JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 116, No. 3, pp. 517–529, March 2003 (© 2003)

Implementing Pure Adaptive Search with Grover's Quantum Algorithm[1]

D. Bulger,[2] W. P. Baritompa,[3] and G. R. Wood[4]

Communicated by P. M. Pardalos

Abstract. Pure adaptive search (PAS) is an idealized stochastic algorithm for unconstrained global optimization. The number of PAS iterations required to solve a problem increases only linearly in the domain dimension. However, each iteration requires the generation of a random domain point uniformly distributed in the current improving region. If no regularity conditions are known to hold for the objective function, then this task requires a number of classical function evaluations varying inversely with the proportion of the domain constituted by the improving region, entirely counteracting the PAS apparent speedup. The Grover quantum computational search algorithm provides a way to generate the PAS iterates. We show that the resulting implementation, which we call the Grover adaptive search (GAS), realizes PAS for functions satisfying certain conditions, and we believe that, when quantum computers will be available, GAS will be a practical algorithm.

Key Words. Discrete optimization, global optimization, Grover iterations, Markov chains, quantum computers, random search.

1. Introduction

Pure adaptive search (PAS) is a theoretical stochastic global optimization algorithm. If PAS could be implemented efficiently, then under certain

[1] The authors would like to thank the Marsden Fund administered by the Royal Society of New Zealand for support of this research.

[2] Postdoctoral Fellow, Institute of Information Sciences and Technology, Massey University, Wellington, New Zealand.

[3] Senior Lecturer, Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand.

[4] Professor, Institute of Information Sciences and Technology, Massey University, Palmerston North, New Zealand.

conditions the number of PAS iterations required to solve a problem would increase only linearly in the dimension of the domain (Refs. 1–2). The algorithm specifies that each iteration requires the generation of a random domain point uniformly distributed in the current improving region, that is, the subset of the domain on which the objective function takes better values than any yet seen. If no regularity conditions are known to hold for the objective function, then the generation of this random point in the improving region requires a number of classical function evaluations varying inversely with the proportion of the domain constituted by the improving region, annulling completely the PAS apparent exponential efficiency gain. However, the Grover quantum computational search algorithm provides a way to generate the PAS iterates. We show that the Grover adaptive research (GAS) realizes PAS for functions satisfying certain conditions, and we believe that, when quantum computers will become available, GAS will be a practical algorithm.

This article merges known results from the theories of global optimization and quantum computation. Section 2 describes the pure adaptive search from the literature of stochastic global optimization theory. Section 3 briefly introduces quantum computation in general, and Section 4 describes the Grover search method, an algorithm for execution on quantum computers. What is known about the durations required for these two algorithms is then combined in Section 5 to explore the future feasibility of implementing PAS on quantum computers using the Grover method. We term the combined algorithm the Grover adaptive search (GAS).

1.1. Terminology and Notation. We consider the following finite global optimization problem:

$$\begin{array}{ll} \min & f(x), \\ \text{s.t.} & x \in S, \end{array}$$

where $f(x)$ is a real-valued function on a finite set S.

The following algorithms for finite optimization are considered in this paper:

(i) Pure random search (PRS, Ref. 3) samples the domain at each iteration according to a fixed distribution.

(ii) Pure adaptive search (PAS, called Strong PAS in Ref. 2) samples from that part of the domain giving a strictly improving objective function value at each iteration.

(iii) Hesitant adaptive search (HAS, Ref. 4) behaves like PAS, except that each iteration is only successful with a certain probability; otherwise, the iteration passes with no improvement found. The success probability in HAS is a function of the best objective function value yet at hand.

(iv) Grover adaptive search (GAS), introduced here, is a stochastic algorithm using at each step a Grover quantum search (Ref. 5) to find a point with a strictly improving objective function value. This is a direct application of the Grover quantum search, where the "marked" points are those with function values better than the best value seen up to that iteration.

