Quantum arithmetics via computation with minimized external control: The half-adder

Liam Eloie,[1] Leonardo Banchi,[1,2] and Sougato Bose[1]
[1]*Department of Physics and Astronomy, University College London, Gower St., London WC1E 6BT, United Kingdom*
[2]*QOLS, Blackett Laboratory, Imperial College London, London SW7 2AZ, United Kingdom*

(Received 3 April 2018; published 13 June 2018)

The while-you-wait computing paradigm combines elements of digital and analog quantum computation with the aim of minimizing the need of external control. In this architecture the computer is split into logic units, each continuously implementing a single recurring multigate operation via the unmodulated Hamiltonian evolution of a quantum many-body system. Here we use evolutionary algorithms to engineer such many-body dynamics, and develop logic units capable of continuously implementing a quantum half-adder in a time-independent four-qubit network, where qubits are coupled with either Ising or Heisenberg interactions. Our results provide a step for the development of larger modules for full quantum arithmetics.

DOI: 10.1103/PhysRevA.97.062321

I. INTRODUCTION

Current quantum computing prototypes can be divided into two categories: either based on digital or analog operations. Digital approaches are normally used in systems with a high degree of control, such as transmon-based superconducting systems [1] or ion traps [2], and have the advantage of enabling the use of error-correcting codes [3]. In digital computers, genuine quantum many-body effects are normally considered detrimental, and ideally one would like to switch on and off the interactions between pairs of constituents at will, e.g., via continuous control [4] on the system. However, spurious interactions (e.g., cross-talk) generally remain, because of either imperfect switching, imperfect isolation from the environment, or from the other qubits. Moreover, the control effectively couples the qubits to a classical device which eventually increases decoherence and heating [5]. On the other hand, analog computers such as quantum annealers [6] exploit collective many-body effects to achieve computation through driving the system to different equilibrium phases. Possible disadvantages are, from a fundamental perspective, the lack of error correction and efficient compilation schemes, and from a practical perspective, the questioned degree of "quantumness" of current devices [6].

While-you-wait (WYW) computing has been proposed as a method to combine elements of digital and analog computation [7]. As in analog computing, some computation is implemented into the physical evolution (here unitary dynamics) of an interacting many-particle system. Moreover, as in the digital approach, the overall algorithm is decomposed into smaller steps. However, unlike traditional gate-based paradigms where one is normally interested in sets of universal gates, in WYW the decomposition is such that the external control is minimized. One possibility is to develop computational units that operate continuously without time-dependent control. Each unit implements a particular multigate operation which recurs multiple times in an algorithm. In WYW computing the only time-dependent operation is the transmission of states between the different units, which continuously operate the same transformation without external action. The transmission can be either obtained via optical means, or with one-dimensional (1D) qubit chains, to avoid the hybrid setups. Combining WYW computing with low-control methods to transfer states in 1D chains [8–16] would allow a minimal control approach to computation [7].

In this paper we focus on developing a logic unit that implements the quantum half-adder gate, a recurring transformation for quantum arithmetics [17–19], and, as such, a key component in the Shor algorithm [20]. Because of its importance, different schemes have been proposed to implement half-adders in different experimental systems, such as linear optics [21], nanographene molecules [22], 1D cellular automaton [23], and superconducting [24] or atomic [25] systems. Successful implementations of a quantum half-adder have also been demonstrated experimentally, for example, using nuclear magnetic resonance 7/2-spin systems [26], although with a 10% error rate, mainly due to radio frequency inhomogeneity and pulse imperfections—since multiple pulses are required to perform the half-adder, these imperfections accumulate. On the other hand, we find a four-qubit quantum network that continuously implements a half-adder with a fidelity of $\approx 97.9\%$. Note that this is already better than the threshold for fault-tolerant classical computation using quantum gates [27].

Since optimizing the dynamics of a many-body system is normally an analytically intractable problem, one has to develop powerful numerical techniques. A fruitful approach is to view the many-body system as a quantum neural network that has to "learn" a desired quantum transformation [7,28–31], or, alternatively, to use powerful optimization algorithms [32–35]. Here we use differential evolution [36], a global optimization algorithm inspired by evolutional biology, which has been recently successfully applied in quantum control problems [34]. We therefore demonstrate its effectiveness also for WYW computing.

II. BACKGROUND

A. Quantum half-adder

Computers are able to perform vast computations, owing to discrete operations on bits called logic gates. Using basic

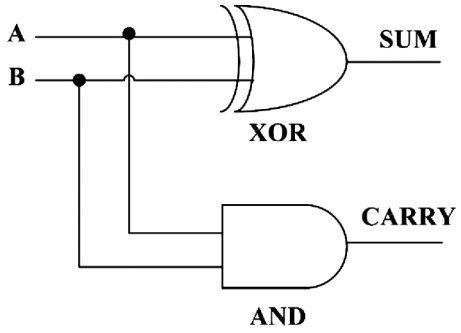

FIG. 1. A half-adder constructed with a XOR and AND gate.

