Error reduction of quantum algorithms

Debajyoti Bera[*] and Tharrmashastha P.V.[†]

Indraprastha Institute of Information Technology, Okhla Industrial Estate Ph-III, New Delhi 110020, India

(Received 20 February 2019; published 19 July 2019)

We present a technique to reduce the error probability of quantum algorithms that determine whether its input has a specified property of interest. The standard process of reducing this error is statistical processing of the results of multiple independent executions of an algorithm. Denoting by ρ an upper bound of this probability (wlog; assume $\rho \leqslant \frac{1}{2}$), classical techniques require $O(\frac{\rho}{[(1-\rho)-\rho]^2})$ executions to reduce the error to a negligible constant. We investigate when and how quantum algorithmic techniques like amplitude amplification and estimation may reduce the number of executions. On one hand, the former idea does not directly benefit algorithms that can err on both yes and no answers and the number of executions in the latter approach is $O(\frac{1}{(1-\rho)-\rho})$. We propose a quantum approach called amplitude separation that combines both these approaches and achieves $O(\frac{1}{\sqrt{1-\rho}-\sqrt{\rho}})$ executions, which betters existing approaches when the errors are high. In the multiple-weight decision problem, the input is an n-bit Boolean function $f()$ given as a black box and the objective is to determine the number of x for which $f(x) = 1$, denoted $wt(f)$, given some possible values $\{w_1, \ldots, w_k\}$ for $wt(f)$. When our technique is applied to this problem, we obtain the correct answer, maybe with a negligible error, using $O(\log_2 k\sqrt{2^n})$ calls to $f()$, which shows a quadratic speedup over classical approaches and currently known quantum algorithms.

DOI: 10.1103/PhysRevA.100.012331

I. INTRODUCTION

Many of the famous problems for which early quantum algorithms were designed are "decision problems," i.e., the solution of the problem requires identifying whether an input satisfies a given property. Inputs which evoke a "yes" answer are called yes inputs, and similarly, those that evoke a "no" answer are called no inputs. Quantum algorithms being inherently probabilistic, it is possible for such algorithms to be error prone. An algorithm that makes the correct decision for every input is termed an "exact algorithm"; otherwise, the algorithm is a probabilistic one. This paper concerns probabilistic quantum algorithms and the techniques to reduce their error. Specifically, we look at algorithms with *bounded nonzero errors* in the following sense: the probability of error for yes inputs is upper bounded by $\rho_y \in (0, 1)$ and the probability of error for no inputs is upper bounded by $\rho_n \in (0, 1)$. Without loss of generality, we assume that $\rho_n \leqslant \rho_y$; otherwise, the notion of yes and no inputs can be interchanged; similarly, we can assume that $\rho_y + \rho_n \leqslant 1$ because, otherwise, $(1 - \rho_y) + (1 - \rho_n) \leqslant 1$ so we can simply swap the yes-no answers.

Setting aside the bespoken error reduction tactics, our focus is on black-box techniques for reducing error that applies to any algorithm. This is routinely done for day-to-day classical algorithms by running them independently enough times and analyzing their output. For example, if ρ_n is 0, then it suffices to simply output yes if any execution outputs yes. In fact the versatile *amplitude amplification* (AA) technique used in quantum algorithms can also be used in such cases [1,2].

However, directly applying AA is inadequate in reducing errors of algorithms if both $\rho_y > 0$ and $\rho_n > 0$. What AA does is nonlinearly multiply the probability that the output state of an algorithm is observed in a particular state (for which the algorithm outputs yes). Therefore, when both ρ_y and ρ_n are nonzero, there is a chance of error for every input. No matter which state is used for amplification, one of ρ_y and ρ_n will decrease but the other will increase, rendering AA ineffective.

There are standard "classical" techniques for handling such algorithms. Suppose that $\mathcal{A}$ denotes the algorithm with error bounds ρ_y and ρ_n. Therefore, for a yes input, the probability of observing a "good" output state would be at least $1 - \rho_y$, and for a no input, the probability of observing the same would be at most ρ_n. One manner in which the error of $\mathcal{A}$ can be reduced (to, say, some δ) is to estimate this probability with a precision of $\pm\frac{1}{2}[(1 - \rho_y) - \rho_n]$ and with an error probability of at most δ. For a yes input, the estimate will be less than $\frac{1}{2}[(1 - \rho_y) + \rho_n]$ with a probability of less than δ, and for a no input, the estimate will be more than the same threshold with a probability of less than δ. Thus, to reduce the error of $\mathcal{A}$, it suffices to estimate the probability and claim that the input is a yes input if the estimate is more than the threshold and a no input otherwise. Estimating the probability requires running $\mathcal{A}$ multiple times and calculating the fraction of times the good state is observed, and to achieve this within the required bounds requires $\tilde{O}((1 - \rho_y)/[(1 - \rho_y) - \rho_n]^2)$ executions[1] of $\mathcal{A}$.

[*] dbera@iiitd.ac.in

[†] tharmasasthapv@gmail.com

[1] $\tilde{O}()$ hides additional insignificant log-factors within $O()$.

Another possibility is the use of quantum *amplitude estimation* to estimate the probability that the output of any algorithm is observed to be in a good state. The probability can be estimated with any required precision; there is a chance of error but that too can be controlled at the expense of more operations. Use of this technique reduces the number of executions to $\tilde{O}(1/[(1-\rho_y)-\rho_n])$.

