

# Synthesizing quantum circuits via numerical optimization

Timothée Goubault de Brugière<sup>1,3,4</sup>, Marc Baboulin<sup>1,3</sup>, Benoît Valiron<sup>2,3</sup>, and Cyril Allouche<sup>4</sup>

<sup>1</sup> University of Paris-Sud, Orsay, France

<sup>2</sup> CentraleSupélec, Gif sur Yvette, France

<sup>3</sup> Laboratoire de Recherche en Informatique, Orsay, France

<sup>4</sup> Atos Quantum Lab, Les Clayes-sous-Bois, France

**Abstract.** We provide a simple framework for the synthesis of quantum circuits based on a numerical optimization algorithm. This algorithm is used in the context of the trapped-ions technology. We derive theoretical lower bounds for the number of quantum gates required to implement any quantum algorithm. Then we present numerical experiments with random quantum operators where we compute the optimal parameters of the circuits and we illustrate the correctness of the theoretical lower bounds. We finally discuss the scalability of the method with the number of qubits.

**Keywords:** quantum circuit synthesis · numerical optimization · quantum simulation · quantum circuit optimization · trapped-ions technology

## 1 Introduction

Quantum computing, introduced and theorized about 30 years ago, is still in its infancy: current technological devices can only handle a few qubits. This new paradigm of computation however shows great promises, with potential applications ranging from high-performance computing [8] to machine learning and big data [9]. Quantum algorithms are usually described via quantum circuits, i.e., series of elementary operations in line with the technological specificity of the hardware. The mathematical formalism for quantum computation is the theory of (finite dimensional) Hilbert spaces: a quantum circuit is represented as a unitary operator [16], independently from the machine support on which the algorithm will be executed. Establishing a link between the unitary operators described as matrices and the unitary operators described as circuits is therefore essential, if only to better understand how to design new algorithms. Obtaining the matrix from the circuit can be done either by running the circuit on a quantum hardware (plus some tomography) or via a simulation on a classical computer [2,24]. Obtaining the circuit from the matrix is more complicated and fits into a more general problematic called *quantum compilation* i.e., how to translate a quantum operator described in an unknown form for the targeted hardware into

a sequence of elementary instructions understandable by the machine. Therefore converting a matrix into a circuit is in all circumstances a compilation step because no hardware can directly accept a matrix as input: this particular task of compilation is called *quantum circuit synthesis* and is the central concern of this work. This process must be as automated and optimized as possible in terms of conventional or quantum resources. This task is inevitably hard in the general case: the classical memory required to store the unitary matrix is exponential in the number of qubits and it has been shown that an exponential number of quantum gates is necessary to implement almost every operator [11]. As a result both classical and quantum complexity follow an exponential growth and any methods will be quickly limited by the size of the problem. Yet being able to synthesize quickly and/or optimally any quantum operator or quantum state on a few qubits is a crucial challenge that could be crucial for some applications. In the NISQC (Noisy Intermediate Scale Quantum Computer) era [18], compressing circuits as much as possible will be crucial. And in general, such a procedure can be integrated as a subtask in a peep-hole optimization framework.

#### *Our contribution*

The quantum circuits addressed in this paper correspond to the specific technology of trapped-ions quantum computer which requires using specific quantum gates. This technology holds great promise to cross the passage to scale: quantum circuits with trapped-ions technology have great fidelities and the qubits have a long decoherence time. In other words the noise in the results is low and long time computations can be performed before the system interferes with the environment and loses its information. Moreover the particular architecture of trapped-ions quantum circuits makes the problem simpler for numerical optimization because as we will see it only involves the optimization of continuous real parameters. We provide a simple optimization framework and we use it to synthesize generic quantum operators. More specifically, we derive a lower bound on the number of entangling gates necessary to implement any quantum algorithm and propose numerical experiments that confirm the correctness of this bound.

#### *Plan of the paper*

In Section 2, we give the main definitions related to quantum circuits and quantum gates and we summarize the state of the art in quantum circuit synthesis. Then in Section 3 we develop in more details the modeling of a circuit as a parameterized topology and the question of the minimum-sized topology necessary to implement any operator. In Section 4 we introduce a generic optimization framework formalizing the circuit synthesis as a classical optimization problem. In Section 5 we apply this framework to optimally synthesize generic unitary matrices with the trapped-ions natural set of gates. We also discuss the scalability of this method and how we can naturally trade the optimality of the final quantum circuit for shorter computational time. We conclude in Section 6.