logic gates in specific configurations, it is possible to construct more complex structures capable of performing arithmetic computation. Fundamental to the construction of computational arithmetic in a classical computer is the half-adder. The half-adder circuit is used for the addition of two bits and is a vital component of the more important adder circuit, which is used to add strings of binary digits together. The construction of a half-adder circuit requires the coupling of two simpler logic gates, the XOR and AND gate, whose design can be seen in Fig. 1. Using this configuration of gates, the addition of two binary digits can be represented with a sum and carry. The truth table is shown in Table I. Analogous to its classical counterpart, quantum half-adders are used for the addition of two quantum bits. From the gate decomposition in Fig. 1, it can be seen that the mapping from input to output in both the XOR and AND gates is many-to-one, and therefore irreversible. Moreover, in Fig. 1 each input bit is copied and sent to two different gates, an impossible operation in the quantum setting because of the no-cloning theorem. Since quantum gates are carried out using unitary operators, and thus must be reversible, it is necessary to introduce an ancillary qubit to construct a quantum half-adder. Therefore, the quantum circuit consists of three qubits: the two target qubits to be summed and a single control qubit, as well as two quantum logic gates: a Toffoli gate followed by a controlled-NOT (CNOT). The schematic of the circuit and the truth table can be seen in Fig. 2 and Table II, respectively, which can be interpreted as the following logical operation:

$$\hat{C}^{ab}_{\text{NOT}}\hat{T}^{abc}(|a\rangle\otimes|b\rangle\otimes|0\rangle) = (\hat{C}_{\text{NOT}}|a\rangle\otimes|b\rangle)\otimes|ab\oplus 0\rangle = |a\rangle\otimes|a\oplus b\rangle\otimes|ab\oplus 0\rangle, \quad (1)$$

where $\hat{T}$ is the Toffoli gate, $\hat{C}_{\text{NOT}}$ is the CNOT gate, $\otimes$ denotes tensor product, and $\oplus$ denotes addition modulo 2.

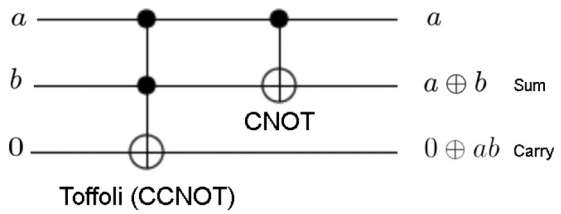

FIG. 2. A quantum half-adder constructed with a Toffoli and CNOT gate.

The sum and carry of this operation are stored in the second and third qubit, respectively. Likewise to a classical computer, a quantum adder can be constructed from quantum half-adders to provide a way to add two unknown quantum states together. These operations are the building blocks of *quantum arithmetics* [17–19], which is a key component (together with the quantum Fourier transform) for implementing Shor's algorithm, capable of factoring numbers in polynomial time—much faster than is possible on a classical computer [20].

B. Quantum gate learning

A quantum algorithm is a unitary operation that transforms the quantum state of a set of qubits into another quantum state. Physically, this corresponds to a quantum evolution via the time-dependent Schrödinger equation $i\frac{\partial U}{\partial t} = \hat{H}(t)U$, where U is a unitary operator capable of a specific transformation, governed by the underlying Hamiltonian $\hat{H}(t)$. If the Hamiltonian is unmodulated in time, namely, $\hat{H}(t) = \hat{H}$, then the unitary evolution of the system is simply related by $U = e^{(i/\hbar)\hat{H}t}$, and one may perform this evolution by setting up a system of qubits where the interactions are modeled by the Hamiltonian, H. Unfortunately, in physical implementations of quantum computing, the range of possible Hamiltonians is severely restricted. As a result of this restriction, the more complicated transformations are usually carried out by breaking the single dynamics of the system and representing the transformation as a series of simpler unitary operations, $U_1U_2\cdots U_N$, governed by interaction Hamiltonians, $H_1, H_2, \ldots, H_N$, that are experimentally achievable. To perform this sort of procedure, it is necessary to execute a series of external pulses to continuously modulate the system's dynamics, such that $\hat{H}(t)$ is piecewise constant to the values $H_1, H_2, \ldots, H_N$, respectively, at different intervals. Consequentially, this leads to possible errors from external interactions which accumulate over time and may affect the outcome of the computation, under the absence of error-correcting codes. Motivated by this, there have been different proposals to directly encode complicated unitary

TABLE I. Truth table of the half-adder.

Input		Output	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

TABLE II. Truth table of the quantum half-adder.

