

# Interference versus success probability in quantum algorithms with imperfections

Daniel Braun and Bertrand Georgeot

*Laboratoire de Physique Théorique, Université de Toulouse, UPS, CNRS, 31062 Toulouse, France*

(Received 9 November 2007; published 14 February 2008)

We study the importance of interference for the performance of Shor's factoring algorithm and Grover's search algorithm using a recently proposed interference measure. To this aim we introduce systematic unitary errors, random unitary errors, and decoherence processes in these algorithms. We show that unitary errors which destroy the interference destroy the efficiency of the algorithm, too. However, unitary errors may also create useless additional interference. In such a case the total amount of interference can increase, while the efficiency of the quantum computation decreases. For decoherence due to phase flip errors, interference is destroyed for small error probabilities, and converted into destructive interference for error probabilities approaching 1, leading to success probabilities which can even drop below the classical value. Our results show that in general, interference is necessary in order for a quantum algorithm to outperform classical computation, but large amounts of interference are not sufficient and can even lead to destructive interference with worse than classical success rates.

DOI: [10.1103/PhysRevA.77.022318](https://doi.org/10.1103/PhysRevA.77.022318)

PACS number(s): 03.67.Lx, 03.65.Yz

## I. INTRODUCTION

Quantum algorithms differ from classical stochastic algorithms by the fact that they have access to entangled quantum states and that they can make use of interference effects between different computational paths [1]. These effects can be exploited for spectacular results. Shor's algorithm factors large integers in a time which is polynomial in the number of digits [2], and Grover's search algorithm finds an item in an unstructured database of size  $N$  with only  $\sim\sqrt{N}$  queries [3]. Many other quantum algorithms have building blocks similar to those developed in those two seminal papers (e.g., [4–6]). But more than twenty years after the discovery of the first quantum algorithm [7] it is still not clear what exactly is at the origin of the speedup of quantum algorithms compared to their classical counterparts. Large amounts of entanglement must necessarily be generated in a quantum algorithm that offers an exponential speedup over classical computation [8], and tremendous effort has been spent to develop methods to detect and quantify entanglement in a given quantum state (see [9,10] for recent reviews). However, the creation of large amounts of entanglement does certainly not suffice for getting an efficient quantum algorithm, and it remains to be elucidated what are both necessary and sufficient requirements.

While there seems to be general agreement that interference plays an important role in quantum algorithms [11–13], surprisingly, it has remained almost unexplored in computational complexity theory. Recently we introduced a measure of interference in order to quantify the amount of interference present in a given quantum algorithm (or, more generally, in any quantum mechanical process in a finite dimensional Hilbert space) [14]. It turned out that both Grover's and Shor's algorithms use an exponential amount of interference when the entire algorithm is considered. Indeed, many useful quantum algorithms start off with superposing coherently all computational basis states at least in one register, which is a process that makes use of massive interference (the number of  $i$ -bits, a logarithmic unit of interference,

equals the number of qubits of the register [14]). Both algorithms differ substantially, however, in their exploitation of interference in the subsequent nongeneric part: Shor's algorithm uses exponentially large interference also in the remaining part of the algorithm due to a quantum Fourier transform (QFT), whereas the remainder of Grover's algorithm succeeds with the surprisingly small amount of roughly three  $i$ -bits, asymptotically independent of the number of qubits.

Recently it was shown that the QFT itself on a wide variety of input states (with efficient classical description) can be efficiently simulated on a classical computer, as the amount of entanglement remains logarithmically bounded [15–17]. As the QFT taken by itself creates exponential interference, it follows that an exponential amount of interference alone does not prevent an efficient classical simulation. This is in fact obvious already from the simple (if practically useless) quantum algorithm which consists of applying a Hadamard gate once on each qubit and then measuring all qubits. By definition, this algorithm uses exponential interference ( $\mathcal{I}=2^n-1$  for  $n$  qubits). When applied to an arbitrary computational basis state, one gets any output between 0 and  $2^n-1$  with equal probability,  $p=1/2^n$ . According to Jozsa and Linden's result [8] this algorithm cannot provide any speedup over its classical counterpart (as it creates zero entanglement), and indeed, it can evidently be efficiently simulated with a simple stochastic algorithm that spits out a random number between 0 and  $2^n-1$  with equal probability, which can be done by choosing each bit randomly and independently equal to 0 or 1 with probability 1/2. We note that the same phenomenon exists also for entanglement: the state  $(|000\cdots000\rangle+|111\cdots111\rangle)/\sqrt{2}$  has a lot of entanglement, but the corresponding probability distribution can be efficiently simulated classically with two registers.

Thus the precise nature of the relationship between interference and the power of quantum computation is not yet fully understood. Surprisingly, there are tasks in quantum information processing which do not require interference in order to give better than classical performance, as was shown

in [18] for quantum state transfer through spin chains. In order to shed light on this question, in this paper we study both Grover's and Shor's algorithms in the presence of various errors. As many other quantum algorithms are built on these two algorithms, our results may apply to a larger class of quantum algorithms. We analyze to what extent the interference in a quantum algorithm changes when the algorithm is subjected to errors, and to what extent these changes reflect a degradation of the performance of the algorithm. We will investigate this question for systematic and random, unitary or nonunitary errors, where the latter include uncorrelated bit flip and phase flip errors. These errors can be taken as representative of, respectively, correlated errors, random unitary noise due to fluctuations of gate parameters, and genuine decoherence, and therefore exhaust the main type of errors which can occur in an experimentally realizable quantum computer. We look at the "potentially available interference" as well as the "actually used interference," where the former means the interference in the entire algorithm, the latter the interference in the part of the algorithm after the application of the initial Hadamard gates [14].

## II. GROVER'S AND SHOR'S ALGORITHMS AND THE INTERFERENCE MEASURE

As we will use Grover's and Shor's algorithms throughout the paper we first review shortly their main components. Grover's algorithm  $U_G$  [3] finds a single marked item  $a$  in an unstructured database of  $N$  items in  $O(\sqrt{N})$  quantum operations, to be compared with  $O(N)$  operations for the classical algorithm. The algorithm starts on a system of  $n$  qubits (Hilbert space of dimension  $N=2^n$ ) with the Walsh-Hadamard transform  $W$ , which transforms the computational basis state  $|0\cdots 0\rangle$  into a uniform superposition of the basis states  $N^{-1/2}\sum_{x=0}^{N-1}|x\rangle$ . Then the algorithm iterates  $k$  times the same operator  $U=WR_2WR_1$ , with an optimal value  $k=\lceil\pi/(4\theta)\rceil$  (where  $\lceil\cdot\rceil$  means the integer part) and  $\sin^2\theta=1/N$ , i.e.,  $U_G=(WR_2WR_1)^k W$ . The oracle  $R_1$  multiplies the amplitude of the marked item  $a$  with a factor  $(-1)$ , and keeps the other amplitudes unchanged. The operator  $R_2$  multiplies the amplitude of the state  $|0\cdots 0\rangle$  with a factor  $(-1)$ , keeping the others unchanged.

Shor's algorithm [2] allows the explicit decomposition of a large integer number  $R$  into prime factors in a polynomial number of operations. The algorithm starts by applying  $2L$  Hadamard gates to a register of size  $2L$  where  $L=\lceil\log_2 R\rceil+1$ , in order to create an equal superposition of all computational basis states  $N^{-1/2}\sum_{x=0}^{N-1}|x\rangle$  where  $N=2^{2L}$ . Then the values of the function  $f(x)=a^x(\bmod R)$ , where  $a$  is a randomly chosen integer with  $0<a<R$ , are built on a second register of size  $L$  to yield the state  $N^{-1/2}\sum_{x=0}^{N-1}|x\rangle|f(x)\rangle$ . The last quantum operation consists in a quantum Fourier transform (QFT) on the first register only which allows one to find the period of the function  $f$ , from which a factor of  $R$  can be found with sufficiently high probability. In the numerical simulations, we did not take into account the workspace qubits which are necessary to perform the modular exponentiation, but are not used elsewhere. It is a reasonable simplifi-

cation in our case, since during this phase of modular exponentiation interference is not modified, as the whole process is effectively a permutation of states in Hilbert space [14]. In order to study numerically the effect of errors for different system sizes, we performed simulations for  $n=12$  qubits, which corresponds, respectively, to factorization of  $R=15$  (with  $a=7$ ), and also for  $n=9$  and  $n=6$ . The cases  $n=9$  and  $n=6$  correspond to order-finding for  $R=7$  (with  $a=3$ ) and  $R=3$  (with  $a=2$ ) and do not exactly correspond to an actual factorization, although the algorithmic operations are the same, and a period is found at the end. In the case  $n=9$  the period found does not divide the dimension of the Hilbert space, so the final wave function is not any more a superposition of equally spaced  $\delta$  peaks, but is composed of broader peaks. This enables reaching the more usual regime of the Shor algorithm, where in general the period is not a power of 2 (although it does not happen for  $R=15$ ). In one case the result was different enough to warrant the display of the corresponding curve for a different value of  $a$  ( $a=6$ ) for which the period is a power of 2 (see the end of Sec. III C).

