



Imperfect Distributed Quantum Phase Estimation

Niels M. P. Neumann^(✉), Roy van Houte, and Thomas Attema

The Netherlands Organisation for Applied Scientific Research,
Anna van Buerenplein 1, 2595 DA The Hague, The Netherlands
{niels.neumann,roy.vanhoute,thomas.attema}@tno.nl

Abstract. In the near-term, the number of qubits in quantum computers will be limited to a few hundreds. Therefore, problems are often too large and complex to be run on quantum devices. By distributing quantum algorithms over different devices, larger problem instances can be run. This distributing however, often requires operations between two qubits of different devices. Using shared entangled states and classical communication, these operations between different devices can still be performed. In the ideal case of perfect fidelity, distributed quantum computing is a solution to achieving scalable quantum computers with a larger number of qubits. In this work we consider the effects on the output fidelity of a quantum algorithm when using noisy shared entangled states. We consider the quantum phase estimation algorithm and present two distribution schemes for the algorithm. We give the resource requirements for both and show that using less noisy shared entangled states results in a higher overall fidelity.

Keywords: Noisy quantum computing · Phase estimation · Distributed quantum computing · Entanglement fidelity · Quantum Fourier transform

1 Introduction

The field of quantum computation already contains an extensive amount of theoretical knowledge and has found more applications in the last decades [10]. The combination of quantum computing and quantum networks opens a whole new world of information and communication technology as new applications emerge. Applications of quantum computing and quantum networks are being developed that are not feasible using classical computers and classical communication, such as applications for security [1, 13], telescopey [5] and clock-synchronization [8].

Current quantum computers are far from solving large practical problems and implementing such quantum computers still comes with many challenges [11]. One of the hurdles of such a universal quantum computer is the number of qubits. The required number of qubits depends on both the application and implementation of the corresponding algorithm. This means that a single quantum computer with only a few qubits will in general not be able to solve larger

problems. However, using a quantum network to link together multiple quantum computers, each with a handful of qubits, larger problem instances can be solved. This concept is called *distributed quantum computing* (DQC) [3, 14].

Using a network of computers to solve large problems is not unique to quantum computers and quantum algorithms. In general, distributed (quantum) computing combines different (quantum) computers, where each machine performs part of the computation. This gives either a speed-up (parallelism) or allows to solve large problem instances (larger computers). These (quantum) computers may be physically separated. In this work, we focus on quantum computers and we consider the effects on the output quality, when distributing a quantum algorithm over multiple devices. The total number of usable qubits depends on the sum of the qubits of each quantum computer separately. Additionally, each device has a communication qubit, used for the shared entanglement with other quantum computers.

Different quantum computers can be linked by shared entangled qubit pairs. This entanglement allows for physically separated machines. Operations involving qubits from a single quantum computer are called *local*, whereas operations using qubits from different devices are called *non-local*. Non-local operations require shared entangled qubit pairs and ideally these pairs are in one of the Bell states [15], often referred to as EPR-pairs. These are used, together with classical communication bits, to perform non-local operations.

Due to noise in the quantum gates and qubit decoherence, the output of a quantum algorithm might differ from the theoretical output. A measure on how well the output quantum state matches with the theoretically expected output, is the fidelity and is given in Eq. (3.2). The lower the fidelity, the more the two states differ. When distributing a quantum algorithm, imperfections in the shared EPR-pair form another potential source of uncertainty. These imperfections can for instance occur due to an imperfect generation process or imperfections in the used quantum channels.

Non-local gates were first described by Eisert et al. [3] in 2000. Later, Yimsiriwattana and Lomonaco showed a modular approach to distributed quantum computing and suggest the use of quantum teleportation to decrease the number of EPR-pairs required [15]. In 2004 they extended their approach to a distributed version of Shor's algorithm [14]. A distributed version of Grover's algorithm was presented by Exman and Levy in 2012 [4]. Only recently, in 2018, Moghadam et al. designed an algorithm to optimize the teleportation cost of distributed quantum circuits [16].

In each of these works, however, only the perfect setting is considered with no gate- and qubit errors. We relax this assumption by allowing imperfect EPR-pairs. Local quantum operations are still assumed to be noiseless and qubits are assumed to not decohere. Hence, errors can only be introduced by the imperfect shared entanglement used for non-local operations. We present two different distribution schemes, one standard implementation and one implementation where operations are combined, thereby requiring less imperfect EPR-pairs.

