Hybrid Decoding – Classical-Quantum Trade-Offs for Information Set Decoding

Andre Esser[1], Sergi Ramos-Calderer[12], Emanuele Bellini[1], José I. Latorre[123], and Marc Manzano[4⋆]

[1] Technology Innovation Institute, UAE
{andre.esser, sergi.ramos, emanuele.bellini, jose.ignacio.latorre}@tii.ae
[2] Departament de Física Quàntica i Astrofísica and Institut de Ciències del Cosmos, Universitat de Barcelona, Spain
[3] Centre for Quantum Technologies, National University of Singapore, Singapore
[4] SandboxAQ, Palo Alto, CA, United States
marc@sandboxaq.com

Abstract. The security of code-based constructions is usually assessed by Information Set Decoding (ISD) algorithms. In the quantum setting, amplitude amplification yields an asymptotic square root gain over the classical analogue. However, already the most basic ISD algorithm by Prange suffers enormous width requirements caused by the quadratic description length of the underlying problem. Even if polynomial, this need for qubits is one of the biggest challenges considering the application of real quantum circuits in the near- to mid-term.

In this work we overcome this issue by presenting the first hybrid ISD algorithms that allow to tailor the required qubits to any available amount while still providing quantum speedups of the form T^δ, $0.5 < \delta < 1$, where T is the running time of the purely classical procedure. Interestingly, when constraining the width of the circuit instead of its depth we are able to overcome previous optimality results on constraint quantum search.

Further we give an implementation of the fully-fledged quantum ISD procedure and the classical co-processor using the quantum simulation library *Qibo* and *SageMath*.

Keywords: decoding, width reduction, hybrid algorithms, code-based cryptography

1 Introduction

The growing threat to modern widespread cryptography posed by the advancing development of quantum computers has led to a focus on other hardness assumptions. One of the leading and most promising proposals for post quantum

⋆ This work was conducted while the author was affiliated with Technology Innovation Institute.

cryptography is code-based cryptography. It has a long history of withstanding classical as well as quantum attacks and is considered to rely on one of the most well understood hardness assumptions. The list of the four KEM finalists of the ongoing NIST standardization process for post quantum cryptography [1] includes one code-based proposal (McEliece [10]) and two more can be found on the alternate candidate list (BIKE [2] and HQC [24]) .

At the heart of all these code-based constructions lies the binary decoding or *syndrome decoding* problem. This problem asks to find a low Hamming weight solution $\mathbf{e} \in \mathbb{F}_2^n$ to the equation $H\mathbf{e} = \mathbf{s}$, where $H \in \mathbb{F}_2^{(n-k)\times n}$ is a random binary matrix and $\mathbf{s} \in \mathbb{F}_2^{n-k}$ a binary vector.

The best known strategy to solve this problem is based on Information Set Decoding (ISD) [27], a technique introduced by Prange in 1962. Since then, there has been a series of works improving on his original algorithm [4,8,11,22,23,28], mostly by leveraging additional memory. In the quantum setting Bernstein showed how to speed up Prange's algorithm by an amplitude amplification routine [5], which results in an asymptotic square root gain over the classical running time. The translation of advanced ISD algorithm to the quantum setting [19,20] yields only small asymptotic improvements. So far these improvements can not compensate for the introduced overhead in terms of width and quantum RAM if looking towards implementations. This is not surprising, since already Prange's algorithm with an only *polynomial* demand for qubits, is limited by its width requirements. This is because all code-based constructions usually involve parity-check matrices consisting of millions of bits.

To overcome this issue we develop hybrid classical-quantum ISD algorithms that enable us to reduce the required amount of qubits to any available amount while still providing quantum speedups. The idea of such classical co-processors has mostly been used to parallelize quantum circuits or instantiate circuits under depth constraints, e.g. when analyzing the quantum security of schemes under the `MAXDEPTH` constraint specified by NIST [2, 6, 7, 14, 18]. Under depth constraints, Zalka [30] showed that the optimal way to perform a quantum search is by partitioning the search space in small enough sets such that the resulting circuit only targeting one set at a time does not exceed the maximum depth. Then the search has to be applied for every set of the partition. However, this optimality result only holds under depth constraints, when instead imposing constraints on the width of the circuit, our trade-offs yield more efficient strategies.

A first attempt to formulate hybrid ISD algorithms were made by Perriello et al. in [26]. However, their construction splits into a classical ISD part and a quantum exhaustive search part allowing to speed up the classical procedure with exponential time T only by a polynomial factor. In comparison our trade-offs achieve speedups of order T^δ for $0.5 < \delta < 1$.

Our Contribution. As a first contribution we design the full circuit performing the quantum version of Prange's algorithm and provide a functional implemen-

tation using the quantum simulation library Qibo [12, 13].[5] Further we describe an optimized circuit that only requires $(n-k)k$ bits to store and operate on the input matrix $H \in \mathbb{F}_2^{(n-k)\times n}$.

Our major contribution is the design of hybrid quantum-classical trade-offs that address the practical limitation on the amount of qubits. In particular, these trade-offs enable quantum speedups for any available amount of qubits. We study the behavior of our trade-offs for various different choices of code parameters. Besides the coding-theoretic motivated settings of full and half distance decoding, these include also the parameter choices made by the NIST PQC candidates McEliece, BIKE and HQC. Our trade-offs perform best on the BIKE and HQC schemes, which is a result of a combination of a very low error weight and a comparably low code rate used by these schemes.

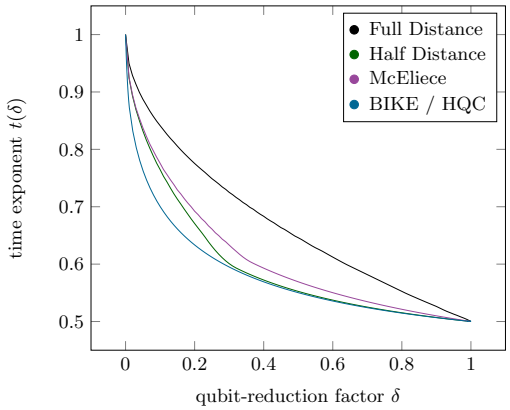


Fig. 1: Comparison of the achieved speedups of our trade-offs $t(\delta)$ (y-axis) plotted as a function of the qubit-reduction factor δ (x-axis).

Our trade-offs allow for a smooth interpolation between purely classical computations at a running time of T_C and a purely quantum based computation taking time $\sqrt{T_C}$. We interpolate between both complexities using a qubit reduction factor δ, where a fully classical computation corresponds to $\delta = 0$ and an entirely quantum based execution implies $\delta = 1$. For each trade-off we then state the running time for a given reduction factor δ as $t(\delta) \in [\![0.5, 1]\!]$, meaning that a reduction of the amount of qubits by a factor of δ implies a total running time of $(T_C)^{t(\delta)}$.

Fig. 1 shows the behavior of our trade-off achieving the best results under limited width. For instance in the BIKE and HQC setting we can reduce the amount of qubits to only 1% ($\delta = 0.01$) of an entire quantum based computa-

[5] Our implementation (available at `https://github.com/qiboteam/qISD`) also includes an implementation of the Lee-Brickel [21] ISD improvement.

tion and still achieve a speedup of roughly $t(\delta) = 0.87$ compared to a classical computation.

2 Preliminaries

For two integers $a, b \in \mathbb{N}$ with $a \leq b$ let $[a, b] := \{a, a+1, \ldots, b\}$. Further we write conveniently $[b] := [1, b]$. Let H be an $m \times n$ matrix and $I \subseteq [n]$, we write H_I to denote the projection of H onto the columns indexed by I. We use the same notation for vectors. For a binary vector $\mathbf{w} \in \mathbb{F}_2^n$ we define $\mathrm{wt}(\mathbf{w}) := |\{i \in [n] \mid w_i = 1\}|$ as the Hamming weight of $\mathbf{w}$. For two reals $c, d \in \mathbb{R}$ we let $[\![c, d]\!] := \{x \in \mathbb{R} \mid c \leq x \leq d\}$ be the (including) interval of all reals between c and d.

