A polynomial-based QCQP solver for encrypted optimization

Sebastian Schlor, Andrea Iannelli, Junsoo Kim, Hyungbo Shim and Frank Allgöwer

Abstract— **In this paper, we present a novel method for solving a class of quadratically constrained quadratic optimization problems using only additions and multiplications. This approach enables solving constrained optimization problems on private data since the operations involved are compatible with the capabilities of homomorphic encryption schemes. To solve the constrained optimization problem, a sequence of polynomial penalty functions of increasing degree is introduced, which are sufficiently steep at the boundary of the feasible set. Adding the penalty function to the original cost function creates a sequence of unconstrained optimization problems whose minimizer always lies in the admissible set and converges to the minimizer of the constrained problem. A gradient descent method is used to generate a sequence of iterates associated with these problems. For the algorithm, it is shown that the iterate converges to a minimizer of the original problem, and the feasible set is positively invariant under the iteration. Finally, the method is demonstrated on an illustrative cryptographic problem, finding the smaller value of two numbers, and the encrypted implementability is discussed.**

I. Introduction

Optimization algorithms are key elements of many modern technologies such as artificial intelligence, machine learning or model predictive control (MPC). At the same time, these applications often deal with sensitive data and handle safety-critical tasks. To enable privacy-preserving control and optimization, homomorphic cryptosystems have been adopted to perform computations entirely on encrypted data. However, one of the limitations of this technology is that only polynomial operations, i.e., addition and multiplication are supported. More complex operations, e.g., divisions and comparisons, have to be approximated, which leads to a higher computational effort and less accuracy. This challenge has prevented powerful state-of-the-art optimization algorithms to be applied under encryption, in particular for constrained optimization problems, since these solvers generally need projections or comparisons, which cannot be natively handled in homomorphic cryptosystems.

In this paper, we present a novel approach to constrained optimization using only polynomial operations. This enables a straightforward implementation in a homomorphically encrypted fashion. In the proposed algorithm, the constraints are replaced by a sequence of polynomial penalty functions, which are added to the original cost function. To gradually approach the minimum of the original problem, a sequential gradient descent on the resulting sequence of unconstrained optimization problems is performed.

A. Related work

For convex unconstrained quadratic optimization problems, encrypted gradient and accelerated gradient methods have been analyzed and implemented in [1], and distributed encrypted alternating direction method of multipliers (ADMM) has been proposed in [2]. The works [3]–[6] consider a linearly constrained quadratic program with private inputs from multiple parties. The solution is obtained via a projected gradient ascent or projected fast gradient descent algorithm, where the critical projection operation is done by a target node in plaintext or a two-party protocol [7], involving more communication steps. To solve quadratically constrained quadratic programs (QCQPs) occurring in encrypted MPC, there are approaches using real-time iterations of the proximal gradient method [8] or ADMM [9]–[11]. Thereby, the projections are done by the plant in plaintext. Alternatives to online optimization for MPC were proposed in [11]–[14], where an explicit MPC solution was computed offline and the identification of the active region was performed by the plant, a two-party protocol, or a garbled circuit. All existing approaches to deal with constraints need a trusted party and decryption or two-party concepts, which are demanding from a computation or communication perspective.

For our approach, we utilize ideas from penalty and barrier methods. For an introduction, see, e.g., [15]. Both types replace the constrained optimization problem by a sequence of unconstrained problems, which are easier to solve and approximate the solution of the original problem. Penalty functions are usually defined as being zero inside the feasible set, and larger than zero outside [16]. Barrier functions approach infinity at the boundary of the feasible set [17]. The auxiliary problems then weigh the original cost function with a growing/decreasing influence of the penalty/barrier function. Because penalty functions are often defined piecewise, and the barrier functions have to grow unbounded on a finite domain, they cannot directly be used with polynomial-based homomorphic encryption.

F. Allgöwer, A. Iannelli, and S. Schlor are thankful that this work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2075 – 390740016. F. Allgöwer further thanks for the grants AL 316/13-2 – 285825138 and AL 316/15-1 – 468094890. Hyungbo Shim and Junsoo Kim gratefully acknowledge funding by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MSIT) under Grant RS-2022-00165417 and Grant RS-2024-00353032. Parts of the paper were prepared when S. Schlor was staying at ASRI, Seoul National University, South Korea.

S. Schlor, A. Iannelli, and F. Allgöwer are with the University of Stuttgart, Institute for Systems Theory and Automatic Control, Germany. {schlor, iannelli, allgower}@ist.uni-stuttgart.de.

Junsoo Kim is with the Seoul National University of Science and Technology, Department of Electrical and Information Engineering, South Korea. junsookim@seoultech.ac.kr.

Hyungbo Shim is with the Seoul National University, ASRI, Department of Electrical and Computer Engineering, South Korea. hshim@snu.ac.kr.

B. Contribution

We propose a novel method to solve a class of constrained optimization problems only using polynomial operations (additions and multiplications). A detailed analysis of the algorithm is provided, in which convergence to a minimizer and positive invariance with respect to the constraints are shown. In particular, we make the following contributions: We introduce a sequence of polynomial penalty functions for convex quadratic constraint sets. When added to the original cost function, we obtain a sequence of unconstrained optimization problems. For the resulting sequence of unconstrained problems, we prove that

- each minimizer always lies inside the feasible set, and
- the sequence of minimizers converges to a minimizer of the original problem as the polynomial degree increases.

To solve the original problem, we then propose a sequential gradient descent method generating a sequence of iterates associated with the aforementioned sequence of unconstrained problems. For this algorithm, we show that

- the feasible set is positively invariant, and
- the iterates converge to a minimizer of the original problem.

Finally, we discuss its benefits for homomorphically encrypted implementations and give an illustrative example.

C. Notation

We define the natural numbers as $\mathbb{N} = \{1, 2, 3 \ldots\}$. By $\langle x, y\rangle$ we denote the inner product of the vectors x and y. By $\succ 0$ ($\succeq 0$) we denote positive (semi-) definiteness. For positive semi-definite matrices, we use the Loewner order, i.e., we say that $A \succeq B$ if $A - B \succeq 0$. The boundary of a set $\mathcal{C}$ is denoted by $\partial\mathcal{C}$. By $\sigma(A)$ we denote the set of singular values, and by $\bar{\sigma}(A)$ ($\underline{\sigma}(A)$) the largest (smallest) singular value of A. The smallest integer greater than or equal to a given number is obtained using the ceiling operation, denoted by $\lceil\cdot\rceil$.