Throughout this paper, we associate with the objective function f the following definitions. Let N denote the size of the domain S. Let $l_1 < \cdots < l_K$ be the distinct objective function values, that is, the distinct levels appearing on the graph of f. Notice that there may be more than K points in S; in fact, the most interesting cases are when $K \ll N$. Given the uniform probability measure μ on S, we define a probability measure $\pi = (\pi_1, \ldots, \pi_K)$ on the range of f as follows. Let π_j be the probability that any iteration of pure random search attains a value of l_j. That is,

$$\pi_j = \mu(f^{-1}(l_j)), \quad \text{for } j = 1, 2, \ldots, K,$$

so more common range values are more likely under π. Let p_j denote $\sum_{i=1}^{j} \pi_i$, the probability that PRS attains a value of l_j or less. In particular, $p_K = 1$.

2. Pure Adaptive Search

The pure adaptive search algorithm, introduced in Ref. 6, is a stochastic algorithm for unconstrained global optimisation which has a very favorable convergence rate, but which has been thought impossible to implement efficiently. The algorithm states simply that each sample point should be distributed uniformly in the current improving region, that is, in the set of points yielding objective function values better than any yet seen. We will consider only PAS in the case of a finite domain and a uniform initial sampling distribution. Formally, we have the following algorithm.

Algorithm 2.1. Pure Adaptive Search.

```
Step 1.  Generate a PRS iterate $x_1 \in S$, and set $y_1 = f(x_1)$.
Step 2.  For $i = 1, 2, \ldots$, as long as $y_i \neq l_1$, do the operations below:

             define $S_i = \{x \in S : f(x) < y_i\}$;
             randomly generate $x'$ uniformly from $S_i$;
             set $y' = f(x')$;
             set $x_{i+1} = x'$ and $y_{i+1} = y'$.
```

The hesitant adaptive search algorithm (Ref. 4) is a simple generalization of PAS. Define a success or bettering probability function $b: R \to (0, 1]$. The algorithm is as follows.

Algorithm 2.2. Hesitant Adaptive Search.

```
Step 1.  Generate a PRS iterate $x_1 \in S$, and set $y_1 = f(x_1)$.
Step 2.  For $i = 1, 2, \ldots$, as long as $y_i \neq l_1$, do the operations below:

             define $S_i = \{x \in S: f(x) < y_i\}$;
             with probability $b(y_i)$, randomly generate $x$ uniformly from $S_i$
             and set $y = f(x) x_{i+1} = x, y_{i+1} = y$;
             otherwise, set $x_{i+1} = x_i$ and $y_{i+1} = y_i$.
```

Pure adaptive search (PAS) is just HAS with b equal to the constant function 1. In Ref. 2, the expected number of iterations before termination of PAS has been shown to equal $\sum_{j=2}^{K} \pi_j / p_j$, and thus to be bounded above by $log(1/p_1)$. This convergence rate of PAS is very appealing. It equates to a linear dependence on domain dimension (see for instance Refs. 1–2), whereas typically the effort required to solve global optimization problems using known practical methods increases exponentially with the domain dimension.

The flaw preventing the practical use of PAS can be seen in the step requiring the generation of the random point x. If the objective function is unknown (as with a problem of the black-box type), or unless it yields readily to analytical study, the improving region may be unknown at each iterate. In addition, even the task of generating uniformly distributed points in known regions can present computational difficulties. Attempts to implement PAS, either exactly or approximately, have concentrated on methods for producing points distributed approximately uniformly in the unknown improving region: for instance, the improving hit-and-run algorithm in Ref. 7 uses the first sample from a Markov chain whose limiting distribution is known to be the required uniform distribution on the improving region.

Recently, physicists studying quantum computing have provided a method that opens up the possibility of an efficient implementation of PAS in the future. A broad overview of this method will be provided in Sections 3 and 4. Some readers may wish to skip directly to the summary at the end of Section 4, where this overview closes and the pertinent quantum result is given.

3. Quantum Computation

The Grover mechanism referred to in this article is one of the major advances to date in the fledgling field of quantum computation, the study of the workings of quantum computers. Quantum computers are predicted technology; research into the hardware implementation of these ideas is still in the very early experimental stages (Ref. 8). But the continuing miniaturization of electronics, combined with our growing awareness of the possible benefits of quantum computation over conventional computation, suggest that these devices are likely to appear within a few decades.