The interference measure for a propagator  $P$  of a density matrix  $\rho$  ( $\rho'_{ij}=\sum_{k,l}P_{ij,kl}\rho_{kl}$ ) derived in [14] is defined as

$$\mathcal{I}(P)=\sum_{i,k,l}|P_{ii,kl}|^2-\sum_{i,k}|P_{ii,kk}|^2, \quad (1)$$

where  $P_{ij,kl}$  are the matrix elements of the propagator in the computational basis  $\{|k\rangle\}$  ( $k=0,\dots,2^n-1$ ), and  $\rho_{kl}=\langle k|\rho|l\rangle$ . While the interference measure is certainly not unique, it quantifies the two basic properties of interference: the coherence of the propagation, and the "equipartition" of the output states, i.e., to what extent the computational basis states are fanned out during propagation. Indeed, the second term in Eq. (1) can be understood as a sum over matrix elements of a classical stochastic map (the map which propagates the diagonal matrix elements of the density matrix, thus the probabilities in the computational basis). This term is subtracted from the more general first term, where the elements  $P_{ii,kl}$  of the propagator are responsible for the propagation of the coherences in the density matrix and their contribution to the final probabilities. Therefore if all coherences get destroyed during propagation (i.e., the map is purely classical), interference vanishes. The squares in Eq. (1) are important, as they allow us to measure the equipartition. The number of  $i$ -bits is defined as  $n_i=\log_2[\mathcal{I}(P)]$ . One Hadamard gate provides one  $i$ -bit of interference [14].

In quantum information theory the propagation of mixed states is generally formulated within the operator sum formalism [1]: A set of operators  $\{E_l\}$  acts on  $\rho$  according to  $\rho'=\sum_l E_l \rho E_l^\dagger=P\rho$ , where the Kraus operators  $E_l$  obey  $\sum_k E_k^\dagger E_k=\mathbf{1}$  for trace-preserving operations. The interference measure then becomes [Eq. (1)]

$$\mathcal{I}=\sum_{i,k,m}\left|\sum_l(E_l)_{ik}(E_l^*)_{im}\right|^2-\sum_{i,k}\left(\sum_l|(E_l)_{ik}|^2\right)^2. \quad (2)$$

In the case of unitary propagation presented by a matrix  $U$ , the interference measure reduces to

$$\mathcal{I}(P(U)) = N - \sum_{i,k} |U_{ik}|^4. \quad (3)$$

This form makes it obvious that the interference is bounded by  $0 \leq \mathcal{I}(P(U)) \leq N-1$ . The interference measure is invariant under permutations of the computational basis states.

### III. PERTURBED QUANTUM ALGORITHMS

In the following we examine the amount of interference in perturbed versions of Grover's algorithm and Shor's algorithm. We will distinguish between "potentially available" and "actually used" interference [14]. These names are motivated by the fact that both algorithms start from the single computational basis state  $|0\rangle$ , such that only the first column of the unitary matrix  $U$ , which represents the algorithms in the computational basis, determines the outcomes and success probabilities. The interference measure, however, counts the interference for all possible input states, i.e., the contributions from all columns in  $U$ —thus the "potentially available" interference is in general much larger than what is needed in the algorithms. The actually used interference on the other hand is the interference in the remainder of an algorithm after all the initial Hadamard gates have been applied. At that point a coherent superposition of all computational basis states has been built up (in the case of Shor's algorithm: a coherent superposition of all computational basis states of the first register), and therefore all the interference measured by  $\mathcal{I}(P(U))$  is actually used. Another motivation to look at these two different measures is the fact that the latter focuses on the "nongeneric" part of the algorithm.

#### A. Systematic unitary errors

We start by replacing each Hadamard gate with a perturbed gate, parametrized with an angle  $\theta$ , as

$$H(\theta) = \begin{pmatrix} \cos \theta & \sin \theta \\ \sin \theta & -\cos \theta \end{pmatrix}. \quad (4)$$

The unperturbed Hadamard gate corresponds to  $\theta = \pi/4$ , while the cases  $\theta = 0$  and  $\theta = \pi/2$  replace the Hadamard gates by the Pauli matrices  $\sigma_z$  and  $\sigma_x$ , respectively, which create no interference. Thus the replacement of Hadamard gates by Eq. (4) amounts to destroying the interference produced in the course of the algorithm in a controllable fashion. This allows us to compare the loss of interference with the efficiency of the algorithm in a systematic way. We measure this efficiency through the success probability  $S$ . For Grover's algorithm the natural definition of  $S$  is the probability to find the searched state  $|\alpha\rangle$ ,  $S = \text{tr}(\rho_f |\alpha\rangle\langle\alpha|)$ , where the density matrix  $\rho_f$  describes the final state at the end of the computation, which may be a mixed state if decoherence strikes during the calculation (see Sec. III C). For Shor's algorithm there are in general many "good" final states which allow us to compute the period of the function  $f$ , and it is therefore more appropriate to define  $S$  through the loss of probability on these "good" final states compared to the unperturbed algorithm. Thus if  $\sum_i \psi_i |i\rangle$  is the final state of the unperturbed algorithm and  $\sum_i \psi_i^{err} |i\rangle$ , we define  $S$  for Shor's algorithm as

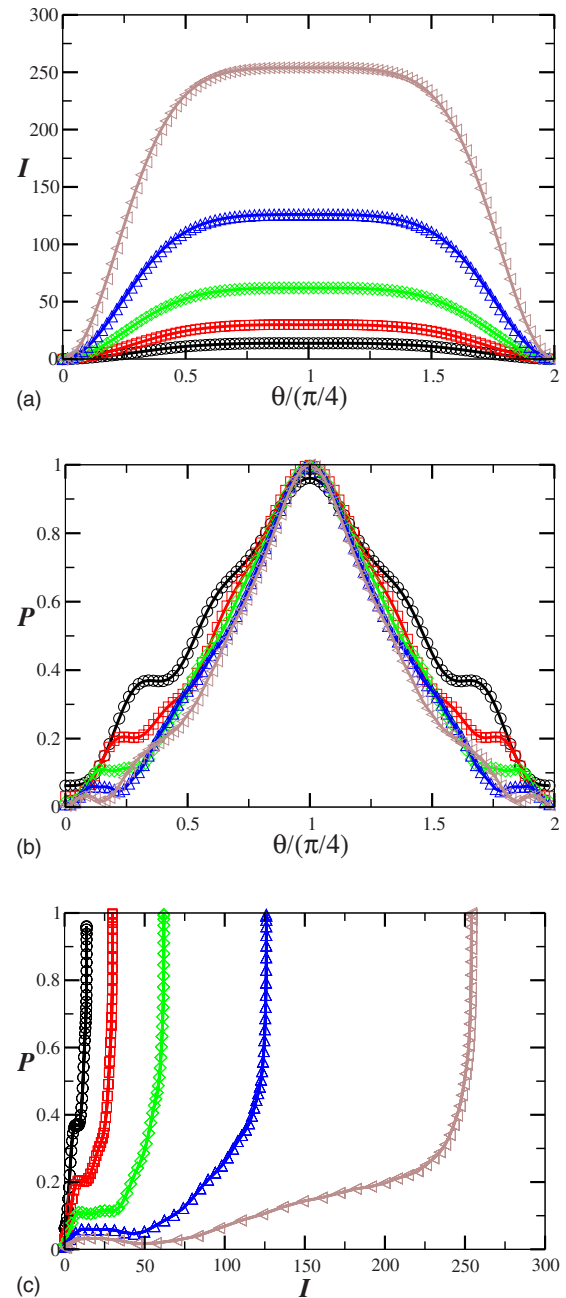


FIG. 1. (Color online) Potentially available interference in the Grover algorithm with systematic unitary errors in the Hadamard gates, parametrized by the angle  $\theta$ , Eq. (4) (a); success probability  $S$  of the algorithm (b); and success probability as function of interference (c). Black circles mean  $n=4$ , red squares  $n=5$ , green diamonds  $n=6$ , blue triangles up  $n=7$ , gray triangles left  $n=8$ . All curves are averaged over all values of the searched item  $\alpha$ .

$$S = 1 - \sum_i \left( |\psi_i|^2 - |\psi_i^{err}|^2 \right) / 2. \quad (5)$$

Figure 1 shows the dependence of the potentially available interference and of the success probability  $S$  of Grover's algorithm on  $\theta$ , as well as the success probability as function of the interference. All curves are averages over all values of  $\alpha$ ,  $\alpha = 0, \dots, 2^n - 1$ . Both interference and success probability