II. Problem Setup and Main Idea

In this section, we introduce the original constrained optimization problem and describe the overall approach, that is detailed in the following sections.

A. Problem setup

We want to solve the constrained optimization problem

$$\begin{aligned} \mathcal{X}^\star = \arg\min_x \;\; & f(x) \\ \text{s.t.} \;\; & x \in \mathcal{C} \end{aligned} \tag{1}$$

with the quadratic cost function

$$f(x) = \frac{1}{2} x^\top Q x + q^\top x$$

and the constraint set

$$\mathcal{C} = \{\, x \mid g(x) \leq 1 \,\}, \quad \partial\mathcal{C} = \{\, x \mid g(x) = 1 \,\},$$
$$g(x) = (x - v)^\top A (x - v)$$

with $x \in \mathbb{R}^n$, $Q \succeq 0$, $A \succ 0$ and $q, v \in \mathbb{R}^n$. This is a special case of a convex QCQP. Further, $\mathcal{C}$ should have nonzero volume, i.e., A should be bounded. Since f is continuous and $\mathcal{C}$ is compact, the set of minimizers $\mathcal{X}^\star$ is nonempty and compact, but the minimizer might not necessarily be a singleton.

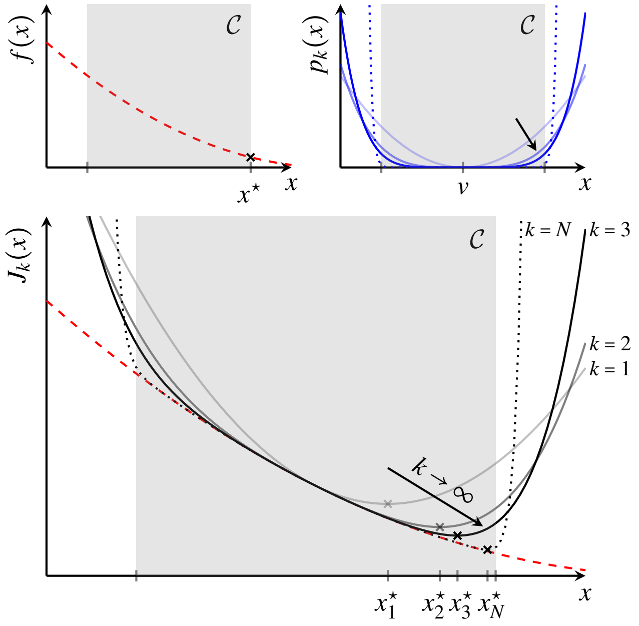

Fig. 1. Example for the sequence of cost functions $\{J_k\}_k$ based on the original cost function f and the sequence of penalty functions $\{p_k\}_k$ for the set $\mathcal{C}$ for $k \in \{1, 2, 3, N\}$ with $N = 15$. The sequence of minimizers $\{x_k^\star\}_k$ converges to the minimizer $x^\star$ of the constrained problem as $k \to \infty$ (cf. Proposition 1).

B. Main idea

Our solution approach is based on the fact that unconstrained convex polynomial optimization problems can be solved by gradient descent algorithms only involving polynomial computations. Since the gradient of a polynomial function is a polynomial, and gradient descent steps only involve multiplication by a step size and addition to the previous value, the overall approach is polynomial and suitable for encrypted evaluation.

However, since the constrained problem cannot be exactly transformed into a polynomial unconstrained problem, we employ techniques from interior point methods and construct a converging sequence of unconstrained auxiliary problems. For this purpose, we introduce a sequence of polynomial penalty functions p_k and a sequence of auxiliary cost functions $J_k(x) = f(x) + p_k(x)$. The construction of the penalty is based on the observation that monomials of increasing power tend towards zero for small values and grow rapidly for large values. By using penalty polynomials that tend to zero inside the allowed set and grow rapidly outside, our approach is a novel intermediate concept between the classical barrier and penalty methods. A key distinction of our approach is that instead of defining the sequence of problems only by adjusting a weighting parameter, the shape itself of our penalty function changes. Specifically, the penalty term is chosen such that the auxiliary cost functions have their minimum within the feasible set and this minimizer converges to a minimizer of the original problem as the index k increases. An exemplary sequence of cost functions is depicted in Figure 1. Since the penalty function cannot approach infinity at the boundary of the feasible set, and we cannot use Newton-style iterations, we need new concepts and proof techniques.

To solve this sequence of problems and thus find a solution to the original constrained problem, we apply a sequential gradient descent algorithm to the sequence of auxiliary problems. After each gradient step, the index of the cost function increases by one. Intuitively, the gradient steps track the auxiliary minimizer, which converges to the original minimizer, resulting in a convergent algorithm.

We make the construction of the penalty function and the minimization algorithm precise in the following sections.

III. Sequence of unconstrained problems

In this section, we introduce the sequence of auxiliary unconstrained problems and analyze its properties.

A. Design of the unconstrained problems

We choose the sequence of penalty function as

$$p_k(x) = m\frac{1}{k}g(x)^k$$

for $k \in \mathbb{N}$ and a parameter $m \geq 0$ that must satisfy a precise relationship described later. Its gradient is given by

$$\nabla p_k(x) = m g(x)^{k-1} \nabla g(x).$$

The auxiliary cost function is

$$J_k(x) = f(x) + p_k(x)$$

and we obtain the auxiliary unconstrained problem

$$x_k^\star = \arg\min_x \; J_k(x). \tag{2}$$

To make sure that every auxiliary Problem (2) has a minimum inside the feasible set, the parameter m has to be chosen such that $-\nabla J_k(x)$ is zero or points towards the interior of $\mathcal{C}$ for every point on the boundary $\partial\mathcal{C}$. Hence, we make the following requirement.