The characteristic feature of a quantum computer is that, in place of the conventional computer bits, it uses quantum bits or qubits. A qubit is a quantum system with two eigenstates, denoted $|0\rangle$ and $|1\rangle$. Dirac notation is used in this and the following section to describe quantum states, as is standard in quantum mechanics. If the possible states of a quantum system are situated in a Hilbert space H, then the general element of H is denoted by $|u\rangle$. Whereas the state of an ordinary bit can be either 0 or 1, the state of a qubit can be any combination

$$\alpha|0\rangle + \beta|1\rangle,$$

where α and β are complex numbers with

$$|\alpha|^2 + |\beta|^2 = 1.$$

When the value stored in a qubit in this state is measured, an output of either 0 or 1 is observed: 0 is observed with probability $|\alpha|^2$, and 1 is observed with probability $|\beta|^2$.

That a qubit can be in a simultaneous superposition of off and on eigenstates presents the possibility of what has been called quantum parallelism, where a single quantum circuit can simultaneously perform a calculation on a superimposed input, corresponding to very many conventional inputs.

The study of quantum computation received a marked increase in attention in 1994 with the publication of the Shor factoring algorithm (Ref. 9). This and other developments of the theory have indicated that quantum computers may one day enjoy a huge speed advantage over conventional computers for many types of problems.

4. Grover Search

The quantum procedure germane to our purposes is the Grover search. This section introduces it and defines the Grover probability distribution (1).

Consider the following general search problem. Let n be a positive integer, and let $S=\{0,1\}^n$, so that the domain size $N=2^n$. Let $g: S\to\{0,1\}$, and assume that we have a black-box quantum circuit implementing g, called the "oracle." We wish to find a point $u\in S$ such that $g(u)=1$. We call such points "marked," we denote the set of marked points by $M\subseteq S$, and we denote the number of marked states by $m=|M|$. We may or may not be aware of the value of m.

Grover introduced in Ref. 5 a means of implementing a certain phase-space rotation of the state of a quantum system encoding points in the domain S. This rotation can be used to move toward the unknown marked states. The description here draws heavily from Ref. 10.

For any real numbers k and l such that

$$mk^2+(N-m)l^2=1,$$

define

$$|\Psi(k,l)\rangle=\sum_{u\in M}k|u\rangle+\sum_{u\in S\backslash M}l|u\rangle.$$

The pivotal step of the Grover algorithm is an efficient method for transforming the quantum state $|\Psi(k,l)\rangle$ into the state

$$\left|\Psi\left(\frac{N-2m}{N}k+\frac{2N-2m}{N}l,\frac{-2m}{N}k+\frac{N-2m}{N}l\right)\right\rangle,$$

using a single (superimposed) oracle query.

As explained in Refs. 5 and 10, this is a rotation in the plane spanned by the vectors

$$(1/\sqrt{m})\sum_{u\in M}|u\rangle \quad\text{and}\quad (1/\sqrt{N-m})\sum_{u\in S\backslash M}|u\rangle,$$

by an angle determined by m and N. Note that the equal amplitude state

$$(1/\sqrt{N})\sum_{\mathrm{u\in S}}|u\rangle,$$

which is relatively simple to prepare within a quantum computer, lies in this plane. When the equal amplitude state is transformed by the Grover rotation operator r times in succession, we call it a Grover run of r rotations, and the resulting quantum state is

$$[\sin((2r+1)\theta)/\sqrt{N}\sin\theta]\sum_{u\in M}|u\rangle+[\cos((2r+1)\theta)/\sqrt{N}\cos\theta]\sum_{u\in S\backslash M}|u\rangle,$$

where $\theta\in[0,\pi/2]$ is such that $\sin^2\theta=m/N$. If this state is then observed, it will collapse to each eigenstate (element of S) with probability equal to the squared modulus of that eigenstate amplitude in the quantum state.

4.1. Grover Run and Distribution. Thus, if S is a finite set with a marked subset $M \subset S$ with $|M| = m$, if we can construct a quantum circuit (the oracle) to test individual domain points for membership of M, if $\theta \in [0, \pi/2]$ is such that $\sin^2\theta = m/N$, and if $r \in N$, then we can execute a Grover run of r rotations for a cost in proportion to that of $r + 1$ oracle queries, in order to generate domain points according to the probability distribution γon S with

$$\gamma(\{x\}) = \begin{cases} \sin^2(2r+1)\theta/N\sin^2\theta, & x \in M, \\ \cos^2(2r+1)\theta/N\cos^2\theta, & x \in S \setminus M. \end{cases} \tag{1}$$

We define $\mathscr{G}(S, M, r)$ the Grover distribution on S with parameters (M, r) to equal this distribution.