### Notations

Throughout this paper we will use the following notations.  $\mathcal{U}(n)$  denotes the set of unitary matrices of size  $n$ , i.e.  $\mathcal{U}(n) = \{M \in \mathbb{C}^{n \times n} \mid M^\dagger M = I\}$ , where  $I$  is the identity matrix and  $M^\dagger$  is the conjugate transpose of the matrix  $M$ . The special unitary group  $S\mathcal{U}(n)$  is the group of  $n \times n$  unitary matrices with determinant 1.  $\|x\| = \sqrt{x^\dagger x}$  refers to the Euclidean norm of a vector  $x \in \mathbb{C}^n$  and  $A \otimes B$  denotes the Kronecker product [5] of two matrices  $A$  and  $B$ .

## 2 Background and state of the art

### 2.1 Main notions in quantum circuits

The basic unit of information in quantum information is the quantum bit and is formalized as a complex linear superposition of two basis states, usually called the state '0' and the state '1'. More generally when manipulating a set of  $n$  qubits we modify a quantum state

$$|\psi\rangle = \sum_{\mathbf{b} \in \{0,1\}^n} \alpha_{\mathbf{b}} |\mathbf{b}\rangle$$

which is a normalized complex linear superposition of all the possible  $n$ -bitstring values. So a quantum state on  $n$  qubits can be written as a unit vector in  $\mathbb{C}^{2^n}$  and by the laws of quantum physics any operation on this quantum state can be represented as a left-multiplication of its vector representation by a unitary matrix  $U \in \mathcal{U}(2^n)$ , if we except the measurement operation.

Quantum algorithms are described via quantum circuits—the quantum analog of logical circuits. An example is given in Figure 1. Each horizontal wire corresponds to one qubit and quantum gates—represented as boxes, or vertical apparatus—are sequentially applied from left to right to different subsets of qubits resulting in a more complex unitary operator. Matrix multiplication  $BA$  of two operators  $A$  and  $B$  enables to apply sequentially the operator  $A$  then  $B$ . To compose two operators  $A$  and  $B$  acting on different qubits we use the Kronecker product  $\otimes$ .

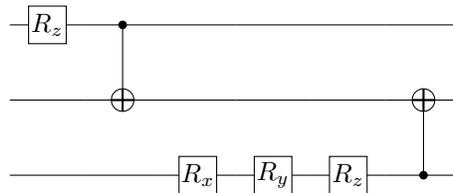


Fig. 1: Example of quantum circuit

For a given technology not all quantum gates are realizable and only a few of them are directly implementable: the so-called elementary gates. For

example, we have the Hadamard gate  $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ , or the one-qubit rotations of angle  $\theta \in \mathbb{R}$  along the x, y and z axis respectively defined by  $R_x(\theta) = \begin{pmatrix} \cos(\theta) & -i \sin(\theta) \\ -i \sin(\theta) & \cos(\theta) \end{pmatrix}$ ,  $R_y(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$ ,  $R_z(\theta) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}$ . Fortunately some subsets of elementary gates have been shown to be “universal”, which means that any quantum operator can be implemented by a quantum circuit containing only gates from this set. Depending on the technology used, the universal sets available will be different. With superconducting qubits or linear optics, the natural set of gates is the set of one-qubit gates  $SU(2)$ , also called local gates because they act on one qubit only, combined with the entangling CNOT (Controlled-NOT) gate which is a NOT gate controlled by one qubit. For instance, in the 2-qubits case, the CNOT gate controlled by the first qubit and

applied to the second one can be represented by the matrix  $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ .

In this paper we focus on the technology of trapped ions which uses a different universal set of gates, the available gates are:

- local  $R_z$  gates,
- global  $R_x$  gates i.e local  $R_x$  are applied to every qubit with the same angle,
- the entangling Mølmer–Sørensen gate (MS gate) defined by

$$MS(\theta) = e^{-i\theta(\sum_{i=1}^n \sigma_x^i)^2/4}.$$

where  $\sigma_x^i$  is the operator  $X$  applied to the  $i$ -th qubit.

Any quantum operator has an abstract representation using a unitary matrix and it is essential to be able to switch from a quantum operator given by a quantum circuit to a quantum operator given by its unitary matrix and vice-versa. From a quantum circuit to a unitary matrix this is the problem of the *simulation of a quantum circuit*. Finding a quantum circuit implementing a unitary matrix is the problem of the *synthesis of a quantum circuit* which is the central concern of this paper. We distinguish two different synthesis problems: the first one consists in implementing a complete unitary matrix and the second one consists in preparing a specific quantum state as output of the circuit applied to the state  $|000\dots 00\rangle$ : this is the *state preparation problem*.

