

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us what having access to this work means to you and why it's important to you. Thank you.

Optimal Entropy Compression and Purification in Quantum Bits

Varad R. Pande*

*Department of Physics, University of Maryland Baltimore County (UMBC), Baltimore, MD 21250, USA,
Department of Physics, Indian Institute of Science Education and Research (IISER) Pune 411008, Maharashtra, India, and
Harish-Chandra Research Institute (HRI), HBNI, Jhansi,
Prayagraj (Allahabad) 211019, Uttar Pradesh, India.*

(Dated: January 3, 2020)

Global unitary transformations (OPTSWAPS) that optimally increase the bias of any mixed computation qubit in a quantum system – represented by a diagonal density matrix – towards a particular state of the computational basis which, in effect, increases its purity are presented. Quantum circuits that achieve this by implementing the above data compression technique – a generalization of the 3B-Comp [Fernandez, Lloyd, Mor, Roychowdhury (2004); arXiv: quant-ph/0401135] used before – are described. These circuits enable purity increment in the computation qubit by maximally transferring part of its von Neumann or Shannon entropy to any number of surrounding qubits and are valid for the complete range of initial biases. Using the optswaps, a practicable new method that algorithmically achieves hierarchy-dependent cooling of qubits to their respective limits in an engineered quantum register opened to the heat-bath is delineated. In addition to multi-qubit purification and satisfying two of DiVincenzo’s criteria for quantum computation in some architectures, the implications of this work for quantum data compression are discussed.

I. MOTIVATION

The advent of quantum information theory has resulted in the development of information processing using quantum computers which employ quantum matter and manipulate it according to the rules of quantum mechanics. Certain criteria need to be satisfied for the physical realization of a quantum computer [1]. Two of these are: (i) initialization of computation quantum bits (or qubits) in a well-defined quantum state, and (ii) error correction to tackle environmental decoherence during information processing. Since continuous supply of pure qubits is required for error-correction, the methods relevant to satisfy the former are indirectly necessary for the latter [2]. One such method for qubit initialization is heat-bath algorithmic cooling (HBAC). In algorithmic cooling (quantified using the definition of spin temperature [3]), we purify or increase the bias of a required number of qubits to make them available for quantum information processing. At the heart of this procedure lies the transfer of entropy from the computation qubits to the reset qubits followed by exposing the reset qubits to a heat-bath which sucks the excess entropy out of them so that the entropy transfer can be repeated.

We first introduce closed-system entropy transfers. The bound [4] on such transfers can be obtained by considering n qubits, say each with equal bias ϵ (or purity $(1 + \epsilon^2)/2$) in the computational basis ($\rho = \{ \{(1 + \epsilon)/2, 0\}, \{0, (1 - \epsilon)/2\} \}$) and total Shannon entropy $H_t = nH$, where $H = [(1 + \epsilon)/2] \log((1 + \epsilon)/2) + [(1 - \epsilon)/2] \log((1 - \epsilon)/2)$. After the (hypothetical) entropy compression, let m qubits be in a pure state so that their Shannon entropy is zero. Consequently, the $n - m$ qubits will each have an entropy H' such that the

total is conserved and we have $(n - m) * H' = n * H$ leading to $H' = (n/(n - m)) * H$. Since the qubits are hotter than before, but not infinitely hot (maximally mixed), we get $H < H' \leq 1$, which leads to a bound on the number of qubits that can be completely purified: $m \leq n(1 - H)$. The bound on entropy compression for an $n + 1$ equal-bias system, obtained by Taylor expanding [5] H to first order in ϵ is also equivalent to the one obtained by conservation of purity (or of “spin order” [6]) and is given by $\epsilon' = \sqrt{n} * \epsilon$, where ϵ and ϵ' are initial and final single target qubit biases respectively with the excess entropy being dumped to the n qubits.

Now we turn to a bound tighter than the entropy bound based on fundamental properties of normal matrices. Density matrices representing quantum states are Hermitian, which are a subset of normal matrices and are therefore diagonalizable by a unitary transformation. The purity of a density matrix ($\text{Tr}(\rho^2)$) does not change under a unitary transformation (the diagonalization). So the ensuing discussion about the possible increase in purity (or bias, once diagonalized) of the target qubit applies most generally. We use the fact that eigenvalues of a Hermitian (more generally, normal) matrix are invariant under any unitary transformation. Therefore, any possible operation to increase the bias of a single target qubit (say) is bound by *exchanges* of diagonal elements (the eigenvalues) of the global (multiqubit) density matrix such that the largest eigenvalues lie in the first half of the resultant density matrix and the smallest ones lie in the bottom half. Using this fact, an analytical bound arises automatically if one arranges all the eigenvalues of the initial state in descending order and require a part of the final state to be such that the purity (or bias) of the target qubit is 1 and the purity of all the rest of the qubits is zero. Let the initial density matrix be ρ_i and the final density matrix be $\rho_f = \alpha\rho_{f1} + \beta\rho_{f2}$, where ρ_{f1} represents part of the Hilbert space which satisfies

* varadrpande@gmail.com

the aforementioned purity requirement. Then, we have $\alpha = \text{Tr}(\rho_i \rho_{f1}) / \text{Tr}(\rho_{f1}^2)$, and the final bias of the target qubit is given by $\epsilon' = \alpha - \beta$ (note that β can be obtained from the conservation of trace with respect to ρ_i). *Sorensen* discovered this fundamentally inviolable bound to be smaller than the entropy bound for special systems where there are qubits with only two different biases of the order of 10^{-5} : the $I_n S$ [6, 7] or the $I_n S_m$ [8] spin systems. Here, it was also found that the entropy bound violates eigenvalue invariance and is therefore automatically disproved. While this treatment neatly tells us what the closed system bound could be for these simple spin systems, it does not tell what those purity-maximizing exchanges are or how to implement them. Ref. [6] was able to do this for the special cases in the nuclear magnetic resonance (NMR) architecture up to 7-qubit systems. Star-topology systems in NMR allow us to do this for systems as large as 37 qubits [3]; in this case several of the diagonal elements bunch together in degenerate energy levels making the exchanges easier.

However, when the default qubit biases (purities) are all unequal, single shot optimal compression is inspired by data compression techniques. In particular, *Schulman and Vazirani* [9] used the reverse of von Neumann’s method of extracting fair coin flips from a biased coin [10]: apply a C-NOT gate on successive pairs of qubits and keep the control qubit conditioned on the measurement outcome of the NOT qubit being 0, resulting in a boosting of its bias; follow this up by segregation of hot and cold qubits. As such, this is a non-unitary method. A single-shot maximally compressive unitary method for purification of the target qubit when three qubits have different biases employs the 3B-Comp [5, 11–22] (see Fig. 1), a data compression circuit introduced and implemented by *Chang et. al.* [23] and identified in its current form by *Fernandez et. al.* [24]. Also, the bonacci series of algorithms [13, 25–27] perform compression by swapping the bottom element in top half of the global density matrix with the top element in bottom half (see discussion above); the k-bonacci does this for density matrices of k successive qubits while the all-bonacci does it for all qubits below the target. So, the first purpose of this paper is to present optimal entropy-compressive unitary transformations (the aforementioned exchanges, *optswaps*) and the quantum circuits to implement those transformations when the initial biases in any number of qubits could be all different and could take any value between 0 and 1. We shall call the quantum circuits as NB-MaxComp to distinguish them from the “NB-Comp” unitaries already used by *Elias et. al.* [13] for the bonacci series of algorithms. We shall provide the circuit LIM-Comp for implementing the “NB-Comp” unitaries as well.

In order to achieve increase in biases beyond the closed-system entropy compression bounds, we need to open the system to a heat-bath which acts as a constant sink of excess entropy by being in contact with some of the qubits involved in the closed system transfer. This was first

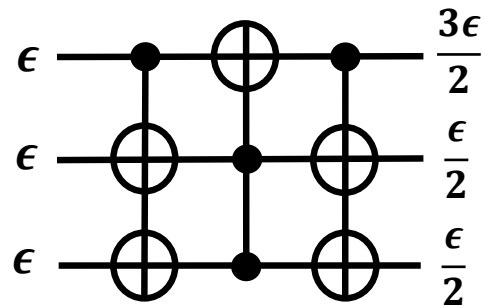


FIG. 1. The 3B-Comp data compression circuit implements the transformation $|011\rangle \leftrightarrow |100\rangle$.

done by *Boykin et. al.* [4] who used the same primitive as Ref. [9] and converted it to an open system method – termed heat-bath algorithmic cooling (HBAC) – to achieve theoretically better cooling with lesser qubits. The question of finding the ultimate limit of cooling a single qubit with any such open-system method was then addressed by *Schulman et. al.* [28, 29], who proposed the partner pairing algorithm (PPA) to achieve that bound. The PPA consists of a SORT step for closed-system entropy compression which sorts diagonal elements in a descending order and a RESET step for refreshing the hotter qubits by swapping their biases with a RESET-qubit in contact with the heat-bath. The limit was proven to be $2^{n-2}\epsilon \leq \epsilon_f \leq 2^{n-1}\epsilon$, where ϵ_f is the final bias of the target qubit [27–29]. Under the approximation that initial biases $\epsilon \ll 1/2^n$ (n is total number of qubits participating in the SORT step), the limiting bias of a single qubit was found to be $2^{n-2}\epsilon$ for the PPA, and the all-bonacci algorithm [13, 27]. The former was proven by *Raeisi and Mosca* by arriving at an optimal asymptotic state (OAS) which is invariant under PPA. The exact HBAC bound was analytically proven using a PPA steady state analogous to the OAS by *Rodriguez-Briones and Laflamme* by using m reset qubits instead of 1, which reduces to the aforementioned low ϵ limit (after substituting $m = 1$). However, exact dynamics of the PPA and method to find transformations needed to implement the sort step have proven elusive [13, 19, 30, 31]. Therefore, the second purpose of this paper is to present a HBAC method distinct from the PPA, but one which leads to the appropriate exact limits. This method is operationally systematic in that it leads all the qubits within a quantum register to their respective multi-round limits – something that gives a high degree of control while purifying the register – by telling us which unitary transformations to do at what stage of the process. Further, the explicit nature of the code (algorithm) allows us to consider cases where all qubits could have different initial (or default) biases, for example because the qubits see different local environments or are acted upon by a different number of heat-causing quantum operations.

