Robust Interior Point Method for Quantum Key Distribution Rate Computation

Hao Hu[1,2], Jiyoung Im[1], Jie Lin[3,4], Norbert Lütkenhaus[3], and Henry Wolkowicz[1]

[1]Department of Combinatorics and Optimization, Faculty of Mathematics, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

[2]Department of Mathematical Sciences, Clemson University, Clemson, SC, United States 29634

[3]Institute for Quantum Computing and Department of Physics and Astronomy, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

[4]Department of Electrical & Computer Engineering, University of Toronto, Toronto, Ontario, Canada M5S 3G4

Friday 2nd September, 2022

Security proof methods for quantum key distribution, **QKD**, that are based on the numerical key rate calculation problem, are powerful in principle. However, the practicality of the methods are limited by computational resources and the efficiency and accuracy of the underlying algorithms for convex optimization. We derive a stable reformulation of the convex nonlinear semidefinite programming, **SDP**, model for the key rate calculation problems. We use this to develop an efficient, accurate algorithm. The stable reformulation is based on novel forms of facial reduction, **FR**, for both the linear constraints and nonlinear quantum relative entropy objective function. This allows for a Gauss-Newton type interior-point approach that avoids the need for perturbations to obtain strict feasibility, a technique currently used in the literature. The result is high accuracy solutions with theoretically proven lower bounds for the original **QKD** from the **FR** stable reformulation. This provides novel contributions for **FR** for general **SDP**.

We report on empirical results that dramatically improve on speed and accuracy, as well as solving previously intractable problems.

Keywords: Key rate optimization, **QKD**, quantum key distribution, Semidefinite programming, **SDP**, Gauss-Newton, **GN**, search direction.

AMS subject classifications: 81P17, 81P45, 81P94, 90C22, 90C25, 90C30, 90C59, 94A60.

Contents

List of Tables

List of Figures

1 Introduction

Quantum key distribution, ***QKD*** (see e.g., [1, 2] for reviews), is a secure communication method that distributes a secret key between two honest parties (traditionally known as Alice and Bob), even in the presence of an eavesdropper (traditionally called Eve). Those keys can be used for secure communication, authentication, secure multi-party computation and other cryptographic applications. Moreover, **QKD** is a quantum-resistant (quantum-safe) key establishment protocol. The need for quantum-resistant cryptography is widely recognized given that the threat of quantum computers to current cryptosystems can become a reality in the future. In contrast to the area of post-quantum cryptography that is also believed to be quantum-resistant, one can actually prove the information-theoretic security of **QKD** protocols, an attractive and important feature. Moreover, in these security proofs, one need only assume that Eve follows the laws of quantum mechanics. Such an all-powerful Eve can be assumed to have access to unlimited computational powers, including not-yet-available quantum computers. The core of a proof of security of any **QKD** protocol is to calculate the secret key rate, which is the number of secret key bits obtained per exchange of a quantum signal. As one needs to take into account all possible eavesdropping attacks from an all-powerful Eve, an analytical calculation of the key rate is extremely challenging. Such analytical calculations are generally limited to protocols with high symmetry like the BB84 protocol [3]. Moreover, one often has to invoke inequalities to obtain an analytical lower bound of the true key rate. Those inequalities can significantly loose the key rate in many practical parameter regimes of a **QKD** protocol.

Fortunately, the key rate problem can be formulated as the optimal value of a convex minimization problem. In general, the size of the problem makes it intractable to find analytical solutions. Therefore, we resort to numerical approaches. The numerical calculation of the key rate can be done by finding a *provable tight lower bound* of this convex optimization problem. This is true for both asymptotic [4, 5] and finite-size regimes [6]. In principle, any (device-dependent) **QKD** protocol can be analyzed in this numerical framework, including measurement-device-independent, and both discrete-variable and continuous-variable protocols. If we are able to solve the problem numerically without loosing tightness, we can potentially provide tighter key rates, even in situations where some valid analytical bounds are known. This is highly relevant for practical implementations of **QKD** protocols as it may enable us to gain key rates without modifying hardwares. This paper continues with the convex optimization approach, and contributes novel robust strategies for numerical calculation of key rates that exhibit strong practical performance.

1.1 Convex Optimization

The secret key rate convex optimization calculations can be typically performed after introducing suitable tools to reduce the dimension of the problem, e.g., the squashing models [7], or the dimension reduction method [8]. These tools are necessary as we are interested in **QKD** protocols that have quantum optical implementations, and thus work with infinite-dimensional Hilbert spaces corresponding to optical modes. In reality, the success of this security proof method is often limited by computational resources, as well as the efficiency and accuracy of underlying algorithms. For a specific protocol, it is also possible to further reduce the dimension of the key rate calculation problem by using properties of the protocol. For example, as done in [7, 9], one can without loss of generality consider only block-diagonal density matrices in the feasible set due to the block-diagonal structure of measurement operators. Then by utilizing properties of the objective function, we can split the key rate calculation problem into several convex optimization problems for individual blocks. Our goal here is to develop a tool that works for

general protocols. In applying our method, one can always choose the problem formulation after applying all applicable aforementioned methods to reduce the dimension as the starting point. Furthermore, it should be noted that the finite-size key rate problem involves a variation of the asymptotic key rate formulation. For the simplicity of our discussion, we focus on the asymptotic formulation in this paper.

The work in [5] provides a reliable framework to compute the key rate using a two-step routine. In the first step, one tries to efficiently find a near optimal, feasible point, of the optimization problem; see (2.2) for the explicit problem formulation. In the second step, one then obtains a reliable lower bound from this feasible point by a linearization and duality argument. In terms of numerical computation, the bottleneck of this approach for large-size **QKD** problems comes from the first step, as it involves semidefinite optimization with a nonlinear objective function. In particular, the work in [5] proposes an algorithm based on the Frank-Wolfe method to solve the first step. However, this method can converge slowly in practice. We note that in Faybusovich and Zhou [10], they also work on providing a more efficient algorithm based on a long-step path-following interior-point method for the **QKD** key rate calculation problem. However, their discussions are restricted to real symmetric matrices, while for **QKD** key rate calculations, it is important to handle Hermitian matrices. Although it might be possible to extend the algorithm in [10] to deal with Hermitian matrices, currently the extension is not done and thus, we cannot directly compare our algorithms with theirs. In addition, the problem formulations used in [5, 10] do not guarantee positive definiteness of the matrices involved in the objective function. Therefore, they perturb current feasible solutions by adding a small multiple of the identity matrix. This perturbation is not required in our new method in this paper due to our regularization using facial reduction, **FR**. [1]

In this paper, we derive a stable reformulation of the convex semidefinite programming, **SDP**, model for the key rate calculation problem of **QKD**. We use this to derive efficient, accurate, algorithms for the problem, in particular, for finding provable lower bounds for the problem. The stable reformulation is based on a novel facial reduction, **FR**, approach. We exploit the Kronecker structure and do **FR** first for the linear constraints to guarantee a positive definite, strictly feasible solution. Second we exploit the properties of the completely positive maps and do **FR** on the nonlinear, quantum relative entropy objective function, to guarantee positive definiteness of its arguments. This allows for a Gauss-Newton type interior-point approach that avoids the need for perturbations to obtain positive definiteness, a technique currently used in the literature. The result is high accuracy solutions with provable lower (and upper) bounds for the convex minimization problem. We note that the convex minimization problem is designed to provide a *lower bound* for the key rate.

1.2 Outline and Main Results

In Section 2 we present the preliminary notations and convex analysis tools that we need. In particular, we include details about the linear maps and adjoints and basic *facial reduction,* ***FR***, needed for our algorithms; see Sections 2.4 and 2.5.

The details and need for facial reduction, **FR**, is discussed in Section 3. We perform **FR** due

[1] Following the original submission of this paper, we have obtained access to the code in [10]. We have observed that the code requires a Slater point, as it uses the analytic center as the starting point of the algorithm. Therefore, we could not use it for many of our instances where strict feasibility fails. Moreover, the random problems solved in the paper have positive definite (strictly feasible) optimal solutions that are relatively close to the analytic center. We hope to fully test the code in [10] on our facially reduced problems in a follow-up paper. Hopefully, we can combine the information on efficient evaluations of the Hessian in [10] with our **FR** techniques and improve both algorithmic approaches.

to the loss of strict feasibility for the linear constraints for some classes of instances, as well as the rank deficiency of the images of the linear maps that consist of the nonlinear objective function. The **FR** guarantees that the reformulated model satisfies the strict feasibility of the linear constraints, and the objective function is evaluated at positive definite density matrices. A partial **FR** that stems from singularity encountered from the *reduced density operator constraint* is discussed in Section 3.2.1. A second type of **FR** performed on the completely positive mappings of the objective function is discussed in Section 3.2.2. These **FR** steps result in a much simplified reformulated model (3.19) for which strict feasibility holds and the objective function arguments preserve positive definiteness. This allows for efficient accurate evaluation of the objective function that uses a spectral decomposition and avoids the less accurate matrix log function evaluation. In addition, we discuss the differentiability of the objective function, both first and second order, in Appendix A.3.

In Section 4 we begin with the optimality conditions and a projected Gauss-Newton, **GN**, interior point method. This uses the modified objective function that is well defined for positive definite density matrices, $\rho \succ 0$. We use the *stable* **GN** search direction for the primal-dual interior-point, **p-d i-p**, method. This avoids unstable backsolve steps for the search direction. We also use a sparsity preserving nullspace representation for the primal feasibility in Appendix B.3.2. This provides for exact primal feasibility steps during the algorithm. Optimal diagonal precondition for the linear system is presented in Appendix B.3.3.

Our upper and lower bounding techniques are given in Section 4.5. In particular, we provide novel theoretical based lower bounding techniques for the **FR** and original problem in Corollaries 4.8 and 4.10, respectively.[2]

Applications to the security analysis of some selected **QKD** protocols are given in Section 5. This includes comparisons with other codes as well as solutions of problems that could not be solved previously. We include the lower bounds and illustrate its strength by including the relative gaps between lower and upper bounds; and we compare with the analytical optimal values when it is possible to do so.

We provide concluding remarks in Section 6. Technical proofs, further references and results, appear in Appendices A and B. The details for six protocol examples used in our tests are given in Appendix C.

A reader, whose main interest is to use the code released from this work to perform **QKD** security analysis, can safely skip Sections 2.3 to 2.5 after reading Section 2.1. One may refer to Section 2.2 for notations if necessary. The main results of Sections 3 and 4 are summarized at the beginning of those sections. The final model of the key rate problem after regularization via **FR** is given in (3.19). The projected Gauss-Newton interior-point algorithm for this model is presented in Algorithm 1 with a less technical explanation given in Section 4.4. In terms of reliable lower bound for the original key rate problem, one may be interested in Remark 4.9 and Corollary 4.10. One can then safely skip details presented in the rest of Sections 3 and 4 and proceed with Section 5 to compare numerical performance of our method with other existing approaches.

2 Preliminaries

We now continue with the terminology and preliminary background for the paper as well as presenting the notations. Sections 2.3 to 2.5 contain the convex optimization background material to understand details of our problem reformulation and our projected Gauss-Newton interior-point algorithm. We also include pointers to additional convex optimization background that

[2]This appears to be a novel contribution for general nonlinear convex **SDP** optimization.

appears in Appendix A.

2.1 QKD key rate calculation background

The asymptotic key rate R^∞ is given by the Devetak-Winter formula [11] that can be written in the following form [5]:

$$R^\infty = \min_\rho D(\mathcal{G}(\rho)\|\mathcal{Z}(\mathcal{G}(\rho))) - p_{\text{pass}}\delta_{\text{EC}}, \tag{2.1}$$

where $D(\delta\|\sigma) =: f(\delta,\sigma) = \text{Tr}\left(\delta(\log\delta - \log\sigma)\right)$ is the quantum relative entropy, p_{pass} is the probability that a given signal is used for the key generation rounds, and δ_{EC} is the cost of error correction per round. The last two parameters are directly determined by observed data and thus are not a part of the optimization. Thus, the essential task of the quantum key distribution rate computation is to solve the following nonlinear convex semidefinite program:

$$\begin{array}{ll} \min_\rho & D(\mathcal{G}(\rho)\|\mathcal{Z}(\mathcal{G}(\rho))) \\ \text{s.t.} & \Gamma(\rho) = \gamma, \\ & \rho \succeq 0, \end{array} \tag{2.2}$$

where $\Gamma : \mathbb{H}^n \to \mathbb{R}^m$ is a linear map defined by $\Gamma(\rho) = (\text{Tr}(\Gamma_i\rho))$; $\mathbb{H}^n$ is the linear space of Hermitian matrices over the reals; and $\gamma \in \mathbb{R}^m$. In this problem, $\{\Gamma_i\}$ is a set of Hermitian matrices corresponding to physical observables. The data pairs Γ_i, γ_i are known observation statistics that include the $\text{Tr}(\rho) = 1$ constraint. The maps $\mathcal{G}$ and $\mathcal{Z}$ are linear, completely positive maps that are specified according to the description of a **QKD** protocol. In general, $\mathcal{G}$ is trace-non-increasing, while $\mathcal{Z}$ is trace-preserving and its Kraus operators are a resolution of identity. The maps are usually represented via the so-called operator-sum (or Kraus operator) representation. (More details on these representation are given below as needed; see also Definition 3.1.)

In other words, the optimization problem is of the form in (2.3):

$$\min\{f(\rho) : \Gamma(\rho) = \gamma,\ \rho \succeq 0\}, \tag{2.3}$$

where the objective function f is the quantum relative entropy function as shown in (2.1), and the constraint set is a spectrahedron, i.e., the intersection of an affine manifold and the positive semidefinite cone. The affine manifold is defined using the linear map for the linear equality constraints in (2.3):

$$\Gamma(\rho) = \left(\text{Tr}(\Gamma_i\rho)\right), i = 1,\ldots,m, \quad \Gamma : \mathbb{H}^n \to \mathbb{R}^m.$$

These are divided into two sets: the observational and reduced density operator constraint sets, i.e., $S_O \cap S_R$.

The set of states ρ satisfying the *observational constraints* is given by

$$S_O = \left\{\rho \succeq 0 \ :\ \langle P_s^A \otimes P_t^B, \rho\rangle = p_{st},\ \forall st\right\}, \tag{2.4}$$

where we let n_A, n_B be the sizes of $P_s^A \in \mathbb{H}^{n_A}, P_t^B \in \mathbb{H}^{n_B}$, respectively; and we denote the *Kronecker product,* $\otimes$. We set $n = n_A n_B$ *which is the size of* ρ.

The set of states ρ satisfying the constraints with respect to the *reduced density operator,* ρ_A, is

$$\begin{array}{rcl} S_R & = & \{\rho \succeq 0 \ :\ \text{Tr}_B(\rho) = \rho_A\} \\ & = & \{\rho \succeq 0 \ :\ \langle \Theta_j \otimes \mathbb{1}_B, \rho\rangle = \theta_j,\ \forall j = 1,\ldots,m_R\}, \end{array} \tag{2.5}$$

where $\theta_j = \langle \Theta_j, \rho_A\rangle$ and $\{\Theta_j\}$ forms an orthonormal basis for the real vector space of Hermitian matrices on system A. This implicitly defines the linear map and constraint in $\text{Tr}_B(\rho) = \rho_A$. Here we denote the identity matrix $\mathbb{1}_B \in \mathbb{H}^{n_B}$.

Here, we may assume that $\Gamma_1 = I$ and $\gamma_1 = 1$ to guarantee that we restrict our variables to *density matrices*, i.e., semidefinite and unit trace. (See [12, Theorem 2.5].)

2.2 Notations

We use $\mathbb{C}^{n\times n}$ to denote the space of n-by-n complex matrices, and $\mathbb{H}^n$ to denote the *subset* of n-by-n Hermitian matrices; we use $\mathbb{H}$ when the dimension is clear. We use $\mathbb{S}^n, \mathbb{S}$ for the subspaces of $\mathbb{H}^n$ of real symmetric matrices. Given a matrix $X \in \mathbb{C}^{n\times n}$, we use $\Re(X)$ and $\Im(X)$ to denote the real and the imaginary parts of X, respectively. We use $\mathbb{H}^n_+, \mathbb{S}^n_+$ ($\mathbb{H}^n_{++}, \mathbb{S}^n_{++}$, resp) to denote the positive semidefinite cone (the positive definite cone, resp); and again we leave out the dimension when it is clear. We use the partial order notations $X \succeq 0, X \succ 0$ for semidefinite and definite, respectively. We let $\mathbb{R}^n$ denote the usual vector space of real n-coordinates; $\mathcal{P}_C(X)$ denotes the projection of X onto the closed convex set C. For a matrix X, we use range(X) and null(X) to denote the *range* and the *nullspace* of X, respectively. We let BlkDiag$(A_1, A_2, \ldots, A_k)$ denote the block diagonal matrix with diagonal blocks A_i.

2.3 Real Inner Product Space $\mathbb{C}^{n\times n}$

In general, $\mathbb{H}^n$ is not a subspace of $\mathbb{C}^{n\times n}$ unless we treat both as vector spaces over $\mathbb{R}$. To do this we define a *real inner product in* $\mathbb{C}^{n\times n}$ that takes the standard inner products of the real and imaginary parts:

$$\begin{aligned} \langle Y, X\rangle &= \langle \Re(Y), \Re(X)\rangle + \langle \Im(Y), \Im(X)\rangle \\ &= \mathrm{Tr}\left(\Re(Y)^\dagger \Re(X)\right) + \mathrm{Tr}\left(\Im(Y)^\dagger \Im(X)\right) \\ &= \Re\left(\mathrm{Tr}(Y^\dagger X)\right). \end{aligned} \tag{2.6}$$

We note that

$$\Re(\langle Y, X\rangle) = \langle \Re(Y), \Re(X)\rangle + \langle \Im(Y), \Im(X)\rangle, \quad \Im(\langle Y, X\rangle) = -\langle \Re(Y), \Im(X)\rangle + \langle \Im(Y), \Re(X)\rangle.$$

Over the reals, $\dim(\mathbb{H}^n) = n^2, \dim(\mathbb{C}^{n\times n}) = 2n^2$. The induced norm is the Frobenius norm $\|X\|_F^2 = \langle X, X\rangle = \mathrm{Tr}\left(X^\dagger X\right)$, where we denote the *conjugate transpose,* $\cdot^\dagger$.

2.4 Linear Transformations and Adjoints

Given a linear map $\mathcal{L} : \mathcal{D} \to \mathcal{R}$, we call the unique linear map $\mathcal{L}^\dagger : \mathcal{R} \to \mathcal{D}$ the *adjoint* of $\mathcal{L}$, if it satisfies

$$\langle \mathcal{L}(X), Y\rangle = \left\langle X, \mathcal{L}^\dagger(Y)\right\rangle, \forall X \in \mathcal{D}, Y \in \mathcal{R}.$$

Often in our study, we use vectorized computations instead of using complex matrices directly. In order to relieve the computational burden, we use isomorphic and isometric realizations of matrices by ignoring the redundant entries. We consider $\mathbb{H}^n$ as a vector space of dimension n^2 over the reals. We define $\mathrm{Hvec}(H) \in \mathbb{R}^{n^2}$ by stacking diag(H) followed by $\sqrt{2}$ times the strict upper triangular parts of $\Re(H)$ and $\Im(H)$, both columnwise:

$$\mathrm{Hvec}(H) = \begin{pmatrix} \mathrm{diag}(H) \\ \sqrt{2}\,\Re(\mathit{upper}(H)) \\ \sqrt{2}\,\Im(\mathit{upper}(H)) \end{pmatrix} \in \mathbb{R}^{n^2}, \quad \mathrm{HMat} = \mathrm{Hvec}^{-1} = \mathrm{Hvec}^\dagger.$$

We note that for the real symmetric matrices $\mathbb{S}^n$, we can use the first *triangular number,* $t(n) = n(n+1)/2$ of elements in Hvec, and we denote this by $\mathrm{svec}(S) \in \mathbb{R}^{t(n)}$, with adjoint sMat.

We use various linear maps in an **SDP** framework. For given $\Gamma_i \in \mathbb{H}^n, i = 1, \ldots, m$, define

$$\Gamma : \mathbb{H}^n \to \mathbb{R}^m \text{ by } \Gamma(H) = (\langle \Gamma_i, H\rangle)_i \in \mathbb{R}^m.$$

The adjoint satisfies

$$\langle \Gamma(H), y \rangle = \sum_i y_i \operatorname{Tr}(\Gamma_i H) = \operatorname{Tr}\left(H \left(\sum_i y_i \Gamma_i \right) \right) = \left\langle H, \Gamma^\dagger(y) \right\rangle .$$

The matrix representation A of Γ is found from

$$(A \operatorname{Hvec}(H))_i = (\Gamma(H))_i = \langle \Gamma_i, H \rangle = \langle \operatorname{Hvec}(\Gamma_i), \operatorname{Hvec}(H) \rangle ,$$

i.e., for $g_i = \operatorname{Hvec}(\Gamma_i),\ \forall i$ and $h = \operatorname{Hvec}(H)$,

$$\Gamma(H) \equiv A(h), \quad \text{where } A = \begin{bmatrix} g_1^\dagger \\ \vdots \\ g_m^\dagger \end{bmatrix} .$$

Specialized adjoints for matrix multiplication are given in Appendix A.1.

2.5 Cones, Faces, and Facial Reduction, **FR**

The facial structure of the semidefinite cone is well understood. We outline some of the concepts we need for facial reduction and exposing vectors (see e.g., [13]). We recall that a *convex cone* K is defined by: $\lambda K \subseteq K, \forall \lambda \geq 0,\ K + K \subset K$, i.e., it is a cone and so contains all rays, and it is a convex set. For a set $S \subseteq \mathbb{H}$ we denote the *dual cone*, $S^\dagger = \{\phi \in \mathbb{H} : \langle \phi, s \rangle \geq 0,\ \forall s \in S\}$.