## 2.2 State of the art in quantum circuit synthesis

Most quantum algorithms are still designed by hand [6,22] even though some circuits have been found via automatic processes [12]. For the automatic synthesis of quantum circuits there are mainly algebraic methods: we can mention the Quantum Shannon Decomposition (QSD) that gives the lowest number of gates in the general case [20] or the use of Householder matrices to achieve the synthesis in a much lower time [4]. The (H,T) framework is used for one-qubit synthesis

[10] although it can be extended to the multi-qubits case [3]. For the particular case of state preparation - the synthesis of one column of the quantum operator - we have devices using multiplexors [15] or Schmidt decomposition [17].

Although the asymptotic complexity of the methods has significantly decreased with the years, some progress can still be made. The motivation behind this article is to use well-known numerical optimization methods in hope of reducing at its maximum the final quantum resources. Using heuristics or classical optimization methods to synthesize circuits is not new. The BFGS (Broyden-Fletcher-Goldfarb-Shanno) algorithm [25] has already been used to synthesize trapped-ions circuits [14] and machine learning techniques have been used in the case of photonic computers [1]. Genetic algorithms have also been used in a general context [13] or for the specific IBM quantum computer [12]. However, these works are purely experimental: the overall optimality of their solution is not discussed.

We tackle the problem of the optimality of the solution by building on the work in [21] that provides a theoretical lower bound of the number of entangling gates necessary in the quantum circuit synthesis problem. The idea is to count the number of degrees of freedom in a quantum circuit and show that this number has to exceed a certain threshold to be sure that an exact synthesis is possible for any operator. To our knowledge numerical methods have not been used in order to address the issue of the lower bound and the more general issue of the minimum quantum resources that are necessary to synthesize a quantum circuit.

### 3 Lower bounds for the synthesis of trapped-ions quantum circuits

The problem of computing the minimal number of quantum gates necessary to implement an operator remains open. Knill [11] showed that for the entire set of quantum operators (up to a zero measure set) we need an exponential number of gates and using a polynomial number of gates is as efficient as using a random circuit in the general case. The special case of circuits built from  $\{SU(2), CNOT\}$  has been analyzed in [21], where quantum circuits are modeled as instantiations of circuit topologies consisting of constant gates and parameterized gates. The unspecified parameters are the degrees of freedom (DOF) of the topology. For instance the circuit given in Figure 1 can be considered as a topology with at most 4 degrees of freedom (one for each rotation) and giving precise angles to the rotations is an instantiation of the topology. As a consequence a topology with  $k$  degrees of freedom can be represented by a smooth function

$$f : \mathbb{R}^k \rightarrow \mathcal{U}(2^n) \tag{1}$$

that maps the values of angles to the space of unitary matrices of size  $2^n$ .

We are interested in the image of the function  $f$ . If a topology  $f$  on  $n$  qubits can implement any  $n$ -qubits operator, i.e., for any operator  $U$  on  $n$  qubits there exists a vector of angles  $x$  such that  $f(x) = U$ , then we say that the topology

is universal. Now what is the minimum number of gates necessary to obtain a universal topology ? The best lower bound for this minimum in the case of  $\{CNOT, SU(2)\}$  circuits is given in [21]. In this section, we derive a lower bound in the case of trapped-ions circuits using a similar reasoning.

**Theorem 1.** *A topology composed of one-qubit rotations and parameterized MS gates cannot be universal with fewer than  $\lceil \frac{4^n - 3n - 1}{2n + 1} \rceil$  MS gates.*

*Proof.* First we use Sard's theorem [7] to claim that the image of  $f$  is of measure 0 if  $k = \#DOF < \dim(\mathcal{U}(2^n))$ . Hence to be sure that we can potentially cover the whole unitary space we need

$$\#DOF \geq \dim(\mathcal{U}(2^n)) = 4^n \quad (2)$$

Next we give a normal form to trapped-ion circuits in order to count the number of DOF. MS gates operate on all qubits, they are diagonal in the basis  $H^{\otimes n} = \bigotimes_{i=1}^n H$  obtained by applying an Hadamard gate to each qubit, the so-called " $|+\rangle/|-\rangle$ " basis. We have

$$MS(\theta) = H^{\otimes n} \times D(\theta) \times H^{\otimes n} \quad (3)$$

with

$$D(\theta) = \text{diag}([e^{(n - \text{Hamm}(i))^2 \times \theta}]_{i=0..2^n-1}) \quad (4)$$

where  $\text{Hamm}$  is the Hamming weight in the binary alphabet.