However, both these techniques become inefficient when $1-\rho_y \approx \rho_n$ and both of these are small. This paper presents the *amplitude separation* (AS) technique, a combination of amplitude amplification and estimation, to reduce both ρ_y and ρ_n, even when they are nonzero, and the number of calls required is only $\tilde{O}(1/[\sqrt{1-\rho_y}-\sqrt{\rho_n}])$. As illustrated in Fig. 1, this method outperforms the earlier techniques when $1-\rho_y \to 0$ and $\rho_n \to 0$.

If the errors for all the yes inputs are the same and equal to ρ_y, and similarly, those for all the no inputs are equal to ρ_n, and if ρ_y and ρ_n are known, then it is possible to perform a better error reduction. Using AA in a sophisticated manner, Bera has shown how to obtain an algorithm that correctly outputs "accept" for all yes inputs and outputs "reject" for all no inputs *without* any probability of error (see the result that EBQP = EQP in [2]). However, that technique crucially uses the information that all error probabilities equal either ρ_y or ρ_n and are known—something which we relax in this paper. Furthermore, the objective of that work was to design an errorless method, whereas we allow error, albeit tunable, as a parameter.

An immediate application of our method is an efficient bounded-error algorithm for the *multiple-weight decision* problem (MWDP). The MWDP is a generalization of the *exact-weight decision* problem (EWDP), which, in turn, generalizes the Deutsch-Jozsa problem and the Grover unordered search problem [3–5]. The input to the MWDP problem is an n-bit Boolean function $f()$ given in the form of a black box and a list of k possible weights of $f()$: $\{0 < w_1 < w_2 < \ldots < w_k < 2^n\}$ along with the promise that $wt(f) = w_i$ for some i. The weight of $f()$ is defined as $wt(f) = |\{x \in \{0,1\}^n : f(x) = 1\}|$. The objective is to determine the actual weight of $f()$ by making very few calls to $f()$. The EWDP can be defined as the MWDP with $k = 2$.

Optimal algorithms for the EWDP are known that determine the weight exactly and make $\Theta(\sqrt{w_2(2^n - w_1)}/(w_2 - w_1))$ calls [5,6], which can be as large as $\sqrt{2^n}$ when $w_1 \approx w_2 \ll 2^n$. Current algorithms for the MWDP give the exact answer and follow two approaches [5]. Either they make $k-1$ calls to an EWDP algorithm and, thus, could make nearly 2^n calls to $f()$ (when $k \approx \sqrt{2^n}$) or they use a quantum counting algorithm [1] to count the number of solutions of $f(x) = 1$ but that could also require nearly 2^n calls [when $wt(f) \approx 2^n$].

We use our amplitude separation technique to give an algorithm for the MWDP with a *small error* that makes $O(\log_2 k\sqrt{2^n})$ calls to $f()$. This is achieved by first designing a bounded-error algorithm for a variation of the EWDP in which we have to determine whether $wt(f) \leqslant w_1$ or $wt(f) \geqslant w_2$ given $0 < w_1 < w_2 < 2^n$.

Our approach uses the concept of amplitude amplification and amplitude estimation. Even though we describe our technique in algorithms that take their input in the form of oracle operators, we can use a method outlined in a work by Bera to apply AA, and hence the technique in this paper, to algorithms that are given their input $x \in \{0,1\}^n$ in the form of an initial state $|x\rangle$ (along with ancillary qubits in a fixed state) [2].

