A Quantum Search Algorithm for a Specified Number of Targets

Mark A. Rubin
Lincoln Laboratory
Massachusetts Institute of Technology
244 Wood Street
Lexington, Massachusetts 02420-9185
rubin@ll.mit.edu

Abstract

The quantum search algorithm of Chen and Diao, which finds with certainty a single target item in an unsorted database, is modified so as to be capable of searching for an arbitrary specified number of target items. If the number of targets, \( \nu_0 \), is a power of four, the new algorithm will with certainty find one of the targets in a database of \( N \) items using

\[
\frac{1}{2} \left( \frac{3(N/\nu_0)^{\log_4 3} - 1}{3} \right) \approx \frac{1}{2} \left( \frac{3(N/\nu_0)^{0.7925} - 1}{3} \right)
\]

oracle calls, where \( N \) is the smallest power of four greater than or equal to \( N \). If \( \nu_0 \) is not a power of four, the algorithm will, with a probability of at least one-half, find one of the targets using no more than

\[
\frac{1}{2} \left( \frac{9(N/\nu)^{\log_4 3} - 1}{3} \right)
\]

calls, where \( \nu \) is the smallest power of four greater than or equal to \( \nu_0 \).

1 Introduction

Recently Chen and Diao [1] presented a quantum algorithm for searching an unsorted database capable of finding, with certainty, a single target item in an \( \mathcal{N} \)-item database after

\[ 2 \lceil \log_4 \mathcal{N} \rceil \]

iterations of certain unitary operations. (\( \lceil x \rceil \) denotes the smallest integer greater than or equal to \( x \).) Grassl [2] and Tu and Long [3] have given a recursive implementation of these unitary operations, and have pointed out that, with this implementation, the number of oracle calls required for the \( j \)th iteration increases exponentially with \( j \).

In this paper I present a modification of the algorithm of [1] for searching an unsorted database of \( \mathcal{N} \) items for \( \nu_0 \geq 1 \) target items, provided that the number of targets \( \nu_0 \) is

\[ \nu_0 \approx 4^k \]
known in advance. In Section 2 below I discuss the case of $\nu_0$ equal to a power of four; in this case the algorithm will find one of the target items with unit probability. In Section 3 I discuss the case of $\nu_0$ not equal to a power of four; in this case the algorithm will find one of the target items with probability of at least one-half. The number of oracle calls required using the recursive implementation is given in Section 4. The notation and terminology follow, in general, those of \[1\] and \[2\].

2 Number of Targets a Power of Four

Denote the $N$ items in the database $D$ by $w_i, i = 1, \ldots, N$. Of these items, a total of $\nu_0$ are members of the subset $T$ of target items. An oracle function $f(w_i)$ indicates whether a selected item is or is not a target:

$$f(w_i) = 1, \quad w_i \in T,$$

$$= 0, \quad \text{otherwise.}$$ (1)

If $N$ is not already a power of four, we embed the database $D$ in a larger database $\tilde{D}$ containing additional non-target items such that the total number of items in $D$ is the smallest power of four larger than $N$:

$$D = D \cup \{w_{N+1}, \ldots, w_N\},$$ (2)

where

$$N = 2^{2n},$$ (3)

$n$ an integer, i.e.,

$$n = [\log_4 N],$$ (4)

so

$$N > N > N/4.$$ (5)

The above enlargement of the database is as in \[1\]. Here, in addition, we embed $D$ in a database $\tilde{D}$ which is four times larger still:

$$\tilde{D} = D \cup \{w_{N+1}, w_{N+2}, \ldots, w_{\tilde{N}}\},$$ (6)

where

$$\tilde{N} = 4N = 2^{2\tilde{n}}.$$ (7)

That is,

$$\tilde{n} = n + 1.$$ (8)

All of the additional items not in $D$ are by definition non-targets, so equation (1) still holds and the cardinality of $T$ is still $\nu_0$.