Requirement 1 (Scaling for minimum inside $\mathcal{C}$): The parameter m satisfies $m \geq m_{\min} := \max(\hat{m}_{\min}, 0)$ with

$$\hat{m}_{\min} = \max_x \; -\frac{\langle \nabla g(x), \nabla f(x)\rangle}{\langle \nabla g(x), \nabla g(x)\rangle} \quad \text{s.t. } g(x) = 1.$$

This condition is equivalent to $\nabla J_k(x)$ and $\nabla g(x)$ forming an acute angle at the boundary ($g(x) = 1$), that is

$$\langle \nabla g(x), \nabla J_k(x)\rangle \geq 0 \quad \Leftrightarrow \quad -\frac{\langle \nabla g(x), \nabla f(x)\rangle}{\langle \nabla g(x), \nabla g(x)\rangle} \leq m. \tag{3}$$

This is clearly satisfied if m is chosen as in Requirement 1. The value $m_{\min}$ can be interpreted as the largest ratio between the directional derivatives of $f(x)$ and $g(x)$ in the direction of $\nabla g(x)$ at the boundary $\partial\mathcal{C}$. While finding the exact $m_{\min}$ can be challenging, we note that any upper bound $m \geq m_{\min}$ is valid. A negative $\hat{m}_{\min}$ means that the minimizer of the original problem is inside the constraint set. We require $m_{\min} \geq 0$, since a negative penalization could render the problem nonconvex.

B. Analysis of the unconstrained problems

Consider the auxiliary unconstrained Problem (2) satisfying Requirement 1. First, let us show that the minimizer is well-defined.

Lemma 1: For every $k \in \mathbb{N}$, Problem (2) has a unique solution.

Proof: The objective function J_k is smooth, radially unbounded, and strictly convex. Hence, the minimum is attained at a finite value $x_k^\star$ and unique. ■

The intuition of a vanishing penalty inside the feasible set as k tends to infinity can be made precise as follows.

Lemma 2: The auxiliary cost function J_k uniformly converges on the set $\mathcal{C}$ to the original cost function f as $k \to \infty$, i.e., $\forall \varepsilon > 0 \, \exists N\colon \forall k \geq N \, \forall x \in \mathcal{C}\colon |J_k(x) - f(x)| < \varepsilon$.

Proof: Pick $\varepsilon > 0$ and $N = 2\left\lceil \frac{m}{\varepsilon} \right\rceil$. Then, $|J_k(x) - f(x)| = p_k(x) = m\frac{1}{k}g(x)^k \leq m\frac{1}{k} \leq m\frac{1}{N} \leq \frac{1}{2}\varepsilon < \varepsilon$, where we used that $g(x) \leq 1$ for $x \in \mathcal{C}$. ■

With this, we arrive at our first important result concerning convergence towards the set of minimizers $\mathcal{X}^\star$.

Proposition 1: Let Requirement 1 hold. Then, the auxiliary minimizer $x_k^\star$ of Problem (2)

1) is contained in the set $\mathcal{C}$ for every $k \in \mathbb{N}$, i.e., $x_k^\star \in \mathcal{C} \forall k \in \mathbb{N}$,
2) converges to $x^\star = \arg\min_{x \in \mathcal{X}^\star} g(x) \in \mathcal{X}^\star$ as $k \to \infty$, i.e., $x_k^\star \to x^\star$ as $k \to \infty$.

Proof: Assume $x_k^\star \notin \mathcal{C}$. Then, there exists a point $x \in \partial\mathcal{C}$, and a vector $d \neq 0$ pointing out of $\mathcal{C}$, such that the directional derivative of J_k at that point x along the vector d is negative. However, by Requirement 1, the directional derivative of J_k along any vector pointing out of $\mathcal{C}$ is nonnegative. By contradiction, $x_k^\star \in \mathcal{C}$. This proves Part 1).

The minimizer $x^\star = \arg\min_{x \in \mathcal{X}^\star} g(x)$ exists and is attained at a unique point since g is strongly convex and $\mathcal{X}^\star$ is convex and compact. Let us define the sets $\Theta = \{x \mid g(x) \leq g(x^\star)\} \subseteq \mathcal{C}$ and $\Omega_k = \{x \mid f(x) \leq f(x^\star) + p_k(x^\star)\}$. Then, $x_k^\star \in \Theta \cap \Omega_k$. This is true since $x_k^\star$ is the minimizer of J_k, in particular,

$$f(x_k^\star) + p_k(x_k^\star) \leq f(x^\star) + p_k(x^\star).$$

Since $f(x^\star) \leq f(x_k^\star)$, we have $p_k(x_k^\star) \leq p_k(x^\star)$, and hence $g(x_k^\star) \leq g(x^\star)$. Thus, $x_k^\star \in \Theta \subseteq \mathcal{C}$. Further, since $p_k(x_k^\star) \geq 0$, also $f(x_k^\star) \leq f(x^\star) + p_k(x^\star)$, and hence, $x_k^\star \in \Omega_k$.

Since due to Lemma 2 $p_k(x^\star)$ uniformly converges to zero, the set Ω_k converges to the set $\mathcal{X}^\star$ as $k \to \infty$. Hence, $\Theta \cap \Omega_k \to \Theta \cap \mathcal{X}^\star$ as $k \to \infty$. Since $x^\star = \arg\min_{x \in \mathcal{X}^\star} g(x)$ is the only element of $\Theta \cap \mathcal{X}^\star$, $x_k^\star \to x^\star$. This proves Part 2). ■

The following property becomes important for the gradient method in the next section.

Lemma 3: For every $k \in \mathbb{N}$, the auxiliary cost function J_k is L_k-smooth inside $\mathcal{C}$, i.e., $\forall k \in \mathbb{N} \, \exists L_k \geq 0 : \nabla^2 J_k(x) \leq L_k I \, \forall x \in \mathcal{C}$, and a smoothness constant is given as $L_k = \bar{\sigma}(Q + m(4k-2)A)$.

Proof: The proof is given in Appendix A. ■

IV. Sequential gradient descent

To find a solution to the problem presented in Section III, and thus to the original problem, we employ gradient descent since it only involves multiplication of the step size and the gradient, and addition to the previous iterate.