Definition 2.1 (*face*)**.** *A convex cone F is a face of a convex cone K, denoted $F \trianglelefteq K$, if*

$$x, y \in K, x + y \in F \implies x, y \in F.$$

Equivalently, for a general convex set K and convex subset $F \subseteq K$, we have $F \trianglelefteq K$, if

$$[x, y] \subset K, z \in \operatorname{relint}[x, y], z \in F \implies [x, y] \subset F,$$

where $[x, y]$ denotes the line segment joining x, y.

Faces of the positive semidefinite cone are characterized by the range or nullspace of any element in the relative interior of the faces. In fact, the following characterizations in Lemma 2.2 hold.

Lemma 2.2. *Let F be a convex subset of $\mathbb{H}_+^n$ with $X \in \operatorname{relint} F$. Let*

$$X = \begin{bmatrix} P & Q \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P & Q \end{bmatrix}^\dagger$$

be the orthogonal spectral decomposition with $D \in \mathbb{H}_{++}^r$. Then the following are equivalent:

1. $F \trianglelefteq \mathbb{H}_+^n$;
2. $F = \{Y \in \mathbb{H}_+^n : \operatorname{range}(Y) \subset \operatorname{range}(X)\} = \{Y \in \mathbb{H}_+^n : \operatorname{null}(Y) \supset \operatorname{null}(X)\}$;
3. $F = P \mathbb{H}_+^r P^\dagger$;
4. $F = \mathbb{H}_+^n \cap (QQ^\dagger)^\perp$.

The matrix P, in Item 3 of Lemma 2.2, allows us to represent any matrix $Y \in F \trianglelefteq \mathbb{H}^n_+$ as a *compact spectral decomposition*, i.e., $Y = PDP^\dagger$ with diagonal $D \in \mathbb{H}^r_+$. This compact representation leads to a reduction in the variable dimension. The matrix $QQ^\dagger$, in Item 4 of Lemma 2.2, is called an *exposing vector* for the face F. Exposing vectors come into play throughout Section 3.

Definition 2.3 (minimal face)**.** *Let K be a closed convex cone and let $X \subseteq K$. Then* $\operatorname{face}(X) \trianglelefteq K$ *is the* minimal face*, the intersection of all faces of K that contain X.*

Facial reduction is a process of identifying the minimal face of $\mathbb{H}^n_+$ containing the spectrahedron $\{\rho : \Gamma(\rho) = \gamma\} \cap \mathbb{H}^n_+$. Lemma 2.4 plays an important role in the heart of the facial reduction process. Essentially, either there exists a strictly feasible ρ, or the alternative holds that there exists a linear combination of the Γ_i that is positive semidefinite but has a zero expectation.

Lemma 2.4 (theorem of the alternative, [13, Theorem 3.1.3])**.** *For the feasible constraint system in (2.3), exactly one of the following statements holds:*

1. *there exists $\rho \succ 0$ such that $\Gamma(\rho) = \gamma$;*
2. *there exists y such that*
$$0 \neq \Gamma^\dagger(y) \succeq 0 \ , \ \langle \gamma, y \rangle = 0. \tag{2.7}$$

In Lemma 2.4, the matrix $\Gamma^\dagger(y)$ is an exposing vector for the face containing the constraint set in (2.3).

3 Problem Formulations and Facial Reduction

The original problem formulation is given in (2.2). Without loss of generality we can assume that the feasible set, a *spectrahedron*, is nonempty. This is because our problem is related to a physical scenario, and we can trivially set the key rate to be zero when the feasible set is empty. Note that the Hermitian (positive semidefinite, density) matrix ρ is the only variable in the above (2.2) optimization problem. Motivated by the fact that the mappings $\mathcal{G}, \mathcal{Z} \circ \mathcal{G}$ are positive semidefinite preserving but possibly not positive definite preserving, we rewrite (2.2) as follows:[3]

$$\begin{array}{ll} \min_{\rho,\sigma,\delta} & \operatorname{Tr}(\delta(\log \delta - \log \sigma)) \\ \text{s.t.} & \Gamma(\rho) = \gamma \\ & \sigma = \mathcal{Z}(\delta) \\ & \delta = \mathcal{G}(\rho) \\ & \rho, \sigma, \delta \succeq 0. \end{array} \tag{3.1}$$

Due to the structure of the linear mapping $\mathcal{G}$, the matrix δ is often singular in (3.1). Therefore, strict feasibility fails in (3.1). This indicates that the objective function, the *quantum relative entropy function* is evaluated on singular matrices in both (2.2) and (3.1), creating theoretical and numerical difficulties. In fact, the domain of the problem that guarantees finiteness for the objective function, requires restrictions on the ranges of the linear mappings. By moving back and forth between equivalent formulations of the types in (2.2) and (3.1), we derive a regularized model that simplifies type (2.2), and where positive definiteness is preserved. In particular, the regularization allows for an efficient interior point method even though the objective function is not differentiable on the boundary of the semidefinite cone. This allows for efficient algorithmic developments. In addition, this enables us to accurately solve previously intractable problems.

[3] This allows us to regularize below using facial reduction, **FR**.

We now present the details on various formulations of **QKD** from (2.2) and (3.1). We show that facial reduction allows for regularization of both the constraints and the objective function, i.e., this means that we have positive *definite* feasible points, and a proper domain for the objective function with positive definite matrices. This obviates the need for adding perturbations of the identity. We include results about **FR** for positive transformations and show that highly accurate **FR** can be done in these cases.

In particular, we provide a regularized reformulation of our problem, see (3.19), and the regularization statement that guarantees positive definiteness, see (3.20).

3.1 Properties of Objective Function and Mappings $\mathcal{G}, \mathcal{Z}$

The *quantum relative entropy function* $D : \mathbb{H}^n_+ \times \mathbb{H}^n_+ \to \mathbb{R}_+ \cup \{+\infty\}$ is denoted by $D(\delta||\sigma)$, and is defined as

$$D(\delta||\sigma) = \begin{cases} \operatorname{Tr}(\delta \log \delta) - \operatorname{Tr}(\delta \log \sigma) & \text{if } \operatorname{range}(\delta) \cap \operatorname{null}(\sigma) = \emptyset \\ \infty & \text{otherwise.} \end{cases} \tag{3.2}$$

That the quantum relative entropy D is finite if $\operatorname{range}(\delta) \subseteq \operatorname{range}(\sigma)$ is shown by extending the matrix log function to be 0 on the nullspaces of δ, σ. (See [14, Definition 5.18].) It is known that D is nonnegative, equal to 0 if, and only if, $\delta = \sigma$, and is jointly convex in both δ and σ, see [12, Section 11.3].

Definition 3.1. *The linear map $\mathcal{G} : \mathbb{H}^n \to \mathbb{H}^k$ is defined as a sum of matrix products (Kraus representation)*

$$\mathcal{G}(\rho) := \sum_{j=1}^{\ell} K_j \rho K_j^\dagger, \tag{3.3}$$

where $K_j \in \mathbb{C}^{k\times n}$ and $\sum_{j=1}^{\ell} K_j^\dagger K_j \preceq I$. The adjoint is $\mathcal{G}^\dagger(\delta) := \sum_{j=1}^{\ell} K_j^\dagger \delta K_j$.

Typically we have $k > n$ with k being a multiple of n; and thus we can have $\mathcal{G}(\rho)$ rank deficient for all $\rho \succ 0$.

Definition 3.2. *The self-adjoint (projection) linear map $\mathcal{Z} : \mathbb{H}^k \to \mathbb{H}^k$ is defined as the sum*

$$\mathcal{Z}(\delta) := \sum_{j=1}^{N} Z_j \delta Z_j, \tag{3.4}$$

where $Z_j = Z_j^2 = Z_j^\dagger \in \mathbb{H}^k_+$ and $\sum_{j=1}^{N} Z_j = I_k$.

Since $\sum_{j=1}^{N} Z_j = I_k$, the set $\{Z_i\}_{j=1}^{N}$ is a *spectral resolution of I*. Proposition 3.3 below states some interesting properties of the operator $\mathcal{Z}$; see also [15, Appendix C, (C1)].

Proposition 3.3. *The linear map $\mathcal{Z}$ in Definition 3.2 is an orthogonal projection on $\mathbb{H}^k$. Moreover, for $\delta \succeq 0$,*

$$\operatorname{Tr}\left(\delta \log \mathcal{Z}(\delta)\right) = \operatorname{Tr}\left(\mathcal{Z}(\delta) \log \mathcal{Z}(\delta)\right). \tag{3.5}$$

Proof. First we show that the matrices of $\mathcal{Z}$ satisfy

$$Z_i Z_j = 0, \ \forall\ i \neq j. \tag{3.6}$$

For $i, j \in \{1, \ldots, N\}$, we have by Definition 3.2 that

$$\begin{array}{rcl} Z_i \left(\sum_{s=1}^{N} Z_s\right) Z_i = Z_i I_k Z_i = Z_i & \implies & 0 = \sum_{s\neq i} Z_i Z_s Z_i = \sum_{s\neq i} (Z_s Z_i)^\dagger (Z_s Z_i) \\ & \implies & Z_j Z_i = 0, \ \forall j \neq i. \end{array} \tag{3.7}$$

We now have $\mathcal{Z} = \mathcal{Z}^2 = \mathcal{Z}^{1/2} = \mathcal{Z}^\dagger$. Thus, $\mathcal{Z}$ is an orthogonal projection. The equality (3.5) holds by the properties of the map $\mathcal{Z}$ that it removes the off-diagonal blocks in its image. □

Using (3.2), Lemma 3.4 below shows that the objective value of the model (2.2) is finite on the feasible set. This also provides insight on the usefulness of **FR** on the variable σ done below.

Lemma 3.4. *Let* $X \succeq 0$. *Then* $\text{range}(X) \subseteq \text{range}(\mathcal{Z}(X))$.

Proof. See Appendix A.2. □

Remark 3.5. *In general, the mapping* $\mathcal{G}$ *in* (3.3) *does not preserve positive definiteness. Therefore the objective function* $f(\rho)$, *see* (A.9) *below, may need to evaluate* $\text{Tr}(\delta \log \delta)$ *and* $\text{Tr}(\delta \log \sigma)$ *with* both $\delta = \mathcal{G}(\rho)$ *and* $\sigma = \mathcal{Z} \circ \mathcal{G}(\rho)$ *always singular. Although the objective function* f *is well-defined at singular points* δ, σ, *the gradient of* f *at singular points* δ, σ *is not well-defined. Our approach using* ***FR*** *within an interior point method avoids these numerical difficulties.*

3.2 Reformulation via Facial Reduction (FR)

Using Proposition 3.3, we can now reformulate the objective function in the key rate optimization problem (3.1) to obtain the following equivalent model:

$$\begin{array}{ll} \min_{\rho,\sigma,\delta} & \text{Tr}(\delta \log \delta) - \text{Tr}(\sigma \log \sigma) \\ \text{s.t.} & \Gamma(\rho) = \gamma \\ & \sigma - \mathcal{Z}(\delta) = 0 \\ & \delta - \mathcal{G}(\rho) = 0 \\ & \rho \in \mathbb{H}^n_+,\ \sigma \in \mathbb{H}^k_+,\ \delta \in \mathbb{H}^k_+. \end{array} \tag{3.8}$$

The new objective function is the key in our analysis, as it simplifies the expressions for gradient and Hessian. Next, we derive facial reduction based on the constraints in (3.8).

3.2.1 Partial FR on the Reduced Density Operator Constraint

Consider the spectrahedron S_R defined by the reduced density operator constraint in (2.5). We now simplify the problem via **FR** by using only (2.5) in the case that $\rho_A \in \mathbb{H}^{n_A}$ is singular. We now see in Theorem 3.6 that we can do this explicitly using the spectral decomposition of ρ_A; see also [16, Sec. II]. Therefore, this step is extremely accurate. Using the structure arising from the reduced density operator constraint, we obtain partial **FR** on the constraint set in Theorem 3.6.

Theorem 3.6. *Let* $\text{range}(P) = \text{range}(\rho_A) \subsetneq \mathbb{H}^{n_A}$, $P^\dagger P = \mathbb{1}_r$ *for* $r < n_A$, *and let* $V = P \otimes \mathbb{1}_B$. *Then the spectrahedron* S_R *in* (2.5) *has the property that*

$$\rho \in S_R \implies \rho = VRV^\dagger, \text{ for some } R \in \mathbb{H}^{r \cdot n_B}_+. \tag{3.9}$$

Proof. Let $\begin{bmatrix} P & Q \end{bmatrix}$ be a unitary matrix such that $\text{range}(P) = \text{range}(\rho_A)$ and $\text{range}(Q) = \text{null}(\rho_A)$. Let $W = QQ^\dagger \succeq 0$. Recall that the adjoint $\text{Tr}_B^\dagger(W) = W \otimes \mathbb{1}_B$. Then $\rho \in S_R$ implies that

$$\langle W \otimes \mathbb{1}_B, \rho \rangle = \langle W, \text{Tr}_B(\rho) \rangle = \langle W, \rho_A \rangle = 0, \tag{3.10}$$

where $\mathbb{1}_B \in \mathbb{H}^{n_B}$ is the identity matrix of size n_B, and we use (2.5) to guarantee that $\text{Tr}_B(\rho) = \rho_A$. Therefore, $W \otimes \mathbb{1}_B \succeq 0$ is an exposing vector for the spectrahedron S_R in (2.5). And we can write $\rho = VRV^\dagger$ with $V = P \otimes \mathbb{1}_B$ for any $\rho \in S_R$. This yields an equivalent representation (3.9) with a smaller positive semidefinite constraint.[4] □

[4] We provide a self-contained alternate proof in Appendix A.4.

We emphasize that facial reduction is not only powerful in reducing the variable dimension, but also in reducing the number of constraints. Indeed, if ρ_A is not full-rank, then at least one of the constraints in (2.5) becomes redundant and can be discarded; see [17, 18]. In this case, it is equivalent to the matrix ρ_A becoming smaller in dimension. (Our empirical observations show that many of the other observational constraints $\Gamma_i(\rho) = \gamma_i$ also become redundant and can be discarded.)

3.2.2 FR on the Constraints Originating from $\mathcal{G}, \mathcal{Z}$

Our motivation is that the domain of the objective function may be restricted to the boundary of the semidefinite cone, i.e., the matrices $\mathcal{G}(\rho), \mathcal{Z}(\mathcal{G}(\rho))$ are singular by the definition of $\mathcal{G}$. We would like to guarantee that we have a well-formulated problem with strictly feasible points in the domain of the objective function so that the derivatives are well-defined. This guarantees basic numerical stability. This is done by considering the constraints in the equivalent formulation in (3.1).

We first note the useful equivalent form for the entropy function.

Lemma 3.7. *Let $Y = VRV^\dagger \in \mathbb{H}_+$, $R \succ 0$ be the compact spectral decomposition of Y with $V^\dagger V = I$. Then*

$$\operatorname{Tr}(Y \log Y) = \operatorname{Tr}(R \log R).$$

Proof. We obtain a unitary matrix $U = \begin{bmatrix} V & P \end{bmatrix}$ by completing the basis. Then $Y = UDU^\dagger$, where $D = \operatorname{BlkDiag}(R, 0)$. We conclude, with $0 \cdot \log 0 = 0$, that $\operatorname{Tr} Y \log Y = \operatorname{Tr} D \log D = \operatorname{Tr} R \log R$. □

We use the following simple result to obtain the exposing vectors of the minimal face in the problem analytically.

Lemma 3.8. *Let $\mathcal{C} \subseteq \mathbb{H}^n_+$ be a given convex set with nonempty interior. Let $Q_i \in \mathbb{H}^{k\times n}, i = 1, \ldots, t$, be given matrices. Define the linear map $\mathcal{A} : \mathbb{H}^n \to \mathbb{H}^k$ and matrix $V \in \mathbb{C}^{k\times r}$ by*

$$\mathcal{A}(X) = \sum_{i=1}^{t} Q_i X Q_i^\dagger, \quad \operatorname{range}(V) = \operatorname{range}\left(\sum_{i=1}^{t} Q_i Q_i^\dagger\right).$$

Then the minimal face,

$$\operatorname{face}(\mathcal{A}(\mathcal{C})) = V \mathbb{H}^r_+ V^\dagger.$$

Proof. First, note that properties of the mapping implies that $\mathcal{A}(\mathcal{C}) \subset \mathbb{H}^k_+$. Nontrivial exposing vectors $0 \neq W \in \mathbb{H}^n_+$ of $\mathcal{A}(\mathcal{C})$ can be characterized by the null space of the adjoint operator $\mathcal{A}^\dagger$:

$$\begin{aligned}
0 \neq W \in \mathbb{H}^n_+, \langle W, \mathcal{A}(\mathcal{C})\rangle = 0 \quad &\Longleftrightarrow \quad 0 \neq W \succeq 0, \langle W, Y\rangle = 0, \forall\, Y \in \mathcal{A}(\mathcal{C}) \\
&\Longleftrightarrow \quad 0 \neq W \succeq 0, \langle \mathcal{A}^\dagger(W), X\rangle = 0, \forall\, X \in \mathcal{C} \\
&\Longleftrightarrow \quad 0 \neq W \succeq 0, W \in \operatorname{null}(\mathcal{A}^\dagger) \\
&\Longleftrightarrow \quad 0 \neq W \succeq 0, Q_i^\dagger W Q_i = 0, \forall i, \\
&\Longleftrightarrow \quad 0 \neq \operatorname{range}(W) \subseteq \operatorname{null}\left(\textstyle\sum_i Q_i Q_i^\dagger\right),
\end{aligned}$$

where the third equivalence follows from $\operatorname{int}(\mathcal{C}) \neq \emptyset$, and the fourth equivalence follows from the properties of the sum of mappings of a semidefinite matrix.

The choice of V follows from choosing a maximal rank exposing vector and constructing V using Lemma 2.2:

$$\operatorname{range}(V) = \operatorname{null}(W) = \operatorname{range}\left(\sum_i Q_i Q_i^\dagger\right).$$

□

We emphasize that the minimal face in Lemma 3.8 means that V has a minimum number of columns, as without loss of generality, we choose it to be full column rank. In other words, this is the greatest reduction in the dimension of the image. In addition, the exposing vectors of $\mathcal{A}(\mathcal{C})$ are characterized by the positive semidefinite matrices in the null space of $\mathcal{A}^\dagger$. This implies that **FR** can be done in one step.

We describe how to apply Lemma 3.8 to obtain $V_\rho, V_\delta, V_\sigma$ of the minimal face of $(\mathbb{H}^n_+, \mathbb{H}^k_+, \mathbb{H}^k_+)$ containing the feasible region of (3.8). By Lemma 2.2, we may write

$$\begin{array}{rcl} \rho & = & V_\rho R_\rho V_\rho^\dagger \in \mathbb{H}^n_+, \quad R_\rho \in \mathbb{H}^{n_\rho}_+ \\ \delta & = & V_\delta R_\delta V_\delta^\dagger \in \mathbb{H}^k_+, \quad R_\delta \in \mathbb{H}^{k_\delta}_+ \\ \sigma & = & V_\sigma R_\sigma V_\sigma^\dagger \in \mathbb{H}^k_+, \quad R_\sigma \in \mathbb{H}^{k_\sigma}_+. \end{array}$$

Define the linear maps

$$\begin{array}{rcl} \Gamma_V : \mathbb{H}^{n_\rho}_+ \to \mathbb{R}^m & \text{by} & \Gamma_V(R_\rho) = \Gamma(V_\rho R_\rho V_\rho^\dagger), \\ \mathcal{G}_V : \mathbb{H}^{n_\rho}_+ \to \mathbb{H}^k_+ & \text{by} & \mathcal{G}_V(R_\rho) = \mathcal{G}(V_\rho R_\rho V_\rho^\dagger), \\ \mathcal{Z}_V : \mathbb{H}^{k_\delta}_+ \to \mathbb{H}^k_+ & \text{by} & \mathcal{Z}_V(R_\delta) = \mathcal{Z}(V_\delta R_\delta V_\delta^\dagger). \end{array}$$

The matrices $V_\rho, V_\delta, V_\sigma$ are obtained as follows.

1. We apply **FR** to $\mathcal{F}_\rho := \{\rho \in \mathbb{H}^n_+ : \Gamma(\rho) = \gamma\}$ to find V_ρ for the minimal face, $\text{face}(\mathcal{F}_\rho) \trianglelefteq \mathbb{H}^n_+$.

2. Define
$$\mathcal{R}_\rho := \{R_\rho \in \mathbb{H}^{n_\rho}_+ : \Gamma_V(R_\rho) = \gamma\}.$$
Note that $\text{int}(\mathcal{R}_\rho) \neq \emptyset$. Applying Lemma 3.8 to $\mathcal{F}_\delta := \{\mathcal{G}_V(R_\rho) \in \mathbb{H}^k_+ : R_\rho \in \mathcal{R}_\rho\}$, the matrix V_δ yields the minimal face, $\text{face}(\mathcal{F}_\delta) \trianglelefteq \mathbb{H}^k_+$ if we choose
$$\text{range}(V_\delta) = \text{range}\left(\mathcal{G}_V(I)\right). \tag{3.11}$$

3. Define
$$\mathcal{R}_\delta := \{R_\delta \in \mathbb{H}^{k_\delta}_+ : V_\delta R_\delta V_\delta^\dagger = \mathcal{G}_V(R_\rho),\ R_\rho \in \mathcal{R}_\rho\}.$$
We again note that $\text{int}(\mathcal{R}_\delta) \neq \emptyset$. Applying Lemma 3.8 to $\mathcal{F}_\sigma := \{\mathcal{Z}_V(R_\delta) \in \mathbb{H}^k_+ : R_\delta \in \mathcal{R}_\delta\}$, we find the matrix V_σ representing the minimal face, $\text{face}(\mathcal{F}_\sigma) \trianglelefteq \mathbb{H}^k_+$. Thus, we choose V_σ satisfying
$$\text{range}(V_\sigma) = \text{range}\left(\mathcal{Z}_V(I)\right). \tag{3.12}$$

As above, this also can be seen by looking at the image of I and the relative interior of the range of $\mathcal{Z}_V$. We note, by Lemma 3.4, that $\text{range}(V_\sigma) \supseteq \text{range}(V_\delta)$. Note that we have assumed the exposing vector of maximal rank for the original constraint set on ρ in the first step is obtained. Without loss of generality, we can assume that the columns in $V_\rho, V_\delta, V_\sigma$ are orthonormal. This makes the subsequent computation easier.