Input			Output		
A	B	C	A	B (sum)	C (carry)
0	0	0	0	0	0
0	1	0	0	1	0
1	0	0	1	1	0
1	1	0	1	0	1

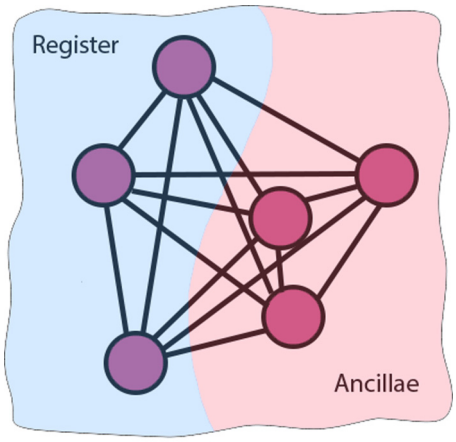

FIG. 3. An undirected graph representing a quantum network split into register and ancillae qubits.

operations into the hardware of a system. Indeed, in Ref. [7] it was shown that both the Toffoli gate and Fredkin gate can be carried out with high fidelity by unmodulated interactions, even when one is restricted to physical two-body couplings between qubits. The procedure for learning the correct physical Hamiltonian that reproduces the given target gate was called quantum gate learning [7,37], while the resulting architecture was called while-you-wait computing [7].

The idea of quantum gate learning is to encode a quantum operation into the unmodulated dynamics of a quantum network without having to provide external control pulses. This is done by constructing a *quantum network* consisting of both regional and ancillae qubits, and then optimizing their two-body interactions in such a way that a unitary gate operation is encoded in the natural reduced dynamics of the system, namely, into the regional qubits.

Consider a quantum network consisting of an undirected graph (V,E) of vertices V and links E as can be seen in Fig. 3. The vertices V are made up of single qubits and the links E correspond to the 2-local spin-coupling interactions between the qubits. The quantum network can be described by the following time-independent Hamiltonian:

$$\mathcal{H} = \sum_{(n,m)\in E}\sum_{\alpha,\beta} J_{nm}^{\alpha\beta}\frac{\sigma_n^\alpha\sigma_m^\beta}{4} + \sum_{n\in V}\sum_{\alpha} h_n^\alpha\frac{\sigma_n^\alpha}{2}, \qquad (2)$$

where σ_n^α, $\alpha = x,y,z$, are the Pauli matrices acting on qubit n, $J_{nm}^{\alpha\beta}$ is the interaction strength between the nth and mth qubit via $\alpha\beta$-spin-coupling interaction, and h_n^α is the strength of the external magnetic field on the nth qubit in the α direction. The idea is to optimize the parameters J and h of the time-independent Hamiltonian that governs the time evolution of the network of qubits in such a way as to maximize the fidelity between the time-evolution operator and the desired gate.

Note the importance of having ancillae qubits: without them the gate learning corresponds to the matrix logarithm of $U = e^{-i\mathcal{H}}$. Nonetheless, since the matrix logarithm is a multivalue function, for degenerate gates one can drastically simplify the resulting Hamiltonian and remove higher order interactions [38]. For nondegenerate gates, these matrix logarithms normally give rise to three-body interaction terms (or even of higher order), which are undesirable, as they unlikely appear naturally in physical systems and must be engineered via a superexchange mechanism [39]. The latter typically yields long gate times, during which the system may be highly affected by decoherence. On the other hand, by adding ancillae qubits, the degree of freedom to implement the gate with simpler interactions is expected to increase, and may give rise to the desired two-body interaction solutions. Different analytical methods have been proposed to find quantum networks that implement a target gate [40,41] using a certain amount of ancillary systems. However, these methods are normally far from optimal in terms of resource usage (i.e., number of ancillae). For instance, via the construction in [40] one normally requires NL ancillae qubits, where N are the number of qubits in the gate and L are the number of CNOT gates in the circuit. For example, since there are three qubits in the construction of a Toffoli gate and it is possible to decompose into five CNOT gates, the number of ancillae required for this method is 15. Since the quantum half-adder requires an additional CNOT gate, the number of ancillae qubits required is 18. Conversely, we will show that the quantum gate learning method of implementing a gate can allow for the construction of 2-local Hamiltonians with far fewer ancillae qubits.

Before tackling the problem at hand, it is necessary to define the optimization problem in an appropriate mathematical framework. Consider a network of qubits consisting of two composite subspaces: register qubits, R and ancillae qubits, A, as represented in Fig. 3. Let the initial quantum systems of Q and A be denoted as $|\psi_Q\rangle$ and $|\psi_A\rangle$, respectively. Note that the time evolution of the overall network is not equal to the tensor product of the individual evolution of both subsystems, displayed in Eq. (3). This is strictly due to the interactions between Q and A, and as such are not independent subsystems.

$$U(t)(|\psi_Q\rangle\otimes|\psi_A\rangle) \neq U_Q(t)|\psi_Q\rangle\otimes U_A(t)|\psi_A\rangle. \qquad (3)$$

To simplify the mathematics we introduce also the density operator formalism

$$|\psi\rangle\langle\psi| = |\psi_Q\rangle\langle\psi_Q|\otimes|\psi_A\rangle\langle\psi_A| = \rho_Q\otimes\rho_A = \rho, \qquad (4)$$

where ρ denotes the state of the overall network, and ρ_Q and ρ_A denote the state of subsystems Q and A, respectively. Using the time evolution governed by Eq. (2), a quantum channel, $\mathcal{E}_{J,h}$, can be defined that describes the system evolution and depends on both $J_{nm}^{\alpha\beta}$ and h_n^α, as well as ρ_A. This is done by first evolving the overall system ρ, then taking the partial trace over A—allowing interactions between both subsystems to be taken into account during the evolution. Mathematically, the time evolution of subsystem Q is described by the quantum channel

$$\mathcal{E}_{J,h}[\rho_Q] = \mathrm{Tr}_A(e^{-i\hat{H}}\rho_Q\otimes\rho_A e^{i\hat{H}}). \qquad (5)$$

It is this (normally nonunitary) operation that ultimately will be optimized to operate like a desired quantum gate.