Consider a sequential gradient descent algorithm

$$x_{k+1} = x_k - \gamma_k \nabla J_k(x_k), \tag{4}$$

where we sequentially update the cost function after every gradient step. We choose $0 < \gamma_k \leq \frac{1}{L_k}$ as step size, and require that the sequence $\{\gamma_k^2\}_k$ is summable, whereas $\{\gamma_k\}_k$ is not. This is fulfilled, e.g., for $\gamma_k = \frac{1}{L_k}$. Further, we make the following standard assumption for interior point methods.

Assumption 1: The initial point is feasible, i.e., $x_1 \in \mathcal{C}$. The center of the constraint ellipsoid $x = v$ is always a feasible starting point; however, better warm starts might be possible.

For the following results, we impose a further requirement on the scaling parameter m.

Requirement 2 (Scaling for invariance of $\mathcal{C}$): The parameter m satisfies $m \geq m_{\text{inv}} := \max(\hat{m}_{\text{inv}}, 0) \geq m_{\min}$ with

$$\begin{aligned} \hat{m}_{\text{inv}} = \min\ & m \\ \text{s.t.}\ & \|\nabla f(x) + m\nabla g(x)\| \leq 2rL_1 \cos(\phi(x))\ \ \forall x \in \partial\mathcal{C}, \end{aligned}$$

$r = \frac{\sqrt{\underline{\sigma}(A)}}{\bar{\sigma}(A)}$, and $\cos(\phi(x)) = \frac{\langle \nabla f(x) + m\nabla g(x), \nabla g(x)\rangle}{\|\nabla f(x) + m\nabla g(x)\| \|\nabla g(x)\|}$ describing the angle between $\nabla f(x) + m\nabla g(x)$ and $\nabla g(x)$.
As for Requirement 1, finding the exact m_{inv} can be challenging, however, any upper bound $m \geq m_{\text{inv}}$ is valid.

Remark 1: In the scalar case $n = 1$, Requirement 1 and Requirement 2 are equivalent.

Then, we can show the following important property.

Proposition 2: Let Requirement 2 hold. Then, the set $\mathcal{C}$ is positively invariant under the gradient descent step (4) with the step size $0 < \gamma_k \leq \frac{1}{L_k}$, i.e., if $x_k \in \mathcal{C}$, then also $x_{k+1} \in \mathcal{C}$.

Proof: The proof is given in Appendix B. ■

This invariance property allows to stop the sequential gradient descent after any finite number of iterations N without violation of the constraints.

Now, we can show our main result, convergence of the sequential gradient descent to a minimizer of the original constrained problem.

Theorem 1: Let Requirement 2 hold. Let $\{x_k\}_k$ be the sequence of iterates resulting from sequential gradient descent (4) and $x^\star \in \mathcal{X}^\star$ a minimizer of the original Problem (1). Then,

$$f(x_k) \to f(x^\star) \text{ as } k \to \infty.$$

Proof: For the proof, we use ideas from [18], [19]. By the update law (4) and convexity of J_k, we have

$$\begin{aligned} \|x_{k+1} - x^\star\|^2 &= \|x_k - \gamma_k \nabla J_k(x_k) - x^\star\|^2 \\ &= \|x_k - x^\star\|^2 - 2\gamma_k \nabla J_k(x_k)^\top (x_k - x^\star) + \gamma_k^2 \|\nabla J_k(x_k)\|^2 \\ &\leq \|x_k - x^\star\|^2 - 2\gamma_k (J_k(x_k) - J_k(x^\star)) + \gamma_k^2 \|\nabla J_k(x_k)\|^2. \end{aligned}$$

By applying this inequality recursively, we obtain

$$\begin{aligned} 0 &\leq \|x_{k+1} - x^\star\|^2 \\ &\leq \|x_1 - x^\star\|^2 - 2\sum_{i=1}^{k} \gamma_i (J_i(x_i) - J_i(x^\star)) + \sum_{i=1}^{k} \gamma_i^2 \|\nabla J_i(x_i)\|^2, \end{aligned}$$

and hence,

$$\begin{aligned} & 2\sum_{i=1}^{k} \gamma_i (J_i(x_i) - J_i(x^\star)) \leq \|x_1 - x^\star\|^2 + \sum_{i=1}^{k} \gamma_i^2 \|\nabla J_i(x_i)\|^2 \\ \Leftrightarrow\ & 2\sum_{i=1}^{k} \gamma_i (J_i(x_i) - f(x^\star)) \leq \|x_1 - x^\star\|^2 + \sum_{i=1}^{k} \gamma_i^2 \|\nabla J_i(x_i)\|^2 \\ & \qquad + 2m\sum_{i=1}^{k} \gamma_i \frac{1}{i} g(x^\star)^i. \end{aligned}$$

Finally, since by Lemma 4 (in Appendix C) $J_k(x_k) \leq J_i(x_i)$ for all $i \leq k$,

$$\begin{aligned} & J_k(x_k) - f(x^\star) \leq \\ & \quad \frac{\|x_1 - x^\star\|^2 + \sum_{i=1}^{k} \gamma_i^2 \|\nabla J_i(x_i)\|^2 + 2m\sum_{i=1}^{k} \gamma_i \frac{1}{i} g(x^\star)^i}{2\sum_{i=1}^{k} \gamma_i}. \end{aligned} \tag{5}$$

From Lemma 5 (in Appendix C), we have that $\gamma_i^2 \|\nabla J_i(x_i)\|^2$ is summable. Further, we know that $\gamma_i \frac{1}{i} \leq \frac{1}{L_i}\frac{1}{i} = \frac{1}{\bar{\sigma}(Q + m(4i-2)A)i} \leq \frac{1}{m(4i-2)\bar{\sigma}(A)i} = \frac{1}{(4i-2)i}\frac{1}{m\bar{\sigma}(A)}$. Thus, $\gamma_i \frac{1}{i} g(x^\star)^i$ is summable. Since we required that the step size γ_i is not summable, the denominator of the right-hand-side of (5) diverges, whereas the nominator converges to a finite value. From this, it follows that the right-hand-side of (5) converges to zero as $k \to \infty$. Hence, $J_k(x_k) - f(x^\star) \to 0$ as $k \to \infty$. Since $J_k(x_k) \geq f(x_k) \geq f(x^\star)$, also $f(x_k) - f(x^\star) \to 0$ as $k \to \infty$. ■

A. Encrypted implementation

The sequential gradient descent steps of (4) only involve polynomial operations. Computing the parameters needed by the algorithm is more difficult. The required parameters are m and the functions' parameters Q, q, A, v, and possibly L_k. If the problem is known beforehand, the analysis for m and L_k can be done offline. In a private implementation without problem knowledge, an upper bound of the parameters m and L_k can be used. Then, any problem that has lower true parameters can be solved by the algorithm; however, it will be conservative and possibly slower.