We use standard Landau notation for complexity statements, where $\tilde{\mathcal{O}}$-notation suppresses polylogarithmic factors, meaning $\tilde{\mathcal{O}}\big(f(x)\big) = \mathcal{O}\big(f(x)\log^i f(x)\big)$ for any constant i. All logarithms are binary if not stated otherwise. We define $\mathrm{H}(x) := -x\log(x) - (1-x)\log(1-x)$ to be the binary entropy function and make use of the well-known approximation

$$\binom{n}{k} = \tilde{\Theta}\left(2^{n\mathrm{H}\left(\frac{k}{n}\right)}\right). \tag{1}$$

Quantum Circuits. Our algorithms are built in the quantum circuit model, where we assume a certain familiarity of the reader (for an introduction see [25]). Note that we use the term circuit depth and time complexity interchangeably when analyzing our quantum circuits.

Decoding and linear codes. A binary linear code $\mathcal{C}$ is a k dimensional subspace of $\mathbb{F}_2^n$ with minimum distance d, which is defined as the minimum Hamming weight of the elements of $\mathcal{C}$. We call n the code length and $R := \frac{k}{n}$ the code rate of $\mathcal{C}$. The code $\mathcal{C}$ can be defined via the kernel of a matrix $H \in \mathbb{F}_2^{(n-k)\times n}$, so that $\mathcal{C} := \{\mathbf{c} \in \mathbb{F}_2^n \mid H\mathbf{c}^T = \mathbf{0}\}$, where H is called a *parity-check matrix*. Note that for ease of exposition, we treat all vectors as column vectors so that we can omit vector transpositions.

A given point $\mathbf{x} = \mathbf{c} + \mathbf{e}$ that differs from a codeword by an error $\mathbf{e}$ can be uniquely decoded to $\mathbf{c}$ as long as $\mathrm{wt}(\mathbf{e}) \leq \left\lfloor \frac{d-1}{2} \right\rfloor$. This setting, in which the error weight is bounded by half of the minimum distance, is also known as *half distance* decoding, while the setting bounding it by d is known as *full distance* decoding. We study the performance of our algorithms in these settings for random codes, which are known to meet the Gilbert-Varshamov bound [17, 29], i.e., $d \approx \mathrm{H}^{-1}(1-R)n$.

The definition of the code via its parity-check matrix allows to treat the decoding procedure independently of the specific codeword by considering the *syndrome* $\mathbf{s}$ of a given faulty codeword $\mathbf{x}$, where $\mathbf{s} := H\mathbf{x} = H(\mathbf{c} + \mathbf{e}) = H\mathbf{e}$. Recovering $\mathbf{e}$ from given H and $\mathbf{s}$ is, hence, equivalent to decoding $\mathbf{x}$ to c. This leads to the definition of the *syndrome decoding problem*.

Algorithm 1 PRANGE

Require: parity-check matrix $H \in \mathbb{F}_2^{(n-k)\times n}$, syndrome $\mathbf{s} \in \mathbb{F}_2^{n-k}$, weight $\omega \in [n]$
Ensure: error vector $\mathbf{e}$ with $\mathrm{wt}(\mathbf{e}) = \omega$ satisfying $H\mathbf{e} = \mathbf{s}$
1: **repeat**
2: choose random permutation matrix $P \in \mathbb{F}_2^{n\times n}$ and set $H_I \leftarrow (HP)_{[n-k]}$
3: solve linear system $H_I\mathbf{e}_1 = \mathbf{s}$ for $\mathbf{e}_1$
4: **until** $\mathrm{wt}(\mathbf{e}_1) = \omega$
5: **return** $P(\mathbf{e}_1, 0^k)$

Definition 1 (Syndrome Decoding Problem). *Let $\mathcal{C}$ be a linear code with parity-check matrix $H \in \mathbb{F}_2^{(n-k)\times n}$ and constant* rate $R := \frac{k}{n}$. *For $\mathbf{s} \in \mathbb{F}_2^{n-k}$ and $\omega \in [n]$, the* syndrome decoding problem $\mathcal{SD}_{n,k,\omega}$ *asks to find a vector $\mathbf{e} \in \mathbb{F}_2^n$ of weight $\mathrm{wt}(\mathbf{e}) = \omega$ satisfying $H\mathbf{e} = \mathbf{s}$. We call any such $\mathbf{e}$ a* solution *while we refer to $(H, \mathbf{s}, \omega)$ as an* instance *of the $\mathcal{SD}_{n,k,\omega}$.*

Prange's Information Set Decoding Given an instance $(H, \mathbf{s}, \omega)$ of the $\mathcal{SD}_{n,k,\omega}$ Prange's algorithm [27] starts by choosing a random set $I \subseteq [n]$ of size $n - k$ and then solves the corresponding linear system

$$H_I\mathbf{e}_1 = \mathbf{s} \tag{2}$$

for $\mathbf{e}_1$.[6] Note that any solution $\mathbf{e}_1$ of weight $\omega' := \mathrm{wt}(\mathbf{e}_1)$ can easily be extended to a vector $\tilde{\mathbf{e}} \in \mathbb{F}_2^n$ of same weight satisfying $H\tilde{\mathbf{e}} = \mathbf{s}$, by setting the corresponding coordinates to zero. Hence, if $\omega' = \omega$ the vector $\tilde{\mathbf{e}}$ forms a solution to the syndrome decoding problem. The algorithm now chooses random subsets I until $\omega' = \omega$ holds.

The algorithm is successful whenever $\mathbf{e}$ projected to the coordinates given by I is a solution to the linear system in Eq. (2), hence if $\mathbf{e}_1 = \mathbf{e}_I$. This happens whenever $\mathbf{e}_I$ covers the full weight of $\mathbf{e}$, in which case I or more precisely $[n]\setminus I$ is called an *information set*. Transferred to Algorithm 1 this applies whenever for the permutation chosen in line 2, it holds that $P^{-1}\mathbf{e} = (\mathbf{e}_1, 0^k)$ for $\mathbf{e}_1 \in \mathbb{F}_2^{n-k}$. The probability that the permutation distributes the weight in such a way is

$$q := \Pr\left[P^{-1}\mathbf{e} = (\mathbf{e}_1, 0^k)\right] = \frac{\binom{n-k}{\omega}}{\binom{n}{\omega}} \ . \tag{3}$$

Hence, the expected number of tries until we draw a suitable permutation P becomes q^{-1} and the expected time complexity is $T = q^{-1} \cdot T_G$, where T_G describes the cost for solving the linear system and performing the weight check.

Remark 1. Note that in the case of S existent solutions the time complexity to retrieve a single solution with Prange's algorithm becomes $\frac{T}{S}$.

[6] Note that in Algorithm 1 we model H_I as the first $n - k$ columns of HP, where P is a random permutation matrix.

3 A quantum ISD circuit design

Let us briefly sketch how we realized the quantum design of Prange's algorithm, a detailed description of every part of the circuit can be found in the full version of this article [16]. Our design is composed of the following three main building blocks:

1) The creation of the uniform superposition over all size-$(n-k)$ subsets of $[n]$ (corresponding to the selection of information sets in line 2 of Algorithm 1).
2) The Gaussian elimination step to derive the error related to a given information set (line 3 of Algorithm 1).
3) A quantum search for an information set yielding an error of the desired weight (substituting the repeat loop in line 1 of Algorithm 1).

Superposition Circuit We realize the creation of the superposition over all size-$(n-k)$ subsets in a bit-by-bit fashion, obtaining a depth of $(n-k)\cdot n$. This is possible since the number of sets including element i is independent of all subsequent elements $j > i$. More recent developments construct this superposition in depth linear in n [3]. However, since this part of the ISD circuit does not dominate the overall depth, we refrain from further optimizations.