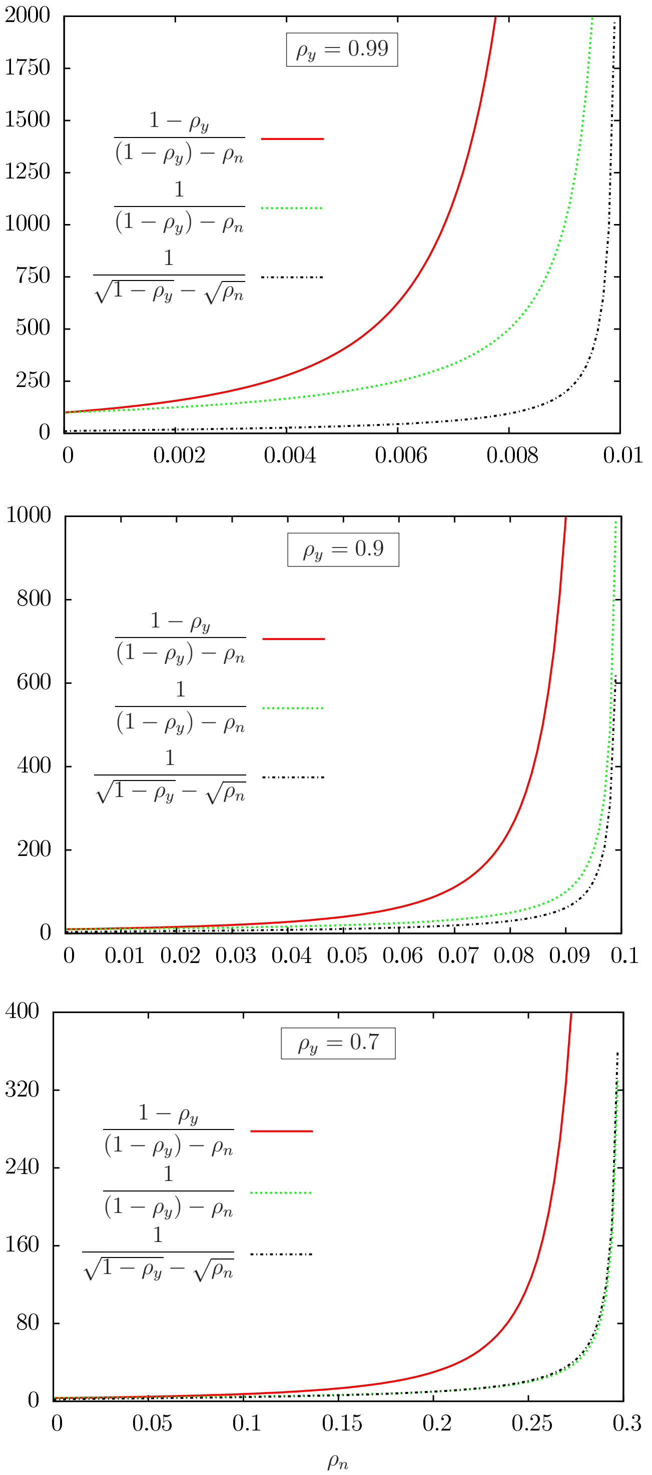

FIG. 1. Comparison of the dominant terms in the query complexity of the classical, amplitude-estimation-based and amplitude-separation-based error reduction algorithms; the Y axis shows the values of the dominant terms and the X axis shows ρ_n. The plot for $\rho_y = 0.5$ (not shown) is identical to that for $\rho_y = 0.7$, indicating that amplitude separation works best for high values of ρ_y and is reasonably good for lower values.

II. BACKGROUND

Our method makes subtle use of the well-known quantum amplitude estimation algorithm so we briefly discuss the relevant results along with the specific extension that we require. Suppose we have an n-qubit quantum algorithm $\mathcal{A}$ that is said to "accept" its input when its output qubit is observed in a specific "good state" upon the final measurement. We use p to denote the probability of observing this good state for a specific input. The value of p can be estimated by purely classical means, e.g., by running the algorithm multiple times and computing the fraction of times the good state is observed. Amplitude estimation is a quantum technique that essentially returns an estimate by making fewer calls to the algorithm compared to this technique.

The estimation method uses two parameters, k and m, that we fix later. The first and basic quantum amplitude estimation algorithm (AmpEst) was proposed by Brassard *et al.* [1]; it acts on two registers of m and n qubits, makes 2^m calls to controlled-$\mathcal{A}$, and outputs a $\tilde{p} \in [0, 1]$ that is a good approximation of p in the following sense.

Theorem II.1. The AmpEst algorithm returns an estimate $\tilde{p}$ that has a confidence interval $|p - \tilde{p}| \leqslant 2\pi k \frac{\sqrt{p(1-p)}}{2^m} + \pi^2 \frac{k^2}{2^{2m}}$ with a probability of at least $\frac{8}{\pi^2}$ if $k = 1$ and with a probability of at least $1 - \frac{1}{2(k-1)}$ if $k \geqslant 2$. If $p = 0$ or 1, then $\tilde{p} = p$ with certainty.

The AmpEst algorithm can be used to estimate p with the desired accuracy (at least 3/4) and error. We now present an extension to the above theorem to obtain an estimation with an *additive error*, say denoted ϵ, that is at most 1/4. We use δ to denote the maximum permissible error. To obtain this estimation, we run AmpEst, presented above, using $k = 1$ and m such that $2^m = \lceil \frac{3\pi}{2\epsilon} \rceil$. AmpEst will be run $7(\ln \frac{1}{\delta})^{1/3} = \Theta(\ln \frac{1}{\delta})$ times to obtain that many estimates of p and the median of these obtained estimates is then returned as $\tilde{p}$. The total number of calls to controlled-$\mathcal{A}$ is, therefore, $O(\frac{1}{\epsilon} \ln \frac{1}{\delta})$. Next we analyze the accuracy of $\tilde{p}$.

Since $p(1 - p) \leqslant 1/4$ for any p, $\frac{1}{2}e^{2\epsilon} \geqslant \sqrt{\frac{1}{4} + \epsilon}$ ($\because$ $\epsilon \leqslant 1/4$), and $3 \geqslant 1 + e^{2\epsilon}$, it can be shown that $\frac{3\pi}{2\epsilon} \geqslant \frac{\pi}{\epsilon}[\sqrt{p(1 - p)} + \sqrt{p(1 - p) + \epsilon}]$, and for $2^m \geqslant \frac{3\pi}{2\epsilon}$ it can be further shown that $2\pi \frac{\sqrt{p(1-p)}}{2^m} + \frac{\pi^2}{2^{2m}} \leqslant \epsilon$. Therefore, for the setting of parameters specified above, using Theorem II.1 we obtain an estimate $\tilde{p}$ of p in each run of AmpEst such that $\Pr[|p - \tilde{p}| \geqslant \epsilon] \leqslant \delta$ with a probability of error of at most $1 - \frac{8}{\pi^2}$, which means that the median of any number of such estimates also satisfies the same upper bound on its additive error. The overall error can be reduced to any desired δ by taking a median of $\Theta(\ln \frac{1}{\delta})$ estimates and this is a standard error reduction technique whose proof uses Chernoff bounds.