In Sect. 2, we will explain the quantum phase estimation algorithm (Sect. 2.1), non-local controlled operations (Sect. 2.2) and the distributed quantum Fourier transform (Sect. 2.3). Afterwards we introduce the setup of our simulations in Sect. 3.1 and we present the corresponding results in Sect. 3.2. Conclusions are given in Sect. 4.

2 Distributed Quantum Computing and Phase Estimation

We will first briefly explain the quantum phase estimation algorithm in Sect. 2.1. Then we explain how to perform non-local controlled U -gates in Sect. 2.2. We end this section by giving two implementation schemes for the distributed quantum Fourier transform in Sect. 2.3.

2.1 Quantum Phase Estimation

The phase estimation algorithm, first presented by Kitaev [9], returns an approximation of an eigenvalue of a given unitary U and a corresponding eigenvector. It has a wide range of applications, the most famous of which is Shor’s algorithm [12].

More formally, if U is a unitary operation on m qubits, and $|\psi\rangle$ is an eigenstate of U , then $U|\psi\rangle = \exp(2\pi i\varphi)|\psi\rangle$ for some phase $\varphi \in [0, 1)$. Let

$$\varphi = \sum_{i=1}^{\infty} \varphi_i 2^{-i} = 0.\varphi_1\varphi_2\dots \tag{2.1}$$

be the binary representation of φ . If we truncate the sum of Eq. (2.1) to n , we have an n -bit approximation of φ given by $0.\varphi_1\varphi_2\dots\varphi_n$. This n -bit approximation of φ is found using the quantum phase estimation algorithm.

A quantum circuit implementation of the quantum phase estimation is given in Fig. 1, with two registers of n and m qubits, respectively. If φ can be represented exactly in at most n -bits, this will be the output of the algorithm with

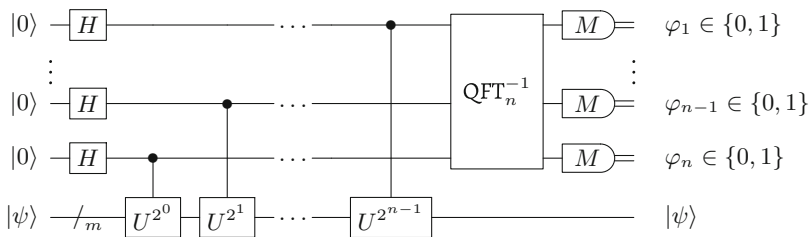


Fig. 1. The quantum phase estimation circuit for a unitary U acting on m qubits. The result is an n -bit approximation of the eigenvalue φ of eigenvector $|\psi\rangle$. The block QFT_n^{-1} is the inverse quantum Fourier transform on n qubits.

certainty. Otherwise, the approximation will round the phase and the correct result is given with probability at least $4/\pi^2$ [2].

First a Hadamard gate is applied on the first n qubits. Afterwards, controlled- $U^{2^{n-i}}$ gates are applied, with control qubit i in the first register and the qubits in the second register as target. Then an inverse quantum Fourier transform on the first register is applied and the qubits are measured. This gives the n -bit phase approximation of φ .

2.2 Distributed Controlled U -gate

A universal gate-set for local operations is given by a CNOT-gate and single qubit rotations [9]. A universal gate-set for non-local operations is thus obtained by a combination of the local universal gate set and non-local CNOT gates. By combining non-local CNOT-gates and local operations, arbitrary non-local operations are constructed. This is similar to how one would construct arbitrary local operations.

Suppose we want to apply a controlled U -gate between two qubits $|\psi\rangle$ and $|\phi\rangle$ on two different devices. Furthermore, let there be two extra qubits, one on each device, that share an entangled state in the Bell state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ and assume the two devices can communicate classically. The quantum circuit given in Fig. 2 performs this non-local controlled U -gate. Here, the two dotted boxes indicate the two quantum devices. The operation E_2 entangles $|0\rangle_1$ and $|0\rangle_2$ in a Bell-state and M indicates a measurement of the corresponding qubit. The double lines indicate classical control by the measured value. This quantum circuit is treated in more detail in [3, 15].

