Quantum Information & Computation
ISSN 1533-7146 Volume 25 (2025) pp.369–384

Cerebration Science Publishing

Article

Experimental Factoring Integers Using Fixed-Point-QAOA with a Trapped-Ion Quantum Processor

Ilia V. Zalivako [1,2], Andrey Yu. Chernyavskiy [2], Anastasiia S. Nikolaeva [1,2,3,*], Alexander S. Borisenko [1,2], Nikita V. Semenin [1,2], Kristina P. Galstyan [1,2], Andrey E. Korolkov [1,2], Sergey V. Grebnev [2], Evgeniy O. Kiktenko [2,3], Ksenia Yu. Khabarova [1,2], Aleksey K. Fedorov [1,2,3], Ilya A. Semerikov [1,2], and Nikolay N. Kolachevsky [1,2]

[1] P.N. Lebedev Physical Institute of the Russian Academy of Sciences, Moscow 119991, Russia
[2] Russian Quantum Center, Skolkovo, Moscow 121205, Russia
[3] Laboratory of Quantum Information Technologies, National University of Science and Technology "MISIS", Moscow 119049, Russia
* Corresponding author: anastasiia.nikolaeva21@gmail.com

Received date: 28 March 2025; Accepted date: 29 July 2025; Published online: 5 September 2025

Abstract: Factoring integers is considered as a computationally hard problem for classical methods, whereas there exists polynomial-time Shor's quantum algorithm for solving this task. However, requirements for running Shor's algorithm for realistic tasks, which are beyond the capabilities of existing and upcoming generations of quantum computing devices, motivate to search for alternative approaches. In this work, we experimentally demonstrate factoring of the integer with a trapped ion quantum processor using the Schnorr approach and a modified version of the quantum approximate optimization algorithm (QAOA). The key difference of our approach in comparison with the recently proposed QAOA-based factoring method is the use of the fixed-point feature, which relies on the use of universal parameters. We present experimental results on factoring $1591 = 37 \times 43$ using 6 qubits as well as simulation results for $74425657 = 9521 \times 7817$ with 10 qubits and $35183361263263 = 4194191 \times 8388593$ with 15 qubits. Although we present all the necessary details for reproducing our results and analysis of the performance of the factoring method, the scalability of this approach in both the classical and quantum domains still requires further studies.

Keywords: Schnorr's algorithm; fixed-point QAOA; trapped-ion processor

1. Introduction

Shor's algorithm [1,2] for factoring integers has become one of the examples of a practically relevant problem, which is hard for classical computers yet amenable for quantum processors. The implication of the integer factorization problem to the widely adopted cryptographic schemes, such as the RSA cryptosystem [3], is a clear motivation for studying its practical complexity within both the classical and quantum approaches [4,5]. Proof-of-concept experimental factoring of 15, 21, and 35 has been demonstrated on superconducting [6], trapped ion [7], and photonic [8–10] quantum computers. However, the implementation of Shor's algorithm for breaking of actually employed cryptosystems requires resources, which seem to be far beyond the capabilities of existing and upcoming generations of quantum computing devices. For example, in order to factor a 2048-bit RSA integer (i.e., an integer $N = pq$ where p, q are distinct primes), one would need 8 hours using 20 million noisy qubits [11]. Recently, this requirement has been updated to less than a week using less than a million noisy qubits [12].

While current quantum computers are not yet capable of running Shor's algorithm in a regime that provides a quantum advantage, various approaches are being developed to implement factorization with fewer resources [13–15] or even with currently available noisy intermediate-scale quantum (NISQ) devices [16–18]. One of the proposed methods is the use of the variational quantum factorization (VQF), which has been applied to factorize 41-, 12-, and 13-bit integers on three, four, and five superconducting qubit systems, respectively [19]. The approach incorporates classical preprocessing that yields variable quantum resource requirements across different integers, with some cases

DOI: 10.2478/qic-2025-0021

achieving particularly significant qubit reductions. The recent paper [20] utilizes an IBM superconducting processor to factorize an 8-bit integer (253) using 9 qubits, without the need for classical preprocessing. This is achieved by applying the variational quantum eigensolver (VQE), which minimizes $(N - pq)^2$ to obtain the desired factorization. Additionally, [20] also presents a simulation of the factorization of up to 20-bit integers using up to 27 qubits.

Another approach, proposed in [21], combines Schnorr's factorization method [22] with the quantum approximate optimization algorithm (QAOA) [23,24], which can be efficiently implemented on NISQ devices [25–28]. The seminal work [21] demonstrated the factorization of a 48-bit integer on 10 trapped ion qubits, describing the solution to a single subproblem using a quantum-classical hybrid quantum QAOA approach. However, as it has been shown [29,30] such an approach encounters a number of pitfalls coming from both classical and quantum domains. In particular, it has been observed [29] that demonstrating this subproblem solution does not equate to full factorization. Building on this, [31] investigated factorizing the same 43-bit integer using a digitized-counterdiabatic quantum algorithm. Their results demonstrated an improved success probability for one Hamiltonian identified in [21], though full factorization was not achieved.

In this study, we show that certain challenges associated with the QAOA-based factorization process can be addressed by switching to a fixed-point variant of QAOA (fpQAOA) [32]. While it became a routine to run QAOA with classical optimization of expectation values with respect to the parameters, such an approach suffers from the problem of global optimization and statistical fluctuations. To our knowledge, the alternative idea to exploit so-called universal angles (parameters) in the QAOA has been presented for the first time in [33]. We follow the latter approach so that in our fixed-point version of QAOA [32] we use fixed optimal parameters from the corresponding training set of tasks, normalize it (i.e., Hamiltonians), and then search for angles providing the maximum minimal increase in the probability of a correct answer, whereas the Max-Min problem is solved via evolution optimization. This allows us to solve reliably the closest vector problem (CVP), which lies in the basis of Schnorr's algorithm, with the use of the quantum device. Within this approach, we demonstrate experimental factoring of the number $1591 = 37 \times 43$ using 6 qubits with a trapped ion quantum processor; see Table 1. We also present simulation results for $74425657 = 9521 \times 7817$ and $35183361263263 = 4194191 \times 8388593$ with 10 and 15 qubits, correspondingly. These results demonstrate an improvement in the required resources, compared to the simulation results presented in [20], which factored a 20-bit integer using 27 qubits. Although we expect that one of the difficulties in the realization of the QAOA-based factoring is resolved, this approach still requires further scalability studies.

Table 1. Comparison of the main results of the NISQ factoring in the current work with previous studies.

Method	N	Qubits	Integer-specific	Full factoring
Schnorr + QAOA [21]	48-bit	10 trapped-ion	no	no[1]
Schnorr + DCQA [31]	48-bit	10 trapped-ion	no	no[1]
VQF [19]	41-bit	3 superconducting	yes[2]	yes
VQE [20]	8-bit	9 superconducting	no	yes
Schnorr + fpQAOA (current work)	11-bit	6 trapped-ion	no	yes

[1] Only solves a single QUBO subproblem; no full factoring shown. [2] Classical preprocessing efficiency varies by integer, causing significant quantum resource variance.

2. Fixed-Point QAOA-Based Factoring

Here we describe the main principles of the considered algorithm, Figure 1 shows the scheme of factoring, and the detailed description is presented in the Supplementary Materials. The crucial component of Schnorr's factoring algorithm is the search for smooth relation pairs of integers, so-called sr-pairs. The definition of sr-pair is based on the *factor base* P_n: the set $\{p_i\}_{i=0,\ldots,n}$ of the first n primes together with -1 ($p_0 = -1,\ p_1 = 2,\ p_2 = 3,\ldots$). A pair of integers (u, v) is called an sr-pair (for fixed P_n and an integer N) if u and $u - vN$ have prime factors only from P_n. Note that the size of the factor base n is the most important hyperparameter of the algorithm (depending on the size of the factorized number N) and is also equal to the number of qubits of the quantum part.

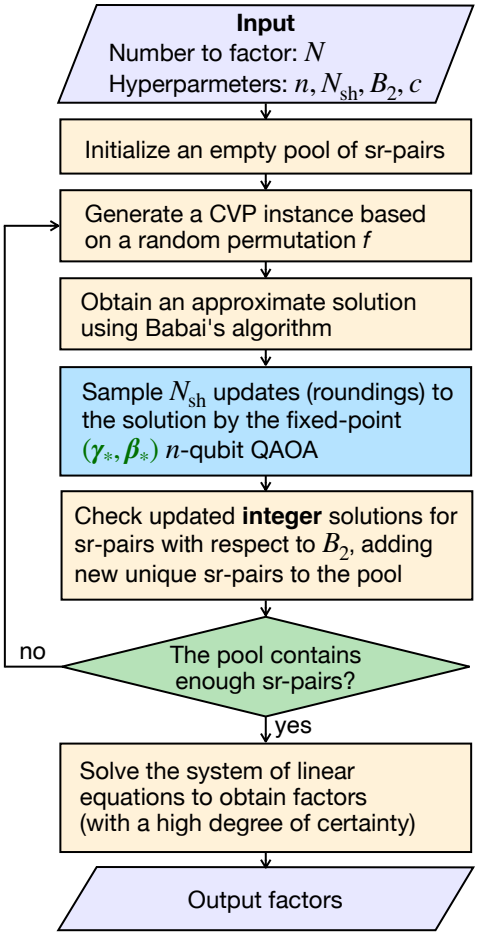


Figure 1. General scheme of the QAOA-based Schnorr factoring algorithm.

