Cholesky Factorization to solve the NES because it has a symmetric positive definite coefficient matrix. Since Cholesky Factorization has $O(n^3)$ time complexity, many researchers studied solution methods for the NES in IPMs theoretically and empirically to improve the total time of IPMs (for more details see Wright, 1997).

More recently, several papers were published on the subject of I-IPMs using inexact search directions. First, Mizuno and his colleagues did a series of research on convergence of Inexact Infeasible IPMs (II-IPMs) (Mizuno and Jarre, 1999; Freund, Jarre, and Mizuno, 1999). The convergence of I-IPM by Kojima, Megiddo, and Mizuno, 1993, known as KMM method, using inexact Newton step was proved by Korzak, 2000 and Baryamureeba and Steihaug, 2005. Korzak, 2000 also showed that the time complexity of this algorithm is polynomial. The usage of Preconditioned Conjugate Gradient (PCG) method in II-IPM was also studied by Al-Jeiroudi and Gondzio, 2009; Monteiro and O’Neal, 2003. Bellavia, 1998 studied the convergence of II-IPM for general convex optimization and the convergence of II-IPM for Semi-definite Optimization (SDO) problems has been proved by Zhou and Toh, 2004. The best time complexity of II-IPMs is $O(n^2L)$ which is more than exact feasible and infeasible IPMs.

Quantum computers have recently emerged as a powerful alternative to classic computers (Childs, Kothari, and Somma, 2017). Starting from Deutsch’s Problem, a series of problems and algorithms have demonstrated theoretically exponential speed-up compared to their classical counterparts (Deutsch, 1985; Deutsch and Jozsa, 1992; Simon, 1997). Having such a powerful tool, it is natural to investigate quantum optimization algorithms. Generally speaking, we can categorize existing quantum continuous optimization methods into QIPMs and non-interior-point methods. Kerenidis and his colleagues presented a series of papers on QIPMs for solving LO, SDO (Kerenidis and Prakash, 2020), and Second-order Cone Optimization (SOCO) problems (Kerenidis, Prakash, and Szilágyi, 2019). Also, Casares and Martin-Delgado, 2020 provide their
scheme of a predictor-corrector QIPM for LO problems. On the non-interior-point side, Brandão and his colleagues proposed their multiplicative-weight-based quantum algorithm and following updates for the SDO problems (Brandão and Svore, 2016; Brandão et al., 2017).

Specifically for LO problems, only two papers propose QIPM (Casares and Martin-Delgado, 2020; Kerenidis and Prakash, 2020). First, Kerenidis and Prakash, 2020 have developed a QIPM for SDO and LO problems. They asserted that their algorithm has $O\left(\frac{n^3 \kappa^3 \log(1/\zeta)}{\epsilon^2}\right)$ worst case running time for SDOs and $O\left(\frac{n^2 \kappa^3 \log(1/\epsilon)}{\zeta^2}\right)$ for LOs where $\zeta$ is the final optimality gap, $\epsilon$ is the precision of the QLSA, Further, $\kappa$ is an upper bound for the condition number of the Newton system. This result indicates polynomial improvement in dependence on the number of variables $n$ over the best classical solvers, while it suffers from high dependence on the condition number and the error. The main idea of this algorithm is to enhance a general F-IPM with building the Newton system in a quantum computer using Block Encoding, solving it by a QLSA, and extracting the classical solution by a Quantum Tomography Algorithm (QTA). Although this paper introduced the great idea of hybrid classical-quantum IPMs, there are some crucial questions about this paper. On of them is that they did not use symmetrization in their algorithm, which is needed for SDO in IPMs. Augustino et al., 2021 has addressed this issue and also used an II-QIPM to solve SDOs. The symmetrization is not an issue for LO problems since diagonal matrices are always symmetric.

On the other hand, Casares and Martin-Delgado, 2020 proposed a predictor-corrector QIPM to solve LO problems. It is also a hybrid classical-quantum algorithm. The corresponding classical algorithm is the well-known predictor-corrector method proposed by Ye, Todd, and Mizuno, 1994. In each step, Casares and Martin-Delgado, 2020 used the QLSA proposed by Chakraborty, Gilyén, and Jeffery, 2018 to solve the Newton systems. The worst-case complexity of their method is $O(L\sqrt{n}(n+m)\|M\|_F\epsilon^2\kappa^2)$, where $m$ is the number of constraints, and $\|M\|_F$ is an upper bound to the Forbenius norm of the coefficient matrix in the Newton system. The authors claimed a quantum speedup with respect to the dimension $n$ compared to comparable classical algorithms. One major concern about this paper is that the authors ignored the fact that a QTA is needed to extract the classical solution from a quantum state in hybrid methods. Thus, we will have more dependence on $n$ if we add the time complexity of the QTA to their total time complexity.

In both mentioned QIPMs, Exact Feasible IPM framework was used although QLSAs deliver inexact solutions inherently. Due to this fact, the convergence of these QIPMs are questionable and their time complexities are not attainable since inexact IPMs have more time complexity. Another challenge for QIPMs using QLSA is that their time complexities depend on the condition number of the Newton system and precision of QLSAs. We know that in IPMs, the condition number of the Newton system goes to infinity as the algorithm approaches an optimal solution (Roos, Terlaky, and Vial, 2005). Also, we need the error to be less than $2^{-2L}$ to reach an exact solution as discussed in Corollary 1. Thus, the time complexities of these methods are not polynomial anymore in the classical sense. In this paper, these challenges arising from using QLSAs in IPMs
are addressed to develop a convergent II-QIPM.

The HHL algorithm was the first QLSA that showed quantum advantage with respect to the dimension compared with classical linear system solvers (Harrow, Hassidim, and Lloyd, 2009). However, this method has unfavorable dependence on the condition number, error, and sparsity of the matrix. Many researches have invested in improving the performance of QLSAs, (Wossnig, Zhao, and Prakash, 2018; Childs, Kothari, and Somma, 2017). In the next section, we discuss the QLSAs and their performance in more detail. In essence, using QLSA comes with challenges such as addressing error of QLSAs and the huge condition number of the Newton system. This paper explores an efficient use of the QLSAs for solving the Newton system at each iteration of IPMs. To reach this goal, an II-QIPM is developed to solve LO problems since QLSAs are not efficient for finding precise solutions. Also, we use an iterative refinement scheme to avoid exponential time complexity in finding an exact optimal solution. This paper is structured as follow:

• Section 2: studying the performance of several QLSAs and how we can use them to solve classical linear systems.
• Section 3: discussing the LO problem and its characteristics.
• Section 4: developing an II-QIPM to solve LO problems, in which QLSA used for solving the NES, and proving the convergence of the proposed II-QIPM.
• Section 5: discussing how we can deal with density of the NES using the structure of input data.
• Section 6: using an Iterative Refinement scheme to find an exact optimal solution of a LO problem without excessive time of QLSAs.
• Section 7: comparing different Newton systems for using in the proposed II-QIPM.
• Section 8: evaluating the performance of the proposed II-QIPM with Iterative Refinement (IR-II-QIPM) through computational experiments.

2 Quantum Linear Algebra

In this section, we want to study how we can use Quantum Computing to solve Linear System problems (LSP) as defined in Definition 1.

**Definition 1** (LSP). *Linear equation system problem is to find* \( z \in \mathbb{R}^p \) *such that* \( Mz = \sigma \) *where* \( M \) *is a matrix with* \( p \) *columns and* \( q \) *rows, and* \( \sigma \in \mathbb{R}^q \).

From linear algebra, the LSP can have one solution, many or no solutions. If \( M \) is non-singular, there exists a unique solution. If \( M \) is singular and \( \sigma \in \text{span}(M) \), the LSP has infinitely many solutions. If \( M \) is singular and \( \sigma \notin \text{span}(M) \), there is no solution for the LSP. In classical computers, one simple solution is finding \( A^{-1} \) but it is shown
that finding $A^{-1}$ has more numerical difficulties than solving a linear equation system. We can also use Cramer's rule and we need to find the determinant of sub matrices which is as expensive as solving linear systems. There are some iterative algorithms for solving such systems numerically better than Cramer's rule or finding $A^{-1}$. The basic approach is Gaussian elimination with $O(p^3)$ time complexity. We can use LU factorization which is similar to Gaussian elimination with $O(p^3)$ complexity. If $M$ is positive semi-definite, we can use Cholesky factorization with $O(p^3)$ complexity. It should be mentioned that Cholesky factorization needs $\frac{1}{3}p^3$ arithmetic operations but LU factorization and Gaussian elimination need $\frac{2}{3}p^3$ arithmetic operations. The best time complexity with respect to $p$ is $O(p^3 \log(1/\epsilon))$ for Conjugate Gradient method where $d$ is the maximum number of non-zero elements in any rows or columns of $M$, $\kappa$ is the condition number of $M$, and $\epsilon$ is the target error.

Solving the linear systems in a quantum computer was first realized by Harrow, Hassidim, and Lloyd, 2009. After this paper, many other papers were published to improve the HHL method. All of these methods solve the Quantum Linear System Problem (QLSP) defined in Definition 2.

**Definition 2 (QLSP).** Let $M$ be a $p$-by-$p$ Hermitian matrix with known condition number $\kappa$, $\|M\|_2 = 1$, and at most $d$ non-zero entries in any rows or columns. Let $|\sigma\rangle$ be a $p$-dimensional vector. A quantum linear system has form $M |z\rangle = |\sigma\rangle$. We define the quantum states $|\sigma\rangle$ and $|z\rangle$ as

$$
|\sigma\rangle = \frac{\sum_{i=1}^{p} \sigma_i |i\rangle}{\|\sigma\|_2}, \quad \text{and} \quad |z\rangle = \frac{\sum_{i=1}^{p} z_i |i\rangle}{\|z\|_2}.
$$

**2.1 Relation between QLSP and LSP**

Based on Definition 1 and Definition 2, QLSP is different form LSP. In IPMs and other practical usage, we have an LSP but QLSAs can solve QLSPs. Thus, we need to translate LSP to QLSP, solve the QLSP by QLSA and convert back the solution. Some people wrongly used the result of QLSAs for solving LSP without considering this translation. Here, we want to analyze this translation and its cost. The following steps show a detailed procedure for translating an LSP to a QLPS and finding the classical solution.