Assumption 3.9. *Without loss of generality, we assume $V_M^\dagger V_M = I$ for $M = \rho, \delta, \sigma$.*

Define $\mathcal{V}_\delta(R_\delta) := V_\delta R_\delta V_\delta^\dagger$ and $\mathcal{V}_\sigma(R_\sigma) := V_\sigma R_\sigma V_\sigma^\dagger$. Applying Lemma 3.7 and substituting for ρ, δ, σ to (3.8), we obtain the equivalent formulation (3.13).

$$\begin{array}{ll} \min & \text{Tr}(R_\delta \log(R_\delta)) - \text{Tr}\left(R_\sigma \log(R_\sigma)\right) \\ \text{s.t.} & \Gamma_V(R_\rho) = \gamma \\ & \mathcal{V}_\sigma(R_\sigma) - \mathcal{Z}_V(R_\delta) = 0 \\ & \mathcal{V}_\delta(R_\delta) - \mathcal{G}_V(R_\rho) = 0 \\ & R_\rho, R_\sigma, R_\delta \succeq 0. \end{array} \tag{3.13}$$

After facial reduction, many of the linear equality constraints in (3.13) end up being redundant. We may delete redundant constraints and keep a well-conditioned equality constraints. In the next section, we show that the removal of the redundant constraints can be performed by *rotating* the constraints.

3.2.3 Reduction on the Constraints

Recall that our primal problem after **FR** is given in (3.13). Moreover, by the work above we can assume that Γ_V is surjective. In Theorem 3.10 and Theorem 3.11 below, we show that we can simplify the last two equality constraints in (3.13) by an appropriate rotation.

Theorem 3.10. *Let $R_\rho \in \mathbb{H}_+^{n_\rho}$ and $R_\delta \in \mathbb{H}_+^{k_\delta}$. It holds that*

$$\mathcal{V}_\delta(R_\delta) = \mathcal{G}_V(R_\rho) \quad \Longleftrightarrow \quad R_\delta = \mathcal{G}_{UV}(R_\rho), \tag{3.14}$$

where $\mathcal{G}_{UV}(\cdot) := V_\delta^\dagger \mathcal{G}_V(\cdot) V_\delta$.

Proof. Let P be such that $U = \begin{bmatrix} V_\delta & P \end{bmatrix}$ is unitary. Rotating the first equality in (3.14) using the unitary matrix U yields an equivalent equality $U^\dagger \mathcal{V}_\delta(R_\delta) U = U^\dagger \mathcal{G}_V(R_\rho) U$. Applying the orthogonality of V_δ, the left-hand side above becomes

$$U^\dagger \mathcal{V}_\delta(R_\delta) U = \begin{bmatrix} R_\delta & 0 \\ 0 & 0 \end{bmatrix}. \tag{3.15}$$

From facial reduction, it holds that $\operatorname{range}(V_\delta) = \operatorname{range}(\mathcal{G}_V)$ and thus $P^\dagger \mathcal{G}_V = 0$. Therefore, the right hand-side becomes

$$U^\dagger \mathcal{G}_V(R_\rho) U = \begin{bmatrix} V_\delta^\dagger \\ P^\dagger \end{bmatrix} \mathcal{G}_V(R_\rho) \begin{bmatrix} V_\delta & P \end{bmatrix} = \begin{bmatrix} V_\delta^\dagger \mathcal{G}_V(R_\rho) V_\delta & 0 \\ 0 & 0 \end{bmatrix}. \tag{3.16}$$

□

Theorem 3.11. *Let $R_\sigma \in \mathbb{H}_+^{k_\sigma}$ and $R_\delta \in \mathbb{H}_+^{k_\delta}$. It holds that*

$$\mathcal{V}_\sigma(R_\sigma) = \mathcal{Z}_V(R_\delta) \quad \Longleftrightarrow \quad R_\sigma = \mathcal{Z}_{UV}(R_\delta), \tag{3.17}$$

where $\mathcal{Z}_{UV}(\cdot) := V_\sigma^\dagger \mathcal{Z}_V(\cdot) V_\sigma$.

Proof. Using the unitary matrix $U = \begin{bmatrix} V_\sigma & P \end{bmatrix}$ in the proof of Theorem 3.10, we obtain the statement. □

With Theorems 3.10 and 3.11, we reduce the number of linear constraints in (3.13) as below.

$$\begin{array}{ll} \min & \operatorname{Tr}(R_\delta \log(R_\delta)) - \operatorname{Tr}(R_\sigma \log(R_\sigma)) \\ \text{s.t.} & \Gamma_V(R_\rho) = \gamma \\ & R_\sigma - \mathcal{Z}_{UV}(R_\delta) = 0 \\ & R_\delta - \mathcal{G}_{UV}(R_\rho) = 0 \\ & R_\rho \in \mathbb{H}_+^{n_\rho}, R_\sigma \in \mathbb{H}_+^{k_\sigma}, R_\delta \in \mathbb{H}_+^{k_\delta}. \end{array} \tag{3.18}$$

We emphasize that the images of $\mathcal{Z}_V$ and $\mathcal{G}_V$ in (3.13) are both in $\mathbb{H}^k$ but the images of $\mathcal{Z}_{UV}$ and $\mathcal{G}_{UV}$ in (3.18) are in $\mathbb{H}^{k_\sigma}$ and $\mathbb{H}^{k_\delta}$, respectively, and $k_\delta \leq k_\sigma \leq k$. The facial reduction performed on the variables δ, σ may yield $k_\delta < k_\sigma$. Hence, the two trace terms in the objective function in (3.18) cannot be consolidated into one trace term in general.

Remark 3.12. *The mapping $\mathcal{G}_{UV}$ satisfies the properties for $\mathcal{G}$ in (3.3). However, the properties in (3.4) do not hold for the mapping $\mathcal{Z}_{UV}$.*

3.3 Final Model for QKD key rate calculation

In this section we have a main result, i.e., the main model that we work on and the derivatives. We eliminate some of variables in the model (3.18) to obtain a simplified formulation. Define $\widehat{\mathcal{Z}} := \mathcal{Z}_{UV} \circ \mathcal{G}_{UV}$ and $\widehat{\mathcal{G}} := \mathcal{G}_{UV}$. We substitute $R_\sigma = \widehat{\mathcal{Z}}(R_\rho)$ and $R_\delta = \widehat{\mathcal{G}}(R_\rho)$ back in the objective function in (3.18). For simplification, and by abuse of notation, we set

$$\boxed{\rho \leftarrow R_\rho,\ \sigma \leftarrow R_\sigma,\ \delta \leftarrow R_\delta.}$$

We obtain the final model for **QKD** key rate calculation problem:

$$\begin{array}{rlll} p^* = & \min & f(\rho) = \operatorname{Tr}\left(\widehat{\mathcal{G}}(\rho)(\log \widehat{\mathcal{G}}(\rho))\right) - \operatorname{Tr}\left(\widehat{\mathcal{Z}}(\rho) \log \widehat{\mathcal{Z}}(\rho)\right) \\ & \text{s.t.} & \Gamma_V(\rho) = \gamma_V \\ & & \rho \in \mathbb{H}^{n_\rho}_+, \end{array} \tag{3.19}$$

where $\gamma_V \in \mathbb{R}^{m_V}$ for some $m_V \leq m$. The final model is essentially in the same form as the original model (2.2); see also Proposition 3.3.

Note that the final model now has smaller number of variables compared to the original problem (2.2). Moreover, the objective function f, with the modified linear maps $\widehat{\mathcal{G}}, \widehat{\mathcal{Z}}$, is well-defined and analytic on $\rho \in \mathbb{H}^{n_\rho}_{++}$, i.e., we have

$$\rho \succ 0 \implies \widehat{\mathcal{G}}(\rho) \succ 0 \implies \widehat{\mathcal{Z}}(\rho) \succ 0.^{5} \tag{3.20}$$

Some derivative background is given in Appendix A.3. We conclude this section by presenting the derivative formulae for gradient and Hessian. The simple formulae in Theorem 3.13 are a direct application of Lemma A.3. Throughout Section 4 we work with these derivatives.

Theorem 3.13 (derivatives of regularized objective)**.** *Let $\rho \succ 0$. The gradient of f in (3.19) is*

$$\nabla f(\rho) = \boxed{\widehat{\mathcal{G}}^\dagger(\log[\widehat{\mathcal{G}}(\rho)]) + \widehat{\mathcal{G}}^\dagger(I)} - \boxed{\widehat{\mathcal{Z}}^\dagger(\log[\widehat{\mathcal{Z}}(\rho)]) + \widehat{\mathcal{Z}}^\dagger(I)}.$$

The Hessian in the direction $\Delta\rho$ is then

$$\nabla^2 f(\rho)(\Delta\rho) \quad = \quad \boxed{\widehat{\mathcal{G}}^\dagger(\log'[\widehat{\mathcal{G}}(\rho)](\widehat{\mathcal{G}}(\Delta\rho))} - \boxed{\widehat{\mathcal{Z}}^\dagger(\log'[\widehat{\mathcal{Z}}(\rho)](\widehat{\mathcal{Z}}(\Delta\rho))}.$$

Given a real-valued convex function f, a *subgradient* ϕ of f at x is a vector satisfying $f(y) \geq f(x) + \langle \phi, y - x \rangle$, $\forall y$. The set of gradients of f at x is called the *subdifferential*, denoted by $\partial f(x)$. For a differentiable function f, the subdifferential is a singleton, $\partial f(x) = \{\nabla f(x)\}$; see e.g., [19].

Theorem 3.14. *Let f be as defined in (3.19) and let $\{\rho_i\}_i \subseteq \mathbb{H}^{n_\rho}_{++}$ with $\rho_i \to \bar{\rho}$. If we have the convergence $\lim_i \nabla f(\rho_i) = \phi$, then*

$$\phi \in \partial f(\bar{\rho}).$$

Proof. The result follows from the characterization of the subgradient as containing the convex hull of all limits of gradients, e.g., [19, Theorem 25.6]. □

[5] This follows from [19, Theorem 6.6], i.e., from $\operatorname{relint}(AC) = A \operatorname{relint}(C)$, where C is a convex set and $A : \mathbb{E}^n \to \mathbb{E}^m$ is a linear map.

4 Optimality Conditions, Bounding, GN Interior Point Method

Arguably, the most popular approach for solving **SDP** problems is by applying a path-following interior point approach to solving perturbed optimality conditions using Newton's method. However, in general, numerical difficulties and instability arise in two ways. First, the optimality conditions for **SDP** problems are overdetermined, and a symmetrization is needed to apply a standard Newton method. Second, block Gaussian elimination is applied to the linearized Newton system to efficiently solve for the Newton direction. This is done without regard to partial pivoting to avoid roundoff error buildup. To avoid these instabilities, we apply a projected Gauss-Newton, **GN**, interior point approach, Section 4.4, to solve the perturbed optimality conditions for our model (3.19).

In this section, we begin by presenting the optimality conditions for the model (3.19); then the **GN** search direction is introduced in Section 4.2; the projected versions are discussed in Section 4.3; we present the algorithm itself in Section 4.4. We finish this section with bounding strategies in Section 4.5. The important provable lower bound is presented in Section 4.5.3.

4.1 Optimality Conditions and Duality

We first obtain perturbed optimality conditions for (3.19) with positive barrier parameters. This is most often done by using a barrier function and adding terms such as $\mu_\rho \log\det(\rho)$ to the Lagrangian. After differentiation we obtain $\mu_\rho \rho^{-1}$ that we equate with the dual variable Z_ρ. After multiplying through by ρ we obtain the *perturbed complementarity equations*, e.g., $Z_\rho \rho - \mu_\rho I = 0$.

Theorem 4.1. *Let L be the Lagrangian for (3.19), i.e.,*

$$L(\rho, y) = f(\rho) + \langle y, \Gamma_V(\rho) - \gamma_V \rangle,\ y \in \mathbb{R}^{m_V}.$$

The following holds for problem (3.19).

1. $$p^* \quad = \quad \max_y \min_{\rho \succeq 0} L(\rho, y).$$

2. *The Lagrangian dual of (3.19) is*

$$d^* \quad = \quad \max_{Z \succeq 0, y} \left(\min_\rho (L(\rho, y) - \langle Z, \rho \rangle) \right),$$

   and strong duality holds for (3.19), i.e., $d^* = p^*$ and d^* is attained for some $(y, Z) \in \mathbb{R}^{m_V} \times \mathbb{H}^{n_\rho}_+$.

3. *The primal-dual pair $(\rho, (y, Z))$, with $\partial f(\rho) \neq \emptyset$, is optimal if, and only if,*

$$\begin{array}{rcll} 0 & \in & \partial f(\rho) + \Gamma_V^\dagger(y) - Z & \textit{(dual feasibility)} \\ 0 & = & \Gamma_V(\rho) - \gamma_V & \textit{(linear primal feasibility)} \\ 0 & = & \langle \rho, Z \rangle & \textit{(complementary slackness)} \\ 0 & \preceq & \rho, Z & \textit{(semidefiniteness primal feasibility).} \end{array} \tag{4.1}$$

Proof. The proof is given in Appendix A.5. □

4.1.1 Perturbed Optimality Conditions

Many interior-point based algorithms try to solve the optimality conditions (3.19) by solving a sequence of perturbed problems while driving the perturbation parameter $\mu \downarrow 0$. The parameter μ gives a measure of the duality gap. In this section, we present the perturbed optimality conditions for **QKD**. Note that the optimality conditions in (4.1) assumed the existence of the subdifferential. This assumption is not required for the perturbed optimality conditions as we can use existing gradients.

Theorem 4.2. *The barrier function for (3.19) with barrier parameter $\mu > 0$ is*

$$B_\mu(\rho, y) = f(\rho) + \langle y, \Gamma_V(\rho) - \gamma_V \rangle - \mu \log\det(\rho).$$

With $Z = \mu\rho^{-1}$ scaled to $Z\rho - \mu I = 0$, we obtain the perturbed optimality conditions for (3.19) at $\rho, Z \succ 0$, y:

$$\begin{array}{lllll} \textit{dual feasibility} & (\nabla B_\rho = 0) & : & F_\mu^d & = & \nabla_\rho f(\rho) + \Gamma_V^\dagger(y) - Z = 0 \\ \textit{primal feasibility} & (\nabla B_y = 0) & : & F_\mu^p & = & \Gamma_V(\rho) - \gamma_V = 0 \\ \textit{perturbed complementary slackness} & & : & F_\mu^c & = & Z\rho - \mu I = 0. \end{array} \tag{4.2}$$

In fact, for each $\mu > 0$ there is a unique primal-dual solution ρ_μ, y_μ, Z_μ satisfying (4.2). This defines the central path as $\mu \downarrow 0$. Moreover,

$$(\rho_\mu, y_\mu, Z_\mu) \underset{\mu\downarrow 0}{\to} (\rho, y, Z) \text{ satisfying (4.1).}$$

Proof. The optimality condition (4.2) follows from the necessary and sufficient optimality conditions of the convex problem

$$\min_\rho \{ f(\rho) - \mu \log\det(\rho) \ : \ \Gamma_V(\rho) = \gamma_V \}$$

and setting $Z = \mu\rho^{-1}$. Note that B_μ is the Lagrangian function of this convex problem. For each $\mu > 0$ there exists a unique solution to (4.2) due to the strict convexity of the barrier term $-\mu \log\det(\rho)$ and boundedness of the level set of the objective. The standard log barrier argument [20, 21] and Theorem 3.14 together give the last claim. □

Theorem 4.2 above provides an interior point path following method, i.e., for each $\mu \downarrow 0$ we solve the pertubed optimality conditions

$$F_\mu(\rho, y, Z) = \begin{bmatrix} \nabla_\rho f(\rho) + \Gamma_V^\dagger(y) - Z \\ \Gamma_V(\rho) - \gamma_V \\ Z\rho - \mu I \end{bmatrix} = 0, \quad \rho, Z \succ 0. \tag{4.3}$$

The question is how to do this efficiently. The nonlinear system is overdetermined as

$$F_\mu : \mathbb{H}^{n_\rho} \times \mathbb{R}^{m_V} \times \mathbb{H}^{n_\rho} \to \mathbb{H}^{n_\rho} \times \mathbb{R}^{m_V} \times \mathbb{C}^{n_\rho \times n_\rho}.$$

Therefore we cannot apply Newton's method directly to the nonlinear system (4.3), since the linearization does not yield a square system.

4.2 Gauss-Newton Search Direction

To avoid the instability that is introduced by symmetrizations for applying Newton's method to the overdetermined optimality conditions, we use a Gauss-Newton approach. That is, to solve the optimality conditions (4.3), we consider the equivalent nonlinear least squares problem

$$\min_{\rho, Z\succ 0, y} g(\rho, y, Z) := \frac{1}{2}\|F_\mu(\rho, y, Z)\|^2 = \frac{1}{2}\|F_\mu^d(\rho, y, Z)\|_F^2 + \frac{1}{2}\|F_\mu^p(\rho)\|^2 + \frac{1}{2}\|F_\mu^c(\rho, Z)\|_F^2.$$

The *Gauss-Newton direction*, d_{GN}, is the least squares solution of the linearization

$$F_\mu'(\rho, y, Z) d_{GN} = -F_\mu(\rho, y, Z),$$

where F_μ' denotes the Jacobian of F_μ.

Remark 4.3. *The Gauss-Newton method is a popular method for solving nonlinear least squares problems. It is arguably the method of choice for overdetermined problems such as the one we have here. It is called Newton's method for overdetermined systems, e.g., [22], see also the classical book [23]. In particular, it is very successful in cases where the residual at optimality is small. And, in our case the residual is zero at optimality. Other symmetrization schemes are discussed in e.g., [24].*

Lemma 4.4. *Under a full rank assumption of $F_\mu'(\rho, y, Z)$, we get*

$$d_{GN} = -((F_\mu')(\rho, y, Z)^\dagger F_\mu'(\rho, y, Z))^{-1}(F_\mu'(\rho, y, Z))^\dagger F_\mu(\rho, y, Z).$$

Moreover, if $\nabla g(\rho, y, Z) \neq 0$, then d_{GN} is a descent direction for g.

Proof. The gradient of g is, omitting the variables,

$$\nabla g = (F_\mu')^\dagger (F_\mu);$$

and the Gauss-Newton direction is the least squares solution of the linearization $F_\mu' d_{GN} = -F_\mu$, i.e., under a full rank assumption, we get the solution from the normal equations as

$$d_{GN} = -((F_\mu')^\dagger F_\mu')^{-1}(F_\mu')^\dagger F_\mu.$$

We see that the inner product with the gradient is indeed negative, hence a descent direction. □

We now give an explicit representation of the linearized system for (4.3). We define the (right/left matrix multiplication) linear maps

$$\mathcal{M}_Z,\ \mathcal{M}_\rho : \mathbb{H}^{n_\rho} \to \mathbb{C}^{n_\rho \times n_\rho}, \quad \mathcal{M}_Z(\Delta X) = Z\Delta X,\ \mathcal{M}_\rho(\Delta X) = \Delta X \rho.$$

Then the linearization of (4.3) is

$$F_\mu' d_{GN} = \begin{bmatrix} \nabla^2 f(\rho)\Delta\rho + \Gamma_V^\dagger(\Delta y) - \Delta Z \\ \Gamma_V(\Delta\rho) \\ Z\Delta\rho + \Delta Z \rho \end{bmatrix} = \begin{bmatrix} \nabla^2 f(\rho) & \Gamma_V^\dagger & -I \\ \Gamma_V & & \\ \mathcal{M}_Z & & \mathcal{M}_\rho \end{bmatrix} \begin{pmatrix} \Delta\rho \\ \Delta y \\ \Delta Z \end{pmatrix} \approx -F_\mu. \tag{4.4}$$

We emphasize that the last term is in $\mathbb{C}^{n_\rho \times n_\rho}$ and the system is overdetermined. The adjoints of $\mathcal{M}_Z, \mathcal{M}_\rho$ are discussed in Section 2.4, Lemmas A.1 and A.2. Solving the system (4.4), we obtain the **GN** *-direction,* $d_{GN} \in \mathbb{H}^{n_\rho} \times \mathbb{R}^{m_V} \times \mathbb{H}^{n_\rho}$.

4.3 Projected Gauss-Newton Directions

The **GN** direction in (4.4) solves a relatively large overdetermined linear least squares system and does not explicitly exploit the zero blocks. We now proceed to take advantage of the special structure of the linear system by using projection and block elimination.