An additional challenge that is common to all encrypted algorithms is the lack of ability to evaluate a stopping criterion. Typically, such stopping conditions involve a comparison of online obtained values to a threshold. Since this comparison is difficult to do for ciphertexts, the number of iterations N of the optimization algorithm has to be set beforehand.

Another aspect to be considered in encrypted implementations is the multiplicative depth of the algorithm and the cryptosystem, respectively. During every multiplication of encrypted numbers, the additive noise in the ciphertext that guarantees the security can be amplified. Therefore, leveled homomorphic cryptosystems only allow for a limited number of multiplications. The number of iterations N has to be chosen accordingly. Fully homomorphic cryptosystems such as [20], however, support an infinite number of operations at the cost of higher computational complexity of the involved bootstrapping operation. For an overview of bootstrapping in different cryptosystems, see [21], and for an analysis of bootstrapping in a dynamic control context, see [22].

V. Example: $\min(a,b)$

An important special case of the considered problem is finding the minimum of two encrypted numbers a and b, i.e.,

$$\begin{aligned} x^\star = \arg\min_{x} \;& x \\ \text{s.t. } & x \in [\min(a,b), \max(a,b)]. \end{aligned}$$

This is a well-known problem for encrypted computations since comparing encrypted numbers is a difficult task (cf. [23], [24]). We can recover the general problem formulation by setting $Q=0$, $q=1$, $A=\frac{4}{(a-b)^2}$, $v=\frac{a+b}{2}$ and choosing the functions $f(x)=x$ and $g(x)=\frac{4}{(a-b)^2}(x-\frac{a+b}{2})^2$.

The optimal slope ratio $m^\star = m_{\text{inv}} = m_{\min}$ is given by

$$m^\star = -\min\left(\frac{\nabla f(a)}{\nabla g(a)}, \frac{\nabla f(b)}{\nabla g(b)}\right) = \frac{|a-b|}{4}.$$

Let us assume that we know an upper bound m on $m^\star$ with $m=\alpha m^\star$ and $\alpha \geq 1$. Then, the auxiliary cost function is

$$J_k(x) = x + \alpha m^\star \frac{1}{k}\left(\frac{4}{(a-b)^2}\left(x-\frac{a+b}{2}\right)^2\right)^k.$$

The step size γ_k can be chosen as

$$\gamma_k = \frac{1}{L_k} = \frac{(a-b)^2}{4(4k-2)m} = \frac{1}{\alpha}\frac{|a-b|}{4k-2}.$$

Then, the gradient descent iterations for every $k \in \mathbb{N}$ are

$$\begin{aligned} x_{k+1} &= x_k - \frac{(a-b)^2}{4(4k-2)m} - 2A^{k-1}\left(x_k - \frac{a+b}{2}\right)^{2k-1} && (6) \\ &= x_k - \frac{1}{\alpha}\frac{|a-b|}{4k-2} - \frac{1}{2k-1}\left(\frac{4}{(a-b)^2}\right)^{k-1}\left(x_k-\frac{a+b}{2}\right)^{2k-1}, && (7) \end{aligned}$$

where (6) is in a form ready for implementation, and (7) will be used for analysis later.

For the encrypted implementation as in (6), we require encrypted values of a, b and A. We note that the availability of A is a strong assumption as it requires division of encrypted numbers. If these values are not available, encrypted division algorithms as in [25] can be used once before the iteration starts. Note that for the iteration with $k=1$, no A is needed. The constant m can be chosen large enough prior to knowing the specific problem. The specific problem just should satisfy $m^\star \leq m$, then it can be solved by the algorithm. Particularly, if we provide an algorithm with a value m, any problem with $|a-b| \leq 4m$ can be solved.

Remark 2: This bound on compatible problems can be understood similar to an approximation interval if we approximated the minimum function by polynomials in the first place. However, here, we can provide guarantees on invariance of the solution and convergence to the true minimum.

Note also that the minimizer $x_k^\star$ of the auxiliary problem would still converge to the true minimizer x_k even if $0 < m < m^\star$. Just the guarantees of the gradient descent do not hold any more but in many cases the iteration still converges.

A. Accuracy of the auxiliary solution

For this example, we can exactly calculate the minimizer $x_k^\star$ of the auxiliary problem depending on how conservative the choice of $m=\alpha m^\star$ is. Due to strict convexity and radially unboundedness of J_k, $x_k^\star$ is a minimizer if and only if $\nabla J_k(x_k^\star)=0$. For the analysis, we assume $a<b$ without loss of generality. Together with the ansatz $x_k^\star = \frac{a+b}{2} - \varepsilon_k$, we obtain

$$\begin{aligned} & 0 = 1 + 2\alpha\frac{|a-b|}{4}\left(\frac{4}{(a-b)^2}\right)^k(-\varepsilon_k)^{2k-1} \\ \Leftrightarrow \qquad & \varepsilon_k = \sqrt[2k-1]{\frac{1}{\alpha}}\,\frac{b-a}{2}. \end{aligned}$$

This means that the distance to the minimizer of the original problem can be expressed as

$$x^\star - x_k^\star = \frac{a-b}{2}\left(1-\sqrt[2k-1]{\frac{1}{\alpha}}\right),$$

which for any $\alpha > 0$ converges to zero as $k \to \infty$. With this result, we can even determine the number of iterations k for a desired precision $\delta < \frac{|a-b|}{2}$ as

$$|x^\star - x_k^\star| \leq \delta \qquad \Leftrightarrow \qquad k \geq \frac{1}{2} - \frac{\ln(\alpha)}{2\ln(1-\frac{2}{|a-b|}\delta)}.$$