For the database to be searched by a quantum computer \[4\], the $\tilde{N}$ items in $\tilde{D}$ are set in one-to-one correspondence with the $\tilde{N}$ computational-basis states $|a_1a_2\ldots a_{2\tilde{n}}\rangle$:

$$w_i \leftrightarrow |a_1(i)a_2(i)\ldots a_{2\tilde{n}}(i)\rangle, \quad i = 1, \ldots, \tilde{N}.$$ (9)
where each of the eigenvalues $a_j(i)$ is either 0 or 1. The $2\tilde{n}$-component vector of $a_j$’s associated with $w_i$ is termed the symbol of $w_i$:

$$S(w_i) = a_1(i)a_2(i) \ldots a_{2\tilde{n}}(i).$$

We also define auxiliary symbol functions

$$S_j(w_i) = a_1(i) \ldots a_j(i), \quad j = 1, \ldots, 2\tilde{n},$$
$$S_{2\tilde{n}}(w_i) = S(w_i).$$

It should be emphasized that the correspondence (9) is not chosen to make the symbol $S(w_i)$ a binary representation of the item index $i$. On the contrary, it is essential for what follows that none of the $N$ items in the set $D$ be represented by states such that $S_2(w_i) = 00$. That is, we require that

$$w_i \in D \Rightarrow S_2(w_i) \neq 00.$$  \hspace{1cm} (12)

(We could, for example, establish the correspondence (9) so that $w_i \in D \Rightarrow S_2(w_i) = 11$.) Condition (12) implies

$$w_i \in T \Rightarrow S_2(w_i) \neq 00.$$  \hspace{1cm} (13)

Extending the technique employed in [1] to the case of multiple targets, we select $\nu_0$ of the items with auxiliary symbols $S_2(w_i) = 00$ to be “ground state items.” Specifically, the $\nu_0$ elements of the set $G$ of ground state items,

$$G = \{w_{G1}, w_{G2}, \ldots, w_{G_{\nu_0}}\},$$

are those with the symbols

$$S(w_{G1}) = 00 \ldots 000000\ldots$$
$$S(w_{G2}) = 00 \ldots 000001\ldots$$
$$S(w_{G3}) = 00 \ldots 000010\ldots$$
$$S(w_{G4}) = 00 \ldots 000011\ldots$$
$$\vdots$$

The rightmost $2p$ entries in $S(w_{G_{\nu_0}})$ are all 1’s and constitute a binary representation of $\nu_0 - 1$, where

$$2^{2p} = \nu_0.$$  \hspace{1cm} (16)

We can now define the auxiliary functions

$$f_j(w_i) = 1 \quad \text{if } S_{2j} = 00 \ldots 00 \text{ but } w_i \not\in G,$$
$$= 0 \quad \text{otherwise}; \quad j = 1, \ldots, \tilde{n} - p,$$

and, in terms of these, the auxiliary oracle functions

$$F_j(w_i) = f(w_i) \lor f_j(w_i).$$

(The symbol “$\lor$” denotes logical OR.) Note that

$$F_{\tilde{n} - p}(w_i) = f(w_i).$$

(3)
The starting state for the iteration is the equally-weighted superposition of computational basis states obtained from the state $|w_{G_1}\rangle = |00\ldots00\rangle$ by a Walsh-Hadamard transformation,

$$|s_0\rangle = \frac{1}{\sqrt{\tilde{N}}} \sum_{i=1}^{\tilde{N}} |w_i\rangle. \quad (20)$$

Starting from $|s_0\rangle$, a total of $n_0^I$ iterations are performed of the transformation

$$|s_{j+1}\rangle = -\mathcal{I}_{s_j} \mathcal{I}_j |s_j\rangle, \quad j = 0, \ldots, n_0^I - 1, \quad (21)$$

where

$$n_0^I = \tilde{n} - p. \quad (22)$$

The unitary operator $\mathcal{I}_j$ in (21) is defined as

$$\mathcal{I}_j = I - 2 \sum_{i|F_{j+1}(w_i)=1} |w_i\rangle\langle w_i|, \quad (23)$$

where $I$ is the identity operator. In terms of its action on computational-basis states,

$$\mathcal{I}_j |w_i\rangle = (-1)^{F_{j+1}(w_i)} |w_i\rangle. \quad (24)$$

The unitary operator $\mathcal{I}_{s_j}$ in (21) is defined as

$$\mathcal{I}_{s_j} = I - 2 |s_j\rangle\langle s_j|. \quad (25)$$

The proof that, after $n_0^I$ iterations, the resulting state $|s_{n_0^I}\rangle$ is an equally-weighted superposition of the $\nu_0$ states $w_i \in T$ proceeds by induction. Using (20), (21), (24) and (25), we find, for $j = 0$,

$$|s_1\rangle = -\frac{1}{\sqrt{\tilde{N}}} \left[ \sum_{i=1}^{\tilde{N}} (-1)^{F_1(w_i)} |w_i\rangle - \frac{2}{\sqrt{\tilde{N}}} \left( \sum_{i=1}^{\tilde{N}} (-1)^{F_1(w_i)} \right) |s_0\rangle \right]. \quad (26)$$