1. Based on the definition of QLSP, the matrix of the system must be Hermitian. If $M$ is not Hermitian, one can construct $\bar{M} \bar{|z\rangle} = |\bar{\sigma}\rangle$ where

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

and find the output $\bar{z} = \begin{pmatrix} 0 \\ z \end{pmatrix}$ by QLSA. The size of problem increases from $p$ to $p + q$ by this step.
2. The QLSP needs $\|M\|_2 = 1$. For general Hermitian matrices, one can construct

$$\bar{M} = \frac{M}{\|M\|_2}, \quad \bar{\sigma} = \frac{\sigma}{\|M\|_2}.$$ 

We also see this scaling in the definitions of states $|z\rangle$ and $|\sigma\rangle$. This scaling is important because it will affect the target precision. Let $|\tilde{z}\rangle$ be the inexact solution of the QLSP such that $\| |\tilde{z}\rangle - |z\rangle \|_2 \leq \epsilon_{QLSP}$. To extract the solution of LSP $\tilde{z}$, we have $\tilde{z} = \|\bar{\sigma}\|_2 |\tilde{z}\rangle$ and the error for the solution of the LSP will be

$$\| \tilde{z} - z \|_2 = \|\bar{\sigma}\|_2 \| |\tilde{z}\rangle - |z\rangle \|_2 \leq \frac{\|\bar{\sigma}\|_2}{\|M\|_2} \epsilon_{QLSP}.$$ 

Thus, the scaling changes the error of a solution. If we want to have error bound $\epsilon$ for solution of LSP, then we must fix the target error of the QLSP as

$$\epsilon_{QLSP} = \frac{\|M\|_2}{\|\bar{\sigma}\|_2} \epsilon.$$ 

3. To implement QLSAs, we need access to a procedure $\mathcal{P}_M$ that computes entries of $M$ and a procedure $\mathcal{P}_\sigma$ that prepares the state $|\sigma\rangle$. Fortunately, there are $\mathcal{P}_M$ and $\mathcal{P}_\sigma$ with $\mathcal{O}(\text{polylog}(p))$ complexity (Chakraborty, Gilyén, and Jeffery, 2018). If the complexity of these procedures was higher than this, we would add the complexity of $\mathcal{P}_M$ and $\mathcal{P}_\sigma$ to the total complexity. Now, the time complexity of these procedures are negligible compared to the time complexity of QLSAs.

4. As we can see in Figure 1, QLSAs can provide a quantum state proportional to the solution. By a single measurement, we can not extract the classical solution. To extract the classical solution, we need Quantum Tomography Algorithms (QTAs). There are many versions of QTAs. Here, we use QTA by Kerenidis and Prakash, 2020. To have $\theta$-close solution, this QTA runs QLSA for $\mathcal{O}\left(\frac{\text{polylog}(p)\|M\|_2^2}{\theta^2}\right)$ times.

2.2 QLSAs

As we can see in Figure (1), QLSAs can solve QLSPs by applying some gates on $|\sigma\rangle$ to reach $|z\rangle$. The HHL algorithm (Harrow, Hassidim, and Lloyd, 2009) was the first algorithm for solving QLSPs with $\mathcal{O}(\text{log}(p)\frac{d^2\kappa^2}{\epsilon})$ complexity. The core idea of this method is using Hamiltonian simulation techniques to apply the unitary operator $e^{iMt}$ to $|\sigma\rangle$ for a superposition of different times $t$ and decomposing $|\sigma\rangle$ into the eigenbasis of $M$ to find the corresponding eigenvalues by quantum phase estimation. Many researchers attempted to improve the performance of the HHL algorithm. In the first attempt, Amplitude Amplification can decrease the dependence on $\kappa^2$ to $\kappa$. Wossnig, Zhao, and Prakash, 2018 proposed a QLSA independent on sparsity with $\mathcal{O}\left(\text{polylog}(p)\|M\|_2^2\right)$ complexity by using Quantum Singular Value Estimation. In another direction, Childs,
Kothari, and Somma, 2017 developed QLSAs with exponentially better dependence on error with $O(\text{polylog}(\frac{p\kappa}{\epsilon})d\kappa)$ complexity. They proposed two approaches using Fourier and Chebyshev series representations. Carrera Vazquez, Hiptmair, and Woerner, 2020 enhanced the HHL algorithm by Richardson Extrapolation in a Hamiltonian simulation with $O(\text{polylog}(\frac{p\kappa}{\epsilon})\kappa)$ time complexity. They also implemented their method on QISKIT simulator which is better than HHL implementation. The best QLSA with respect to time complexity uses Block Encoding and QRAM with $O(\text{polylog}(\frac{p}{\epsilon})\kappa)$ complexity (Chakraborty, Gilyén, and Jeffery, 2018). The details of these methods are not explained in this paper but Dervovic et al., 2018 reviewed most of these QLSAs in detail.

![QLSA Circuit](image)

Figure 1: The QLSA in a circuit

As we can see in Table 1, QLSAs have better dependence on the dimension $p$ compared with classical solver but worse dependence on the condition number $\kappa$, and the error $\epsilon$ when we apply them to QLSPs. To have a classical solution, we need more time since we must also apply QTAs. QLSAs have better dependence on the dimension $p$ compared with factorization and elimination techniques but worse dependence on the condition number $\kappa$. On the other hand, their time complexity is roughly similar to Conjugate Gradient (CG) method with respect to $p$ and some QLSAs have better dependence on sparsity than CG method. In the following section, we will discuss how we can deal with error and condition number to use QLSA in IPMs.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>QLSA</th>
<th>LSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factorization methods (e.g. Cholesky)</td>
<td>(\mathcal{O}(p^3))</td>
<td>(\mathcal{O}(p^3))</td>
</tr>
<tr>
<td>Conjugate Gradient</td>
<td>(\mathcal{O}(p\kappa\log(1/\epsilon)))</td>
<td>(\mathcal{O}(p\kappa\log(1/\epsilon)))</td>
</tr>
<tr>
<td>Fast sparse solver (Peng and Vempala, 2021)</td>
<td>(\mathcal{O}(p\kappa\log(1/\epsilon)))</td>
<td>(\mathcal{O}(p\kappa\log(1/\epsilon)))</td>
</tr>
<tr>
<td>HHL (Harrow, Hassidim, and Lloyd, 2009)</td>
<td>(\mathcal{O}(p\kappa\log(1/\epsilon)))</td>
<td>(\mathcal{O}(p\kappa\log(1/\epsilon)))</td>
</tr>
<tr>
<td>VTAA-HHL (Childs, Kothari, and Somma, 2017)</td>
<td>(\mathcal{O}(p\kappa\log(1/\epsilon)))</td>
<td>(\mathcal{O}(p\kappa\log(1/\epsilon)))</td>
</tr>
<tr>
<td>QLSA (Wossnig, Zhao, and Prakash, 2018)</td>
<td>(\mathcal{O}(p\kappa\log(1/\epsilon)))</td>
<td>(\mathcal{O}(p\kappa\log(1/\epsilon)))</td>
</tr>
<tr>
<td>QLSA (Childs, Kothari, and Somma, 2017)</td>
<td>(\mathcal{O}(p\kappa\log(1/\epsilon)))</td>
<td>(\mathcal{O}(p\kappa\log(1/\epsilon)))</td>
</tr>
<tr>
<td>QLSA (Carrera Vazquez, Hiptmair, and Woerner, 2020)</td>
<td>(\mathcal{O}(p\kappa\log(1/\epsilon)))</td>
<td>(\mathcal{O}(p\kappa\log(1/\epsilon)))</td>
</tr>
<tr>
<td>QLSA (Chakraborty, Gilyén, and Jeffery, 2018)</td>
<td>(\mathcal{O}(p\kappa\log(1/\epsilon)))</td>
<td>(\mathcal{O}(p\kappa\log(1/\epsilon)))</td>
</tr>
</tbody>
</table>
3 Problem Definition

Here, we use the standard form of Linear Optimization (LO) problems as

\[
\begin{align*}
\text{(P)} \quad & \min c^T x \\
\text{s.t.} \quad & Ax = b, \\
& x \geq 0,
\end{align*}
\]

\[
\begin{align*}
\text{(D)} \quad & \max b^T y \\
\text{s.t.} \quad & A^T y + s = c, \quad s \geq 0,
\end{align*}
\]

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

\[
\mathcal{PD}^* = \{ (x,y,s) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n | Ax = b, \ A^T y + s = c, \ x^T s = 0, \ (x,s) \geq 0 \}.
\]

Let \( \mu = \frac{x^T s}{n} \) then the central path indicates the line where \( x_i s_i = \mu \) for any \( i \in \{1, \ldots, n\} \).

**Definition 3.** Let \( \zeta \geq 0 \), we define the set of \( \zeta \)-optimal solution as

\[
\mathcal{PD}_\zeta = \{ (x,y,s) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n | ||Ax - b||_2 \leq \zeta, \ ||A^T y + s - c||_2 \leq \zeta, \ \mu \leq \zeta, \ (x,s) \geq 0 \}.
\]

Now, for any \( \gamma > 0 \), we can define an infeasible neighborhood of the central path (Kojima, Megiddo, and Mizuno, 1993) as

\[
\mathcal{N}(\gamma, \zeta) = \{ (x,y,s) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n | x \geq 0, \ s \geq 0, \ x_i s_i \geq \gamma \mu \text{ for } i \in \{1, \ldots, n\}, \ \mu \geq \gamma ||Ax - b||_2 \text{ or } ||Ax - b||_2 \leq \zeta, \ \mu \geq \gamma ||A^T y + s - c||_2 \text{ or } ||A^T y + s - c||_2 \leq \zeta \}.
\]

By Theorem 1, we can have a condition indicating how a point in the neighborhood of the central path can be a \( \zeta \)-optimal solution.

**Theorem 1.** For any \( (x,y,s) \in \mathcal{N}(\gamma, \zeta) \), if \( \mu \leq \zeta \gamma \), then \( (x,y,s) \) is a \( \zeta \)-optimal solution.

**Proof.** One can easily verify that

\[
\begin{align*}
(x,s) & \geq 0, \\
\zeta & \geq \gamma \zeta \geq \mu, \\
\gamma \zeta & \geq \mu \geq \gamma ||Ax - b||_2 \text{ or } ||Ax - b||_2 \leq \zeta, \\
\gamma \zeta & \geq \mu \geq \gamma ||A^T y + s - c||_2 \text{ or } ||A^T y + s - c||_2 \leq \zeta.
\end{align*}
\]

Thus, \( (x,y,s) \) is a \( \zeta \)-optimal solution.