So, to summarize this section, we have explained a method that returns an estimate $\tilde{p}$ of the success probability p of a quantum algorithm $\mathcal{A}$ such that $\tilde{p} - \epsilon \leqslant p \leqslant \tilde{p} + \epsilon$ with a probability of at least $1 - \delta$. The method makes altogether $O(\frac{1}{\epsilon} \ln \frac{1}{\delta})$ calls to $\mathcal{A}$.

III. AMPLITUDE SEPARATION ALGORITHM

Now we introduce the amplitude separation (AS) problem and describe an algorithm that is our main technical tool. Suppose we are given a quantum algorithm $\mathcal{A}$ for a decision problem; without loss of generality, we can assume that the algorithm outputs "yes" if the output qubit is observed in state $|1\rangle$ and "no" if the observed state is $|0\rangle$. Let p denote the probability of observing the output qubit in state $|1\rangle$. Suppose it is also given that for yes inputs $p \geqslant t$ and for no inputs $p \leqslant t'$ given $0 < t' < t < 1$. The AS problem is to determine whether a given input is a yes input or a no input by making black-box calls to $\mathcal{A}$.

There are, of course, several alternative strategies. Consider the completely classical method of making multiple observations of $\mathcal{A}$ and deciding based on the number of times the output qubit is observed in state $|1\rangle$; the number of required queries to $\mathcal{A}$ can be obtained using probabilistic techniques (involving the Chernoff bound) and scales as $O(\frac{1}{t-t'})$. Another possibility would have been to use the quantum amplitude estimation methods. They come in various flavors and a quick summary of the relevant ones is presented in Sec. II. If we use the additive-accuracy estimation, then also the number of queries scales as in the previous case. One can also design an estimator with a relative accuracy, but to obtain an upper bound on the number of queries, one would require a lower bound on p which need not be known.

The decision algorithm is Algorithm 1. For simplicity of analysis, we use a *separation* variable β chosen such that $t' = \beta^2 t$. On a high level, our algorithm first amplifies the amplitude of state $|1\rangle$ of the output qubit and only after that applies amplitude estimation since amplified probabilities have a larger gap and, therefore, are easier to distinguish. Recall that applying AA k_i times increases the corresponding probability from any $\sin^2\theta$ to $\sin^2[(2k_i + 1)\theta]$. We see below how this allows us to solve the problem with a number of queries to $\mathcal{A}$ that scales as $O(\frac{1}{\sqrt{t}})$. For amplitude estimation we use the additive-accuracy estimator with additive error ϵ' and error δ', which is explained in Sec. II.

Algorithm 1 Amplitude separation ($\mathcal{A}$)

```
**Parameter:** $0 < t' < t \leqslant 1$ (thresholds)
**Parameter:** $\delta$ (error)
**Denote:** $|in\rangle$ as the initial state of $\mathcal{A}$
Set $\beta = \sqrt{t/t'}$, $\tau = \sin^{-1}\sqrt{t}$, $s = \lfloor \log_3 \frac{\pi}{4\tau} \rfloor$, $\delta' = \frac{\delta}{(1+s)}$.
Set $\epsilon' = \frac{1}{2}(\sin^2 3^s\tau - \sin^2 3^s\beta\tau)$.
Set $\epsilon^* = \frac{1}{2}(\sin^2 3^s\tau + \sin^2 3^s\beta\tau)$.
**for** $i = 0$ to $s$ **do**
    Set $k_i = \frac{1}{2}(3^i - 1)$.
    $|\phi\rangle \leftarrow$ apply amplitude amplification $k_i$ times to $A|in\rangle$.
    $\tilde{p} \leftarrow$ estimate probability of observing the output qubit of $|\phi\rangle$ in state $|1\rangle$ using "amplitude estimation with additive error $\epsilon'$ and error $\delta'$."
    **If** $\tilde{p} \geqslant \epsilon^*$:
    **return** "accept" (i.e., claim that $p \geqslant t$).
**end for**.
**return** "reject" (i.e., claim that $p \leqslant t' = \beta^2 t$).
```

Now we explain how Algorithm 1 makes $\tilde{O}(\frac{1}{\sqrt{t}-\sqrt{t'}} \log \frac{1}{\delta})$ calls to $\mathcal{A}$ (and $\mathcal{A}^\dagger$) and, with a probability of error of at most δ, returns accept if $p \geqslant t$ or returns reject if $p \leqslant t'$.

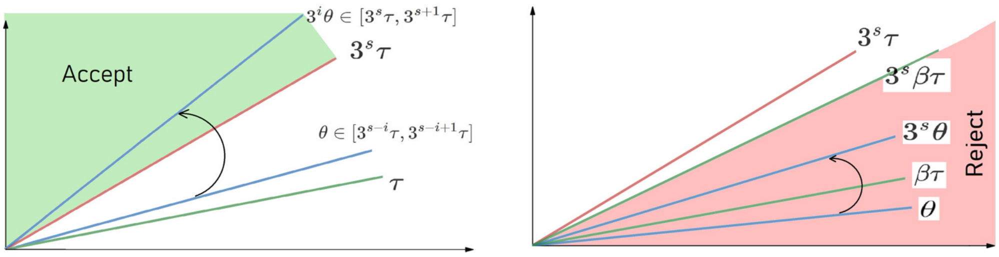

FIG. 2. The case $\sin\theta \geqslant \sin\tau$ (left) and the case $\sin\theta \leqslant \beta\sin\tau$ (right) of Algorithm 1, before and after amplification. The Y axis represents the probability p of observing $|1\rangle$ and the X axis represents the probability of observing $|0\rangle$.