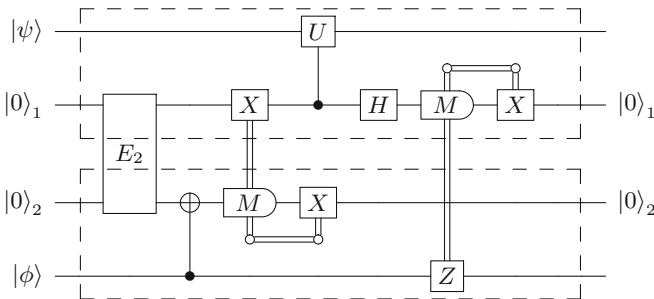


Fig. 2. A quantum circuit implementation of a non-local controlled U -gate between $|\phi\rangle$ and $|\psi\rangle$. The block E_2 creates an entangled qubit pair in state $(|00\rangle + |11\rangle)/\sqrt{2}$.

There are other ways of applying a non-local U operation. We can first teleport $|\psi\rangle$ to the other device, do all operations locally, and then teleport the resulting state back. This, however, requires one extra qubit per device, more operations and two shared EPR-pairs instead of one.

2.3 Distributed Quantum Fourier Transform

The quantum Fourier transform maps an n -qubit state $|k\rangle$ to $\sum_{j=0}^{2^n-1} e^{2\pi ijk/2^n} |j\rangle$, with i the complex unit. A recursive implementation of the quantum Fourier transform is given in Fig. 3. We see that the implementation can be decomposed in Hadamard gates and controlled R_k -gates. These rotation gates R_k are given by

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix}. \tag{2.2}$$

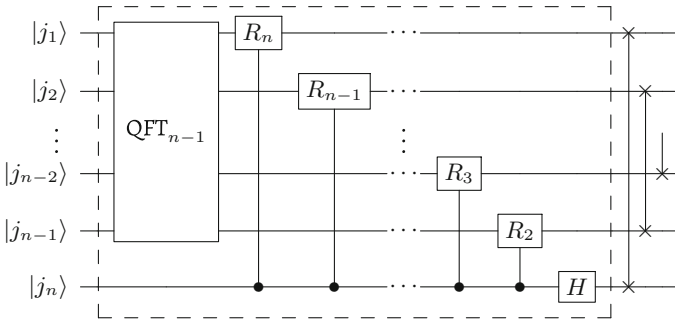


Fig. 3. A recursive quantum circuit for an n -qubit quantum Fourier transform. The dotted rectangle represents the QFT_n . By definition $\text{QFT}_1 = H$.

Table 1. The resources requirements for the local quantum Fourier transform on $2n$ qubits and the non-local quantum Fourier transform distributed over two devices of n qubits each.

Method	Number of qubits	Gates	Communication (EPR-pairs)	Communication (classical bits)
Local gates	$2n$	$2n^2 + n$	0	0
Standard implementation	$2n + 2$	$10n^2 + n$	n^2	$2n^2$
Combined implementation	$2n + 2$	$2n^2 + 9n$	n	$2n$

Note that the last operations in Fig. 3 can be omitted, as they only swap the order of the qubits. Performing non-local SWAP-gates gives a high computational overhead, whereas reversing the order of the measurement results is easily accounted for classically. Even if the output of the quantum Fourier transform is used as input for further computations, these operations can be accounted for without using SWAP gates.

A non-local implementation of the quantum Fourier transform is obtained by combining the quantum circuits shown in Fig. 2 and Fig. 3, with $U = R_k$. We refer to the approach of simply replacing each controlled gate by a non-local

one if necessary, as the *standard* approach. Instead of replacing each controlled operation by a non-local one, we can also use a single shared entangled state to perform multiple non-local gates, by grouping all operations on one computer that are controlled by a single qubit from another quantum computer. This quantum circuit is given in Fig. 4, where only a single qubit of the second device is shown. This *combined* approach uses less shared entangled states and has less communication overhead.

In Table 1 we show the resource requirements to run a quantum Fourier transform on $2n$ qubits for both the local implementation and the two presented non-local implementations. In the non-local situation, two additional qubits are used for the shared entanglement, as well as additional classical communication bits.