We show the binary length of 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 1 comes from Lemma 3.1 from (Wright, 1997) and Corollary 1 is its immediate result.
Lemma 1. For any \((x^*, y^*, s^*) \in PD^*\), we have

\[
\min_i \{x^*_i | x^*_i > 0\} \geq 2^{-L},
\]
\[
\min_i \{s^*_i | s^*_i > 0\} \geq 2^{-L},
\]
\[
\max_i \{x^*_i\} \leq 2L,
\]
\[
\max_i \{s^*_i\} \leq 2L.
\]

Corollary 1. The exact optimal solution can be attained by rounding the solution \((x, y, s)\) if

\[\mu \leq 2^{-2L}.\]

In the following, we assume that the set \(PD \neq \emptyset\) and there exists \(\omega > 1\) such that \(||(x^*, s^*)||_\infty \leq \omega\) for any \((x^*, y^*, s^*) \in PD\). Based on Lemma 1, a trivial bound is \(\omega = 2L\).

4 Inexact Infeasible Quantum IPM

To speed up IPMs, we want to 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, Megiddo, and Mizuno, 1993 with the inexact Newton steps calculated by a QLSA. Let \(\mu_k = \frac{x^T_k s_k}{n}\) and \(0 < \beta_1 < 1\) be the centering parameter then the Newton system is defined as

\[
A \Delta x_k = b - Ax_k,
\]
\[
A^T \Delta y_k + \Delta s_k = c - A^T y_k - s_k,
\]
\[
X_k \Delta s_k + S_k \Delta x_k = \beta_1 \mu_k e - X_k s_k,
\]

where \(X_k\) and \(S_k\) are diagonal matrices of \(x_k\) and \(s_k\), respectively. We can solve the Augmented system or Normal Equation System (NES) instead of solving the full Newton system. In the following, we use the NES and discuss the effect of different systems on the results later in Section 7. From the Newton system (2), the NES is formulated as

\[
M_k \Delta y_k = \sigma_k,
\]

where \(M_k = A X_k S_k^{-1} A^T\) and \(\sigma_k = A X_k S_k^{-1} c - A X_k S_k^{-1} A^T y_k - \beta_1 \mu k A S_k^{-1} e + b - A x_k\).

As we can see, the NES has a smaller size, \(m\), than the full Newton system. On the other hand, the matrix of NES is symmetric, positive definite, and hermitian, enabling QLSAs to solve it efficiently. After finding \(\Delta y_k\) inexactly by solving the NES with QLSA, we can find \(\Delta x_k\) and \(\Delta s_k\) by

\[
\Delta s_k = c - A^T y_k - s_k - A^T \Delta y_k,
\]
\[
\Delta x_k = \beta_1 \mu_k S_k^{-1} e - x_k - S_k^{-1} X_k \Delta s_k.
\]
By its nature, QLSA generates an inexact solution $\tilde{\Delta}y_k$ with error bound $\|\Delta y_k - \tilde{\Delta}y_k\|_2 \leq \epsilon_k$. This error leads to residual $r_k$ as

$$M_k \Delta y_k = \sigma_k + r_k,$$

where $r_k = M_k(\tilde{\Delta}y_k - \Delta y_k)$. As $\tilde{\Delta}s_k$ and $\tilde{\Delta}x_k$ are directly calculated by equations (4), $(\tilde{\Delta}x_k, \tilde{\Delta}s_k, \tilde{\Delta}y_k)$ satisfies

$$A\tilde{\Delta}s_k = b - Ax_k + r'_k, \quad A^T\tilde{\Delta}y_k + \tilde{\Delta}s_k = c - A^T y_k - s_k, \quad X_k\tilde{\Delta}s_k + S_k\tilde{\Delta}x_k = \mu_k e - X_k s_k.$$ (5)

We can show that $r'_k = r_k$ as follows.

$$A\tilde{x}_k = A(\Delta x_k + \mu_k S_k^{-1} e - S_k^{-1} \Delta y_k),$$

$$= A(\Delta x_k + \mu_k S_k^{-1} e - S_k^{-1} X_k (c - s_k - A^T y_k - A^T \Delta y_k)),$$

$$= -\Delta x_k + \mu_k S_k^{-1} e - S_k^{-1} X_k c + S_k^{-1} X_k s_k + A S_k^{-1} X_k \Delta y_k + A S_k^{-1} X_k A^T \Delta y_k,$$

$$= \mu_k S_k^{-1} e - S_k^{-1} X_k c + A S_k^{-1} X_k A^T y_k + A X_k S_k^{-1} c - A X_k S_k^{-1} A^T y_k - \mu_k A S_k^{-1} e - b - Ax_k + r_k,$$

$$= b - \Delta x_k + r_k.$$

**Assumption 1.** Let assume in all iterations of II-QIPM, we have

$$\|r_k\|_2 \leq \eta \mu_k,$$

where $\eta$ is an enforcing parameter with $0 \leq \eta \leq 1$.

In the following, we show that the II-QIPM is convergent under Assumption 1. To do so, we need to find an appropriate $\epsilon_k$ error bound for QLSA such that Assumption (1) is satisfied. We have

$$\|r_k\|_2 = \|M_k \Delta y_k - M_k \tilde{\Delta}y_k\|_2 \leq \|M_k\|_2 \cdot \|\Delta y_k - \tilde{\Delta}y_k\|_2 \leq \|M_k\|_2 \epsilon_k.$$ (6)

So, we need

$$\|r_k\|_2 \leq \|M_k\|_2 \epsilon_k \leq \eta \mu_k.$$ (6)

Thus, the error of the QLSA must be bounded as inequality (6) to satisfy Assumption 1.

$$\epsilon_k \leq \eta \frac{\mu_k}{\|M_k\|_2}$$ (6)

The following Theorem is obtained from results of QLSA by Chakraborty, Gilyén, and Jeffery, 2018. We can also use other QLSAs discussed in Section 2, leading to different time complexities.

**Theorem 2.** Given the NES $M_k \Delta y_k = \sigma_k$, then QLSA and QTA can produce the solution $\tilde{\Delta}y_k$ with residual $r_k$ such that $\|r_k\|_2 \leq \eta \mu_k$. Thus, the worst case time complexity is $O(m \text{polylog}(\frac{\|\sigma_k\|_2}{\mu_k} \cdot \frac{\|r_k\|_2}{\mu_k} \cdot \|\sigma_k\|_2^2))$. 

10
Proof. First we need to find the target precision for QSLA and QTA. To have $\|r_k\| \leq \eta \mu_k$, we need

$$
\epsilon_{LSP} \leq \eta \frac{\mu_k}{\|M_k\|_2},
$$
$$
\epsilon_{QLSP} \leq \eta \frac{\mu_k}{\|\sigma_k\|_2},
$$
$$
\epsilon_{QLSA} \leq \eta \frac{\mu_k}{2\|\sigma_k\|_2},
$$
$$
\epsilon_{QTA} \leq \eta \frac{\mu_k}{2\|\sigma_k\|_2}.
$$

With this target precision, QLSA by Chakraborty, Gilyén, and Jeffery, 2018 and QTA by Kerenidis and Prakash, 2020 have totally $O\left(m \text{polylog}\left(\frac{m}{\mu_k}\right) \frac{\alpha_k\|\sigma_k\|_2^2}{\mu_k^2}\right)$ time complexity.

Algorithm 1 is the II-QIPM for solving LO problems using QLSA and QTA to solve the NES.

**Algorithm 1** II-QIPM using the NES

1: Choose $\zeta > 0$, $\gamma \in (0,1), 0 < \beta_1 < \beta_2 < 1$, $\eta \in (0,1)$ satisfying (7).
2: Choose $\omega \geq \max\{1, \|x^*, s^*\|_\infty\}$.
3: $k \leftarrow 0$
4: $(x_0, y_0, s_0) \leftarrow (\omega e, 0e, \omega e) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ with $e$ as a all-one vector
5: while $(x_k, y_k, s_k) \notin \mathcal{P}_\zeta$ do
6: $\mu_k \leftarrow \frac{x_k^T s_k}{n}$
7: $(M_k, \sigma_k) \leftarrow \text{build NES}(\mu_k)$ as Equation (3)
8: $\epsilon_{QLSA_k} \leftarrow \eta \frac{\mu_k}{\|\sigma_k\|_2}$
9: $\epsilon_{QTA_k} \leftarrow \eta \frac{\mu_k}{\|\sigma_k\|_2}$
10: $(\Delta x_k, \Delta y_k, \Delta s_k) \leftarrow \text{solve NES}(\mu_k)$ by QLSA+QTA with $\epsilon_{QLSA_k}$ and $\epsilon_{QTA_k}$
11: $\alpha_k^* \leftarrow \max\{\alpha \in \mathbb{R} \mid x_k + \alpha \Delta x_k \geq 0 \text{ and } s_k + \alpha \Delta s_k \geq 0\}$
12: if $(x_k, y_k, s_k) + \alpha_k^*(\Delta x_k, \Delta y_k, \Delta s_k) \in \mathcal{P}_\zeta$ then
13: break
14: $\hat{\alpha}_k \leftarrow \max\{\hat{\alpha} \in [0, 1] \mid \text{for any } \alpha \in [0, \hat{\alpha}] \text{ we have}
15: \{(x_k, y_k, s_k) + \alpha (\Delta x_k, \Delta y_k, \Delta s_k) \in N(\gamma, \zeta) \text{ and}
16: (x_k + \alpha \Delta x_k)^T (s_k + \alpha \Delta s_k) \leq (1 - \alpha(1 - \beta_2)) x_k^T s_k\}\}
17: $(x_{k+1}, y_{k+1}, s_{k+1}) \leftarrow (x_k, y_k, s_k) + \hat{\alpha}_k (\Delta x_k, \Delta y_k, \Delta s_k)$
18: if $\|x_{k+1}, s_{k+1}\|_\infty \geq \omega$ then
19: return Problem is infeasible.
20: $k \leftarrow k + 1$
21: return $(x_k, y_k, s_k)$
Remark 1. In Theorem 3, we show that there exists αL such that for any αk ∈[αL, ˆαk], the proposed II-QIPM converges to an optimal solution. So, we can use αk instead of ˆαk in Line (15) of the proposed II-QIPM.

We need to give appropriate values to parameters (β1, β2, η, γ) to have

\[
\begin{align*}
\delta_1 &= \frac{(1 - \gamma)\beta_1}{n} > 0, \\
\delta_2 &= \beta_1 - \eta\gamma > 0, \\
\delta_3 &= \beta_2 - \beta_1 > 0.
\end{align*}
\]

(7)

It can be easily verified that e.g., β1 = 0.5, β2 = 0.9995, η = 0.5 and γ = 0.5 yield a valid choice. In the following, we prove the convergence of Algorithm 1. To do so, we use δ1, δ2, δ3 as defined in (7). The parameter ω is hard to compute as described in Algorithm 1. We can use ω = 2L from Lemma 1, but in practice we can find a smaller bound.