4.3.1 First Projected Gauss-Newton Direction

Given the system (4.4), we can make a substitution for ΔZ using the first block equation

$$\Delta Z = F_\mu^d + \nabla^2 f(\rho)\Delta\rho + \Gamma_V^\dagger(\Delta y). \tag{4.5}$$

This leaves the two blocks of equations

$$\begin{aligned}
\left(F_\mu^{pc}\right)' \begin{pmatrix} \Delta\rho \\ \Delta y \end{pmatrix} &= \begin{bmatrix} \Gamma_V(\Delta\rho) \\ Z\Delta\rho + \left(\nabla^2 f(\rho)\Delta\rho + \Gamma_V^\dagger(\Delta y)\right)\rho \end{bmatrix} \\
&= \begin{bmatrix} \Gamma_V \\ \mathcal{M}_Z + \mathcal{M}_\rho \nabla^2 f(\rho) & \mathcal{M}_\rho \Gamma_V^\dagger \end{bmatrix} \begin{pmatrix} \Delta\rho \\ \Delta y \end{pmatrix} \\
&\approx -\begin{bmatrix} F_\mu^p \\ F_\mu^c + F_\mu^d \rho \end{bmatrix},
\end{aligned}$$

where the superscript in F_μ^{pc} stands for the primal and complementary slackness constraints.

The adjoint equation follows:

$$\left[\left(F_\mu^{pc}\right)'\right]^\dagger \begin{pmatrix} r_p \\ R_c \end{pmatrix} = \begin{bmatrix} \Gamma_V^\dagger & \mathcal{M}_Z{}^\dagger + \nabla^2 f(\rho)\mathcal{M}_\rho{}^\dagger \\ 0 & \Gamma_V \mathcal{M}_\rho{}^\dagger \end{bmatrix} \begin{pmatrix} r_p \\ R_c \end{pmatrix}.$$

In addition, we can evaluate the condition number of the system using $\left(\left(F_\mu^{pc}\right)'\right)^\dagger \left(F_\mu^{pc}\right)'$. Note that we include the adjoints as they are needed for matrix free methods that exploit sparsity.

4.3.2 Second Projected Gauss-Newton Direction

We can further reduce the size of the linear system by making further variable substitutions. Recall that in Section 4.3.1 we solve the system with a variable in $\mathbb{H}^{n_\rho} \times \mathbb{R}^{m_V}$, i.e., $n_\rho^2 + m_V$ number of unknowns. In this section, we make an additional substitution using the second block equation in (4.4) and reduce the number of the unknowns to n_ρ^2.

Theorem 4.5. *Let $\hat{\rho} \in \mathbb{H}^{n_\rho}$ be a feasible point for $\Gamma_V(\cdot) = \gamma_V$. Let $\mathcal{N}^\dagger : \mathbb{R}^{n_\rho^2 - m_V} \to \mathbb{H}^{n_\rho}$ be an injective linear map in adjoint form so that, again by abuse of notation and redefining the primal residual, we have the nullspace representation:*

$$F_\mu^p = \Gamma_V(\rho) - \gamma_V \iff F_\mu^p = \mathcal{N}^\dagger(v) + \hat{\rho} - \rho, \text{ for some } v.$$

Then the second projected **GN** *direction, $d_{GN} = \begin{pmatrix} \Delta v \\ \Delta y \end{pmatrix}$, is found from the least squares solution of*

$$\boxed{\left[Z\mathcal{N}^\dagger(\Delta v) + \nabla^2 f(\rho)\mathcal{N}^\dagger(\Delta v)\rho\right] + \left[\Gamma_V^\dagger(\Delta y)\rho\right] = -F_\mu^c - ZF_\mu^p - \left(F_\mu^d + \nabla^2 f(\rho)F_\mu^p\right)\rho.} \tag{4.6}$$

Proof. Using the new primal feasibility representation, the perturbed optimality conditions in (4.3) become:

$$F_\mu(\rho, v, y, Z) = \begin{bmatrix} F_\mu^d \\ F_\mu^p \\ F_\mu^c \end{bmatrix} = \begin{bmatrix} \nabla_\rho f(\rho) + \Gamma_V^\dagger(y) - Z \\ \mathcal{N}^\dagger(v) + \hat{\rho} - \rho \\ Z\rho - \mu I \end{bmatrix} = 0, \quad \rho, Z \succ 0. \tag{4.7}$$

After linearizing the system (4.7) we use the following to find the **GN** search direction:

$$F_\mu' d_{GN} = \begin{bmatrix} \nabla^2 f(\rho)\Delta\rho + \Gamma_V^\dagger(\Delta y) - \Delta Z \\ \mathcal{N}^\dagger(\Delta v) - \Delta\rho \\ Z\Delta\rho + \Delta Z\rho \end{bmatrix} \approx -F_\mu.$$

From the first block equation we have

$$\begin{aligned} \Delta Z &= F_\mu^d + \nabla^2 f(\rho)\Delta\rho + \Gamma_V^\dagger(\Delta y) \\ &= F_\mu^d + \nabla^2 f(\rho)(F_\mu^p + \mathcal{N}^\dagger(\Delta v)) + \Gamma_V^\dagger(\Delta y). \end{aligned}$$

From the second block equation, we have

$$\Delta\rho = F_\mu^p + \mathcal{N}^\dagger(\Delta v).$$

Substituting ΔZ and $\Delta\rho$ into $Z\Delta\rho + \Delta Z\rho$ gives

$$\begin{aligned} Z\Delta\rho + \Delta Z\rho &= Z(F_\mu^p + \mathcal{N}^\dagger(\Delta v)) + \left[F_\mu^d + \nabla^2 f(\rho)(F_\mu^p + \mathcal{N}^\dagger(\Delta v)) + \Gamma_V^\dagger(\Delta y)\right]\rho \\ &= \left[Z\mathcal{N}^\dagger(\Delta v) + \nabla^2 f(\rho)\mathcal{N}^\dagger(\Delta v)\rho\right] + \left[\Gamma_V^\dagger(\Delta y)\rho\right] + ZF_\mu^p + \left(F_\mu^d + \nabla^2 f(\rho)F_\mu^p\right)\rho. \end{aligned}$$

Rearranging the terms, the third block equation becomes

$$\begin{aligned} F_\mu^{c\prime}\begin{pmatrix} \Delta v \\ \Delta y \end{pmatrix} &= \left[Z\mathcal{N}^\dagger(\Delta v) + \nabla^2 f(\rho)\mathcal{N}^\dagger(\Delta v)\rho\right] + \left[\Gamma_V^\dagger(\Delta y)\rho\right] \\ &= -F_\mu^c - ZF_\mu^p - \left(F_\mu^d + \nabla^2 f(\rho)F_\mu^p\right)\rho. \end{aligned}$$

□

The matrix representation of (4.6) is presented in Appendix B.2. It is easy to see that the adjoint satisfying $\langle F_\mu^{c\prime}(d_{GN}), R_c\rangle = \langle d_{GN}, (F_\mu^{c\prime})^\dagger(R_c)\rangle$ now follows:

$$(F_\mu^{c\prime})^\dagger(R_c) = \begin{bmatrix} \mathcal{N}\,\mathrm{Hvec}\,\mathcal{M}_Z{}^\dagger + \mathcal{N}\nabla^2 f(\rho)\,\mathrm{Hvec}\,\mathcal{M}_\rho{}^\dagger \\ \Gamma_V\mathcal{M}_\rho{}^\dagger \end{bmatrix}(R_c).$$

After solving the system (4.6), we make back substitutions to recover the original variables. In other words, once we get $(\Delta v, \Delta y)$ from solving (4.6), we obtain $(\Delta\rho, \Delta y, \Delta Z)$ using the original system:

$$\Delta\rho = F_\mu^p + \mathcal{N}^\dagger(\Delta v), \quad \Delta Z = F_\mu^d + \nabla^2 f(\rho)(F_\mu^p + \mathcal{N}^\dagger(\Delta v)) + \Gamma_V^\dagger(\Delta y).$$

Theorem 4.6 below illustrates cases where we maintain the exact primal feasibility.

Theorem 4.6. *Let α be a step length and consider the update*

$$\rho_+ \leftarrow \rho + \alpha\Delta\rho = \rho + F_\mu^p + \alpha\mathcal{N}^\dagger(\Delta v).$$

1. *If a step length one is taken ($\alpha = 1$), then the new primal residual is exact, i.e.,*

$$F_\mu^p = \mathcal{N}^\dagger(v_+) + \hat{\rho} - \rho_+ = 0.$$

2. *Suppose that the exact primal feasibility is achieved. Then the primal residual is* 0 *throughout the iterations regardless of the step length.*

Proof. If a step length one is taken for updating

$$\rho_+ \leftarrow \rho + \Delta\rho = \rho + F_\mu^p + \mathcal{N}^\dagger(\Delta v),$$

then the new primal residual

$$\begin{aligned}(F_\mu^p)_+ &= \mathcal{N}^\dagger(v_+) + \hat{\rho} - \rho_+ \\ &= \mathcal{N}^\dagger(v + \Delta v) + \hat{\rho} - \rho - F_\mu^p - \mathcal{N}^\dagger(\Delta v) \\ &= \mathcal{N}^\dagger(v) + \hat{\rho} - \rho - \mathcal{N}^\dagger(v) - \hat{\rho} + \rho \\ &= 0.\end{aligned}$$

In other words, as for Newton's method, a step length of one implies that the new residuals are zero for linear equations.

We can now change the line search to maintain $\rho_+ = \mathcal{N}^\dagger(v + \alpha\Delta v) - \hat{\rho} \succ 0$ and preserve exact primal feasibility. Assume that $F_\mu^p = 0$.

$$\rho_+ \leftarrow \rho + \alpha\Delta\rho = \rho + \alpha(F_\mu^p + \mathcal{N}^\dagger(\Delta v)) = \rho + \alpha\mathcal{N}^\dagger(\Delta v)$$

Now, we see that

$$\Gamma_V(\rho_+) = \Gamma_V(\rho + \alpha\mathcal{N}^\dagger(\Delta v)) = \Gamma_V(\rho) = \gamma,$$

where the last equality follows from the exactly feasibility assumption. □

4.4 Projected Gauss-Newton Primal-Dual Interior Point Algorithm

We now present the pseudocode for the Gauss-Newton primal-dual interior point method in Algorithm 1.

It is a series of steps that find the least squares solution of the over-determined linear system (4.6), while decreasing the perturbation parameter $\mu \downarrow 0$, and maintaining the positive definiteness of ρ, Z. Algorithm 1 is summarized as follows:

1. At each iteration, we find the projected **GN** direction described in Section 4.3.2; See Algorithm 1 lines: 2, 3 and 4.

2. We then choose a step length that maintains strict feasibility. Whenever the step length is one, we attain primal feasibility for all future iterations; See Algorithm 1 line: 5.

3. We decrease the perturbation parameter μ appropriately, and proceed to the next iteration; See Algorithm 1 line: 7.

4. The algorithm stops when the relative duality gap reaches a prescribed tolerance or we reach our prescribed maximum number of iterations; See Appendix B.3 for implementation details on stopping criteria, preconditioning, etc.

4.5 Dual and Bounding

We first look at upper bounds[6] found from feasible solutions in Proposition 4.7. Then we use the dual program to provide provable lower bounds for the **FR** problem (2.2) thus providing lower bounds for the original problem with the accuracy of **FR**.

[6]Our discussion about upper bounds here is about upper bounds for the given optimization problem, which are not necessarily key rate upper bounds of the **QKD** protocol under study. This is because the constraints that one feeds into the algorithm might not use all the information available to constrain Eve's attacks.

Algorithm 1 Projected Gauss-Newton Interior Point Algorithm for **QKD**

Require: $\rho \succ 0$, $\mu \in \mathbb{R}_{++}$, $\eta \in (0,1)$

while stopping criteria is not met **do**
  solve (4.6) for $(\Delta v, \Delta y)$
  $\Delta\rho = F_\mu^p + \mathcal{N}^\dagger(\Delta v)$
  $\Delta Z = F_\mu^d + \nabla^2 f(\rho)(F_\mu^p + \mathcal{N}^\dagger(\Delta v)) + \Gamma_V^\dagger(\Delta y)$
  choose step length α
  $(\rho, y, Z) \leftarrow (\rho, y, Z) + \alpha(\Delta\rho, \Delta y, \Delta Z)$
  $\mu \leftarrow \langle \rho, Z\rangle / n_\rho$; $\mu \leftarrow \eta\mu$
end while

4.5.1 Upper Bounds

A trivial upper bound is obtained as soon as we have a primal feasible solution $\hat{\rho}$ by evaluating the objective function. Our algorithm is a primal-dual *infeasible* interior point approach. Therefore we typically have approximate linear feasibility $\Gamma_V(\hat{\rho}) \approx \gamma_V$; though we do maintain positive definiteness $\hat{\rho} \succ 0$ throughout the iterations. Therefore, once we are close to feasibility we can project onto the affine manifold and hopefully maintain positive definiteness, i.e., we apply iterative refinement by finding the projection

$$\min_\rho \left\{ \frac{1}{2}\|\rho - \hat{\rho}\|^2 \ : \ \Gamma_V(\rho) = \gamma_V \right\}.$$

Proposition 4.7. *Let $\hat{\rho} \succ 0$, $F_\mu^p = \Gamma_V(\hat{\rho}) - \gamma_V$. Then*

$$\rho = \hat{\rho} - {\Gamma_V}^{-1} F_\mu^p = \operatorname*{argmin}_\rho \left\{ \frac{1}{2}\|\rho - \hat{\rho}\|^2 \ : \ \Gamma_V(\rho) = \gamma_V \right\},$$

where we denote ${\Gamma_V}^{-1}$, generalized inverse. If $\rho \succeq 0$, then $p^* \leq f(\rho)$.

In our numerical experiments below we see that we obtain valid upper bounds starting in the early iterations and, as we use a Newton type method, we maintain exact primal feasibility throughout the iterations resulting in a zero primal residual, and no further need for the projection. As discussed above, we take a step length of one as soon as possible. This means that exact primal feasibility holds for the remaining iterations and we keep improving the upper bound at each iteration.

4.5.2 Lower Bounds for **FR** Problem

Facial reduction for the affine constraint means that the corresponding feasible set of the original problem lies within the minimal face $V_\rho \mathbb{H}_+^{n_\rho} V_\rho^\dagger$ of the semidefinite cone. Since we maintain positive definiteness for ρ, Z during the iterations, we can obtain a lower bound using weak duality. Note that $\rho \succ 0$ implies that the gradient $\nabla f(\rho)$ exists.

Corollary 4.8 (lower bound for **FR** (3.19))**.** *Consider the problem (3.19). Let $\hat{\rho}, \hat{y}$ be a primal-dual iterate with $\hat{\rho} \succ 0$. Let*

$$\bar{Z} = \nabla f(\hat{\rho}) + \Gamma_V^\dagger(\hat{y}).$$

If $\bar{Z} \succeq 0$, then a lower bound for problem (3.19) is

$$p^* \geq f(\hat{\rho}) + \langle \hat{y}, \Gamma_V(\hat{\rho}) - \gamma_V \rangle - \langle \hat{\rho}, \bar{Z} \rangle.$$

Proof. Consider the dual problem

$$d^* = \max_{y,Z\succeq 0} \min_{\rho\in\mathbb{H}^{n_\rho}} L(\rho, y) - \langle Z, \rho\rangle.$$

We now have dual feasibility

$$\bar{Z} \succeq 0,\ \nabla f(\hat{\rho}) + \Gamma_V^\dagger(\hat{y}) - \bar{Z} = 0 \implies \hat{\rho} \in \operatorname*{argmin}_{\rho} L(\rho, \hat{y}) - \langle \bar{Z}, \rho\rangle.$$

Since we have dual feasibility, weak duality in Theorem 4.1, Item 2 as stated in the dual problem above yields the result. □

Remark 4.9. *We note that the lower bound in Corollary 4.8 is a simplification of the approach in [5], where after a near optimal solution is found, a dual problem of a linearized problem is solved using CVX in MATLAB. Then a strong duality theorem is assumed to hold and is applied along with a linearization of the objective function. Here we do not assume strong duality, though it holds for the facially reduced problem. And we apply weak duality to get a theoretically guaranteed lower bound.*

*We emphasize that this holds within the margin of error of the **FR**. Recall that we started with the problem in (2.3). If we only apply the accurate **FR** based on spectral decompositions, then the lower bound from Corollary 4.8 is accurate and theoretically valid up to the accuracy of the spectral decompositions.*[7] *In fact, in our numerics, we can obtain tiny gaps of order 10^{-13} when requested; and we have never encountered a case where the lower bound is greater than the upper bound. Thus the bound applies to our original problem as well. Greater care must be taken if we had to apply **FR** to the entire constraint $\Gamma(\rho) = \gamma$. The complexity of **SDP** feasibility is still not known. Therefore, the user should be aware of the difficulties if the full **FR** is done.*

A corresponding result for a lower bound for the original problem is given in Corollary 4.10.

4.5.3 Lower Bounds for the Original Problem

We can also obtain a lower bound for the case where **FR** is performed with some error. Recall that we assume that the original problem (2.3) is feasible. We follow the same arguments as in Section 4.5.2 but apply it to the original problem. All that changes is that we have to add a small perturbation to the optimum $V_\rho \hat{R} V_\rho^\dagger$ from the **FR** problem in order to ensure a positive definite ρ for differentiability. The exposing vector from **FR** process presents an intuitive choice for the perturbation.

Corollary 4.10. *Consider the original problem (2.3) and the results from the theorem of the alternative, Lemma 2.4, for fixed y:*

$$0 \neq W = \Gamma^\dagger(y) \succeq 0,\ \gamma^\dagger y = \epsilon_\gamma, \quad \epsilon_\gamma \geq 0. \tag{4.8}$$

Let the orthogonal spectral decomposition be

$$W = \begin{bmatrix} V & N \end{bmatrix} \begin{bmatrix} D_\delta & 0 \\ 0 & D_> \end{bmatrix} \begin{bmatrix} V & N \end{bmatrix}^\dagger,\ D_> \in \mathbb{S}^r_{++}.$$

Let $0 \preceq \eta \approx W$ be the (approximate) exposing vector obtained as the nearest rank r positive semidefinite matrix to W,

$$W = N D_> N^\dagger + V D_\delta V^\dagger = \eta + V D_\delta V^\dagger.$$

[7] Note that the condition number of the spectral decomposition of Hermitian matrices is 1; see e.g., [25].

*Let $\hat{R}, \hat{y}$ be a primal-dual iterate for the **FR** problem, with $\hat{R} \succ 0$. Add a small perturbation matrix $\Phi \succ 0$ to guarantee that the approximate optimal solution*

$$\hat{\rho}_\phi = V\hat{R}V^\dagger + N\Phi N^\dagger \succ 0.$$

Without loss of generality, let $\hat{y}$ be a dual variable for (2.3), adding zeros to extend the given vector if needed. Set

$$\bar{Z}_\phi = \nabla f(\hat{\rho}_\phi) + \Gamma^\dagger(\hat{y}). \tag{4.9}$$

If $\bar{Z}_\phi \succeq 0$, then a lower bound for the original problem (2.3) is

$$p^* \geq f(\hat{\rho}_\phi) + \langle \hat{y}, \Gamma(\hat{\rho}_\phi) - \gamma \rangle - \langle \hat{\rho}_\phi, \bar{Z}_\phi \rangle. \tag{4.10}$$

Proof. By abuse of notation, we let f, L be the objective function and Lagrangian for (2.3). Consider the dual problem

$$d^* = \max_{y, Z \succeq 0} \min_{\rho \in \mathbb{H}^n} \left(L(\rho, y) - \langle Z, \rho \rangle \right).$$

We now have dual feasibility

$$\bar{Z}_\phi \succeq 0,\ \nabla f(\hat{\rho}_\phi) + \Gamma^\dagger(\hat{y}) - \bar{Z}_\phi = 0 \implies \hat{\rho}_\phi \in \operatorname*{argmin}_\rho \left(L(\rho, \hat{y}) - \langle \bar{Z}_\phi, \rho \rangle \right).$$

Since we have dual feasibility, weak duality in Theorem 4.1, Item 2 as stated in the dual problem above yields the result. □

Remark 4.11. *We note that (4.9) with $\bar{Z}_\phi$ is dual feasibility (stationarity of the Lagrangian) for an optimal $\hat{\rho}_\phi$. Therefore, under continuity arguments, we expect $\bar{Z}_\phi \succeq 0$ to hold as well.*

*In addition, for implementation we need to be able to evaluate $\nabla f(\hat{\rho}_\phi)$. Therefore, we need to form the positive definite preserving maps $\widehat{\mathcal{G}}, \widehat{\mathcal{Z}}$, but without performing **FR** on the feasible set. That we can do this accurately using a spectral decomposition follows from Lemma 3.8.*

5 Numerical Testing

We compare our algorithm to other algorithms by considering six **QKD** protocols including four variants of the Bennett-Brassard 1984 (BB84) protocol, twin-field **QKD** and discrete-modulated continuous-variable **QKD** . In Appendix C we include the descriptions of protocol examples that we use to generate instances for the numerical tests. We note that while it is possible to simplify the optimization problem of some protocols using protocol-specific properties as discussed in Section 1.1, we have not performed those protocol-specific simplifications since we aim to demonstrate the generality of our method for a wide class of protocols and in particular the ability to handle problems with considerably large problem sizes.

We continue with the tests in Sections 5.1 to 5.3. This includes security analysis of some selected **QKD** protocols and comparative performances among different algorithms. In particular, in Section 5.1, we compare the results obtained by our algorithm with the analytical results for selected test examples where tight analytical results can be obtained. In Section 5.2, we present results where it is quite challenging for the previous method in [5] to produce tight lower bounds. In particular, we consider the discrete-modulated continuous-variable **QKD** protocol and compare results obtained in [26]. In Section 5.3, we compare performances among different algorithms in terms of accuracy and running time.