As soon as we have a sufficient number of such pairs, which is larger than the size of the factoring base (hyperparameter of the algorithm), we can form a system of linear equations that appears to be degenerate and always has a solution. This solution, by a classical Fermat's method (see, e.g., [34]), provides a factorization with a high probability.

The problem of sr-pairs search can be reduced to the closest vector problem on a lattice: one needs to find the linear combination of vectors

$$\begin{pmatrix} \mathbf{b}_1 & \dots & \mathbf{b}_n \end{pmatrix} := \begin{pmatrix} \lceil f(1)/2 \rfloor & 0 & \dots & 0 \\ 0 & \lceil f(2)/2 \rfloor & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lceil f(n)/2 \rfloor \\ \lceil 10^c \ln p_1 \rfloor & \lceil 10^c \ln p_2 \rfloor & \dots & \lceil 10^c \ln p_n \rfloor \end{pmatrix} \tag{1}$$

with integer coefficients closest to the target vector

$$\mathbf{t} = \begin{pmatrix} 0 & \dots & 0 & \lceil 10^c \ln N \rfloor \end{pmatrix}^\top \tag{2}$$

Here $f(i)$ is the random permutation of the first n natural numbers, c is the smoothness hyperparameter, and N is the integer to factorize. The closer the found solution to the desired vector, the greater the chance of obtaining an sr-pair.

Schnorr's method relies on solving the CVP with the classic approximate Babai's algorithm [35] with LLL-reduction (Lenstra–Lenstra–Lovász) [36]. As this algorithm gives only an approximate real-valued solution, in the original paper by Schnorr it was rounded to the closest integer value at the last step. The idea behind the recent proposal [21] is to choose the rounding side for each variable to find the closest integer-valued solution, which in turn

directly reduces to a quadratic unconstrained binary optimization (QUBO) problem. The example of the rounding refinement is presented in Figure 2. Such class of problems is amenable to solving with QAOA.

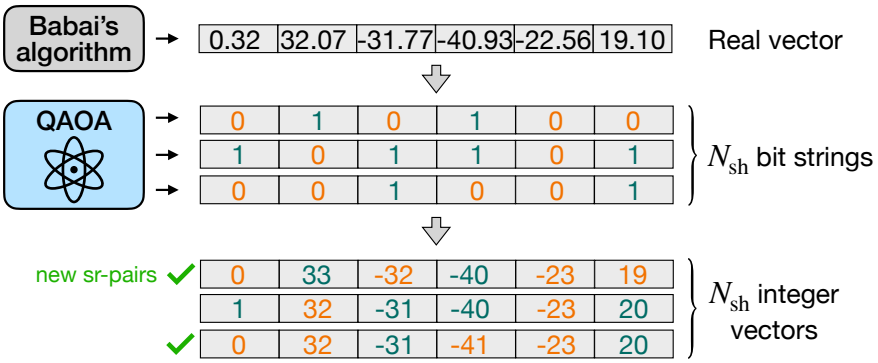


Figure 2. The example of the rounding refinement in the QAOA-based Schnorr factoring algorithm.

QAOA is based on the trotterization of the adiabatic evolution of the following form:

$$|\beta,\gamma\rangle = U(\beta[p],\gamma[p])\dots U(\beta[1],\gamma[1])|+\rangle^{\otimes n},$$
$$U(\beta[j],\gamma[j]) = e^{-i\beta[j]H_M}e^{-i\gamma[j]H_P} \tag{3}$$

where $\beta = \{\beta[j]\}$ and $\gamma = \{\gamma[j]\}$ are circuit parameters (angles), hyperparameter p is the number of layers, $|+\rangle$ is the $+1$ eigenstate of σ_x Pauli matrix, $H_M = \sum_k \sigma_x^{(k)}$ is the mixing Hamiltonian (here $\sigma_x^{(k)}$ is σ_x acting on k-th qubit), and H_P is the problem Hamiltonian, which in most cases directly encodes the Ising form of a QUBO problem to be solved.

The most common approach to QAOA is to classically optimize the expectation value $E(\beta,\gamma) = \langle\beta,\gamma|H_P|\beta,\gamma\rangle$ being estimated by the set of measurements (shots) on a quantum processor (see, e.g., Refs. [37–39]). In contrast, in the seminal QAOA paper [23] relies on searching optimal angles utilizing the efficient exact classical calculation of E (which was presented for Max-Cut problems on 3-regular graphs [23]) followed by sampling on a quantum processor. We have used an alternative approach based on the empirical hypothesis of close optimal angles for different instances of the same problem type [33,40,41].

To find fixed QAOA parameters, we use the training set consisting of 100 QUBO subproblems arised during factoring $N = 48567227$ on $n = 10$ qubits. As optimal QAOA problem angles γ scale together with QUBO coefficients, we normalize every QUBO coefficient matrix by its maximal value [32]. The ratio P_q/P_c of the probability P_q to measure the optimal (minimal) answer to its classical random sampling counterpart P_c was used as an optimization metric, and its minimum over the training set was maximized using random mutations optimization algorithm [42,43]. To minimize the quantum circuit depth, we use just a single layer of QAOA ($p = 1$), which significantly increases robustness of the quantum part of the algorithm. The quantum circuit for a single layer of QAOA used in the algorithm has the form presented in Figure 3, and Figure 4 demonstrates the scheme of the fpQAOA.

The resulting single-layer QAOA parameters used in the factorization are $\gamma_1 := \gamma_* = 2.64$ and $\beta_1 := \beta_* = 0.33$. The fixed-parameters approach allows avoiding the classical-quantum hybrid optimization procedure and fits well with the demands of Schnorr's method: one does not need to obtain the exact or suboptimal solution of CVP, but sample solutions close to the target vector to increase the probability of forming a set of sr-pairs.

In the classical part of the algorithm, we directly follow Refs. [21] and [29]. We use the main factor base of the size $B_1 = 6$ (which is equal to the number of qubits), the relaxed factor base size for sr-pairs verification is $B_2 = 11$, the rounding parameter of lattice/target formation procedure is $c = 1.5$, and the parameter of LLL-reduction is $\delta = 0.75$. For each lattice (which is formed by a random permutation of the diagonal), we conduct 5 measurements (shots) of each circuit. Due to a strongly stochastic nature of the algorithm the required number of circuits varies. The details of a single run of the factorization algorithm including the exact form of the circuit and corresponding parameters are provided in the Supplemental Material.

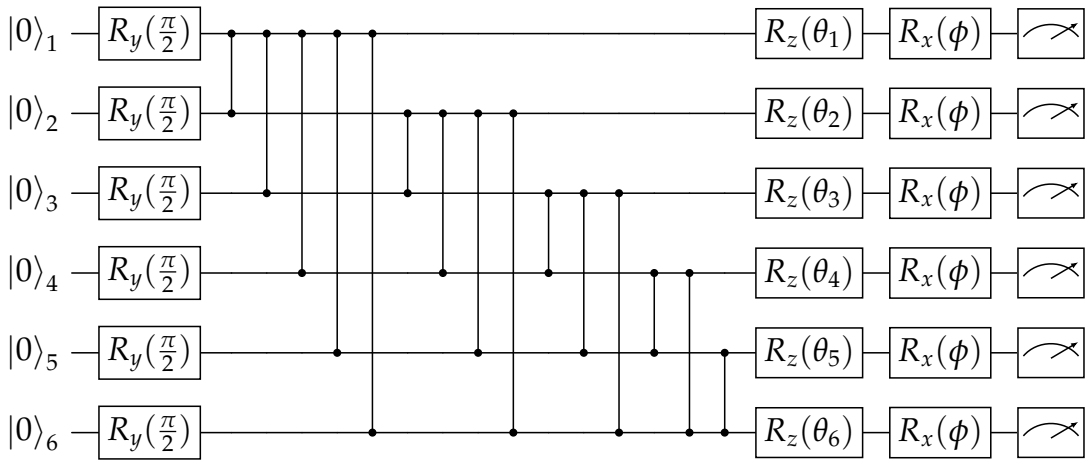


Figure 3. Architecture of the executed quantum circuits in fixed-point-QAOA algorithm. Each pair of connected black circles corresponds to $ZZ(\chi_{ij})$ gate acting on i-th and j-th qubits, where for each involved qubit pair χ_{ij} is unique. Angles θ_i in $R_z(\theta_i)$ gates are also different for each i-th qubit in each circuit. β in $R_x(\phi)$ is equal to 2.64. For each of the 9 executed circuits parameters of these gates are given in Table 5 of Supplementary Materials.