C. Fidelity-based cost function

A similarity measure needs to be defined that will allow a direct comparison between the operation of the quantum channel and the ideal quantum gate G. Here we use the fidelity

metric defined below for this comparison:

$$F(G,\mathcal{E}_{J,h},\rho_Q) = \langle\psi_Q|G^\dagger\mathcal{E}_{J,h}[\rho_Q]G|\psi_Q\rangle = \mathrm{Tr}(G\rho_Q G^\dagger\,\mathcal{E}_{J,h}[\rho_Q]). \quad (6)$$

If the fidelity between both operations is close to 1, this demonstrates that the quantum channel and the ideal quantum gate have similar actions on the quantum state, $|\psi_Q\rangle$. It is possible that both the quantum channel and the ideal quantum gate have similar actions on a single state, $|\psi_Q^{(1)}\rangle$, but have completely different actions on every other quantum state, $|\psi_Q^{(i)}\rangle$. A better representation of the similarity between the operations is the average gate fidelity, which can simply be obtained by integrating Eq. (6) over all quantum states, $|\psi\rangle$. The average gate fidelity allows for the comparison of two quantum operations acting on all possible quantum states, providing a strong metric for similarity. It is defined as

$$\bar{F}(G,\mathcal{E}_{J,h}) = \int F(G,\mathcal{E}_{J,h},|\psi_Q\rangle\langle\psi_Q|)d\psi_Q, \quad (7)$$

where $d\psi$ is the uniform (Haar) integration over the possible states. The above Haar integral can be computed exactly to write [42,43]

$$\bar{F}(G,\mathcal{E}_{J,h}) = \frac{1}{D+1} + \frac{1}{D(D+1)}\sum_{i,j,k,l} G^*_{ik}\langle q_i|\mathcal{E}_{J,h}[|q_k\rangle\langle q_l|]|q_j\rangle G_{jl}, \quad (8)$$

where the states $|q_j\rangle$ form a basis of the Hilbert space Q, and D its dimension. Unfortunately, the numerical implementation of Eq. (8) requires an explicit sum over four different variables, which slows down the computation significantly in high-level programming languages. For this reason, a vectorized form of the average gate fidelity would be preferred, as a computer is highly efficient when dealing with matrix manipulation.

Vectorization of the average gate fidelity may be performed by first expressing Eq. (8) in terms of the Choi matrix, $\rho_\mathcal{E} = I\otimes\mathcal{E}|\Phi\rangle\langle\Phi|$, where $|\Phi\rangle$ is the maximally entangled quantum states, $\frac{1}{\sqrt{D}}\sum_i|q_i,q_i\rangle$. Expanding the Choi matrix into its matrix representation, the quantum channel can be expressed in terms of the Choi matrix as

$$\langle q_i|\mathcal{E}_{J,h}[|q_k\rangle\langle q_l|]|q_j\rangle = D\langle q_k,q_i|\rho_\mathcal{E}|q_l,q_j\rangle. \quad (9)$$

Substituting Eq. (9) into Eq. (8), and defining $|G\rangle = \sum_{ij}G_{ij}|q_j,q_i\rangle/\sqrt{D}$, the average gate fidelity can be expressed in a concise vectorized form, as

$$\bar{F}(G,\mathcal{E}) = \frac{1}{D+1} + \frac{D}{D+1}\langle G|\rho_\mathcal{E}|G\rangle. \quad (10)$$

This expression is more useful when $\rho_\mathcal{E}$ can be expressed in terms of the basis functions and ancillae states. This will require the explicit evaluation of the Choi matrix for the channel $\mathcal{E}$. Conclusively, by defining a new state, $|\Xi\rangle = |\Phi\rangle\otimes|\psi_A\rangle = \frac{1}{\sqrt{D}}\sum_i|q_i\rangle|q_i\rangle|\psi_A\rangle$, and expressing the time evolution of the system as $\tilde{U} = \hat{I}_Q\otimes e^{-it\mathcal{H}}$, where $\hat{I}_Q$ is the $D\times D$ identity matrix acting on subsystem Q, the final form of the Choi matrix can be found as $\rho_\mathcal{E} = \mathrm{Tr}_A[\tilde{U}|\Xi\rangle\langle\Xi|\tilde{U}^\dagger]$. Ultimately, the average gate fidelity can be fully vectorized into the following expression:

$$\bar{F}(G,\mathcal{E}_{J,h}) = \frac{1}{D+1} + \frac{D}{D+1}\langle\Xi|\tilde{U}^\dagger\,\rho^G\,\tilde{U}|\Xi\rangle, \quad (11)$$

where $\rho^G = |G\rangle\langle G|\otimes\hat{I}_A$ and $\hat{I}_A$ acts on the ancillae.