B. Accuracy of a single gradient step

For the case that the constraint parameter A is not available, let us consider a gradient step for $k=1$, where only encrypted values of a and b are needed, since in

$$x_2 = \frac{a+b}{2} - \frac{(a-b)^2}{8m}$$

all nonpolynomial operations are done with public numbers. For $a<b$ this yields

$$\begin{aligned} x_2 &= \frac{a+b}{2} - \frac{1}{\alpha}\frac{|a-b|}{2} && (8) \\ &= \frac{a\left(1+\frac{\alpha-1}{2}\right)+b\left(\frac{\alpha-1}{2}\right)}{\alpha}, \end{aligned}$$

which is the exact minimum a for $\alpha=1$. Further, $x_2 \in [a, \frac{b+a}{2})$ for $\alpha \in [1,\infty)$. Note that for $\alpha=1$, (8) recovers the well-known formula $\min(a,b)=\frac{a+b}{2}-\frac{|a-b|}{2}$ (cf. [23], [24]). However, if we replace $|a-b|$ naïvely by the same upper bound $\frac{m}{4}=\alpha|a-b| \geq |a-b|$, we get

$$\begin{aligned} x &= \frac{a+b}{2} - \frac{m}{8} \\ &= \frac{1+\alpha}{2}a + \frac{1-\alpha}{2}b, \end{aligned}$$

which is also the exact minimum a for $\alpha=1$, but for $\alpha \in [1,\infty)$ takes values in $(-\infty, a]$. This might be less desirable than the invariance property of our proposed algorithm.

VI. Summary and Outlook

In this paper, we presented a novel optimization algorithm to solve a special class of QCQP. It is tailored to encrypted implementations as it explicitly only uses addition and multiplication, which are the natively supported operations of homomorphic cryptosystems. With this, we demonstrated how this class of constrained optimization problems can be solved in an encrypted fashion without the need of a trusted third party, a multi-party protocol or naïve polynomial approximations of standard optimization algorithms. For our proposed method, we showed several desirable properties, such as that the unique minimizers of the auxiliary unconstrained problems as well as the gradient descent iterates always stay inside the feasible set and converge towards a minimizer of the original constrained problem. Further, we showed how finding the minimum of two numbers can be formulated in our framework, and explicitly analyzed the relationship between accuracy, conservatism, and the number of iterations.

In future work, we plan to analyze the convergence speed for the general algorithm and further compare it with existing nonpolynomial barrier and penalty methods. It would also be interesting to improve the handling of encrypted parameters in the constraints and to extend the idea to more general classes of optimization problems.

Appendix

A. Proof of Lemma 3:

From the definition of $J_k(x)$, we get

$$\begin{aligned}
&\nabla^2 J_k(x) = \nabla^2 f(x) + \nabla^2 p_k(x)\\
&= Q + m((k-1)g(x)^{k-2}\nabla g(x)\nabla g(x)^\top + g(x)^{k-1}\nabla^2 g(x))\\
&\preceq Q + m((4k-4)A(x-v)(x-v)^\top A^\top + 2A),
\end{aligned}$$

where in the last inequality we used that the largest Hessian in the Loewner order is found at the boundary $\partial\mathcal{C}$, where $g(x)=1$. Now, we parameterize x on the boundary $\partial\mathcal{C}$ as $(x-v)=\sqrt{A^{-1}}y$ with $\|y\|=1$ and $A=\sqrt{A^{-1}}\sqrt{A^{-1}}$. Further, we observe that the singular values of $yy^\top$ fulfill $\sigma(yy^\top)=\{1,0,\dots,0\}$. This yields

$$\begin{aligned}
\nabla^2 J_k(x) &\preceq Q + m((4k-4)A\sqrt{A^{-1}}yy^\top \sqrt{A^{-1}}^\top A^\top + 2A)\\
&\preceq Q + m((4k-4)A\sqrt{A^{-1}}\bar{\sigma}(yy^\top)I\sqrt{A^{-1}}^\top A^\top + 2A)\\
&= Q + m(4k-2)A\\
&\preceq \bar{\sigma}(Q + m(4k-2)A)I.
\end{aligned}$$

■

B. Proof of Proposition 2:

The proof works in three steps. First, we show that under the gradient step (4), the image of a levelset of g, which is an ellipsoidal surface, is again an ellipsoidal surface. Second, we show that images that correspond to a lower level of g, are contained in ellipsoids that correspond to a level of $g(x)=1$. In the third step, we explicitly show that if $x_k \in \partial\mathcal{C}$, then $x_{k+1}\in\mathcal{C}$, which, according to the first part of the proof, bound all other levelsets inside the ellipsoid C, which is the 1-sublevelset of g.

For the current iterate $x_k\in\mathcal{C}$ with $g(x_k)=c\in[0,1]$, we define the ellipsoidal levelset $\partial\mathcal{C}_c=\{x\mid g(x)=c\}$ and the ellipsoidal levelset $\partial\mathcal{C}'_c=\{x\mid g'_c(x)=c\}$ with the same level c for a quadratic function $g'_c(x)=(x-v')^\top A'_c(x-v')$ with parameters v' and A'_c. The center v' of the ellipsoid $\partial\mathcal{C}'_c$ is given as $v'=v-\gamma_k(q+Qv)$. The matrix A'_c can be computed as $A'_c=T_c^{-1}AT_c^{-1}$ with the symmetric matrix $T_c=I-\gamma_k(Q+2mc^{k-1}A)$. Then,

$$g'_c(x_{k+1}) = (x_k-v)^\top TT^{-1}AT^{-1}T(x_k-v) = g(x_k) = c.$$

Thus, if $x_k\in\partial\mathcal{C}_c$, then $x_{k+1}\in\partial\mathcal{C}'_c$.