Gaussian Elimination Our Gaussian elimination circuit mostly resembles its classical analogue. The integration of the superposition and Gaussian elimination circuit works by first swapping all selected columns (determined by the superposition) to the back of the matrix and then implementing the Gaussian Elimination only on the last $n-k$ columns.

Quantum Search The square root gain over the classical algorithm is achieved by employing an amplitude amplification procedure. Here the diffusion layer consists of our initial superposition circuit, while the sign flip is performed based on the Hamming weight of the error obtained by performing the Gaussian elimination circuit.

We find that our circuit has a depth of $\mathcal{O}\left(\frac{n^3 \log n}{\sqrt{q}}\right)$, where q is the probability detailed in Eq. (3). This corresponds to only a logarithmic overhead compared to a classical implementation. The width of the circuit is dominated by the space required for storing the parity-check matrix, which is $(n-k)\cdot n$. In the next section, we detail a procedure to reduce the width to about $(n-k)\cdot k = (1-R)\cdot R\cdot n^2$, relying on first transforming the parity check matrix into *systematic form* $H := (I_{n-k} \mid H')$, where $H' \in \mathbb{F}_2^{(n-k)\times k}$ via Gaussian elimination. We then show that the circuit can be adapted to work only with H' as input. However, the required amount of qubits is still quadratic in the code length n and, hence, one of the most limiting factors in terms of concrete implementations.

3.1 Reducing the Width for free

In the following we assume the parity-check matrix H to be in systematic form, as shown in Fig. 2. We now describe how to adapt the quantum circuit to only require the matrix H' as well as the corresponding syndrome as an input, which saves $(n-k)^2$ qubits. Recall that the goal of the Gaussian elimination procedure is to obtain the identity matrix on the matrix projected to the columns of the currently selected subset by elementary row operations. Our previous quantum circuit achieved this by fist swapping all columns that belong to the selected subset (determined by the superposition) to the back of the matrix and then performing the Gaussian elimination always on the last $n-k$ columns. But since we now only obtain H' as input this is not possible anymore.

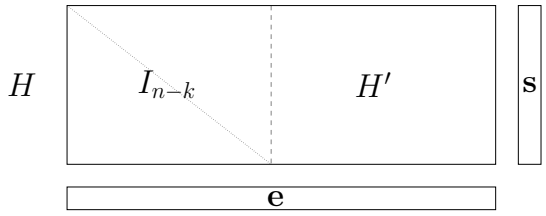


Fig. 2: Problem shape for input matrix in systematic form.

However, note that if any of the first $n-k$ columns, which are already unit vectors, belongs to the selected subset a single row swap is sufficient to obtain the desired unit vector in that column. Hence, we only implement a corresponding row swap on H' and $\mathbf{s}$. Furthermore, the necessary swaps are fully determined by the index of the respective column and its position in the selected subset. Thus, we can embed them into the quantum circuit a priori. After the necessary row-swaps are performed, all columns of H' belonging to the corresponding subset are swapped to the back. Subsequently we perform the Gaussian elimination only on the last columns of H' that belong to the current selection. This procedure is depicted in Fig. 3, which shows the state of the matrix after all three operations have been performed for the chosen subset. Note that the first $n-k$ columns only serve an illustrative purpose and are not part of the input.

4 Classical-time quantum-memory trade-offs

Next we introduce our trade-offs, allowing for an adaptive scaling of the algorithm to the available amount of qubits. Our trade-offs are divided in a classical and quantum computation part, where a decrease of the amount of qubits comes at the cost of an increased classical running time. Since the increase in running time is exponential we neglect polynomial factors by employing $\tilde{\mathcal{O}}$-notation. Our trade-offs allow for a smooth interpolation between purely classical computations

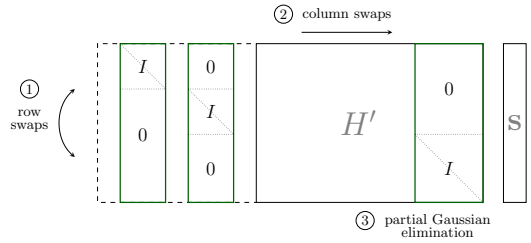


Fig. 3: Procedure to perform quantum version of Prange without first $n-k$ columns as input. Colored framed parts indicate columns belonging to the current selected subset.

at a running time of

$$T_{\mathrm{C}} := \tilde{\mathcal{O}}\left(\frac{\binom{n}{\omega}}{\binom{n-k}{\omega}}\right), \tag{4}$$

(compare to Eq. (3)) and a purely quantum based computation taking time $\sqrt{T_{\mathrm{C}}}$. Recall that we interpolate between both complexities using a qubit reduction factor δ and state the running time for a given reduction factor as $t(\delta) \in [\![0.5, 1]\!]$; meaning that a reduction of the amount of qubits by a factor of δ implies a total running time of $(T_{\mathrm{C}})^{t(\delta)}$.

We start with a trade-off based on shortening the underlying code, which already achieves a better than linear dependence between δ and $t(\delta)$. After that, we present a second trade-off based on puncturing the code which asymptotically outperforms the first one. However, for concrete parameters in medium scale both trade-offs remain superior to each other for certain values of δ. Finally, we obtain improvements by combining both methods.

4.1 Shortening the code

Our first trade-off is based on shortening the underlying code before using it as input to the quantum circuit. In Prange's original algorithm k zero positions of $\mathbf{e}$ are guessed and then the linear system corresponding to the non-zero positions is solved in polynomial time. In our hybrid version the classical part consists in guessing $\alpha n \leq k$ zero coordinates of $\mathbf{e}$, which allows to shorten the code and, hence, reduce the problem to a code of length $(1-\alpha)n$ and dimension $k-\alpha n$, while the error weight remains unchanged (compare to Fig. 4). This reduced instance is then solved with our previously constructed quantum circuit. Should the quantum computation not result in an actual solution, the initial guess of zero coordinates was incorrect and we proceed with a new guess. Algorithm 2 gives a pseudocode description of our Shortened-Hybrid.

Theorem 1 (Shortened Hybrid). *Let $n \in \mathbb{N}$, $\omega = \tau n$ and $k = Rn$ for $\tau, R \in [\![0, 1]\!]$ and let T_{C} be as defined in Eq. (4). Then for any qubit reduction factor $\delta \in$*

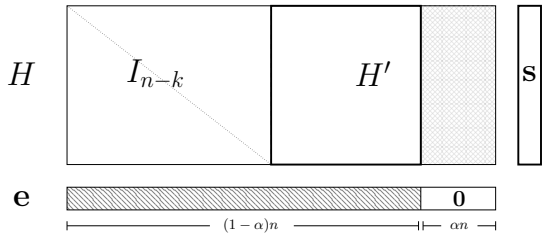


Fig. 4: Parity-check matrix in sysetmatic form where αn zero positions of $\mathbf{e}$ are guessed. Striped region of $\mathbf{e}$ indicates parts containing weight, crosshatched columns of H' do not affect $\mathbf{s}$. Framed parts are used as input to the quantum algorithm.

Algorithm 2 SHORTENED-HYBRID

Require: parity-check matrix $H \in \mathbb{F}_2^{(n-k)\times n}$, syndrome $\mathbf{s} \in \mathbb{F}_2^{n-k}$, weight $\omega \in [n]$, qubit reduction factor $\delta \in [\![0,1]\!]$
Ensure: error vector $\mathbf{e}$ with $\mathrm{wt}(\mathbf{e}) = \omega$ satisfying $H\mathbf{e} = \mathbf{s}$
1: $\alpha := (1-\delta)\frac{k}{n}$
2: **repeat**
3: choose random permutation matrix $P \in \mathbb{F}_2^{n\times n}$ and set $\tilde{H} \leftarrow HP$
4: solve instance $(\tilde{H}_{[(1-\alpha)n]}, \mathbf{s}, \omega)$ via quantum algorithm returning $\mathbf{e}_1 \in \mathbb{F}_2^{(1-\alpha)n}$
5: $\mathbf{e} \leftarrow P(\mathbf{e}_1, 0^{\alpha n})$
6: **until** $H\mathbf{e} = \mathbf{s}$
7: **return** $\mathbf{e}$

$[\![0,1]\!]$ *Algorithm 2 solves the* $\mathcal{SD}_{n,k,\omega}$ *problem in time* $(T_{\mathrm{C}})^{t(\delta)}$ *using* $\delta(1-R)Rn^2$ *qubits for the matrix representation, where*

$$t(\delta) = 1 - \frac{\frac{1}{2}\left((1-\alpha)\mathrm{H}\left(\frac{\tau}{1-\alpha}\right) - (1-R)\mathrm{H}\left(\frac{\tau}{1-R}\right)\right)}{\mathrm{H}(\tau) - (1-R)\mathrm{H}\left(\frac{\tau}{1-R}\right)},$$

for $\alpha = (1-\delta)R$.