D. Differential evolution

For training neural networks [44], the first choice algorithm for optimizing the cost function is often the stochastic gradient descent (SGD) and its variations. For this reason, SGD was also used in [7] for obtaining a quantum network that accurately approximates a Toffoli gate. SGD has some benefits: it is fast and has the ability to escape, in principle, from some local minima. However, SGD rarely converges to the true global optimum but rather to a "good enough" optimum. This is particularly important in neural networks, since the true global optimum normally overfits the data. On the other hand, quantum networks are useful only if they can provide a very high-fidelity implementation of the target quantum operation, because of the remarkable precision required in quantum computation. For this reason, it is important to optimize the quantum network by finding a true global maximum of the fidelity Eq. (7). Fortunately, there exists a class of optimization algorithms that are well suited for dealing with the global optimiszation of many parameter, nonconvex functions. A good example of one of these algorithms is differential evolution (DE). DE is a robust, stochastic, population-based optimization algorithm first introduced by Storn and Price that can converge quickly to find the global optimum of functions that are nondifferentiable, noncontinuous, or have many local optima [36]. For that reason, DE will be the algorithm of choice to find the global optimum for the average gate fidelity subject to the spin couplings, J, and external magnetic field, h, parameters.

DE consists of four main steps: initialization, mutation, recombination, and selection. Initialization defines the search-space domain in which your parameters exist. Mutation and recombination introduces diversity and allows the parameter search space to be traversed. Selection ensures that the best parameters of each generation are kept within the population. There are three main control parameters that determine how fast the algorithm converges: the number of agents, N, the mutation factor, F, and the crossover rate, CR. These parameters are often chosen by trial and error and remain constant throughout the evolution of the algorithm. Unfortunately, this approach is not viable when dealing with computationally expensive functions, such as average gate fidelity, as it requires many evaluations of the function each iteration. An alternative approach, known as self-adaptive DE, is to vary F and CR after each generation of the algorithm. This has been shown to be an effective way of decreasing the likelihood of getting stuck in a local optima and also achieving better results in quantum control problems [34].

III. RESULTS

The design of the quantum network to encode the quantum half-adder consists of four qubits: three regional and one ancilla. There is no specific reason for only introducing one

ancilla, other than this is the minimum number of ancillae that can be used. As mentioned earlier, the reason for not using N qubits (in this case, three) to encode the N-dimensional gate is because three-body interaction terms are present in the analytical solution. This is demonstrated by taking the principal matrix logarithm of the half-adder gate G:

$$-i\frac{8}{\pi}\ln(G) = \hat{I}^{\otimes 3} - \sigma_1^Z - \sigma_2^X - \sigma_2^Y + \sigma_3^X - \sigma_2^X \otimes \sigma_3^X + \sigma_2^X \otimes \sigma_1^Z - \sigma_3^X \otimes \sigma_1^Z + \sigma_2^Y \otimes \sigma_1^Z - \sigma_2^X \otimes \sigma_3^X \otimes \sigma_1^Z - \sigma_3^X \otimes \sigma_2^Y \otimes \sigma_1^Z, \quad (12)$$

where the last two terms have three-body interactions.

The number of free parameters to optimize normally increases exponentially with the number of qubits considered in a network. In order to keep the discussion realistic, and also to minimize the number of parameters, cross-pairwise interactions between qubits will be ignored. This is preferable as these types of interactions are uncommon in nature. Therefore, the problem is simplified to only XX, YY, and ZZ pairwise interactions between qubits. As a result, the Hamiltonian for this network design can be extracted from the generalized form in Eq. (2), resulting in Eq. (13) below, and the optimization problem is reduced from a 48- to a 30-dimensional problem.

$$\begin{aligned}\mathcal{H} = \tfrac{1}{4}\big(&J_{12}^{XX}\sigma_1^X \otimes \sigma_2^X + J_{12}^{YY}\sigma_1^Y \otimes \sigma_2^Y + J_{12}^{ZZ}\sigma_1^Z \otimes \sigma_2^Z \\ &+ J_{13}^{XX}\sigma_1^X \otimes \sigma_3^X + J_{13}^{YY}\sigma_1^Y \otimes \sigma_3^Y + J_{13}^{ZZ}\sigma_1^Z \otimes \sigma_3^Z \\ &+ J_{14}^{XX}\sigma_1^X \otimes \sigma_4^X + J_{14}^{YY}\sigma_1^Y \otimes \sigma_4^Y + J_{14}^{ZZ}\sigma_1^Z \otimes \sigma_4^Z \\ &+ J_{23}^{XX}\sigma_2^X \otimes \sigma_3^X + J_{23}^{YY}\sigma_2^Y \otimes \sigma_3^Y + J_{23}^{ZZ}\sigma_2^Z \otimes \sigma_3^Z \\ &+ J_{34}^{XX}\sigma_3^X \otimes \sigma_4^X + J_{34}^{YY}\sigma_3^Y \otimes \sigma_4^Y + J_{34}^{ZZ}\sigma_3^Z \otimes \sigma_4^Z\big) \\ &+ \tfrac{1}{2}\big(h_1^X\sigma_1^X + h_1^Y\sigma_1^Y + h_1^Z\sigma_1^Z \\ &+ h_2^X\sigma_2^X + h_2^Y\sigma_2^Y + h_2^Z\sigma_2^Z \\ &+ h_3^X\sigma_3^X + h_3^Y\sigma_3^Y + h_3^Z\sigma_3^Z \\ &+ h_4^X\sigma_4^X + h_4^Y\sigma_4^Y + h_4^Z\sigma_4^Z\big). \end{aligned} \quad (13)$$

