Portfolios of Quantum Algorithms

Sebastian M. Maurer

Physics Department, Stanford University, Stanford, California 94043

Tad Hogg and Bernardo A. Huberman

HP Labs, Palo Alto, California 94304

(Received 25 June 2001; published 27 November 2001)

Quantum computation holds promise for the solution of many intractable problems. However, since many quantum algorithms are stochastic in nature they can find the solution of hard problems only probabilistically. Thus the efficiency of the algorithms has to be characterized by both the expected time to completion and the associated variance. In order to minimize both the running time and its uncertainty, we show that portfolios of quantum algorithms analogous to those of finance can outperform single algorithms when applied to the NP-complete problems such as 3-satisfiability.

DOI: 10.1103/PhysRevLett.87.257901 PACS numbers: 03.67.Lx, 02.70.Rr, 89.65.Gh

While the quantum restarts involved in this method are similar to restart strategies for classical Las Vegas algorithms, there are two distinctions. First, the number of iterations \( t \) one performs before having a chance of finding a solution is chosen ahead of time. This is not so in the classical case, where the algorithm terminates as soon as a solution is found. Second, while the probability of finding a solution increases with time in the classical case, due to a monotonic cumulative probability distribution \( F(t) \) of finding a solution after \( t \) iterations, this is not so in the quantum case. For a quantum algorithm, the amplitudes after a trial with \( t \) iterations determine the probability of finding a solution upon a measurement. This probability is nonmonotonic and periodic [2].

Suppose there are \( N \) possible states overall, and \( S \) solutions to the search problem. After \( t \) iterations of amplitude amplification the probability of finding a solution when measuring the state is

\[
p_t = \sin^2[(2t + 1)\theta],
\]

where \( \theta \) satisfies \( \sin^2\theta = S/N \). As pointed out in [9], if the number of solutions and hence \( \theta \) is known, it is possible to determine the number of iterations \( t^* \) necessary before a measurement is done such that the result is a solution state with near absolute certainty. Simply put, we need

\[
(2t^* + 1)\theta = \frac{\pi}{2}
\]

so that

\[
t^* = \frac{\pi}{4} \sqrt{\frac{N}{S}} = 0.785 \sqrt{\frac{N}{S}}
\]

iterations are required in the limit of a small number of solutions.

Since the probability of finding a solution is already high for somewhat lower values of \( t \), a lower expected running time can be obtained by a restart strategy, i.e., repeatedly running the quantum algorithm with a smaller number of iterations [9]. In particular, with \( t \) iterations between every measurement, let \( \eta_t \) be the number of iterations required to find a solution, which in itself is a random variable due to...
to the variability in the number of trials. The expected number of iterations to find the solution is
\[
\langle \eta_i \rangle = \frac{t}{p_i} = \frac{t}{\sin^2[(2t + 1)\theta]}.
\] (3)

In the limit of small \( \theta = \sqrt{S/N} \), the optimal expected waiting time is [9]
\[
\langle \eta^* \rangle = \frac{z}{4\theta \sin^2(c/2)} = 0.690\sqrt{N/S}.
\] (4)

Boyer et al. [9] conclude that the restart strategy is indeed better, since the expected number of iterations \( \eta^* \) is about 12\% smaller. However, this result is misleading if variance is considered as well. Specifically, a smaller number of Grover iterations between measurements no longer seems as attractive, since an improvement in the expected performance is traded off against an increase in the variance in the running time.