Proof. Assume that the permutation P distributes the error such that

$$P^{-1}\mathbf{e} = (\mathbf{e}_1, 0^{\alpha n}), \tag{5}$$

for α as defined in Algorithm 2. Then it follows, that $\mathbf{e}_1$ is a solution to syndrome decoding instance $((HP)_{[(1-\alpha)n]}, \mathbf{s}, \omega)$. By the correctness of our quantum circuit the solution $\mathbf{e}_1$ is returned in line 4 and finally $\mathbf{e} = P(\mathbf{e}_1, 0^{\alpha n})$ is recovered.

Next let us analyze the running time of the algorithm. The probability of a random permutation distributing the error weight as given in Eq. (5) is

$$q_{\mathrm{C}} := \Pr\left[P^{-1}\mathbf{e} = (\mathbf{e}_1, 0^{\alpha n})\right] = \frac{\binom{(1-\alpha)n}{\omega}}{\binom{n}{\omega}}.$$

Hence, we expect that after q_{C}^{-1} random permutations one of them induces the desired weight-distribution. The asymptotic time complexity for the execution of the quantum circuit to solve the corresponding $\mathcal{SD}_{(1-\alpha)n,(R-\alpha)n,\omega}$ problem is given as (compare to Section 3).

$$T_{\mathrm{Q}} = \tilde{\mathcal{O}}\left(\sqrt{\frac{\binom{(1-\alpha)n}{\omega}}{\binom{(1-R)n}{\omega}}}\right).$$

Since for each classically chosen permutation we need to execute our quantum circuit the total running time becomes

$$T = q_{\mathrm{C}}^{-1} \cdot T_{\mathrm{Q}} = \tilde{\mathcal{O}}\left(\frac{\binom{n}{\omega}}{\sqrt{\binom{(1-\alpha)n}{\omega}\binom{(1-R)n}{\omega}}}\right).$$

Now let us determine $t(\delta) := \frac{\log T}{\log T_{\mathrm{C}}}$. First observe that $T = \frac{T_{\mathrm{C}}}{T_{\mathrm{Q}}}$, which can be rewritten as

$$\begin{aligned} &\log T_{\mathrm{C}} - \log T_{\mathrm{Q}} = \log T \\ \Leftrightarrow \qquad &1 - \frac{\log T_{\mathrm{Q}}}{\log T_{\mathrm{C}}} = \frac{\log T}{\log T_{\mathrm{C}}} =: t(\delta). \end{aligned}$$

An approximation of T_{Q} and T_{C} via the approximation for binomial coefficients given in Eq. (1) together with $\omega := \tau n$ and $k := Rn$ then yields

$$t(\delta) = 1 - \frac{\frac{1}{2}\left((1-\alpha)\mathrm{H}\left(\frac{\tau}{1-\alpha}\right) - (1-R)\mathrm{H}\left(\frac{\tau}{1-R}\right)\right)}{\mathrm{H}(\tau) - (1-R)\mathrm{H}\left(\frac{\tau}{1-R}\right)},$$

as claimed. Note that the input matrix of a code of length $(1-\alpha)n$ and dimension $(R-\alpha)n$ requires $(1-R)(R-\alpha)n^2$ qubits for the matrix representation (compare to Section 3). Hence, by setting $\alpha = (1-\delta)R$ we obtain a qubit reduction by

$$\frac{(1-R)(R-\alpha)n^2}{(1-R)Rn^2} = \frac{R-(1-\delta)R}{R} = \delta. \qquad \square$$

Next we simplify the statement of Theorem 1 for sublinear error-weight, which is, e.g., the case for McEliece, BIKE and HQC. Note that in the case of a sublinear error-weight, T_{C} can be expressed as

$$T_{\mathrm{C}} := \tilde{\mathcal{O}}\left(\frac{\binom{n}{\omega}}{\binom{(1-R)n}{\omega}}\right) = \tilde{\mathcal{O}}\left((1-R)^{-\omega}\right), \tag{6}$$

see, e.g., [15, Remark A.1].

This allows us to give the following simplified corollary.

Corollary 1 (Shortened Hybrid for sublinear error weight). *Let all parameters be as in Theorem 1. For $\tau = o(1)$, we have*

$$t(\delta) = \frac{1}{2} \cdot \left(1 + \frac{\log(1-(1-\delta)R)}{\log(1-R)}\right).$$

Proof. First we approximate T_{Q} similar to T_{C} in Eq. (6) as

$$T_{\mathrm{Q}} = \tilde{\mathcal{O}}\left(\sqrt{\frac{\binom{(1-\alpha)n}{\omega}}{\binom{(1-R)n}{\omega}}}\right) = \tilde{\mathcal{O}}\left(\left(\frac{1-\alpha}{1-R}\right)^{\frac{\omega}{2}}\right).$$

Now we can derive the statement of the corollary as

$$\begin{aligned} t(\delta) &= 1 - \frac{\log T_{\mathrm{Q}}}{\log T_{\mathrm{C}}} = 1 - \frac{\frac{\omega}{2}\left(\log(1-\alpha) - \log(1-R)\right)}{-\omega \log(1-R)} \\ &= \frac{1}{2} \cdot \left(1 + \frac{\log(1-(1-\delta)R)}{\log(1-R)}\right). \end{aligned}$$

□

Fig. 5 visualizes the relation between the qubit-reduction factor and the speedup for the full distance decoding setting with rate $R = 0.5$ and $\tau = \mathrm{H}^{-1}(R) \approx 0.11$ and the parameters of the McEliece scheme, which are $R = 0.8$ and $\tau = o(1)$. We observed that the trade-off behavior is very insensitive to changes in the error-rate. Therefore the behavior for the settings of full and half distance as well as BIKE and HQC are almost identical, such that we only included the full distance case for the sake of clarity.

However, the trade-off is more sensitive to changes in the code-rate. We observe better performance the higher the code-rate, which lies in favour to mounting an attack against codes using McEliece parameters. To give a concrete example, our Shortened-Hybrid algorithm allows for a reduction of the necessary qubits by 80% (corresponding to $\delta = 0.2$), while still achieving a speedup of $t(\delta) \approx 0.82$ in the McEliece setting.

4.2 Puncturing the code

While our Shortened-Hybrid decreases the amount of necessary qubits by shortening the code, our second trade-off instead aims at puncturing the code. In a nutshell, we consider only $(1-\beta)n-k$ parity-check equations, rather than all $n-k$, i.e., we omit βn rows of the parity-check matrix. The subsequently applied quantum circuit, hence, needs fewer qubits to represent matrix and syndrome. The advantage over Shortened-Hybrid partly comes form the fact that each row saves n instead of only $n-k$ bits. Also the generated classical overhead is significantly smaller. This variant has similarities with the Canteaut-Chabaud improvement [9] in the classical setting. Here only a certain amount of columns (originally only one) of the identity part are exchanged in each iteration rather

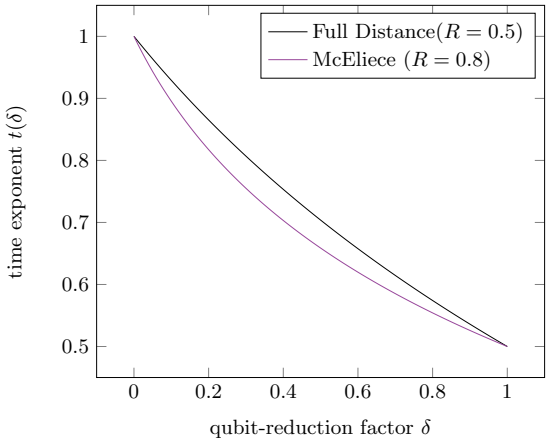


Fig. 5: Time exponent (y-axis) achieved by Theorem 1 for different code parameters plotted as a function of the qubit-reduction factor δ (x-axis).

than drawing a completely new permutation. In our case we fix βn columns of the permutation classically and then search for the remaining $n-k-\beta n$ quantumly. In addition we expect weight p on the fixed βn coordinates, where p has to be optimized.