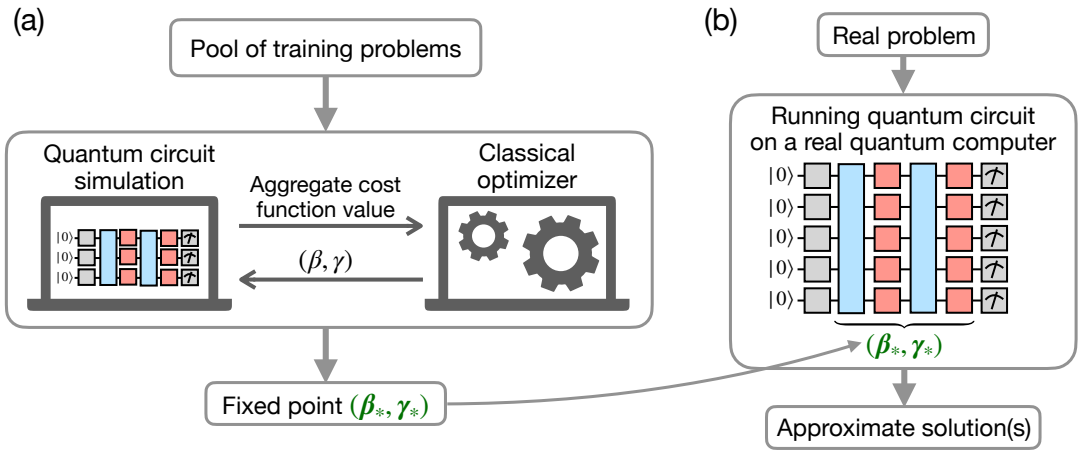


Figure 4. Scheme of the fixed-parameters QAOA algorithm: (**a**) training—search for fixed parameters; (**b**) using for real problems.

3. Experimental Setup

Experimental demonstration of the algorithm was performed with a quantum processor based on a chain of ten ultracold $^{171}Yb^+$ ions in a linear Paul trap. Its characteristics are summarized in Table 2. Qubits are encoded in states $|0\rangle = {}^2S_{1/2}(F = 0, m_F = 0)$ and $|1\rangle = {}^2D_{3/2}(F = 2, m_F = 0)$, coupled by an optical E2 transition at wavelength $\lambda = 435.5$ nm. While the setup supports usage of all five Zeeman sublevels of the upper state for the information encoding (i.e., we have the qudit processor [44]); in this work we have used the processor in the qubit regime.

Before each experimental shot ions are Doppler cooled to the temperatures of approximately 1.5 mK, which is followed by the sideband-cooling of all radial motional modes close to the ground state and initialization to the $|0\rangle$ state by optical pumping [44]. On the next stage the target native gates sequence is being implemented. In our system single-qubit native gates are $R_\phi(\theta) = \exp\left(-i\sigma_\phi\theta/2\right)$ and $R_z(\theta) = \exp(i\theta|1\rangle\langle 1|)$, where $\sigma_\phi = \cos\phi\sigma_x + \sin\phi\sigma_y$, and ϕ, θ — arbitrary angles. The first operation is performed by applying a laser pulse, resonant to the $|0\rangle \rightarrow |1\rangle$ transition. In this case ϕ is determined by the relative phase of the laser field and the qubit, while θ is determined by the pulse duration. The $R_z(\theta)$ is a virtual gate [45] and is performed by shifting phases of all successive laser pulses applied to this ion. A native two-qubit operation for this system is a Mølmer-Sørensen gate [46–49] $R_{xx}(2\chi) \equiv XX(\chi) = \exp(-i\chi\sigma_x \otimes \sigma_x)$. This gate is implemented by illuminating a target pair of ions with a bichromatic laser fields, coupling their electronic states with a collective motional degrees of freedom (in our

case we use radial motional modes). These common motional modes serve as mediator, coupling both qubits. The laser fields are amplitude-modulated to decouple all electronic degrees of freedom from motional ones at the end of the gate and reduce sensitivity of the operation to the experimental parameters [50]. The processor supports $XX(\chi)$ gates with arbitrary χ and all-to-all connectivity. We also include $R_{zz}(2\chi) \equiv ZZ(\chi) = \exp(-i\chi\sigma_z \otimes \sigma_z)$ gate in the list of supported operations, which is automatically hardware-efficiently transpiled as $ZZ(\chi) = (R_y(\pi/2) \otimes R_y(\pi/2))XX(\chi)(R_y(-\pi/2) \otimes R_y(-\pi/2))$ in the processor. At the end of each experimental shot the quantum register readout is performed using electron-shelving technique on the $|^2S_{1/2}\rangle \rightarrow |^2P_{1/2}\rangle$ transition at 369 nm [44, 51]. Ions fluorescence in this process is collected with a high numeric aperture lens and is sent via an array of multimode fibers to the multichannel photomultiplier tube.

Fidelities of the single-qubit and two-qubit operations are 99.946(6)% and 96.3(3)%, which are measured using randomized benchmarking [52], and parity oscillations observation [53], correspondingly. The qubits coherence time $T_2^* = 30(2)$ ms was extracted from decay of Ramsey fringes contrast with increasing delay between $\pi/2$ pulses. To reduce cross-talk during single-qubit operations all $R_\phi(\theta)$ gates in the circuits are substituted with their composite analogues using SK1 scheme [54]. Particularly, two 2π rotations around specific axes are added after each single-qubit gate, which are known to suppress both cross-talks and rotation angle fluctuations.

Table 2. Parameters of the experimental setup.

Parameter	Value
Number of qubits	10
Single-qubit gate fidelity	99.946(6)%
Two-qubit gate fidelity[1]	96.3(3)%
T_2^*	30(2) ms
Connectivity	Full
Single-qubit gate duration	20 μs
Two-qubit gate duration	1.14 ms
Secular frequencies $(\omega_x, \omega_y, \omega_z)$	$2\pi \times (3.7, 3.6, 0.13)$ MHz

[1] SPAM-corrected Bell-state preparation fidelity averaged over all qubit pairs.

More details on the experimental setup can be found in [44,55].

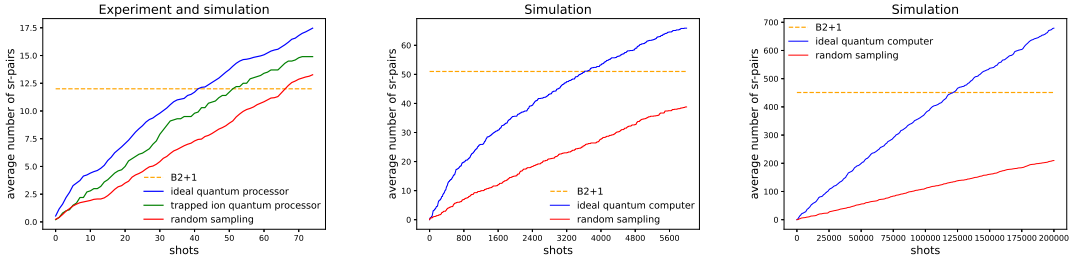
4. Experimental Results

In the experiment, we use Schnorr's approach assisted with the fixed-angles QAOA to factorize number $1591 = 37 \times 43$ using 6 qubits.

In a single sample run of the experiment (for details, see Supplemental Material), $B_2 + 1$ sr-pairs required to deterministically factorize the number were found in 43 steps (shots) using 9 different quantum circuits (each circuit repeated 5 times followed by the next circuit). However, in this particular sample run the first 39 shots appeared to be already sufficient to factorize the number. We have compared the average speed of collecting unique sr-pairs in three cases: (i) random sampling; (ii) experimentally obtained samples; (iii) samples obtained with noiseless emulator (see Figure 5(a)). The figure demonstrates the advantage of the quantum processor sampling results over the random sampling. However, the presence of the noise in the system decreases the efficiency of the method in comparison with a noiseless emulator. To illustrate the level of the noise in the quantum processor we also measured the output states probability distributions for several used circuits with better averaging and compared it with results expected in the absence of errors (for details, see Supplemental Material).

In this experiment we chose to use 6 qubits as a trade-off between the problem size and quantum circuits fidelity. Numeric simulations show that the expected advantage over random sampling in QUBO-subproblems increases with the growth of qubits number and magnitude of a number to factorize (e.g., see Figure 5). At the same time as the number of two-qubit operations in each circuit is equal to $n(n-1)/2$, where n is the number of qubits, the quantum sampling fidelity decreases with larger n. In our experiments $n = 6$ was the smallest number of qubits, where the advantage over random sampling was observed experimentally despite the better sampling fidelity at $n < 6$.

A number of shots per circuit was chosen using numerical simulations. It was set to be sufficient to find enough sr-pairs, keeping the total number of shots minimal.