4.1 Convergence of II-QIPM

First, we study basic properties of II-QIPM in Lemma 2 and Lemma 3. Then Lemma 4 shows that \{ ˆαk \} sequence is always strictly positive for all k. The convergence of II-QIPM is proved in Theorem 3 and an upper bound for the maximum number of iterations is provided in Theorem 4. In the proof, we need to analyze values of α such that the Newton step satisfies conditions in line (14) at iteration k of Algorithm 1. For the ease of notation, we define the following functions

\[
\begin{align*}
f_k^i(\alpha) &= (x_k^i + \alpha\Delta x_k^i)(s_k^i + \alpha\Delta s_k^i) - \gamma\frac{(x_k + \alpha\Delta x_k)^T(s_k + \alpha\Delta s_k)}{n} \quad \text{for } i \in \{1, 2, \ldots, n\}, \\
g_k^P(\alpha) &= \frac{(x_k + \alpha\Delta x_k)^T(s_k + \alpha\Delta s_k)}{n} - \gamma\|A(x_k + \alpha\Delta x_k) - b\|_2, \\
g_k^D(\alpha) &= \frac{(x_k + \alpha\Delta x_k)^T(s_k + \alpha\Delta s_k)}{n} - \gamma\|A^T(y_k + \alpha\Delta y_k) + s_k + \alpha\Delta s_k - c\|_2, \\
h_k(\alpha) &= (1 - \alpha(1 - \beta_2))x_k^T s_k - (x_k + \alpha\Delta x_k)^T(s_k + \alpha\Delta s_k).
\end{align*}
\]

For step length 0 < α < 1, positive values of f_k^i(\alpha), g_k^P(\alpha) and g_k^D(\alpha) show that the algorithm will remain in the neighborhood of the central path by taking this step. By h_k(\alpha) > 0, the optimality gap will sufficiently decrease after this step. Based on line (14) of iteration k, for any α ∈[0, ˆαk], we have

\[
\begin{align*}
f_k^i(\alpha) &\geq 0, \quad \text{for } i \in \{1, 2, \ldots, n\}, \quad (8a) \\
g_k^P(\alpha) &\geq 0, \quad \text{or } \|A(x_k + \alpha\Delta x_k) - b\|_2 \leq \zeta, \quad (8b) \\
g_k^D(\alpha) &\geq 0, \quad \text{or } \|A^T(y_k + \alpha\Delta y_k) + s_k + \alpha\Delta s_k - c\|_2 \leq \zeta, \quad (8c) \\
h_k(\alpha) &\geq 0. \quad (8d)
\end{align*}
\]

Lemma 3 and Lemma 4 indicate that 0 < ˆαk ≤α^*_k for all k. The following lemma shows some properties of the proposed II-QIPM.
Lemma 2. In Algorithm 1, at line (10) of iteration \(k\), we have
\[
(x_k, y_k, s_k) \in \mathcal{N}(\gamma, \zeta),
\]
\[
\|x_k, s_k\|_\infty \leq \omega,
\]
\[
x_k^T s_k \leq (1 - \alpha_k(1 - \beta_2))(x_{k-1}^T s_{k-1}).
\]

At lines (11) and (14) of iteration \(k\), for any \(\alpha \in [0, 1]\), we have
\[
\|A(x_k + \alpha \Delta x_k) - b\|_2 \leq (1 - \alpha)\|Ax_k - b\|_2 + \alpha \eta \frac{x_k^T s_k}{n},
\]
\[
\|A^T(y_k + \alpha \Delta y_k) + s_k + \alpha \Delta s_k - c\|_2 = (1 - \alpha)\|A^T y_k + s_k - c\|_2,
\]
\[
(x_k + \alpha \Delta x_k)^T(s_k + \alpha \Delta s_k) = (1 + \alpha(\beta_1 - 1))x_k^T s_k + \alpha^2 \Delta x_k^T \Delta s_k,
\]
\[
(x_k^T s_k) + \alpha \Delta s_k = (1 - \alpha)x_k^T s_k^i + \alpha \beta_1 \frac{x_k^T s_k}{n} + \alpha^2 \Delta x_k^i \Delta s_k^i,
\]
for \(i \in \{1, 2, \ldots, n\}\).

Proof. At line (10) of iteration \(k\), ((9a), (9c)) are valid because the step length is calculated in step (14) of the previous iteration such that conditions (8) hold. Since the algorithm did not stop at step 16 of the previous iteration, (9b) is also valid.

To prove (10a), for any \(\alpha \in [0, 1]\), we have
\[
\|A(x_k + \alpha \Delta x_k) - b\|_2 = \|b - Ax_k - \alpha A\Delta x_k\|_2,
\]
\[
= \|b - Ax_k - \alpha(b - Ax_k + r_k)\|_2,
\]
\[
= \|(1 - \alpha)(Ax_k - b) - \alpha r_k\|_2,
\]
\[
\leq (1 - \alpha)\|Ax_k - b\|_2 + \alpha\|r_k\|_2,
\]
\[
\leq (1 - \alpha)\|Ax_k - b\|_2 + \alpha \eta \mu_k,
\]
\[
\leq (1 - \alpha)\|Ax_k - b\|_2 + \alpha \eta \frac{x_k^T s_k}{n}. \tag{11f}
\]

In the equations (11), (11b) follows from (5), (11d) from the triangular inequality, (11e) from Assumption 1, and (11f) from definition of \(\mu_k\).

To prove (10b), based on (5), we have
\[
\|A^T(y_k + \alpha \Delta y_k) + s_k + \alpha \Delta s_k - c\| = \|c - A^T y_k - s_k - \alpha(A^T \Delta y + \Delta s)\|,
\]
\[
= \|c - A^T y_k - s_k - \alpha(c - A^T y_k - s_k)\|,
\]
\[
= (1 - \alpha)\|A^T y_k + s_k - c\|.
\]

To prove (10c), based on (5), we have
\[
(x_k + \alpha \Delta x_k)^T(s_k + \alpha \Delta s_k) = x_k^T s_k + \alpha(x_k^T \Delta s_k + s_k^T \Delta x_k) + \alpha^2 \Delta x_k^T \Delta s_k,
\]
\[
= x_k^T s_k + \alpha(n\beta_1 \mu_k - x_k^T s_k) + \alpha^2 \Delta x_k^T \Delta s_k,
\]
\[
= x_k^T s_k + \alpha(\beta_1 x_k^T s_k - x_k^T s_k) + \alpha^2 \Delta x_k^T \Delta s_k,
\]
\[
= (1 + \alpha(\beta_1 - 1))x_k^T s_k + \alpha^2 \Delta x_k^T \Delta s_k.
\]
Similarly, we can prove (10d) for all \( i \in \{1, 2, \ldots, n \} \) as follows:

\[
(x_i^k + \alpha \Delta x_i^k)(s_k + \alpha \Delta s_i^k) = x_i^k s_k^i + \alpha (x_i^k \Delta s_i^k + s_i^k \Delta x_i^k) + \alpha^2 \Delta x_i^k \Delta s_i^k,
\]

\[
= x_i^k s_k^i + \alpha (x_i^k \Delta s_i^k + s_i^k \Delta x_i^k) + \alpha^2 \Delta x_i^k \Delta s_i^k,
\]

\[
= x_i^k s_k^i + \alpha \beta (x_i^k s_k^i) - \alpha x_i^k s_k^i + \alpha^2 \Delta x_i^k \Delta s_i^k,
\]

\[
= (1 - \alpha) x_i^k s_k^i + \alpha \beta \frac{x_i^k s_k^i}{n} + \alpha^2 \Delta x_i^k \Delta s_i^k.
\]

Equations (12) follow from (5) and the definition of \( \mu_k \). Thus, the proof is complete. \( \square \)

We will use the results from Lemma 2 in the following Lemmas and Theorems.

**Lemma 3.** If II-QIPM does not terminate at line (12) of iteration \( k \), then

\[ 0 < \alpha_k \leq \alpha_k^* \]

**Proof.** First, we want to show that \( \hat{\alpha}_k < \alpha_k^* \). Suppose \( \hat{\alpha}_k \geq \alpha_k^* \), then conditions (8) hold for \( \alpha_k^* \). Based on the definition of \( \alpha_k^* \), there exist an index \( i \) with \( (x_i^k + \alpha_k^* \Delta x_i^k)(s_i^k + \alpha_k^* \Delta s_i^k) = 0 \). Then from condition (8a), we have

\[
0 \leq f_k^i(\alpha_k^*),
\]

\[
\leq (x_i^k + \alpha_k^* \Delta x_i^k)(s_i^k + \alpha_k^* \Delta s_i^k) - \gamma \frac{(x_k + \alpha_k^* \Delta x_k)(s_k + \alpha_k^* \Delta s_k)}{n},
\]

\[
\leq 0 - \gamma \frac{(x_k + \alpha_k^* \Delta x_k)(s_k + \alpha_k^* \Delta s_k)}{n}.
\]

From inequality (13), we have \( (x_k + \alpha_k^* \Delta x_k)^T(s_k + \alpha_k^* \Delta s_k) \leq 0 \). We also know that \( (x_k + \alpha_k^* \Delta x_k)^T(s_k + \alpha_k^* \Delta s_k) \geq 0 \) then

\[
(x_k + \alpha_k^* \Delta x_k)^T(s_k + \alpha_k^* \Delta s_k) = 0.
\]

Based on condition (8b), we have

\[ \|A(x_k + \alpha_k^* \Delta x_k) - b\|_2 \leq \zeta, \quad \text{or} \quad g_k^P(\alpha_k^*) \geq 0. \]

When \( g_k^P(\alpha_k^*) \geq 0 \), by (14), we have

\[
\frac{(x_k + \alpha_k^* \Delta x_k)(s_k + \alpha_k^* \Delta s_k)}{n} - \gamma \|A(x_k + \alpha_k^* \Delta x_k) - b\|_2 \geq 0,
\]

\[
0 - \gamma \|A(x_k + \alpha_k^* \Delta x_k) - b\|_2 \geq 0,
\]

\[
\|A(x_k + \alpha_k^* \Delta x_k) - b\|_2 = 0.
\]

Thus, both cases of (8b) lead to

\[ \|A(x_k + \alpha_k^* \Delta x_k) - b\|_2 \leq \zeta. \]
Similarly, from condition (8c), we have
\[ \|A^T(y_k + \alpha_k^* \Delta y_k) + s_k + \alpha_k^* \Delta s_k - c\|_2 \leq \zeta. \] (16)

By equality (14) and inequalities (15) and (16), we conclude that
\[ (x_k, y_k, s_k) + \alpha_k^* (\Delta x_k, \Delta y_k, \Delta s_k) \in PD_\zeta. \]

This contradicts the assumption that II-QIPM does not terminate at line (12). Thus, we can conclude that \( \hat{\alpha}_k < \alpha_k^* \). The inequality \( 0 < \alpha_k \leq \hat{\alpha}_k \) holds because of line (14) of iteration \( k \). Thus, we have \( 0 < \alpha_k \leq \hat{\alpha}_k < \alpha_k^* \) and the proof is complete.