We again start with a parity-check matrix H in systematic form. Now consider the projection of H onto its first $n-k-\beta n$ rows, we call the resulting matrix $\tilde{H}$. Clearly, a solution $\mathbf{e}$ to the instance $(H, \mathbf{s}, \omega)$ is still a solution to the instance $(\tilde{H}, \mathbf{s}_{[n-k-\beta n]}, \omega)$. Moreover, the matrix $\tilde{H}$ includes βn zero columns, which can safely be removed (compare to Fig. 6). This results in a matrix $\tilde{H}' = (I_{n-k-\beta n} \mid H') \in \mathbb{F}_2^{(n-k-\beta n)\times(1-\beta)n}$ corresponding to a code of length $(1-\beta)n$ and dimension k. Still, by removing the corresponding coordinates from $\mathbf{e}$ we obtain a solution $\mathbf{e}'$ to the instance $(\tilde{H}', \mathbf{s}_{[n-k-\beta n]}, \omega - p)$, where $p := \mathrm{wt}(\mathbf{e}_{[n-k-\beta n+1, n-k]})$ is the weight of coordinates removed from $\mathbf{e}$. Eventually, once $\mathbf{e}'$ is recovered we can obtain $\mathbf{e}$ in polynomial time by solving the respective linear system.

A crucial observation is that disregarding βn parity-check equations could lead to the existence of multiple solutions to the reduced instance, i.e. multiple $\mathbf{e}'$ satisfying $\tilde{H}'\mathbf{e}' = \mathbf{s}_{[n-k-\beta n]}$ but yielding an $\mathbf{e}$ with $\mathrm{wt}(\mathbf{e}) > \omega$. Not that we can control this amount of solutions by increasing p. Also, our algorithm compensates for multiple solutions by recovering all solutions to the reduced instance by repeated executions of the quantum circuit. A pseudocode description of our Punctured-Hybrid trade-off is given by Algorithm 3.

In the following theorem we first state the time complexity of our Punctured-Hybrid in dependence on the qubit reduction factor δ. After this we derive the speedup $t(\delta)$ in a separate corollary.

Theorem 2 (Punctured Hybrid). *Let $n \in \mathbb{N}$, $\omega \in [n]$ and $k = Rn$ for $R \in [\![0, 1]\!]$. Then for any qubit reduction factor $\delta \in [\![0, 1]\!]$ Algorithm 3 solves the*

Algorithm 3 Punctured Hybrid

Require: parity-check matrix $H \in \mathbb{F}_2^{(n-k)\times n}$, syndrome $\mathbf{s} \in \mathbb{F}_2^{n-k}$, weight $\omega \in [n]$, qubit reduction factor $\delta \in [\![0, 1]\!]$
Ensure: error vector $\mathbf{e}$ with $\mathrm{wt}(\mathbf{e}) = \omega$ satisfying $H\mathbf{e} = \mathbf{s}$

choose p accordingly
$\beta := (1-\delta)(1-\frac{k}{n})$, $S := \frac{\binom{(1-\beta)n}{\omega-p}}{2^{(1-\beta)n-k}}$
repeat
  choose random permutation matrix $P \in \mathbb{F}_2^{n\times n}$ and set $\tilde{H} \leftarrow HP$
  transform $\tilde{H}$ to systematic form, $\tilde{H} = \begin{pmatrix} I_{n-k-\beta n} & \mathbf{0} & H_1' \\ \mathbf{0} & I_{\beta n} & H_2' \end{pmatrix}$ with syndrome $\tilde{\mathbf{s}}$
  $\tilde{H}' \leftarrow (I_{n-k-\beta n} \mid H_1')$, $\mathbf{s}' \leftarrow \tilde{\mathbf{s}}_{[(1-\beta)n-k]}$
  for $i = 1$ **to** $\mathrm{poly}(n) \cdot S$ **do**
    solve instance $(\tilde{H}', \mathbf{s}', \omega - p)$ via quantum algorithm returning $\mathbf{e}' \in \mathbb{F}_2^{(1-\beta)n}$
    $\mathbf{e}'' \leftarrow H_2'\mathbf{e}'_{[n-k-\beta n+1,(1-\beta)n]} + \tilde{\mathbf{s}}_{[n-k-\beta n+1,n-k]}$
    if $\mathrm{wt}(\mathbf{e}'') \leq p$ **then**
      $\mathbf{e} \leftarrow P(\mathbf{e}'_{[n-k-\beta n]}, \mathbf{e}'', \mathbf{e}'_{[n-k-\beta n+1,(1-\beta)n]})$
      break
until $H\mathbf{e} = \mathbf{s}$
return $\mathbf{e}$

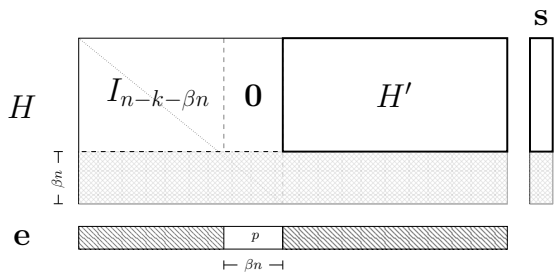


Fig. 6: Parity-check matrix where βn rows are omitted and $\mathbf{e}$ contains weight p on βn coordinates. Framed parts are used as input to the quantum algorithm.

$\mathcal{SD}_{n,k,\omega}$ problem in expected time T_{PH} using $\delta(1-R)Rn^2$ qubits for the matrix representation, where

$$T_{\mathrm{PH}} = \tilde{\mathcal{O}}\left(\frac{\binom{n}{\omega}}{\sqrt{\binom{(1-\beta)n}{\omega-p}\binom{(1-\beta-R)n}{\omega-p}\binom{\beta n}{p}}} \cdot \max\left(1, \sqrt{\binom{(1-\beta)n}{\omega-p} \cdot 2^{-(1-\beta-R)n}}\right)\right)$$

with $\beta = (1-\delta)(1-R)$ and $p \in [\min(\omega, \beta n)]$.

Proof. Assume that the permutation distributes the error weight, such that for $P^{-1}\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e_3}) \in \mathbb{F}_2^{(1-\beta-R)n} \times \mathbb{F}_2^{\beta n} \times \mathbb{F}_2^{Rn}$ it holds $\mathrm{wt}(\mathbf{e}_2) = p$. Now consider the permuted parity-check matrix in systematic form $\tilde{H}$ as given in line 5 of Algorithm 3 with corresponding syndrome $\tilde{\mathbf{s}}$. We obtain

$$\tilde{H}P^{-1}\mathbf{e} = (\mathbf{e}_1 + H_1'\mathbf{e}_3, \mathbf{e}_2 + H_2'\mathbf{e}_3) = \tilde{\mathbf{s}}.$$

This implies that $(\mathbf{e}_1, \mathbf{e}_3)$ is a solution to the syndrome decoding instance $(\tilde{H}', \mathbf{s}', \omega - p)$ with $\tilde{H}' = (I_{(1-\beta-R)n} \mid H_1')$ and $\mathbf{s}' = \tilde{\mathbf{s}}_{[(1-\beta-R)n]}$. The solution is then recovered by the application of our quantum circuit in line 8. Note that in expectation there exist

$$S := \binom{(1-\beta)n}{\omega - p} \cdot 2^{-(1-\beta-R)n}$$

solutions to our reduced instance. Since we apply our quantum circuit $\mathrm{poly}(n) \cdot S$ times and in each execution a random solution is returned, a standard coupon collector argument yields that we recover all S solutions with high probability. Now, when $\mathbf{e}' = (\mathbf{e}_1, \mathbf{e}_3)$ is returned by the quantum circuit, we recover $\mathbf{e}_2 = \tilde{\mathbf{s}}_{[(1-\beta-R)n+1,(1-R)n]} + H_2' \mathbf{e}_3$ and eventually return $\mathbf{e} = P(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$.