The search-space domain for each parameter was set bounded by the values $[-20,20]$, chosen to reflect the domain of the solution for the Toffoli gate [7]. The number of initial agents was first chosen to be 75, as recommended for the dimensionality of the problem by Pedersen [45], but was prone to getting stuck in local minima. For that reason, the number of initial agents was set to 156. The self-adaptive parameters were set to $\kappa = 0.1$, $\kappa_2 = 0.9$, $\mu_1 = 0.1$, and $\mu_2 = 0.1$, the same as those seen in the . successful implementation of self-adaptive DE on quantum control problems by Zahedinejad *et al.* [46].

After around 40 000 iterations of DE, optimization came to a halt at a fidelity of 0.9792, displayed by the optimization curve in Fig. 4, yielding the following parameters.

$$\begin{aligned}
J_{12}^{XX} &= 0.0124, & J_{12}^{YY} &= 0.0331, & J_{12}^{ZZ} &= 6.9978,\\
J_{13}^{XX} &= 0.0122, & J_{13}^{YY} &= -0.0272, & J_{13}^{ZZ} &= -17.456,\\
J_{14}^{XX} &= -0.0078, & J_{14}^{YY} &= -0.0021, & J_{14}^{ZZ} &= -0.9188,\\
J_{23}^{XX} &= -9.5369, & J_{23}^{YY} &= -11.163, & J_{23}^{ZZ} &= 6.9448,\\
J_{24}^{XX} &= 0.1043, & J_{24}^{YY} &= 0.0139, & J_{24}^{ZZ} &= 0.0693,
\end{aligned}$$

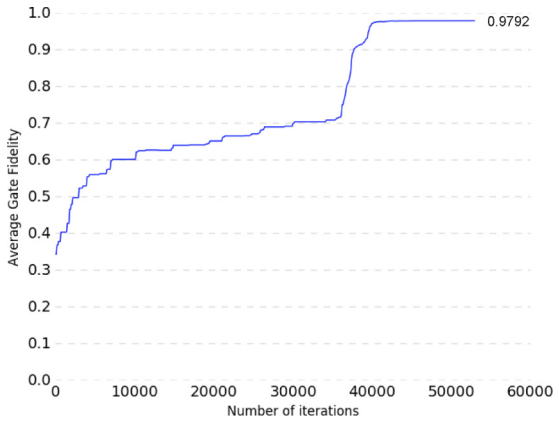

FIG. 4. The optimization curve for the average gate fidelity using differential evolution.

$$\begin{aligned}
J_{34}^{XX} &= -17.972, & J_{34}^{YY} &= -0.1634, & J_{34}^{ZZ} &= 0.2029,\\
h_1^X &= 0.0132, & h_1^Y &= 0.0084, & h_1^Z &= -14.935,\\
h_2^X &= -4.1712, & h_2^Y &= -2.2730, & h_2^Z &= 3.5019,\\
h_3^X &= -19.8358, & h_3^Y &= 4.7550, & h_3^Z &= -8.6666,\\
h_4^X &= -18.8257, & h_4^Y &= 0.0239, & h_4^Z &= 0.3716.
\end{aligned}$$

Interestingly, a high proportion of these parameters are close to zero. Motivated by this observation, a following optimization procedure was carried out. A top-down elimination approach was taken, by iteratively setting the weaker parameters to zero and then further optimizing the average gate fidelity using a local optimization algorithm, Broyden-Fletcher-Goldfarb-Shanno (BFGS). This fine-tuning is stopped when there is a significant drop in the fidelity. The optimal parameters are then the ones before this drop. Figure 5 illustrates the effect this process has on the average gate fidelity, ultimately removing redundant connections in the quantum network without general loss in fidelity. Sixteen parameters were removed, resulting in a much simplified network, visualized in Fig. 6, and the remaining interactions

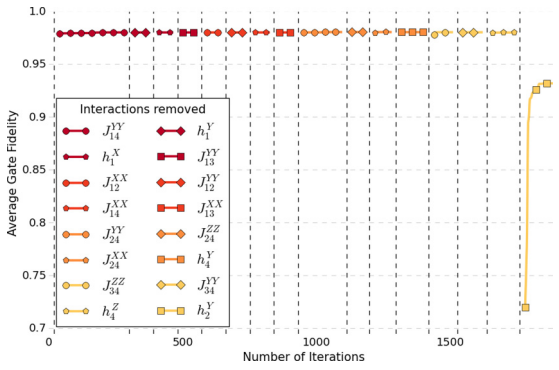

FIG. 5. The optimization curve for the average gate fidelity when the smallest parameters are iteratively forced to zero. At the iteration marked by vertical dotted lines a new interaction, specified in the legend, is removed. A local optimization algorithm is then carried out for a corresponding reduced set of parameters; see discussion in the text.