By now, we proved that \( \alpha_k^* \) is upper bound for \( \alpha_k \) and \( \hat{\alpha}_k \). In the next Lemma we find a strictly positive lower bound for \( \hat{\alpha}_k \).

**Lemma 4.** Let \( \nu > 0 \), \( \left| \Delta x_k^i \Delta s_k^i - \frac{\gamma \Delta x_k^T \Delta s_k}{n} \right| \leq \nu \) for \( i \in \{1, 2, \ldots, n\} \) and \( |\Delta x_k^T \Delta s_k| \leq \nu \). Then, at line (14) of iteration \( k \), we have
\[ \hat{\alpha}_k \geq \tilde{\alpha}_k := \min \left\{ 1, \min\{\delta_1, \delta_2, \delta_3, \beta_1\} \frac{x_k^T s_k}{\nu} \right\}. \]

**Proof.** It is enough to show that conditions (8) hold for any \( \alpha \in [0, \tilde{\alpha}_k] \). Based on Lemma 2, for any \( \alpha \in [0, \tilde{\alpha}_k] \), we have
\[
\begin{align*}
f'_k(\alpha) &= (x_k^i + \alpha \Delta x_k^i)(s_k^i + \alpha \Delta s_k^i) - \gamma \frac{(x_k + \alpha \Delta x_k)^T (s_k + \alpha \Delta s_k)}{n}, \\
&= (1 - \alpha) x_k^T s_k + \alpha \beta_1 \frac{x_k^T s_k}{n} + \alpha^2 \Delta x_k^i \Delta s_k^i - \gamma \frac{(1 + \alpha(\beta_1 - 1)) x_k^T s_k + \alpha^2 \Delta x_k^i \Delta s_k^i}{n}, \\
&= \alpha^2 (\Delta x_k^i \Delta s_k^i - \frac{\gamma}{n} \Delta x_k^T \Delta s_k) + (1 - \alpha) \left( x_k^T s_k^i - \frac{\gamma}{n} x_k^T s_k \right) + \alpha \beta_1 (1 - \gamma) \frac{x_k^T s_k}{n}, \\
&\geq -\alpha^2 \nu + \alpha \delta_1 x_k^T s_k, \\
&\geq \alpha (\delta_1 x_k^T s_k - \nu \tilde{\alpha}_k), \\
&\geq 0.
\end{align*}
\]

In the above equations, equality (17b) follows from Lemma 2 and inequality (17d) comes from the Lemma’s assumption and definition of the neighborhood \( \mathcal{N}(\gamma, \zeta) \). For \( g_k^F \) there are two cases. Suppose \( g_k^F(0) = \frac{x_k^T s_k}{n} - \gamma \|Ax_k - b\|_2 \geq 0 \). Similarly, by Lemma 2 and
the assumption, for all $\alpha \in [0, \bar{\alpha}_k]$ we have
\[
g_k^P(\alpha) = \frac{(x_k + \alpha \Delta x_k)^T(s_k + \alpha \Delta s_k)}{n} - \gamma \|A(x_k + \alpha \Delta x_k) - b\|_2,
\]
\[
= \left(1 + \alpha(\beta_1 - 1)\right)\frac{x_k^T s_k}{n} + \alpha^2 \Delta x_k^T \Delta s_k - \gamma(1 - \alpha)\|A x_k - b\|_2 - \gamma \alpha \eta \frac{x_k^T s_k}{n},
\]
\[
\geq \left(1 + \alpha(\beta_1 - 1)\right)\frac{x_k^T s_k}{n} - \alpha^2 \nu + (1 - \alpha)(g_k^P(0) - \frac{x_k^T s_k}{n}) - \gamma \alpha \eta \frac{x_k^T s_k}{n},
\]
\[
\geq \alpha \delta_2 \frac{x_k^T s_k}{n} - \frac{\alpha^2 \nu}{n} + (1 - \alpha)g_k^P(0),
\]
\[
\geq \frac{\alpha}{n} (\delta_2 x_k^T s_k - \nu \bar{\alpha}_k),
\]
\[
\geq 0.
\]

In the other case, if $g_k^P(0) < 0$, we have $\|A x_k - b\|_2 \leq \zeta$ since $(x_k, y_k, s_k) \in \mathcal{N}(\gamma, \zeta)$. Thus
\[
\frac{x_k^T s_k}{n} < \gamma \|A x_k - b\|_2 \leq \gamma \zeta.
\]

Based on Lemma 2, for any $\alpha \in [0, \bar{\alpha}_k]$, we have
\[
\|A(x_k + \alpha \Delta x_k) - b\|_2 \leq (1 - \alpha)\|A x_k - b\|_2 + \alpha \frac{x_k^T s_k}{n} \leq (1 - \alpha)\zeta + \alpha \eta \zeta \leq \zeta.
\]

Similarly, for $g_k^D$ there are two cases. Suppose $g_k^D(0) = \frac{x_k^T s_k}{n} - \gamma \|A^T y_k + s_k - c\|_2 \geq 0$. By Lemma 2 and the assumption, we have for all $\alpha \in [0, \bar{\alpha}_k]$
\[
g_k^D(\alpha) = \frac{(x_k + \alpha \Delta x_k)^T(s_k + \alpha \Delta s_k)}{n} - \gamma \|A^T(y_k + \alpha \Delta y_k) + s_k + \alpha \Delta s_k - c\|_2,
\]
\[
= \left(1 + \alpha(\beta_1 - 1)\right)\frac{x_k^T s_k}{n} + \alpha^2 \Delta x_k^T \Delta s_k - \gamma(1 - \alpha)\|A^T y_k + s_k - c\|_2,
\]
\[
\geq \left(1 + \alpha(\beta_1 - 1)\right)\frac{x_k^T s_k}{n} - \alpha^2 \nu + (1 - \alpha)(g_k^D(0) - \frac{x_k^T s_k}{n}),
\]
\[
\geq \alpha \beta_1 \frac{x_k^T s_k}{n} - \frac{\alpha^2 \nu}{n} + (1 - \alpha)g_k^D(0),
\]
\[
\geq \frac{\alpha}{n} (\beta_1 x_k^T s_k - \nu \bar{\alpha}_k),
\]
\[
\geq 0.
\]

In the other case, if $g_k^D(0) < 0$, we have $\|A^T y_k + s_k - c\|_2 \leq \zeta$ since $(x_k, y_k, s_k) \in \mathcal{N}(\gamma, \zeta)$. Based on Lemma 2, for any $\alpha \in [0, \bar{\alpha}_k]$, we have
\[
\|A^T(y_k + \alpha \Delta y_k) + s_k + \alpha \Delta s_k - c\|_2 = (1 - \alpha)\|A^T y_k + s_k - c\|_2 \leq (1 - \alpha)\zeta \leq \zeta.
\]
By Lemma 2 and the assumption, we have for all \( \alpha \in [0, \tilde{\alpha}_k] \)

\[
    h_k(\alpha) = (1 - \alpha(1 - \beta_2))x_k^T s_k - (x_k + \alpha \Delta x_k)^T (s_k + \alpha s_k),
    
    = (1 - \alpha(1 - \beta_2))x_k^T s_k - (1 + \alpha(\beta_1 - 1))x_k^T s_k - \alpha^2 \Delta x_k^T \Delta s_k,
    
    \geq \alpha(\beta_2 - \beta_1) x_k^T s_k - \alpha^2 \nu,
    
    \geq \alpha(x_k^T s_k - \nu \tilde{\alpha}_k),
    
    \geq 0.
\]

We showed that for all \( \alpha \in [0, \tilde{\alpha}_k] \), conditions (8) hold. Thus, we can conclude that \( \tilde{\alpha}_k \leq \hat{\alpha}_k \) and the proof is complete.

So, by Lemma 4, we have a strictly positive step length \( \tilde{\alpha}_k \) to remain in the neighborhood of the central path as we decrease the optimality gap. In what follows, we want to show the convergence of the Algorithm 1 using the results of previous lemmas.

**Theorem 3.** II-QIPM terminates after a finite number of iterations.

**Proof.** Suppose that II-QIPM does not terminate. Thus, for any \( k \), we have \((x_k, y_k, s_k) \in N(\gamma, \zeta) \) and based on Theorem 1

\[
    \mu_k = \frac{x_k^T s_k}{n} > \zeta \gamma.
\]

Based on Lemma 2 and \( \|x_k, s_k\|_\infty \leq \omega \), we can write

\[
    (x_k, s_k) \in \Gamma := \left\{(x, s) \in \mathbb{R}^n \times \mathbb{R}^n \mid \gamma^2 \zeta \leq x_i s_i \leq \omega^2 \text{ for } i \in \{1, 2, \ldots, n\}\right\}.
\]

Now, we want to bound the right-hand side vector of the Newton system. Similar to the proof of Lemma 2, we can write

\[
    \|b - Ax_k + r_k\|_2 \leq \|Ax_k - b\|_2 + \|r_k\|_2,
    
    \leq \|Ax_k - b\|_2 + \eta x_k^T s_k/n,
    
    \leq \|Ax_k - b\|_2 + \eta \omega^2,
    
    \leq \|Ax_0 - b\|_2 + k\eta \omega^2.
\]

For the second part of the right-hand side, we also have

\[
    \|c - A^T y_k - s_k\|_2 = \prod_{i=0}^{k-1} (1 - \alpha_k) \|c - A^T y_0 - s_0\|_2,
    
    \leq \|c - A^T y_0 - s_0\|_2.
\]
Similarly, we have
\[
\|\beta_1 \mu_k e - X_k s_k\|_2^2 = \sum_{i=1}^{n} \left( \beta_1 x_k^T s_k - x_i s_i \right)^2,
\]
\[
\leq \sum_{i=1}^{n} \left( \frac{\beta_1 n \omega}{n} + \omega^2 \right)^2,
\]
\[
\leq (\beta_1 + 1) \sqrt{n} \omega^2.
\]