Next let us consider the time complexity of the algorithm. Observe that the probability, that $\mathrm{wt}(\mathbf{e}_2) = p$ for a random permutation holds is

$$q_{\mathrm{C}} := \Pr\left[\mathrm{wt}(\mathbf{e}_2) = p\right] = \frac{\binom{(1-\beta)n}{\omega-p}\binom{\beta n}{p}}{\binom{n}{\omega}}.$$

Hence, after q_{C}^{-1} iterations we expect that there is at least one iteration where $\mathrm{wt}(\mathbf{e}_2) = p$. In each iteration we apply our quantum circuit $\tilde{\mathcal{O}}(S)$ times to solve the reduced instance $(\tilde{H}', \mathbf{s}', \omega - p)$, corresponding to a code of length $(1-\beta)n$ and dimension Rn. Since there exist S solutions the expected time to retrieve one of them at random is

$$T_{\mathrm{Q}} = \tilde{\mathcal{O}}\left(\sqrt{\frac{\binom{(1-\beta)n}{\omega-p}}{\max(1, S) \cdot \binom{(1-\beta-R)n}{\omega-p}}}\right),$$

according to Remark 1. The maximum follows since we know that there exists at least one solution. In summary the running time becomes $T_{\mathrm{PH}} = q_{\mathrm{C}}^{-1} \cdot T_{\mathrm{Q}} \cdot \max(1, S)$, as stated in the theorem.

The required amount of qubits of the quantum circuit for solving the syndrome decoding problem related to the reduced code of length $(1-\beta)n$ and dimension $(1-R)n$ are roughly $R(1-\beta-R)n^2$ (compare to Section 3). Thus, for $\beta := (1-\delta)(1-R)$ this corresponds to a qubit reduction of

$$\frac{R(1-\beta-R)}{R(1-R)} = \frac{1-R-(1-\delta)(1-R)}{1-R} = \delta. \qquad \square$$

Theorem 2 allows to easily determine the corresponding speedup, whose exact formula we give in the following corollary.

Corollary 2 (Punctured Hybrid Speedup). *Let $n \in \mathbb{N}$, $\omega = \tau n$ and $k = Rn$, $p = \rho n$ for $\tau, R, \rho \in [\![0, 1]\!]$ and let T_{C} be as defined in Eq. (4). Then for any qubit reduction factor $\delta \in [\![0, 1]\!]$ Algorithm 3 solves the $\mathcal{SD}_{n,k,\omega}$ problem in time $(T_{\mathrm{C}})^{t(\delta)}$ using $\delta(1-R)Rn^2$ qubits for the matrix representation, where*

$$t(\delta) = \frac{\mathrm{H}(\tau) - \beta \mathrm{H}\left(\frac{\rho}{\beta}\right) - \frac{1-\beta}{2} \cdot \mathrm{H}\left(\frac{\tau-\rho}{1-\beta}\right) - \frac{(1-\beta-R)}{2} \cdot \mathrm{H}\left(\frac{\tau-\rho}{1-\beta-R}\right) + \max(0, \sigma)}{\mathrm{H}(\tau) - (1-R)\mathrm{H}\left(\frac{\tau}{1-R}\right)}$$

for $\beta = (1-\delta)(1-R)$ *and* $\sigma = (1-\beta)\mathrm{H}\left(\frac{\tau-\rho}{1-\beta}\right) - (1-\beta-R)$.

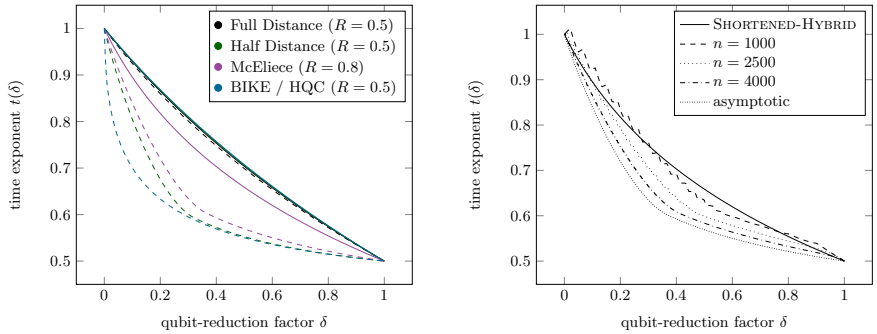
Proof. Recall that $t(\delta) = \frac{\log T_{\mathrm{PH}}}{\log T_{\mathrm{C}}}$, where T_{PH} is the running time of Algorithm 3, given in Theorem 2. Now the statement of the corollary follows immediately by approximating the binomial coefficients in T_{PH} and T_{C} via Stirling's formula (see Eq. (1)). □

In Fig. 7a we compare the behavior of our new trade-off to our previously obtained Shortened-Hybrid. Recall that the performance of Shortened-Hybrid is not very sensitive to changes in the error-rate. Thus, for settings with the same code-rate, i.e. full and half distance as well as BIKE / HQC, the solid lines are almost on top of each other. The dashed lines represent our new trade-off (Theorem 2) for which we optimized p numerically. It can be observed, that this trade-off outperforms the Shortened-Hybrid for all parameters. Here, we observe the best behaviour for low code-rates and small error-rates, which correspond to the case, where the solution is very unique. In these cases our Punctured-Hybrid algorithm can disregard parity-check equations without introducing multiple solutions to the reduced instance. Hence, still a single execution of the quantum circuit suffices to recover the solution. The significance of the amount of solutions can be well observed by comparing the full and half distance settings. In the full distance setting there exists already one random solution in expectation, therefore any omitted parity equation leads to the existence of multiple solution and in turn leads to only a small improvement over Shortened-Hybrid. Contrary, the half distance setting allows for a significant improvement, which is due to the exponentially small probability of existing random solutions. Note that in the McEliece, BIKE and HQC setting the error weight is only sublinear, which lies in favour of our new trade-off, since the probability for existing random solutions is again exponentially small. BIKE and HQC furthermore use a very small error weight of only $\mathcal{O}(\sqrt{n})$ and specify a code-rate of $R = 0.5$, which results in a very unique solution. Consequently, in Fig. 7a it can be observed, that asymptotically for these settings the second trade-off improves drastically on Shortened-Hybrid.

Note that our formulation of the speedup for Punctured-Hybrid in contrast to Shortened-Hybrid (see Corollary 1) still depends on the error-rate, not exactly allowing for $\omega = o(n)$. Thus, to obtain the asymptotic plot we compared the result of Corollary 1 to Theorem 2 for McEliece ($n = 6688, k = 5024, \omega = 128$), BIKE ($n = 81946, k = 40973, \omega = 264$) and HQC ($n = 115274, k = 57637, \omega = 262$), which are the suggested parameters for 256-bit security from the corresponding NIST submission documentations [2, 10, 24].

To quantify the result of our new trade-off take e.g. the case of McEliece and a qubit reduction by 80% ($\delta = 0.2$), as before. Here we improve to a speedup of $t(\delta) \approx 0.74$, compared to 0.82 for Shortened-Hybrid.

However, for concrete medium sized parameters this asymptotic behaviour is not necessarily obtained. In Fig. 7b we show a comparison of both trade-offs for concrete McEliece parameter sets. Note that for all parameter sets the performance of Shortened-Hybrid is almost identical, which is why there is only a single solid line.