FIG. 6. Complex network transforms into a simplified network with much fewer interactions after iteratively setting small parameters to 0 and minimizing.

are as follows:

$$
\begin{aligned}
&J_{12}^{ZZ} = 6.9997, \quad J_{13}^{ZZ} = -17.396, \quad J_{14}^{ZZ} = 2.4969,\\
&J_{23}^{XX} = -9.5116, \quad J_{23}^{YY} = -11.060, \quad J_{23}^{ZZ} = 6.8754,\\
&J_{34}^{XX} = -13.281,\\
&h_1^Z = -14.924, \quad h_2^Z = 3.5001, \quad h_3^Z = -8.7004,\\
&h_2^X = -4.1277, \quad h_3^X = -22.231, \quad h_4^X = -18.786,\\
&h_2^Y = -2.2985, \quad h_3^Y = 4.5632.
\end{aligned}
$$

It is worth noting that the majority of the network is made up of ZZ interactions, which is obtainable in superconducting circuits, among others, as shown by Geller *et al.* [47]. Additionally, McKay *et al.* showed that it was possible to obtain $(XX + YY)$ interactions in a superconducting circuit [48], lending itself nicely for the interactions between the second and third qubits of this network. This makes the quantum half-adder with this network configuration almost fully implementable in a superconducting circuit. Unfortunately, it has yet to be known how to implement an XX interaction, seen between the third register qubit and the ancilla qubit.

The performance of the implemented quantum half-adder can be seen in Fig. 7, where its operation has been applied to many random quantum states, and compared to the ideal quantum half-adder. Around the optimal value, the operation of the implemented quantum half-adder is almost identical to the ideal gate. As a single parameter moves away from the optimal value, the fidelity vastly decreases, demonstrating the extreme nonconvexity of the average gate fidelity.

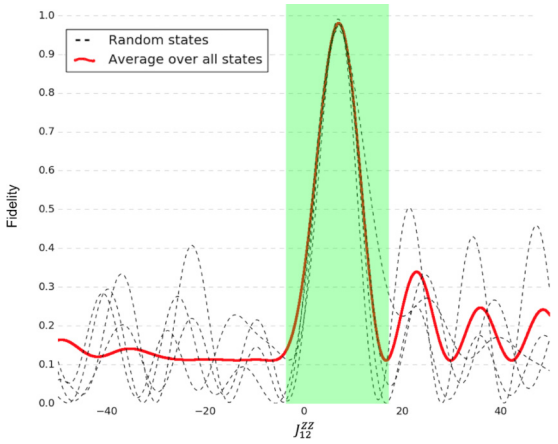

FIG. 7. The average gate fidelity and fidelity of random states given by the implemented quantum half-adder when varying one parameter, J_{12}^{ZZ}, around its optimal value.

IV. CONCLUSION

We optimized an unmodulated quantum network to operate as a continuous logic module for while-you-wait computing. Specifically, we obtained a four-qubit network which accurately acts as a quantum half-adder, a three-qubit operation used to construct quantum arithmetics. The optimal qubit-qubit interactions are found by mapping the problem into a global optimization of the formal average process fidelity, and we optimize the latter by adapting the *differential evolution* algorithm. Our results show that evolutionary algorithms are suitable to optimize over the complex parameter manifolds defined by the process fidelity, and provide a first step for the development of larger modules for full quantum arithmetics.

[1] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell *et al.*, Nature (London) **508**, 500 (2014).

[2] K. R. Brown, A. C. Wilson, Y. Colombe, C. Ospelkaus, A. M. Meier, E. Knill, D. Leibfried, and D. J. Wineland, Phys. Rev. A **84**, 030303 (2011).

[3] D. A. Lidar and T. A. Brun, *Quantum Error Correction* (Cambridge University Press, Cambridge, UK, 2013).

[4] C. Brif, R. Chakrabarti, and H. Rabitz, New J. Phys. **12**, 075008 (2010).

[5] M. H. Mohammady, M. Mohseni, and Y. Omar, New J. Phys. **18**, 015011 (2016).

[6] S. Boixo, T. F. Rønnow, S. V. Isakov, Z. Wang, D. Wecker, D. A. Lidar, J. M. Martinis, and M. Troyer, Nat. Phys. **10**, 218 (2014).

[7] L. Banchi, N. Pancotti, and S. Bose, npj Quantum Inf. **2**, 16019 (2016).

[8] L. Banchi, T. J. G. Apollaro, A. Cuccoli, R. Vaia, and P. Verrucchi, Phys. Rev. A **82**, 052321 (2010).