Put another way, one obtains a small gain in the expected waiting time by increasing the probability that we will need at least twice as many iterations. Quantitatively,
\[
\langle \eta_i^2 \rangle = \sum_{k=1}^{\infty} (kt)^2 p_i (1 - p_i)^{k-1} = \frac{2 - 3p_i + p_i^2}{(1 - p_i)p_i^2} t^2.
\] (5)

Since \( \langle \eta_i \rangle = t/p_i \), the standard deviation in the number of iterations, \( \sigma \), is given by \( \frac{t}{p_i^2} \sqrt{1 - p_i} \). Figure 1 shows the “return vs risk” curve in this case. The efficient frontier contains all the points for which the mean cannot be reduced without increasing the variance, and the variance cannot be reduced without increasing the mean. Any point on the efficient frontier is an acceptable strategy, and the choice of a strategy depends on a person’s preferences. The optimum choice will depend on the application. For example, in real-time applications, reducing variance will be more important than reducing the expected waiting time. On the other hand, in other domains, we may care only about the mean performance because we average over a large number of trials. Another possibility, common in finance, is to maximize the ratio of return to risk, \( \langle \eta_i \rangle/\sigma \), called the Sharpe ratio [10]. For the quantum search algorithm presented here, the restart strategy has a Sharpe ratio near unity, while the strategy without restarts has a Sharpe ratio near infinity, because \( p_i \) is close to one.

This discussion also applies to any quantum algorithm whose performance varies with the run time. This is the case of the Grover algorithm when the number of solutions is not known \textit{a priori} [9] and quantum adiabatic algorithms of the type recently introduced by Farhi et al. [11].

Until now we considered variation in the number of iterations in each trial. Another approach applies to quantum algorithms with fixed trial lengths but with a variety of parameters that must be set prior to starting each trial. For a given problem instance, the choice of these parameters determines the probability for success of each trial. For example, instead of just adjusting phases based on whether states are solutions, such algorithms can use other efficiently computable properties, as often used in conventional heuristic methods. In general, the best parameter choices for a given instance are not known \textit{a priori}. Thus an important issue for such “quantum heuristics” is how to find choices that work well for typical problem instances encountered in practice.

As a specific example, we consider the \( k \)-satisfiability (\( k \)-SAT) problem. It consists of \( n \) Boolean variables and \( m \) clauses. A clause is a logical OR of \( k \) variables, each of which may be negated. A solution is an assignment, i.e., a value, true or false, for each variable, satisfying all the clauses. An assignment is said to conflict with any clause it does not satisfy. An example 2-SAT problem instance with three variables and two clauses is \( v_1 \lor \neg v_2 \) and \( v_2 \lor v_3 \), which has four solutions, e.g., \( v_1 = \text{false}, v_2 = \text{false}, \text{and} v_3 = \text{true} \). For \( k \geq 3 \), \( k \)-SAT is NP complete [12]. To consider typical rather than worst-case behavior, we focus on random 3-SAT, in which each clause is selected randomly, and take \( m = 4.25n \), which gives a high concentration of hard instances.

For a quantum algorithm [13], we use a superposition of all \( N = 2^n \) assignments, adjust phases based on the number of conflicts in each state, and mix amplitudes among states based on their Hamming distances. The algorithm’s performance depends on how well the phase adjustments match the structure of the particular instance, which is not known \textit{a priori}. One approach finds phase choices that work well on average for random 3-SAT problems [13], and gives better performance than amplitude amplification, which ignores problem structure.

However, phase adjustments that work well on average do not work well for all instances. Hence we can improve the performance by using a variety of choices, thereby reducing the chance of encountering instances that perform particularly poorly for any single choice of algorithm parameters. Such a “portfolio” of choices can reduce both the mean and variance of the time to find a solution.

In the simplest portfolio approach, we use different phase adjustments in each trial, instead of using the same ones every time. As we will show, this portfolio method is

![Figure 1](image-url)
guaranteed to perform better than a fixed strategy, provided we have no other information concerning the performance of different choices.

To quantify this improvement, consider a variety of phase choices for a given instance. Each choice gives a particular success probability \( p \) for a single trial. Let \( f(p) \) be the distribution of success probabilities when selecting phase choices from among a prespecified set of possibilities.

The default strategy simply picks a single choice for the phase adjustments to use for every trial. In this case, the expected running time is \( \langle \frac{1}{p} \rangle \), while the variance is \( \langle \frac{1}{p^2} \rangle - \langle \frac{1}{p} \rangle \).