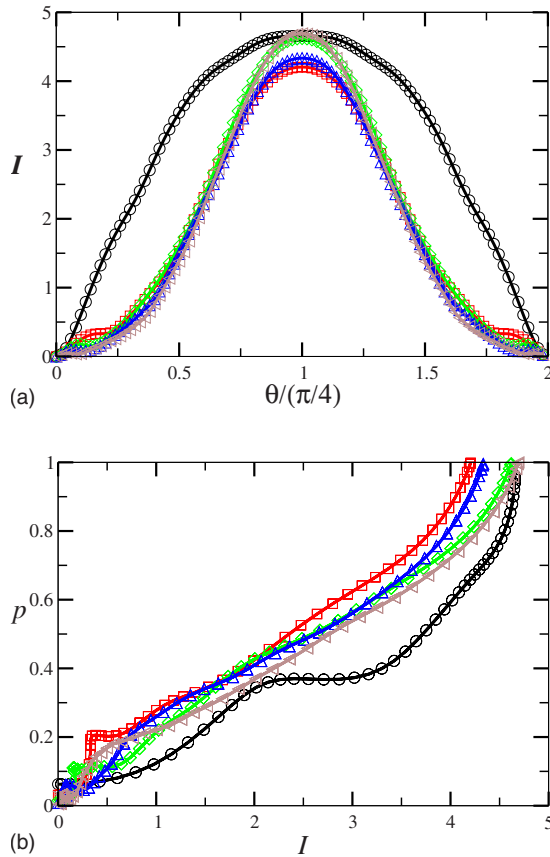


FIG. 2. (Color online) Same as Fig. 1, but for actually used interference.

peak at  $\theta = \pi/4$ . For a small number of qubits,  $S(\theta)$  shows some additional modulation in the wings of the  $n$  curve, which are pushed further and further out for increasing  $n$ . The broad maxima of  $\mathcal{I}(\theta)$  lead to steep increases of  $S(\mathcal{I})$  close to the maximum possible value for the interference  $\mathcal{I} = 2^n - 1$ . At  $\theta = 0$  or  $\theta = \pi/2$ , the interference vanishes, as in that case the algorithm degenerates to a combination of permutations and phase shifts, so that no two computational states get superposed. Figure 1 shows that for this example an exponential amount of interference is necessary even in order to obtain a success probability of the order 1/2, and by squeezing out a small additional amount of interference,  $S$  can be boosted to its optimal value.

The actually used interference gives a similar picture: For  $\theta = 0$  or  $\theta = \pi/2$  the interference vanishes, and interference reaches its maximum value  $\mathcal{I} \approx 4$  for  $\theta = \pi/4$ . As the success probability remains unchanged whether we calculate the interference for the entire algorithm or only after the initial Hadamard gates, we find again that the success probability increases with increasing interference (see Fig. 2).

Figure 3 shows the potentially available interference and the success probability for Shor’s algorithm. Again, both interference and success probability peak at  $\theta = \pi/4$ . The additional modulation in the wings of the curve for  $S(\theta)$  for a small number of qubits is much less pronounced than for Grover’s algorithm, but the main fact remains that the broad maximum of  $\mathcal{I}(\theta)$  corresponds to a sharp peak for  $S(\theta)$ , lead-

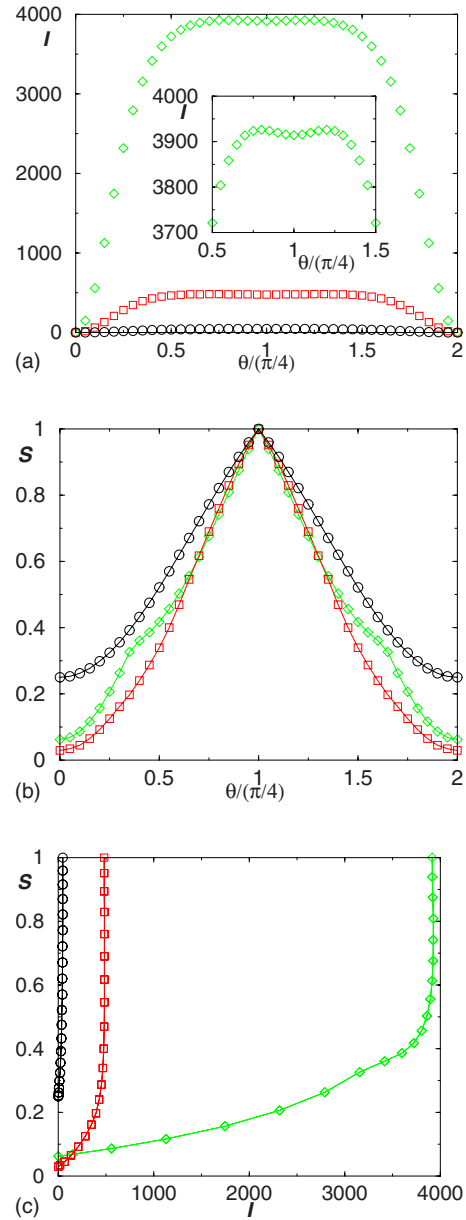


FIG. 3. (Color online) Potentially available interference in the Shor algorithm with systematic unitary errors in the Hadamard gates, parametrized by the angle  $\theta$ , Eq. (4) (a); success probability  $S$  of the algorithm (b); and success probability as a function of interference (c). Black circles mean  $n=6$  [ $f(x)=2^x(\text{mod } 3)$ ], red squares  $n=9$  [ $f(x)=3^x(\text{mod } 7)$ ], green diamonds  $n=12$  [ $f(x)=7^x(\text{mod } 15)$ ]. The inset is a closeup of the case  $n=12$  close to the maximum.

ing to the same steep increase of  $S(\mathcal{I})$  close to the maximum possible value of the interference. However, the exact algorithm does not lead to the maximum possible amount of interference. Close to  $\theta = \pi/4$ , the interference slightly increases while the success probability goes down, indicating that some interference which is “useless” in terms of the algorithm efficiency is generated. When  $\theta$  increases and the interference is reduced by a large amount, the algorithm has a low success probability. Even though the success probability globally goes down with an increasing number of qubits, it is not the case for all  $\theta$  values. This can be attributed to the



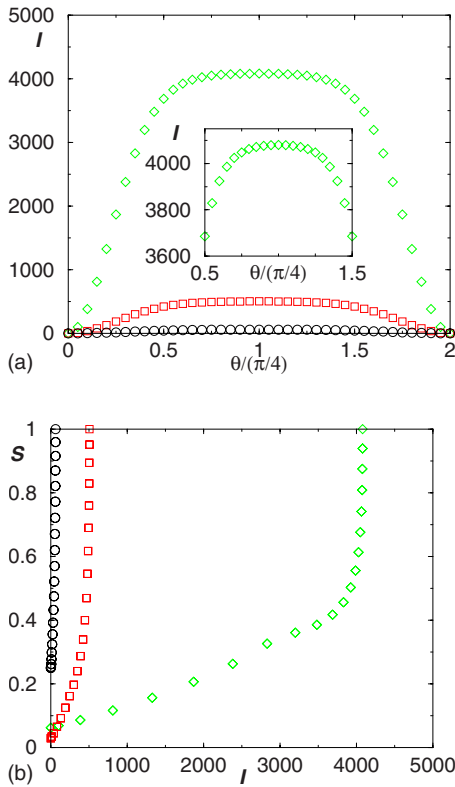


FIG. 4. (Color online) Same as Fig. 3, but for actually used interference.

fact that the three curves on the figure do not exactly describe the same problem for different numbers of qubits, but are instances of order-finding for different values of the number  $R$ . Thus when different values of the number of qubits are used, the precise problem investigated depends on the number theoretical properties of the integers chosen, which can be different. On the contrary, Grover's algorithm run on different numbers of qubits is essentially the same problem run on a computer of different size.

In [14], it was pointed out that Grover's and Shor's algorithms use a very different amount of actually used interference. Indeed, for Grover's algorithm it remains bounded for all values of the number of qubits  $n$ , while for Shor's algorithm it grows exponentially with  $n$ . This may be related to the fact that Shor's algorithm is exponentially faster than all known classical algorithms, while Grover's is only quadratically faster. In Fig. 4 the actually used interference is plotted for Shor's algorithm, showing that for  $\theta = \pi/4$  it reaches its maximal value which grows exponentially with the number of qubits. In this case, any decrease in the interference corresponds to a decrease of the success probability.

**B. Random unitary errors**

Let us now consider what happens if we replace each Hadamard gate with a gate given by Eq. (4), where each  $\theta$  is chosen randomly, uniformly, and independently from all other gates in an interval  $\pi/4 - \epsilon/2, \pi/4 + \epsilon/2$ .