In section II, we shall find the optswaps resulting from

a general compression subroutine, conjecture their optimality in section II A, find a numerical proof of these transformations in section II B, and quantum circuits for implementing these unitary transformations in section III. Then we work on open-system compression in section IV where we introduce the limiting swap in section IV A, analytically derive the multi-round limit for hierarchical cooling of a multiqubit quantum register in section IV B, find the algorithm which provides numerical limits supplementing the above and generalizes to the case where the qubits have different default biases in section IV C, and build the register compression subroutine (RCS) characterizing the dynamics of HBAC and leading to the multi-round limits in section IV D. Finally, we shall discuss the complexity of RCS-HBAC and the NB-MaxComp in section V and end with a discussion of this work in section VI.

II. GENERAL COMPRESSION SUBROUTINE

Consider an array of qubits in a quantum register. Generally these qubits would be in a mixed state diagonal in some “natural” basis. Thus, the state of the i^{th} computation qubit in the register is,

$$\rho_i = \begin{pmatrix} \frac{1+\epsilon_i}{2} & 0 \\ 0 & \frac{1-\epsilon_i}{2} \end{pmatrix} = \begin{pmatrix} p_i & 0 \\ 0 & 1-p_i \end{pmatrix} \quad (1)$$

where $p_i \in (1/2, 1)$ maps to $\epsilon_i \in (0, 1)$ and its Shannon entropy is given by $H_i = (1 + \epsilon_i)/2 \log((1 + \epsilon_i)/2) + (1 - \epsilon_i)/2 \log((1 - \epsilon_i)/2)$. Hereafter, we shall denote $(1 + \epsilon_i)/2 \equiv \epsilon_i^+$ and $(1 - \epsilon_i)/2 \equiv \epsilon_i^-$. The object of algorithmic cooling is to compress entropy out from some of the qubits in the register to the remaining qubits by increasing the bias (ϵ_i) or probability (p_i) of the computation qubits towards state $|0\rangle$. This is akin to increasing their purity $\text{Tr}(\rho_i^2)$.

For a given array of qubits, we seek to increase the bias of the TARGET qubit towards the state $|0\rangle$ at the expense of decreasing the corresponding bias of the ancilla qubits. Therefore, the probability amplitudes (PROBAMPS) of the state can be divided into two subspaces – one corresponding to density matrix diagonal elements (DIAG-ELS) where the TARGET qubit is $|0\rangle$ and the other corresponding to the DIAG-ELS where the target qubit is $|1\rangle$. Hereafter, we will call them the $|0T\rangle$ and the $|1T\rangle$ subspaces respectively (see pictorial representation in Fig. 2). Thus, the system of n qubits, each represented by Eq. 1, can be expressed as:

$$|\psi_t\rangle = \bigotimes_{i=1}^n |\psi_i\rangle = \sum_{j=0}^{2^n-1} \prod_{\{k_j\}} \prod_{\{l_j\}} \epsilon_{k_j}^+ \epsilon_{l_j}^- |j\rangle, \quad (2)$$

where j is the decimal number corresponding to the respective DIAG-ELS. k_j and l_j index 0's and 1's respectively in the n -bit binary equivalent of j . Thus, $k_j, l_j \in \{1, \dots, n\}$, $\{k_j\} \cup \{l_j\} = \{1, \dots, n\}$ and $\{k_j\} \cap \{l_j\} = \emptyset$. For

the sake of brevity, in the above and subsequent equations, we set $|j\rangle \langle j| \equiv |j\rangle$, $\rho_t \equiv |\psi_t\rangle$, and $\rho_i \equiv |\psi_i\rangle$. Without loss of generality, we choose to cool the first qubit. The subspace division can be expressed as:

$$|\psi_{0T}\rangle = \sum_{j=0}^{2^{n-1}-1} \epsilon_1^+ \prod_{\{k_j\}} \prod_{\{l_j\}} \epsilon_{k_j}^+ \epsilon_{l_j}^- |j\rangle,$$

$$|\psi_{1T}\rangle = \sum_{j=2^{n-1}}^{2^n-1} \epsilon_1^- \prod_{\{k_j\}} \prod_{\{l_j\}} \epsilon_{k_j}^+ \epsilon_{l_j}^- |j\rangle.$$

The exchange of particular PROBAMPS between these two subspaces using certain entropy compressive unitary transformations (OPTSWAPS) is the building block of the subroutine. The OPTSWAPS conform to the following prescription:

Definition 1. Let the set of unitary transformations $J \equiv \{V_j\}$, where $V_j \equiv |j\rangle_{0T} \leftrightarrow |2^n - 1 - j\rangle_{1T} \forall j \in \{0, \dots, 2^{n-1} - 1\}$. Then $V_j \in J_\theta \subseteq J$ if and only if $R_{j,0T} < R_{j,1T}$, where $R_{j,0T} \equiv (\epsilon_1^+ \prod_{\{k_j\}} \prod_{\{l_j\}} \epsilon_{k_j}^+ \epsilon_{l_j}^-)_{0T}$ and $R_{j,1T} \equiv (\epsilon_1^- \prod_{\{k_j\}} \prod_{\{l_j\}} \epsilon_{k_j}^+ \epsilon_{l_j}^-)_{1T}$.

For example, in a 5-qubit system, if $\epsilon_1^+ \epsilon_2^- \epsilon_3^+ \epsilon_4^- \epsilon_5^- < \epsilon_1^- \epsilon_2^+ \epsilon_3^- \epsilon_4^+ \epsilon_5^+$, then the corresponding OPTSWAP is $|0_1 1_2 0_3 1_4 1_5\rangle \leftrightarrow |1_1 0_2 1_3 0_4 0_5\rangle$ or simply $|01011\rangle \leftrightarrow |10100\rangle$. In this case, $\{k_j\} = \{2, 4, 5\}$ and $\{l_j\} = \{1, 3\}$. In terms of decimals it can be simply represented as: $|11\rangle \leftrightarrow |20\rangle$. Upon performing all the OPTSWAPS that satisfy the above prescription and denoted by the set $J_\theta = \{j_c\}$, the increase in bias of the TARGET is given by $X_n = \epsilon'_1 - \epsilon_1 = 2 \sum_{\{j_c\}} (R_{j_c,1T} - R_{j_c,0T})$

$$= 2 \sum_{\{j_c\}} \left(\frac{1}{2^n} \sum_m \sum_{A_m, B_m} (-1)^{\sum_{B_m} k_{j_c m} \bmod (k_{j_c m} - 1)} \prod_{A_m} \epsilon_{l_{j_c m}} \prod_{B_m} \epsilon_{k_{j_c m}} \right), \quad (3)$$

where, $A_m \equiv \{l_{j_c m}\}$, $B_m \equiv \{k_{j_c m}\}$, and $m = \text{card}(A_m \cup B_m) \in 2\mathbb{N} + 1$. Also, $l_{j_c m}, k_{j_c m} \in \mathcal{C}(A_m \cup B_m)$, where \mathcal{C} denotes the set of all possible combinations of elements in $A_m \cup B_m$. This expression is derived by making empirical observations for some examples.

A. Optimality

The optimality of the OPTSWAPS for increasing the bias of the TARGET is based on Def. 1. The following theorem is based on ruling out all the non-complementary swaps (those which are violative of Def. 1) by establishing the fact that they are suboptimal for our task:

Theorem 1. If

$$1. R_{a,1T} - R_{a,0T} \geq S_{b,1T} - R_{a,0T} \forall a \in J_\theta \text{ and } \forall b \in J \setminus J_\theta, \text{ and}$$

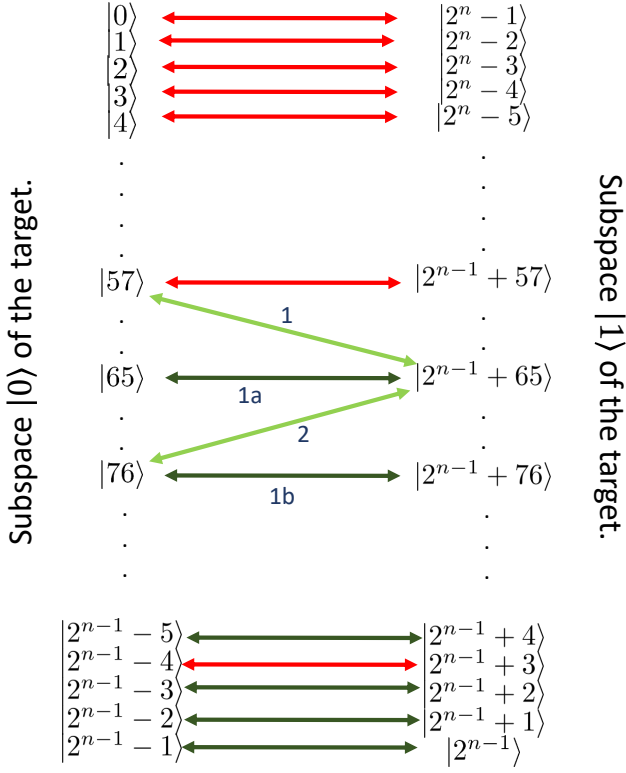


FIG. 2. Graphical representation of Theorem. 1. All double-edged arrows represent complementary swaps. Dark green arrows are OPTSWAPS. 1 and 2 are examples of beneficial non-complementary swaps which are ruled out by statements 1 and 2 of Theorem 1 respectively.

2. given $R = R_{a,1T} - R_{a,0T} > 0$ and $S = S_{b,1T} - S_{b,0T} > 0$, $S_{b,1T} - R_{a,0T} < R + S$, $\forall a, b \in J_\theta$ such that $a \neq b$, and

3. given $J_\theta = \emptyset$, $S_{b,1T} - R_{a,0T} < 0$, $\forall a, b \in J$,

then X_n is maximal.

where $R_{a,1T} \equiv (\epsilon_1^- \prod_{\{l_a\}} \prod_{\{k_a\}} \epsilon_{k_a}^- \epsilon_{l_a}^+)_{1T}$, $R_{a,0T} \equiv (\epsilon_1^+ \prod_{\{l_a\}} \prod_{\{k_a\}} \epsilon_{k_a}^+ \epsilon_{l_a}^-)_{0T}$, $S_{b,0T} \equiv (\epsilon_1^+ \prod_{\{l_b\}} \prod_{\{k_b\}} \epsilon_{k_b}^+ \epsilon_{l_b}^-)_{0T}$ and $S_{b,1T} \equiv (\epsilon_1^- \prod_{\{l_b\}} \prod_{\{k_b\}} \epsilon_{k_b}^- \epsilon_{l_b}^+)_{1T}$. We note that for case 3, $X_n = 0$. Fig. 2 provides a graphical visualization of this theorem.