To evaluate the second sum in (26), divide the set of $\tilde{N}$ states into two groups, those for which $S_2(w_i) = 00$ and those for which $S_2(w_i) \neq 00$. The first group contains $2^{2(\tilde{n}-1)}$ states, of which the $2^{2(\tilde{n}-1)} - \nu_0$ states not in $G$ have $F_1(w_i) = 1$, and the remaining $\nu_0$ states in $G$ have $F_1(w_i) = 0$ (see eqs. (17), (18)). Of the $3 \cdot 2^{2(\tilde{n}-1)}$ states with $S_2(w_i) \neq 00$, $\nu_0$ of these have $F_1(w_i) = 1$ by virtue of being target states ($f(w_i) = 1$), and the remaining $3 \cdot 2^{2(\tilde{n}-1)} - \nu_0$ have $F_1(w_i) = 0$. So,

$$\sum_{i=1}^{\tilde{N}} (-1)^{F_1(w_i)} = \frac{\tilde{N}}{2} = 2^{2\tilde{n}-1}, \quad (27)$$

and (26) reduces to

$$|s_1\rangle = 2^{-\tilde{n}+1} \sum_{i|F_1(w_i)=1} |w_i\rangle. \quad (28)$$
We now assume that for some \( j \),

\[
|s_j\rangle = 2^{-\bar{n}+j} \sum_{i|F_j(w_i)=1} |w_i\rangle,
\]

and derive the form of \( |s_{j+1}\rangle \). From (29), (21), (24) and (25),

\[
|s_{j+1}\rangle = -2^{-\bar{n}+j} \left[ \sum_{i|F_{j+1}(w_i)=1} (-1)^{F_{j+1}(w_i)} |w_i\rangle - 2^{-\bar{n}+j+1} \left( \sum_{i|F_j(w_i)=1} (-1)^{F_j(w_i)} \right) |s_j\rangle \right]. \tag{30}
\]

The second sum in (30) can again be evaluated by counting. The items \( w_i \) for which \( F_j(w_i) = 1 \) fall into two disjoint groups, those for which \( f_j(w_i) = 1 \), and the elements of \( T \). Of the former group, \( 2^{2(\bar{n}+1)-\nu_0} \) have \( F_{j+1}(w_i) = 1 \) (those with \( S_{2j+2}(w_i) = 00\ldots00 - \) recall that the elements of \( G \) are not members of \( \{w_i|F_k(w_i) = 1\} \) for any \( k \)), and the remaining \( 3 \cdot 2^{2(\bar{n}+1)-1} \) have \( F_{j+1}(w_i) = 0 \). As for the elements of \( T \), all \( \nu_0 \) have \( F_{j+1}(w_i) = 1 \). Therefore,

\[
\sum_{i|F_j(w_i)=1} (-1)^{F_{j+1}(w_i)} = 2^{2(\bar{n}+1)-1}, \quad j = 1, \ldots, \bar{n} - p - 1. \tag{31}
\]

Using (31) in (30), we obtain

\[
|s_{j+1}\rangle = 2^{-\bar{n}+j+1} \sum_{i|F_{j+1}(w_i)=1} |w_i\rangle. \tag{32}
\]

After applying \( n_0^I \) iterations (21) to the starting state (27), we therefore obtain (keeping in mind that \( F_{n_0^I}(w_i) = f(w_i) \))

\[
|s_{n_0^I}\rangle = 2^{-\nu} \sum_{i|w_i \in T} |w_i\rangle. \tag{33}
\]

A measurement of \( |s_{n_0^I}\rangle \) in the computational basis will with certainty yield one of the states corresponding to a target item.