5.1 Comparison between the Algorithmic Lower Bound and the Theoretical Key Rate

We compare results from instances for which there exist tight analytical key rate expressions to demonstrate that our Gauss-Newton method can achieve high accuracy with respect to the analytical key rates. There are known analytical expressions for entanglement-based BB84, prepare-and-measure BB84 as well as measurement-device-independent BB84 variants mentioned in Appendix C. We take the measurement-device-independent BB84 as an example since it involves the largest problem size among these three examples and therefore more numerically challenging. In Figure 5.1, we present instances with different choices of parameters for data generation. The instances are tested with a desktop computer that runs with the operating system Ubuntu 18.04.4 LTS, MATLAB version 2019a, Intel Xeon CPU E5-2630 v3 @ 2.40GHz × 32 and 125.8 Gigabyte memory. We set the tolerance $\epsilon = 10^{-12}$ for the Gauss-Newton method.

In Figure 5.1, the numerical lower bounds from the Gauss-Newton method are close to the analytical results to at least 12 decimals and in many cases they agree up to 15 decimals.

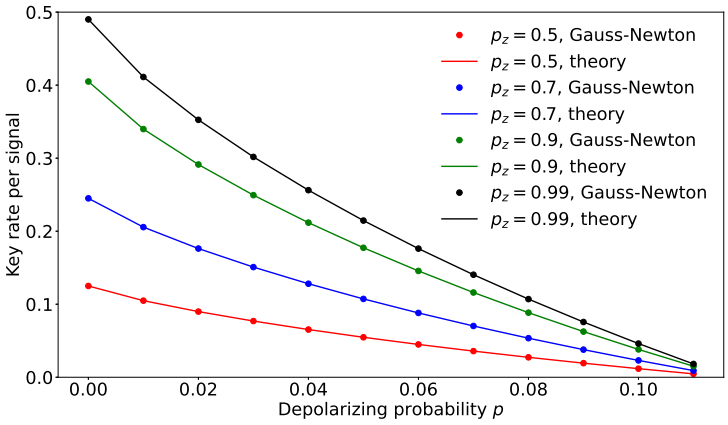

Figure 5.1: Comparisons of key rate for measurement-device-independent BB84 (Appendix C.3) between our Gauss-Newton method and the known analytical key rate.

As noted in Appendix C.5, analytical results are also known when the channel noise parameter ξ is set to zero since in this case, one may argue the optimal eavesdropping attack is the generalized beam splitting attack. This means the feasible set contains essentially a single ρ up to unitaries. Since our objective function is unitarily invariant, one can analytically evaluate the key rate expression. In Figure 5.2, we compare the results from the Gauss-Newton method with the analytical key rate expressions for different choices of distances L (See Appendix C.5 for the description about instances of this protocol example). These instances were run in the same machine as in Figure 5.1. We set the tolerance $\epsilon = 10^{-9}$ for the Gauss-Newton method.

5.2 Solving Numerically Challenging Instances

We show results where the Frank-Wolfe method without **FR** has difficulties in providing tight lower bounds in certain instances. In Figure 5.2, we plot results obtained previously in [26, Figure 2(b)] by the Frank-Wolfe method without **FR**. In particular, results from Frank-Wolfe method have visible differences from the analytical results starting from distance $L = 60$ km. In addition the lower bounds are quite loose once the distance reaches 150 km. In fact, there are points like the one around 180 km where the Frank-Wolfe method cannot produce nontrivial (nonzero) lower bounds. On the other hand, the Gauss-Newton method provides much tighter lower bounds.

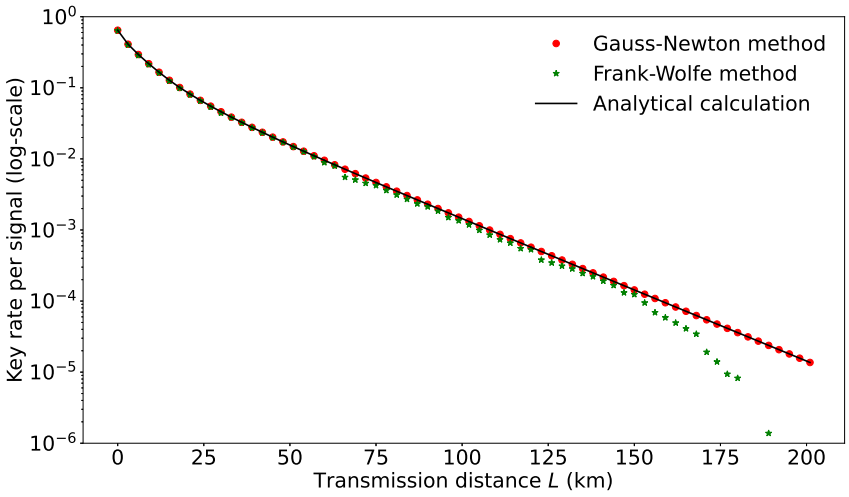

Figure 5.2: Comparison of key rate for discrete-modulated continuous-variable **QKD** (Appendix C.5) among our Gauss-Newton method, the Frank-Wolfe method and analytical key rate for the noise $\xi = 0$ case.

In Figure 5.3, we show another example to demonstrate the advantages of our method. These instances were run in the same machine as in Figure 5.2. For this discrete-phase-randomized BB84 protocol with 5 discrete global phases (see Appendix C.6 for more descriptions), the previous Frank-Wolfe method was unable to find nontrivial lower bounds. This is because the previous method can only achieve an accuracy around 10^{-3} for this problem due to the problem size. This is insufficient to produce nontrivial lower bounds for many instances since the key rates are on the order of 10^{-3} or lower as shown in Figure 5.3. On the other hand, due to high accuracy of our method, we can obtain meaningful key rates. The advantage of high accuracy achieved by our method enables us to perform security analysis for protocols that involve previously numerically challenging problems. Like the discrete-phase-randomized BB84 protocol, these protocols involve more signal states, which lead to higher-dimensional problems.

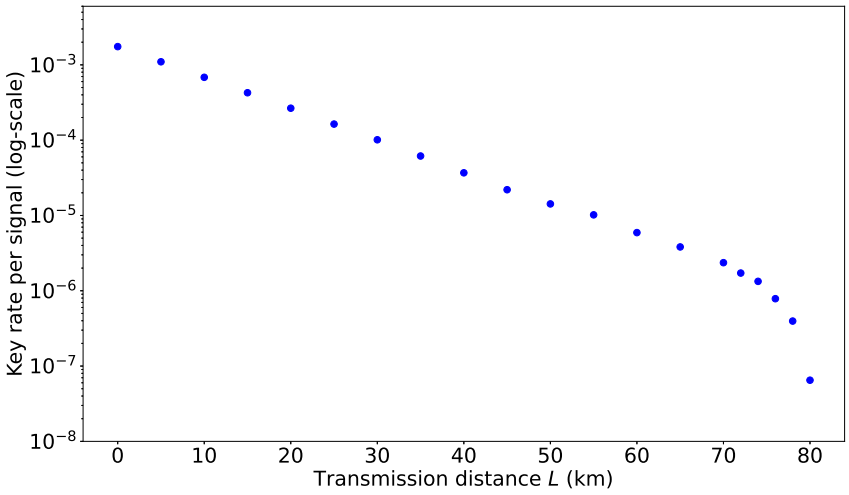

Figure 5.3: Key rate for discrete-phase-randomized BB84 (Appendix C.6) with the number of discrete global phases $c = 5$. In this plot, the coherent state amplitude is optimized for each distance by a simple coarse-grained search over the parameter regime.

5.3 Comparative Performance

In this section we examine the comparative performance among three algorithms; the Gauss-Newton method, the Frank-Wolfe method and cvxquad. The Gauss-Newton method refers to the algorithm developed throughout this paper. The Frank-Wolfe method refers to the algorithm developed in [5] and cvxquad is developed in [27]. We use Table 5.1 to present detailed reports on some selected instances. More numerics are reported throughout Tables C.1 to C.6 in Appendix C.7.

The instances are tested with MATLAB version 2021a using Dell PowerEdge R640 Two Intel Xeon Gold 6244 8-core 3.6 GHz (Cascade Lake) with 192 Gigabyte memory. For the instances corresponds to the DMCV protocol, we used the tolerance $\epsilon = 10^{-9}$ and the tolerance $\epsilon = 10^{-12}$ was used for the remaining instances. The maximum number of iteration was set to 80 for the Gauss-Newton method.

Problem Data			**Gauss-Newton**		**Frank-Wolfe with FR**		**Frank-Wolfe w/o FR**		**cvxquad with FR**	
protocol	**parameter**	**size**	**gap**	**time**	**gap**	**time**	**gap**	**time**	**gap**	**time**
ebBB84	(0.50,0.05)	(4,16)	5.98e-13	0.40	1.01e-04	92.49	1.17e-04	93.05	5.46e-01	214.02
ebBB84	(0.90,0.07)	(4,16)	1.42e-12	0.20	2.71e-04	91.26	2.75e-04	94.49	7.39e-01	177.64
pmBB84	(0.50,0.05)	(8,32)	5.51e-13	0.23	1.12e-04	1.38	6.47e-04	1.91	5.26e-01	158.64
pmBB84	(0.90,0.07)	(8,32)	5.13e-13	0.17	7.31e-05	1.29	6.25e-04	38.65	6.84e-01	233.43
mdiBB84	(0.50,0.05)	(48,96)	1.14e-12	1.09	4.99e-05	104.31	5.22e-04	134.05	1.82e-01	557.08
mdiBB84	(0.90,0.07)	(48,96)	2.96e-13	0.96	2.04e-04	106.61	2.85e-03	126.62	4.57e-01	537.52
TFQKD	(0.80,100.00,0.70)	(12,24)	1.15e-12	0.79	2.60e-09	1.21	1.57e-03	124.48	n/a	0.01
TFQKD	(0.90,200.00,0.70)	(12,24)	1.04e-12	0.44	3.98e-09	1.13	1.68e-04	2.25	n/a	0.00
DMCV	(10.00,60.00,0.05,0.35)	(44,176)	2.71e-09	507.83	4.35e-06	467.41	3.57e-06	657.08	n/a	0.01
DMCV	(11.00,120.00,0.05,0.35)	(48,192)	3.24e-09	700.46	2.35e-06	194.62	2.15e-06	283.06	n/a	0.01
dprBB84	(1.00,0.08,30.00)	(12,48)	4.92e-13	1.19	3.85e-06	96.74	9.43e-05	141.38	⋆⋆	118.81
dprBB84	(2.00,0.14,30.00)	(24,96)	1.04e-12	11.76	5.71e-06	17.66	5.38e-06	34.60	⋆⋆	106.24
dprBB84	(3.00,0.10,30.00)	(36,144)	4.96e-13	63.26	6.48e-04	7.38	2.08e-02	29.00	⋆⋆	582.64
dprBB84	(4.00,0.12,30.00)	(48,192)	3.80e-13	330.39	4.42e-05	13.78	9.79e-04	175.39	⋆⋆	3303.23

Table 5.1: Numerical Report from Three Algorithms

In Table 5.1 **Problem Data** refers to the data used to generate the instances. **Gauss-Newton** refers to the Gauss-Newton method. **Frank-Wolfe** refers to the Frank-Wolfe algorithm used in [5] and we use 'with **FR** (w/o **FR**, resp)' to indicate the model is solved with **FR** (without **FR**, resp). The header **cvxquad with FR** refers to the algorithm provided by [27] with **FR** reformulation. If a certain algorithm fails to give a reasonable answer within a reasonable amount of time, we give a '⋆⋆' flag in the gap followed by the time taken to obtain the error message. We use 'n/a' to indicate the instances for which cvxquad is not applicable due to the size differences in the images under $\widehat{\mathcal{G}}$ and $\widehat{\mathcal{Z}}$ due to **FR**.

The following provides details for the remaining headers in Table 5.1.

1. **protocol**: the protocol name; refer to Appendix C;
2. **parameter**: the parameters used for testing; see Appendix C.1 - Appendix C.6 for the ordering of the parameters;
3. **size**: the size (n, k) of original problem; n, k are defined in (3.3);
4. **gap**: the relative gap between the bestub and bestlb;

$$\frac{\text{bestub - bestlb}}{1 + \frac{|\text{bestub}|+|\text{bestlb}|}{2}}. \tag{5.1}$$

5. **time**: time taken in seconds.

We make some discussions on the formula (5.1). The best upper bound from Gauss-Newton algorithm is used for all instances for 'bestub' in (5.1). The Gauss-Newton algorithm computes

the lower bounds as presented in Corollary 4.8. The Frank-Wolfe algorithm presented in [5] obtains the lower bound by a linearization technique near the optimal. As presented in [27], cvxquad uses the semidefinite approximations of the matrix logarithm. The lower bounds from cvxquad are often larger than the theoretical optimal values. This observation indicates that the lower bounds from cvxquad are not reliable. Therefore, we adopt the lower bound strategy used in [5] for cvxquad.

We now discuss the results in Table 5.1. Comparing the two columns **gap** and **time** among the different methods, we see that the Gauss-Newton method outperforms other algorithms in both the accuracy and the running time. For example, comparing **Gauss-Newton** and **Frank-Wolfe with FR**, the gaps and running times from **Gauss-Newton** are competitive. There are three instances that **Gauss-Newton** took longer time. We emphasize that the gap values with **Gauss-Newton** illustrate much higher accuracy.

We now illustrate that the reformulation strategy via **FR** contributes to superior algorithmic performances. For the columns **Frank-Wolfe with FR** and **Frank-Wolfe w/o FR** in Table 5.1, the **FR** reformulation contributes to not only giving tighter gaps but also reducing the running time significantly. We now consider the column corresponding to **cvxquad with FR** in Table 5.1. We see that the algorithm fails (marked with '$\star\star$') with some instances due to the memory shortage. Facial reduction indeed contributes to the reduction on the problem sizes. For example, we reduced the problem sizes in Table 5.2.

protocol	parameter	(n, m)	(n_ρ, m_V)
pmBB84	$(0.5, 0.05)$	$(8, 21)$	$(4, 8)$
mdiBB84	$(0.5, 0.05)$	$(48, 305)$	$(12, 34)$

Table 5.2: Reduction in Problem Sizes

6 Conclusion

6.1 Summary

In this paper we have presented a robust numerical method for finding *provable tight lower bounds* for the convex optimization problem that finds the key rate for the **QKD** problem. Our empirical evidence illustrates consistent significant improvements in solution time and accuracy over previous methods. In particular, we solve many problems close to machine accuracy and provide theoretical provable accurate lower bounds. This includes previously unsolved problems. (See e.g., Table 5.1.)

This paper used novel convex optimization techniques applied specifically to the **QKD** problem. This includes reformulations of the convex optimization problem that finds the key rate. The result is a regularized problem that avoids the need for previously used perturbations and resulting possible instabilities. Below, we give a summary of the contributions for the model reformulation and for the algorithm.

6.1.1 Summary of the Model Reformulation

We have reformulated, simplified, and stabilized the model for **QKD** key rate calculation through the sequence

$$(2.2) \xrightarrow{(1)} (3.1) \xrightarrow{(2)} (3.8) \xrightarrow{(3)} (3.13) \xrightarrow{(4)} (3.18) \xrightarrow{(5)} (3.19),$$

via (1) variable substitutions; (2) property of $\mathcal{Z}$ from Proposition 3.3; (3) facial reduction on ρ, δ, σ; (4) rotation of the constraints; (5) substituting the constraints back to the objective.

6.1.2 Summary of Algorithm 1

Our algorithm Algorithm 1 is based on a standard primal-dual interior-point approach applied to the **FR** stabilized model. However, it differs in several ways.

1. We modify the primal feasibility to use a nullspace representation. Therefore, both primal and dual feasibility have a similar representation.

2. We use a projected Gauss-Newton search direction to account for the overdetermined least squares problem arising from the optimality conditions. This means we project the Gauss-Newton direction after substituting using the primal-dual linear feasibility equations.

3. We exploit the exact feasibility of linear constraints after a step length one for the Gauss-Newton method. Therefore, we attempt to take a primal and/or dual step length one as soon as possible. Exact feasibility results.

4. We use a modified form of the dual to obtain a lower bound that is used along with an upper bound from the objective function to stop the algorithm when the duality gap is provably small. Our lower bound is a provably lower bound for the original problem as it is using a feasible dual point to evaluate the dual value. This is needed, as an approximate primal optimal value is not exactly optimal.

6.2 Future Plans

There are still many improvements that can be made. Exact primal feasibility was quickly obtained and maintained throughout the iterations. However, accurate dual feasibility was difficult to maintain. This is most likely due to the rounding errors in the numerical computation of the Hessian $H_f(\rho)$ when ρ is near the boundary of the positive semidefinite cone. This approximation can be improved by including a quasi-Newton approach, as we have accurate gradient evaluations. We maintain high accuracy results even in the cases where the Jacobian was not full rank at the optimum. This appears to be due to the special data structures and more theoretical analysis at the optimum can be done.

In this work, we considered a model with the linear constraints $\Gamma(\rho) = \gamma$ restricted to be equalities. While this model covers many interesting **QKD** protocols, there are scenarios where inequality constraints are needed, e.g., when using the flag-state squasher [7] to reduce the dimension. Moreover, there can be additional matrix inequality constraints when the dimension reduction method [8] is applied. It is interesting and important to address possible numerical instabilities introduced by those inequality constraints as was done in this paper.

Acknowledgements

We thank Kun Fang and Hamza Fawzi for discussions and Cunlu Zhou for kindly providing us the code presented in [10]. The authors H.H., J.I. and H.W. thank the support of the National Sciences and Engineering Research Council (NSERC) of Canada. Part of this work was done at the Institute for Quantum Computing, University of Waterloo, which is supported by Innovation, Science and Economic Development Canada. J.L. and N.L. are supported by NSERC under the Discovery Grants Program, Grant No. 341495, and also under the Collaborative Research and Development Program, Grant No. CRDP J 522308-17. Financial support for this work has been partially provided by Huawei Technologies Canada Co., Ltd.

Code Availability

The code developed in this work is currently available at this link.[8] It will be integrated into the open-source **QKD** security software project, which can be accessed via the link.[9]

[8] https://www.math.uwaterloo.ca/~hwolkowi/henry/reports/ZGNQKDmainsolverUSEDforPUBLCNJuly31/

[9] https://openqkdsecurity.wordpress.com/

A Background Results and Proofs

A.1 Adjoints for Matrix Multiplication

Adjoints are essential for our interior point algorithm when using matrix-free methods. We define the *symmetrization linear map*, $\mathcal{S}$, as $\mathcal{S}(M) = (M + M^\dagger)/2$. The *skew-symmetrization linear map*, $\mathcal{SK}$, is $\mathcal{SK}(M) = (M - M^\dagger)/2$.

Lemma A.1 (adjoint of $\mathcal{W}(R) := WR$)**.** *Let $W \in \mathbb{C}^{n\times n}$ be a given square complex matrix, and define the (left matrix multiplication) linear map $\mathcal{W} : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ by $\mathcal{W}(R) = WR$. Then the adjoint $\mathcal{W}^\dagger : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ is defined by*

$$\mathcal{W}^\dagger(M) = \Re(W)^\dagger\Re(M) + \Im(W)^\dagger\,\Im(M) + i\left(\Re(W)^\dagger\,\Im(M) - \Im(W)^\dagger\Re(M)\right). \tag{A.1}$$

If $W \in \mathbb{H}^n$ and $\mathcal{W} : \mathbb{H}^n \to \mathbb{C}^{n\times n}$, then the adjoint $\mathcal{W}^\dagger : \mathbb{C}^{n\times n} \to \mathbb{H}^n$ is defined by

$$\mathcal{W}^\dagger(M) = \mathcal{S}\left[\Re(W)\Re(M) - \Im(W)\,\Im(M)\right] + i\,\mathcal{SK}\left[\Im(W)\Re(M) + \Re(W)\,\Im(M)\right]. \tag{A.2}$$

Proof. We have

$$\begin{aligned} WR &= (\Re(W) + i\,\Im(W))(\Re(R) + i\,\Im(R)) \\ &= \Re(W)\Re(R) - \Im(W)\,\Im(R) + i\Re(W)\,\Im(R) + i\,\Im(W)\Re(R). \end{aligned}$$

Hence,

$$\Re(WR) = \Re(W)\Re(R) - \Im(W)\,\Im(R), \quad \Im(WR) = \Re(W)\,\Im(R) + \Im(W)\Re(R).$$

Then the inner product (2.6) yields

$$\begin{aligned} \langle \mathcal{W}(R), M\rangle &= \langle WR, M\rangle \\ &= \langle \Re(WR), \Re(M)\rangle + \langle \Im(WR), \Im(M)\rangle. \end{aligned}$$

We first focus on the first term.

$$\begin{aligned} \langle \Re(WR), \Re(M)\rangle &= \mathrm{Tr}(\Re(WR)^\dagger\Re(M)) \\ &= \mathrm{Tr}\left([\Re(W)\Re(R) - \Im(W)\,\Im(R)]^\dagger\Re(M)\right) \\ &= \mathrm{Tr}\left([\Re(W)\Re(R)]^\dagger\Re(M)\right) - \mathrm{Tr}\left([\Im(W)\,\Im(R)]^\dagger\Re(M)\right) \\ &= \mathrm{Tr}\left(\Re(R)^\dagger\Re(W)^\dagger\Re(M)\right) - \mathrm{Tr}\left(\Im(R)^\dagger\,\Im(W)^\dagger\Re(M)\right) \\ &= \left\langle \Re(R), \Re(W)^\dagger\Re(M)\right\rangle - \left\langle \Im(R), \Im(W)^\dagger\Re(M)\right\rangle \end{aligned}$$

We now focus on the second term.

$$\begin{aligned} \langle \Im(WR), \Im(M)\rangle &= \mathrm{Tr}(\Im(WR)^\dagger\,\Im(M)) \\ &= \mathrm{Tr}\left([\Re(W)\,\Im(R) + \Im(W)\Re(R)]^\dagger\,\Im(M)\right) \\ &= \mathrm{Tr}\left([\Re(W)\,\Im(R)]^\dagger\,\Im(M)\right) + \mathrm{Tr}\left([\Im(W)\Re(R)]^\dagger\,\Im(M)\right) \\ &= \mathrm{Tr}\left(\Im(R)^\dagger\Re(W)^\dagger\,\Im(M)\right) + \mathrm{Tr}\left(\Re(R)^\dagger\,\Im(W)^\dagger\,\Im(M)\right) \\ &= \left\langle \Im(R), \Re(W)^\dagger\,\Im(M)\right\rangle + \left\langle \Re(R), \Im(W)^\dagger\,\Im(M)\right\rangle \end{aligned}$$

Therefore,

$$\begin{aligned} \langle \mathcal{W}(R), M\rangle &= \left\langle \Re(R), \Re(W)^\dagger\Re(M)\right\rangle - \left\langle \Im(R), \Im(W)^\dagger\Re(M)\right\rangle \\ &\quad + \left\langle \Im(R), \Re(W)^\dagger\,\Im(M)\right\rangle + \left\langle \Re(R), \Im(W)^\dagger\,\Im(M)\right\rangle \\ &= \left\langle \Re(R), \Re(W)^\dagger\Re(M) + \Im(W)^\dagger\,\Im(M)\right\rangle \\ &\quad + \left\langle \Im(R), \Re(W)^\dagger\,\Im(M) - \Im(W)^\dagger\Re(M)\right\rangle \\ &= \langle R, \mathcal{W}^\dagger(M)\rangle. \end{aligned} \tag{A.3}$$

This proves the first general adjoint expression (A.1).