B. Numerical proof

Numerical proof of the theorem is obtained through the following pseudocode. With the biases of all qubits in the register as the input, it outputs the exact swaps that need to be performed, and verifies the optimality of these swaps by ruling out all other swaps by demonstrating cases 1 and 2 presented in Theorem. 1. As a consequence of Algorithm 1, all the PROBAMPSin $|0T\rangle$ subspace would be greater than or equal to all the PROBAMPSin the $|1T\rangle$

subspace of the TARGET qubit. Line 4 of algorithm 1 describes the DIAG-ELS and PROBAMPS, line 11 implements definition 1, and lines 21, 30 and 41 initiate numerical proof statements 1, 2, and 3 of theorem 1 respectively. The conjecture has been demonstrated for the several combinations of initial biases and number of qubits (see Appendix).

Algorithm 1 Optswaps

Require: Starting biases of the qubits: $\{\epsilon_i\}$

Ensure: Cool the m^{th} qubit in the register.

```

1: Bring the bias of the  $m^{th}$  qubit to position 1: swap( $\epsilon_1, \epsilon_m$ )
2: for  $j \leftarrow 0$  to  $2^n - 1$  do
3:    $\triangleright$  DIAG-EL $_j \leftarrow (j)_{10}$  converted to its binary equivalent  $(j)_2$ 
      composed of  $n$  binary digits by appropriately
      padding 0's to the left.
4:   for  $i \leftarrow 1$  to  $n$  do
5:      $\triangleright \epsilon_i^+ \leftarrow (1 + \epsilon_i)/2$  and  $\epsilon_i^- \leftarrow (1 - \epsilon_i)/2$ .
6:      $\triangleright$  PROBAMPS $_{ij} \leftarrow$  Assigning  $\epsilon_i^+$  for 0i and  $\epsilon_i^-$  for 1i
      in DIAG-EL $_j$ .
7:   end for
8:    $\triangleright$  PROBAMP $_j \leftarrow \prod_i$  PROBAMPS $_{ij}$ 
9: end for
10: The number of swaps  $n_s \leftarrow 0$ .
11: for  $k \leftarrow 0$  to  $2^{n-1} - 1$  do
12:   if PROBAMP $_k <$  PROBAMP $_{2^n-k-1}$  then
13:      $\triangleright$  swap(DIAG-EL $_k$ , DIAG-EL $_{2^n-k-1}$ ).
14:      $\triangleright n_s \leftarrow n_s + 1$ .
15:      $\triangleright$  swap(PROBAMP $_k$ , PROBAMP $_{2^n-k-1}$ )
16:      $\triangleright$  Print: Swapped  $|k\rangle$  and  $|2^n - 1 - k\rangle$  or DIAG-EL $_k$ 
      and DIAG-EL $_{2^n-k-1}$ .
17:   end if
18: end for
19:  $\triangleright$  Final bias  $\epsilon'_m = \sum_{j=0}^{2^{n-1}-1} \text{PROBAMP}_j - \sum_{j=2^{n-1}}^{2^n-1} \text{PROBAMP}_j$ 
20: if  $n_s \neq 0$  then
21:   for  $k, l \leftarrow 0$  to  $2^{n-1} - 1$  do
22:     if PROBAMP $_k <$  PROBAMP $_{2^n-k-1}$  and
      PROBAMP $_l >$  PROBAMP $_{2^n-l-1}$  then
23:        $\triangleright v \leftarrow \text{PROBAMP}_{2^n-k-1} - \text{PROBAMP}_k$ 
24:       if  $\text{PROBAMP}_{2^n-l-1} - \text{PROBAMP}_k > v$  then
25:         Print : Exception found; case 1. disproved;
      Swaps are not optimal.
26:       else
27:         Print : Exception not found; case 1.
      demonstrated.
28:       end if
29:     end if
30:   if PROBAMP $_k <$  PROBAMP $_{2^n-k-1}$  and
      PROBAMP $_l <$  PROBAMP $_{2^n-l-1}$  then
31:      $\triangleright v_1 \leftarrow \text{PROBAMP}_{2^n-k-1} - \text{PROBAMP}_k$ 
32:      $\triangleright v_2 \leftarrow \text{PROBAMP}_{2^n-l-1} - \text{PROBAMP}_l$ 
33:     if  $\text{PROBAMP}_{2^n-l-1} - \text{PROBAMP}_k > v_1 + v_2$  then
34:       Print : Exception found; case 2. disproved;
      Swaps are not optimal.
35:     else
36:       Print : Exception not found; case 2.
      demonstrated.

```

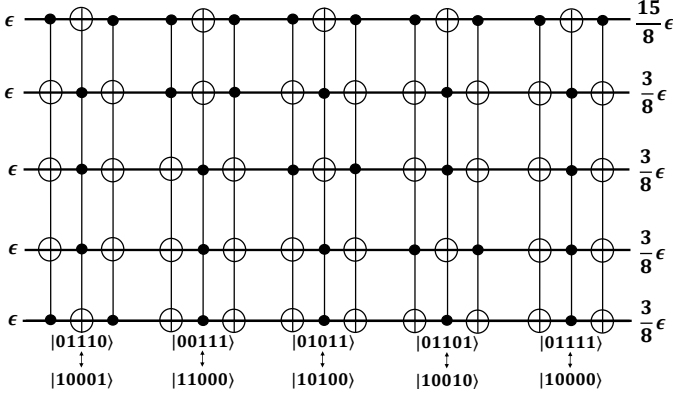


FIG. 3. An example of NB-MaxComp: the 5B-MaxComp compression circuit. The OPTSWAP corresponding to each unitary transformation is shown below the 3-line gate set which implements it. There are five such sets in this case.

```

37:     end if
38:     end if
39:   end for
40: else if  $n_s = 0$  then
41:   for  $k, l \leftarrow 0$  to  $2^{n-1} - 1$  do
42:     if  $\text{PROBAMP}_k < \text{PROBAMP}_{2^n-l-1}$  then
43:       Print : Exception found; case 3. disproved;
         Swaps are not optimal.
44:     else
45:       Print : The qubit cannot be cooled.
46:     end if
47:   end for
48: end if

```

III. NB-MAXCOMP

The quantum circuits that implement the unitary transformations proposed above are multi-qubit analogs of the *C-NOT* or *Toffoli* [32] gates. For each swapped *DIAG-EL* in the $0T$ subspace, we put a *CONTROL* gate for each individual swap $|0\rangle_{0T} \leftrightarrow |1\rangle_{1T}$ and a *NOT* gate for each $|1\rangle_{0T} \leftrightarrow |0\rangle_{1T}$. The circuit corresponding to optimal entropy compression in a 5 qubit register with equal initial biases is shown in Fig. 3.

IV. OPEN-SYSTEM COMPRESSION

Upon implementing the OPTSWAP on a set of qubits, bias of the first qubit increases and is compensated by the decrease in bias for rest of the qubits because entropy is conserved in the closed system. For example, see the 5-qubit register in Fig. 3. In order to again cool the first qubit, we need to bring the bias in rest of the qubits back to their default or initial (terms used interchangeably in the text) value by bringing the register in contact with an environment which acts as a heat-bath and therefore as an entropy sink. This can be done by surrounding all the qubits with satellite qubits like in a star-topology regis-

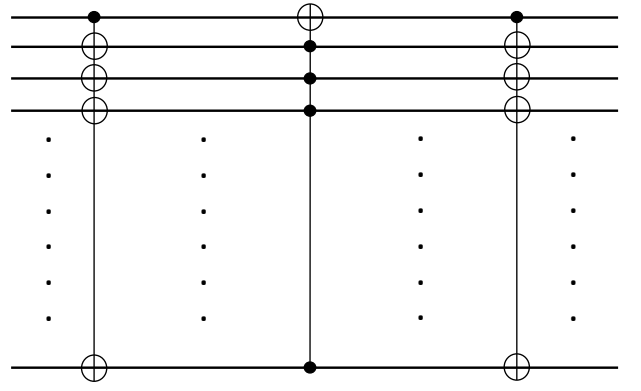


FIG. 4. The n -qubit LIM-Comp circuit implements the transformation corresponding to the limiting swap: $|0111\dots11\rangle \leftrightarrow |1000\dots00\rangle$

ter [3], or by swapping their biases with a single [28, 30] or multiple [33] refrigerant qubits whose sole purpose is to serve as an intermediary between the bath and computation qubits which are sought to be cooled.

A. Limiting Swap

To find the limit of cooling a single qubit given a set of ancilla qubits to which its entropy can be transferred, we need to find the OPTSWAP which is *last* beneficial OPTSWAP. This OPTSWAP, termed as the limiting swap is given by $|011\dots1\rangle_{0T} \leftrightarrow |100\dots0\rangle_{1T}$. It is formalized below:

Proof 1. Given that $a \in \{0, \dots, 2^{n-1} - 2\}$ and $b \in \{2^{n-1} + 1, \dots, 2^n - 1\}$, $\forall a$ and b , we have $R_{a,0T} > R_{2^{n-1}-1,0T}$ and $R_{b,1T} < R_{2^{n-1},1T}$. It implies that, $R_{2^{n-1},1T} \leq R_{2^{n-1}-1,0T}$ if and only if, $\forall a$ and b , $R_{a,0T} \geq R_{b,1T}$ which violates Def. 1.