Now, we show that $\partial\mathcal{C}'_c=\{x\mid g'_c(x)=c\}$ is contained $\mathcal{C}'_1=\{x\mid g'_1(x)\le c\}$. Since $\mathcal{C}'_c$ and $\mathcal{C}'_1$ have the same center v', the condition $\partial\mathcal{C}'_c\subseteq\mathcal{C}'_1$ is satisfied if and only if

$$\frac{1}{c}A'_c \succeq A_1 \qquad \Leftrightarrow \qquad A \succeq \sqrt{c}T_cT_1^{-1}AT_1^{-1}T_c\sqrt{c}.$$

This holds if and only if $\bar{\sigma}(T_1^{-1}T_c\sqrt{c})\le 1$. Since $T_1^{-1}T_c\sqrt{c}\succeq 0$ and $T_1^{-1}\succ 0$, this is equivalent to

$$\begin{aligned}
& T_1^{-1}T_c\sqrt{c}-I \preceq 0\\
\Leftrightarrow\quad & (\sqrt{c}-1)(I-\gamma_k Q)+\gamma_k m2A(\sqrt{c}c^{k-1}-1)\preceq 0\\
\Leftarrow\quad & (\sqrt{c}-1)(I-\frac{1}{L_k}(Q-m2A)\preceq 0\\
\Leftarrow\quad & (\sqrt{c}-1)\big(I-\frac{1}{\bar{\sigma}(Q+m2(2k-1)A)}(Q-m2A)\big)\preceq 0,
\end{aligned}$$

which is satisfied for all $k\ge 1$.

Finally, we show that if $x_k\in\partial\mathcal{C}$, then $x_{k+1}\in\mathcal{C}$, which implies that $\partial\mathcal{C}'_1\subseteq\mathcal{C}$. Consider $x_k\in\partial\mathcal{C}$. If $\nabla J_k(x_k)=0$, then $x_{k+1}=x_k\in\mathcal{C}$. Now consider the case $\nabla J_k(x_k)\neq 0$. The cosine of the angle between $\nabla J_k(x_k)$ and $\nabla g(x_k)$ is given as

$$\cos(\phi(x_k)) = \frac{\langle\nabla J_k(x_k),\nabla g(x_k)\rangle}{\|\nabla J_k(x_k)\|\,\|\nabla g(x_k)\|}.$$

The radius of maximum curvature of the ellipsoid $\mathcal{C}$ is given by $r=\frac{\sqrt{\underline{\sigma}(A)}}{\bar{\sigma}(A)}$. For every point $x_k\in\partial\mathcal{C}$, a ball B_r with radius r can be placed such that $x_k\in\partial B_r$ and $B_r\subseteq\mathcal{C}$. Consider a line from x_k in the direction of $\nabla J_k(x_k)$, i.e., with angle ϕ from the normal $\nabla g(x_k)$ of the ellipsoid and the ball on x_k. Then, the length of the line inside B_r is given as $2r\cos(\phi(x_k))$. Thus, if the gradient step $\|x_{k+1}-x_k\|=\|-\gamma_k\nabla J_k(x_k)\|$ is not longer than the line length inside the ball, invariance is guaranteed. The condition is equivalent to

$$\begin{aligned}
& \|-\gamma_k\nabla J_k(x_k)\| \le 2r\cos(\phi(x_k))\\
\Leftarrow\quad & \|\nabla J_k(x_k)\| \le 2rL_1\cos(\phi(x_k)),
\end{aligned}$$

where we used that $\gamma_k\le\frac{1}{L_k}\le\frac{1}{L_1}\ \forall k\in\mathbb{N}$. Hence, by Requirement 2, the gradient descent step (4) leads to $x_{k+1}\in\mathcal{C}$ if $x_k\in\partial\mathcal{C}$. With this, we have shown $x_k\in\partial\mathcal{C}_c\subseteq\mathcal{C}\implies x_{k+1}\in\partial\mathcal{C}'_c\subseteq\mathcal{C}'_1\subseteq\mathcal{C}$.

■

C. Auxiliary lemmas

In this section, we provide some intermediate results that we need for the proof of Theorem 1.

Lemma 4: The sequence $\{J_k(x_k)\}_k$ is nonincreasing, particularly, $J_{k+1}(x_{k+1})\le J_k(x_{k+1})\le J_k(x_k)$.

Proof: The descent relation $J_k(x_{k+1}) \leq J_k(x_k)$, is a standard property of gradient descent with the chosen sequence of step sizes. From the construction of J_k, it follows that

$$\begin{aligned} J_k(x_{k+1}) &= J_{k+1}(x_{k+1}) - m\left(\frac{1}{k+1}g(x_{k+1}) - \frac{1}{k}\right)g(x_{k+1})^k \\ &\geq J_{k+1}(x_{k+1}) - m\left(\frac{1}{k+1} - \frac{1}{k}\right)g(x_{k+1})^k \\ &= J_{k+1}(x_{k+1}) + m\frac{1}{(k+1)k}g(x_{k+1})^k \\ &\geq J_{k+1}(x_{k+1}), \end{aligned}$$

where we used that $0 \leq g(x_{k+1}) \leq 1$. ■

Lemma 5: The sequence $\{\gamma_k^2 \|\nabla J_k(x_k)\|^2\}_k$ is summable, i.e., $\sum_{k=1}^{\infty} \gamma_k^2 \|\nabla J_k(x_k)\|^2 < \infty$.

Proof: From L_k-smoothness and the gradient step $x_{k+1} = x_k - \gamma_k \nabla J_k(x_k) \Leftrightarrow \nabla J_k(x_k) = \frac{x_k - x_{k+1}}{\gamma_k}$ it follows that

$$\begin{aligned} J_k(x_{k+1}) &\leq J_k(x_k) + \nabla J_k(x_k)^\top (x_{k+1} - x_k) + \frac{L_k}{2}\|x_{k+1} - x_k\|^2 \\ &= J_k(x_k) - \frac{1}{\gamma_k}\|x_{k+1} - x_k\|^2 + \frac{L_k}{2}\|x_{k+1} - x_k\|^2 \\ &= J_k(x_k) - \delta_k \|x_{k+1} - x_k\|^2 \end{aligned}$$

with $\delta_k = \left(\frac{1}{\gamma_k} - \frac{L_k}{2}\right) \geq \frac{L_k}{2}$. With $J_k(x_{k+1}) \geq J_{k+1}(x_{k+1})$ we get

$$\delta_k \|x_{k+1} - x_k\|^2 \leq J_k(x_k) - J_{k+1}(x_{k+1}).$$

If we sum from $k = 1$ to $N - 1$, we obtain

$$\begin{aligned} \sum_{k=1}^{N-1} \delta_k \|x_{k+1} - x_k\|^2 &\leq J_1(x_1) - J_N(x_N) \\ &\leq J_1(x_1) - f(x_N) < \infty. \end{aligned}$$