If $X \sim \mathscr{G}(S, M, r)$, then

$$\begin{aligned} \mathrm{P}[X \in M] &= \sum_{u \in M} \mathrm{P}[X = m] \\ &= m\sin^2(2r+1)\theta/N\sin^2\theta \\ &= \sin^2(2r+1)\theta. \end{aligned}$$

We define a new function g_r and summarize this section.

Axiom 4.1. There is a search procedure on a quantum computer, called a Grover run of r rotations, that will find a marked state with probability

$$g_r(p) = \sin^2[(2r+1)\arcsin\sqrt{p}]$$

when the proportion of marked states is p.

The general form of g_r is illustrated in Fig. 1; the polynomial shown is g_5 of degree 11. The number of maxima is equal to $r + 1$.

5. Grover Adaptive Search

The Grover mechanism described in the previous section is the key opening up the possibility of efficient general implementation of PAS (that is, implementation for use in the absence of known regularity conditions). In this section, we define an algorithm using the Grover distribution and provide the main result of this paper.

Let r be a fixed positive integer, called the rotation count. Then, the Grover adaptive search with r rotations per step [GAS(r)] is as follows.

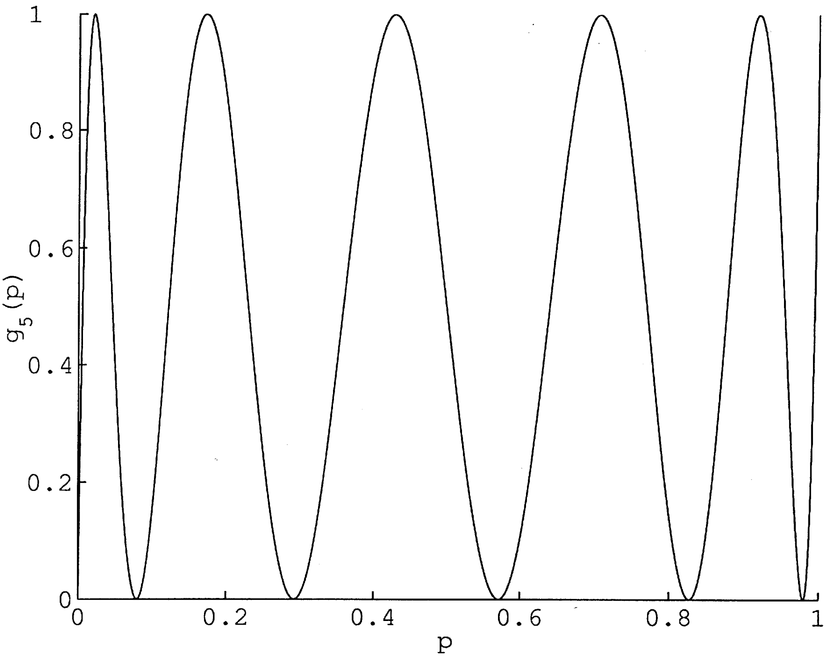

Fig. 1. Maximizers of g_5 are projections of points evenly spaced on a circle.

Algorithm 5.1. Grover Adaptive Search.

```
Step 1. Generate a PRS iterate $x_1 \in S$ and set $y_1 = f(x_1)$.
Step 2. For $i = 1, 2, \ldots$, as long as $y_i \neq l_1$, do the following operations:
        define $S_i = \{x \in S: f(x) < y_i\}$;
        set $x$ to the observed output of a Grover run of $r$ rotations,
        starting from the equal amplitude vector, with the states in $S_i$
        considered "marked";
        do a classical evaluation to determine $y = f(x)$;
        if $y < y_i$, then set $x_{i+1} = x$ and $y_{i+1} = y$;
        otherwise, if $y \geq y_i$, set $x_{i+1} = x_i$ and $y_{i+1} = y_i$.
```

In the algorithm as presented, the number r of Grover rotations used in each step is constant. Variations of this algorithm are possible, in which the number of rotations per step varies, either according to a fixed schedule or adaptively.