To explain the claim we use the following two trigonometric facts: (i) for any $a < 1$ and $t \leqslant \pi/2$, $\sin\theta \leqslant a\sin t$ implies $\theta \leqslant at$, and (ii) for any $a < 1$ and $t \leqslant \pi/4$, $a\sin t \leqslant \sin at \leqslant \sqrt{a}\sin t$ (proof of these are included in the Appendix).

Consider $\theta \in [0, \frac{\pi}{2}]$ such that $p = \sin^2\theta$ and $\tau \in [0, \frac{\pi}{2}]$ such that $t = \sin^2\tau$. Then the two cases of θ that are under consideration would be (i) $\sin\theta \geqslant \sin\tau$ and (ii) $\sin\theta \leqslant \beta\sin\tau$. Following a common technique of analyzing amplitude amplification techniques [7], it will be helpful to break the interval $[\tau, \frac{\pi}{2}]$ into these intervals:

$$R_0 = \left[\psi, \frac{\pi}{2}\right], R_1 = \left[\frac{1}{3}\psi, \psi\right], R_2 = \left[\frac{1}{3^2}\psi, \frac{1}{3}\psi\right]$$
$$\dots R_i = \left[\frac{1}{3^i}\psi, \frac{1}{3^{i-1}}\psi\right]\dots R_s = \left[\tau = \frac{1}{3^s}\psi, \frac{1}{3^{s-1}}\psi\right],$$

where $\psi = 3^s\tau$ and $s = \lfloor\log_3\frac{\pi}{4\tau}\rfloor$. It can quickly be verified that $3^s\tau \in (\frac{\pi}{12}, \frac{\pi}{4}]$.

First, consider the case of $\sin\theta \geqslant \sin\tau$, which is equivalent to $\theta \geqslant \tau$ (refer to Fig. 2). Note that for any $\theta \in [\tau, \frac{\pi}{2}]$, there exists some R_i such that $\theta \in R_i$. Consider the ith iteration in the algorithm, in which we set $k_i = \frac{1}{2}(3^i - 1)$. For any $\theta \in R_{i\neq 0}$, $(2k_i + 1)\theta = 3^i\theta \in [3^s\tau, 3\cdot 3^s\tau] \subseteq [3^s\tau, 3\pi/4]$, and for $\theta \in R_0$, $(2k_i + 1)\theta \in [\frac{\pi}{12}, \frac{\pi}{2}] \subseteq [3^s\tau, 3\pi/4]$. Since $3^s\tau \in (\frac{\pi}{12}, \frac{\pi}{4}]$, therefore, after amplification $\sin^2[(2k_i + 1)\theta] \geqslant \sin^2 3^s\tau$. So, the probability p of observing the output qubit in $|1\rangle$ satisfies $p \geqslant \sin^2 3^s\tau$. Therefore, using additive amplitude estimation with ϵ' and δ' as specified in the algorithm will ensure that $\tilde{p} \geqslant p - \epsilon' \geqslant \epsilon^*$ holds with a probability of at least $1 - \delta'$. Hence, the probability that the algorithm will return accept in the ith iteration is at least $1 - \delta'$ and the probability that the algorithm will *correctly* return accept eventually is also at least $1 - \delta' \geqslant 1 - \delta$.

Next, consider the case where $\sin\theta \leqslant \beta\sin\tau$ (refer to Fig. 2). As per the trigonometric claim above, this implies that $\theta \leqslant \beta\tau$. Therefore, for any $i = 1\dots s$, $(2k_i + 1)\theta \leqslant (2k_i + 1)\beta\tau \leqslant 3^s\beta\tau$. This implies that the probability p defined above satisfies $p \leqslant \sin^2(3^s\beta\tau)$. Again, using the additive amplitude estimation in a manner similar to that above will ensure that $\tilde{p} \leqslant p + \epsilon' \leqslant \epsilon^*$ with a probability of at least $1 - \delta'$. Hence, the probability that the algorithm will return accept in a specific iteration is at most δ'. Therefore, the probability that the algorithm will return accept in *any* of the $i = 0\dots s$ iterations is at most $(1 + s)\delta' = \delta$, which means that the probability that the algorithm will *correctly* return reject is also at least $1 - \delta$.