(a) $N = 43 \times 37$, $n = 6$, $B_2 = 11$, $c = 1.5, N_{\text{sh}} = 5$

(b) $N = 9521 \times 7817$, $n = 10$, $B_2 = 50$, $c = 4$, $N_{\text{sh}} = 20$

(c) $N = 4194191 \times 8388593$, $n = 15, B_2 = 450, c = 4$, $N_{\text{sh}} = 1000$

Figure 5. A comparison of sr-pairs collection rates between cases where QUBO-subproblems samples are generated with a random sampling (red lines), a noiseless quantum emulator (blue), and a real trapped-ion quantum processor (green) for different number of qubits. The left sub-figure shows both experimental (averaged over 10 runs) and simulation data (averaged over 30 runs), while other figures contain only simulation results (averaged over 10 trajectories). Here N stands for the factorized number, n is for the number of qubits, and N_{sh} is for the number of shots per circuit. The dashed horizontal line shows a $B_2 + 1$ sr-pairs threshold which guarantees the factorization.

5. Scalability Analysis

The initial complexity estimates presented in Schnorr's work [22] did not lead to practical results for factoring large numbers; however, the effectiveness of the method has still neither been proven nor strictly disproved. Based on Refs. [29,30,56,57] and own numerical experiments, the following difficulty can be noted: the probability that estimates obtaining an sr-pair by suboptimal solutions of CVP problem (obtained by classical or quantum methods) does not directly lead to the probability of obtaining a set of *unique* sr-pairs needed for the factorization. Further, we present a brief empirical analysis of scalability of the presented method. We use the number of measurement shots as a proxy for computational cost, as we adopt an exponential complexity model that ignores polynomial factors from quantum circuit depth (though this can be extended with circuit-level details). Figure 6 shows the logarithmic shot count versus the bit-length n_b of the factored integers.

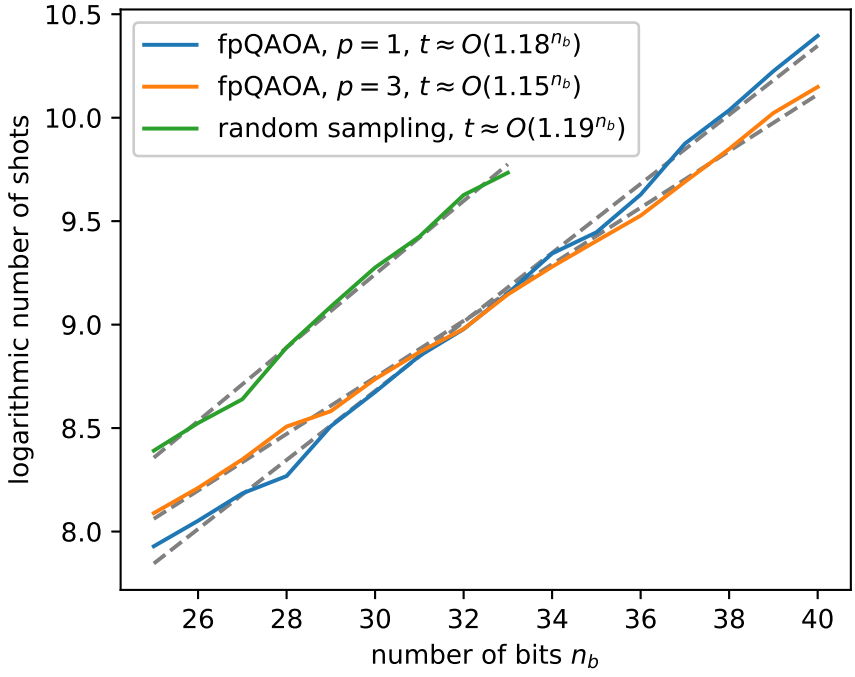


Figure 6. Dependency of computational complexity (number of shots) on the number of bits.

For fpQAOA-based factoring, we trained on 50 QUBO instances generated from random 24-bit integers using $n = 10$ (where n denotes the lattice size and the number of qubits). Since optimal parameter selection for n remains an open problem, we performed a grid search to determine the best n for each bit-length and algorithm variant; optimal values of n are presented in Table 3.

Table 3. Optimal n for different bit-lengths and QUBO-solving methods.

method / n_b	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
fpQAOA, $p = 1$	11	10	11	11	11	12	12	11	12	12	12	13	12	14	12	13
fpQAOA, $p = 3$	12	14	12	14	15	14	14	14	14	14	15	15	15	15	16	16
random sampling	9	9	9	9	9	10	10	10	10							

Numerical experiments on 25–40-bit integers reveal approximately exponential scaling within the tested range (Figure 6)—consistent with other NISQ-era quantum heuristics [58,59]. Results indicate that the shot count scales roughly as $O(1.15^{n_b})$ for $p = 3$ QAOA, even for such a low depth outperforming random sampling ($m \approx 1.19$) and classical brute-force ($m \approx 1.414$). While this suggests a quantum advantage for mid-scale factoring, we emphasize that this model is provisional—its validity at cryptographic scales (e.g., $n_b > 100$) remains untested (the true complexity could be strictly better or worse than this exponential estimate depending on the interaction between Schnorr's lattice reduction and quantum optimization dynamics). We also want to note that for exponential scaling $O(m^{n_b})$, values like $m \approx 1.02$ (enabling classical record 829-bit factoring in 10^7 shots) may yield practical utility despite asymptotic limitations, as the base m dominates at intermediate scales.

The improvement of complexity of the presented method can possibly be achieved by hyperparameter optimization, increasing the depth p, dynamic p-scaling via Fourier encoding [38], and the enlargement of search spaces beyond single steps from classical approximations. We also emphasize that the quantum advantage should be verified against classical QUBO-solving methods more sophisticated than random sampling.

6. Conclusion and Outlook

We have considered the Schnorr factoring scheme, where following the idea from [21], we adopt the QAOA method at the last step of Babai's algorithm. However, for the first time we used a fixed-point feature of QAOA [32,33] for the factoring problem and were able to factor a specific integer. To the best of our knowledge, it is the first successful experimental factoring of a particular integer with fixed-point QAOA-assisted Schnorr approach, whereas previously it was only experimentally presented how to obtain some sr-pairs for this task using quantum computers.

To confirm both the overall scheme and the fixed-point approach, we experimentally factor $1591 = 37 \times 43$ using 6 qubits of the 10-qubit trapped-ion processor. Thus, this work provides the first complete implementation of quantum-enhanced Schnorr's lattice-based factorization, filling a critical development gap among alternative quantum factoring methods. The fixed-angle QAOA approach enables scalable factoring of general integers, with simulations up to 45 bits on 15 qubits suggesting practical utility for near-term NISQ experiments and benchmarks.

For further research we leave the questions of the algorithm's efficiency and thorough comparison with classical methods, as well as a more detailed investigation of the quantum processor noise influence.

Note added. After completion of this work, we became aware of [59], which also suggests using fixed-point QAOA in the same context. Authors have presented an alternative approach of fixed angles search and scaling, and conducted a thorough numerical analysis of QAOA-augmented refinement of CVP problem. In contrast, in our work we consider the complete factorization algorithm and its experimental trapped-ion implementation.

Author Contributions

A.Yu.Ch. implemented the fpQAOA subroutine. S.V.G. implemented classical subparts of the algorithm. A.S.N. adapted quantum part of the algorithm for execution on the trapped-ion processor. I.V.Z and A.S.B. performed the experiment. I.V.Z., A.S.B., I.A.S., K.P.G., N.V.S., and A.E.K. built, optimized and operated the quantum processor. E.O.K., A.S.N., and I.V.Z, conducted a theoretical analysis of the experimental results. E.O.K. created the visual illustrations for the article. K.Yu.Kh., A.K.F., I.A.S., and N.N.K. supervised the project. *Writing—original draft*, all authors have contributed equally to this task. *Writing—review and editing*, all authors have contributed equally to this task. All authors have read and agreed to the published version of the manuscript.

Funding
A.S.N., E.O.K., and A.K.F. acknowledge support from the Priority 2030 program at the NIST "MISIS" under the project K1-2022-027. The experimental part of this work was supported by the Russian Roadmap on Quantum Computing (Contract No. 868-1.3-15/15-2021, October 5, 2021).

Conflicts of Interest Statement
The authors declare no conflicts of interest.

Data Availability Statement
The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments
The authors would like to thank B. I. Bantysh for insightful comments and anonymous referees for their constructive feedback.