(a) Asymptotically achieved time exponents. New Theorem 2 (Punctured-Hybrid) depicted as dashed lines, Theorem 1 (Shortened-Hybrid) as solid lines.

(b) Time exponents for concrete parameter sets. McEliece parameter sets satisfy $k = 0.8n$ and $\omega = \left\lfloor \frac{n}{5 \log n} \right\rfloor$. Non-solid lines correspond to Punctured-Hybrid.

Fig. 7: Comparison of time exponents of Shortened-Hybrid and Punctured-Hybrid (y-axis) plotted as a function of the qubit-reduction factor δ (x-axis).

For these concrete computations we used the more accurate time complexity formula involving binomial coefficients rather than its asymptotic approximation to determine the speedup $t(\delta)$. Note that the discontinuity for our new trade-off is due to the restriction to discrete choices of p. We find that for parameters up to $n \approx 2500$ both trade-offs remain superior to each other for certain reduction factors δ. For larger values of n the Punctured-Hybrid algorithm becomes favourable for all δ.

In the BIKE and HQC settings the Punctured-Hybrid algorithm is favourable already for small parameters corresponding to $n \geq 1000$.

4.3 Combined Hybrid

Next let us outline how to combine both previous trade-offs to achieve an improved version. We first reduce the code length and dimension, again by guessing αn zero coordinates of $\mathbf{e}$ and removing the corresponding columns form H, i.e., we shorten the code. The remaining instance is then solved using our Punctured-Hybrid algorithm, i.e., by first omitting βn parity-check equations (compare also to Fig. 8) and then using the reduced instance as input to the quantum circuit.

We give the pseudocode description of the procedure in Algorithm 4. Note that here we use β and p as input parameters to Punctured-Hybrid, rather than to the choice made in Algorithm 3 (Punctured-Hybrid). Further, since for an incorrect guess of αn zero positions the call to Punctured-Hybrid will not finish,, we introduce an abort after the expected amount of iterations on a correct guess.

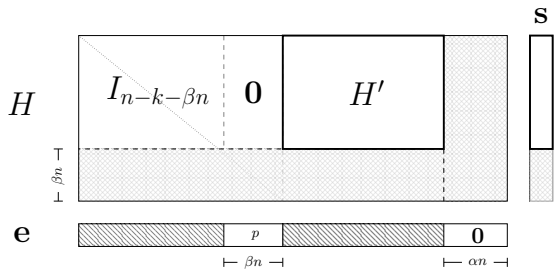


Fig. 8: Input matrix in systematic form where βn parity-check equations are omitted and αn zeros of $\mathbf{e}$ are known. The vector $\mathbf{e}$ is assumed to contain weight p on βn coordinates. Framed parts are used as input to the quantum algorithm.

Algorithm 4 COMBINED-HYBRID

Require: parity-check matrix $H \in \mathbb{F}_2^{(n-k)\times n}$, syndrome $\mathbf{s} \in \mathbb{F}_2^{n-k}$, weight $\omega \in [n]$, qubit reduction factor $\delta \in [\![0,1]\!]$

Ensure: error vector $\mathbf{e}$ with $\mathrm{wt}(\mathbf{e}) = \omega$ satisfying $H\mathbf{e} = \mathbf{s}$

choose α and p accordingly

$\beta := (1 - \frac{k}{n}) \left(\frac{\delta \frac{k}{n}}{\frac{k}{n} - \alpha} \right)$, $E := \frac{\binom{(1-\alpha)n}{\omega}}{\binom{(1-\alpha-\beta)n}{\omega-p}\binom{\beta n}{p}}$

repeat

  choose random permutation matrix $P \in \mathbb{F}_2^{n\times n}$ and set $\tilde{H} \leftarrow HP$

  $\mathbf{e}' \leftarrow$ PUNCTURED-HYBRID$(\tilde{H}_{[(1-\alpha)n]}, \mathbf{s}, \omega, \delta, \frac{\beta}{1-\alpha}, p)$ ▷ abort after E iterations of the outer loop

  $\mathbf{e} \leftarrow P(\mathbf{e}', 0^{\alpha n})$

until $H\mathbf{e} = \mathbf{s}$

return $\mathbf{e}$

Theorem 3 (Combined Hybrid). *Let $n \in \mathbb{N}$, $\omega \in [n]$ and $k = Rn$ for $R \in [\![0,1]\!]$. Then for any qubit reduction factor $\delta \in [\![0,1]\!]$ the $\mathcal{SD}_{n,k,\omega}$ problem can be solved in expected time T_{CH} using $\delta(1-R)Rn^2$ qubits for the matrix representation, where*

$$T_{\mathrm{CH}} = \tilde{\mathcal{O}}\left(\frac{\binom{n}{\omega}}{\sqrt{\binom{(1-\alpha-\beta)n}{\omega-p}\binom{(1-\beta-R)n}{\omega-p}\binom{\beta n}{p}}} \cdot \max\left(1, \sqrt{\binom{(1-\alpha-\beta)n}{\omega-p} \cdot 2^{-(1-\beta-R)n}}\right)\right)$$

with $\alpha \in [\![0, R]\!]$, $\beta = (1-R)\left(1 - \frac{\delta R}{R-\alpha}\right)$ and $p \in [\min(\omega, \beta n)]$.

Proof. The correctness follows from the correctness of Algorithm 2 and Algorithm 3. Therefore observe that for a correct guess of αn zero positions of $\mathbf{e}$, the expected amount of permutations needed by PUNCTURED-HYBRID to find the solution is

$$E := \frac{\binom{(1-\alpha)n}{\omega}}{\binom{(1-\alpha-\beta)n}{\omega-p}\binom{\beta n}{p}}.$$

Also note that PUNCTURED-HYBRID is called on a code of length $n' = (1-\alpha)n$. Hence, setting $\beta' = \frac{\beta}{1-\alpha}$ guarantees that $\beta' n' = \beta n$ parity equations are omitted.

For the time complexity we have again with probability

$$q_{\mathrm{C}} := \Pr\left[P^{-1}\mathbf{e} = (\mathbf{e}_1, 0^{\alpha n})\right] = \frac{\binom{(1-\alpha)n}{\omega}}{\binom{n}{\omega}},$$

a correct guess for αn zero positions (compare to the proof of Theorem 1). In each iteration of our combined algorithm we call the Punctured-Hybrid algorithm. Inside this subroutine E iterations of the outer loop are executed, each performing

$$S = \tilde{\Theta}\left(\max\left(1, \frac{\binom{1-\beta-\alpha}{\omega-p}}{2^{-(1-R-\beta)n}}\right)\right)$$

calls to the quantum circuit. This quantum circuit is applied to solve the syndrome decoding problem defined on a code of length $(1-\alpha-\beta)n$ and dimens $(R-\alpha)n$ with error-weight $\omega - p$ (compare to Fig. 8), which takes time

$$T_{\mathrm{Q}} = \tilde{\mathcal{O}}\left(\sqrt{\frac{\binom{(1-\alpha-\beta)n}{\omega-p}}{S \cdot \binom{(1-\beta-R)n}{\omega-p}}}\right).$$

Thus, eventually, the time complexity of the whole algorithm summarizes as $T_{\mathrm{CH}} = q_{\mathrm{C}}^{-1} \cdot E \cdot T_{\mathrm{Q}} \cdot S$, as claimed. Finally, note that for given $\beta = (1-R)\left(1 - \frac{\delta R}{R-\alpha}\right)$ we obtain a qubit reduction by

$$\frac{(R-\alpha)(1-R-\beta)}{R(1-R)} = \frac{(R-\alpha)(1-R)\left(1-(1-\frac{\delta R}{R-\alpha})\right)}{R(1-R)} = \frac{(R-\alpha)\cdot\frac{\delta R}{R-\alpha}}{R} = \delta. \quad \square$$

Our combination achieves the improved trade-off behavior depicted as dashed lines in Fig. 9. Here the values of p and α were optimized numerically. It shows that the combination of both trade-offs for most parameters improves on Punctured-Hybrid (solid lines). Especially in the full distance decoding setting an improvement for nearly all δ is achieved. This is due to the fact, that the guessing of zero coordinates is an additional possibility to control the amount of solutions to the reduced instance and therefore to optimize the complexity of the Punctured-Hybrid subroutine. This is also the reason why we achieve no (asymptotic) improvement in the BIKE and HQC settings, here the solution is already so unique that the trade-off can not benefit from the new degree of freedom.