Figure 5 shows the interference and success probability of Grover's algorithm as a function of  $\epsilon$ . All curves are aver-

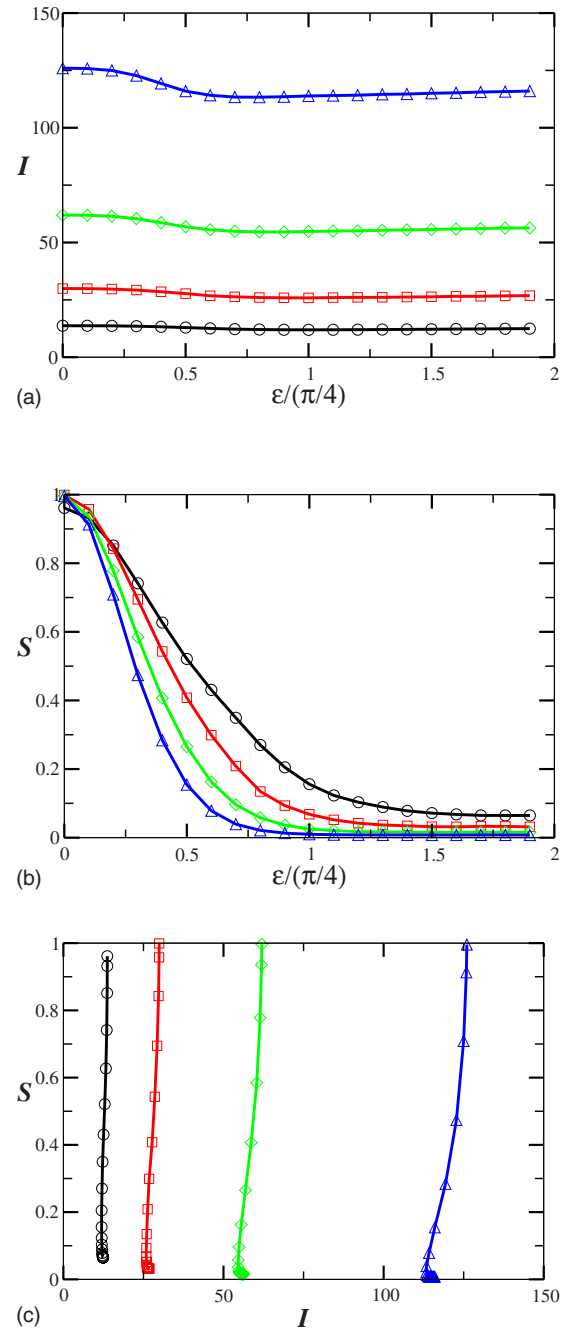


FIG. 5. (Color online) Potentially available interference in the Grover algorithm with random unitary errors in the Hadamard gates, parametrized by the interval  $\epsilon$ , Eq. (4) (a); success probability  $S$  of the algorithm (b); and success probability as function of interference (c) for  $n=4$  to  $n=7$ . Same symbols as in Fig. 1.

aged over all possible values of  $\alpha$  as well as over  $n_r$  random realizations of the algorithm ( $n_r=1000$  for  $n=4$ ,  $n_r=100$  for  $n=5, \dots, 7$ ). Again, the maximum amount of interference is obtained for the unperturbed algorithm,  $\epsilon=0$ , but the maximum is not very prominent. It is followed by a shallow minimum close to  $\epsilon = \pi/4$ , which gets shifted to smaller values for increasing  $n$ . Altogether, the interference is little affected by the random unitary errors. This can be understood from the fact that the unitary matrix  $U$  representing the algorithm

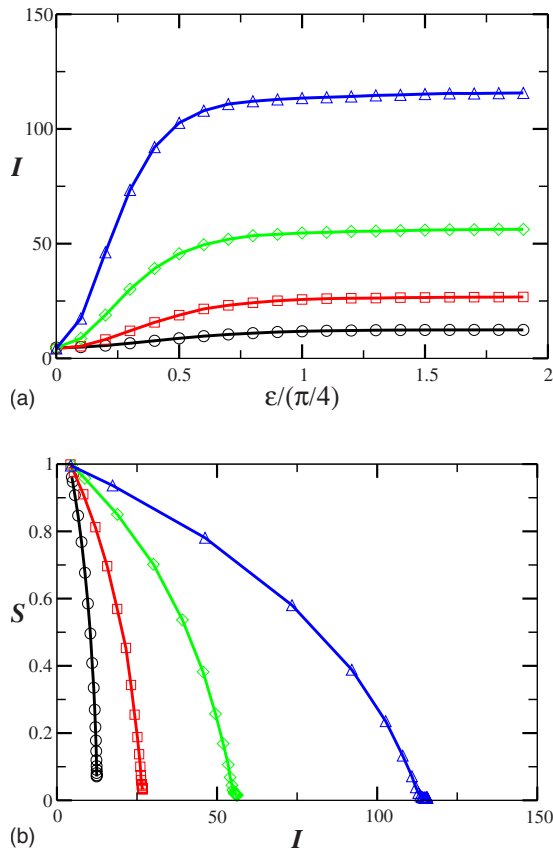


FIG. 6. (Color online) Actually used interference in the Grover algorithm with random unitary errors in the Hadamard gates, parametrized by the interval  $\epsilon$ , Eq. (4) (a), and success probability  $S$  as a function of interference (b). Same symbols as in Fig. 1. The success probability as a function of  $\epsilon$  is the same as in Fig. 5.

is already almost full in the unperturbed algorithm [14], with the exception of the first column, which propagates the initial state  $|0\rangle$ , and presents a strong peak on the searched item. Randomly replacing the Hadamard gates by  $H(\theta)$  increases the equipartition in the first column, as is witnessed by the decay of  $S$ , but reduces on average the equipartition in the other columns, leading to a slight overall decrease of interference close to  $\epsilon=0$ . This means again that a very large amount of interference is necessary in order to get even a modest performance of the algorithm, and a very steep increase of the success probability occurs when interference is boosted to its maximum value.

The situation for the actually used interference is quite different, as shown in Fig. 6. For  $\epsilon=0$ , we have an interference  $I \sim 4$  at the end of the algorithm (after the first diffusion gate it reaches its maximum value of  $I \sim 8-24/N$  and then oscillates and decays with each subsequent diffusion gate to the final value  $I \sim 4$  [14]). Thus the unperturbed algorithm leads to remarkably low equipartition in the entire matrix  $U$ , a highly unlikely situation for any random matrix. Indeed it was shown in [19] that a random unitary  $N \times N$  matrix drawn from the circular unitary ensemble (CUE) gives, with almost certainty, an interference  $I \sim N-2$ . Thus it is not surprising that with growing  $\epsilon$ ,  $I$  rapidly increases to a value  $I \sim N$ . As the success probability decreases with  $\epsilon$ , this leads to the

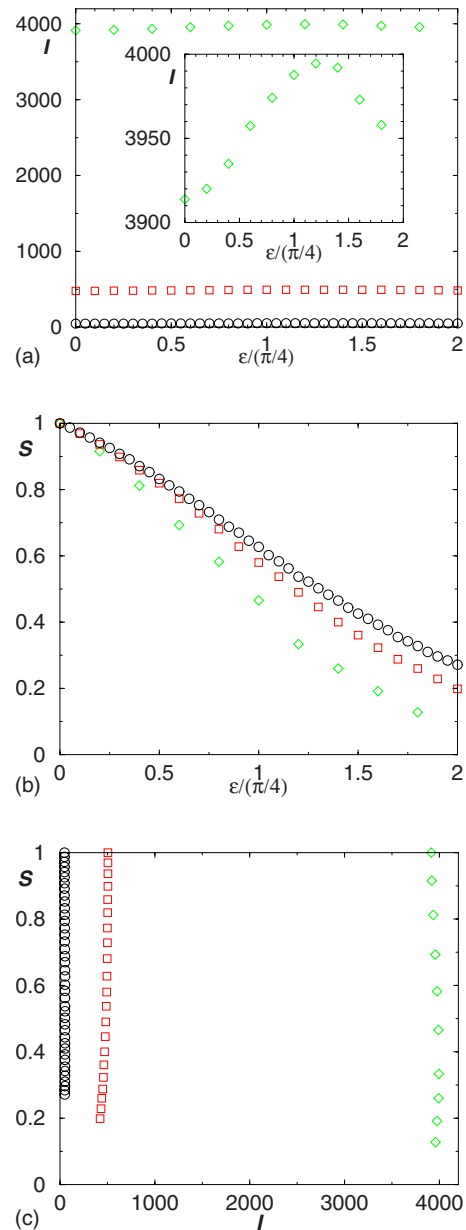


FIG. 7. (Color online) Potentially available interference in Shor's algorithm with random unitary errors in the Hadamard gates, parametrized by the interval  $\epsilon$ , Eq. (4) (a); success probability of the algorithm (b); and success probability as a function of interference (c). Symbols as in Fig. 3. The inset shows a closeup of the curve for  $n=12$ .

counterintuitive situation that the success probability decays with increasing actually used interference.