Supplemental Materials

In this supplemental section we provide the details of a single run of the factoring algorithm. Let's consider the first random permutation $(1,3,2,5,6,4)$ used in the algorithm. The corresponding CVP is defined by the lattice

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 22 & 35 & 51 & 62 & 76 & 81 \end{pmatrix}$$

and the target vector

$$t = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 233 \end{pmatrix}.$$

The approximate solution given by Babai's algorithm based on LLL-reduction is

$$\begin{pmatrix} 19 & -23 & -41 & -32 & 32 & 0 \end{pmatrix},$$

all elements were rounded *up* to the nearest integer. The corresponding normalized (to the maximal value) matrix of QUBO coefficients (rounded to 10^{-3}) is

$$Q = \begin{pmatrix} -0.929 & -0.286 & 0.143 & 0.071 & 0.143 & 0.286 \\ 0.000 & 1.000 & -0.286 & 0.143 & -0.286 & -0.571 \\ 0.000 & 0.000 & -1.643 & -0.286 & 0.643 & 0.071 \\ 0.000 & 0.000 & 0.000 & -0.143 & 0.000 & -0.429 \\ 0.000 & 0.000 & 0.000 & 0.000 & -2.571 & 0.643 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & -1.429 \end{pmatrix}.$$

Fixed-Point-QAOA Circuits

To factor $1591 = 37 \times 43$ with fixed-point QAOA we implemented 9 quantum 6-qubit quantum circuits on a trapped-ion processor. Due to the fixed-point feature there is no need for classical-quantum hybrid optimization, therefore all necessary parameters for circuit construction can be obtained before execution on a quantum hardware. Exact architecture of executed quantum circuits with native for the processor single-qubit and two-qubit gates is presented in Figure 3. Parameters of the circuits, which correspond to angles in the gates $R_z(\theta_i)$ and $ZZ(\chi_{ij})$, are given in Table 5. When χ_{ij} is equal to zero, $ZZ(\chi_{ij})$ is not implemented. We note that to get sufficient statistics it was enough to perform 5 shots for each circuit. In total, 45 experimental shots were executed on a trapped-ion processor. To collect 12 sr-pairs 43 shots were enough.

Experimental QUBO Sampling Accuracy

In this section we present a comparison between experimentally obtained output states probabilities for circuits 1 and 6 from the Table 5 and ones calculated on a noiseless emulator (Figure 7).

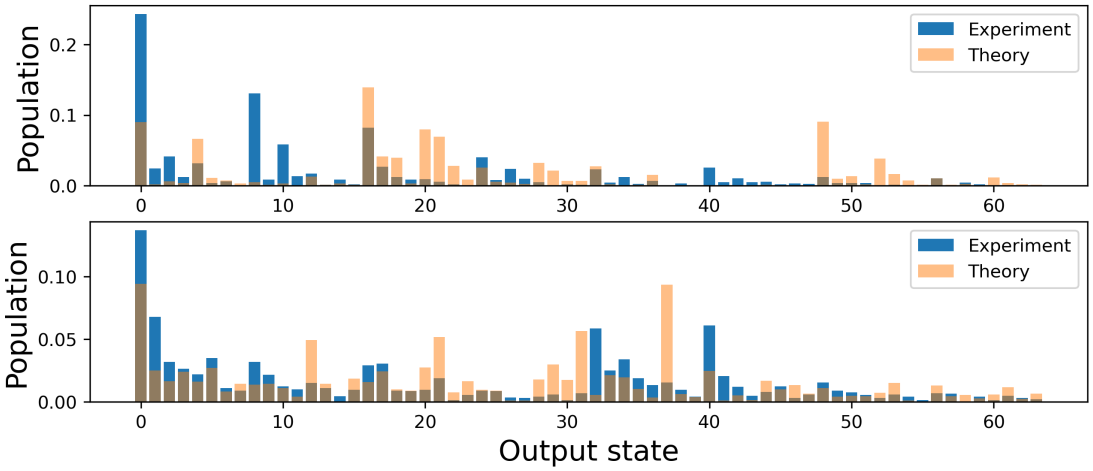


Figure 7. Output states probabilities for circuits 1 and 6 from Table 5 sampled by the quantum processor and the noiseless emulator. Output states are numbered as a decimal representation of the output bitstrings. The first qubit corresponds to the high-order digit in the bitstrings. Each histogram is an average of 2000 shots.

In Figure 8 we also show analogous output probability distributions for the circuits where we use only 5 qubits to factorize number 437. It can be seen that the sampling fidelity is generally higher than for a 6 qubit case due to smaller circuit depth. However, for such a small problem size no quantum advantage over random sampling was observed.

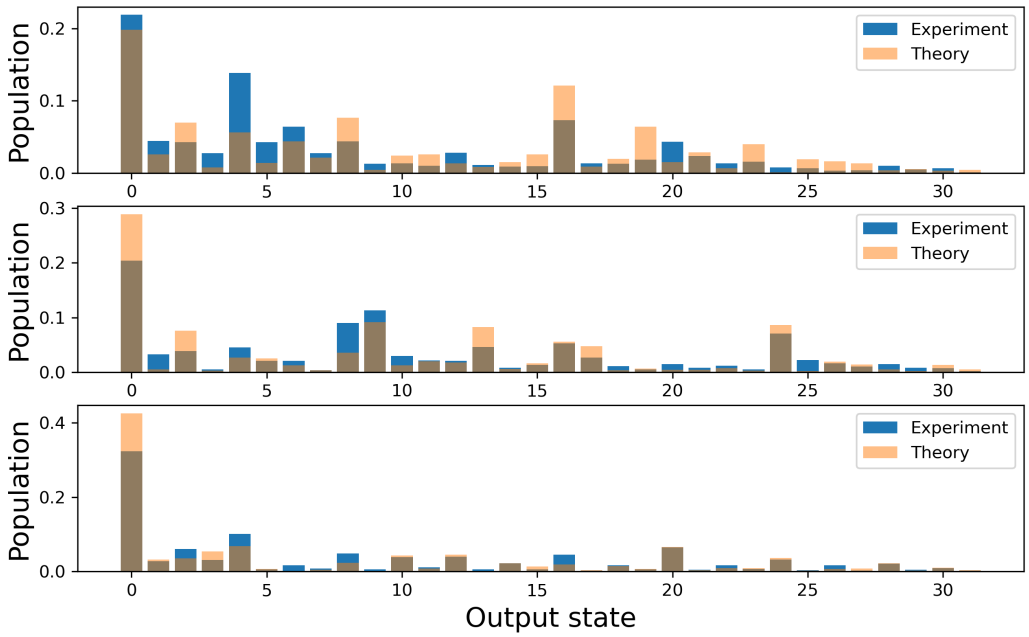


Figure 8. Output states probabilities sampled by the quantum processor and the noiseless emulator for a set of circuits used to factorize number 437 using 5 qubits. Output states are numbered as a decimal representation of the output bitstrings. The first qubit corresponds to the high-order digit in the bitstrings. Each histogram is an average of 2000 shots.

Table 4. Steps of the factoring.

Step	Permutation	Circuit	Measurement Result	sr-pair	#sr-pairs	Factoring
1	(1, 3, 2, 5, 6, 4)	1	010001		0	
2	(1, 3, 2, 5, 6, 4)	1	101000		0	
3	(1, 3, 2, 5, 6, 4)	1	000100		0	
4	(1, 3, 2, 5, 6, 4)	1	001010		0	
5	(1, 3, 2, 5, 6, 4)	1	000001		0	
6	(4, 1, 3, 6, 5, 2)	2	000010		0	
7	(4, 1, 3, 6, 5, 2)	2	001101		0	
8	(4, 1, 3, 6, 5, 2)	2	000000	(1521, 1)	1	
9	(4, 1, 3, 6, 5, 2)	2	000000		1	
10	(4, 1, 3, 6, 5, 2)	2	100000	(1690, 1)	2	
11	(3, 5, 2, 6, 4, 1)	3	001000	(5005, 3)	3	
12	(3, 5, 2, 6, 4, 1)	3	101000		3	
13	(3, 5, 2, 6, 4, 1)	3	100001		3	
14	(3, 5, 2, 6, 4, 1)	3	001100		3	
15	(3, 5, 2, 6, 4, 1)	3	000001		3	
16	(1, 4, 2, 6, 5, 3)	4	000010		3	
17	(1, 4, 2, 6, 5, 3)	4	000000	(1625, 1)	4	
18	(1, 4, 2, 6, 5, 3)	4	001000		4	
19	(1, 4, 2, 6, 5, 3)	4	001000		4	
20	(1, 4, 2, 6, 5, 3)	4	100000		4	
21	(1, 5, 4, 2, 3, 6)	5	000000	(1540, 1)	5	
22	(1, 5, 4, 2, 3, 6)	5	000000		5	
23	(1, 5, 4, 2, 3, 6)	5	100000		5	
24	(1, 5, 4, 2, 3, 6)	5	010000		5	
25	(1, 5, 4, 2, 3, 6)	5	100000		5	
26	(6, 5, 1, 2, 3, 4)	6	000001		5	
27	(6, 5, 1, 2, 3, 4)	6	101101	(41503, 25)	6	
28	(6, 5, 1, 2, 3, 4)	6	000011		6	
29	(6, 5, 1, 2, 3, 4)	6	100110	(5775, 4)	7	
30	(6, 5, 1, 2, 3, 4)	6	010011		7	
31	(5, 4, 2, 3, 1, 6)	7	000100		7	
32	(5, 4, 2, 3, 1, 6)	7	001010	(1375, 1)	8	
33	(5, 4, 2, 3, 1, 6)	7	000000	(1573, 1)	9	
34	(5, 4, 2, 3, 1, 6)	7	110000		9	
35	(5, 4, 2, 3, 1, 6)	7	100100	(3185, 2)	10	✓
36	(5, 6, 2, 4, 1, 3)	8	010100		10	✓
37	(5, 6, 2, 4, 1, 3)	8	100000		10	✓
38	(5, 6, 2, 4, 1, 3)	8	100010	(3125, 2)	11	✓
39	(5, 6, 2, 4, 1, 3)	8	011000		11	✓
40	(5, 6, 2, 4, 1, 3)	8	011000		11	✓
41	(5, 4, 3, 1, 2, 6)	9	011010		11	✓
42	(5, 4, 3, 1, 2, 6)	9	001000		11	✓
43	(5, 4, 3, 1, 2, 6)	9	000000	(1617, 1)	12	✓

Table 5. R_z and ZZ gates rotation angles of quantum circuits used in the factorization of 1591.