The proof establishes that the limiting swap $|011\dots1\rangle_{0T} \leftrightarrow |100\dots0\rangle_{1T}$ is counterproductive if and only if all other swaps are also counterproductive making it the last productive swap. It can be implemented using the data compression circuit LIM-Comp shown in Fig. 4. Bias of the TARGET qubit at the limit can be found by equating the PROBAMPs corresponding to the limiting swap: $(\epsilon_l)_+ \epsilon_-^m = (\epsilon_l)_- \epsilon_+^m$, which gives us:

$$\epsilon_l = \frac{(1 + \epsilon)^m - (1 - \epsilon)^m}{(1 + \epsilon)^m + (1 - \epsilon)^m}, \quad (4)$$

assuming that default bias of all the ancilla qubits is identically ϵ and bias of the computation TARGET qubit is denoted by ϵ_l . In the example of a star-topology register mentioned above, this limiting bias can act as default bias of the computation qubits in the idealized scenario of infinite relaxation time for the computation qubits. As mentioned, in other cases the default bias would be $m\epsilon$ for multiple refrigerant qubits or ϵ for a single refrigerant qubit.

B. Multi-round limit – Analytical Proof

Above, we found the limit of cooling a single qubit to its limit assuming that the initial biases of the rest of the qubits are all identically ϵ . Here, our purpose is to cool a register of multiple qubits where all qubits are cooled to their respective limits. We begin by cooling the first qubit to its limit by utilizing the minimum bias of the rest of the qubits at each compression step. We continue cooling each remaining qubit in the register with the help of respective number of qubits below it, i.e., our subspace would contain one less qubit as we go down the hierarchy. At the end of this procedure, all the qubits in the register are cooled to their first-round limits. Using the first-round limits of all the qubits in the register, we again cool the first qubit, this time to its second-round limit. Again, we proceed to cool each remaining qubit to its respective second-round limit, where the subspace being utilized sees a reduction of one qubit as we go down the hierarchy. We continue this procedure to obtain the multi-round limit of cooling the quantum register. Thus, the expression for the limit of purifying the k^{th} qubit in the r^{th} limiting round in a register of size n is given by equating the PROBAMPS corresponding to the limiting swap in respective rounds (similar to the Eq. 4):

$$\epsilon_{rkn} = \frac{(1 + \epsilon)^{f(r,k,n)} - (1 - \epsilon)^{f(r,k,n)}}{(1 + \epsilon)^{f(r,k,n)} + (1 - \epsilon)^{f(r,k,n)}}, \quad (5)$$

where the function $f(r, k, n)$ is given by recursive relations which we shall derive here. Without loss of generality about the specific refrigerant qubit scenario, we assume the default bias to be ϵ .

Consider the first qubit in the first limiting round: it is purified to its limit by transferring entropy to the $n - 1$ qubits lower in the hierarchy (see limiting swap 1): $(\epsilon_{11n})_+ \epsilon_-^{n-1} = (\epsilon_{11n})_- \epsilon_+^{n-1}$. We thus have $f(1, 1, n) = n - 1$. The second qubit would be purified using only the $n - 2$ qubits lower than itself: $f(1, 2, n) = n - 2$. For k^{th} qubit, we thus have $f(1, k, n) = n - k$. For the last and the penultimate qubit, the function equals just 1. The second round limit for the k^{th} qubit is obtained by using the first round limits of all the qubits lower in the hierarchy: $[(\epsilon_{2kn})_+] \prod_{i=k+1}^{n-2} \epsilon_-^{n-i} \epsilon_-^2 = [(\epsilon_{2kn})_-] \prod_{i=k+1}^{n-2} \epsilon_+^{n-i} \epsilon_+^2$. Solving this, we find $f(2, k, n) = \sum_{i=k+1}^{n-2} f(1, i, n) + 2$, where the 2 is added for the last two qubits. The third round limit is obtained by adding the function for the second round up to the $(n - 3)^{rd}$ qubit which is added to the function for the first round: $f(3, k, n) = \sum_{i=k+1}^{n-3} f(2, i, n) + f(1, n - 2, n) + 2$. Similarly for the 4^{th} qubit, we have $f(4, k, n) = \sum_{i=k+1}^{n-4} f(3, i, n) + f(2, n - 3, n) + f(1, n - 2, n) + 2$, and for the 5^{th} qubit, we get $f(5, k, n) = \sum_{i=k+1}^{n-5} f(4, i, n) + f(3, n - 4, n) + f(2, n - 3, n) + f(1, n - 2, n) + 2$ and so on.

We notice that as one goes further into the limiting rounds, the limit up to only the $(n - r - 1)^{th}$ qubit shows an increment. Also, we note the difference in expressions of f corresponding to the 1^{st} and 2^{nd} rounds, and that

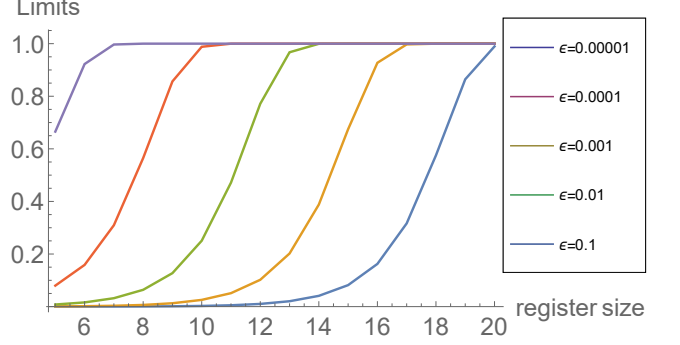


FIG. 5. The y-axis represents limits ϵ_{rkn} for $r = n - 2$ and $k = 1$ plotted against the register size n on the x-axis.

of round 3 and further. Based on this observation, the recursive relation corresponding to $r > 2$ & $k < n - r$ is given by:

$$f(r, k, n) = \sum_{j=1}^{r-2} f(j, n - j - 1, n) + \sum_{i=k+1}^{n-r} f(r - 1, i, n) + 2. \quad (6)$$

Similar observation for $r = 2$ & $k < n - r$ yields

$$f(r, k, n) = \sum_{i=k+1}^{n-r} f(r - 1, i, n) + 2. \quad (7)$$

Finally, for $r \geq 2$ & $k \geq n - r$, we get

$$f(r, k, n) = f(r - 1, k, n). \quad (8)$$

The initial condition for f corresponds to the first-round limit of the respective qubits, where $k < n - 1$: $f(1, k, n) = n - k$. For $k \geq n - 1$, we simply have $f(1, k, n) = 1$. We also note that for a given n , $r_{max} = n - 2$. In the low initial bias case, Eqn.5 can be expanded to first order in ϵ for the case $k = 1, r = r_{max}$, to obtain $\epsilon_{rkn} = 2^{n-2}\epsilon$, which is consistent with Refs. [27, 28, 30, 33].

From eqn. 5, together with eqns. 6, 7 or 8, we can find the limit of cooling a particular qubit or a particular set of qubits in the quantum register. Thus our quantum information processor achieves flexibility due to its ability to separate the computation space of any size (dependent on the nature of the computation) from the qubits that are just meant to cool the computation qubits. Further, the formula can be used to tailor our needs by fixing two or three of the four variables, ϵ_{rkn} , r , k , and n to obtain a space of the remaining variables satisfying the chosen constraint. In Fig. 5, we show the limits for different values of initial biases when $r = r_{max}$ and $k = 1$. In Fig. 6, we show the change in limits with different limiting rounds when the initial bias is fixed at $\epsilon = 0.00001$ for several values of n . Fig. 7 shows variance of Limits with respect to the register size and the number of rounds. We

C. Numerical Limits

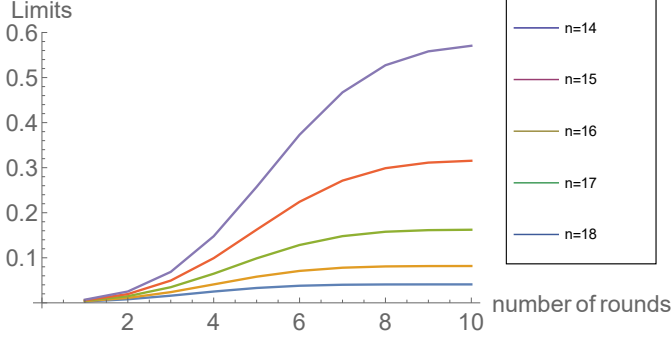


FIG. 6. The y-axis represents limits ϵ_{rkn} for $\epsilon = 0.00001$ and $k = 1$ plotted against the number of rounds r on the x-axis.

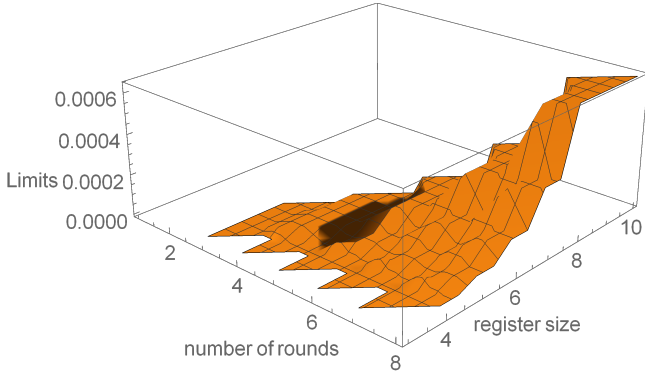


FIG. 7. The y-axis represents limits ϵ_{rkn} for $\epsilon = 0.00001$ and $k = 1$ plotted against the register size n on the z-axis and number of rounds r on the x-axis. Surfaces for other values of initial biases ϵ are shown in the appendix.

can also plot (see Fig. 8) the limits with respect to the number of rounds.

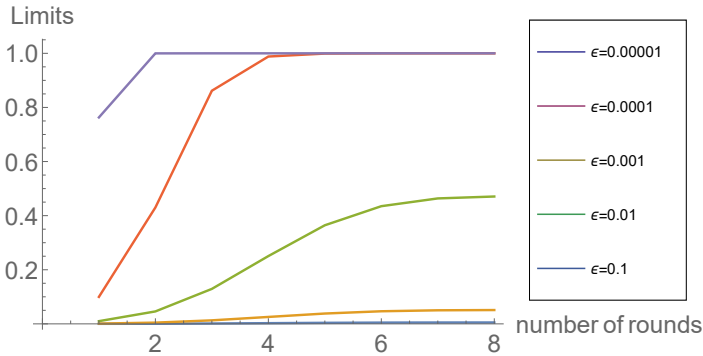


FIG. 8. The y-axis represents limits ϵ_{rkn} with $k = 1$ and $n = 11$ for different values of initial biases plotted against the number of rounds r on the x-axis when register size is $n = 11$.