Figures 7 and 8 display the effect of random unitary errors on Shor's algorithm with the number of random realizations  $n_r=5000$  ( $n=6$ ),  $n_r=1000$  ( $n=9$ ), and  $n_r=100$  ( $n=12$ ). Besides changing the Hadamard gates, random phases with the same distribution were added to the two-qubit gates in the quantum Fourier transform. In this way in both algorithms, all Fourier transforms and Walsh-Hadamard transforms are randomized in a comparable way. Figure 7 shows that the potentially available interference oscillates slowly as a func-

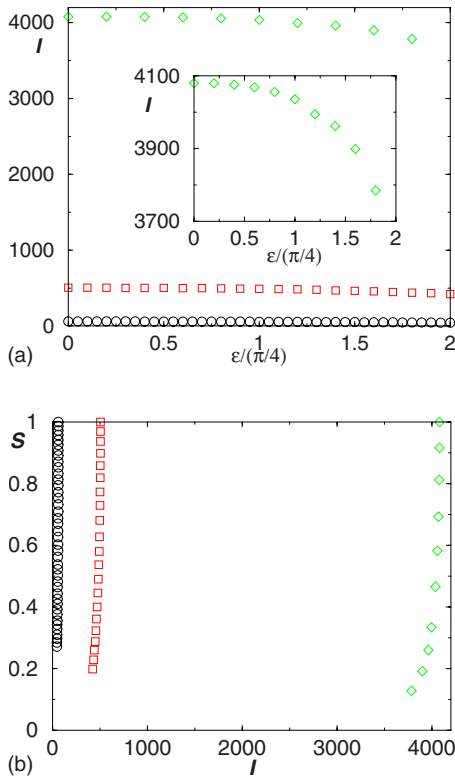


FIG. 8. (Color online) Actually used interference in Shor’s algorithm with random unitary errors in the Hadamard gates, parametrized by the interval  $\epsilon$ , Eq. (4) (a), and success probability as a function of interference (b). Same symbols as in Fig. 3. The inset shows a closeup of the curve for  $n=12$ . The success probability as a function of  $\epsilon$  is the same as in Fig. 7.

tion of  $\epsilon$ , on a scale which seems independent of the number of qubits and also larger than for Grover’s algorithm. The situation is similar to the one in Fig. 6 (although Fig. 6 deals with actually used interference), since interference increases for small values of  $\epsilon$  and reaches a maximal value around  $\epsilon=0.6-0.7$ , while the success probability decreases.

The situation is different in the case of actually used interference, shown in Fig. 8. Indeed, interference starts from its maximum possible value and decreases with increasing  $\epsilon$  at the same time as the success probability decreases. In the same way as for potentially available interference, the variation of interference is relatively small compared to the case of systematic errors, which were explicitly designed to destroy interference. Nevertheless, Fig. 8 shows that contrary to the case of Grover’s algorithm, interference and success probability decrease in a correlated way.

### C. Decoherence

We finally consider a class of errors which create true decoherence. We distinguish between phase flips and bit flips, and consider a (somewhat artificial) situation, where the errors occur only during the first Walsh-Hadamard transformation, i.e., the sequence of Hadamard gates on all qubits at the beginning of the algorithm in the case of Grover’s algorithm, and on all qubits of the first register of length  $2L$

in the case of Shor’s algorithm. We will assume that  $n_f$  out of  $n$  qubits are affected by errors, and study interference and success probability as a function of  $n_f$ ,  $n_f=1, \dots, n$ . Note that if all Hadamard gates in the entire algorithm were prone to error, one would need to calculate  $2^{(2k+1)n}$  Kraus operators for Grover’s algorithm (see Sec. II), each of which is a  $2^n \times 2^n$  matrix, which makes the numerical calculation rapidly too costly. The former number is reduced to a more bearable  $2^{n_f}$  in our case. For Shor’s algorithm, the number of Hadamard gates depends on the implementation of the modular exponentiation and the calculation of the function  $f$ , but grows exponentially with the number of qubits as well, if all qubits can be affected by the decoherence process. Contrary to the calculations for unitary errors, in the simulation of Grover’s algorithm we restricted ourselves to a fixed value of the searched item  $\alpha$ , but checked for a few different values of  $\alpha$  that the results are insensitive to the value of  $\alpha$ .

A Hadamard gate prone to errors is followed with probability  $p$  by a bit flip (or by a phase flip—we consider only one type of error at a time), and we calculate again both the potentially available and actually used interference. In the latter case, only the Pauli operators  $\sqrt{p}\sigma_z$  and  $\sqrt{p}\sigma_x$  which represent the phase flip and bit flip errors, respectively, with probability  $p$  on a given qubit are included in the Kraus operators, but not the initial Hadamard gates themselves.

For Grover’s algorithm, Fig. 9 shows the result in the case of bit flip errors, and Fig. 10 for phase flip errors. For both types of error, the interference has maximal value for  $p=0$  or  $p=1$ , which corresponds to completely coherent propagation, and minimal value for  $p=0.5$ . The minimal value decreases rapidly with the number of qubits prone to error. The potentially available interference reaches zero in the case of bit flip errors on all  $n$  qubits, whereas for phase flip errors a finite value remains. The actually used interference shows the opposite behavior. It has zero minimal value for phase flip errors on all  $n$  qubits, whereas it remains finite for bit flip errors even with probability 0.5. Phase flip errors rapidly destroy the operability of the algorithm. The success probability decreases linearly with  $p$  for  $n_f=1$  to reach a value close to zero for  $p=1$ , and more and more rapidly for increasing  $n_f$ . In fact, for  $n=4$ ,  $S(p=1) \approx 0.0025$ , independent of  $n_f$ , which is even smaller than the classical value  $1/16 = 0.0625$ . The algorithm is completely coherent in this case, and the large amount of interference is used in a destructive way, *subtracting* probability from the searched item.

Remarkably, bit flip errors do not affect the success probability at all, such that  $S(p)$  remains constant at the optimal value, independently of the number of qubits affected. The behavior is easily understood, as in fact the bit flip errors leave the state obtained after applying the Hadamard gates invariant (and in particular: pure). The interference goes down to zero, nevertheless, as it measures coherence using superpositions of all computational states [14], and not pureness of the final state. We have therefore the peculiar situation where in spite of decoherence processes one particular state remains pure (the perfectly equipartitioned superposition of all computational basis states), and since it is that state which is used in the algorithm, the success probability remains unaffected. On the other hand, the interference measure was constructed to measure coherence by the sensitivity

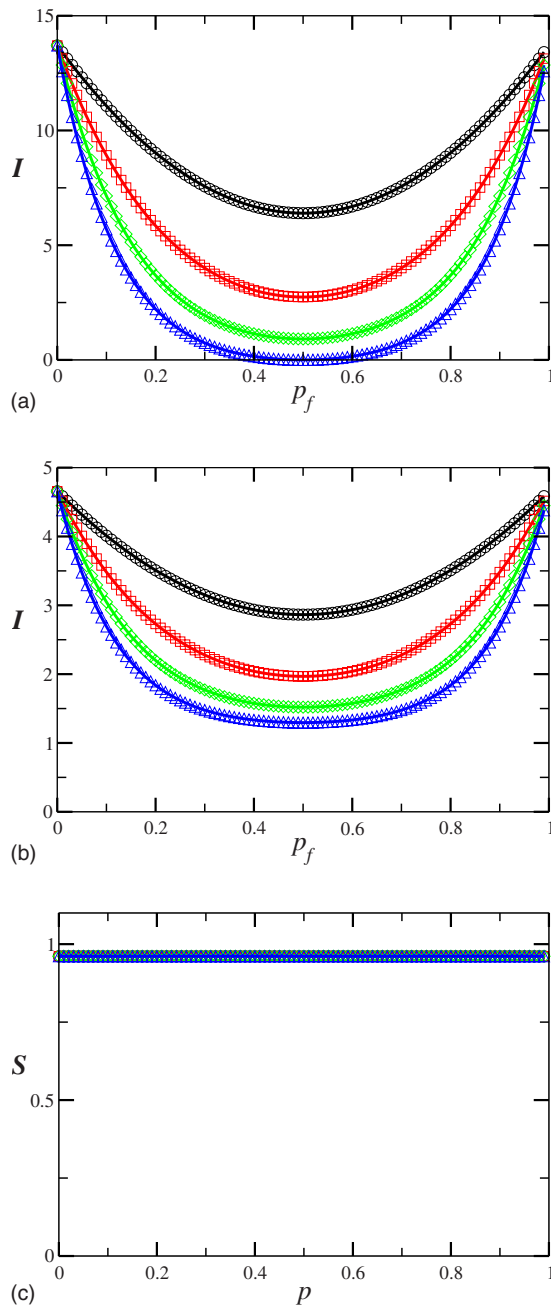


FIG. 9. (Color online) Potentially available interference in the Grover algorithm with decoherence through bit flips during the first Walsh-Hadamard transformation, as a function of the bit flip probability  $p$  after each Hadamard gate (a). Same but for actually used interference (b). Success probability as a function of  $p$  (c). All curves are for  $n=4$ ,  $\alpha=2$ ;  $n_f=1$  black circles,  $n_f=2$  red squares,  $n_f=3$  green diamonds,  $n_f=4$  blue triangles.