	Circuit1	Circuit2	Circuit3	Circuit4	Circuit5	Circuit6	Circuit7	Circuit8	Circuit9
θ_1	-0.619	0.190	-0.513	-0.619	-0.867	-1.667	0.400	-1.133	0.476
θ_2	0.667	-1.429	-0.308	0.667	0.133	-0.444	-1.067	-3.000	-0.857
θ_3	-1.095	-0.714	-1.436	-1.095	0.667	-0.556	-0.867	-1.267	-1.143
θ_4	-0.095	-1.381	-0.205	-0.095	0.067	0.333	-0.933	-2.067	-0.095
θ_5	-1.714	-1.571	-1.026	-1.714	-0.267	-1.444	-0.867	-1.200	-0.190
θ_6	-0.952	-2.095	-0.308	-0.952	-0.733	0.444	-0.067	-1.067	-0.190
χ_{12}	-0.095	-0.190	-0.026	-0.095	0.300	0.333	-0.233	0.233	-0.286
χ_{13}	0.048	0.095	0.128	0.048	-0.233	0.333	-0.133	0.067	-0.190
χ_{14}	0.024	-0.048	0.128	0.024	-0.200	0	-0.033	0.033	-0.238
χ_{15}	0.048	-0.024	0.103	0.048	-0.067	0.278	0	0.167	-0.238

Table 5 (continued).

χ_{16}	0.095	−0.095	−0.128	0.095	0.067	−0.389	−0.133	−0.133	0.238
χ_{23}	−0.095	−0.167	−0.231	−0.095	0.100	−0.167	0.200	0.200	0.333
χ_{24}	0.048	0.190	−0.205	0.048	−0.233	0.056	0.067	0.233	−0.286
χ_{25}	−0.095	0.167	0.077	−0.095	−0.200	0.056	0.167	0.167	0.048
χ_{26}	−0.190	0.190	0.077	−0.190	−0.200	−0.389	0	0.167	0.048
χ_{34}	−0.095	−0.048	0.333	−0.095	−0.033	−0.333	−0.167	−0.167	−0.095
χ_{35}	0.214	0.071	0.179	0.214	−0.167	−0.278	−0.133	−0.133	−0.190
χ_{36}	0.024	0.071	−0.128	0.024	−0.200	0.222	0.133	0.133	0.143
χ_{45}	0	−0.119	−0.128	0	−0.167	0	0.333	0.333	−0.286
χ_{46}	−0.143	0.333	−0.179	−0.143	−0.100	−0.389	−0.233	−0.067	−0.048
χ_{56}	0.214	0.119	−0.128	0.214	0	−0.444	−0.100	−0.267	−0.286

Factoring Algorithm Details

Here we provide the description of the factoring algorithm adapted from [29]. The formal detailed description of Babai's algorithm for CVP-approximation subroutine is presented in Algorithm 1. Overall factoring procedure is presented in Algorithm 2. Further, we consider the QUBO refinement of CVP-approximation. Generally, for any CVP defined by a lattice $B = \{b_{ij}\}$ and a target vector t_j, we need to find integer coefficients $a_i \in \mathbb{Z}$ minimizing $\sum_j (\sum_i a_i b_{ij} - t_j)^2$. Babai's algorithm (however, it can be replaced by any other real approximation algorithm) provides a real-valued approximation $c_i \in \mathbb{R}$. Then the rounded values $\lceil c_i \rfloor$ are being taken as an approximate integer solution. Let us define variables that specify the rounding side:

$$r_i = \begin{cases} 1 & \lceil c_i \rfloor \geq c_i \\ -1 & \lceil c_i \rfloor < c_i \end{cases}$$

.

Then the problem of choosing the side of each rounding has the form:

$$\min_{x_i \in \{0,1\}} \sum_j (\sum_i (\lceil c_i \rfloor - r_i x_i) b_{ij} - t_j)^2,$$

which is a QUBO problem with integer coefficients and can be approximately solved by the fpQAOA algorithm.

Algorithm 1 Babai's CVP algorithm Babai_δ (adapted from [29])

Require: $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{Z}^{n+1}$ ▷ Basis vectors of the considered lattice Λ
$\mathbf{t} \in \mathbb{Z}^{n+1}$ ▷ Target vector
$\delta \in (0.25, 1)$ [$\delta = 3/4$ by default] ▷ Parameter of the LLL-reduction
Ensure: $\mathbf{c} = (c_1, \ldots, c_n) \in \mathbb{Z}^n$ ▷ Rounded coefficient of the solution in LLL-reduced basis
$\bar{} = (\mu_1, \ldots, \mu_n) \in \mathbb{R}^n$ ▷ Unrounded coefficient of the solution in LLL-reduced basis
$(\mathbf{d}_1, \ldots, \mathbf{d}_n)$ ▷ LLL-reduced basis
$\mathbf{d}_1, \ldots, \mathbf{d}_n \leftarrow \mathsf{LLL}_\delta(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ ▷ Constructing LLL-reduced basis via standard procedure
$(\widetilde{\mathbf{d}}_1, \ldots, \widetilde{\mathbf{d}}_n) \leftarrow$ Gram-Schmidt orthogonalization of $(\mathbf{d}_1, \ldots, \mathbf{d}_n)$
$\mathbf{b} \leftarrow \mathbf{t}$
for $j = n, n-1, \ldots, 1$ **do**
    $\mu_i \leftarrow \left\langle \mathbf{b}, \widetilde{\mathbf{d}}_j \right\rangle / \left\langle \widetilde{\mathbf{d}}_j, \widetilde{\mathbf{d}}_j \right\rangle$
    $c_j \leftarrow \lceil \mu_j \rfloor$
    $\mathbf{b} \leftarrow \mathbf{b} - c_j \mathbf{d}_j$
end for
return $\mathbf{c}, \boldsymbol{\mu}, (\mathbf{d}_1, \ldots, \mathbf{d}_n)$

Algorithm 2 fpQAOA-enhanced factorization algorithm (adapted from [29])