The afore-derived limits are obtained under the assumption that the initial or default biases of all the qubits in the register are equal (ϵ) to begin with. However, to make a statement about the limiting entropy distribution in a quantum register with different default biases (say, due to connection with heat baths at different temperatures), we construct the pseudocode Limits 2. As expected, the afore-derived limits can be numerically obtained by implementing this pseudocode.

Within a given limiting round (r), the *for* loop (line 8) truncates the subspace of the qubit register from the v^{th} qubit to the $n - r - 1^{th}$ qubit. The upper limit on the subspace size is imposed by noting that within a given limiting round, the qubit index $k \leq n - r - 1$. Line 10 enters a *while* loop which repeatedly implements compression using the OPTSWAP still the point where the bias/purity of the v^{th} qubit can no longer be increased. The *while* loop terminates when the ratio of purities (line 31) before and after compression (single iteration of the *while* loop) asymptotically reaches 1. When the purity of the v^{th} qubit reaches its limit within a given round (line 33), it is disconnected from the subspace by the next $-v + 1^{th}$ iteration of the *for* loop. Further, when the biases/purities of all the qubits reach their limits within a given limiting round, they serve as initial biases of the register for the next $-r + 1^{th}$ round (lines 35 through 39).

Lets call the first qubit within a compression step as the target and the remaining qubits, lower down in the register, as ancilla (note that this characterization is valid only *within* a given compression step). It should be noted that although we are arriving at the respective limits in this pseudocode within a given compression step (the *while* loop), we do not account for the purity/bias decrease of the ancilla when we increase the bias of the target qubit by transferring its entropy to the remaining qubits within the compression subspace. This ignorance can be justified with the argument that, within a given round (say r), when an ancilla qubit's purity decreases than its limit in the previous round ($r - 1$), one can do a series of compression rounds to bring it back to the level of the previous round ($r - 1$) using the qubits lower in the hierarchy with respect to the particular ancilla qubit before proceeding to increase the purity of the target within the compression step in the r^{th} round. The same observation holds for all the ancilla qubits within the compression subspace. This argument is bolstered in the register compression algorithm (see next section) in which the pseudocode Limits serves as a subroutine providing target purities/biases of the register for each limiting round.

Algorithm 2 Limits

Require: Number of qubits, n .
Require: Starting biases of the qubits, $\{\epsilon_i^{in}\}$.
Require: Number of *rounds*; highest possible is $n - 2$.
Require: Desired *precision*.
Ensure: Sequentially arrive at the cooling limit of all the qubits in the register.

```

1:  $\triangleright \omega_i \leftarrow \epsilon_i^{in}$ 
2: for  $r \leftarrow 1$  to rounds do
3:   if  $r > 1$  then
4:      $\{\epsilon_i^{in}\} \leftarrow \{\epsilon_{r-1,i}^f\}$ 
5:   else
6:      $\{\epsilon_i^{in}\} \leftarrow \{\omega_1, \epsilon_{2:n}^{in}\}$ 
7:   end if
8:   for  $v \leftarrow 1$  to  $n - r - 1$  do
9:      $s \leftarrow 0$ 
10:    while Purity increase ratio  $\neq 1$  within the given precision requirement. do
11:       $s \leftarrow s + 1$ 
12:      if  $s > 1$  then
13:         $\{\epsilon_i^{transient}\} \leftarrow \{\epsilon^{increase}, \epsilon_{v+1}^{in}, \dots, \epsilon_n^{in}\}$ 
14:      end if
15:      for  $j \leftarrow 0$  to  $2^{n-v+1} - 1$  do
16:         $\triangleright \text{DIAG-EL}_j \leftarrow (j)_{10}$  converted to its binary equivalent  $(j)_2$  composed of  $n$  binary digits by appropriately padding 0's to the left.
17:        for  $i \leftarrow 1$  to  $n - v + 1$  do
18:           $\triangleright \epsilon_i^+ \leftarrow (1 + \epsilon_i)/2$  and  $\epsilon_i^- \leftarrow (1 - \epsilon_i)/2$ .
19:           $\triangleright \text{PROBAMP}_{ij} \leftarrow$  Assigning  $\epsilon_i^+$  for 0's and  $\epsilon_i^-$  for 1's in  $\text{DIAG-EL}_j$ .
20:        end for
21:         $\triangleright \text{PROBAMP}_j \leftarrow \prod_i \text{PROBAMP}_{ij}$ 
22:      end for
23:      for  $k \leftarrow 0$  to  $2^{n-v} - 1$  do
24:        if  $\text{PROBAMP}_k < \text{PROBAMP}_{2^{n-v+1}-k-1}$  then
25:           $\triangleright \text{swap}(\text{DIAG-EL}_k, \text{DIAG-EL}_{2^{n-v+1}-k-1})$ .
26:           $\triangleright \text{swap}(\text{PROBAMP}_k, \text{PROBAMP}_{2^{n-v+1}-k-1})$  or  $\text{DIAG-EL}_k$  and  $\text{DIAG-EL}_{2^{n-v+1}-k-1}$ .
27:        end if
28:      end for
29:       $\triangleright \text{Final bias } \epsilon^{increase} \leftarrow \frac{\sum_{j=0}^{2^{n-v}-1} \text{PROBAMP}_j - \sum_{j=2^{n-v}}^{2^{n-v+1}-1} \text{PROBAMP}_j}{\epsilon_v^{in}}$ 
30:      Purity increase ratio  $\leftarrow \frac{\epsilon^{increase}}{\epsilon_v^{in}}$ 
31:    end while
32:     $\epsilon_{r,v}^f \leftarrow \epsilon^{increase}$ 
33:  end for
34:  if  $r = 1$  then
35:     $\epsilon_{r,v+1:n}^f \leftarrow \omega_{v+1:n}$ 
36:  else
37:     $\epsilon_{r,v+1:n}^f \leftarrow \epsilon_{r-1,v+1:n}^f$ 
38:  end if
39: end for
40: Print :  $\epsilon^f$ , which displays the limit for each qubit in each round as a  $r \times n$  matrix.

```

Fig. 9 provides a graphical visualization of the above subroutine.

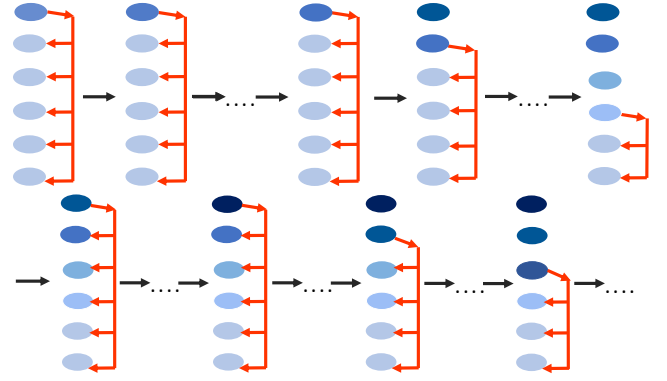


FIG. 9. This graphic represents working of the subroutine Limits 2 for a 6-qubit register. Lightest shade of blue represents qubits with default or initial biases. Darker shades of blue represent cooler qubits. The top half (L-R) of the figure represents purifying the quantum register to round 1 limits of the respective qubits. Bottom half (L-R) represents the use of round 1 limits to purify respective qubits to their round 2 limits.

D. Register Compression Subroutine

The limits obtained in the previous subroutine serve as targets for *adaptively* initializing the quantum register in each round. This program ensures that all the qubits in the register are exactly initialized till the desired limit in that particular round. This allows us to systematically obtain better cooling in successive rounds. Here we shall account for the reduction in purities of the ancilla qubits within a compression subspace during iteration of the while loop. In effect, we are able to find and count all of the swaps or unitary transformations needed to initialize the quantum register.

Within a given limiting round (line 2), we enter the compression subspace of a particular target qubit (x ; line 4) where all the qubits from 1 to $x - 1$ are disconnected from the subspace because they are purified to the limit corresponding to that round. We then enter the subspace compression subroutine to purify the chosen subspace to its limits corresponding to round r .

Algorithm 3 Register Compression

Require: Number of qubits, n .
Require: Starting biases of the qubits, $\{\epsilon_i^{in}\}$.
Require: Number of *rounds*; highest possible is $n - 2$.
Require: Desired *precision*.
Require: Round limits (RL) \leftarrow Zero matrix of dimensions $rounds \times n$
Require: *Targets* \leftarrow Limits($n, \{\epsilon_i^{in}\}, rounds, precision$)
Ensure: Sequentially arrive at the cooling limit of all the qubits in the register and count the number of unitary transformation required in the process.

```

1:  $\triangleright$  complexity  $\leftarrow$  0
2: for  $r \leftarrow 1$  to rounds do
3:    $\triangleright$  Swaps in this round (totswaps)  $\leftarrow$  0
4:   for  $x \leftarrow 1$  to  $n - r - 1$  do
5:      $\triangleright$   $z \leftarrow 0$ 
6:      $\triangleright$  Swaps for subspace compression (tttswaps)  $\leftarrow$  0
7:      $\triangleright$  [tttswaps, Subspace limits (SL)]  $\leftarrow$ 
       SI( $n, rounds, Targets, x, z, tttswaps, precision, RL$ )
8:      $\triangleright$  totswaps  $\leftarrow$  totswaps + tttswaps
9:      $\triangleright$  RL  $\leftarrow$  SL
10:   end for
11:   if  $r < rounds$  then
12:     RL $_{r+1}$   $\leftarrow$  RL $_r$ 
13:   end if
14:   complexity  $\leftarrow$  complexity + totswaps
15: end for
16: Print: complexity – Total number of unitaries leading up
    to the limits.
17: Print: Round limits – Matrix of dimensions  $rounds \times n$ 
    with the limit of each qubit in every round.
```

Within the subspace compression subroutine, we enter another subspace – let's call it the subsubspace (line 4 in 4) – with the v^{th} qubit as the target. The compressive while loop (line 11) now accounts for the reduction in purities/biases of the ancilla qubits (lines 39-43) due to entropy transfer out of the target. Line 9. ensures that we enter the compression loop only if the bias/purity of the particular target qubit is less than the limit for that particular round, which is loaded from the subroutine Limits2. Further, we put conditions for termination of the for loop (responsible for the OPTSWAPS) within the while loop if the bias/purity of the target qubit in any given subsubspace overshoots the database Limits (lines 28, 30, and 32). This ensures that the first qubit within the subsubspace assumes precedence in the order of purification over the rest leading to a systematic descending order of biases in the qubit register after each limiting round. The subsubspace purities are updated (lines 56-58) only when we enter the compression while loop indicated by the value of 'w'. This information is then fed into our main database (RL) for processing in the next $((v+1)^{th})$ iteration of the for loop.