of final probabilities to *relative* initial phases between the computational basis input states, and it correctly picks up that the phase coherence *between* all states got lost. Thus in this particular situation, one can have a perfectly well working algorithm which uses, according to our measure, zero potentially available interference. We believe, however, that this case where the coherence of the propagation cannot be measured by the influence of relative phases but only

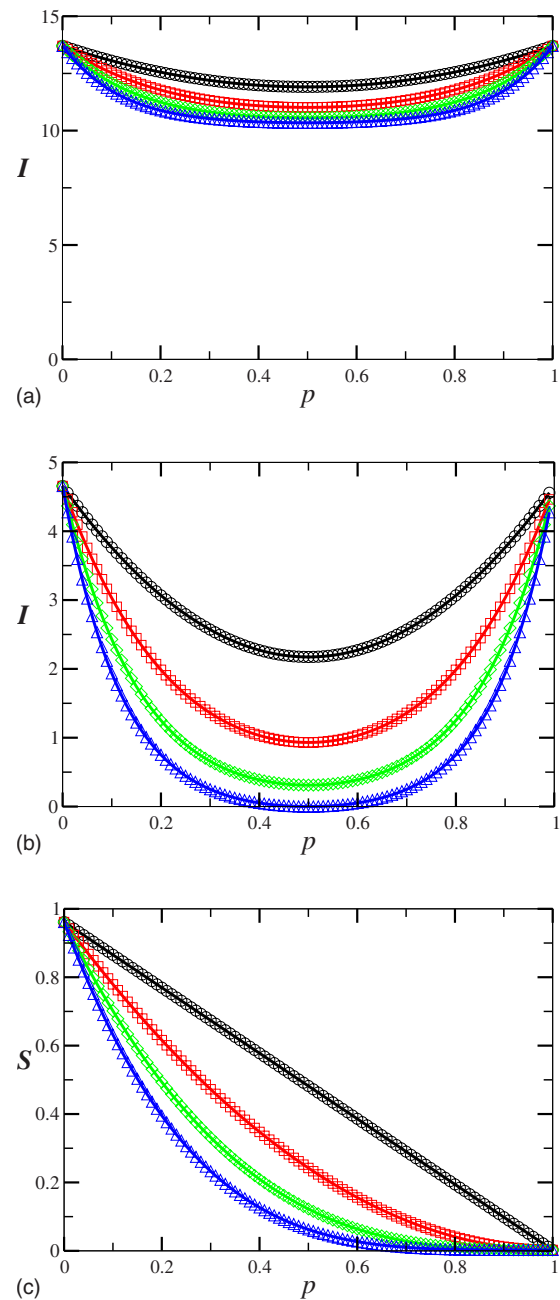


FIG. 10. (Color online) Same as Fig. 9, but for phase flip errors.

through the purity of the final state is highly exceptional and should not exclude a proof that exponential speedup needs exponential interference, if one restricts attention to unitary algorithms, or gives special attention to the exceptional single pure state mentioned. It is also important to note that the actually used interference remains finite at  $p=0.5$  and  $n_f=n$ , and below the already small value  $\mathcal{I} \sim 4$  for the unperturbed algorithm and thus never goes to zero for bit flip errors.

Figures 11 and 12 show the result of decoherence due to bit flips in the Shor algorithm for  $n=12$ ,  $n=9$ , and  $n=6$  qubits. As for the Grover algorithm, bit flips are performed after each of the initial Hadamard gates. However, as these



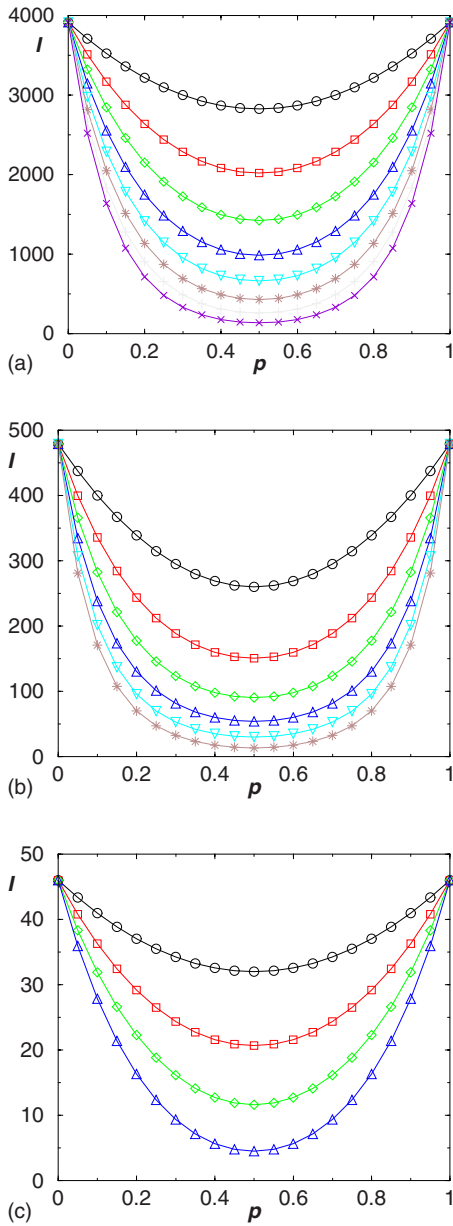


FIG. 11. (Color online) Potentially available interference in the Shor algorithm with decoherence through bit flips during the first Walsh-Hadamard transformation, as a function of the bit flip probability  $p$  after each Hadamard gate, for  $n=12$  (a),  $n=9$  (b),  $n=6$  (c). The symbols are  $n_f=1$  black circles,  $n_f=2$  red squares,  $n_f=3$  green diamonds,  $n_f=4$  blue up triangles,  $n_f=5$  cyan down triangles,  $n_f=6$  brown stars,  $n_f=7$  gray  $\times$ ,  $n_f=8$  violet  $+$ . The corresponding success probability  $S$  is a constant equal to  $S=1$  for all values of  $p$  (data not shown). Here and in the following figures all quantities are averaged over all possible choices of the  $n_f$  qubits in the first register.

Hadamard gates concern only one of the registers, the decoherence process affects only the first two-thirds of qubits. The curves show the effect of decoherence on a growing number of qubits, from  $n_f=1$  to  $n_f=(2/3)n$ , with data averaged over the choice of the  $n_f$  affected qubits. The success probability  $S$  is computed as in Eq. (5) where now  $|\psi_i^{err}|^2$  is replaced by the probability for state  $i$  in the final mixed state.

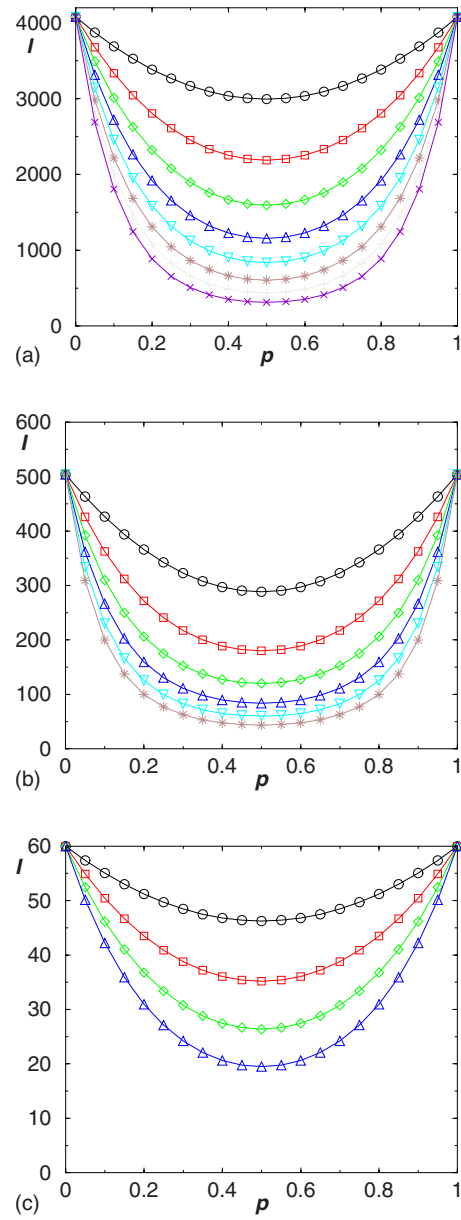


FIG. 12. (Color online) Actually used interference in the Shor algorithm with decoherence through bit flips during the first Walsh-Hadamard transformation, as a function of the bit flip probability  $p$  for  $n=12$  (a),  $n=9$  (b),  $n=6$  (c). The corresponding success probability  $S$  is a constant equal to  $S=1$  for all values of  $p$  (data not shown); symbols as in Fig. 11.

