Response to Ramesh & Vinay, (2003)

String Matching in $\tilde{O}(\sqrt{n} + \sqrt{m})$ Quantum Time

Matthew Evans, Ariz Siddiqui, Nathan Puskuri

I. Overview

Ramesh & Vinay, (2003) [**?**] addresses the problem of efficiently determining whether a smaller string (pattern) appears within a larger one (text). Traditional solutions to this problem on classical computers take time roughly proportional to the total length of both the pattern and the text. The authors propose a quantum algorithm that determines the existence of a match with probability better than $\frac{1}{2}$. While constrained in scope, their approach demonstrates how quantum computing methods, particularly quantum search techniques, can potentially achieve faster-than-classical performance for specific string matching scenarios.

We will first cover the concepts and dependencies to which the authors appeal, and then explain how these pieces fit together to support the authors' claims.

II. Preliminaries

A. Grover's Algorithm

Grover's algorithm [**?**] addresses the problem of locating one of t marked items in an unsorted database of N entries with a quadratic speedup over classical search. The procedure comprises the following steps:

1) Initialize a superposition over all N basis states.
2) Apply the oracle O_f to flip the phase of the t marked states.
3) Perform the diffusion operator to reflect amplitudes about their average.
4) Repeat the oracle and diffusion operations approximately $\frac{\pi}{4}\sqrt{N}$ times.
5) Measure the register to obtain a marked element with high probability.

Running in $O(\sqrt{N})$ time, Grover's algorithm provides the core quantum search subroutine underpinning the $\widetilde{O}(\sqrt{n} + \sqrt{m})$ string-matching algorithm.

B. Tight Bounds on Grover's (BBHT96)

[**?**, Boyer, Brassard, Høyer, and Tapp] show that even when the exact number of target items t in an unsorted database of size N is unknown, one can still locate a marked element in $O(\sqrt{(N/t)})$ oracle calls with constant success probability. Their procedure adaptively adjusts the number of Grover iterations to achieve the same quadratic speedup without prior knowledge of t. In our string-matching algorithm, this result is crucial: when searching a text block of length $m/2$ for any matching alignment, we do not know in advance how many alignments will succeed, so we invoke BBHT96's method to retain the overall $\widetilde{O}(\sqrt{n} + \sqrt{m})$ runtime bound.

C. Deterministic Sampling (VI)

Deterministic sampling, introduced by [**?**, Vishkin] is a procedure for reducing the number of expensive pattern-matching checks on an aperiodic pattern p of length m. When matching p against a text block of length $m/2$, there are $m/2$ possible alignments. Deterministic sampling picks $O(\log m)$ indices in the pattern such that at most one alignment can match at all of these sample positions. The procedure is:

1) Form $m/2$ "copies" of the pattern, each shifted by one position.
2) Identify a column (i.e. pattern index) where at least two copies have different symbols.
3) Choose one of the observed symbols as the sample value and discard all copies that do not match it.
4) Repeat the previous two steps $O(\log m)$ times until only one copy remains.
5) Collect the chosen columns; these indices constitute the deterministic sample S.

Because only the single surviving alignment must undergo a full (quantum) check of cost $\widetilde{O}(\sqrt{m})$, deterministic sampling reduces the number of such expensive checks per text block from m to 1. This reduction is crucial to achieving the overall $\widetilde{O}(\sqrt{n}+\sqrt{m})$ runtime of the quantum string-matching algorithm.

D. Minimum Finding Oracle

The minimum-finding oracle given by [**?**, Durr & Høyer] locates the index of the smallest element in an unsorted database of size n in $O(\sqrt{n})$ time using a comparison oracle. Beginning from a random index k, it invokes Grover's algorithm to find any index i such that database$[i]$ < database$[k]$. If one is found, k is updated to i and the search repeats; if not, k is the minimum. This primitive underlies every "pick the smallest (or leftmost) index satisfying some condition" step in the paper. It is used to build the deterministic sample for aperiodic patterns—repeatedly halving the $m/2$ candidate alignments in $\widetilde{O}(\sqrt{m}\log m)$ time—and is invoked a second time to pinpoint the earliest text block containing the match, thereby preserving the overall $\widetilde{O}(\sqrt{n}+\sqrt{m})$ runtime bound.