If the compressive while loop, and therefore the for loop, succeeds (indicated by the difference in initial and final purities (line 64)), we enter the recursive conditions taking us back to the subspace compression subroutine if we fall short of the target for the corresponding round (line 68 for the first qubit in the target subsubspace and

line 73 for the rest). The subspace compression subroutine is repeated till these conditions fail. The output goes to the register compression subroutine as the total number of the unitaries (tttswaps) and the RL database as we move to the next $((x+1)^{th})$ target subspace in the register compression subroutine.

Algorithm 4 Subspace Compression

Require: $\{\epsilon_i^{in}\}$, RL, n , *rounds* (r), *Targets*, tttswaps, precision, x , z .
Ensure: The subspace spanning the x^{th} qubit to the n^{th} qubit is initialized for the round r .

```

1:  $\triangleright$  Swaps initializing a single target subspace (ttswaps)  $\leftarrow$  0
2:  $\triangleright$   $\alpha \leftarrow$  RL $_r$ 
3:  $\triangleright$   $a \leftarrow$  RL $_r$ 
4: for  $v \leftarrow x$  to  $n - 1$  do
5:    $\triangleright$   $s \leftarrow 0$ 
6:    $\triangleright$  Purity increase ratio  $\leftarrow$  0
7:    $\triangleright$   $\beta \leftarrow \alpha_{v:n}$ 
8:    $\triangleright$  Swaps initializing the first qubit within the target subspace (tswaps)  $\leftarrow$  0
9:   if  $\alpha_v < Targets_{r,v}$  then
10:     $w \leftarrow 1$ 
11:    while Purity increase ratio  $\neq 1$  within the desired
      precision do
12:       $s = s + 1$ 
13:      if  $s > 1$  then
14:         $\beta \leftarrow \gamma$ 
15:      end if
16:      for  $j \leftarrow 0$  to  $2^{n-v+1} - 1$  do
17:         $\triangleright$  DIAG-EL $_j \leftarrow (j)_{10}$  converted to its binary
          equivalent  $(j)_2$  composed of  $n$  binary digits by
          appropriately padding 0's to the left.
18:        for  $i \leftarrow 1$  to  $n$  do
19:           $\triangleright$   $\epsilon_i^+ \leftarrow (1 + \epsilon_i)/2$  and  $\epsilon_i^- \leftarrow (1 - \epsilon_i)/2$ .
20:           $\triangleright$  PROBAMP $_{ij} \leftarrow$  Assigning  $\epsilon_i^+$  for 0i and  $\epsilon_i^-$ 
            for 1i in DIAG-EL $_j$ .
21:        end for
22:         $\triangleright$  PROBAMP $_j \leftarrow \prod_i$  PROBAMP $S_{ij}$ 
23:      end for
24:       $\triangleright$   $\gamma \leftarrow \beta$ 
25:       $\triangleright$   $\gamma' \leftarrow \gamma$ 
26:       $\triangleright$  swap  $\leftarrow$  0
27:      for  $k \leftarrow 0 : 2^{n-v}$  do
28:        if  $z=0$  &  $r > 1$  &  $v > x$  &  $\gamma_1 \geq targets_{r-1,v}$  then
29:          break
30:        else if  $z=1$  &  $r > 1$  &  $\gamma_1 \geq targets_{r-1,v}$  then
31:          break
32:        else if  $z=0$  &  $v=x$  &  $\gamma_1 \geq targets_{r-1,v}$  then
33:          break
34:        else if PROBAMP $_k <$  PROBAMP $_{2^{n-v+1}-k-1}$  then
35:           $\triangleright$  swap(DIAG-EL $_k$ , DIAG-EL $_{2^{n-v+1}-k-1}$ ).
36:           $\triangleright$  swap  $\leftarrow$  swap + 1
37:           $\triangleright$  swap(PROBAMP $_k$ , PROBAMP $_{2^{n-v+1}-k-1}$ )
38:          for  $i \leftarrow 1$  to  $n - v + 1$  do
39:             $\triangleright$   $c \leftarrow 2^{n-v+1-i}$ 
40:             $\triangleright$   $\gamma_i \leftarrow \sum_{j=1}^c$  PROBAMP $_j - \sum_{j=1+c}^{2*c} PROBAMP_j$ 
```

```

41:   for  $m \leftarrow 1$  to  $2^{i-1} - 1$  do
42:      $\triangleright \gamma_i \leftarrow \gamma_i + \sum_{j=1+(2^m)*c}^{(2^{m+1})*c} \text{PROBAMP}_j -$ 
        $\sum_{j=1+(2^m+1)*c}^{(2^m+2)*c} \text{PROBAMP}_j$ 
43:   end for
44:   if  $\gamma_i < \epsilon_i^{in}$  then
45:      $\gamma_i \leftarrow \epsilon_i^{in}$ .
46:   end if
47:   end for
48:   end if
49:   end for
50:    $\triangleright$  Purity increase ratio  $\leftarrow \frac{\gamma_i}{\gamma_1}$ 
51:    $\triangleright$  tswaps  $\leftarrow$  tswaps + swaps
52:   end while
53:   else
54:      $w \leftarrow 0$ 
55:   end if
56:   if  $w = 1$  then
57:      $\alpha_{v:n} \leftarrow \gamma_{1:n-v+1}$ 
58:   end if
59:    $\triangleright \text{RL}_{r,v:n} \leftarrow \alpha_{v:n}$ 
60:    $\triangleright$  tswaps  $\leftarrow$  tswaps + tswaps
61:   end for
62:    $\triangleright b \leftarrow \text{RL}_r$ 
63:    $\triangleright g \leftarrow 0$ 
64:   if  $a = b$  then
65:      $g \leftarrow 1$ 
66:   end if
67:   tswaps  $\leftarrow$  tswaps + tswaps
68:   if  $\alpha_x < \text{Targets}_{r,x}$  &  $g = 0$  then
69:      $\triangleright z \leftarrow 0$ 
70:      $\triangleright [\text{tswaps}, \text{RL}] \leftarrow \text{SI}(\text{n}, \text{rounds}, \text{Targets}, x, z, \text{tswaps},$ 
        $\text{precision}, \text{RL})$ 
71:   end if
72:   for  $i \leftarrow x + 1 : n - 1$  do
73:     if  $r > 1$  &  $\alpha_i < \text{Targets}_{r-1,i}$  &  $g = 0$  then
74:        $\triangleright z \leftarrow 1$ 
75:        $\triangleright [\text{tswaps}, \text{RL}] \leftarrow \text{SI}(\text{n}, \text{rounds}, \text{Targets}, i, z, \text{tswaps},$ 
        $\text{precision}, \text{RL})$ 
76:     end if
77:   end for
78:    $\triangleright$  Subroutine outcome 1  $\leftarrow$  tswaps
79:    $\triangleright$  Subroutine outcome 2  $\leftarrow$  RL

```

Fig. 10 provides a graphical visualization of the register compression subroutine in case of a 4 qubit register.

We note that just using the limiting swap (implemented with the LIM-Comp 4) will converge to the limits in the idealized scenario of relaxation times, but for the case of finite relaxation times for the computation and the reset qubit(s) all the optswaps would be necessary to achieve better cooling in lesser steps as demonstrated before [3]. This would be especially pertinent during the first few steps which provide the most significant boost in the biases. Even if only the LIM-Comp is available to be used (say due to limitations of the architecture Hamiltonian), the RCS will continue to determine dynamics of HBAC because it would be necessary to make the appropriate transitions within subspaces for reaching the cool-

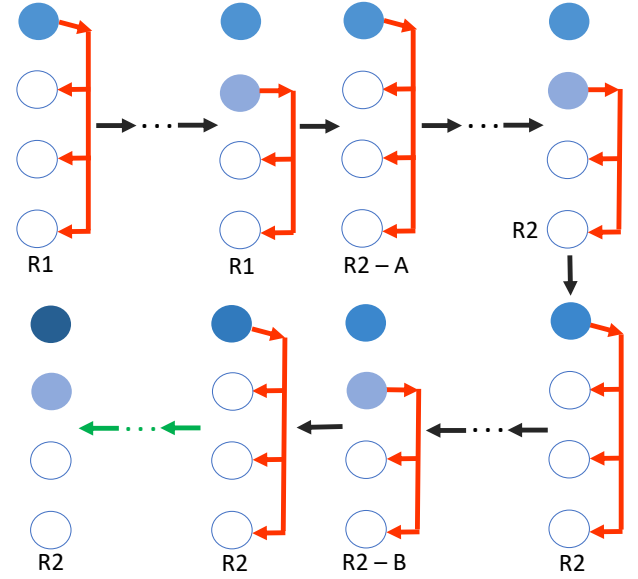


FIG. 10. Figure representing working of Register Compression 3 in tandem with Subspace Compression 4 subroutine. The empty circles represent default/initial biases of the qubits. R2—A represents the point when subspace initialization subroutine is activated to bring the second qubit back to its round 1 limit. The dashed green arrow represents repeat of steps from R2—A to R2—B till the first qubit is purified to the round 2 limit and the second qubit is purified to the round 1 limit.

ing limit for all qubits in the register. In this case, the checkpoint in line 34 of Subspace Compression 4 should be replaced with just the limiting swap for that subspace and the for loop (line 27) would not be necessary.

V. COMPLEXITY

The classical space and time complexity of implementing the register compression algorithm comprising the subroutines Limits and subspace compression can be analyzed block by block. It may be noted that the algorithm doesn't pose any hindrance if existing computing resources such as workstations with multi-core processors are utilized.