The success probability is constant equal to 1 for all values of  $p$  (data not shown). In contrast, both potentially available and actually used interference are strongly affected by the decoherence. Both quantities decrease from their maximum value at  $p=0$  and  $p=1$  to a minimum at  $p=0.5$ , the potentially available interference decreasing faster. However, the interference never goes down to zero in this setting, as can be seen in the insets of Figs. 11 and 12, contrary to the case of the Grover algorithm above.

In Figs. 11 and 12, only two-thirds of the qubits are affected by the decoherence. This may appear to be the main reason why the potentially available interference does not

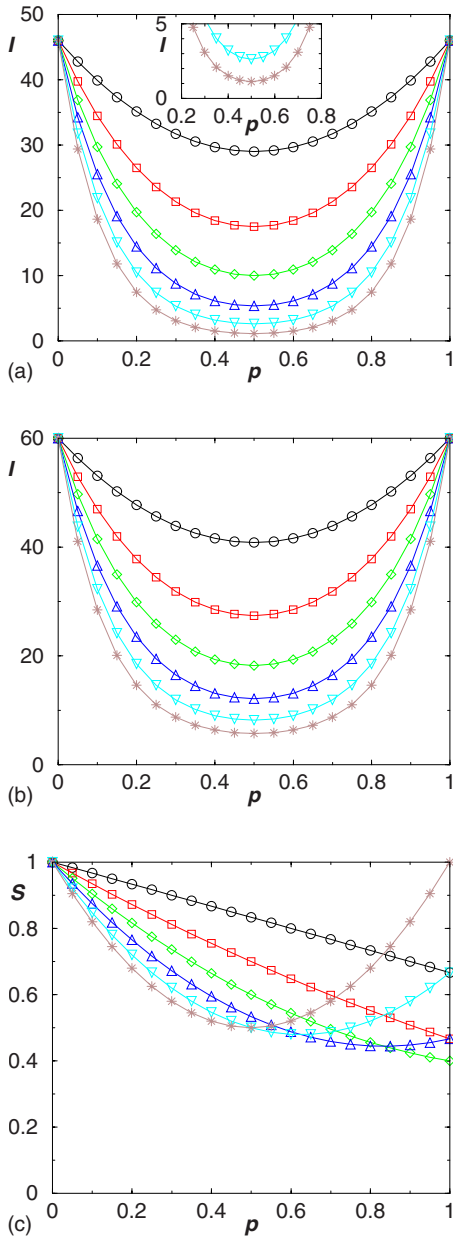


FIG. 13. (Color online) Interference and success probability in the Shor algorithm with decoherence through bit flips during the first Walsh-Hadamard transformation, as a function of the bit flip probability  $p$  after each Hadamard gate, with all  $n$  qubits flipped, for  $n=6$ : potentially available interference (a), actually used interference (b), success probability (c). The inset is a closeup close to the minimum; symbols as in Fig. 11.

decrease to zero in the presence of bit flip decoherence, contrary to the case above with the Grover algorithm. In order to investigate this question in more detail, we studied the interference produced when *all*  $n$  qubits are affected by bit flip decoherence. The results are shown in Fig. 13 for  $n=6$ . Although the two types of interference decrease faster than in Figs. 11 and 12, none of them reaches exactly zero over the whole interval of  $p$  values. Contrary to the case of Figs. 11 and 12, the success probability is now strongly affected by the decoherence and is not preserved: the initial superposed

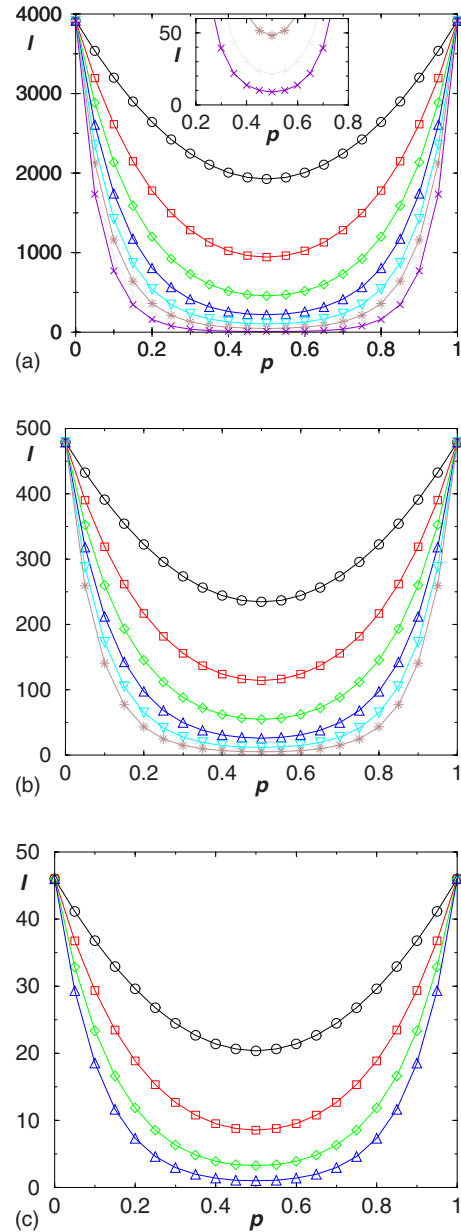


FIG. 14. (Color online) Potentially available interference in the Shor algorithm with decoherence through phase flips during the first Walsh-Hadamard transformation, as a function of the phase flip probability  $p$  after each Hadamard gate, for  $n=12$  (a),  $n=9$  (b),  $n=6$  (c). The inset is a closeup close to the minimum; symbols as in Fig. 11.

state is no longer protected against this type of decoherence, since the second register is not supposed to be in an equal superposition state in the exact algorithm. The data displayed in Figs. 11–13 suggest that for Shor’s algorithm, although it is possible to decrease the interference by a large amount while keeping the success probability constant, it does not seem possible to perform efficiently the computation with zero interference.

Figures 14–16 display data obtained for decoherence through phase flips. As before, phase flips are introduced with probability  $p$  on qubits of the first register after appli-

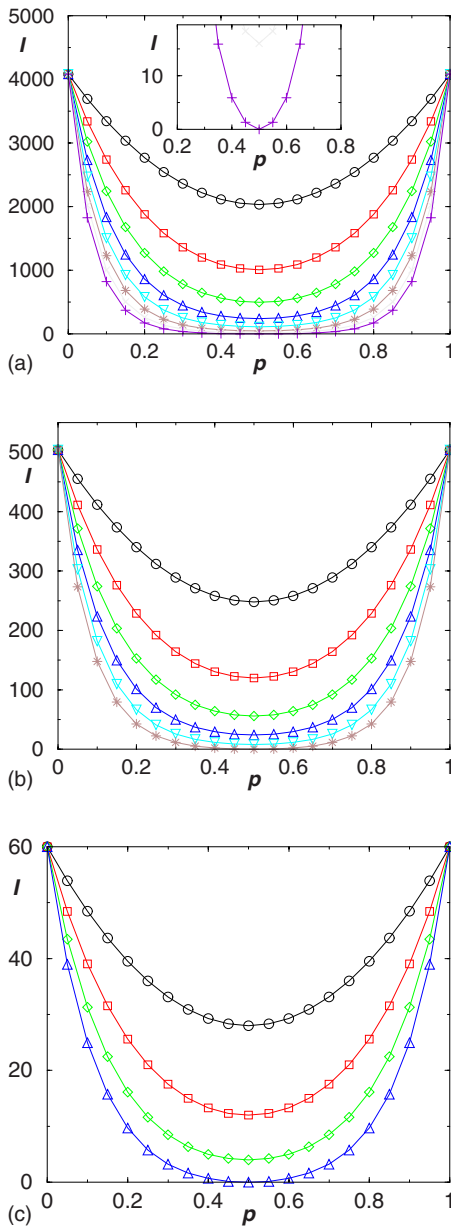


FIG. 15. (Color online) Actually used interference in the Shor algorithm with decoherence through phase flips during the first Walsh-Hadamard transformation, as a function of the phase flip probability  $p$  after each Hadamard gate for  $n=12$  (a),  $n=9$  (b),  $n=6$  (c). The inset is a closeup of the case  $n=12$  close to the minimum, showing that the value  $\mathcal{I}=0$  is indeed reached for  $p=0.5$ .

cation of the Hadamard gates. The data displayed in Figs. 14 and 15 show that interference, both potentially available and actually used, decreases to a minimum at  $p=0.5$ . The minimum is lower than in the case of bit flips, and reaches zero for actually used interference. In the case of potentially available interference, some residual interference is still present when all qubits of the first register are affected. Figure 16 shows that in contrast to the case of bit flip errors, success probability is strongly affected for phase flip errors. It decreases with the number of qubits affected and the value of  $p$  until the algorithm is totally destroyed.