But also in the McEliece setting we achieve notable improvements. If we again consider a reduction-factor of $\delta = 0.2$ the combination improves the speedup to $t(\delta) \approx 0.69$ from 0.74 achieved by Punctured-Hybrid. Furthermore, when focusing on near future realizations, i.e., the regime of small reduction factors, it is for example possible with just one percent of the qubits ($\delta = 0.01$) to achieve a speedup of $t(\delta) \approx 0.92$.

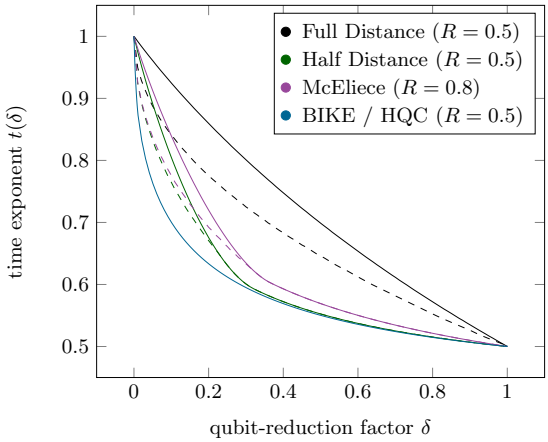


Fig. 9: Asymptotically achieved time exponents. The combined trade-off is depicted as dashed line, PUNCTURED-HYBRIDis illustrated as a solid line.

References

1. Alagic, G., Alperin-Sheriff, J., Apon, D., Cooper, D., Dang, Q., Kelsey, J., Liu, Y.K., Miller, C., Moody, D., Peralta, R., et al.: Status report on the second round of the NIST post-quantum cryptography standardization process. US Department of Commerce, NIST (2020)
2. Aragon, N., Barreto, P., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.C., Gaborit, P., Gueron, S., Guneysu, T., Melchor, C.A., et al.: BIKE: bit flipping key encapsulation (2020)
3. Bärtschi, A., Eidenbenz, S.: Deterministic preparation of Dicke states. In: International Symposium on Fundamentals of Computation Theory. pp. 126–139. Springer (2019)
4. Becker, A., Joux, A., May, A., Meurer, A.: Decoding random binary linear codes in $2^{n/20}$: How 1+1=0 improves information set decoding. In: Annual international conference on the theory and applications of cryptographic techniques. pp. 520–536. Springer (2012)
5. Bernstein, D.J.: Grover vs. McEliece. In: International Workshop on Post-Quantum Cryptography. pp. 73–80. Springer (2010)
6. Biasse, J.F., Bonnetain, X., Pring, B., Schrottenloher, A., Youmans, W.: A trade-off between classical and quantum circuit size for an attack against CSIDH. Journal of Mathematical Cryptology **15**(1), 4–17 (2020)
7. Biasse, J.F., Pring, B.: A framework for reducing the overhead of the quantum oracle for use with Grover's algorithm with applications to cryptanalysis of SIKE. Journal of Mathematical Cryptology **15**(1), 143–156 (2020)
8. Both, L., May, A.: Decoding linear codes with high error rate and its impact for LPN security. In: International Conference on Post-Quantum Cryptography. pp. 25–46. Springer (2018)
9. Canteaut, A., Chabaud, F.: A new algorithm for finding minimum-weight words in a linear code: application to McEliece's cryptosystem and to narrow-sense BCH

codes of length 511. IEEE Transactions on Information Theory **44**(1), 367–378 (1998)
10. Chou, T., Cid, C., UiB, S., Gilcher, J., Lange, T., Maram, V., Misoczki, R., Niederhagen, R., Paterson, K.G., Persichetti, E., et al.: Classic mceliece: conservative code-based cryptography 10 october 2020 (2020)
11. Dumer, I.: On minimum distance decoding of linear codes. In: Proc. 5th Joint Soviet-Swedish Int. Workshop Inform. Theory. pp. 50–52 (1991)
12. Efthymiou, S., Ramos-Calderer, S., Bravo-Prieto, C., Pérez-Salinas, A., García-Martín, D., Garcia-Saez, A., Latorre, J.I., Carrazza, S.: Qibo: a framework for quantum simulation with hardware acceleration. arXiv preprint arXiv:2009.01845 (2020)
13. Efthymiou, S., Ramos-Calderer, S., Bravo-Prieto, C., Pérez-Salinas, A., García-Martín, D., Garcia-Saez, A., Latorre, J.I., Carrazza, S.: Quantum-tii/qibo: Qibo (Aug 2020). https://doi.org/10.5281/zenodo.3997195, https://doi.org/10.5281/zenodo.3997195
14. Esser, A., Bellini, E.: Syndrome decoding estimator. In: IACR International Conference on Public-Key Cryptography. pp. 112–141. Springer (2022)
15. Esser, A., May, A., Verbel, J., Wen, W.: Partial key exposure attacks on BIKE, Rainbow and NTRU. Cryptology ePrint Archive (2022)
16. Esser, A., Ramos-Calderer, S., Bellini, E., Latorre, J.I., Manzano, M.: An optimized quantum implementation of ISD on scalable quantum resources. arXiv preprint arXiv:2112.06157 (2021)
17. Gilbert, E.N.: A comparison of signalling alphabets. The Bell system technical journal **31**(3), 504–522 (1952)
18. Jaques, S., Naehrig, M., Roetteler, M., Virdia, F.: Implementing grover oracles for quantum key search on AES and LowMC. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 280–310. Springer (2020)
19. Kachigar, G., Tillich, J.P.: Quantum information set decoding algorithms. In: International Workshop on Post-Quantum Cryptography. pp. 69–89. Springer (2017)
20. Kirshanova, E.: Improved quantum information set decoding. In: International Conference on Post-Quantum Cryptography. pp. 507–527. Springer (2018)
21. Lee, P.J., Brickell, E.F.: An observation on the security of McEliece's public-key cryptosystem. In: Workshop on the Theory and Application of of Cryptographic Techniques. pp. 275–280. Springer (1988)
22. May, A., Meurer, A., Thomae, E.: Decoding random linear codes in $\tilde{\mathcal{O}}(2^{0.054n})$. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 107–124. Springer (2011)
23. May, A., Ozerov, I.: On computing nearest neighbors with applications to decoding of binary linear codes. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 203–228. Springer (2015)
24. Melchor, C.A., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.C., Gaborit, P., Persichetti, E., Zémor, G., Bourges, I.: Hamming quasi-cyclic (HQC) (2020)
25. Nielsen, M.A., Chuang, I.L.: Quantum information and quantum computation. Cambridge: Cambridge University Press **2**(8), 23 (2000)
26. Perriello, S., Barenghi, A., Pelosi, G.: A quantum circuit to speed-up the cryptanalysis of code-based cryptosystems. In: International Conference on Security and Privacy in Communication Systems. pp. 458–474. Springer (2021)
27. Prange, E.: The use of information sets in decoding cyclic codes. IRE Transactions on Information Theory **8**(5), 5–9 (1962)

28. Stern, J.: A method for finding codewords of small weight. In: International Colloquium on Coding Theory and Applications. pp. 106–113. Springer (1988)
29. Varshamov, R.R.: Estimate of the number of signals in error correcting codes. Docklady Akad. Nauk, SSSR **117**, 739–741 (1957)
30. Zalka, C.: Grover's quantum searching algorithm is optimal. Physical Review A **60**(4), 2746 (1999)