However, the quantum computational complexity comprises the total number of OPTSWAPS and the quantum time/depth and space complexity [34–36] of implementing each OPTSWAP using the NB-MaxComp. The gate complexity of the NB-MaxComp can be obtained from the multi-qubit version of the Solovay-Kitaev theorem [32] (the Universality Theorem [34]) and the several algorithms using specific generating gate sets for approximating the relevant unitaries to a certain target accuracy [37]. A useful result of note in this context is the gate and depth complexity for reversible circuits constructed with NOTs, C–NOTs and 2–CNOTs [38], which if generalized to k–C–l–NOT (with k+l = N) gates would be ap-

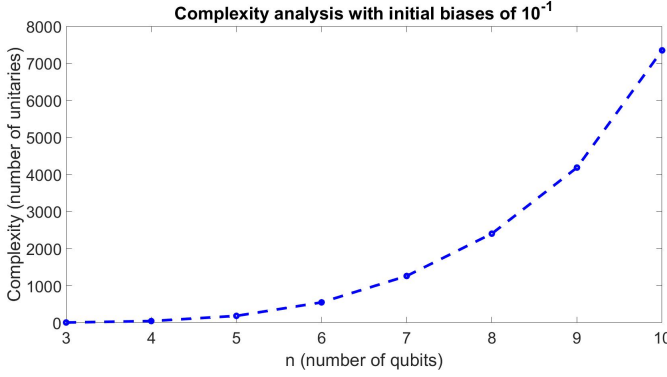


FIG. 11. Y-axis represents the number of unitaries it takes to reach the limit of RCS-HBAC for different register sizes (number of qubits) plotted on x-axis for the case where initial biases of all qubits are 0.1.

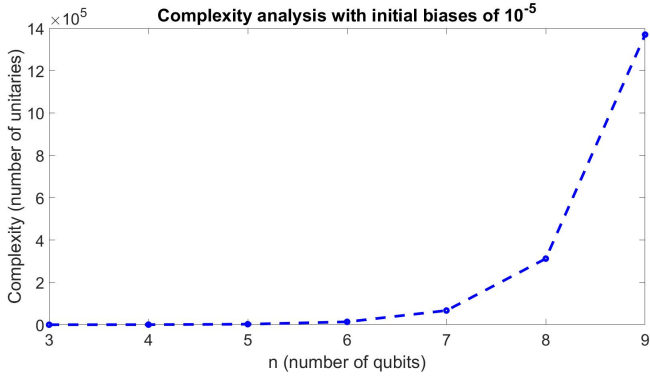


FIG. 12. Y-axis represents the number of unitaries it takes to reach the limit of RCS-HBAC for different number of qubits plotted on x-axis for the case where initial biases of all qubits are 0.00001.

plicable to the NB-MaxComp. Assuming this quantum time or space complexity to be some $\nu(N)$, the complexity of implementing register compression on a quantum information processor can be found by counting the number of OPTSWAPS required to reach the TARGET:

$$C = O(e^{\lambda N} \nu(N)) \quad (9)$$

The exponential scaling of the complexity with the number of qubits can be inferred from the following graphs. In Fig. 11, we plot the complexity to reach the limit when initial biases are all 0.1. In Fig. 12, we plot the complexity to reach the limit when initial biases are all 0.00001. In Fig. 13, the complexity is plotted for $n = 5$ against changing initial biases by an order of magnitude. While we did this to get the big-O complexity, specific cases, for example, where one starts with unequal biases and different targets can be straightforwardly obtained from the classical algorithm.

Additionally, for the LIM-Comp and NB-MaxComp, it would be interesting to minimize the quantum cost [39–41] as done for other reversible logic gates [42, 43] and to optimize their implementation as done for the Toffoli circuit family [44–47].

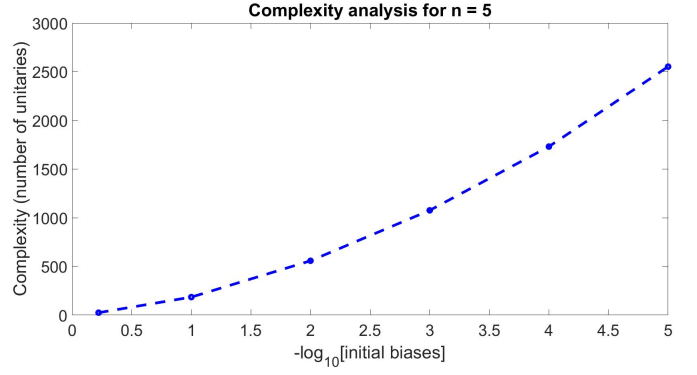


FIG. 13. Y-axis represents number of unitaries it takes to reach the limit of RCS-HBAC for cases where initial biases of all qubits are 0.1, 0.01, 0.001, 0.0001, and 0.00001 plotted on a log scale on the x-axis.

VI. DISCUSSION

In summary, we constructed an explicit heat-bath algorithmic cooling algorithm based on optimal closed-system entropy compression among qubits at each cooling step which asymptotically reaches the HBAC limit of cooling each qubit in a quantum register. In doing so, we were able to account for the reduction in entropy of each qubit while cooling a target qubit within the subspace and thus counted the total number of unitary transformations needed to achieve a desired level of cooling in the register. To reach zero temperature (purity 1), we require infinite precision in the RCS which would take infinite number of steps to reach making this consistent with the third law of thermodynamics [48]. In that context, it would be interesting to find how the temperature scales with the time required to cool as we move away from the idealized scenario to finite relaxation times, since we are able to count the exact number of steps in this operational HBAC protocol (Eqn. (7) in Ref. [48]). Also, the work cost of implementing the NB-MaxComp specific to particular architectures would help check the relation with final attainable temperature (Eqn. (1) and (13) in Ref [48]).

The optswaps introduced for the closed-system entropy compression may have implications for quantum data compression [49] when the signal states are known [50]. Particularly, lossy data compression with 3 qubit - 2 qubit block coding proposed [51, 52] and implemented [53] before uses the transformation $|011\rangle \leftrightarrow |100\rangle$. Optswaps and the NB-MaxComp would allow this to be optimally extended to block sizes of more qubits; particularly we could apply the optswaps on Hilbert spaces of sequentially decreasing dimension as one decreases the size of the block depending on how much information loss can be tolerated [54, 55]. It would be interesting to compare this method with fixed or variable-length coding schemes which work with unknown but identical signal states using the quantum Schur-Weyl transform [56–59].

The current HBAC method could be extended to cool a

register of qudits (d-level quantum systems), but the definition of optswaps for such systems adds a new layer of complexity. Further the circuits needed to implement the requisite exchanges in d-level systems would need multi-level NOT and multi-level CONTROL gates, leading to the interesting challenge of defining the ND-MaxComp quantum circuits.

ACKNOWLEDGMENTS

Helpful, encouraging remarks from and discussions with Gerardo Adesso, Sougato Bose, Kalyan Chakraborty, Nilanjana Datta, Sebastian Deffner, Azizul Haque, Dintomon Joy, Jason Kestner, Samuel J. Lomonaco, T. S. Mahesh, Jonathan Oppenheim, Aditi Sen De, Ujjwal Sen, Uttam Singh, and Kapil Paranjape are gratefully acknowledged. This work was presented at the “International Symposium on New Frontiers in Quantum Correlations” in SN Bose National Center for Basic Sciences, Kolkata (poster), at “Quantum Roundabout” in the University of Nottingham (poster), and at the APS Meeting of the Mid-Atlantic section in the University of Maryland, College Park (talk). At different points, the author acknowledges financial support from the DST-INSPIRE fellowship, Govt. of India, at HRI from the DAE, Govt. of India, and from the State of Maryland (USA) through UMBC Physics Department.

Appendix A: Numerical proof examples

Theorem (1) has been numerically verified through Algorithm 1 for the following combinations of initial biases. These combinations are chosen to reflect several scenarios which could stretch the theorem:

(i) $n = 5$: $\{0.0147, 0.0893, 0.0881, 0.0448, 0.0737\}$, $\{0.1405, 0.4176, 0.9066, 0.7843, 0.9499\}$, $\{0.0800,$

$0.00001, 0.00001, 0.1000, 0.1000\}$, $\{0.8000, 0.0001, 0.0001, 0.0002, 0.0001\}$, $\{0.00002, 0.9000, 0.1000, 0.2000, 0.0700\}$;

(ii) $n = 9$: $\{0.6492, 0.0354, 0.8406, 0.9247, 0.6720, 0.7502, 0.7357, 0.3883, 0.6489\}$, $\{0.0121, 0.0495, 0.0023, 0.0195, 0.0033, 0.0069, 0.0577, 0.0487, 0.0223\}$, $\{0.00002, 0.9000, 0.1000, 0.2000, 0.0700, 0.0900, 0.7450, 0.2000, 0.5000\}$, $\{0.2000, 0.0001, 0.0000, 0.0002, 0.0007, 0.0001, 0.000745, 0.0020, 0.5000\}$, $\{0.7000, 0.0009, 0.0001, 0.0020, 0.0007, 0.0090, 0.0075, 0.0020, 0.0006\}$;

(iii) $n = 14$: $\{0.0665, 0.0025, 0.0308, 0.0268, 0.0536, 0.0557, 0.0132, 0.0343, 0.0313, 0.0453, 0.0497, 0.0529, 0.0194, 0.0476\}$, $\{0.4586, 0.1139, 0.0833, 0.3489, 0.6718, 0.2383, 0.4097, 0.1567, 0.5259, 0.1786, 0.3542, 0.4894, 0.6236, 0.6715\}$, $\{0.7000, 0.0009, 0.0001, 0.0020, 0.000745, 0.0090, 0.0075, 0.0020, 0.0006, 0.0009, 0.00023, 0.0001, 0.0036, 0.0001\}$, $\{0.0003, 0.0900, 0.0100, 0.0020, 0.0007, 0.0900, 0.0075, 0.0020, 0.0060, 0.0009, 0.0230, 0.7000, 0.0036, 0.0500\}$, $\{0.0003, 0.0900, 0.1000, 0.2000, 0.0700, 0.9000, 0.7450, 0.2000, 0.6000, 0.0900, 0.2300, 0.7000, 0.3560, 0.5000\}$;