5.1. Markov Chain Description. The stochastic process $\{Y_k: k = 1, 2, \ldots\}$ for GAS can be modeled as a Markov chain with states $l_1, \ldots, l_K$, where the state l_1 represents the global optimum. The initial probability distribution Y_1 is given by π (i.e., we begin with a PRS step). In the standard Markov chain terminology (Refs. 2 and 11), l_1 is the absorbing state of this chain and all other states are transient. GAS converges when the chain reaches the absorbing state.

A GAS iteration will always return a value less than or equal to the best so far. If the best so far is l_{m+1}, the Grover run of r rotations succeeds with probability $g_r(p_m)$. In that event, all domain points in the improving region are equally likely, so the K by K transition matrix P, in standard form, having the ijth element $\mathrm{P}[Y_k = l_j | Y_{k-1} = l_i]$, for GAS is as follows. Note that the (2, 1)-entry is equal to $g_r(p_1)$, but is written as $g_r(p_1)\pi_1/p_1$ to clarify the structure of the matrix,

$$\begin{bmatrix} 1 & 0 & 0 & \ldots & 0 & 0 \\ g_r(p_1)\dfrac{\pi_1}{p_1} & 1-g_r(p_1) & 0 & \ldots & 0 & 0 \\ g_r(p_2)\dfrac{\pi_1}{p_1} & g_r(p_2)\dfrac{\pi_2}{p_2} & 1-g_r(p_2) & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ g_r(p_{K-1})\dfrac{\pi_1}{p_{K-1}} & g_r(p_{K-1})\dfrac{\pi_2}{p_{K-1}} & g_r(p_{K-1})\dfrac{\pi_3}{p_{K-1}} & \ldots & g_r(p_{K-1})\dfrac{\pi_{K-1}}{p_{K-1}} & 1-g_r(p_{K-1}) \end{bmatrix}.$$

It follows that $\mathrm{P}[Y_k = l_i]$, the probability of objective function value l_i for GAS on the kth iteration, is the ith entry of πP^{k-1}.

Note that, for the variation of GAS with a schedule of different rotation counts, the corresponding stochastic process would be modelled by a nonhomogeneous Markov chain.

5.2. Realization of HAS. Hitherto, HAS and PAS have been viewed primarily as theoretical tools, but quantum computing casts these algorithms in a new light.

Theorem 5.1. GAS is an implementation of HAS. The number of GAS(r) iterations before termination has expectation

$$\sum_{j=2}^{K} \frac{\pi_j}{g_r(p_j)p_j}$$

and variance

$$\sum_{j=2}^{K} \left(\frac{2}{g_r(p_j)} \frac{\pi_j}{g_r(p_j)\,p_j} - 1\right) \frac{\pi_j}{g_r(p_j)\,p_j}.$$

Proof. Using any improvement probability function b satisfying

$$b(l_i) = g_r(p_i), \qquad \text{for each } i \in \{1, \dots, K\},$$

reveals GAS as an implementation of HAS. The moment results follow from Corollary 3.2 in Ref. 12. □

Note that all the higher moments are also available (Ref. 12).

5.3. Realization of PAS. The exact expressions given in the theorem are valid for an arbitrary distribution reflected by π. However, if the distribution is particularly well behaved, GAS becomes PAS.

Corollary 5.1. If a function f has associated p_j satisfying $g_r(p_j) = 1$, then GAS(r) realizes PAS for this f.