Thus, $\{\delta_k \|x_{k+1} - x_k\|^2\}_k$ is a summable sequence. Since $\delta_k \geq \frac{L_k}{2}$, and $\|x_{k+1} - x_k\| = \gamma_k \|\nabla J_k(x_k)\|$, also $\{\frac{1}{2}L_k \gamma_k^2 \|\nabla J_k(x_k)\|^2\}_k$ is summable, and also $\{L_k \gamma_k^2 \|\nabla J_k(x_k)\|^2\}_k$ is summable. Since $L_k \to \infty$ as $k \to \infty$, also $\{\gamma_k^2 \|\nabla J_k(x_k)\|^2\}_k$ is summable. ■

References

[1] A. Bertolace, K. Gatsis, and K. Margellos, “Homomorphically encrypted gradient descent algorithms for quadratic programming,” in *Proc. 62nd IEEE Conf. Decision and Control (CDC)*, 2023, pp. 3844–3849.

[2] P. Binfet, J. Adamek, N. Schlüter, and M. Schulze Darup, “Towards privacy-preserving cooperative control via encrypted distributed optimization,” *at - Automatisierungstechnik*, vol. 71, no. 9, pp. 736–747, 2023.

[3] Y. Shoukry, K. Gatsis, A. Alanwar, G. J. Pappas, S. A. Seshia, M. Srivastava, and P. Tabuada, “Privacy-aware quadratic optimization using partially homomorphic encryption,” in *Proc. 55th IEEE Conf. Decision and Control (CDC)*, 2016, pp. 5053–5058.

[4] A. B. Alexandru, K. Gatsis, and G. J. Pappas, “Privacy preserving cloud-based quadratic optimization,” in *Proc. 55th Annual IEEE Allerton Conf. Communication, Control, and Computing*, 2017, pp. 1168–1175.

[5] A. B. Alexandru, K. Gatsis, Y. Shoukry, S. A. Seshia, P. Tabuada, and G. J. Pappas, “Cloud-based quadratic optimization with partially homomorphic encryption,” *IEEE Trans. Autom. Control*, pp. 2357–2364, 2020.

[6] A. B. Alexandru, M. Morari, and G. J. Pappas, “Cloud-based MPC with encrypted data,” in *Proc. 57th IEEE Conf. Decision and Control (CDC)*, 2018, pp. 5014–5019.

[7] I. Damgard, M. Geisler, and M. Kroigard, “Homomorphic encryption and secure comparison,” *Int. J. Applied Cryptography*, vol. 1, no. 1, pp. 22–31, 2008.

[8] M. Schulze Darup, A. Redder, and D. E. Quevedo, “Encrypted cloud-based MPC for linear systems with input constraints,” *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 535–542, 2018.

[9] M. Schulze Darup, “Encrypted MPC based on ADMM real-time iterations,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 3508–3514, 2020.

[10] M. Schulze Darup, G. Book, and P. Giselsson, “Towards real-time ADMM for linear MPC,” in *Proc. 18th European Control Conf. (ECC)*. IEEE, 2019, pp. 4276–4282.

[11] M. Schulze Darup, “Encrypted model predictive control in the cloud,” in *Privacy in Dynamical Systems*. Springer Singapore, 2019, pp. 231–265.

[12] M. Schulze Darup, A. Redder, I. Shames, F. Farokhi, and D. Quevedo, “Towards encrypted MPC for linear constrained systems,” *IEEE Control Systems Letters*, vol. 2, no. 2, pp. 195–200, 2018.

[13] N. Schlüter and M. Schulze Darup, “Encrypted explicit MPC based on two-party computation and convex controller decomposition,” in *Proc. 59th IEEE Conf. Decision and Control (CDC)*, 2020, pp. 5469–5476.

[14] K. Tjell, N. Schlüter, P. Binfet, and M. Schulze Darup, “Secure learning-based MPC via garbled circuit,” in *Proc. 60th IEEE Conf. Decision and Control (CDC)*, 2021, pp. 4907–4914.

[15] Y. Nesterov, *Introductory Lectures on Convex Optimization*. Springer New York, 2004.

[16] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming*. SIAM, 1990.

[17] A. Forsgren, P. E. Gill, and M. H. Wright, “Interior methods for nonlinear optimization,” *SIAM Review*, vol. 44, no. 4, pp. 525–597, 2002.

[18] S. Boyd and J. Park, “Subgradient methods,” Lecture notes for EE364b, Stanford University, 2014, https://web.stanford.edu/class/ee364b/lectures/subgrad_method_notes.pdf [Online; accessed 28-January-2025].

[19] B. T. Poljak, *Introduction to optimization*, ser. Translations Series in mathematics and engineering. New York: Optimization Software, 1987.

[20] J. H. Cheon, A. Kim, M. Kim, and Y. Song, “Homomorphic Encryption for Arithmetic of Approximate Numbers,” in *Advances in Cryptology – ASIACRYPT 2017*, 2017, pp. 409–437.

[21] A. A. Badawi and Y. Polyakov, “Demystifying bootstrapping in fully homomorphic encryption,” Cryptol. ePrint Arch., Paper 2023/149, 2023.

[22] S. Schlor and F. Allgöwer, “Bootstrapping guarantees: Stability and performance analysis for dynamic encrypted control,” *IEEE Control Systems Letters*, vol. 8, pp. 2235–2240, 2024.

[23] J. H. Cheon, D. Kim, D. Kim, H. H. Lee, and K. Lee, “Numerical method for comparison on homomorphically encrypted numbers,” in *Advances in Cryptology – ASIACRYPT 2019*, 2019, pp. 415–445.

[24] J. H. Cheon, D. Kim, and D. Kim, “Efficient Homomorphic Comparison Methods with Optimal Complexity,” in *Advances in Cryptology – ASIACRYPT 2020*, 2020, pp. 221–256.

[25] J. Adamek, P. Binfet, N. Schlüter, and M. Schulze Darup, “Encrypted system identification as-a-service via reliable encrypted matrix inversion,” in *Proc. 63rd IEEE Conf. Decision and Control (CDC)*, 2024, pp. 4582–4588.