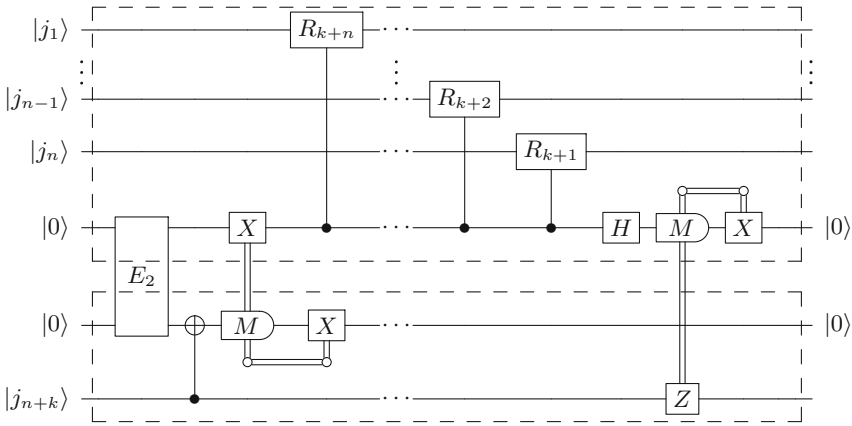


Fig. 4. Part of a distributed quantum Fourier transform, where n non-local operations, on n qubits are performed using a single shared entangled state. The control is $|j_{n+k}\rangle$, the targets are $|j_1\rangle, \dots, |j_n\rangle$. Only a single qubit of the second quantum computer is shown, others are omitted. The dashed boxes indicate the quantum computers and the double lines indicate classical communication.

3 Non-local Quantum Circuits with Imperfect Entanglement

In this section we will first explain the setup of our simulations (Sect. 3.1) and then present the results of these simulations (Sect. 3.2).

3.1 The Simulation Setup

We implemented the two distributed quantum circuits presented in the previous section, as well as an implementation of a local circuit. For these implementations we used Python 3.6 and the QuTiP Python package [6, 7]. Simulations are run in

the density state formalism. Depolarizing noise is applied to the shared entangled states using noise parameter α . The density representation of the noisy EPR-pair is given by

$$\eta(\alpha) = \frac{(1-\alpha)}{2}(|00\rangle + |11\rangle)(\langle 00| + \langle 11|) + \frac{\alpha}{4}I_2 \otimes I_2. \quad (3.1)$$

If $\alpha = 0$, the state corresponds to one of the Bell-states, whereas for increasing α , the state becomes more ideally mixed. In our simulations we consider k quantum computers, each with n_i qubits. Each device has one additional qubit used for the shared entanglement.

For different topologies we compare the output of the quantum circuit $\eta_{out}(\alpha)$ with the output in a noiseless situation $\eta_{out}(0)$. The quality is expressed in terms of the fidelity between a pair of density matrices ρ and σ and is given by

$$F(\rho, \sigma) = \left[\text{Tr} \sqrt{\rho^{1/2} \sigma \rho^{1/2}} \right]^2. \quad (3.2)$$

3.2 Results

We run our simulations for the unitary operation

$$R_\varphi = \begin{pmatrix} 1 & 0 \\ 0 & \exp(2\pi i\varphi) \end{pmatrix}$$

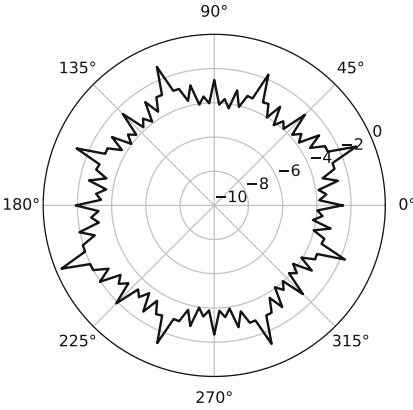
with eigenvector $|\psi\rangle = |1\rangle$. The corresponding eigenvalue is $\exp(2\pi i\varphi)$ which has phase φ . We consider different noise rates $\alpha \in [0, 1]$.