Now assume that we randomly chose a different set of phases every time. Then the probability of success for a trial is simply \( \langle p \rangle \). It follows that the mean and standard deviation are given by \( \frac{1}{\langle p \rangle} \) and \( \frac{1}{\langle p \rangle \sqrt{1 - \langle p \rangle}} \), respectively.

Instead of using different choices for each trial, we can also use a “quantum portfolio” of our quantum algorithms. That is, all choices are evaluated simultaneously in superposition by using additional qubits to specify the particular phase choice.

For simplicity, consider such a portfolio of only two algorithms, i.e., choices for phases. A single qubit \( |a\rangle = \alpha |+\rangle + \beta |-\rangle \) will determine which of the two algorithms is to be used.

Suppose the first choice gives amplitudes \( c_i \) for state \( i \) and the second gives \( d_i \), when run individually. Then the result for the quantum portfolio is

\[
\alpha \sum_i c_i |i, +\rangle + \beta \sum_i d_i |i, -\rangle.
\]

The probability of measuring a solution state is

\[
p = |\alpha|^2 \sum_{i \in A} |c_i|^2 + |\beta|^2 \sum_{i \in A} |d_i|^2.
\]

This probability is just a weighted sum of the individual success probabilities of each algorithm on its own. This discussion generalizes to combinations of additional phase choices. As a result, quantum portfolios of quantum algorithms, and mixed quantum algorithms (i.e., a classical portfolio), in which we randomly use a different quantum algorithm after each iteration, are equivalent, since they both result in the same probability of success at every measurement.

The interesting result is that such portfolios, i.e., both classical and quantum, complete faster on average than a computation with a fixed set of parameters. This is because

\[
\langle \frac{1}{p} \rangle \geq \frac{1}{\langle p \rangle}
\]

always holds, since \( p \) is strictly positive. Equality occurs only for a trivial probability distribution with only one outcome. This can easily be shown using the Schwarz inequality [14]:

\[
(AB)^2 \leq \langle A^2 \rangle \langle B^2 \rangle.
\]

for which equality holds only if there is a linear combination \( aA + bB \) that is equal to zero with unit probability.

The performance improvement obtained with such a strategy is shown in Fig. 2, in which we randomly chose a different set of phases to use for each quantum computation between measurements. The histogram shows a substantial probability of choosing sets of phases with near zero probability of finding a solution after each measurement. This leads to a large expected waiting time if a portfolio is not used.

In practice, one would not use entirely random phases since their typical performance is rather low. Rather, one would use phases that are already known to work well for similar problems, e.g., obtained by optimizing the choices for a sample of similar problem instances. Optimizing choices for these sample instances multiple times, starting from a variety of initial choices, gives a set of phase choices that can be expected to perform much better than random choices on new instances drawn from the same problem ensemble as the training sample.

Since such optimization can be computationally demanding, a more feasible scenario performs this optimization on a large sample of small problems and then applies the resulting choices to new, larger instances. Thus algorithms are optimized on easily solvable instances, and applied to more difficult problems that have not been solved before.

For random \( k \)-SAT, the ratio of clauses to variables, \( m/n \), characterizes the concentration of difficult instances so a reasonable scaling approach optimizes phases for small problems and then applies those to larger problems with the same ratio. Figure 3 shows the performance improvement of such a strategy. Here the set of phase choices available to select was created by optimizing for instances of 3-SAT with 8 and 12 variables and clause to variable ratio, \( m/n \), of 4.25. These optimized phases were used to solve larger instances, with 20 variables. In this case, the advantages of a portfolio strategy are less dramatic than in Fig. 2, but are still quite impressive.
Thus the advantages of classical portfolios can be improved on even further by a truly quantum portfolio. More generally, as with combining different conventional heuristics, we could consider operators within each trial that mix amplitude among the different quantum algorithms. Such operations provide a broader range of possible techniques than available with classical portfolios, though it remains to be seen whether such extensions give significant additional improvements.

S.M.M. thanks the Fannie and John Hertz Foundation for financial support.