First we can merge the Hadamard gates appearing in Equation (3) with the local gates so that we can consider that our circuits are only composed of local gates and diagonal gates given by Equation (4). Then we can write each local gate  $U$  as

$$U = R_z(\alpha) \times R_x(-\pi/2) \times R_z(\beta) \times R_x(\pi/2) \times R_z(\gamma) \quad (5)$$

where  $\alpha, \beta, \gamma$  parameterize the unitary matrix  $U$ . Because the MS gates are now diagonal we can commute the first  $R_z$  so that it merges with the next local unitary matrices. By doing this until we reach the end of the circuit we finally get a quantum circuit for trapped-ions hardware with the following basic subcircuit:

- a layer of local  $R_z$  gates,
- a layer of global  $R_x(\pi/2)$  gates,
- a layer of local  $R_z$  gates,
- a layer of global  $R_x(-\pi/2)$  gates,
- an MS gate (given in its diagonal form)

This subcircuit is repeated  $k$  times for a circuit with  $k$  MS gates. Ultimately, the circuit ends with a layer of rotations following the decomposition (5) on each qubit. An example of a quantum circuit on 3 qubits with 2 MS gates is given in Figure 2. The angles of the  $R_z$  rotations are omitted for clarity. The only parameters of such generic circuits are:

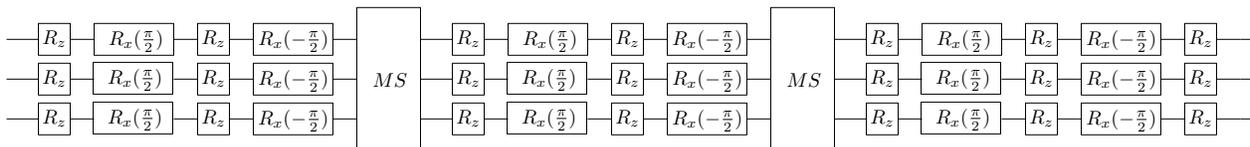


Fig. 2: Generic quantum circuit on 3 qubits for the trapped-ions technology

- the angles of the  $R_z$  rotations,
- the number and the angles of the MS gates.

Each elementary rotation (around the  $x, y, z$  axis) is parameterized by one angle so it can only bring one additional degree of freedom, as the MS gates if they are parameterized. Including the global phase, a circuit containing  $k$  parameterized MS gates can have at most  $(2n + 1) \times k + 3n + 1$  DOF. In the example given figure 2 the topology has 24 DOF. To reach universality we must verify equation (2), which leads to the lower bound

$$\#MS \geq \left\lceil \frac{4^n - 3n - 1}{2n + 1} \right\rceil.$$

This proof easily transposes to the state preparation problem with a few changes:

- a quantum state is completely characterized by  $2^{n+1} - 2$  real parameters,
- starting from the state  $|0\rangle^{\otimes n}$  we can only add one degree of freedom to each qubit on the first rotations because the first  $R_z$  result in an unnecessary global phase.

Consequently the total number of DOF a topology on  $n$  qubits with  $k$  MS gates can have is at most  $(2n + 1)k + 2n$ . We get the lower bound

$$\#MS \geq \left\lceil \frac{2^{n+1} - 2n - 2}{2n + 1} \right\rceil.$$

To our knowledge this is the first calculus of a lower bound in the context of trapped-ions circuits. In the next section we propose an algorithm based on numerical optimization that achieves accurate circuit synthesis using a number of gates corresponding to the above lower bounds. This gives a good indication of the tightness of these bounds.

## 4 The optimization framework

Given a function  $f$  representing a topology on  $n$  qubits, finding the best possible synthesis of a unitary matrix  $U$  with the topology  $f$  can be reformulated as solving

$$\arg \min_x \|f(x) - U\| := \arg \min_x g(x),$$

where  $\|\cdot\|$  is an appropriate norm. We decompose the cost function  $g$  as

$$g(x) = \frac{1}{k} \sum_{i=1}^k \|f(x) |e_{\phi(i)}\rangle - u_{\phi(i)}\|,$$

where  $\phi$  is a permutation of  $\llbracket 1, 2^n \rrbracket$  and  $u_j$  denotes the  $j$ -th column of  $U$ . In other words we generalize our problem to the synthesis of  $k$  given columns of our operator  $U$ . For simplicity now we can only study the case where  $k = 1$ : this is the state preparation problem.

Our goal is to rely on various algorithms for non linear optimization. We choose the norm to be the Euclidean norm so that with this choice of norm we have a simple expression of the cost error:

$$g(x) = 2 \times (1 - \operatorname{Re}(\langle 0 | f(x)^\dagger | \psi \rangle)). \quad (6)$$

Hence the cost to compute the error function is equivalent to the simulation cost of the circuit. Starting from the state  $|\psi\rangle$  we simulate the circuit  $f(x)^\dagger$ , then the real part of the first component gives the error. Many methods for simulating a quantum circuit have been designed and we can rely on them to perform efficient calculations.