### 3 Number of Targets Not a Power of Four

Only a small number of changes are required in the analysis presented above to produce an algorithm which will yield one of the target states with a probability greater than one-quarter when the number of targets is not a power of four, and which reduces to the algorithm of Section 2 when the number of targets is a power of four. All of the definitions through the selection of the ground-state items, eq. (13), remain applicable. However, the integer \( p \) defined in (14) must be everywhere replaced with \( \tilde{p} \)

\[
2^{2\tilde{p}} = \nu, \tag{34}
\]

where \( \nu \) is the smallest power of four larger than \( \nu_0 \). I.e.,

\[
\tilde{p} = \lceil \log_4 \nu_0 \rceil, \tag{35}
\]
The rightmost $2\tilde{p}$ entries in $S(w_{G,\nu_0})$ constitute a binary representation of $\nu_0 - 1$, but they will be not all 1’s.  The definitions (17) and (18) of the auxiliary functions $f_j(w_i)$ and the auxiliary oracle functions $F_j(w_i)$ remain unchanged.  However, most significantly, eq. (19) is replaced with

\[
\{w_i | F_{\tilde{n} - \tilde{p}}(w_i) = 1\} \supset T,
\]

since not all items with $S_{2(\tilde{n} - \tilde{p})} = 00 \ldots 00$ are in $G$.

So, a derivation parallel to that in Section 2 leads to the conclusion that, by beginning with the initial state (20) and performing $\tilde{n} - \tilde{p}$ iterations (21), we obtain the state

\[
|s_{\tilde{n} - \tilde{p}}\rangle = 2^{-\tilde{p}} \sum_{i | F_{\tilde{n} - \tilde{p}}(w_i) = 1} |w_i\rangle.  \tag{38}
\]

If a measurement in the computational basis is made of the state (38), the probability that one of the target states will be obtained is

\[
P_0(\rho) = \rho, \tag{39}
\]

where

\[
\rho = \frac{\nu_0}{\nu}. \tag{40}
\]

The probability of finding a target state is thus between one, when $\nu_0 = \nu$ ($\rho = 1$), and somewhat above one-quarter, when $\nu_0 = \nu/4 + 1$ ($\rho = 1/4 + 1/\nu$).

Now suppose that, rather than making a measurement after $\tilde{n} - \tilde{p}$ iterations, we perform an “extra” iteration, i.e., compute

\[
|s_{\tilde{n} - \tilde{p} + 1}\rangle = -I_{s_{\tilde{n} - \tilde{p}}} |s_{\tilde{n} - \tilde{p}}\rangle. \tag{41}
\]

before measuring.  The definitions (17), (18) of $f_j(w_i)$ and $F_j(w_i)$ work for $j > \tilde{n} - \tilde{p}$ and, with the relations (34), (36), imply that, regardless of the value of $\nu_0$,

\[
F_{\tilde{n} - \tilde{p} + q}(w_i) = f(w_i), \quad q \geq 1. \tag{42}
\]

For $j = \tilde{n} - \tilde{p}$ the summation formula corresponding to (31) is

\[
\sum_i |F_{\tilde{n} - \tilde{p}}(w_i) = 1(-1)^F_{\tilde{n} - \tilde{p} + 1}(w_i) = \sum_i |F_{\tilde{n} - \tilde{p}}(w_i) = 1(-1)^f(w_i) = 2^{2\tilde{p}} - 2\nu_0 \tag{43}
\]

The state resulting after one extra iteration is

\[
|s_{\tilde{n} - \tilde{p} + 1}\rangle = 2^{-\tilde{p} + 1} \left[ (1 - \delta) \sum_{i | f(w_i) = 1} |w_i\rangle - \delta \sum_{i | f_{\tilde{n} - \tilde{p}} = 1} |w_i\rangle \right] \tag{44}
\]

where

\[
\delta = (4\rho - 1)/2. \tag{45}
\]

The probability of obtaining a target state upon measuring $|s_{\tilde{n} - \tilde{p} + 1}\rangle$ is

\[
P_1(\rho) = \rho(3 - 4\rho)^2. \tag{46}
\]
For $1/4 < \rho < 1/2$, $P_1(\rho) > P_0(\rho)$, while, for $1/2 < \rho < 1$, $P_1(\rho) < P_0(\rho)$. So, the
appropriate strategy is to make a measurement after
\[ n_\tilde{I} = \tilde{n} - \tilde{p} \] (47)
iterations if $1/2 \leq \rho < 1$, and to make a measurement after
\[ n_\tilde{I} = \tilde{n} - \tilde{p} + 1 \] (48)
iterations if $1/4 < \rho < 1/2$. The probability of obtaining a target state will in this way be
at least as large as $P_0(1/2) = P_1(1/2) = 1/2$ (see Fig. 1).