Require: $N \in \mathbb{N}_+$ ▷ RSA integer to be factorized
$n \in \mathbb{N}_+$ ▷ Number of qubits used in the QAOA
$B_2 \in \mathbb{N}_+$ (s.t. $B_2 \geq n$, $p_{B_2} < N$) ▷ Size of a factor base used for checking sr-pairs
$c \in \mathbb{R}_+$ ▷ Rounding parameter
$\delta \in (0.25, 1)$ [$\delta = 3/4$ by default] ▷ Parameter of the LLL-reduction in Babai's algorithm
$\mathsf{sort} \in \{\mathrm{True}, \mathrm{False}\}$ [$\mathsf{sort} = \mathrm{False}$ by default] ▷ Rearranging parameter for Babai's algorithm
Ensure: $p, q \in \mathbb{N}$ **or Fail** ▷ Factors of N
$i \leftarrow 0$ ▷ Initialize a counter for found relations
while $i < n+1$ **do** ▷ Accumulating sr-pairs
  Fix a random permutation $f : [1, \ldots, n] \mapsto [f(1), \ldots, f(n)]$
  for $j = 1, 2, \ldots, n$ **do** ▷ Setting lattice's basis vectors
    $\mathbf{b}_j[k] \leftarrow \begin{cases} \lceil f(k)/2 \rfloor & k = j \\ 0 & k \neq j \end{cases}$ for $k = 1, 2, \ldots, n$
    $\mathbf{b}_j[n+1] \leftarrow \lceil 10^c \ln p_k \rfloor$
  end for
  $\mathbf{t}[k] \leftarrow 0$ for $k = 1, 2, \ldots, n$, $\mathbf{t}[n+1] \leftarrow \lceil 10^c \ln N \rfloor$ ▷ Setting the target vector
  $\mathbf{c}, \boldsymbol{\mu}, (\mathbf{d}_1, \ldots, \mathbf{d}_n) \leftarrow \mathsf{Babai}_\delta(\mathbf{b}_1, \ldots, \mathbf{b}_n; \mathbf{t})$ ▷ Applying Babai's algorithm to obtain rounded ($\mathbf{c}$) and unrounded ($\boldsymbol{\mu}$) coefficients of solution in the LLL-reduced basis $\mathbf{d}_1, \ldots, \mathbf{d}_n$
  $\mathbf{x} \leftarrow \mathsf{fpQAOA}(\mathbf{c}; \boldsymbol{\mu}; \mathbf{d}_1, \ldots, \mathbf{d}_n; \mathbf{t})$ ▷ Applying the QAOA to obtain corrections $x_i \in \{\pm 1, 0\}$
  $\mathbf{v}_{\mathrm{new}} \leftarrow \sum_{j=1}^{n} (c_j + x_j)\mathbf{d}_j$ ▷ Updating CVP solution
  $\mathbf{v}_{\mathrm{new}} = \sum_{j=1}^{n} e_j \mathbf{b}_j$ ▷ Rewriting the updated CVP solution in the initial basis
  $u \leftarrow \prod_{e_j > 0} p_j^{e_j}$, $v \leftarrow \prod_{e_j < 0} p_i^{-e_j}$ ▷ Constructing candidates for an sr-pair
  if $u - v \cdot N$ is P_{B_2}-smooth **and** $(u, v) \notin \{(u_j, v_j)\}_{j=1}^{i}$ **then** ▷ A new sr-pair has been found
    $u_i \leftarrow u$, $v_i \leftarrow v$ ▷ Adding the sr-pair
    $u_i = \prod_{j=0}^{B_2} p_j^{e_{j,i}}$ $u_i - N v_i = \prod_{j=0}^{B_2} p_j^{e'_{j,i}}$ ▷ Decomposing u_i and $u_i - N v_i$ in P_{B_2}
    $i \leftarrow i + 1$ ▷ Increasing counter and proceed to the next permutation
  end if
end while
$a_{i,j} \leftarrow (e'_{i,j} - e_{i,j}) \bmod 2$ ▷ Constructing the system of linear equations
$\mathrm{Ker}[(a_{i,j})] = \mathrm{Span}(\tau_1, \ldots, \tau_m)$ ▷ Extracting linearly independent solutions
for $\tau = \tau_1, \ldots, \tau_m$ **do** ▷ Looking through the solutions
  $X_\tau \leftarrow \prod_{i=0}^{B_2} p^{\frac{1}{2}\sum_{j=1}^{m}(e'_{i,j} - e_{i,j})\tau_j} \pmod N$ ▷ Constructing X_τ satisfying $X_\tau^2 = 1 \pmod N$
  if $X_\tau \not\equiv \pm 1 \pmod N$ **then** ▷ Success!
    return $\gcd(X_\tau + 1, N), \gcd(X_\tau - 1, N)$
  end if
end for
return Fail ▷ No solutions found

References

1. Shor, P. Algorithms for quantum computation: discrete logarithms and factoring. *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* **1994**, 124–134. https://doi.org/10.1109/SFCS.1994.365700
2. Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review* **1999**, *41*, 303–332. https://doi.org/10.1137/S0036144598347011
3. Rivest, R. L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM* **1978**, *21*, 120–126. https://doi.org/10.1145/359340.359342
4. Bernstein, D. J.; Lange, T. Post-quantum cryptography. *Nature* **2017**, *549*, 188–194. https://doi.org/10.1038/nature23461
5. Yunakovsky, S. E.; Kot, M.; Pozhar, N.; Nabokov, D.; Kudinov, M.; Guglya, A.; Kiktenko, E. O.; Kolycheva, E.; Borisov, A.; Fedorov, A. K. Towards security recommendations for public-key infrastructures for production environments in the post-quantum era. *EPJ Quantum Technology* **2021**, 8, 14. https://doi.org/10.1140/epjqt/s40507-021-00104-z6.

6. Lucero, E.; Barends, R.; Chen, Y.; Kelly, J.; Mariantoni, M.; Megrant, A.; O'Malley, P.; Sank, D.; Vainsencher, A.; Wenner, J.; White, T.; Yin, Y.; Cleland, A. N.; Martinis, J. M. Computing prime factors with a Josephson phase qubit quantum processor. *Nature Physics* **2012**, *8*, 719–723. https://doi.org/10.1038/nphys2385
7. Monz, T.; Nigg, D.; Martinez, E. A.; Brandl, M. F.; Schindler, P.; Rines, R.; Wang, S. X.; Chuang, I. L.; Blatt, R. Realization of a scalable Shor algorithm. *Science* **2016**, *351*, 1068. https://doi.org/10.1126/science.aad9480
8. Lu, C.-Y.; Browne, D. E.; Yang, T.; Pan, J.-W. Demonstration of a compiled version of Shor's quantum factoring algorithm using photonic qubits. *Physical Review Letters* **2007**, *99*, 250504. https://doi.org/10.1103/PhysRevLett.99.250504
9. Lanyon, B. P.; Weinhold, T. J.; Langford, N. K.; Barbieri, M.; James, D. F. V.; Gilchrist, A.; White, A. G. Experimental demonstration of a compiled version of Shor's algorithm with quantum entanglement. *Physical Review Letters* **2007**, *99*, 250505. https://doi.org/10.1103/PhysRevLett.99.250505
10. Martín-López, E.; Laing, A.; Lawson, T.; Alvarez, R.; Zhou, X.-Q.; O'Brien, J. L. Experimental realization of Shor's quantum factoring algorithm using qubit recycling. *Nature Photonics* **2012**, *6*, 773. https://doi.org/10.1038/nphoton.2012.259
11. Gidney, C.; Ekera, M. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *Quantum* **2021**, *5*, 433. https://doi.org/10.22331/q-2021-04-15-433
12. Gidney, C. How to factor 2048 bit RSA integers with less than a million noisy qubits. *arXiv preprint* **2025**, arXiv:2505.15917. https://arxiv.org/abs/2505.15917
13. Coppersmith, D. An approximate Fourier transform useful in quantum factoring. *arXiv preprint* **2002**, arXiv:quant-ph/0201067. https://arxiv.org/abs/quant-ph/0201067
14. Bocharov, A.; Roetteler, M.; Svore, K. M. Factoring with qutrits: Shor's algorithm on ternary and metaplectic quantum architectures. *Physical Review A* **2017**, *96*, 012306. https://doi.org/10.1103/PhysRevA.96.012306
15. Regev, O. An efficient quantum factoring algorithm. *arXiv preprint* **2024**, arXiv:2308.06572. https://arxiv.org/abs/2308.06572
16. Anschuetz, E.; Olson, J.; Aspuru-Guzik, A.; Cao, Y. Variational quantum factoring. In *Quantum Technology and Optimization Problems*, Feld, S.; Linnhoff-Popien, C. (Eds.), Springer International Publishing: Cham, 2019; pp. 74–85.
17. Peng, W.; Wang, B.; Hu, F.; Wang, Y.; Fang, X.; Chen, X.; Wang, C. Factoring larger integers with fewer qubits via quantum annealing with optimized parameters. *Science China Physics, Mechanics & Astronomy* **2019**, *62*, 60311. https://doi.org/10.1007/s11433-018-9307-1
18. Wang, B.; Hu, F.; Yao, H.; Wang, C. Prime factorization algorithm based on parameter optimization of Ising model. *Scientific Reports* **2020**, *10*, 7106. https://doi.org/10.1038/s41598-020-62802-5
19. Karamlou, A.H.; Simon, W.A.; Katabarwa, A.; Scholten, T.L.; Peropadre, B.; Cao, Y. Analyzing the performance of variational quantum factoring on a superconducting quantum processor. *npj Quantum Information* **2021**, *7*, 156. https://doi.org/10.1038/s41534-021-00478-z
20. Sobhani, M.; Chai, Y.; Hartung, T.; Jansen, K. Variational quantum eigensolver approach to prime factorization on IBM's noisy intermediate scale quantum computer. *Physical Review A* **2025**, *111*, 042413.
21. Yan, B.; Tan, Z.; Wei, S.; Jiang, H.; Wang, W.; Wang, H.; Luo, L.; Duan, Q.; Liu, Y.; Shi, W.; et al. Factoring integers with sublinear resources on a superconducting quantum processor. *arXiv* **2022**, arXiv:2212.12372. https://arxiv.org/abs/2212.12372
22. Schnorr, C.P. Fast factoring integers by SVP algorithms, corrected. *Cryptology ePrint Archive* **2021**, Paper 2021/933. https://eprint.iacr.org/2021/933
23. Farhi, E.; Goldstone, J.; Gutmann, S. A quantum approximate optimization algorithm. *arXiv* **2014**, arXiv:1411.4028. https://arxiv.org/abs/1411.4028
24. Farhi, E.; Harrow, A.W. Quantum supremacy through the quantum approximate optimization algorithm. *arXiv* **2019**, arXiv:1602.07674. https://arxiv.org/abs/1602.07674
25. Pagano, G.; Bapat, A.; Becker, P.; Collins, K.S.; De, A.; Hess, P.W.; Kaplan, H.B.; Kyprianidis, A.; Tan, W. L.; Baldwin, C.; et al. Quantum approximate optimization of the long-range Ising model with a trapped-ion quantum simulator. *Proceedings of the National Academy of Sciences of the United States of America* **2020**, *117*, 25396. https://doi.org/10.1073/pnas.2006373117
26. Harrigan, M. P.; Sung, K. J.; Neeley, M.; Satzinger, K. J.; Arute, F.; Arya, K.; Atalaya, J.; Bardin, J. C.; Barends, R.; Boixo, S.; et al. Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. *Nature Physics* **2021**, *17*, 332.https://doi.org/10.1038/s41567-020-01105-y