Since  $g$  is  $C^\infty$  we can use more performant optimization algorithms if we can compute the gradient and if possible the Hessian. To compute the gradient, we need to give a more explicit formula for  $g(x)$ . For some  $j$  we have

$$\frac{\partial}{\partial x_j} g(x) = -2 \frac{\partial}{\partial x_j} \operatorname{Re}(\langle 0 | f(x)^\dagger | \psi \rangle),$$

and by linearity we get

$$\frac{\partial}{\partial x_j} g(x) = -2 \operatorname{Re} \left( \frac{\partial}{\partial x_j} \langle 0 | f(x)^\dagger | \psi \rangle \right).$$

Let us suppose we have  $K$  gates in our circuit. Then we can write

$$f(x)^\dagger = \prod_{i=1}^K A_i$$

where  $(A_i)_{i=1..K}$  is the set of gates on  $n$  qubits that compose the circuit  $f$ . In all circuits encountered the parameterized gates are of the form  $e^{i\theta\Omega}$  where  $\Omega$  can be either  $X, Y, Z$  (tensored with the identity if necessary) or more complex hermitians (see the MS gate for instance),  $\theta$  is the parameter of the gate. We assume that two gates are not parameterized by the same variable. Let  $k_1$  be the index of the gate depending on  $x_j$ , then we have

$$\begin{aligned}
\frac{\partial}{\partial x_j} \langle 0 | f(x)^\dagger | \psi \rangle &= \frac{\partial}{\partial x_j} \langle 0 | \prod_{i=1}^{k_1-1} A_i \times e^{i \times x_j \times \Omega} \times \prod_{i=k_1+1}^K A_i | \psi \rangle \\
&= \langle 0 | \prod_{i=1}^{k_1-1} A_i \times \frac{\partial}{\partial x_j} e^{i \times x_j \times \Omega} \times \prod_{i=k_1+1}^K A_i | \psi \rangle \\
&= i \times \langle 0 | \prod_{i=1}^{k_1-1} A_i \times \Omega \times A_{k_1} \times \prod_{i=k_1+1}^K A_i | \psi \rangle
\end{aligned}$$

Therefore we have a simple expression for the partial derivatives

$$\frac{\partial}{\partial x_j} g(x) = 2 \operatorname{Im} \left( \langle 0 | \prod_{i=1}^{k_1-1} A_i \times \Omega \times A_{k_1} \times \prod_{i=k_1+1}^K A_i | \psi \rangle \right).$$

To compute the whole gradient vector we propose the following algorithm:

---

**Algorithm 1** Computing the gradient of the cost function

---

**Require:**  $n > 0, |\psi\rangle \in \mathbb{C}^{2^n}, f : \mathbb{R}^k \rightarrow \mathcal{U}(2^n)$ ,  
**Ensure:** Computes  $\partial f$

```

 $\langle \psi_1 | \leftarrow \langle 0 |$ 
 $|\psi_2\rangle \leftarrow f(x)^\dagger |\psi\rangle$ 
 $m \leftarrow 1$ 
//  $N$  is the total number of gates in the circuit
for  $i = 1, N$  do
  //  $A_i$  is the  $i$ -th gate in the circuit
  if  $A_i$  is parameterized then
    //  $H_i$  is the Hamiltonian associated to the gate  $A_i$ 
     $df[m] \leftarrow 2 \operatorname{Im} (\langle \psi_1 | H_i | \psi_2 \rangle)$ 
     $m \leftarrow m + 1$ 
  end if
   $\langle \psi_1 | \leftarrow \langle \psi_1 | A_i$ 
   $|\psi_2\rangle \leftarrow A_i^\dagger |\psi_2\rangle$ 
end for

```

---

Therefore computing the gradient is equivalent to simulating two times the circuit. On total we need 3 circuit simulations to compute the error and the gradient.

Note that we need to store two vectors when computing the gradient. If  $k > 2^{n-1}$  then we use more memory than if we have only stored the entire matrix. As memory is not a big concern here compared to the computational time, we keep this framework even for  $k = 2^n$  for example.

## 5 Numerical experiments

The experiments have been carried out on one node of the QLM (Quantum Learning Machine) located at ATOS/BULL. This node is a 24-core Intel Xeon(R) E7-8890 v4 processor at 2.4 GHz. Our algorithm is implemented in C with a Python interface and has been executed on 12 cores using OpenMP multithreading. The uniform random unitary matrices are generated according to the Haar’s measure [23]. For the numerical optimization we use the BFGS algorithm provided in the SciPy [19] package.