Hence, the step \((\Delta x_k, \Delta y_k, \Delta s_k)\) is the solution of system
\[
\begin{pmatrix}
A & 0 & 0 \\
0 & A^T & I \\
S_k & 0 & X_k
\end{pmatrix}
\begin{pmatrix}
\Delta x_k \\
\Delta y_k \\
\Delta s_k
\end{pmatrix}
= \begin{pmatrix}
b - Ax_k + r_k \\
c - A^T y_k - s_k
\end{pmatrix},
\]
where
\[
\|b - Ax_k + r_k\|_2 \leq \|Ax_0 - b\|_2 + k \eta \omega^2,
\]
\[
\|c - A^T y_k - s_k\|_2 \leq \|c - A^T y_0 - s_0\|_2,
\]
\[
\|\mu_k e - X_k s_k\|_2 \leq (\beta_1 + 1) \sqrt{n} \omega^2.
\]

Since \(\Gamma\) is a compact set, there exists a compact set containing the coefficient matrix of the linear system (18) where all the matrices are non-singular. The inverses of all matrices are bounded, and thus, \((\Delta x_k, \Delta y_k, \Delta s_k)\) is bounded for all \(k \geq 0\). Then, there exist \(\nu > 0\) with \(|\Delta x_k^T \Delta s_k - \gamma \Delta x_k^T \Delta s_k| \leq \nu\) for \(i \in \{1, 2, \ldots, n\}\) and \(|\Delta x_k^T \Delta s_k| \leq \nu\).

Based on Lemma 4, we have
\[
\hat{\alpha}_k \geq \alpha_k \geq \alpha^L := \min \left\{ 1, \min \{\delta_1, \delta_2, \delta_3, \beta_1\} \frac{\zeta^*}{\nu} \right\} \in (0, 1].
\]

Hence, by the definition of the neighborhood, we have
\[
x_k^T s_k \leq (1 - \alpha^L(1 - \beta_2))^k x_0^T s_0.
\]
This implies that \(\lim_{k \to \infty} x_k^T s_k = 0\) which contradicts with the initial assumption that \(x_k^T s_k \geq \zeta^*\). Thus, the algorithm is convergent.

We see that the II-QIPM is convergent and we can easily drive the following corollaries.

**Corollary 2.** For any iteration \(k\), we have \(\alpha_k > 0\).

**Remark 2.** One option for finding \(\alpha_k\) is \(\alpha^L\). By this choice, we do not need to find \(\hat{\alpha}_k\) in II-QIPM.

**Corollary 3.** The sequence \(\{\mu_k\}\) generated by II-IPM converges linearly to zero. In consequence, primal infeasibility \(\|Ax_k - b\|_2\) and dual infeasible \(\|A^T y_k + s_k - c\|_2\) converge to zero Linearly.
4.2 Time complexity of II-QIPM

In this section, we analyze the time complexity of the II-QIPM. The following Theorem gives a polynomial bound for the number of iterations of the II-QIPM.

**Theorem 4.** The number of iterations of the II-QIPM has upper bound $O(n^2 \log \zeta^{-1})$.

**Proof.** From Corollary 2, the optimality gap converges to zero. Also, based on the definition of $N$, the primal and dual infeasibility converge to zero. Based on Theorem 1, the algorithm stops when

$$\mu_k \leq (1 - \alpha^L (1 - \beta_2))^k \mu_0 \leq \zeta$$

By the definition of $\alpha^L$, one can verify that $1/\alpha^L = O(n^2)$ (Korzak, 2000). Since $\mu_0 = \omega^2$, we can conclude that $k = O(n^2 \log \zeta^{-1})$.

We have shown that the number of iterations of the II-QIPM is bounded by $O(n^2 \log \zeta^{-1})$ as Theorem 4. Also, we need to solve the NES at each iteration of the II-QIPM by the QLSA+QTA with $O(m \text{ polylog} (\| \sigma_k \|/\mu_k) \kappa_k \| \sigma_k \|^2_{\mu_k})$ computational cost as discussed in Theorem 2. We can calculate the total time complexity of the II-QIPM as the product of the time complexity of the QLSA and the number of iterations of the II-QIPM. The computational cost of the QLSA depends on the condition number of the NES $\kappa_k$ and $\mu_k$, which change through the algorithm. In the following section we try to find an appropriate bound for them.

4.3 Bounds on $\| \sigma \|$, $\| M \|$ and $\kappa$

Before analyzing bounds, we review how matrix $M$ evolves through the iterations. We solve the NES with $M_k = AX_k S_k^{-1} A^T$ as coefficient matrix. As in (Wright, 1997) with partition $N \cup B = \{1, 2, \ldots, n\}$ where $B \cap N = \emptyset$, we have

$$\frac{x_i^k}{s_i^k} = O(\mu_k^{-1}) \to \infty \text{ for } i \in B \quad \text{and} \quad \frac{x_i^k}{s_i^k} = O(\mu_k) \to 0 \text{ for } i \in N.$$ (19)

Appropriate bounds are provided in the following results.

- Based on the Theorem 3, we have $1/\mu_k = O(\zeta^{-1})$ and $\mu_k = O(\mu_0) = O(\omega^2)$.
- Since $\| M_k \|_2 \leq \| A \|_2 \cdot \| X_k S_k^{-1} \|_2 \cdot \| A^T \|_2$, $\| A^T \|_2 = \| A \|_2$, and $\| X_k S_k^{-1} \|_2 = O(\zeta^{-1})$ by (19), we have $\| M \|_2 = O(\zeta^{-1} \| A \|_2^2)$.
- Let $\kappa_A$ be the condition number of matrix $A$. By using $\kappa_k = \| M_k \| \cdot \| M_k^{-1} \|$, we have

$$\kappa_k = O(\zeta^{-2} \kappa_A^2).$$
• We can easily verify that
\[
\|\sigma_k\|_2 \leq \|(b - Ax_k)\| + \|AX_kS_k^{-1}\| \|c - A^T y_k - s_k\| + \|Ax_k\| + \|\mu_k AS_k^{-1} e\|
\leq \frac{\mu_k}{\gamma} + \|AX_kS_k^{-1}\| \|\mu_k\| \gamma + \|AX_kS_k^{-1}\| \|\mu_k\| \gamma + \|\mu_k\| \gamma + \|\mu_k\| \gamma.
\]

Thus, we have \(\|\sigma_k\|_2 = O(\|A\| \omega + \|b\|)\).

• To find the bound for \(\frac{1}{\epsilon_{QLSP_k}}\), we can use following bounds for \(\mu_k\) and \(\|\sigma_k\|_2\) leading to
\[
\frac{1}{\epsilon_{QLSP_k}} = O(\frac{\|\sigma_k\|}{\mu_k}) = O(\frac{\|A\| \omega^2 + \|b\|}{\zeta}).
\]
Since both \(\mu_k\) and \(\|\sigma_k\|_2\) are decreasing during the iteration of algorithm, we can get tighter bound directly as
\[
\frac{1}{\epsilon_{QLSP_k}} = O(\frac{\|A\| + \|b\|}{\zeta}).
\]
Similarly, we have \(\frac{1}{\epsilon_{QLSA_k}} = O(\frac{\|A\| + \|b\|}{\zeta})\) and \(\frac{1}{\epsilon_{QTA_k}} = O(\frac{\|A\| + \|b\|}{\zeta})\).

• The time complexity of QLSA by Chakraborty, Gilyén, and Jeffery, 2018 for solving the NES is \(O(\text{polylog}(m\|A\| + \|b\|) \frac{\kappa_A^2}{\zeta^2})\).

• The time complexity of QTA by Kerenidis and Prakash, 2020 for solving the NES is \(O(\frac{m\|A\| + \|b\|}{\zeta^2})\).

Thus, the detailed time complexity of the proposed II-QIPM with QLSA by Chakraborty, Gilyén, and Jeffery, 2018 and QTA by Kerenidis and Prakash, 2020 is
\[
O(n^2 \log \frac{\omega}{\zeta} (m^2 n + \text{polylog}(m\|A\| + \|b\|) \frac{\kappa_A^2}{\zeta^2} m\|A\| + \|b\|)^2)).
\]

This time complexity is multiplication of number of iteration of II-QIPM and cost of each iteration, including building the NES in classical and solving it by QLSA+QTA.

By assumption that \(m = O(n)\), the time complexity of II-QIPM can be simplified \(O(n^5 \text{polylog}(\frac{\omega}{\zeta}) \zeta^{-4} \|A\|^2 \kappa_A^2)\). In complexity of II-QIPM, there is \(\zeta^{-1}\) which can be exponentially big to find an exact solution. We discuss how we can solve this issue by an Iterative Refinement scheme in Section 6. Also, most of the dependence on \(n\) comes from building the NES in classical computer. In Section 7, we investigate how other systems can have better time complexity.

5 Sparsity of the Matrix of the Normal Equation System

Some QLSAs are dependent on the sparsity of the NES such as (Harrow, Hassidim, and Lloyd, 2009) and (Childs, Kothari, and Somma, 2017).

Since the sparsity of \(M_k = AX_kS_k^{-1}A^T\) is independent of \(X_k\) and \(S_k\), \(d_k = d\) for all \(k\), where \(d\) is the maximum number of nonzero elements in any row or column of matrix.
Generally, we know that $d \leq m$ and when the matrix is fully dense we have $d = m$. We can rewrite matrix $M_k$ as

$$M_k = \sum_{i=1}^{n} \frac{x_i^k}{s_k^i} A_i A_i^T,$$

where $A_i$ is the $i$th column of matrix $A$. It should be noted that matrix $A$ determines the sparsity of matrix $M_k$, so we can take advantage of the structure of matrix $A$ if matrix $A$ has sufficiently more sparse columns than dense columns. Let $D = \{ i \in \{1, \ldots, n\} | A_i \text{ is dense} \}$ and $S = \{1, 2, \ldots, n\} \setminus D$. So we have

$$M_k = \sum_{i \in D} \frac{x_i^k}{s_k^i} A_i A_i^T + \sum_{i \in S} \frac{x_i^k}{s_k^i} A_i A_i^T.$$

As we can see $M_k$ has a sparse part $M_k^S = \sum_{i \in S} \frac{x_i^k}{s_k^i} A_i A_i^T$ and we can find the solution of $M_k^S w = \sigma_k$ by QLSA by Childs, Kothari, and Somma, 2017 with complexity $O(\text{polylog}(\frac{m\kappa_k}{\epsilon_k}d'\kappa_k))$ where the sparsity of matrix $M_k^S$, $d' \ll m$. Then, we can find the solution $\Delta y_k$ by Sherman–Morrison–Woodbury formula. Here, it is important to know that whether $M_k^S$ is singular or not. We can apply the mentioned approach if $M_k^S$ is nonsingular. However, in case of singularity, we need to build a nonsingular matrix. For more detail of classical procedure, the reader is referred to (Andersen et al., 2004). The total time complexity is determined by solving the sparse system because extracting $\Delta y_k$ by Sherman–Morrison–Woodbury formula only needs some matrix and vector multiplications. This method will decrease $d$ the sparsity of matrix $A$ to $d'$ the sparsity of sparse columns of matrix $A$. The structure of matrix $A$ determines how much this approach is effective.