Proof. The matrix P reduces to the transition matrix for strong PAS in Ref. 2, since

$$g_r(p_j) = 1, \qquad \text{for all } j = 1, \dots, K. \qquad \square$$

Example 5.1. As an example, consider the univariate function depicted in Fig. 2. This is a function on a domain of size 2^{13}, whose image

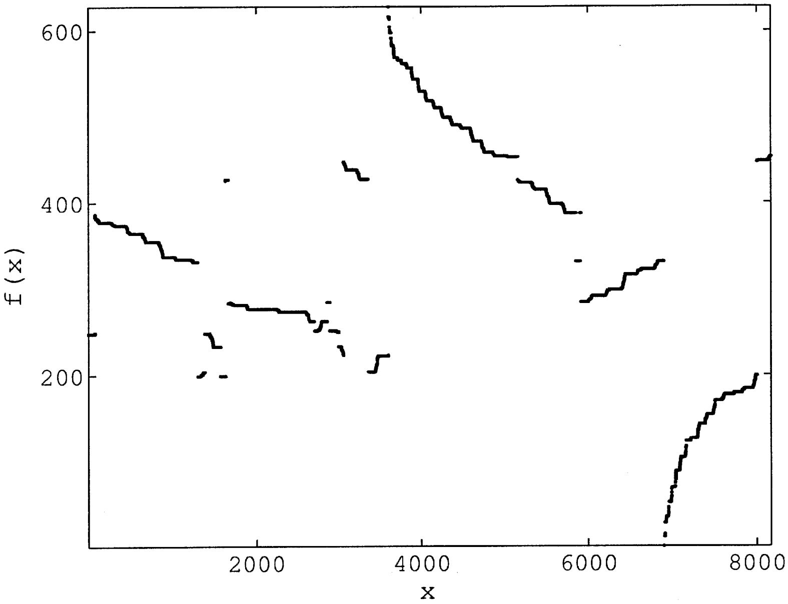

Fig. 2. Computer-generated example of objective function.

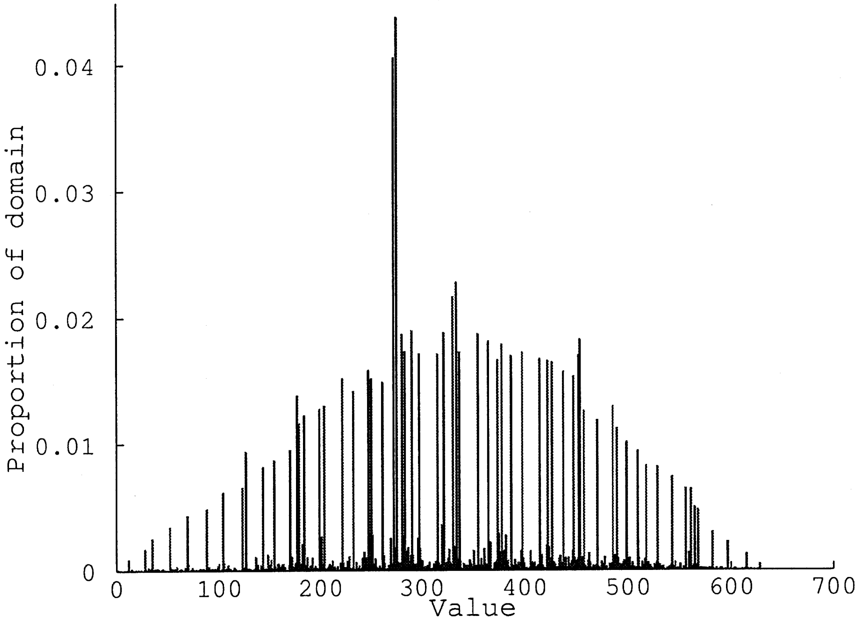

Fig. 3. Relative frequency distribution of values taken by the objective function in Fig. 2.

consists of 624 distinct values. The relative frequency distribution of these values is shown in Fig. 3.

When applied to this function, the number of iterations required by GAS(65) before termination has expectation 8.9 and standard deviation 3.0. For the same function, the number of PRS iterations before termination has expectation 6736 and standard deviation 8495.

To compare these results fairly, recall that each iteration performed by GAS(65) uses 65 Grover rotations, each requiring a single oracle query, followed by a classical evaluation, that is, 66 function evaluations. Thus, GAS(65) has an expected effort comparable to 587.1 PRS iterations, with a standard deviation comparable to 198.5 PRS iterations.

Note that the behavior of GAS on this function depends only on the distribution given in Fig. 3. We generated the example by first building a monotone function with the distribution shown in Fig. 3 and then shuffling to produce the function shown in Fig. 2. This example was made using some very preliminary results aimed at identifying the characteristics of functions for which GAS(r) will be particularly suited. A more thorough study is planned for future work. Briefly, though, we arbitrarily chose $r = 65$ and constructed a distribution for which g_r gave values of at least 0.8 at all points. As expected, this example required very few iterations of GAS(65), and it is encouraging to see that this advantage remains pronounced when total effort is considered.