### 5.1 Synthesizing generic circuits

A clear asset of trapped-ion technology is that there is no notion of topology. Contrary to superconducting circuits where we have to deal with the placement of the CNOT gates, we just have to minimize the number of MS gates to optimize the entangling resources. Local rotations are less expensive to realize experimentally [14]. So, as a first approach, we can always apply one-qubit gates on every qubits between two MS gates such that a quantum circuit for trapped ions always follow a precise layer decomposition.

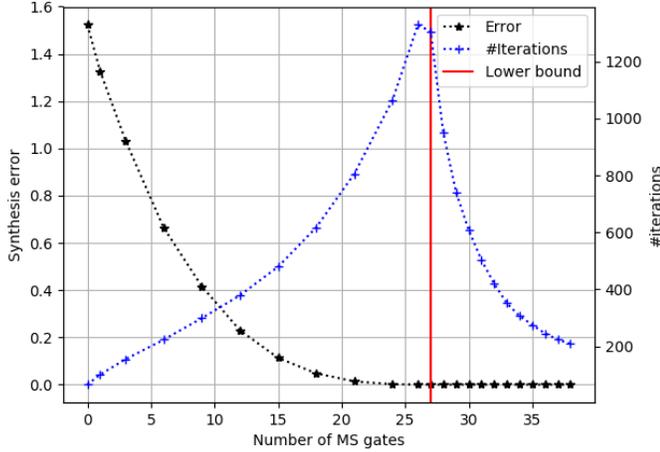
For 50 random unitary matrices on  $k \in \{2, 3, 4\}$  qubits, and quantum states on  $k \in \{2, 3, 4, 5, 6, 7\}$  qubits, we execute Algorithm 1 with circuits containing various numbers of parameterized MS gates. The stopping criterion for the optimization is the one chosen in SciPy i.e., the norm of the gradient must be below  $10^{-5}$ . We repeat the optimization process several times per matrix with different starting points to maximize our chance to reach a global minimum. Then we plot, for various numbers of MS gates,

- the error expressed in Formula (6), maximized over the sample of matrices,
- the number of iterations needed for convergence, averaged over the sample of matrices.

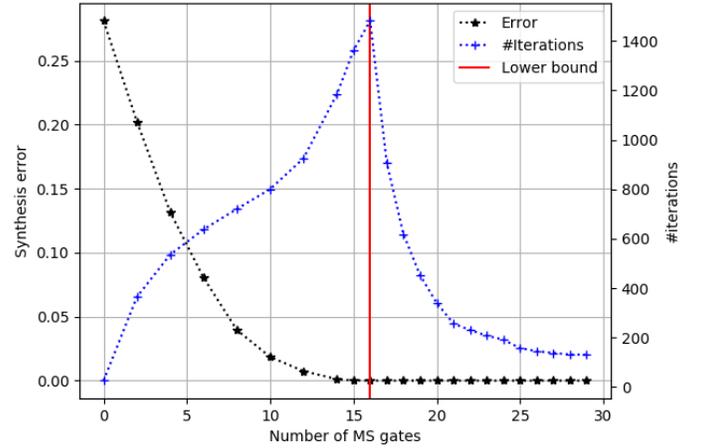
We summarize our results in Figures 3a and 3b corresponding respectively to the circuit synthesis problem on 4 qubits and the state preparation problem on 7 qubits. The results obtained for lower number of qubits are similar. The amount of time required to perform such experiments for larger problems is too important (more than a day).

For both graphs, we observe an exponential decrease of the error with the number of MS gates. This strong decrease shows that there is a range of MS gates count for which we have an acceptable accuracy without being minimal. Although we can save only a small number of MS gates by this trade, we conjecture that for a given precision the number of saved MS gates will increase with the number of qubits. In other words if we want to synthesize an operator up to a given error, the range of “acceptable” number of MS gates will increase with the number of qubits.

What is the experimental lower bound ? Interestingly, the number of iterations is a better visual indicator of this lower bound : once the exact number of DOF is



(a) 4-qubits Quantum circuit synthesis problem.



(b) 7-qubits State preparation problem.

Fig. 3: Evolution of the synthesis error/number of iterations with the number of MS gates.

reached, we add redundant degrees of freedom into the optimization problem. We make the hypothesis that this leads to a bigger search space but with many more global minima. Hence the search is facilitated and the algorithm can converge in fewer iterations. So the peak of the number of iterations corresponds to the experimental lower bound, which is confirmed by the computed errors.