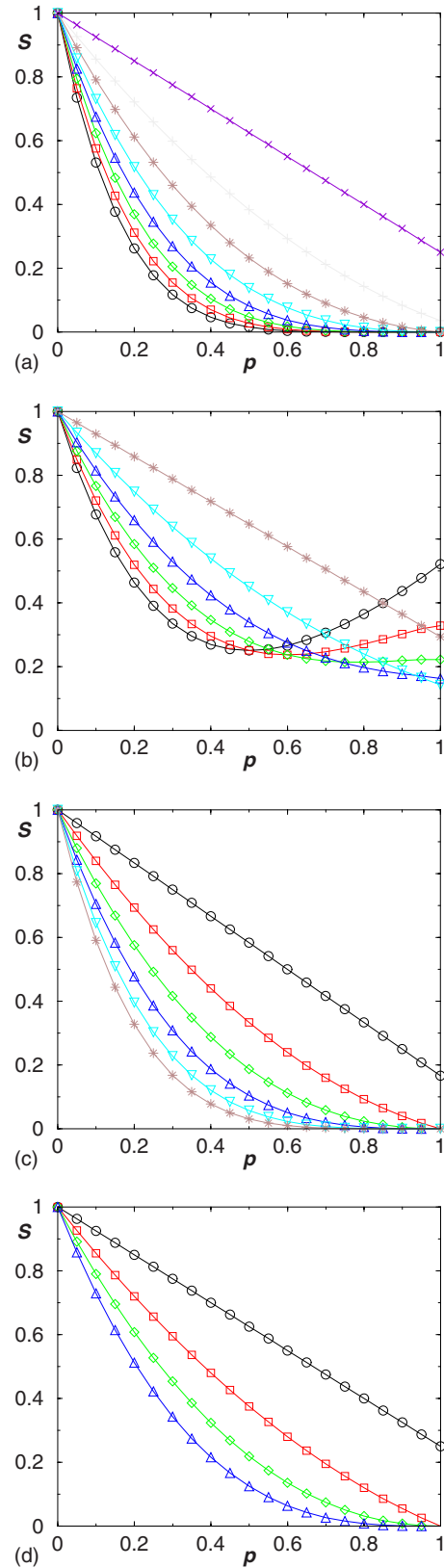


FIG. 16. (Color online) Success probability in the Shor algorithm with decoherence through phase flips during the first Walsh-Hadamard transformation, as a function of the phase flip probability  $p$  after each Hadamard gate for  $n=12$  (a),  $n=9$ ,  $a=3$  (b),  $n=9$ ,  $a=6$  (c),  $n=6$  (d); symbols as in Fig. 11.

Figs. 14 and 15 shows that the increase of interference between  $p=0.5$  and  $p=1$  is not reflected in a similar increase in success probability. The interference produced in this case is useless and does not serve the algorithm. It is similar to the one found for random algorithms in [19], where it was shown that random algorithms produce on average an interference close to the maximum value. The case  $n=9$ ,  $a=3$  is peculiar: in this particular instance (second figure in Fig. 16), after an initial decrease for small values of  $p$ , the success probability increases for larger values of  $p$ , although it never reaches values close to 1. We think this is due to the fact that in this case the period is not a power of 2, and therefore the final wave function is composed of broad peaks which for such small sizes have a significant projection on many basis states of the Hilbert space. A random-type wave function produced by the destroyed Shor algorithm therefore has a much larger projection on such a state than on a state composed of sharp  $\delta$  peaks as in the two other values of  $n$ . To sustain this hypothesis, we computed the success probability for  $n=9$  and  $a=6$ , where the period does divide the Hilbert space dimension, and the final wave function is composed of  $\delta$  peaks. In this case (last figure of Fig. 16) the success probability indeed goes to zero for large  $p$  values.

#### IV. CONCLUSIONS

In this paper we have investigated the role interference plays in quantum algorithms by analyzing its change when errors affect the performance of the algorithms. To this end we subjected Grover's search algorithm and Shor's factoring algorithm to different kinds of errors, namely systematic unitary errors, random unitary errors, and decoherence due to bit flips or phase flips. The study of systematic unitary errors showed that in both algorithms the controlled destruction of interference goes hand in hand with the decay of the success probability. This reinforced the idea that interference is an important ingredient necessary for the functioning of these algorithms.

The case of random unitary errors shows, however, that a large amount of interference is by no means sufficient for the success of a quantum algorithm, since in some cases the interference increases with decreasing success probability. This effect is particularly pronounced for the actually used interference in the case of Grover's algorithm, where the interference increases from about two i-bits for the unperturbed algorithm to an amount of the order  $n$  i-bits, close to the maximum possible value, for sufficiently strong errors. The success probability may decrease even below the classi-

cal value corresponding to unbiased random guessing, meaning that the interference has become destructive. This can also be understood in the context of the recent result that a randomly chosen quantum algorithm leads with very high probability to an amount of interference close to the maximum value [19], such that randomizing a given quantum algorithm with limited interference is very likely to increase the interference, even if the algorithm itself is destroyed in the process. Thus as to be expected, interference needs to be exploited in the proper way to be useful.

The study of decoherence led to more complex results. Phase flip errors destroy interference and in parallel decrease the success probability, in the same way as systematic errors. Interference decreases for small error probabilities, and reappears again as destructive interference for error probabilities approaching 1, leading to success probabilities which can even drop below the classical value. In contrast, bit flip errors performed after each initial Hadamard gate do not reduce the success probability of the algorithm, while affecting the interference produced. In the case of Grover's algorithm, the potentially available interference can even go all the way down to zero while the performance of the algorithm is unaffected. This surprising result is due to the symmetry of the equipartitioned state used as the initial state in Grover's algorithm, which is invariant under bit flips. It should be remarked, however, that the actually used interference does not go to zero for Grover's algorithm. As concerns Shor's algorithm, the bit flips destroy part but not all of the interference, both potentially available and actually used, while also keeping constant the success probability of the algorithm. These results show that in general it is possible to reduce substantially the interference produced while keeping the efficiency of the algorithm. Grover's algorithm seems to run correctly with zero potentially available interference, but not with zero actually used interference. In contrast, in none of our simulations was Shor's algorithm found to run efficiently without some interference left, potentially available or actually used. Thus while the relationship between interference and success probability is a complex one, our results show quantitatively that interference is an important ingredient of a quantum algorithm.

#### ACKNOWLEDGMENTS

We thank IDRIS in Orsay and CALMIP in Toulouse for the use of their computers. This work was supported in part by the Agence National de la Recherche (ANR), project INFOSYSQQ, Contract No. ANR-05-JCJC-0072, and the EC IST-FET project EUROSQIP.

- 
- [1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000).
- [2] P. W. Shor, in Proceedings of the 35th Annual Symposium of Foundations of Computer Science, edited by S. Goldwasser

(IEEE Computer Society, Los Alamitos, CA, 1994), pp. 124–134.

- [3] L. K. Grover, Phys. Rev. Lett. **79**, 325 (1997).
- [4] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, Fortschr. Phys. **46**, 493 (1998).



- [5] G. Brassard, M. M. P. Høyer, and A. Tapp, in *Quantum Computation and Quantum Information: A Millenium Volume*, Contemporary Mathematics Series Vol. 305, edited by S. J. Lomonaco, Jr. and H. E. Brandt (AMS, Providence, 2002), pp. 53-74.
- [6] B. Georgeot, Phys. Rev. A **69**, 032301 (2004).
- [7] D. Deutsch, Proc. R. Soc. London, Ser. A **400**, 97 (1985).
- [8] R. Jozsa and N. Linden, Proc. R. Soc. London, Ser. A **459**, 2011 (2003).
- [9] D. Bruß, J. Math. Phys. **43**, 4237 (2002).
- [10] A. De, U. Sen, M. Lewenstein, and A. Sanpera, in *Lectures on Quantum Information*, edited by D. Bruss and G. Leuchs (Wiley-VCH, Weinheim, 2007), pp. 123-146.
- [11] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, Proc. R. Soc. London, Ser. A **454**, 339 (1998).
- [12] C. H. Bennett and D. P. DiVincenzo, Nature (London) **404**, 247 (2000).
- [13] M. Beaudry, J. M. Fernandez, and M. Holzer, Theor. Comput. Sci. **345**, 206 (2005).
- [14] D. Braun and B. Georgeot, Phys. Rev. A **73**, 022314 (2006).
- [15] D. Aharonov, Z. Landau, and J. Makowsky, e-print arXiv:quant-ph/0611156.
- [16] N. Yoran and A. J. Short, Phys. Rev. A **76**, 060302(R) (2007).
- [17] D. E. Browne, New J. Phys. **9**, 146 (2007).
- [18] A. O. Lyakhov, D. Braun, and C. Bruder, Phys. Rev. A **76**, 022321 (2007).
- [19] L. Arnaud and D. Braun, Phys. Rev. A **75**, 062314 (2007).