Yet another iteration before measurement gives
\[ |s_{\tilde{n}+2} \rangle = 2^{-\tilde{p}+1} \left[ (1 - \delta)(1 - C) \sum_{i|f(w_i)=1} |w_i \rangle + \delta(1 + C) \sum_{i|f_{\tilde{n}-\tilde{p}}(w_i)=1} |w_i \rangle \right] \] (49)
where
\[ C = 8 \left[ (1 - \delta)^2 \rho - \delta^2 (1 - \rho) \right], \] (50)
and a probability of target-finding of
\[ P_2(\rho) = 4\rho (1 - \delta)^2 (1 - C)^2. \] (51)

Despite the extra iteration, the probability of obtaining a target state when $\rho = 1/2$ is not increased; $P_2(1/2) = 1/2$. This is true for an arbitrary number of additional iterations. The quantum state obtained after $\tilde{n} - \tilde{p} + q$ iterations, $q \geq 1$, is of the form
\[ |s_{\tilde{n}+2} \rangle = 2^{-\tilde{p}+1} \left[ A_q \sum_{i|f(w_i)=1} |w_i \rangle + B_q \sum_{i|f_{\tilde{n}-\tilde{p}}(w_i)=1} |w_i \rangle \right], \] (52)
where $A_q$ and $B_q$ satisfy the recursion relations
\[ A_{q+1} = \left( 1 - 8 \left[ A_q^2 \rho - B_q^2 (1 - \rho) \right] \right) A_q, \] (53)
\[ B_{q+1} = - \left( 1 + 8 \left[ A_q^2 \rho - B_q^2 (1 - \rho) \right] \right) B_q. \] (54)
The probability of finding a target upon measurement is
\[ P_q(\rho) = 4A_q^2 \rho. \] (55)

From (44) and (52) we see that $A_1 = 1/2$ and $B_1 = -1/2$ when $\rho = 1/2$. The relations (53)-(55) then show that
\[ P_q(1/2) = 1/2 \quad \forall \ q \geq 1. \] (56)

This is not in any sense to claim that iteration algorithms different than those considered here might not improve on the probability of finding a target when $\rho = 1/2$. Nor is it to say that iterations beyond $\tilde{n} - \tilde{p} + 1$ necessarily have no use. Probability functions $P_q(\rho)$, $q \geq 2$, can, for values of $\rho \neq 1/2$, be larger than either $P_0(\rho)$ or $P_1(\rho)$, indeed as large as 1 (see Fig. 1).
4 Required Number of Oracle Calls

Grassl [2] and Tu and Long [3] have presented the following implementations of the operators $I_j$ and $I_{s_j}$, and have evaluated the number of oracle calls required each time these operators are applied. From eq. (24) we see that $I_j$ can be written as

$$I_j = \sum_i (-1)^{F_{j+1}(w_i)} |w_i\rangle \langle w_i|.$$  \hspace{1cm} (57)

From the condition (12) on the representation of elements of $D$ (and, therefore, on all elements of the target set $T$), and the definitions (17), (18) of $f_j$, $F_j$, it follows that

$$(-1)^{F_{j+1}(w_i)} = (-1)^{f(w_i)} (-1)^{f_{j+1}(w_i)}.$$  \hspace{1cm} (58)

Therefore

$$I_j = \left( I - 2 \sum_{i|w_i\in T} |w_i\rangle \langle w_i| \right) \left( \sum_k (-1)^{f_{j+1}(w_k)} |w_k\rangle \langle w_k| \right),$$  \hspace{1cm} (59)

and we see that each application of $I_j$ requires a single call to the oracle, since the $f_j$’s are independent of $f$.