(iv) $n = 19$: $\{0.322, 0.93, 0.212, 0.232, 0.607, 0.569, 0.985, 0.2, 0.1, 0.732, 0.42, 0.35, 0.75, 0.23, 0.369, 0.785, 0.12, 0.21, 0.832\}$, $\{0.4, 0.000093, 0.0000212, 0.000232, 0.000607, 0.000569, 0.000985, 0.0002, 0.0001, 0.0000732, 0.000042, 0.000035, 0.000075, 0.000023, 0.000369, 0.0000785, 0.00012, 0.00021, 0.0000832\}$, $\{0.00004, 0.93, 0.212, 0.232, 0.607, 0.569, 0.985, 0.2, 0.1, 0.732, 0.42, 0.35, 0.75, 0.23, 0.369, 0.785, 0.12, 0.21, 0.832\}$, $\{0.04, 0.00093, 0.00212, 0.00232, 0.0607, 0.569, 0.0985, 0.0002, 0.1, 0.0732, 0.0042, 0.35, 0.075, 0.023, 0.000369, 0.0785, 0.0012, 0.21, 0.00832\}$, $\{0.000003, 0.000093, 0.0000212, 0.000000232, 0.00000607, 0.0000569, 0.0000985, 0.00002, 0.000001, 0.00000732, 0.000042, 0.000035, 0.000075, 0.0000023, 0.0000369, 0.00000785, 0.0000012, 0.000021, 0.0000832\}$;

(v) $n = 23$: $\{0.5359, 0.5566, 0.1308, 0.3429, 0.3119, 0.4524, 0.4966, 0.5283, 0.1932, 0.4758, 0.4586, 0.1139, 0.0833, 0.3489, 0.6718, 0.2383, 0.4097, 0.1567, 0.5259, 0.1786, 0.3542, 0.4894, 0.6236\}$

-
- [1] D. P. DiVincenzo, *Fortschritte der Physik* **48**, 771 (2000).
 - [2] E. Knill and R. Laflamme, *Physical Review A* **55**, 900 (1997).
 - [3] V. R. Pande, G. Bhole, D. Khurana, and T. S. Mahesh, *Phys. Rev. A* **96**, 012330 (2017).
 - [4] P. O. Boykin, T. Mor, V. Roychowdhury, F. Vatan, and R. Vrijen, *Proceedings of the National Academy of Sciences* **99**, 3388 (2002).
 - [5] Y. Atia, Y. Elias, T. Mor, and Y. Weinstein, *Physical Review A* **93**, 012325 (2016).
 - [6] O. W. Sørensen, *Progress in Nuclear Magnetic Resonance Spectroscopy* **21**, 503 (1989).
 - [7] O. W. Sørensen, *Journal of Magnetic Resonance* (1969) **86**, 435 (1990).
 - [8] O. W. Sørensen, *Journal of Magnetic Resonance* (1969) **93**, 648 (1991).
 - [9] L. J. Schulman and U. V. Vazirani, in *Proceedings of the thirty-first annual ACM symposium on Theory of computing* (ACM, 1999) pp. 322–329.
 - [10] J. Von Neumann, *Appl. Math Ser* **12**, 5 (1951).
 - [11] J. Baugh, O. Moussa, C. Ryan, A. Nayak, and R. Laflamme, *Nature* **438**, 470 (2005).
 - [12] G. Brassard, Y. Elias, J. M. Fernandez, H. Gilboa, J. A. Jones, T. Mor, Y. Weinstein, and L. Xiao, *arXiv preprint quant-ph/0511156* (2005).
 - [13] Y. Elias, J. M. Fernandez, T. Mor, and Y. Weinstein, *Israel Journal of Chemistry* **46**, 371 (2006).
 - [14] T. Mor, V. Roychowdhury, S. Lloyd, J. M. Fernandez, and Y. Weinstein, “Algorithmic cooling,” (2005), *US Patent 6,873,154*.
 - [15] D. K. Park, G. Feng, R. Rahimi, S. Labruière, T. Shibata, S. Nakazawa, K. Sato, T. Takui, R. Laflamme,

- and J. Baugh, *Quantum Information Processing* **14**, 2435 (2015).
- [16] D. K. Park, N. A. Rodriguez-Briones, G. Feng, R. Rahimi, J. Baugh, and R. Laflamme, in *Electron Spin Resonance (ESR) Based Quantum Computing* (Springer, 2016) pp. 227–255.
 - [17] P. Kaye, *Quantum Information Processing* **6**, 295 (2007).
 - [18] C. Ryan, O. Moussa, J. Baugh, and R. Laflamme, *Physical review letters* **100**, 140501 (2008).
 - [19] Y. Elias, H. Gilboa, T. Mor, and Y. Weinstein, *Chemical Physics Letters* **517**, 126 (2011).
 - [20] S. Zaiser, B. Masatth, D. Rao, S. Raeisi, and J. Wrachtrup, arXiv preprint arXiv:1812.06252 (2018).
 - [21] Y. Atia, Y. Elias, T. Mor, and Y. Weinstein, *International Journal of Quantum Information* **12**, 1450031 (2014).
 - [22] T. Mor, in *Encyclopedia of Algorithms*, edited by M.-Y. Kao (Springer US, New York, 2008) pp. 30–37.
 - [23] D. E. Chang, L. M. Vandersypen, and M. Steffen, *Chemical physics letters* **338**, 337 (2001).
 - [24] J. M. Fernandez, S. Lloyd, T. Mor, and V. Roychowdhury, *International Journal of Quantum Information* **2**, 461 (2004).
 - [25] G. Brassard, Y. Elias, T. Mor, and Y. Weinstein, *The European Physical Journal Plus* **129**, 258 (2014).
 - [26] Y. Elias, J. M. Fernandez, T. Mor, and Y. Weinstein, in *International Conference on Unconventional Computation* (Springer, 2007) pp. 2–26.
 - [27] Y. Elias, T. Mor, and Y. Weinstein, *Physical Review A* **83**, 042340 (2011).
 - [28] L. J. Schulman, T. Mor, and Y. Weinstein, *Physical review letters* **94**, 120501 (2005).
 - [29] L. J. Schulman, T. Mor, and Y. Weinstein, *SIAM Journal on Computing* **36**, 1729 (2007).
 - [30] S. Raeisi and M. Mosca, *Physical review letters* **114**, 100404 (2015).
 - [31] S. Raeisi, M. Kieferová, and M. Mosca, *Physical Review Letters* **122**, 220501 (2019).
 - [32] M. A. Nielsen and I. Chuang, “Quantum computation and quantum information,” (2002).
 - [33] N. A. Rodríguez-Briones and R. Laflamme, *Physical review letters* **116**, 170501 (2016).
 - [34] J. Watrous, *Encyclopedia of complexity and systems science*, 7174 (2009).
 - [35] E. Bernstein and U. Vazirani, *SIAM Journal on computing* **26**, 1411 (1997).
 - [36] T. Koike and Y. Okudaira, *Physical Review A* **82**, 042305 (2010).
 - [37] C. M. Dawson and M. A. Nielsen, arXiv preprint quant-ph/0505030 (2005).
 - [38] D. V. Zakablukov, *Journal of Computer and System Sciences* **84**, 132 (2017).
 - [39] A. Banerjee and A. Pathak, arXiv preprint arXiv:0910.2129 (2009).
 - [40] M. M. Rahman, A. Banerjee, G. W. Dueck, and A. Pathak, in *2011 41st IEEE International Symposium on Multiple-Valued Logic* (IEEE, 2011) pp. 86–92.
 - [41] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, *Phys. Rev. A* **52**, 3457 (1995).
 - [42] R. Saligram, S. S. Hegde, S. A. Kulkarni, H. Bhagyalakshmi, and M. Venkatesha, *International Journal of VLSI Design & Communication Systems* **4**, 53 (2013).
 - [43] V. Pareek, arXiv preprint arXiv:1410.2373 (2014).
 - [44] B. P. Lanyon, M. Barbieri, M. P. Almeida, T. Jennewein, T. C. Ralph, K. J. Resch, G. J. Pryde, J. L. O’Brien, A. Gilchrist, and A. G. White, *Nature Physics* **5**, 134 (2009).
 - [45] R. D. a. Nabila Abdessaied, *Reversible and Quantum Circuits: Optimization and Complexity Analysis*, 1st ed. (Springer International Publishing, 2016).
 - [46] D. M. Miller, R. Wille, and Z. Sasanian, in *2011 41st IEEE International Symposium on Multiple-Valued Logic* (IEEE, 2011) pp. 288–293.
 - [47] Y. Nam, N. J. Ross, Y. Su, A. M. Childs, and D. Maslov, *npj Quantum Information* **4**, 23 (2018).
 - [48] L. Masanes and J. Oppenheim, *Nature communications* **8**, 14538 (2017).
 - [49] C. H. Bennett, *Physics Today* **48**, 24 (1995).
 - [50] B. Schumacher, *Physical Review A* **51**, 2738 (1995).
 - [51] R. Jozsa and B. Schumacher, *Journal of Modern Optics* **41**, 2343 (1994).
 - [52] J. A. Vaccaro, Y. Mitsumori, S. M. Barnett, E. Andersson, A. Hasegawa, M. Takeoka, and M. Sasaki, in *International Symposium on Stochastic Algorithms* (Springer, 2003) pp. 98–107.
 - [53] Y. Mitsumori, J. A. Vaccaro, S. M. Barnett, E. Andersson, A. Hasegawa, M. Takeoka, and M. Sasaki, *Physical review letters* **91**, 217902 (2003).
 - [54] N. Datta, J. M. Renes, R. Renner, and M. M. Wilde, *IEEE Transactions on Information Theory* **59**, 8057 (2013).
 - [55] J. Langford, *Physical Review A* **65**, 052312 (2002).
 - [56] M. Plesch and V. Bužek, *Physical Review A* **81**, 032317 (2010).
 - [57] D. Bacon, I. L. Chuang, and A. W. Harrow, *Physical review letters* **97**, 170502 (2006).
 - [58] L. A. Rozema, D. H. Mahler, A. Hayat, P. S. Turner, and A. M. Steinberg, *Physical review letters* **113**, 160504 (2014).
 - [59] M. Hayashi and K. Matsumoto, *Physical Review A* **66**, 022311 (2002).