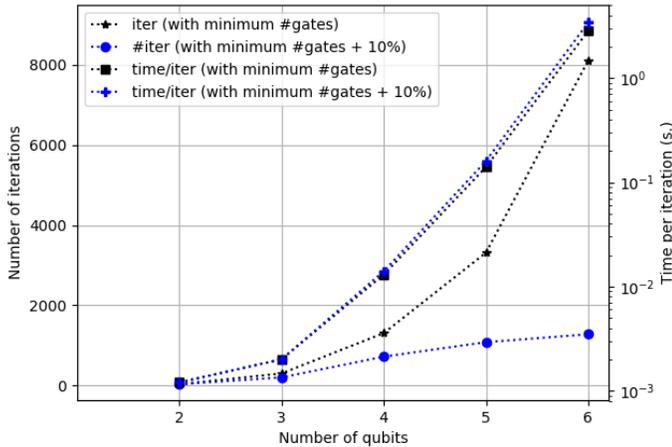
The experimental lower bounds follow the formulas given in Section 3 except for the 2-qubits quantum circuit synthesis case where we need 1 more MS gate to reach universality. This gives strong indications about the correctness of the theoretical lower bounds and the relevance of the approach of counting the degrees of freedom to estimate the resources necessary for implementing a quantum operator.

## 5.2 Tradeoff quantum/classical cost

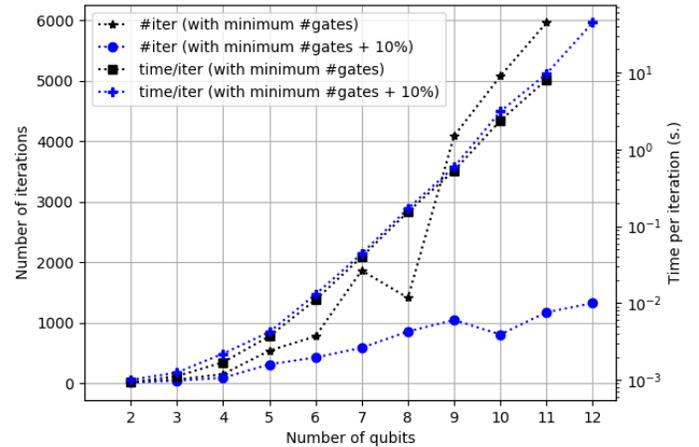
In the previous experiments, we could synthesize unitary operators on 6 qubits in 7 hours and quantum states on 11 qubits in 13 hours, both with circuits of optimal size. In the graphs plotted in Figures 3a and 3b, we also observe an exponential decrease of the number of iterations after we have reached the optimal number of MS gates. We can exploit this behavior if we want to accelerate the time of the synthesis at the price of a bigger circuit. By adding only a few more MS gates the number of iterations can be significantly reduced, allowing us to address potentially bigger problems in the synthesis. In Figure 4 we show, for different problem sizes, how the iteration count and the time per iteration evolve when

we add 10% more MS gates to the optimal number of gates. We observe that the number of iterations is reduced at most by a factor 5 in the case of 6 qubits for generic operators and 11 qubits for quantum states. More importantly, with such an augmentation in the number of MS gates, the number of iterations only slightly increases, almost linearly, in contrary to circuits of optimal size where the number of iterations increases exponentially. This means that the time per iteration, also increasing exponentially with the number of qubits, is the main limiting factor in the good scalability of the method. Thus any improvement in the classical simulation of a quantum circuit (i.e., reducing the time per iteration) will lead to significant acceleration in the synthesis time.

Finally in our experiments we achieved circuit synthesis on 6 qubits in about an hour, and a state preparation on 11 qubits in about 3 hours. Despite this sacrifice in the quantum cost (since we use more gates), this is still to our knowledge the best method to synthesize generic quantum circuits and quantum states for the trapped-ions technology.



(a) Quantum circuit synthesis problem.



(b) State preparation problem.

Fig. 4: Number of iterations and time per iteration for different problem sizes.

## 6 Conclusion

We have explained how a numerical optimization algorithm can be used to synthesize generic quantum circuits adapted to the trapped-ions technology. We

have provided a simple algorithm to compute the error and the gradient such that the scalability of our method relies on the simulation cost of the circuits we want to optimize. We have also highlighted a possible tradeoff between the classical time necessary to perform the synthesis and the final size of the circuits resulting in more flexibility in the application of the framework. Finally we have shown that the lower bounds computed experimentally with this framework follow the theoretical results with the hope that it would help for future formal proofs.