6 Iterative Refinement to find Exact Solution

To have an exact solution, we need $\zeta \leq 2^{-2L}$ based on Corollary 1. Thus, the proposed II-QIPM has exponential time complexity. An Iterative Refinement scheme can be employed to have a polynomial complexity. So, a LO problem is solved by the II-QIPM with large $\zeta$ such as $\zeta = 10^{-2}$ and the Iterative Refinement algorithm improves the precision to reach an exact solution. In the classical IPM literature, several studies address the numerical analysis of reaching an exact solution, such as the Iterative Refinement scheme by A. M. Gleixner, Steffy, and Wolter, 2016 and the Rational Reconstruction method by A. Gleixner and Steffy, 2019. This paper uses the Iterative Refinement scheme as the general procedure to find an exact optimal solution. Theorem 5 is the foundation of the Iterative Refinement method.

**Theorem 5** (A. M. Gleixner, Steffy, and Wolter, 2016). Let the primal problem $(P)$ be given as (1). For $\bar{x} \in \mathbb{R}^n$ and $\bar{y} \in \mathbb{R}^m$ and scaling factor $\nabla > 0$ consider the transformed problem $(\bar{P})$

$$\max \{ \nabla c^T x | Ax = \nabla b \text{ and } x \geq 0 \},$$
where \( \tilde{c} = c - A^T\tilde{y} \) and \( \tilde{b} = b - A\tilde{x} \). Then \( \tilde{x} \) and \( \tilde{y} \) are \( \hat{\zeta} \)-optimal solution for problem \( \tilde{P} \) if and only if \( \tilde{x} + \frac{1}{\rho}\tilde{x} \) and \( \tilde{y} + \frac{1}{\rho}\tilde{y} \) are the \( \hat{\zeta} \)-optimal solution for problem \( P \).

By this idea, we develop the following Iterative Refinement algorithm, which calls II-QIPM at each iteration. We used the scheme as described by A. M. Gleixner, Steffy, and Wolter, 2016.

**Algorithm 2 Iterative Refinement (IR-II-QIPM)**

**Require:** \((A \in \mathbb{Z}^{m \times n}, b \in \mathbb{Z}^m, c \in \mathbb{Z}^n, \hat{\zeta} = 10^{-2}, \zeta = 2^{-2L})\)

1: Choose incremental scaling limit \( \rho \in \mathbb{N} \) such that \( \rho > 1 \)
2: \( k \leftarrow 0 \)
3: \((x_0^*, y_0^*) \leftarrow \text{solve} (A, b, c) \text{ using the II-QIPM 1 with } \hat{\zeta} \text{ precision} \)
4: loop
5: \( \bar{b}_k \leftarrow b - Ax_k^* \) and \( \bar{c}_k \leftarrow c - A^Ty_k^* \)
6: \( r_k \leftarrow \max \left\{ \max_j |\bar{b}_k^j|, \max_i (-\bar{c}_k^i), \sum_i |\bar{c}_k^i| x_k^*| \right\} \)
7: if \( r_k \leq \zeta \) then
8: return \((x_k^*, y_k^*)\)
9: \( \bar{r}_k \leftarrow \max \left\{ r_k, \frac{1}{\rho \nabla k} \right\} \)
10: \( k \leftarrow k + 1 \)
11: \( \nabla k \leftarrow 2^{\left\lfloor \log(1/\bar{r}_{k-1}) \right\rfloor} \)
12: \((\hat{x}_k, \hat{y}_k) \leftarrow \text{solve} (A, \nabla_k \bar{b}_{k-1}, \nabla_k \bar{c}_{k-1}) \text{ using II-QIPM 1 with } \hat{\zeta} \text{ precision} \)
13: \( x_k^* \leftarrow x_{k-1}^* + \frac{1}{\nabla_k} \hat{x}_k \) and \( y_k^* \leftarrow y_{k-1}^* + \frac{1}{\nabla_k} \hat{y}_k \)

As we can see in Algorithm 2, we use the II-QIPM 1 to solve the refining model and update the solution with a smart scaling procedure. The Theorem 6 shows the polynomial number of iteration is needed to reach an exact optimal solution.

**Theorem 6** (A. M. Gleixner, Steffy, and Wolter, 2016). *Number of iterations of Algorithm 2 is at most

\[
\left\lceil \log(\zeta) \over \log(\hat{\zeta}) \right\rceil = \left\lceil -2L \log(2) \over -2 \right\rceil = O(L).
\]

As we can see that all parameters in the time complexity of II-QIPM are constant in all iterations of Iterative Refinement except \( \omega \). To have a detailed time complexity for IR-II-QIPM, Lemma 5 provides a bound for \( \omega_k \) for any iteration \( k \) of Iterative Refinement.

**Lemma 5.** *In iteration \( k \) of Iterative Refinement, 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, we have \( \omega_k = O((2\rho)^L) \).*
Proof. From Theorem 6 and last line of Algorithm 2, we have
\[ x^* = x_0^* + \sum_{k=1}^{L} \frac{1}{\nabla_k} \hat{x}_k, \]
\[ s^* = s_0^* + \sum_{k=1}^{L} \frac{1}{\nabla_k} \hat{s}_k. \]

Based on Lemma 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 \( \nabla_k = O(\rho^L) \). We can conclude that \( \omega_k = O((2\rho)^L) \).

In Theorem 7, we have the total time complexity of the Iterative Refinement using the proposed II-QIPM to find an exact optimal solution for LO problems.

**Theorem 7.** Let \( \hat{\zeta} = 10^{-2} \), the total time complexity of finding an exact optimal solution using the II-QIPM Algorithm 1 and Iterative Refinement for the LO problem (1) is polynomial with
\[ O\left(n^2 L (m^2 n + (\text{polylog}(m||A|| + \|b\|))n^2) m(||A|| + \|b\|)^2)\right), \]
the arithmetic operations.

*Proof.* The proof comes from combining the result of Theorem 6, total time complexity of II-QIPM in (20), and Lemma 5.

From the result of Theorem 7, the simplified time complexity of IR-II-QIPM using the NES is \( O(n^5 L||A||^2 n^2) \). As the result indicates, building the NES in classical computers has more complexity than solving it by quantum computer with respect to the dimension. In next section, other Newton systems are discussed to compare their adaptability for using in the II-QIPM.

### 7 Effect of different Newton Systems

In Section 4, II-QIPM is proposed with the NES which has \( m\)-by-\( m \) positive definite and Hermitian matrix appropriate for QLSAs. The full Newton system is
\[ A\Delta x_k = b - Ax_k + r_P, \]
\[ A^T \Delta y_k + \Delta s_k = c - A^T y_k - s_k + r_D, \]
\[ X_k \Delta s_k + S_k \Delta x_k = \beta_1 \mu_k e - X_k s_k + r_C, \]
\[ (21) \]
where \((r_P, r_D, r_C)\) are residuals generated by inexact QLSAs. As mentioned in equation (5), when the NES is solved instead of the full Newton system, then \( r_C = r_D = 0 \).
One can easily update the proofs for II-QIPM with the full Newton system. The augmented system is
\[
A \Delta x_k = b - Ax_k + r'_P, \\
A^T \Delta y_k + X_k^{-1} S_k \Delta x_k = c - A^T y_k - \beta_1 \mu_k X_k^{-1} e + r'_D. 
\] (22)

After solving the Augmented system, \( \Delta s_k \) can be calculated by the last line of system (21). When we solve the Augmented system, we have \( r_C = 0 \). Again, one can redo the convergence proof for II-QIPM with Augmented system. Since the full Newton system and the Augmented system are neither symmetric nor Hermitian, we need the first step of the procedure mentioned in Section 2 to make the matrix of system Hermitian which increases the size of system. Due to limitation on available Qbits on current quantum computers and simulators, we prefer to use the NES which has smaller dimension. Table 2 shows that the Augmented system and the full Newton system have better dependence on \( n \) since building them in classical computer has smaller cost than building the NES. The difference in the time complexities of these system comes from difference in the norm of right-hand side vector which highly affects the precision of QLSA and QTA. The full Newton system has the best complexity since its right-hand side vector decreases with same rate \( \mu_k \). The issue of full Newton system is its huge size which can not be implemented for even very small problem with current quantum hard wares and simulators. Although the Augmented system has better dependence on \( n \), its time complexity is not polynomial since the norm of right-hand side vector for this system is dependent on \( \omega \). Similar to theoretical analysis, we will use the NES in our implementation in next section since it is much smaller than other systems. Maybe in future, by development in quantum simulators and hard wares the full Newton system get more interest due its better complexity and sparsity.