Having shown that Algorithm 1 returns the correct answer to its decision problem with an error of at most δ, now we explain the query complexity of the algorithm. We use M to denote the number of queries made by the additive amplitude estimation algorithm with parameters ϵ' and δ'; it is shown in Sec. II that $M = O(\frac{\pi}{\epsilon'}\log\frac{1}{\delta'})$. We first need a lower bound on $\epsilon' = \frac{1}{2}(\sin^2 3^s\tau - \sin^2 3^s\beta\tau)$. Using the fact that $3^s\tau \in (\frac{\pi}{12}, \frac{\pi}{4}]$ and the trigonometric facts stated above we derive the following:

$$\sin^2 3^s\tau - \sin^2 3^s\beta\tau \geqslant \sin^2 3^s\tau - \beta\sin^2 3^s\tau$$
$$= (1 - \beta)\sin^2 3^s\tau > (1 - \beta)\frac{\pi}{12}.$$

Therefore, hiding all constants in the big-O notation, $M = \tilde{O}(\frac{1}{(1-\beta)}\log\frac{1}{\delta})$. Now, in Algorithm 1, we can see that the oracle $\mathcal{A}$ is called a total of $(1 + M)k_i$ times at each iteration as the oracle is explicitly called k_i times during the amplitude amplification and the amplitude estimation subroutine itself calls the amplitude amplification M times. So, the total number of calls to the oracle in the algorithm can be expressed as

$$\sum_{i=0}^{s}(1 + M)k_i = \frac{1}{2}(1 + M)\sum_{i=0}^{s}(3^i - 1) < \frac{1}{2}(1 + M)\frac{3\pi}{8\tau}$$
$$= \tilde{O}\left(\frac{1}{(1 - \beta)\tau}\log\frac{1}{\delta}\right)$$
$$= \tilde{O}\left(\frac{1}{(1 - \beta)\sqrt{t}}\log\frac{1}{\delta}\right),$$

where we have used $\frac{1}{\sin^{-1}\sqrt{t}} < \frac{1}{\sqrt{t}}$ in the last inequality.

Suppose $\mathcal{A}$ has bounded errors, say ρ_n and ρ_y; then for no inputs $p \leqslant \rho_n$, and for yes inputs $p \geqslant (1 - \rho_y)$. Further suppose that we want to reduce its error to at most $\delta < \{\rho_n, \rho_y\}$. Algorithm 1 can be applied to $\mathcal{A}$ by setting parameters t to $1 - \rho_y$ and t' to ρ_n and, as shown above, will return accept for yes inputs, as well as returning reject for no inputs, both with a probability of at least $1 - \delta$. What we obtain is an algorithm that acts on the same input state as $\mathcal{A}$, and is observed using the same measurement operators, but makes at most an error of δ in identifying yes and no inputs. This is our proposal to reduce the error of $\mathcal{A}$ in a generic manner. The number of calls

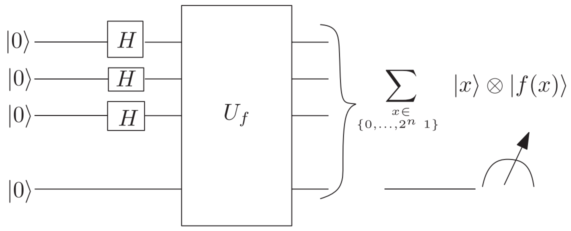
$|0\rangle$
$|0\rangle$
$|0\rangle$
$|0\rangle$
H
H
H
U_f
$\sum_{x \in \{0,\ldots,2^n\ 1\}}$
$|x\rangle \otimes |f(x)\rangle$

FIG. 3. Quantum circuit for the WDP with bounded error.

that will be made to $\mathcal{A}$ (and $\mathcal{A}^\dagger$) in the reduced error algorithm will be at most $O(\frac{1}{\sqrt{1-\rho_y}-\sqrt{\rho_n}} \log \frac{1}{\delta})$.[2]

IV. WEIGHT DECISION ALGORITHM

Given an n-bit Boolean function $f(\,)$ and two parameters, $0 < k_1 < k_2 < 2^n$, suppose it is given that either $wt(f) \leqslant k_1$ or $wt(f) \geqslant k_2$. We define the weight decision problem, denoted WDP_{k_1,k_2}, as the question of determining whether $wt(f) \leqslant k_1$ or $wt(f) \geqslant k_2$. The objective is to minimize the number of calls to $f(\,)$ that is given as input in the usual form of a black-box operator $U_f : |x\rangle|b\rangle \mapsto |x\rangle|b \oplus f(x)\rangle$, where $x \in \{0,1\}^n$, $b \in \{0,1\}$.

The WDP is fairly versatile in its applicability to Boolean function problems. For example, the EWDP is a restricted version of the WDP, where it is given that either $wt(f) = k_1$ or $wt(f) = k_2$ and the problem is to identify which case it is. The decision version of the unordered "Grover's" search problem is to identify whether $wt(f) = 0$ or $wt(f) \geqslant 1$, which is $\text{WDP}_{0,1}$. The Deutsch problem and the Deutsch-Jozsa problem act on Boolean functions that are either constant or balanced and their objective is to determine which one it is; for n-bit functions this is equivalent to identifying whether $wt(f) \in \{0, 2^n\}$ or $wt(f) = 2^{n-1}$. Following the technique suggested by Bera [8], one can define the function $g(x) = f(x) \oplus f(0)$; both problems can now be reformulated as the EWDP with weights 0 and 2^{n-1} and with the function $g(\,)$ as input.