As future work we plan to extend our analysis to specific quantum operators for which we expect shorter circuits. We also plan to extend this analysis to  $\{CNOT, SU(2)\}$  based circuits, especially in order to answer the following questions : is the lower bound given in [21] a tight one ? Is there a universal topology that reaches that lower bound ? We also investigate a way to efficiently compute the Hessian in order to use more complex optimization methods and we plan to address problems with more qubits thanks to GPU and distributed computing.

## Acknowledgement

This work was supported in part by the French National Research Agency (ANR) under the research project SoftQPRO ANR-17-CE25-0009-02, and by the DGE of the French Ministry of Industry under the research project PIA-GDN/QuantEx P163746-484124.

## References

1. Arrazola, J.M., Bromley, T.R., Izaac, J., Myers, C.R., Bradler, K., Killoran, N.: Machine learning method for state preparation and gate synthesis on photonic quantum computers. *Quantum Science and Technology* (2018)
2. Bravyi, S., Browne, D., Calpin, P., Campbell, E., Gosset, D., Howard, M.: Simulation of quantum circuits by low-rank stabilizer decompositions. *arXiv preprint arXiv:1808.00128* (2018)
3. Giles, B., Selinger, P.: Exact synthesis of multiqubit clifford+ $t$  circuits. *Phys. Rev. A* **87**, 032332 (Mar 2013)
4. Goubault de Brugière, T., Baboulin, M., Valiron, B., Allouche, C.: Quantum circuits synthesis using householder transformations (2018), submitted
5. Graham, A.: *Kronecker products and matrix calculus with application*. Wiley, New York (1981)
6. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*. pp. 212–219. STOC '96, ACM, New York, NY, USA (1996)
7. Guillemin, V., Pollack, A.: *Differential topology*, vol. 370. American Mathematical Soc. (2010)
8. Harrow, A.W., Hassidim, A., Lloyd, S.: Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **103**, 150502 (2009)
9. Kerenidis, I., Prakash, A.: Quantum recommendation systems. In: Papadimitriou, C.H. (ed.) *8th Innovations in Theoretical Computer Science Conference, ITCS'17*. LIPIcs, vol. 67, pp. 49:1–49:21. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2017)

10. Kliuchnikov, V., Maslov, D., Mosca, M.: Fast and efficient exact synthesis of single-qubit unitaries generated by Clifford and T gates. *Quantum Information and Computation* **13**(7-8), 607–630 (2013)
11. Knill, E.: Approximation by quantum circuits. arXiv preprint quant-ph/9508006 (1995)
12. Li, R., Alvarez-Rodriguez, U., Lamata, L., Solano, E.: Approximate quantum adders with genetic algorithms: An ibm quantum experience. arXiv preprint arXiv:1611.07851 (2016)
13. Lukac, M., Perkowski, M., Goi, H., Pivtoraiko, M., Yu, C.H., Chung, K., Jeech, H., Kim, B.G., Kim, Y.D.: Evolutionary approach to quantum and reversible circuits synthesis. *Artificial Intelligence Review* **20**(3-4), 361–417 (2003)
14. Martinez, E.A., Monz, T., Nigg, D., Schindler, P., Blatt, R.: Compiling quantum algorithms for architectures with multi-qubit gates. *New Journal of Physics* **18**(6), 063029 (2016)
15. Mottonen, M., Vartiainen, J.J.: Decompositions of general quantum gates. arXiv preprint quant-ph/0504100 (2005)
16. Nielsen, M.A., Chuang, I.L.: *Quantum Computation and Quantum Information*. Cambridge University Press (2011)
17. Plesch, M., Brukner, i.c.v.: Quantum-state preparation with universal gate decompositions. *Phys. Rev. A* **83**, 032302 (Mar 2011)
18. Preskill, J.: Quantum Computing in the NISQ era and beyond. *Quantum* **2**, 79 (Aug 2018)
19. Python: Scientific Computing Tools for Python (SciPy). <https://www.scipy.org>
20. Shende, V.V., Bullock, S.S., Markov, I.L.: Synthesis of quantum-logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **25**(6), 1000–1010 (2006)
21. Shende, V.V., Markov, I.L., Bullock, S.S.: Minimal universal two-qubit controlled-not-based circuits. *Physical Review A* **69**(6), 062321 (2004)
22. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review* **41**(2), 303–332 (1999)
23. Stewart, G.: The efficient generation of random orthogonal matrices with an application to condition estimators. *SIAM Journal on Numerical Analysis* **17**(3), 403–409 (1980)
24. Vidal, G.: Efficient classical simulation of slightly entangled quantum computations. *Phys. Rev. Lett.* **91**, 147902 (Oct 2003)
25. Wright, S., Nocedal, J.: Numerical optimization. *Springer Science* **35**(67-68), 7 (1999)