First, we consider a random fixed angle $\varphi = 72/128$, and two quantum computers of 4 qubits each, 7 qubits of which we use for the approximation. We run the simulations for different noise rates and the results for $\alpha = 0.1$ and 0.5 are shown in Fig. 5. Note that φ translates to a fraction of 2π and hence to an angle of $\varphi * 2\pi = 202.5^\circ$ in these plots. Results are presented using log-radar plots for both the standard and the combined implementation. The shown results are the log-values of the output probabilities. The results for $\alpha = 0$ and $\alpha = 1$ are not shown. For $\alpha = 0$, no errors occur and φ is retrieved with certainty. For $\alpha = 1$, the result is uniform for all states.

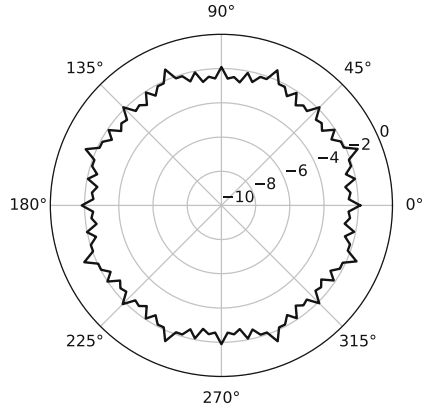
The results for both implementations are similar and show a repetitive pattern, with spikes every 45° . The largest spike is found at 202.5° , corresponding to the phase φ to be found. These effects were also found for quantum computers of different sizes and when distributing over more than two devices.

For the 7-bit approximation, 128 different measurement outcomes are possible. For both the standard and the combined implementation, we found that the results are independent from the initial angle up to rotations with steps of $1/128$. Therefore, the probability distribution of $\varphi = 72/128$, is the rotated probability distribution of $\varphi = 0$. More generally, we also found that the probability distributions for m -bit approximations are equivalent up to rotation for $m > n$. For example, the 9-bit phases $\varphi = 1/512$ and $\varphi = 3/512$, give the same probability distribution in a 7-bit approximation.

Standard phase estimation
 $\varphi = 72/128$ and $\alpha = 0.1$



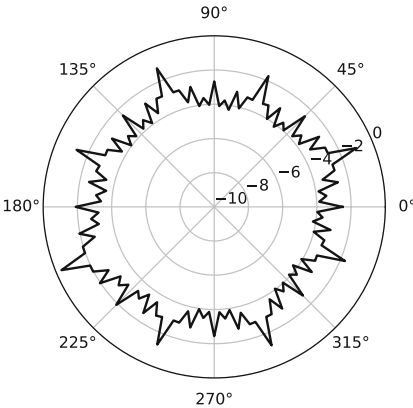
Standard phase estimation
 $\varphi = 72/128$ and $\alpha = 0.5$



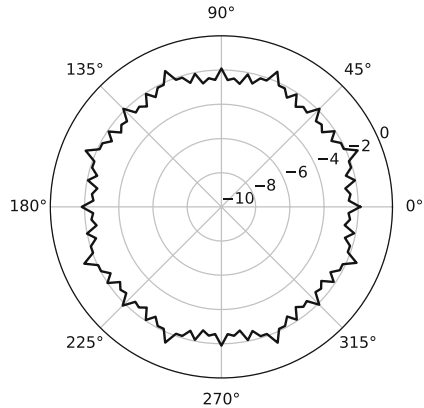
(a) Standard implementation with noise parameter $\alpha = 0.1$.

(b) Standard implementation with noise parameter $\alpha = 0.5$.

Combined phase estimation
 $\varphi = 72/128$ and $\alpha = 0.1$



Combined phase estimation
 $\varphi = 72/128$ and $\alpha = 0.5$



(c) Combined implementation with noise parameter $\alpha = 0.1$.

(d) Combined implementation with noise parameter $\alpha = 0.5$.

Fig. 5. The probability distributions for the standard and combined distributed phase estimation circuit for $\alpha = 0.1$ and $\alpha = 0.5$

Even though the probability distributions for the standard and combined approach seem similar, they are not the same. We found that the probability of correct retrieval of angle φ was highest for the combined implementation.

In Fig. 6 we show the fidelity for both implementations for varying α -values for the same network as before: two quantum computers with 4 qubits each. As expected, the results are the same for $\alpha = 0$. For $\alpha = 1$, the probability distributions are uniform and hence equal for both the standard and the combined

implementation. Note that with increasing noise rate, the fidelity drops off quickly. However, also note that the fidelity will not become zero, due to the uniform distribution obtained for $\alpha = 1$.