From the iteration condition (21), the definition (25) of $I_{s_j}$, and the unitarity of $I_j$ and $I_{s_j}$, we see that the operators $I_{s_j}$ satisfy the relation

$$I_{s_{j+1}} = I_{s_j} I_j I_{s_j} I_j I_{s_j}.$$  \hspace{1cm} (60)

Let $t(j)$ denote the number of oracle calls required by $I_{s_j}$. Since $I_j$ requires one oracle call, (60) implies

$$t(j + 1) = 3t(j) + 2.$$  \hspace{1cm} (61)

For $j = 0$,

$$I_{s_0} = I - 2 |s_0\rangle \langle s_0|,$$  \hspace{1cm} (62)

which is independent of $f$, so

$$t(0) = 0$$  \hspace{1cm} (63)

and $t(j)$ has the closed form

$$t(j) = 3^j - 1.$$  \hspace{1cm} (64)

Taking into account the single oracle call required by $I_j$, the total number of oracle calls required for $n_I$ iterations of (21) is

$$C(n_I) = \sum_{j=0}^{n_I-1} t(j) + n_I$$  \hspace{1cm} (65)

which, using (64), has the value

$$C(n_I) = (1/2) (3^{n_I} - 1).$$  \hspace{1cm} (66)

It follows from the results of Section 2 that, for $\nu_0$ a power of four, the required number of oracle calls to obtain a target with unit probability is

$$C^0 = C(n_I^0) = (1/2) \left( 3\left( N/\nu_0 \right)^{\log_4 3} - 1 \right).$$  \hspace{1cm} (67)
If \( \nu_0 \) is not a power of four, the results of Section 3 imply that the number of oracle calls to obtain a target state with probability of at least one half is

\[
C^\geq = C(n^\geq_I) = (1/2) \left( 3(N/\nu)^{\log_4 3} - 1 \right)
\]

(68)

if \( \rho = \nu_0/\nu \) is between 1/2 and 1, and

\[
C^\leq = C(n^\leq_I) = (1/2) \left( 9(N/\nu)^{\log_4 3} - 1 \right)
\]

(69)

if \( \rho \) is between 1/4 and 1/2.

The original algorithm of Chen and Diao \[1\] performs two series of \( n \) iterations of \( I_i \), so the number of oracle calls required to find the unique target item by that method is

\[
C_{CD} = 2C(n) = 3N^{\log_4 3} - 1.
\]

(70)

The exponent \( \log_4 3 \) is approximately equal to 0.7925. So, with this particular implementation of the operators \( I_j \) and \( I_s \), the computational complexity of the algorithms of \[1\] and the present paper scales more slowly than that of the best possible classical algorithm \( (O(N)) \), but not as slowly as that of Grover’s algorithm \[5\] \( (O(\sqrt{N})) \). Unlike Grover’s algorithm, these algorithms will find a target item with certainty \(^1\) if the number of targets is a power of four. It is not known at present whether the implementation employed here is the most efficient possible, or if implementations requiring fewer oracle calls may exist.

**Acknowledgments**

I would like to thank Markus Grassl for providing a prepublication copy of \[2\], and the anonymous reviewer of the previous version of this paper for also pointing out the relation \( (60) \).

**References**

[1] G. Chen and Z. Diao, “Exponentially fast quantum search algorithm,” quant-ph/0011109 v3 (2000).

[2] M. Grassl, “Comment on ‘An exponentially fast quantum search algorithm,’ ” unpublished (2001).

[3] C. C. Tu and G. L. Long, “Chen and Diao’s quantum search algorithm is not exponentially fast,” quant-ph/0110098 (2001).

[4] N. D. Mermin, “Lecture notes for physicists on the theory of quantum computation,” http://www.lassp.cornell.edu/lassp_data/NMermin.html

[5] L. K. Grover, in *Proc. 28th Annual Symposium on the Theory of Computing* (ACM Press, New York, 1996), p. 212; *Phys. Rev. Lett.* 78, 325 (1997); *Phys. Rev. Lett.* 80, 4329 (1998).

\(^1\)Versions of Grover’s algorithm which find targets with certainty have been presented in \[6, 7, 8\].
[6] G. Brassard, P. Høyer, M. Mosca, A. Tapp, “Quantum amplitude amplification and estimation,” quant-ph/0005055 (2000).

[7] P. Høyer, “On arbitrary phases in quantum amplitude amplification,” Phys. Rev. A62, 052304 (2000); quant-ph/0006031.

[8] G. L. Long, “Grover algorithm with zero theoretical failure rate,” quant-ph/010607 (2001).
Figure Caption

Figure 1. Probability $P_q$ of finding a target with $q$ “extra” iterations, as a function of $\rho$. Solid line: $q = 0$. Dashed line: $q = 1$. Dotted line: $q = 2$. 