III. Quantum String Matching

A. The Algorithm

1) **Deterministic-Sampling Preprocessing.**
   - Run Vishkin's deterministic-sampling on p of length m to obtain an $O(\log m)$-sized sample set S.
   - Cost: $\widetilde{O}(\sqrt{m}\,\log^2 m)$.
2) **Partition the text.**
   - Divide the text t into
   $$B = \left\lceil \frac{2(n-m+1)}{m} \right\rceil$$
   blocks, each of size $\approx m/2$.
3) **Quantum search for a "hit" block.**
   - Define oracle $h(i)$: tests if block i has at least one alignment matching on all positions in S in $\widetilde{O}(\sqrt{m}\,\log m)$ time.
   - Use Grover search over $i = 1, \ldots, B$ with oracle h; time $\widetilde{O}(\sqrt{n}\,\log m)$. If none found, conclude "no occurrence."
4) **Locate surviving alignment in block.**
   - Define oracle $k(i^*, j)$: checks alignment at shift j in block i^* on sample S in $O(\log m)$ time.
   - Use Grover search over $j = 0, \ldots, \lfloor m/2 \rfloor$ with oracle k; time $\widetilde{O}(\sqrt{m}\,\log m)$. If none survives, conclude "no occurrence."
5) **Full quantum verification.**
   - Run Grover search over $\ell = 1, \ldots, m$ with oracle "$t[i^* + j^* + \ell] \neq p[\ell]$" to find any mismatch; time $\widetilde{O}(\sqrt{m})$.
   - If no mismatch is found, report occurrence at $i^* + j^*$; otherwise, conclude "no occurrence."

B. Concerning Periodicity

C. Constant Two-Sided Failure Probability

D. Achieving $\tilde{O}(\sqrt{n}+\sqrt{m})$

References

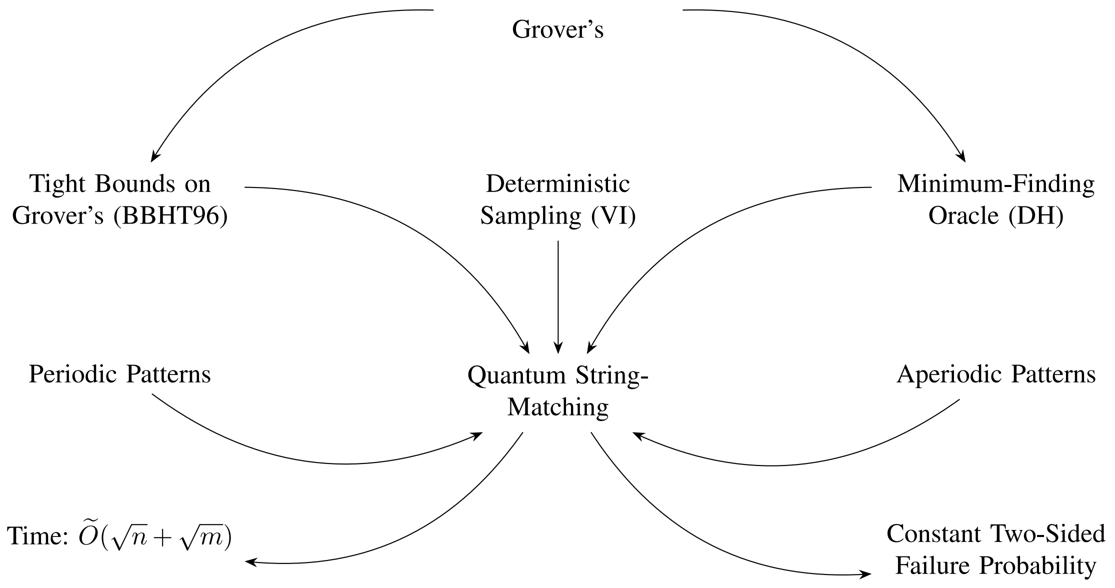
Grover's
Tight Bounds on Grover's (BBHT96)
Deterministic Sampling (VI)
Minimum-Finding Oracle (DH)
Periodic Patterns
Quantum String-Matching
Aperiodic Patterns
Time: $\widetilde{O}(\sqrt{n}+\sqrt{m})$
Constant Two-Sided Failure Probability

Fig. 1. Conceptual dependencies in the $\widetilde{O}(\sqrt{n}+\sqrt{m})$ quantum string-matching algorithm.