<table>
<thead>
<tr>
<th>Size</th>
<th>Full Newton System</th>
<th>Augmented System</th>
<th>NES</th>
</tr>
</thead>
<tbody>
<tr>
<td>QLSA</td>
<td>( O((2n + m)) )</td>
<td>( O((2n + m)) )</td>
<td>( O(m) )</td>
</tr>
<tr>
<td>QTA</td>
<td>( O((2n + m)n) )</td>
<td>( O((2n + m)n) )</td>
<td>( O(m(n + m)\omega^2 )</td>
</tr>
<tr>
<td>II-QIPM</td>
<td>( O(n^3 \text{polylog}(\frac{\omega^2}{\epsilon} \kappa^{-2} \eta_\delta A)) )</td>
<td>( O(n^3 \text{polylog}(\frac{\omega^2}{\epsilon} \kappa^{-2} \eta_\delta A)) )</td>
<td>( O(n^3 \text{polylog}(\frac{\omega^2}{\epsilon} A \kappa^{-2} \eta_\delta A)) )</td>
</tr>
<tr>
<td>IR-II-QIPM</td>
<td>( O(n^4 \text{polylog}(n L \kappa^{-4} \eta_\delta A)) )</td>
<td>( O(n^4 \text{polylog}(n L^2 \kappa^{-4} \eta_\delta A)) )</td>
<td>( O(n^5 L \text{polylog}(\frac{\omega^2}{\epsilon} A \kappa^{-4} \eta_\delta A)) )</td>
</tr>
</tbody>
</table>

Table 2: II-QIPM+Iterative Refinement for different systems

8 Numerical Experiments

The proposed II-QIPM is implemented in Python using QISKIT AQUA simulator of the HHL algorithm. First, the performance of the simulator is examined in Section 8.1 to tune it well for solving the NES in II-QIPM. In Section 8.2, We analyze how II-QIPM behave throw the iterations. Finally, the Iterative Refinement is adapted to increase the precision in Section 8.3.
8.1 Evaluation of the Quantum Simulator

The simulator has a bit different configuration from theoretical HHL algorithm. There is no direct parameter for tuning precision. First parameter which highly affects the precision is the time of the Hamiltonian simulation \( t \). The best value is \( t = \frac{\pi}{\lambda_{max}} \) where simulator can almost find solution with small error for scaled system. Another value for this parameter leads to considerable error affecting convergence of II-QIPM. Finding maximum eigenvalue is not computationally expensive by using Power Iteration method. In our implementation we used \( t = \frac{\pi}{\lambda_{max}} \) where the maximum eigenvalue of the NES \( \lambda_{max} \) is calculated by Power Iteration method. We also need to tune other parameters to increase the precision of HHL simulator and decrease the time of solution. Another parameter affecting the performance of the HHL simulator is the number of ancillae qubits. Figure 2 depicts the time and error of solving randomly generated systems with different numbers of ancillae qubits. We can see that by increasing the number of ancillae qubits, the error decreases but time increases. For this case, the appropriate number of ancillae qubits is four since after that error slightly decreases but time rapidly increases.

![Figure 2: Effect of the number of ancillae qubits on the error and time of the HHL simulator](image)

It is necessary to evaluate the performance of HHL simulator based on features of the NES such as condition number, norm of coefficient matrix and right-hand side vector. As expected, Figure 3 shows that the dimension of system does not affect the error but the time of solution is increasing like a step function since the HHL simulator build a system where dimension is a power of two.
Figure 3: Effect of dimension on the time and the error of the HHL simulator

To confirm the theoretical result, Fig 4 illustrates that the norm of right-hand side vector affects the error due to scaling but it does not affect the time of solution. In theoretical result it affects the time complexity since we fixed the target error and QLSAs need amount of time proportional to the norm of the right-hand side vector. Contrary to theoretical analysis of HHL method, HHL simulator is almost independent on the condition number of the coefficient matrix as depicted in Figure 5. The reason is that the number of iterations of HHL method is $O(\kappa)$ to get solution with high probability and in the simulator we do not need to replicate the computation since the probability can be calculated numerically. This seems favorable for II-QIPM since the growing condition number of NES does not affect the performance of the simulator. Figure 6 indicates that the norm of coefficient matrix of the NES does not affect the time and the error of solution.

Figure 4: Effect of the norm of right-hand side vector on the time and the error of the HHL simulator
8.2 Evaluation of II-QIPM

As we studied in Section 8.1, there are some opportunities and challenges using quantum simulator. Since simulators need much more time than real quantum computers, evaluating time of solution is not an appropriate empirical analysis. In addition we can analyze how the II-QIPM behaves using HHL simulator. In Figure 7, we can see how the II-QIPM converges to an optimal solution. As the algorithm approaches to optimal solution, the complementarity decreases along with primal and dual residuals. Although the condition number is increasing, the error of QLSA is decreasing since the norm of right-hand side is decreasing rapidly. This is a favorable phenomenon since as the algorithm gets near to optimal solution, higher precision is required for the QLSA.
One important parameter which highly affects the norm of right-hand side vector and consequently the error of HHL simulator is $\omega$ which is bound for the norm of optimal solution. Figure 8 shows how increasing $\omega$ leads to increase in the error and time of II-QIPM. By increasing $\omega$, not only the norm of the right-hand side vector in initial steps increases, but also the II-QIPM needs more iteration to decrease the optimality gap.

### 8.3 Evaluation of IR-II-QIPM

Even with smart parameter tuning and prefect implementation, the HHL simulator has limited precision. For the case of Table 3, the II-QIPM can get the $10^{-1}$ precision and with iterative refinement we could improve the precision as we want but the time of solution will increases but not exponentially.

### Table 3: Effect of iterative refinement in improving the precision

<table>
<thead>
<tr>
<th>Method</th>
<th>Error</th>
<th>Time (s)</th>
<th># Iterations</th>
<th>Complementarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>II-IPM</td>
<td>1e-01</td>
<td>0.13</td>
<td>12</td>
<td>3.16e-02</td>
</tr>
<tr>
<td>II-QIPM</td>
<td>1e-01</td>
<td>785.91</td>
<td>13</td>
<td>5.08e-02</td>
</tr>
<tr>
<td>IR-II-IPM</td>
<td>1e-04</td>
<td>0.26</td>
<td>2</td>
<td>1.98e-05</td>
</tr>
<tr>
<td>IR-II-QIPM</td>
<td>1e-04</td>
<td>4948.85</td>
<td>5</td>
<td>1.10e-05</td>
</tr>
</tbody>
</table>
All the methods are implemented in a Python package available for public at url: https://github.com/Fakhimi/interior_point_methods.
This solver gives the option of choosing quantum or classical linear solver to solve LO problems. Also it has option for using Iterative Refinement to improve the precision.

9 Conclusion

In general, this paper proposes a convergent hybrid algorithm by using an Iterative Refinement scheme, Inexact Infeasible IPM, Quantum Tomography, and Quantum Linear System algorithms to find an exact optimal solution for LO problems in a polynomial time. Table 4 compares the best time complexity for IR-II-QIPM with two analogous classical II-IPMs and two QIPMs.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Simplified Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>II-IPM with Cholesky</td>
<td>$O(n^5L)$</td>
</tr>
<tr>
<td>II-IPM with CG</td>
<td>$O(n^3L^{2\frac{3}{2}L^2})$</td>
</tr>
<tr>
<td>QIPM of (Kerenidis and Prakash, 2020)</td>
<td>$O(n^3L^{2\frac{4}{L^2}})$</td>
</tr>
<tr>
<td>QIPM of (Casares and Martin-Delgado, 2020)</td>
<td>$O(n^2L^{2\frac{4}{L^2}})O(n^{2L})$</td>
</tr>
<tr>
<td>Proposed IR-II-QIPM</td>
<td>$O(n^4\text{polylog}(n)L^{2A^2})$</td>
</tr>
</tbody>
</table>

Table 4: Time complexity of finding exact solution

The proposed IR-II-QIPM has a polynomial complexity while the other QIPMs cannot find an exact optimal solution due to exponential complexity caused by the error of QLSA and the large condition number of the Newton system. Classical II-IPM with CG method can face with exponential time complexity and some classical papers used preconditioning techniques to avoid it (e.g. Al-Jeiroudi and Gondzio, 2009). It seems that two other QIPMs have better time complexity but these time complexities can not be attained since they contain the iteration complexity of exact IPMs while these QIPMs solve the Newton system inexactly. They also need appropriate bound for $\kappa$ and $\epsilon$ based on their setting of QIPM. To correct the time complexity of QIPMs, $O(n^{2.5})$ must be added for inexact Newton steps and appropriate bound for the error of QLSAs. The second part of time complexity of QIPM by Casares and Martin-Delgado, 2020 is time complexity of QTA which was ignored by the authors of that paper. By using QLSA and QTA, the time complexity of proposed method has better dependence on $n$ than complexity of its classical counterparts and realistic complexity of its quantum complexity. It is worth mentioning that the performance of the proposed method depends on $\kappa_A$ which is constant. For LO problem with large $\kappa_A$, scaling and preconditioning techniques can be applied to decrease $\kappa_A$. These techniques must be used carefully since they can affect the final precision.

Based on the complexity analysis of the proposed method, building the Newton systems in classical computers has more dependence on $n$ than solving these systems by
quantum computers. To have more speed-up in hybrid QIPMs, building the Newton system can be investigated in future researches. In quantum part, improving the complexity of QTAs and QLSAs can lead to better time complexity for QIPMs. In the classical part, we can also investigate Inexact-Feasible IPM more adaptable with QLSAs since Feasible IPMs have better complexity than infeasible IPMs. Another direction can be developing pure QIPM which all calculations happen in the quantum setting. Although there are some limitations in current quantum linear algebra to have a pure QIPM, such a method does not need QTA inside and can get an advantage from fast QLSAs. One interesting result from both theoretical and empirical analysis is the considerable effect of the right-hand side vector on the performance of both QLSAs and QTAs due to scaling. In consequence, the norm of this vector can highly affect the performance of QIPM although it was not such important parameter in the classical setting.

Some of the major contributions of this paper are analyzing the advantages and challenges of using QLSAs inside the IPMs in detail and proposing a convergent II-QIPM adaptable with QLSAs. Although prior papers ignored the fact that the required precision and the condition number of Newton system in the complexity of QLSAs go to infinity with exponential speed as approaching an exact optimal solution, an Iterative Refinement scheme is used to find an exact solution in polynomial time without excessive time of QLSAs. After addressing issues in the existing QIPMs, we proved the correctness and convergence of our method and analyzed its performance both theoretically and empirically.

Based on the result of implementing II-QIPM augmented with Iterative Refinement using the QLSA simulator of QISKIT AQUA, we can solve problems with large number of variables to user-defined precision while the current simulator cannot find the solution of a linear system with precision higher than $10^{-2}$. There is a limitation in the number of constraints since we can simulate only few Qbits by a classical computer. The time of solution is not an appropriate measure to judge quantum methods since simulating quantum computers in classical computers requires dramatically more time than a real quantum computer. Although there are Feasible IPMs with better complexity than the proposed IR-II-QIPM, this paper is a great step to use quantum solvers in classical methods correctly and efficiently. Also, LO problems practically can be solved using quantum solvers for the first time.

10 Acknowledgement

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


Wossnig, Leonard, Zhikuan Zhao, and Anupam Prakash (Jan. 2018). “Quantum Linear System Algorithm for Dense Matrices”. In: Physical Review Letters 120.5. DOI: 10.1103/physrevlett.120.050502. URL: https://doi.org/10.1103/physrevlett.120.050502.