There is a very simple quantum algorithm for $\text{WDP}_{\text{k}_1,\text{k}_2}$, illustrated in Fig. 3. For ease of explanation, we recast the problem as a decision problem; we denote functions for which $wt(f) \geqslant k_2$ as yes inputs and functions for which $wt(f) \leqslant k_1$ as no inputs. Consider the algorithm that first runs the above circuit and then outputs yes [i.e., claims that the function satisfies $wt(f) \geqslant k_2$] if the last qubit is observed in state $|1\rangle$ upon measurement and outputs no otherwise. If the input is a yes input, then the probability of error is at most $\rho_y = (1 - k_2)/2^n$, and if the input is a no input, then the probability of error is at most $\rho_n = k_1/2^n$. These errors can be reduced to any δ by using the above algorithm (in Fig. 3) as $\mathcal{A}$ in Algorithm 1. The number of calls to $\mathcal{A}$, and so to $f(\,)$, would be $O(\frac{\sqrt{2^n}}{\sqrt{k_2}-\sqrt{k_1}})$; this is asymptotically optimal in n for constant k_1 and k_2 due to the fact that the WDP generalizes the unordered search problem which has a $\Omega(\sqrt{2^n})$ lower bound.

Algorithm 2 MWDP(f, $[w_1, w_2, \ldots, w_j]$)

```
Require: $0 < w_1 < w_2 < \ldots < w_j < 2^n$
Global parameter: $\delta$ (error), $k$ (number of possible weights)
if $j == 1$, then
    return $w_k$.
else
    $m = \lfloor j/2 \rfloor$, $t = w_{m+1}/2^n$, $t' = w_m/2^n$, $\delta' = \delta/\log_2(k)$.
    $\mathcal{A}$ : quantum circuit for WDP (Figure 3) using $f(\,)$.
    /* Determine if $wt(f) \leqslant w_m$ or $\geqslant w_{m+1}$ */.
    if AS($\mathcal{A}$, $t$, $t'$, $\delta'$) accepts, then
        MWDP($f$, $[w_{m+1}, \ldots, w_j]$).
    else
        MWDP($f$, $[w_1, \ldots, w_m]$).
    end if
end if
```

A similar idea can be used to design an algorithm for the MWDP problem with k possible weights $\{0 < w_1 < w_2 < \ldots < w_k\}$. Our bounded-error algorithm for determining $wt(f)$ is described in Algorithm 2. The algorithm recursively searches for the correct weight in the list L that it maintains. In each recursive call, it uses AS to determine whether $wt(f)$ lies in the lower half of the weights in L or in the upper half and, accordingly, discards half of the possible weights from L. Specifically, if $wt(f) \leqslant w_m$, then $\mathcal{A}$'s probability of success is at most $w_m/2^n$, and otherwise, it is at least $w_{m+1}/2^n$; therefore, t and t' are set to $w_{m+1}/2^n$ and $w_m/2^n$, respectively. The algorithm makes an error if and only if any of the AS makes an error, and since there are $\log_2(k)$ such calls, the maximum error that Algorithm 2 can make is $\log_2(k) \cdot \delta' = \delta$.

The trivial classical complexity of the exact MWDP (without any error) with k possible weights is $O(2^n)$. The best-known quantum method for the exact MWDP was also proposed by Choi *et al.* [5]; the authors made $k - 1$ calls to the EWDP. Since the optimal query complexity of the EWDP is $\Theta(\sqrt{2^n})$, therefore, their approach yields a better-than-classical approach only when $k \ll \sqrt{2^n}$. Compared to those, our approach has a complexity $\tilde{O}(\sqrt{2^n} \log_2 k \log \frac{1}{\delta})$ that we explain next and suffers from a negligible probability of error δ—the dependency of the complexity on δ being logarithmic; it is possible to set a very low δ without a heavy increase in the complexity. Recall that the MWDP(f, $[w_1, w_2, \ldots, w_k]$) makes altogether $\log_2(k)$ calls to AS in a recursive manner. When AS is called with parameters $t' = w_m/2^n$ and $t' = w_{m+1}/2^n$, the number of calls to $f(\,)$ is at most $O(\frac{\sqrt{2^n}}{\sqrt{w_{m+1}-\sqrt{w_m}}} \log \frac{1}{\delta'}) = \tilde{O}(\sqrt{2^n} \log \frac{1}{\delta})$, leading us to the complexity stated before. In particular, when $k = \Theta(n)$, existing quantum algorithms have the same asymptotic complexity of $O(2^n)$ as classical algorithms but our approach uses only $O(n\sqrt{2^n})$ calls to $f(\,)$.

[2] The exact expression, along with all constants, turns out to be

$$\frac{1}{2}\frac{3\pi}{8\sin^{-1}\sqrt{t}}\left(1+7\left\lceil\frac{36}{1-\beta}\right\rceil\left(\ln\tfrac{1+s}{\delta}\right)^{1/3}\right)$$
$$\lessapprox \frac{3\pi}{16\sqrt{1-\rho_y}} + \frac{48\pi}{\sqrt{1-\rho_y}-\sqrt{\rho_n}}\left(\ln\tfrac{1+s}{\delta}\right)^{1/3}.$$