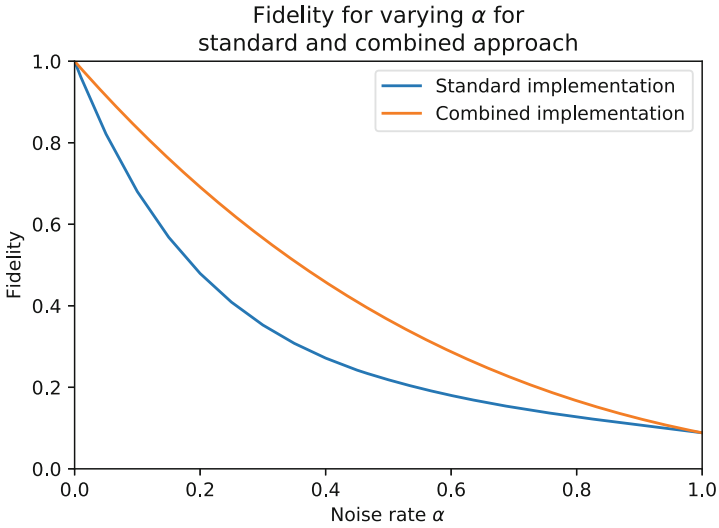


Fig. 6. Comparison between the fidelity for the standard and combined approach for varying noise rates α for two quantum computers of 4 qubits each.

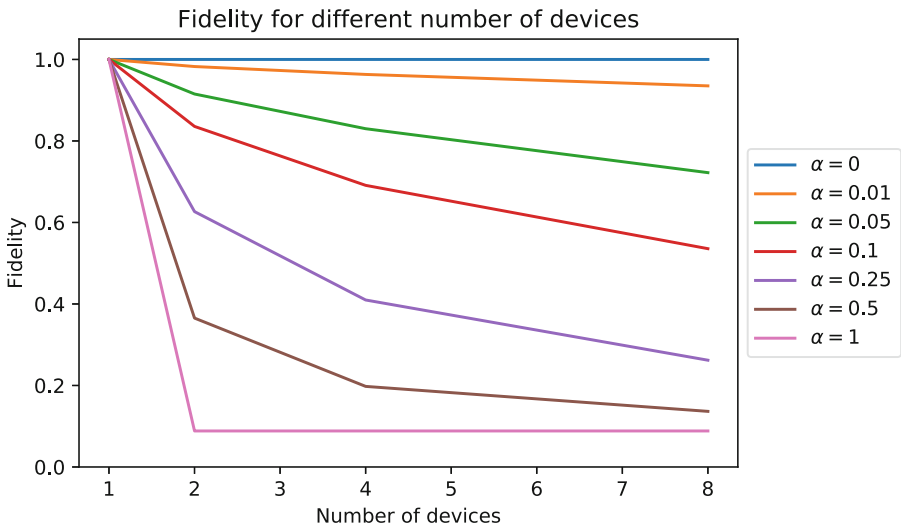


Fig. 7. The fidelity for a varying number of devices the phase estimation algorithm is distributed over. Different noise rates α are shown.

Finally, we consider the effects of distributing the algorithm over more devices. We consider 8 qubits in total and distributed the quantum phase estimation algorithm over k quantum computers for $k \in \{1, 2, 4, 8\}$. The results are shown in Fig. 7 for varying noise rates $\alpha \in [0, 1]$. Naturally, the fidelity is 1 when doing all computations locally ($k = 1$), independent of the noise rate α . When distributing the algorithm ($k > 1$), the fidelity becomes smaller quickly even for small error rates. For noise rates $\alpha = 1$, we see that distributing the algorithm over two or more devices, results in a uniform distribution.

4 Conclusions

In this work, we considered the effect of imperfect shared entanglement on the output fidelity of distributed quantum algorithms. We used the phase estimation algorithm and proposed two distribution schemes for the corresponding quantum Fourier transform. One is a standard approach, where every non-local operation uses a shared entangled pair; the other is a combined approach, where different non-local operations are grouped to use only a single shared entangled pair.