Now, suppose that W Hermitian is given, and consider $\mathcal{W} : \mathbb{H}^n \to \mathbb{C}^{n\times n}$, i.e., a mapping from $\mathbb{H}^n$. Then (A.3) becomes

$$\begin{aligned}\langle \mathcal{W}(R), M\rangle &= \Big\langle \Re(R), \Re(W)^\dagger \Re(M) + \Im(W)^\dagger \Im(M)\Big\rangle \\ &\quad + \Big\langle \Im(R), \Re(W)^\dagger \Im(M) - \Im(W)^\dagger \Re(M)\Big\rangle \\ &= \langle \Re(R), \Re(W)\Re(M) - \Im(W)\,\Im(M)\rangle \\ &\quad + \langle \Im(R), \Re(W)\,\Im(M) + \Im(W)\Re(M)\rangle \\ &= \langle \Re(R), \mathcal{S}(\Re(W)\Re(M) - \Im(W)\,\Im(M))\rangle \\ &\quad + \langle \Im(R), \mathcal{S}\mathcal{K}(\Re(W)\,\Im(M) + \Im(W)\Re(M))\rangle\,.\end{aligned} \tag{A.4}$$

This yields the second term in (A.2). □

Lemma A.2 (adjoint of $\rho(S) = S\rho$)**.** *Let $\rho \in \mathbb{C}^{n\times n}$ be a given square complex matrix, and define the (right matrix multiplication) linear map $\rho : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ by $\rho(S) = S\rho$. Then the adjoint $\rho^\dagger : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ is defined by*

$$\rho^\dagger(M) = \Re(M)\Re(\rho)^\dagger + \Im(M)\,\Im(\rho)^\dagger + i\left(-\Re(M)\,\Im(\rho)^\dagger + \Im(M)\Re(\rho)^\dagger\right). \tag{A.5}$$

If $\rho \in \mathbb{H}^n$ and $\rho : \mathbb{H}^n \to \mathbb{C}^{n\times n}$, then the adjoint $\rho^\dagger : \mathbb{C}^{n\times n} \to \mathbb{H}^n$ is defined by

$$\rho^\dagger(M) = \mathcal{S}\left[\Re(M)\Re(\rho) - \Im(M)\,\Im(\rho)\right] + i\,\mathcal{S}\mathcal{K}\left[\Re(M)\,\Im(\rho) + \Im(M)\Re(\rho)\right]. \tag{A.6}$$

Proof. The proof is similar to the proof of Lemma A.1. □

A.2 Proof of Lemma 3.4

Proof. Recall that

$$A, B \succeq 0 \implies \operatorname{range}(A+B) = \operatorname{range}(A) + \operatorname{range}(B). \tag{A.7}$$

Let X be a positive semidefinite matrix with rank r and spectral decomposition

$$X = \sum_{i=1}^{r} \lambda_i u_i u_i^\dagger.$$

We only focus on the first term $\lambda_1 u_1 u_1^\dagger$. Then

$$\mathcal{Z}(\lambda_1 u_1 u_1^\dagger) = \sum_{j=1}^{n} Z_j(\lambda_1 u_1 u_1^\dagger)Z_j = \sum_{j=1}^{n} \lambda_1 (Z_j u_1)(Z_j u_1)^\dagger.$$

We note, from (A.7), that

$$\begin{aligned}\operatorname{range}(\mathcal{Z}(\lambda_1 u_1 u_1^\dagger)) &= \operatorname{range}(\lambda_1 (Z_1 u_1)(Z_1 u_1)^\dagger + \lambda_1 (Z_2 u_1)(Z_2 u_1)^\dagger + \cdots + \lambda_1 (Z_n u_1)(Z_n u_1)^\dagger) \\ &= \operatorname{range}(Z_1 u_1) + \cdots + \operatorname{range}(Z_n u_1).\end{aligned}$$

We also note that

$$u_1 = I u_1 = \left(\sum_{j=1}^{n} Z_j\right) u_1 = \sum_{j=1}^{n} Z_j u_1 \in \operatorname{range}(Z_1 u_1) + \cdots + \operatorname{range}(Z_n u_1).$$

Hence,

$$\operatorname{range}(\lambda_1 u_1 u_1^\dagger) = \operatorname{range}(u_1) \subseteq \operatorname{range}(Z_1 u_1) + \cdots + \operatorname{range}(Z_n u_1) = \operatorname{range}(\mathcal{Z}(\lambda_1 u_1 u_1^\dagger)).$$

We now consider the first two terms in X, $\lambda_1 u_1 u_1^\dagger + \lambda_2 u_2 u_2^\dagger$. Similarly,

$$\operatorname{range}(\lambda_1 u_1 u_1^\dagger) \subseteq \operatorname{range}(\mathcal{Z}(\lambda_1 u_1 u_1^\dagger)) \quad \text{and} \quad \operatorname{range}(\lambda_2 u_2 u_2^\dagger) \subseteq \operatorname{range}(\mathcal{Z}(\lambda_2 u_2 u_2^\dagger)). \tag{A.8}$$

Then

$$\begin{array}{lll} \operatorname{range}(\lambda_1 u_1 u_1^\dagger + \lambda_2 u_2 u_2^\dagger) & = \operatorname{range}(\lambda_1 u_1 u_1^\dagger) + \operatorname{range}(\lambda_2 u_2 u_2^\dagger) & \text{by (A.7)} \\ & \subseteq \operatorname{range}(\mathcal{Z}(\lambda_1 u_1 u_1^\dagger)) + \operatorname{range}(\mathcal{Z}(\lambda_2 u_2 u_2^\dagger)) & \text{by (A.8)} \\ & = \operatorname{range}(\mathcal{Z}(\lambda_1 u_1 u_1^\dagger) + \mathcal{Z}(\lambda_2 u_2 u_2^\dagger)) & \text{by (A.7)} \\ & = \operatorname{range}(\mathcal{Z}(\lambda_1 u_1 u_1^\dagger + \lambda_2 u_2 u_2^\dagger)) & \text{by linearity of } \mathcal{Z}. \end{array}$$

This completes the proof (The induction steps are clear.). □

A.3 Derivatives for Quantum Relative Entropy under Positive Definite Assumptions

We can reformulate the quantum relative entropy function defined in the key rate optimization (2.2) as

$$\begin{array}{rcl} f(\rho) & = & D(\mathcal{G}(\rho) \| \mathcal{Z}(\mathcal{G}(\rho))) \\ & = & \operatorname{Tr}\left(\mathcal{G}(\rho) \log \mathcal{G}(\rho)\right) - \operatorname{Tr}(\mathcal{G}(\rho) \log \mathcal{Z}(\mathcal{G}(\rho))) \\ & = & \operatorname{Tr}\left(\mathcal{G}(\rho) \log \mathcal{G}(\rho)\right) - \operatorname{Tr}(\mathcal{Z}(\mathcal{G}(\rho)) \log \mathcal{Z}(\mathcal{G}(\rho))) \end{array} \tag{A.9}$$

Here, the linear map $\mathcal{Z}$ is added to the second term in (A.9) above, and the equality follows from Proposition 3.3.

In this section, we review the gradient (Fréchet derivative), and the image of the Hessian, for the reformulated relative entropy function f defined in (A.9). We obtain the derivatives of f under the assumption that the matrix-log is acting on positive definite matrices. This assumption is needed for differentiability. Note that the difficulty arising from the singularity is handled by using perturbations in [5, 10]. This emphasizes the need for the regularization below as otherwise f in (A.9) is *never* differentiable. We avoid using perturbations in this paper by applying **FR** in the sections below.

We now use the chain rule and derive the first and the second order derivatives of the composition of a linear and entropy function.

Lemma A.3. *Let $\mathcal{H} : \mathbb{H}^n \to \mathbb{H}^k$ be a linear map that preserves positive semidefiniteness. Assume that $\mathcal{H}(\rho) \in \mathbb{H}^k_{++}$. Define the composite function $g : \mathbb{H}^k_+ \to \mathbb{R}$ by*

$$g(\rho) = \operatorname{Tr}\left(\mathcal{H}(\rho) \log(\mathcal{H}(\rho))\right).$$

Then the gradient of g at ρ is

$$\nabla g(\rho) = \mathcal{H}^\dagger(\log[\mathcal{H}(\rho)]) + \mathcal{H}^\dagger(I),$$

and the Hessian of g at ρ acting on $\Delta\rho$ is

$$\nabla^2 g(\rho)(\Delta\rho) = \mathcal{H}^\dagger\left(\log' \mathcal{H}(\rho)(\mathcal{H}(\Delta\rho))\right),$$

where $\log'$ denotes the Fréchet derivative.

Proof. We first work on the first-order derivative.

$$\begin{aligned}
\langle \nabla g(\rho), \Delta\rho \rangle &= \Big\langle \frac{d}{d\rho} \operatorname{Tr}\left(\mathcal{H}(\rho)\log(\mathcal{H}(\rho))\right), \Delta\rho \Big\rangle \\
&= \operatorname{Tr}\Big(\frac{d}{d\rho}\left(\mathcal{H}(\rho)\log(\mathcal{H}(\rho))\right)(\Delta\rho)\Big) \\
&= \operatorname{Tr}\left(\frac{d}{d\rho}\Big(\mathcal{H}(\rho)\Big)(\Delta\rho)\log(\mathcal{H}(\rho)) + \mathcal{H}(\rho)\frac{d}{d\rho}\Big(\log(\mathcal{H}(\rho))\Big)(\Delta\rho)\right) \\
&= \Big\langle \frac{d}{d\rho}\Big(\mathcal{H}(\rho)\Big)\Delta\rho, \log(\mathcal{H}(\rho))\Big\rangle + \Big\langle \mathcal{H}(\rho), \frac{d}{d\rho}\Big(\log(\mathcal{H}(\rho))\Big)\Delta\rho\Big\rangle \\
&= \Big\langle \Delta\rho, \mathcal{H}^\dagger\Big(\log(\mathcal{H}(\rho))\Big)\Big\rangle + \left\langle \left(\frac{d}{d\rho}\log(\mathcal{H}(\rho))\right)^\dagger \mathcal{H}(\rho), \Delta\rho \right\rangle \\
&= \Big\langle \Delta\rho, \mathcal{H}^\dagger\Big(\log[\mathcal{H}(\rho)]\Big)\Big\rangle + \left\langle \frac{d\mathcal{H}(\rho)}{d\rho}^\dagger (I), \Delta\rho \right\rangle \\
&= \Big\langle \mathcal{H}^\dagger\Big(\log[\mathcal{H}(\rho)]\Big), \Delta\rho\Big\rangle + \Big\langle \mathcal{H}^\dagger(I), \Delta\rho\Big\rangle .
\end{aligned} \tag{A.10}$$

Note that we used the fact that the directional derivative of matrix-log at ρ in the direction ρ is:

$$\log'(\delta)(\delta) = \log'(\delta;\delta) = I.$$

Similarly, the Hessian g at ρ acting on $\Delta\rho$ can be obtained as follows.

$$\nabla^2 g(\rho)(\Delta\rho) = \tfrac{\partial}{\partial\rho}\mathcal{H}^\dagger\Big([\log \mathcal{H}(\rho)]\Big) = \mathcal{H}^\dagger \tfrac{\partial}{\partial\rho}\Big([\log \mathcal{H}(\rho)]\Big) = \mathcal{H}^\dagger\Big([\log' \mathcal{H}(\rho)(\mathcal{H}\Delta\rho)]\Big). \tag{A.11}$$

□

Under the assumption that $\mathcal{G}(\rho) \succ 0$, we can use Lemma 3.4 and show that $\mathcal{Z}(\mathcal{G}(\rho)) \succ 0$. Using Lemma A.3 and (3.5), we obtain the first and the second order derivatives of the objective function f in (A.9).

Corollary A.4. *Suppose that $\rho \in \mathbb{H}_+^n$ and $\mathcal{G}(\rho) \succ 0$. Then the gradient of f at ρ is*

$$\nabla f(\rho) = \mathcal{G}^\dagger\Big(\log[\mathcal{G}(\rho)]\Big) - (\mathcal{Z}\circ\mathcal{G})^\dagger\Big(\log[(\mathcal{Z}\circ\mathcal{G})(\rho)]\Big). \tag{A.12}$$

The Hessian at $\rho \in \mathbb{H}_+^n$ acting on the direction $\Delta\rho \in \mathbb{H}^n$ is

$$\nabla^2 f(\rho)(\Delta\rho) \;=\; \mathcal{G}^\dagger\Big([\log' \mathcal{G}(\rho)(\mathcal{G}\Delta\rho)]\Big) - (\mathcal{Z}\circ\mathcal{G})^\dagger\Big([\log'(\mathcal{Z}\circ\mathcal{G})(\rho)((\mathcal{Z}\circ\mathcal{G})(\Delta\rho))\Big). \tag{A.13}$$

A.4 Proof of Theorem 3.6

Proof. We provide an alternative, self-contained proof. We note that the key is finding an *exposing vector* for S_R, i.e., $Z_\Gamma \succeq 0$ such that $\langle Z_\Gamma, \rho\rangle = 0$, $\forall \rho \in S_R$. See e.g., [13]. The standard theorem of the alternative for strict feasibility, Lemma 2.4, yields the following equalities for Z_Γ:

$$0 \neq Z_\Gamma = \sum_j y_j\Gamma_j = \sum_j y_j\,(\Theta_j \otimes \mathbb{1}_B) = \left(\sum_j y_j\Theta_j\right) \otimes \mathbb{1}_B \succeq 0; \quad y^\dagger\theta = 0.$$

It is equivalent to look at the smaller problem and find y so that

$$0 \neq Z_\Theta = \sum_j y_j\Theta_j \succeq 0; \quad y^\dagger\theta = 0.$$

Since the reduced density operator constraint requires that $\theta_j = \operatorname{Tr} \rho_A \Theta_j$, we get

$$0 = \sum_j y_j \theta j = \operatorname{Tr}\left(\rho_A \sum_j y_j \Theta_j\right) \iff \rho_A \left(\sum_j y_j \Theta_j\right) = \rho_A Z_\Theta = 0,$$

i.e., the exposing vector $Z_\Theta = Q R_\Theta Q^\dagger$, for some R_Θ. Conversely, we can set $Z_\Theta = QQ^\dagger, R_\Theta = I$, by the basis property of the Θ_i, i.e., the basis property means we can always find an appropriate y so that $\sum_j y_j \Theta_j = QQ^\dagger$. We get that $\operatorname{rank} Z_\Theta = n_A - r$. Therefore, $Z_\Gamma = Z_\Theta \otimes \mathbb{1}_B$, with $\operatorname{rank} Z_\Theta = n_B(n_A - r)$, is an exposing vector as desired, i.e., we have

$$S_R \subset \{\rho \succeq 0 \,:\, \langle Z_\Theta \otimes \mathbb{1}_B, \rho\rangle = 0\}\,.$$

Thus we get the conclusion that $\rho = VRV^\dagger$, as desired. □

A.5 Proof of Theorem 4.1

Proof. The dual in Item 2 is obtained from from standard min-max argument; See [28, Chapter 5].

$$\begin{aligned} d^* = \max_y \min_{\rho \in \mathbb{H}_+^{n_\rho}} L(\rho, y) &= \max_y \left\{ L(\rho, y) : Z \in \partial f(\rho) + \Gamma_V^\dagger(y),\ Z \in (\mathbb{H}_+^{n_\rho} - \rho)^\dagger \right\} \\ &= \max_{y, Z \succeq 0} \min_{\rho \in \mathbb{H}_+^{n_\rho}} L(\rho, y) - \langle Z, \rho\rangle. \end{aligned}$$

That strong duality holds comes from our regularization process, i.e., the existence of a Slater point; see [29, Chapter 8].

Item 3 is the standard optimality conditions for convex programming, where the dual feasibility $0 \in \partial f(\rho) + \Gamma_V^\dagger(y) - Z$ holds from Theorem 3.14. □

B Implementation Details

In this section we look at simplifications for evaluations of the objective function and its derivatives.

B.1 Matrix Representations of Derivatives

We now include a matrix representation for the derivatives.

Let A, B, C be given compatible matrices. If X is Hermitian, then the linear system $AXB = C$ can be written as

$$\left((B^\dagger)^T \otimes A\right) \operatorname{vec}(X) = \operatorname{vec}(C).$$

Note that M^T is the transpose of M, i.e., without conjugation.

Let $g : \mathbb{R} \to \mathbb{R}$ be a continuously differentiable function. The first divided difference $h^{[1]}(\lambda, \mu)$ of g at $\lambda, \mu \in \mathbb{R}$ is defined as

$$h^{[1]}(\lambda, \mu) = \begin{cases} \frac{g(\lambda) - g(\mu)}{\lambda - \mu} & \text{if } \lambda \neq \mu \\ g'(\lambda) & \text{if } \lambda = \mu\,. \end{cases} \tag{B.1}$$

If D is a diagonal matrix with diagonal entries $\lambda_1, \ldots, \lambda_n$, then we define $h^{[1]}(D)$ to be the symmetric $n \times n$ matrix given by $h^{[1]}(\operatorname{diag}(D))$.

Lemma B.1. *Let $\mathcal{A} : \mathbb{H}^s \to \mathbb{H}^t$ be a linear map, $\rho, \Delta\rho \in \mathbb{H}^s, \mathcal{A}(\rho) \in \mathbb{H}^t_{++}$, and $f(\rho) = \mathrm{Tr}(\mathcal{A}(\rho)\log\mathcal{A}(\rho))$. Let $\mathcal{A}(\rho) = UDU^\dagger$ be the spectral decomposition of f at ρ, and the Hessian of f at ρ in the direction $\Delta\rho$ are given by*

$$\nabla f(\rho) = \mathcal{A}^\dagger(\log\mathcal{A}(\rho)) + \mathcal{A}^\dagger(I),$$

and

$$\left(H_f(\rho)\right)(\triangle\rho) = \mathcal{A}^\dagger\left(U(h^{[1]}(D)\circ U^\dagger\mathcal{A}(\triangle\rho)U)U^\dagger\right),$$

where $h^{[1]}(D)$ is the first divided difference of the logarithm function $g(x) = \ln x$, see (B.1) and the paragraph below.

In the actual computation, it is more convenient to express the gradient and Hessian in matrix form. Let A be the matrix representation of $\mathcal{A}$. The Hessian in matrix form is

$$H_f(\rho) \;=\; A^\dagger(U^\dagger\otimes U)\,\mathrm{Diag}(h^{[1]}(D))((U^\dagger)^\dagger\otimes U^\dagger)A.$$

B.2 Matrix Representation of the Second Projected Gauss-Newton System

We present the matrix representation of (4.6). Let N_i be a basis element of $\mathrm{null}(\Gamma_V)$. Then $\mathcal{N}^\dagger(w)$ has the representation $\sum_i w_i N_i$. Then the LHS of (4.6) becomes

$$\begin{array}{rl} & \left[Z\mathcal{N}^\dagger(\Delta v) + \nabla^2 f(\rho)[\mathcal{N}^\dagger(\Delta v)]\ \rho\right] + \left[\Gamma_V^\dagger(\Delta y)\rho\right] \\ = & Z\sum_i N_i\Delta v_i + \nabla^2 f(\rho)[\sum_i N_i\Delta v_i]\ \rho + \Gamma_V^\dagger(\Delta y)\rho \\ = & \sum_i ZN_i\Delta v_i + \sum_i \nabla^2 f(\rho)N_i\rho\Delta v_i + \sum_i \Gamma_i\rho\Delta y_i. \end{array} \tag{B.2}$$

Applying Cvec[10] to the terms related to Δv, we have the following matrix representation:

$$\left[\mathrm{Cvec}\left(ZN_1 + \nabla^2 f(\rho)N_1\rho\right) \quad \cdots \quad \mathrm{Cvec}\left(ZN_{n_\rho^2-m_V} + \nabla^2 f(\rho)N_{n_\rho^2-m_V}\rho\right)\right]\begin{bmatrix}\Delta v_1\\ \vdots \\ \Delta v_{n_\rho^2-m_V}\end{bmatrix}.$$

Similarly, applying Cvec on the terms related to Δy, we have the following matrix representation:

$$\left[\mathrm{Cvec}\left(\Gamma_1\rho\right) \quad \cdots \quad \mathrm{Cvec}\left(\Gamma_m\rho\right)\right]\begin{bmatrix}\Delta y_1\\ \vdots \\ \Delta y_m\end{bmatrix}.$$

The RHS of (4.6) becomes $\mathrm{Cvec}\left(F_\mu^c + ZF_\mu^p + \left(F_\mu^d + \nabla^2 f(\rho)F_\mu^p\right)\rho\right)$. Thus, d_{GN} is obtained be solving the system

$$\begin{array}{rl} & \left[\left[\mathrm{Cvec}\left(ZN_i + \nabla^2 f(\rho)N_i\rho\right)\right]_{i=1,\ldots,n_\rho^2-m_V} \quad \left[\mathrm{Cvec}\left(\Gamma_j\rho\right)\right]_{j=1,\ldots,m_V}\right]\begin{bmatrix}\Delta v\\ \Delta y\end{bmatrix} \\ = & \mathrm{Cvec}\left(F_\mu^c + ZF_\mu^p + \left(F_\mu^d + \nabla^2 f(\rho)F_\mu^p\right)\rho\right). \end{array}$$

B.3 Implementation Heuristics

We now discuss the implementation details. This involves preprocessing for a nullspace representation and preconditioning. The details follow.