V. CONCLUSION

In this paper we have described a technique to reduce error in quantum algorithms in a black-box manner, akin to the classical approaches of running an algorithm multiple times. We showed how to use our approach for designing an efficient low-error algorithm for the multiple-weight decision problem. At the core of our approach is a new quantum algorithm for amplitude separation that can distinguish between two classes of inputs, say class G and class B, and given a quantum algorithm $\mathcal{A}$ that accepts an input from class G with a probability in the range $[p_2, 1]$ but accepts an input from class B with a probability in the range $[0, p_1]$ (where $p_1 < p_2$). It would be interesting and beneficial to solve its multiclass version, i.e., given possible ranges $[0, p_1], (p_1, p_2], \ldots, (p_k, 1]$ of acceptance probabilities of different classes of inputs to $\mathcal{A}$, identify between the classes, maybe with a low error. The number of times the algorithm $\mathcal{A}$ is repeated in our two-class amplitude separation algorithm is at most $\tilde{O}(\frac{1}{\sqrt{p_2}-\sqrt{p_1}})$, which is better than the classical methods based on Chernoff's bound or the quantum technique of amplitude estimation when $p_2 \ll 1$; we think that our bound is tight but we leave open the question of proving a lower bound on the required number of calls to $\mathcal{A}$.

ACKNOWLEDGMENTS

T.P.V. would like to thank Indraprastha Institute of Information Technology Delhi (IIIT-Delhi) for hosting him while this work was accomplished.

APPENDIX: PROOF OF TRIGONOMETRIC FACTS

We include a quick geometric proof of the trigonometric identity that, for any $a < 1$ and $t \leqslant \pi/2$,

$$\sin\theta \leqslant a \sin t \quad \text{implies that} \quad \theta \leqslant at.$$

For this, consider the right-angled triangles ABE and CDE in Fig. 4. E is the point where the line segment BD intersects the X axis and B and D are points on the $\sin(x)$ curve corresponding to $x = t$ and $x = at$, respectively.

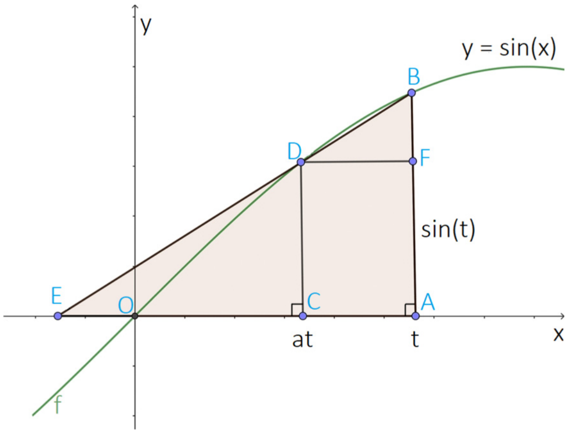

FIG. 4. Proof of the fact that $\sin\theta \leqslant a \sin t \Rightarrow \theta \leqslant at$.

We know from geometry that CDE is similar to ABE, that is, $\frac{\sin at}{\sin t} = \frac{CD}{AB} = \frac{EC}{EA}$. From the figure, $EC = EO + at$ and $EA = EO + t$, which implies that $\frac{\sin at}{\sin t} = \frac{EO+at}{EO+t} \geqslant a$. Therefore, $a \sin t \leqslant \sin at$. Furthermore, we are given that $\sin\theta \leqslant a \sin t$. Combining the latter two facts we get that $\sin\theta \leqslant \sin at$, which in turn implies that $\theta \leqslant at$, settling the fact.

In our analysis we make use of the fact that $\sin at \geqslant a \sin t$ for $a \in (0, 1)$ and $t \in [0, \pi/2]$, which follows from the above result.

We make use of another fact which states that $\sin at \leqslant \sqrt{a} \sin t$ for $t \in [0, \frac{\pi}{4}]$ and $a < 1$, whose proof we discuss now. Consider the real-valued continuous function $f(t) = a \sin^2 t - \sin^2 at$. We now show that $f(t)$ is nonnegative for $t \in [0, \frac{\pi}{4}]$. To show this, first observe that $f(0) = 0$. Furthermore, the first derivative satisfies $f'(t) = a(\sin 2t - \sin 2at) \geqslant 0$ since $a \in (0, 1)$ and $t \in [0, \frac{\pi}{4}]$. This shows that for the specified values of t, $f(t) \geqslant 0$ or, equivalently, $\sin at \leqslant \sqrt{a} \sin t$.

[1] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, Contemp. Math. **305**, 53 (2002).

[2] D. Bera, Amplitude amplification for operator identification and randomized classes, in *Computing and Combinatorics (COCOON)* (Springer International, Cham, Switzerland, 2018), pp. 579–591.

[3] D. Qiu and S. Zheng, Generalized Deutsch-Jozsa problem and the optimal quantum algorithm, Phys. Rev. A **97**, 062331 (2018).

[4] S. L. Braunstein, B.-S. Choi, S. Ghosh, and S. Maitra, Exact quantum algorithm to distinguish Boolean functions of different weights, J. Phys. A: Math. Theor. **40**, 8441 (2007).

[5] B.-S. Choi and S. L. Braunstein, Quantum algorithm for the asymmetric weight decision problem and its generalization to multiple weights, Quant. Info. Proc. **10**, 177 (2011).

[6] B.-S. Choi, Optimality proofs of quantum weight decision algorithms, Quant. Info. Proc. **11**, 123 (2012).

[7] K. Chakraborty and S. Maitra, Application of Grover's algorithm to check non-resiliency of a Boolean function, Cryptogr. Commun. **8**, 401 (2016).

[8] D. Bera, A different Deutsch–Jozsa, Quant. Info. Proc. **14**, 1777 (2015).