Of course, each particular value of r is ideally suited only to a small class of functions, namely, those functions for whose distributions g_r gives only values close to 1. To others, it can be poorly suited. In some sense, it is good on average, but it is not robust, in that the variation in performance is very great. It is expected, though, that variable rotation count versions of GAS will bring similar efficiency gains to objective functions without special suitability properties.

6. Discussion

Throughout this article, it has been assumed that the domain cardinality is a power of two. Where this is not the case, note that the domain can be artifically couched within a larger domain whose cardinality is a power of two. If the size of the domain is known, the artificial domain need not be larger than twice the size of the true domain. An objective function can be extended to have maximal value on the artifically added points, and similarly a marking oracle can merely ignore (not mark) the artificial points. The values of π_i will change by this construction.

The study of stochastic algorithms has often been seen as less worthwhile than that of deterministic ones. In the future, with quantum machines intrinsically based on physics showing stochastic behavior, it would appear that algorithms such as PAS and HAS will come to the fore.

Further Work. As mentioned earlier, it might be expected that efficiency gains could be made by allowing the rotation count to vary from one GAS step to another, either according to a fixed schedule or dynamically, depending on which Grover runs yielded improvements. A future paper explores this and the effect of domain expansion as mentioned above.

A future paper will investigate convergence time bounds for GAS which apply under certain favorable conditions.

7. Summary

Physicists studying quantum computing have provided a method that opens up the future possibility of an efficient global optimization algorithm. The Grover adaptive search (GAS) realizes hesitant adaptive search (HAS) and on occasions pure adaptive search (PAS), which up to now have been thought of primarily as theoretical tools.

References

1. Zabinsky, Z. B., and Smith, R. L., *Pure Adaptive Search in Global Optimization*, Mathematical Programming, Vol. 53, pp. 323–338, 1992.
2. Zabinsky, Z. B., Wood, G. R., Steel, M. A., and Baritompa, W. P., *Pure Adaptive Search for Finite Global Optimisation*, Mathematical Programming, Vol. 69, pp. 443–448, 1995.
3. Brooks, S. H., *A Discussion of Random Methods for Seeking Maxima*, Operations Research, Vol. 6, pp. 244–251, 1958.
4. Bulger, D. W., and Wood, G. R., *Hesitant Adaptive Search for Global Optimization*, Mathematical Programming, Vol. 81, pp. 89–102, 1998.
5. Grover, L. K., *A Fast Quantum Mechanical Algorithm for Database Search*, Proceedings of the 28th Annual ACM Symposium on the Theory of Computing, 1996.
6. Patel, N. R., Smith, R. L., and Zabinsky, Z. B., *Pure Adaptive Search in Monte Carlo Optimization*, Mathematical Programming, Vol. 43, pp. 317–328, 1988.
7. Zabinsky, Z. B., Smith, R. L., Mcdonald, J. F., Romeijn, H. E., and Kaufman, D. E., *Improving Hit-and-Run for Global Optimization*, Journal of Global Optimization, Vol. 3, pp. 171–192, 1993.
8. Laflamme, R., *Los Alamos Scientists Make Seven Bit Quantum Leap*, available at http://www.lanl.gov/worldview/news/releases/archive/00-041.html, 2000.
9. Shor, P. W., *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*, SIAM Journal on Computing, Vol. 26, pp. 1484–1509, 1997.
10. Boyer, M., Brassard, G., Høyer, P., and Tapp, A., *Tight Bounds on Quantum Searching*, Fortschritte der Physik, Vol. 46, pp. 493–506, 1998.
11. Kemeny, J. G., and Snell, J. L., *Finite Markov Chains*, Springer Verlag, New York, NY, 1960.
12. Wood, G. R., Zabinsky, Z. B., and Kristinsdottir, B.P., *Hesitant Adaptive Search: The Distribution of the Number of Iterations to Convergence*, Mathematical Programming, Vol. 89, pp. 479–486, 2001.