[10] The operator vec maps a real matrix to a column vector by stacking the columns on top of one another. Cvec is a generalization of vec for complex matrices. It maps a complex matrix M to the column vector $\begin{bmatrix}\mathrm{vec}(\Re(M))\\ \mathrm{vec}(\mathrm{Im}\,(M))\end{bmatrix}$.

B.3.1 Stopping Criteria

We terminate the algorithm when the optimality condition (4.3) is approximately satisfied. Denote the residual in Theorem 4.5 by

$$\text{RHS} = -F_\mu^c - ZF_\mu^p - \left(F_\mu^d + \nabla^2 f(\rho)F_\mu^p\right)\rho,$$

and the denominator term by

$$\text{denom} = 1 + \frac{1}{2}\min\{\ \|\rho\| + \|Z\|\ ,\ |\text{bestub}| + |\text{bestlb}|\ \}.$$

Then

$$\text{relstopgap} = \frac{1}{\text{denom}}\max\{\ \text{bestub} - \text{bestlb}\ ,\ \|\text{RHS}\|\ \}. \tag{B.3}$$

In other words, for a pre-defined tolerance ϵ, we terminate the algorithm when the relstopgap$< \epsilon$. If the algorithm computes lower and upper bounds of the optimal value throughout its execution, we may terminate the algorithm when the gap between lower and upper bounds is within ϵ. Finally, a common way to terminate an algorithm is to impose restrictions on the running time, e.g., setting an upper bound on the number of iterations or the physical running time.

B.3.2 **GN** Direction using Sparse Nullspace Representation

We let $r = \text{Hvec}(\rho)$, and construct a matrix representation H for the Hessian, and a matrix representation M for the linear constraints that includes a permutation of rows and columns rp, cp with inverse column permutation icp, so that

$$r = \text{Hvec}(\rho): \quad r(cp) = P_{cp}r,\ r = P_{icp}r(cp), \quad P_{cp}P_{icp} = P_{icp}P_{cp} = I,\ P_{icp} = P_{cp}^\dagger.$$

We can ignore the row permutations. We have

$$\begin{aligned}
\Gamma_V(\rho) &= (\Gamma_V\,\text{HMat})\,\text{Hvec}(\rho)\\
&= (\Gamma_V\,\text{HMat})P_{icp}P_{cp}\,\text{Hvec}(\rho)\\
&= (P_{cp}(\Gamma_V\,\text{HMat})^\dagger)^\dagger P_{cp}\,\text{Hvec}(\rho)\\
&= Mr(cp)\\
&= MP_{cp}\,\text{Hvec}(\rho).
\end{aligned}$$

We now get the nullspace representation:

$$\hat{r} = \text{Hvec}(\hat{\rho});\ \Gamma_V(\hat{\rho}) = M\hat{r}(cp) = \gamma_V,\ M = \begin{bmatrix} B & E \end{bmatrix},\ N^\dagger = \begin{bmatrix} B^{-1}E \\ -I \end{bmatrix};$$

$$r = \text{Hvec}(\rho): \quad \Gamma_V(\rho) = \gamma_V \iff \mathcal{M}_{\Gamma_V}P_{cp}r = \gamma_V \iff r = \hat{r} + P_{icp}N^\dagger(w),\ \text{for some } w. \tag{B.4}$$

The permutation of rows and columns are done in order to obtain a simple, near triangular, well conditioned B so that $B^{-1}E$ can be done simply and maintain sparsity if possible. The permutation of the rows does not affect the problem and we can ignore it. However the permutation of the columns cannot be ignored. We get the following

$$\mathcal{N}^\dagger(v) = \text{HMat}\left(P_{icp}N^\dagger(v)\right),\ \Gamma_V^\dagger(\Delta y) = P_{icp}M(\Delta y),\ \nabla^2 f(\rho)\mathcal{N}^\dagger(\Delta v) = \text{HMat}\left(HP_{icp}N^\dagger(\Delta v)\right).$$

By abuse of notation, the Gauss-Newton direction $d_{GN} \in \mathbb{R}^{n_\rho^2}$ can now be found from:

$$\begin{aligned}
F_\mu^{c\prime}d_{GN} &= Z\,\text{HMat}\left(P_{icp}N^\dagger(\Delta v)\right) + \left(\text{HMat}\left(\nabla^2 f(\rho)(P_{icp}N^\dagger(\Delta v))\right) + \Gamma_V^\dagger(\Delta y)\right)\rho\\
&= \left[\mathcal{M}_Z\left(\text{HMat}\left(P_{icp}N^\dagger\bullet\right)\right) + \mathcal{M}_\rho\left(\text{HMat}\left(\nabla^2 f(\rho)P_{icp}N^\dagger\bullet\right)\right) \quad \mathcal{M}_\rho\Gamma_V^\dagger\bullet\right]\begin{pmatrix}\Delta v\\ \Delta y\end{pmatrix}\\
&= -(F_\mu^c + F_\mu^d\rho + ZF_\mu^p) - \left(\nabla^2 f(\rho)(F_\mu^p)\right)\rho.
\end{aligned} \tag{B.5}$$

B.3.3 Preconditioning

The overdetermined linear system in (B.5) can be ill-conditioned. We use diagonal preconditioning, i.e., we let $d_i = \|F_\mu^{c\prime}(e_i)\|$, for unit vectors e_i and then column precondition using

$$F_\mu^{c\prime} \leftarrow F_\mu^{c\prime} \operatorname{Diag}(d)^{-1}.$$[11]

This diagonal preconditioning has been shown to be the optimal diagonal preconditioning for the so-called Ω-condition number, [30]. It performs exceptionally well in our tests below.

B.3.4 Step Lengths

The **GN** method is based on a linearization that suggests a step length of one. However, long step methods are known to be more efficient in practice for interior point methods for linear **SDP**s. Typically step lengths are found using backtracking to ensure primal-dual positive definiteness of ρ, Z.

In our case we do not have a linear objective and we typically experience Maratos type situations, i.e., we get fast convergence for primal feasibility but slow and no convergence for dual feasibility. However, we do have the gradient and Hessian of the objective function and therefore can minimize the quadratic model for the objective function in the search direction $\Delta\rho$

$$\min_\alpha f(\rho) + \alpha\langle\nabla f(\rho), \Delta\rho\rangle + \frac{1}{2}\alpha^2\langle\Delta\rho\nabla^2 f(\rho), \Delta\rho\rangle, \quad \alpha^* = -\langle\nabla f(\rho), \Delta\rho\rangle / \langle\Delta\rho, \nabla^2 f(\rho)\Delta\rho\rangle.$$

Therefore, we begin the backtracking with this step.

Moreover, we take a step length of one as soon as possible, and only after this do we allow step lengths larger than one. This means that exact primal feasibility holds for all further steps. This happens relatively early for our numerical tests.

C Descriptions and Further Numerics of the Protocols

We briefly describe six **QKD** protocols where we compare our algorithm to other algorithms. We also describe how the data (γ in (2.2)) is generated. In addition, we remark on the level of numerical difficulty for each example. We consider four variants of the Bennett-Brassard 1984 (BB84) protocol [3] including single-photon based variants: entanglement-based (Appendix C.1), prepare-and-measure (Appendix C.2), measurement-device-independent [31] (Appendix C.3) and a coherent-state based variant with discrete global phase randomization [32] (Appendix C.6). We also consider the single-photon version of the twin-field **QKD** [33] (Appendix C.4). Another interesting protocol in our numerical tests is the quadrature phase-shift keying scheme of discrete-modulated continuous-variable **QKD** with heterodyne detection (Appendix C.5), see [26, Protocol 2]. These protocols correspond to numerical problems with the level of difficulty ranging from easy to difficult. In the descriptions below, we use the Dirac notation for quantum states which are vectors in the underlying Hilbert space. We skip the description about some common classical postprocessing steps in a **QKD** protocol like error correction and privacy amplification since they are unimportant for our discussions here. We note that the description of linear maps $\mathcal{G}$ and $\mathcal{Z}$ directly follow from the protocol description by following the simplification procedure explained in [26, Appendix A]. We omit those detailed descriptions here and note that the explicit expressions for some of protocols can also be found in [6, Appendix D].

[11] The MATLAB command is: $d_{GN} = ((F_\mu^{c\prime}/\operatorname{Diag}(d))\backslash\text{RHS})./d$.

C.1 Entanglement-Based BB84

We consider this protocol with a single-photon source and restrict our discussions to the qubit space. In the quantum communication phase, Alice and Bob each receive one half of a bipartite state. This is supposed to be a two-qubit maximally entangled state before Eve's tampering. And the measure in the Z basis is with probability p_z, or in X basis with a probability $1-p_z$. In the classical communication phase, they announce their basis choices for each round and perform sifting to keep those rounds where they both chose the same basis. In the end, they generate keys from both Z and X bases.

In the simulation, we assume both bases have the same error rate $e_z = e_x = Q$. In particular, Γ (in (2.2)) contains the Z-basis error rate, X-basis error rate constraints as well as one coarse-grained constraint for each mismatched basis choice scenario. This is to ensure that Alice and Bob get completely uncorrelated outcomes in that case. In other words, the data γ are determined by Q. This test example is supposed to be numerically easy, since it involves the smallest possible size of ρ for **QKD**, i.e., four. Moreover, there is no reduced density operator ρ_A constraint for this example. In Table 5.1, instances of this test example are labeled as ebBB84(p_z, Q) for different values of p_z and Q.

C.2 Prepare-and-Measure BB84

Another protocol example in our numerical tests is the prepare-and-measure version of BB84 with a single-photon source. In the quantum communication phase, Alice chooses the Z basis with a probability p_z or the X basis with a probability $1-p_z$. When she chooses the Z basis, she sends either $|0\rangle$ or $|1\rangle$ at random, where $|0\rangle$ and $|1\rangle$ are eigenstates of the Pauli σ_Z operator. When she chooses the X basis, she sends either $|+\rangle$ or $|-\rangle$ at random, where $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$. After Alice sends the state of her choice to Bob, Bob chooses to measure in the Z basis with a probability p_z or the X basis with a probability $1-p_z$. The rest of the protocol is exactly the same as the entanglement-based BB84 protocol described in Appendix C.1. We call $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$ as stated in BB84.

For the security analysis, we use the source-replacement scheme [16] to convert it to its equivalent entanglement-based scheme. Therefore, the main differences between this example and the one in Appendix C.1 are: (1) the dimension of Alice's system for this example is four due to the source-replacement scheme, while it is two for the entanglement-based BB84; (2) there is the reduced density operator constraint ρ_A which is of size 4 and translated to 16 linear constraints. In this test example, the size of ρ is 8 and the size of $\mathcal{G}(\rho)$ is 32 before **FR**.

The data simulation is done in a similar way as that in the entanglement-based BB84 protocol, i.e., $e_x = e_z = Q$. In Table 5.1, instances of this test example are labeled as pmBB84(p_z, Q).

C.3 Measurement-Device-Independent BB84

In the measurement-device-independent variant of BB84 with single-photon sources, Alice and Bob each prepare one of four BB84 states (with the probability of choosing the Z basis as p_z). Then they both send this to an untrusted third-party Charlie for measurements. He ideally then performs the Bell-state measurements and announces the outcomes. We consider a setup where Charlie only uses linear optics, and thus can only measure two out of four Bell states. In this protocol, Charlie announces either a successful Bell-state measurement or a failure. If a successful measurement, Charlie then also announces the Bell state. Therefore, there are three possible announcement outcomes. After the announcement, Alice and Bob perform the basis sifting, as well as discard rounds that are linked to unsuccessful events. They then generate

keys from rounds where they both chose the Z basis and Charlie's announcement is one of the successful events.

We now consider the measurement-device-independent type of protocols. As described in [4], the optimization variable ρ involves three parties as ρ_{ABC}. Here, registers AB together serve the role of A in the reduced density operator constraint set (2.5). The dimension of Alice's system is 4 and so is Bob's dimension. The register C is a classical register that stores the announcement outcome. Thus it is three-dimensional with three possible announcement outcomes. In the data simulation, we assume that each qubit sent to Charlie goes through a depolarizing channel, with the depolarizing probability p.

In the numerical tests, we label instances of this protocol example as mdiBB84(p_z, p). The size of ρ is 48 and that of $\mathcal{G}(\rho)$ is 96 before **FR**.

C.4 Twin-Field **QKD**

As above, this protocol also uses the measurement-device-independent setup. The exact protocol description can be found in [34, Protocol 1]. In this protocol, Alice and Bob each prepare a state $|\phi_q\rangle_{Aa} = \sqrt{q}|0\rangle_A|0\rangle_a + \sqrt{1-q}|1\rangle_A|1\rangle_a$ ($|\phi_q\rangle_{Bb}$) with $0 \leq q \leq 1$, where the register A is a qubit system and the register a is an optical mode with the vacuum state $|0\rangle_a$ and the single-photon state $|1\rangle_a$. After they send states to the intermediate station, Charlie at the intermediate station is supposed to perform the single-photon interference of these two signal pulses and then announces the measurement outcome for each of two detectors: click or no-click. Then Alice and Bob each perform the X-basis measurement on their local qubits with a probability p_x or the Z-basis measurement with a probability $1 - p_x$. They generate keys from rounds where they both choose the X basis and where Charlie announces a successful measurement outcome, that is, having exactly one of two detectors click.

In the simulation, we consider a lossy channel, with the transmittance $10^{-0.02L}$, for the distance L in kilometers between Alice and Bob. We consider the symmetric scenario where Charlie is at an equal distance away from Alice and Bob. We also consider detector imperfections: each detector at Charlie's side has detector efficiency $\eta_d = 14.5\%$ and dark count probability $p_d = 10^{-8}$. In instances of this protocol, data is generated as a function of: q that appears in the states $|\phi_q\rangle_{Aa}$ and $|\phi_q\rangle_{Bb}$; the total distance L in kilometers between Alice and Bob; and the probability of choosing X basis p_x. The instances of this test example are labeled as TFQKD(q, L, p_x).

C.5 Discrete-Modulated Continuous-Variable **QKD**

The exact protocol description can be found in [26, Protocol 2]. We use the same simulation method described in [26, Equation (30)] to generate the data γ. In this protocol, Alice sends Bob one of four coherent states $|\alpha e^{i\theta_j}\rangle$, where $\theta_j = \frac{j\pi}{2}$ for $j = 0, 1, 2, 3$. And Bob performs the heterodyne measurement, i.e., measuring both X- and P-quadratures after splitting the signal into two halves by a 50/50 beamsplitter. The first and second moments of X- and P-quadratures are used to constrain ρ. The data simulation uses a phase-invariant Gaussian channel with transmittance η_t and excess noise ξ to generate those values. We use the same photon-number cutoff assumption used there to truncate the infinite-dimensional Hilbert space. For the calculation, it is typically sufficient to choose $N_c \geq 10$ to minimize the effects of errors due to the truncation. For simplicity, we assume the detector at Bob's side is an ideal detector. The channel transmittance, η_t, is related to the transmission distance L between Alice and Bob, by $\eta_t = 10^{-0.02L}$.

Let N_c be an integer that represents the cutoff photon number. Before **FR**, the sizes of ρ and

$\mathcal{G}(\rho)$ are $4(N_c+1)$ and $16(N_c+1)$, respectively. In Table 5.1, we label instances of this example as $\mathrm{DMCV}(N_c, L, \xi, \alpha)$.

When the noise $\xi = 0$, this problem can be solved analytically via physical arguments. The detailed instructions for analytical calculation can be found in [26, Appendix C]. We use this special case to demonstrate that our interior-point method can reproduce the analytical results to high precision.

C.6 Discrete-Phase-Randomized BB84

We consider the phase-encoding BB84 protocol with c (a parameter) discrete global phases evenly spaced between $[0, 2\pi]$ [32]. A detailed protocol description can also be found in [6, Sec. IV D]. In particular, each of four BB84 states is realized by a two-mode coherent state $|\alpha e^{i\theta}\rangle_r |\alpha e^{i(\theta+\phi_A)}\rangle_s$, where the first mode is the phase reference mode and the second mode encodes the private information. In particular, the θ is a global phase that involves discrete phase randomization, i.e., $\theta \in \{\frac{2\pi\ell}{c} : \ell = 0, \ldots, c-1\}$. The relative phase for encoding is $\phi_A \in \{\frac{j\pi}{2} : j = 0, 1, 2, 3\}$, where $\{0, \pi\}$ correspond to the Z basis and $\{\frac{\pi}{2}, \frac{3\pi}{2}\}$ correspond to the X basis.

Data simulation is done in exactly the same way as in [6, Section IV D], where we consider detector imperfections and a lossy channel with a misalignment error due to phase drift.

We remark that the instances of this test example become more challenging as one increases the number of discrete phases c, since the size of ρ is $12c$ and the size of $\mathcal{G}(\rho)$ is $48c$ before **FR**. In all instances of this protocol, we choose $p_z = 0.5$ and the data are simulated with detector efficiency $\eta_d = 0.045$, dark count probability $p_d = 8.5 \times 10^{-7}$ and relative phase drift of $11°$. The final key rate values are presented by taking the error correction efficiency as 1.16. In the numerical tests, we label instances of this protocol as $\mathrm{dprBB84}(c, \alpha, L)$.

C.7 Additional Numerical Results

Problem Data			Gauss-Newton		Frank-Wolfe with FR		Frank-Wolfe w/o FR		cvxquad with FR	
protocol	parameter	size	gap	time	gap	time	gap	time	gap	time
ebBB84	(0.50,0.01)	(4,16)	1.14e-12	0.42	5.96e-05	95.32	5.88e-05	99.09	6.37e-01	216.75
ebBB84	(0.50,0.03)	(4,16)	8.35e-13	0.20	6.37e-05	93.35	6.24e-05	99.47	5.88e-01	258.98
ebBB84	(0.50,0.05)	(4,16)	5.98e-13	0.18	1.01e-04	95.31	1.17e-04	101.36	5.46e-01	213.04
ebBB84	(0.50,0.07)	(4,16)	1.05e-12	0.19	1.66e-04	96.92	1.65e-04	100.46	5.07e-01	179.38
ebBB84	(0.50,0.09)	(4,16)	1.35e-12	0.18	1.43e-04	96.35	2.55e-04	100.64	4.70e-01	170.14
ebBB84	(0.70,0.01)	(4,16)	1.21e-13	0.21	7.60e-05	96.33	7.62e-05	99.14	7.06e-01	172.09
ebBB84	(0.70,0.03)	(4,16)	6.36e-13	0.18	9.16e-05	96.77	9.15e-05	99.78	6.59e-01	160.88
ebBB84	(0.70,0.05)	(4,16)	5.34e-13	0.18	1.67e-04	97.03	1.03e-04	100.73	6.14e-01	173.77
ebBB84	(0.70,0.07)	(4,16)	1.26e-12	0.20	1.80e-04	96.22	1.74e-04	100.71	5.70e-01	261.77
ebBB84	(0.70,0.09)	(4,16)	2.06e-13	0.18	3.85e-04	97.08	2.61e-04	101.55	5.26e-01	225.94
ebBB84	(0.90,0.01)	(4,16)	5.40e-13	0.17	1.02e-04	97.68	1.03e-04	98.45	8.73e-01	141.93
ebBB84	(0.90,0.03)	(4,16)	7.05e-13	0.21	1.25e-04	97.97	1.43e-04	99.50	8.27e-01	164.26
ebBB84	(0.90,0.05)	(4,16)	3.48e-13	0.18	1.48e-04	96.91	9.81e-05	100.99	7.83e-01	186.87
ebBB84	(0.90,0.07)	(4,16)	1.42e-12	0.17	2.71e-04	96.55	2.75e-04	101.79	7.39e-01	179.55
ebBB84	(0.90,0.09)	(4,16)	1.09e-12	0.21	3.51e-04	96.40	3.42e-04	101.99	6.94e-01	228.63