[9] A. Zwick, G. A. Álvarez, J. Stolze, and O. Osenda, Phys. Rev. A **84**, 022311 (2011).

[10] C. Ramanathan, P. Cappellaro, L. Viola, and D. G. Cory, New J. Phys. **13**, 103015 (2011).

[11] S. Lorenzo, T. J. G. Apollaro, A. Sindona, and F. Plastina, Phys. Rev. A **87**, 042313 (2013).

[12] L. Banchi, Eur. Phys. J. Plus **128**, 137 (2013).

[13] L. Campos Venuti, S. M. Giampaolo, F. Illuminati, and P. Zanardi, Phys. Rev. A **76**, 052328 (2007).

[14] L. Campos Venuti, C. Degli Esposti Boschi, and M. Roncaglia, Phys. Rev. Lett. **99**, 060401 (2007).

[15] D. Burgarth, arXiv:0704.1309.

[16] G. M. Nikolopoulos and I. Jex, *Quantum State Transfer and Network Engineering* (Springer, New York, 2014).

[17] V. Vedral, A. Barenco, and A. Ekert, Phys. Rev. A **54**, 147 (1996).

[18] P. Gossett, arXiv:quant-ph/9808061.
[19] R. Van Meter and K. M. Itoh, Phys. Rev. A **71**, 052320 (2005).
[20] P. W. Shor, in Proceedings of the 35th Annual Symposium on Foundations of Computer Science*(SFCS '94)* (IEEE Computer Society, Washington, DC, 1994), pp. 124–134.
[21] G. A. Barbosa, Phys. Rev. A **73**, 052321 (2006).
[22] S. Srivastava, H. Kino, and C. Joachim, Chem. Phys. Lett. **667**, 301 (2017).
[23] C. Wu and C. Cain, Physica E **59**, 243 (2014).
[24] D. Chatterjee and A. Roy, Prog. Theor. Exp. Phys. **2015**, 093A02 (2015).
[25] G. Dridi, R. Julien, M. Hliwa, and C. Joachim, Nanotechnology **26**, 344003 (2015).
[26] K. V. R. M. Murali, N. Sinha, T. S. Mahesh, M. H. Levitt, K. V. Ramanathan, and A. Kumar, Phys. Rev. A **66**, 022313 (2002).
[27] B. Cruikshank and K. Jacobs, Phys. Rev. Lett. **119**, 030503 (2017).
[28] Z. Rigui, J. Nan, and D. Qiulin, Neural Process. Lett. **24**, 261 (2006).
[29] M. Altaisky, arXiv:quant-ph/0107012.
[30] K. H. Wan, O. Dahlsten, H. Kristjánsson, R. Gardner, and M. Kim, npj Quantum Inf. **3**, 36 (2017).
[31] M. Schuld, I. Sinayskiy, and F. Petruccione, Quantum Inf. Process. **13**, 2567 (2014).
[32] R. Li, U. Alvarez-Rodriguez, L. Lamata, and E. Solano, Quantum Meas. Quantum Metrol. **4**, 1 (2017).
[33] U. Las Heras, U. Alvarez-Rodriguez, E. Solano, and M. Sanz, Phys. Rev. Lett. **116**, 230504 (2016).
[34] E. Zahedinejad, J. Ghosh, and B. C. Sanders, Phys. Rev. Lett. **114**, 200502 (2015).
[35] P. Palittapongarnpim, P. Wittek, E. Zahedinejad, S. Vedaie, and B. C. Sanders, Neurocomputing **268**, 116 (2017).
[36] R. Storn and K. Price, J. Global Optim. **11**, 341 (1997).
[37] L. Banchi, N. Pancotti, and S. Bose, arXiv:1607.06146.
[38] L. Innocenti, L. Banchi, A. Ferraro, S. Bose, and M. Paternostro, arXiv:1803.07119.
[39] S. Bravyi, D. P. DiVincenzo, D. Loss, and B. M. Terhal, Phys. Rev. Lett. **101**, 070503 (2008).
[40] D. Nagaj, Phys. Rev. A **85**, 032330 (2012).
[41] T. Cubitt, A. Montanaro, and S. Piddock, arXiv:1701.05182.
[42] M. A. Nielsen, Phys. Lett. A **303**, 249 (2002).
[43] L. Banchi, A. Bayat, P. Verrucchi, and S. Bose, Phys. Rev. Lett. **106**, 140501 (2011).
[44] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, MA, 2016).
[45] M. E. H. Pedersen, Good parameters for differential evolution, Technical Report HL1002 (Hvass Laboratories, 2010).
[46] E. Zahedinejad, J. Ghosh, and B. C. Sanders, Phys. Rev. Appl. **6**, 054005 (2015).
[47] M. R. Geller, E. Donate, Y. Chen, C. Neill, P. Roushan, and J. M. Martinis, Phys. Rev. A **92**, 012320 (2015).
[48] D. C. McKay, S. Filipp, A. Mezzacapo, E. Magesan, J. M. Chow, and J. M. Gambetta, Phys. Rev. Appl. **6**, 064007 (2016).