27. Bharti, K.; Cervera-Lierta, A.; Kyaw, T.H.; Haug, T.; Alperin-Lea, S.; Anand, A.; Degroote, M.; Heimonen, H.; Kottmann, J.S.; Menke, T.; Mok, W.-K.; Sim, S.; Kwek, L.-C.; Aspuru-Guzik, A. Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics* **2022**, *94*, 015004. https://doi.org/10.1103/RevModPhys.94.015004
28. Cerezo, M.; Arrasmith, A.; Babbush, R.; Benjamin, S.C.; Endo, S.; Fujii, K.; McClean, J.R.; Mitarai, K.; Yuan, X.; Cincio, L.; Coles, P.J. Variational quantum algorithms. *Nature Reviews Physics* **2021**, *3*, 625. https://doi.org/10.1038/s42254-021-00348-9
29. Grebnev, S.V.; Gavreev, M.A.; Kiktenko, E.O.; Guglya, A.P.; Efimov, A.R.; Fedorov, A.K. Pitfalls of the sublinear QAOA-based factorization algorithm. *IEEE Access* **2023**, *11*, 134760. https://doi.org/10.1109/ACCESS.2023.3336989
30. Khattar, T.; Yosri, N. A comment on "Factoring integers with sublinear resources on a superconducting quantum processor". *arXiv* **2023**, arXiv:2307.09651.
31. Hegade, N.N.; Solano, E. Digitized-counterdiabatic quantum factorization. *arXiv* **2023**, arXiv:2301.11005.
32. Chernyavskiy, A.; Bantysh, B. A method to compute QAOA fixed angles. *Russian Microelectronics* **2023**, *52*, S352–S356.
33. Brandao, F.G.; Broughton, M.; Farhi, E.; Gutmann, S.; Neven, H. For fixed control parameters the quantum approximate optimization algorithm's objective function value concentrates for typical instances. *arXiv* **2018**, arXiv:1812.04170.
34. Yan, S. Y. *Cryptanalytic attacks on RSA*. Springer New York, NY **2008**. https://doi.org/10.1007/978-0-387-48742-7
35. Babai, L. On Lovász's lattice reduction and the nearest lattice point problem. *Combinatorica* **1986**, 6, 1–13.
36. Lenstra, A. K.; Lenstra, H. W.; Lovász, L. Factoring polynomials with rational coefficients. *Mathematische Annalen* **1982**, *261*, 515–534.
37. Guerreschi, G. G.; Matsuura, A. Y. QAOA for Max-Cut requires hundreds of qubits for quantum speed-up. *Scientific Reports* **2019**, *9*, 6903.
38. Zhou, L.; Wang, S.-T.; Choi, S.; Pichler, H.; Lukin, M. D. Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. *Physical Review X* **2020**, *10*, 021067.
39. Fernández-Pendás, M.; Combarro, E.F.; Vallecorsa, S.; Ranilla, J.; Rúa, I.F. A study of the performance of classical minimizers in the quantum approximate optimization algorithm. *Journal of Computational and Applied Mathematics* **2022**, *404*, 113388.
40. Galda, A.; Liu, X.; Lykov, D.; Alexeev, Y.; Safro, I. Transferability of optimal QAOA parameters between random graphs. In *Proceedings IEEE International Conference on Quantum Computing and Engineering (QCE)*, IEEE: 2021; pp. 171–180.
41. Wurtz, J.; Lykov, D. The fixed angle conjecture for QAOA on regular MaxCut graphs. *arXiv* **2021**, arXiv:2107.00677.
42. Chernyavskiy, A.Y. Calculation of quantum discord and entanglement measures using the random mutations optimization algorithm. *arXiv* **2013**, arXiv:1304.3703.
43. Bantysh, B.; Bogdanov, Y.I. Quantum tomography of noisy ion-based qudits. *Laser Physics Letters* **2020**, *18*, 015203.
44. Zalivako, I.V.; Nikolaeva, A.S.; Borisenko, A.S.; Korolkov, A.E.; Sidorov, P.L.; Galstyan, K.P.; Semenin, N.V.; Smirnov, V.N.; Aksenov, M.A.; Makushin, K.M.; Kiktenko, E.O.; Fedorov, A.K.; Semerikov, I.A.; Khabarova, K. Y.; Kolachevsky, N.N. Towards a multiqudit quantum processor based on a 171Yb+ ion string: Realizing basic quantum algorithms. *Quantum Reports* **2025**, *7*, 19. https://doi.org/10.3390/quantum7020019
45. McKay, D.C.; Wood, C.J.; Sheldon, S.; Chow, J.M.; Gambetta, J.M.; Efficient Z gates for quantum computing. *Physical Review A* **2017**, *96*, 022330.
46. Schmidt-Kaler, F.; Häffner, H.; Riebe, M.; Gulde, S.; Lancaster, G.P.T.; Deuschle, T.; Becher, C.; Roos, C.F.; Eschner, J.; Blatt, R. Realization of the Cirac–Zoller controlled-NOT quantum gate. *Nature* **2003**, *422*, 408. https://doi.org/10.1038/nature01494
47. Mølmer, K.; Sørensen, A. Multiparticle entanglement of hot trapped ions. *Physical Review Letters* **1999**, *82*, 1835. https://doi.org/10.1103/PhysRevLett.82.1835
48. Sørensen, A.; Mølmer, K. Quantum computation with ions in thermal motion. *Physical Review Letters* **1999**, *82*, 1971. https://doi.org/10.1103/PhysRevLett.82.1971

49. Sørensen, A.; Mølmer, K. Entanglement and quantum computation with ions in thermal motion. *Physical Review A* **2000**, *62*, 022311. https://doi.org/10.1103/PhysRevA.62.022311
50. Choi, T.; Debnath, S.; Manning, T.; Figgatt, C.; Gong, Z.X.; Duan, L.M.; Monroe, C. Optimal quantum control of multimode couplings between trapped ion qubits for scalable entanglement. *Physical Review Letters* **2014**, *112*, 190502.
51. Semenin, N.V.; Borisenko, A.S.; Zalivako, I.V.; Semerikov, I.A.; Khabarova, K.Y.; Kolachevsky, N.N. Optimization of the readout fidelity of the quantum state of an optical qubit in the $^{171}Yb^{+}$ ion. *JETP Letters* **2021**, *114*, 486.
52. Magesan, E.; Gambetta, J.M.; Emerson, J. Characterizing quantum gates via randomized benchmarking. *Physical Review A* **2012**, *85*, 042311.
53. Benhelm, J.; Kirchmair, G.; Roos, C.F.; Blatt, R. Towards fault-tolerant quantum computing with trapped ions. *Nature Physics* **2008**, *4*, 463–466. https://doi.org/10.1038/nphys961
54. Brown, K.R.; Harrow, A.W.; Chuang, I.L. Arbitrarily accurate composite pulse sequences. *Physical Review A* **2004**, *70*, 052318.
55. Zalivako, I.V.; Semenin, N.V.; Zhadnov, N.O.; Galstyan, K.P.; Kamenskikh, P.A.; Smirnov, V.N.; Evgenyevich, K. A.; Leonidovich, S.P.; Borisenko, A.S.; Anosov, Y.P. Quantum computing with trapped ions: principles, achievements, and prospects. *Physics-Uspekhi* **2025**, *68*.
56. Aboumrad, W.; Widdows, D.; Kaushik, A. Quantum and classical combinatorial optimizations applied to lattice-based factorization. *arXiv* **2023**, arXiv:2308.07804.
57. Luan, L.; Gu, C.; Zheng, Y.; Shi, Y. Lattice enumeration with discrete pruning: Improvements, cost estimation and optimal parameters. *Mathematics* **2023**, *11*, 766.
58. Shaydulin, R.; Li, C.; Chakrabarti, S.; DeCross, M.; Herman, D.; Kumar, N.; Larson, J.; Lykov, D.; Minssen, P.; Sun, Y.; et al. Evidence of scaling advantage for the quantum approximate optimization algorithm on a classically intractable problem. *Science Advances* **2024**, *10*, eadm6761.
59. Priestley, B.; Wallden, P. A practically scalable approach to the closest vector problem for sieving via QAOA with fixed angles. *arXiv* **2025**, arXiv:2503.08403. https://arxiv.org/abs/2503.08403