Table C.1: Numerical Report for ebBB84 Instances

Problem Data			Gauss-Newton		Frank-Wolfe with FR		Frank-Wolfe w/o FR		cvxquad with FR	
protocol	parameter	size	gap	time	gap	time	gap	time	gap	time
pmBB84	(0.50,0.01)	(8,32)	5.96e-13	0.38	6.19e-06	1.95	4.64e-04	25.18	6.30e-01	190.28
pmBB84	(0.50,0.03)	(8,32)	1.01e-12	0.18	1.71e-05	1.37	6.54e-04	90.80	5.74e-01	181.83
pmBB84	(0.50,0.05)	(8,32)	5.51e-13	0.19	1.12e-04	1.31	6.47e-04	1.94	5.26e-01	159.28
pmBB84	(0.50,0.07)	(8,32)	8.88e-14	0.16	5.89e-05	1.30	8.77e-04	1.98	4.81e-01	161.49
pmBB84	(0.50,0.09)	(8,32)	9.38e-13	0.19	6.71e-05	1.42	9.04e-04	2.03	4.40e-01	179.01
pmBB84	(0.70,0.01)	(8,32)	7.69e-13	0.25	7.62e-06	1.33	2.39e-04	130.73	7.03e-01	213.43
pmBB84	(0.70,0.03)	(8,32)	4.75e-13	0.18	2.38e-05	1.32	2.68e-04	133.52	6.51e-01	246.50
pmBB84	(0.70,0.05)	(8,32)	6.16e-13	0.17	3.52e-05	1.37	3.37e-04	134.59	6.04e-01	281.84
pmBB84	(0.70,0.07)	(8,32)	6.30e-13	0.19	7.43e-05	1.26	3.04e-04	141.97	5.60e-01	259.33
pmBB84	(0.70,0.09)	(8,32)	8.47e-13	0.16	9.25e-05	1.29	3.55e-04	7.07	5.18e-01	297.32
pmBB84	(0.90,0.01)	(8,32)	3.68e-13	0.18	7.07e-06	1.33	3.27e-04	4.98	8.60e-01	247.82
pmBB84	(0.90,0.03)	(8,32)	1.27e-12	0.18	2.29e-05	1.35	5.43e-04	137.74	7.96e-01	230.35
pmBB84	(0.90,0.05)	(8,32)	1.36e-12	0.16	4.62e-05	1.35	5.96e-04	72.04	7.38e-01	291.83
pmBB84	(0.90,0.07)	(8,32)	5.13e-13	0.17	7.31e-05	1.35	6.25e-04	40.11	6.84e-01	235.27
pmBB84	(0.90,0.09)	(8,32)	7.84e-13	0.19	1.06e-04	1.32	7.39e-04	142.32	6.32e-01	244.81

Table C.2: Numerical Report for pmBB84 Instances

Problem Data			Gauss-Newton		Frank-Wolfe with FR		Frank-Wolfe w/o FR		cvxquad with FR	
protocol	parameter	size	gap	time	gap	time	gap	time	gap	time
mdiBB84	(0.50,0.01)	(48,96)	1.25e-12	1.27	1.64e-05	109.60	3.75e-04	498.82	2.11e-01	719.99
mdiBB84	(0.50,0.03)	(48,96)	8.37e-13	0.81	3.53e-05	107.90	5.31e-04	2811.15	1.95e-01	630.89
mdiBB84	(0.50,0.05)	(48,96)	1.14e-12	0.83	4.99e-05	111.08	5.22e-04	464.06	1.82e-01	586.22
mdiBB84	(0.50,0.07)	(48,96)	1.35e-12	1.08	4.87e-05	335.57	4.60e-04	2169.93	1.71e-01	569.14
mdiBB84	(0.50,0.09)	(48,96)	1.25e-12	0.91	8.27e-05	342.04	4.37e-04	829.61	1.60e-01	568.97
mdiBB84	(0.70,0.01)	(48,96)	1.20e-12	0.68	2.20e-05	5.68	5.53e-04	901.29	3.79e-01	719.27
mdiBB84	(0.70,0.03)	(48,96)	4.24e-13	1.05	1.11e-04	5.83	1.21e-03	256.79	3.55e-01	670.49
mdiBB84	(0.70,0.05)	(48,96)	1.06e-12	1.10	6.92e-05	340.21	1.75e-03	865.68	3.31e-01	651.44
mdiBB84	(0.70,0.07)	(48,96)	5.71e-13	0.95	1.37e-04	337.17	1.55e-03	864.25	3.09e-01	604.94
mdiBB84	(0.70,0.09)	(48,96)	1.57e-13	0.92	1.61e-04	347.21	2.24e-03	872.37	2.88e-01	604.71
mdiBB84	(0.90,0.01)	(48,96)	8.44e-13	0.66	4.42e-05	343.38	3.21e-03	175.20	5.53e-01	710.65
mdiBB84	(0.90,0.03)	(48,96)	1.39e-12	0.90	9.15e-05	343.08	3.66e-03	301.29	5.16e-01	671.95
mdiBB84	(0.90,0.05)	(48,96)	9.88e-13	0.84	1.73e-04	345.79	4.64e-03	519.80	4.85e-01	644.23
mdiBB84	(0.90,0.07)	(48,96)	2.96e-13	1.05	2.04e-04	338.15	2.85e-03	148.24	4.57e-01	580.20
mdiBB84	(0.90,0.09)	(48,96)	5.21e-13	1.02	2.52e-04	343.95	3.26e-03	193.04	4.31e-01	595.07

Table C.3: Numerical Report for mdiBB84 Instances

Problem Data			Gauss-Newton		Frank-Wolfe with FR		Frank-Wolfe without FR	
protocol	parameter	size	gap	time	gap	time	gap	time
TFQKD	(0.75,50.00,0.70)	(12,24)	8.45e-13	1.33	2.72e-09	2.04	1.83e-03	155.08
TFQKD	(0.75,100.00,0.70)	(12,24)	1.42e-12	0.86	3.75e-09	1.35	1.53e-03	153.57
TFQKD	(0.75,150.00,0.70)	(12,24)	9.94e-13	0.78	2.82e-09	1.39	7.82e-04	162.13
TFQKD	(0.75,200.00,0.70)	(12,24)	1.15e-12	1.02	3.98e-09	1.37	7.19e-04	155.11
TFQKD	(0.75,250.00,0.70)	(12,24)	6.79e-13	0.79	8.21e-09	1.32	3.14e-04	172.53
TFQKD	(0.80,50.00,0.70)	(12,24)	1.15e-12	0.77	2.82e-09	1.39	1.52e-03	156.17
TFQKD	(0.80,100.00,0.70)	(12,24)	1.15e-12	1.08	2.60e-09	1.33	1.57e-03	156.11
TFQKD	(0.80,150.00,0.70)	(12,24)	1.25e-12	0.94	2.97e-09	1.29	8.30e-04	158.12
TFQKD	(0.80,200.00,0.70)	(12,24)	9.23e-13	0.73	4.23e-09	1.32	5.60e-04	155.45
TFQKD	(0.80,250.00,0.70)	(12,24)	3.91e-13	0.55	2.22e-09	1.35	1.97e-04	164.83
TFQKD	(0.90,50.00,0.70)	(12,24)	8.08e-13	0.79	4.30e-09	1.30	1.55e-03	156.38
TFQKD	(0.90,100.00,0.70)	(12,24)	1.38e-12	0.37	3.62e-09	1.31	1.23e-03	154.08
TFQKD	(0.90,150.00,0.70)	(12,24)	1.14e-12	0.61	2.87e-09	1.28	6.02e-04	161.73
TFQKD	(0.90,200.00,0.70)	(12,24)	1.04e-12	0.34	3.98e-09	1.36	1.68e-04	2.63
TFQKD	(0.90,250.00,0.70)	(12,24)	5.77e-13	0.43	2.77e-09	1.37	1.08e-05	2.07
TFQKD	(0.95,50.00,0.70)	(12,24)	9.84e-13	0.77	4.00e-09	1.43	1.38e-03	156.09
TFQKD	(0.95,100.00,0.70)	(12,24)	1.06e-12	0.70	4.36e-09	1.41	7.47e-04	153.19
TFQKD	(0.95,150.00,0.70)	(12,24)	1.36e-12	0.71	3.92e-09	1.35	8.38e-04	149.27
TFQKD	(0.95,200.00,0.70)	(12,24)	7.63e-13	0.65	2.60e-09	1.36	3.13e-04	150.75
TFQKD	(0.95,250.00,0.70)	(12,24)	1.02e-12	0.64	3.61e-09	1.42	5.56e-06	1.81

Table C.4: Numerical Report for TFQKD Instances

Problem Data			Gauss-Newton		Frank-Wolfe with FR		Frank-Wolfe without FR	
protocol	parameter	size	gap	time	gap	time	gap	time
DMCV	(10.00,60.00,0.05,0.35)	(44,176)	2.71e-09	1016.19	4.35e-06	612.62	3.57e-06	919.84
DMCV	(10.00,120.00,0.05,0.35)	(44,176)	2.70e-09	1090.41	2.27e-06	216.61	2.16e-06	277.47
DMCV	(10.00,180.00,0.05,0.35)	(44,176)	2.87e-09	1173.14	1.90e-07	23.64	1.36e-07	32.50
DMCV	(11.00,60.00,0.05,0.35)	(48,192)	3.05e-09	1419.71	2.50e-06	768.52	1.43e-06	1095.78
DMCV	(11.00,120.00,0.05,0.35)	(48,192)	3.24e-09	1484.12	2.35e-06	261.89	2.15e-06	380.63
DMCV	(11.00,180.00,0.05,0.35)	(48,192)	3.40e-09	1796.16	2.53e-07	26.98	1.59e-07	38.62
DMCV	(10.00,150.00,0.02,0.70)	(44,176)	2.07e-09	1143.23	1.67e-06	83.32	1.82e-06	99.96
DMCV	(10.00,200.00,0.02,0.70)	(44,176)	1.96e-09	1214.62	7.29e-07	21.26	7.06e-07	28.04
DMCV	(10.00,150.00,0.02,0.80)	(44,176)	3.27e-09	1113.17	1.10e-06	105.03	1.85e-06	106.68
DMCV	(10.00,200.00,0.02,0.80)	(44,176)	1.63e-09	1106.38	3.18e-07	19.61	2.90e-07	26.11
DMCV	(11.00,150.00,0.02,0.70)	(48,192)	3.31e-09	1607.78	1.72e-06	103.49	1.36e-06	150.76
DMCV	(11.00,200.00,0.02,0.70)	(48,192)	3.05e-09	1573.12	7.12e-07	27.40	6.68e-07	35.62
DMCV	(11.00,150.00,0.02,0.80)	(48,192)	3.37e-09	1597.15	1.36e-06	93.86	1.57e-06	115.21
DMCV	(11.00,200.00,0.02,0.80)	(48,192)	3.38e-09	1541.27	3.38e-07	25.34	2.99e-07	34.36

Table C.5: Numerical Report for DMCV Instances

Problem Data			Gauss-Newton		Frank-Wolfe with FR		Frank-Wolfe without FR	
protocol	parameter	size	gap	time	gap	time	gap	time
dprBB84	(1.00,0.08,15.00)	(12,48)	9.42e-13	1.66	3.85e-06	117.56	1.03e-04	189.23
dprBB84	(1.00,0.08,30.00)	(12,48)	4.92e-13	1.70	3.85e-06	114.97	9.43e-05	178.82
dprBB84	(1.00,0.14,15.00)	(12,48)	2.96e-13	0.91	3.63e-04	115.55	1.16e-02	178.91
dprBB84	(1.00,0.14,30.00)	(12,48)	5.21e-13	1.43	2.60e-04	1.89	8.31e-03	2.51
dprBB84	(2.00,0.08,15.00)	(24,96)	1.10e-12	22.06	9.58e-05	41.11	2.11e-03	82.96
dprBB84	(2.00,0.08,30.00)	(24,96)	9.58e-13	22.26	1.17e-04	26.10	7.32e-06	114.66
dprBB84	(2.00,0.14,15.00)	(24,96)	1.35e-12	24.93	1.89e-05	7.56	5.11e-04	16.59
dprBB84	(2.00,0.14,30.00)	(24,96)	1.04e-12	24.63	5.71e-06	20.79	5.38e-06	42.06
dprBB84	(2.00,0.14,30.00)	(24,96)	1.04e-12	24.82	5.71e-06	20.19	5.38e-06	44.72
dprBB84	(3.00,0.08,15.00)	(36,144)	1.38e-12	129.39	2.36e-04	12.94	7.41e-03	33.59
dprBB84	(3.00,0.08,30.00)	(36,144)	6.33e-13	139.34	2.26e-04	12.54	7.04e-03	36.53
dprBB84	(3.00,0.14,15.00)	(36,144)	1.30e-12	118.84	4.32e-05	41.49	1.43e-04	50.07
dprBB84	(3.00,0.14,30.00)	(36,144)	3.32e-13	127.94	5.80e-06	11.32	5.74e-06	37.63
dprBB84	(4.00,0.08,15.00)	(48,192)	6.98e-09	766.17	2.88e-04	61.19	8.10e-03	235.02
dprBB84	(4.00,0.08,30.00)	(48,192)	2.13e-09	786.01	2.97e-04	21.23	8.45e-03	232.35
dprBB84	(4.00,0.14,15.00)	(48,192)	2.85e-12	539.08	1.29e-04	18.50	3.85e-03	201.96
dprBB84	(4.00,0.14,30.00)	(48,192)	1.17e-12	545.04	1.26e-04	25.67	3.73e-03	209.05

Table C.6: Numerical Report for dprBB84 Instances

We remark on the behavior of the Frank-Wolfe method when used without **FR**. Table 5.1 shows that two instances of the TFQKD protocol have vastly different running times. Similar situations without **FR** also occur in the tables presented here. The Frank-Wolfe method that is used and described in [5] adopts a two-step procedure. The running time of this method is mainly dependent on the number of iterations, since each iteration solves a linear **SDP** problem using CVX. There are two stopping conditions: the first depends on the value of the gradient information; the second is the maximum number of iterations, currently set at 300. For the instances with short running time, termination typically occurred due to the gradient condition; the other instances stopped after reaching the maximum number of iterations. A likely reason for early termination is the perturbation approach used to preserve positive definiteness of the matrices involved in the objective function. We emphasize that early termination still leads to reliable (but pessimistically lower) key rates. More explanations can be found in [5]. As discussed in this paper, a general motivation for **FR** is to avoid numerical instability associated with such perturbation approaches, i.e., without **FR**, we typically have the erratic behavior due to ill-posedness of the problems.

Index

References

[1] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev. The security of practical quantum key distribution. *Rev. Mod. Phys.*, 81:1301, 2009. DOI: 10.1103/RevModPhys.81.1301. 4

[2] F. Xu, X. Ma, Q. Zhang, H.-K. Lo, and J.-W. Pan. Secure quantum key distribution with realistic devices. *Rev. Mod. Phys.*, 92:025002, 2020. DOI: 10.1103/RevModPhys.92.025002. 4

[3] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *International Conference on Computers, Systems & Signal Processing, Bangalore, India, Dec 9-12, 1984*, pages 175–179, 1984. DOI: 10.1016/j.tcs.2014.05.025. Reprint of the 1984 original. 4, 39

[4] P. J. Coles, E. M. Metodiev, and N. Lütkenhaus. Numerical approach for unstructured quantum key distribution. *Nat. Commun.*, 7:11712, 2016. DOI: 10.1038/ncomms11712. 4, 41

[5] A. Winick, N. Lütkenhaus, and P. J. Coles. Reliable numerical key rates for quantum key distribution. *Quantum*, 2:77, 2018. DOI: 10.22331/q-2018-07-26-77. 4, 5, 7, 24, 25, 28, 29, 34, 44

[6] I. George, J. Lin, and N. Lütkenhaus. Numerical calculations of the finite key rate for general quantum key distribution protocols. *Physical Review Research*, 3:013274, 2021. DOI: 10.1103/PhysRevResearch.3.013274. 4, 39, 42

[7] Y. Zhang, P. J. Coles, A. Winick, J. Lin, and N. Lütkenhaus. Security proof of practical quantum key distribution with detection-efficiency mismatch. *Phys. Rev. Research*, 3:013076, 2021. DOI: 10.1103/PhysRevResearch.3.013076. 4, 30

[8] T. Upadhyaya, T. van Himbeeck, J. Lin, and N. Lütkenhaus. Dimension reduction in quantum key distribution for continuous- and discrete-variable protocols. *PRX Quantum*, 2: 020325, 2021. DOI: 10.1103/PRXQuantum.2.020325. 4, 30

[9] N. K. H. Li and N. Lütkenhaus. Improving key rates of the unbalanced phase-encoded bb84 protocol using the flag-state squashing model. *Phys. Rev. Research*, 2:043172, 2020. DOI: 10.1103/PhysRevResearch.2.043172. 4

[10] L. Faybusovich and C. Zhou. Long-step path-following algorithm for solving symmetric programming problems with nonlinear objective functions. *Computational Optimization and Applications*, 72(3):769–795, 2019. ISSN 15732894. DOI: 10.1007/s10589-018-0054-7. 5, 30, 34

[11] I. Devetak and A. Winter. Distillation of secret key and entanglement from quantum states. *Proc. R. Soc. A*, 461:207–235, 2005. DOI: 10.1098/rspa.2004.1372. 7

[12] M. A. Nielsen and I. L. Chuang, editors. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, UK, 2000. DOI: 10.1017/CBO9780511976667. 7, 11

[13] D. Drusvyatskiy and H. Wolkowicz. The many faces of degeneracy in conic optimization. *Foundations and Trends® in Optimization*, 3(2):77–170, 2017. ISSN 2167-3888. DOI: /10.1561/2400000011. 9, 10, 35

[14] J. Watrous. *The Theory of Quantum Information*. Cambridge University Press, Cambridge, UK, 2018. ISBN 1107180562. DOI: 10.1017/9781316848142. 11

[15] P. J. Coles. Unification of different views of decoherence and discord. *Phys. Rev. A*, 85: 042103, 2012. DOI: 10.1103/PhysRevA.85.042103. 11

[16] A. Ferenczi and N. Lütkenhaus. Symmetries in quantum key distribution and the connection between optimal attacks and optimal cloning. *Phys. Rev. A*, 85:052310, 2012. DOI: 10.1103/PhysRevA.85.052310. 12, 40

[17] J. M. Borwein and H. Wolkowicz. Regularizing the abstract convex program. *J. Math. Anal. Appl.*, 83(2):495–530, 1981. ISSN 0022-247X. DOI: 10.1017/S1446788700017250. 13

[18] S. Sremac, H. J. Woerdeman, and H. Wolkowicz. Error bounds and singularity degree in semidefinite programming. *SIAM J. Optim.*, 31(1):812–836, 2021. ISSN 1052-6234. DOI: 10.1137/19M1289327. 13

[19] R. T. Rockafellar. *Convex analysis*. Princeton Mathematical Series, No. 28. Princeton University Press, Princeton, N.J., 1970. DOI: 10.1515/9781400873173. 16

[20] D. G. Luenberger and Y. Ye. *Linear and Nonlinear Programming*, volume 116 of *International series in operations research & management science*. Springer, Boston, USA, 2008. ISBN 9781441945044. DOI: 10.1007/978-0-387-74503-9. 18

[21] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, NY, USA, second edition, 2006. ISBN 978-0387-30303-1; 0-387-30303-0. DOI: 10.1007/978-0-387-40065-5. 18

[22] J. P. Dedieu and M. Shub. Newton's method for overdetermined systems of equations. *Math. Comp.*, 69(231):1099–1115, 2000. ISSN 0025-5718. DOI: 10.1090/S0025-5718-99-01115-1. 19

[23] J. E. Dennis Jr. and R. B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*, volume 16 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996. ISBN 0-89871-364-1. DOI: 10.1137/1.9781611971200. Corrected reprint of the 1983 original. 19

[24] R. D. C. Monteiro and M. J. Todd. Path-following methods. In *Handbook of Semidefinite Programming*, volume 27 of *International Series in Operations Research & Management Science*, pages 267–306. Springer, Boston, MA, 2000. DOI: 10.1007/978-1-4615-4381-7˙10. 19

[25] J. W. Demmel. *Applied numerical linear algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997. ISBN 0-89871-389-7. DOI: 10.1137/1.9781611971446. 24

[26] J. Lin, T. Upadhyaya, and N. Lütkenhaus. Asymptotic security analysis of discrete-modulated continuous-variable quantum key distribution. *Phys. Rev. X*, 9:041064, 2019. DOI: 10.1103/PhysRevX.9.041064. 25, 26, 39, 41, 42

[27] H. Fawzi, J. Saunderson, and P. A. Parrilo. Semidefinite approximations of the matrix logarithm. *Foundations of Computational Mathematics*, 19:259–296, 2019. DOI: 10.1007/s10208-018-9385-0. Package cvxquad at `https://github.com/hfawzi/cvxquad`. 28, 29

[28] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004. DOI: 10.1017/CBO9780511804441. 36

[29] D. G. Luenberger. *Optimization by Vector Space Methods*. John Wiley & Sons, New York, USA, 1969. 36

[30] J. E. Dennis Jr. and H. Wolkowicz. Sizing and least-change secant methods. *SIAM J. Numer. Anal.*, 30(5):1291–1314, 1993. ISSN 0036-1429. DOI: 10.1137/0730067. 39

[31] H.-K. Lo, M. Curty, and B. Qi. Measurement-device-independent quantum key distribution. *Phys. Rev. Lett.*, 108:130503, 2012. DOI: 10.1103/PhysRevLett.108.130503. 39

[32] Z. Cao, Z. Zhang, H.-K. Lo, and X. Ma. Discrete-phase-randomized coherent state source and its application in quantum key distribution. *New J. Phys.*, 17:053014, 2015. DOI: 10.1088/1367-2630/17/5/053014. 39, 42

[33] M. Lucamarini, Z. L. Yuan, J. F. Dynes, and A. J. Shields. Overcoming the rate-distance limit of quantum key distribution without quantum repeaters. *Nature*, 557:400–403, 2018. DOI: 10.1038/s41586-018-0066-6. 39

[34] M. Curty, K. Azuma, and H.-K. Lo. Simple security proof of twin-field type quantum key distribution protocol. *npj. Quantum Inf.*, 5:64, 2019. DOI: 10.1038/s41534-019-0175-6. 41