The output probability distributions for both schemes are very similar and independent of the input angle, up to rotations. However, the combined approach gives the highest probability of correct retrieval of the phase φ . Also in terms of fidelity, the differences are more prominent, especially for smaller α -values. Again the combined approach shows the highest fidelity. For high noise rates, the fidelity of both is very similar and near uniform.

We thus found that using less shared entangled states is beneficial for the output in terms of fidelity. Note however, that the results presented in this paper are based on simulations and hence a formal proof of the result is still needed. Furthermore, we assumed perfect local operations and not qubit decoherence. In practice, both will play a role.

The fidelity of the shared entangled pair is related to the noise rate α , with $\alpha = 0$ resulting in a fidelity of 1. As the fidelity of the output drops quickly with increased noise rates α , the fidelity of this shared entangled pair must be close to 1. Different techniques can be used to obtain a higher fidelity, such as entanglement purification. This allows for higher fidelity, but also introduces overhead. In our case of no gate errors and no qubit decoherence, this overhead will have no effect. In practical cases, we may however not neglect these two effects and there is a trade-off between the output fidelity of the algorithm and the fidelity of the shared entangled qubit pair.

Acknowledgments. We would like to thank Rob Knegjens for his helpful advice and insightful discussions. These helped improve the paper.

References

1. Bennett, C.H., Brassard, G.: Quantum cryptography: public key distribution and coin tossing. *Theor. Comput. Sci.* **560**, 7–11 (2014)
2. Cleve, R., Ekert, A., Macchiavello, C., Mosca, M.: Quantum algorithms revisited. *Proc. R. Soc. London. Ser. A Math. Phys. Eng. Sci.* **454**(1969), 339–354 (1998)
3. Eisert, J., Jacobs, K., Papadopoulos, P., Plenio, M.B.: Optimal local implementation of nonlocal quantum gates. *Phys. Rev. A* **62**, 052317 (2000)
4. Exman, I., Levy, E.: Quantum probes reduce measurements: application to distributed Grover algorithm (2012)
5. Gottesman, D., Jennewein, T., Croke, S.: Longer-baseline telescopes using quantum repeaters. *Phys. Rev. Lett.* **109**, 070503 (2012)
6. Johansson, J., Nation, P., Nori, F.: QuTiP: an open-source Python framework for the dynamics of open quantum systems. *Comput. Phys. Commun.* **183**(8), 1760–1772 (2012)
7. Johansson, J., Nation, P., Nori, F.: QuTiP 2: a Python framework for the dynamics of open quantum systems. *Comput. Phys. Commun.* **184**(4), 1234–1240 (2013)
8. Jozsa, R., Abrams, D.S., Dowling, J.P., Williams, C.P.: Quantum clock synchronization based on shared prior entanglement. *Phys. Rev. Lett.* **85**, 2010–2013 (2000)
9. Kitaev, A.Y.: Quantum computations: algorithms and error correction. *Russ. Math. Surv.* **52**(6), 1191–1249 (1997)
10. Nielsen, M.A., Chuang, I.L.: *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 10th edn. Cambridge University Press, New York (2010)
11. Preskill, J.: Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018)
12. Shor, P.W.: Polynomial time algorithms for discrete logarithms and factoring on a quantum computer. In: Adleman, L.M., Huang, M.-D. (eds.) ANTS 1994. LNCS, vol. 877, pp. 289–289. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-58691-1_68
13. Stucki, D., et al.: Long-term performance of the SwissQuantum quantum key distribution network in a field environment. *New J. Phys.* **13**(12), 123001 (2011)
14. Yimsiriwattana, A., Lomonaco Jr., S.J.: Distributed quantum computing: a distributed Shor algorithm. In: Donkor, E., Pirich, A.R., Brandt, H.E. (eds.) *Quantum Information and Computation II*, vol. 5436, pp. 360–372. International Society for Optics and Photonics, SPIE (2004)
15. Yimsiriwattana, A., Lomonaco, Jr., S.J.: Generalized GHZ states and distributed quantum computing (2004)
16. Zomorodi-Moghadam, M., Houshmand, M., Houshmand, M.: Optimizing teleportation cost in distributed quantum circuits. *Int. J. Theor. Phys.* **57**(3), 848–861 (2018)