Pramana – J. Phys. (2021) 95:22
https://doi.org/10.1007/s12043-020-02062-0

Quantum pattern matching oracle construction

VIKRAM MENON[1] and AYAN CHATTOPADHYAY[2],*

[1]Confident Bellatrix, Bengaluru 562 125, India
[2]Assetz East Point, Bengaluru 562 103, India
*Corresponding author. E-mail: ayankc@yahoo.com

MS received 10 May 2020; revised 25 September 2020; accepted 15 October 2020

Abstract. We propose a couple of oracle construction methods for quantum pattern matching. One of the constructs is based upon the conventional string comparison method. This, along with a unique input state preparation, when combined with the Grover's search algorithm, results in a deterministic exact and partial pattern matching logic. The other method generates a superposition of Hamming distances between the searched pattern and all the substrings formed from the input string. The measurement statistics from a large ensemble would provide data on the closest match. We show that this method can leverage parallel computing for enhanced performance. Alternatively, it can also be combined with the minimum finding algorithm for a deterministic outcome.

Keywords. Pattern matching; quantum search; quantum pattern matching.

PACS Nos 12.60.Jv; 12.10.Dm; 98.80.Cq; 11.30.Hv

1. Introduction

Pattern searching has wide applications in various modern fields like data analysis, molecular biology especially in DNA sequencing, search engines, AI, etc. In fact, this is a basic tool to extract useful information for various cases. Usually, the size of the text string is moderate to very large and, matching it with a given or a set of patterns, involves large time complexity. Till date, many well performing classical algorithms have been devised, most notably Knuth Morris Pratt (KMP) and Boyer Moore algorithms. Classical algorithms are best run on classical computers, but they cannot exploit the power of quantum mechanical principle. Efforts are on to create efficient pattern matching algorithms using quantum mechanical principles, so that they run on quantum computers or on quantum simulators. To enhance the searching performance, it is important to effectively pre-process the available data so as to exploit quantum parallelism that can improve the run-time. Ramesh and Vinay [1] initially proposed a quantum method for substring matching in $\tilde{O}(\sqrt{N}+\sqrt{M})$ running time, combining Grover's search algorithm [2] and deterministic sampling. Quantum algorithms for closest pattern matching were given in [3,4]. Their attempts allow one to search for many distinct patterns in a given string, using a query function per symbol of the pattern alphabet. It returns the position of the closest substring to a given pattern with non-negligible probability in $O(\sqrt{N})$ queries, where N is the size of the string.

Motivated by the works cited, our intention is to articulate a methodology to leverage parallel processing using Grover's search algorithm which is the most efficient algorithm discovered till date. Here, we propose two oracle construction methods. The first construction will use the Grover's search algorithm for exact pattern (substring of size M) match in a given input string (size N). The search will return the position of one of the occurrences (in the case of multiple match) of the substring, in $\sqrt{N-M}$ oracle queries. We shall later extend this to variable length search. The input quantum state preparation will use the naive pattern search technique. This technique will be efficient as it uses Grover's search algorithm.

The second oracle will generate the Hamming distances of the pattern to be searched and all the possible substrings in the given input. The measurement outcome will be probabilistic. To get all the matched indices, a large number of iterations $(N-M)$ need to be performed in proportion to the running time. But we shall show that

Published online: 06 February 2021

by leveraging quantum parallelism, running time can be reduced.

2. Oracle construction using Grover's search algorithm

Let us consider an input string of size N, over an alphabet set $\sum$. For any pattern P of size, say $M \ll N$, the oracle can be defined as

$$f_P(|x\rangle) = \begin{cases} 1, & \text{if } |x\rangle \text{ is a solution} \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where $|x\rangle$ is a solution, if the given pattern matches the substring at position x of the input string. A substring at position x, S_x, will constitute the alphabets in the range $[x, x+M-1]$.

This construction has a search space of size $(N-M+1)$. The input state is given by

$$\frac{1}{\sqrt{T}}\sum_{x=0}^{T-1}|x\rangle, \tag{2}$$

where $2^t = T = (N-M+1)$. The basis states, $|0\rangle$ to $|T-1\rangle$, map directly to all the possible search positions in the input string.

The Grover's search algorithm, with the above construction, will require $\frac{\pi}{4}\sqrt{T/K}$ invocations of the oracle, where K is the number of solutions, i.e. how many substrings exactly match the given pattern. The solution count, if unknown, can be found out using the methods proposed in [5–7].

The measurement will yield the state index corresponding to the position of the matched pattern in the input string. In the case of multiple matches, this corresponds to one of the indices at random. Running multiple iterations of the algorithm will retrieve all the solutions.

By modifying the oracle to match L alphabets instead of M, with $L < M$, a prefix pattern matching can be achieved.

3. Oracle construction to generate Hamming distance

In this method, the oracle is defined as

$$f_P(|x\rangle \otimes |S_x\rangle \otimes |0\rangle^{\otimes m}) = |x\rangle \otimes |S_x\rangle \otimes |P \oplus S_x\rangle, \tag{3}$$

where S_x is the substring at index x and $2^m = M$ ancillas to hold the result $|P \oplus S_x\rangle$.

The input state, derived from eq. (2) in the previous method, is given as follows:

$$\frac{1}{\sqrt{T}}\sum_{x=0}^{T-1}|x\rangle \otimes |S_x\rangle \otimes |0\rangle^{\otimes m}. \tag{4}$$

It is assumed that the oracle knows the pattern to be searched. Therefore, when applied to the state given by eq. (2), it will generate the Hamming distances of all the substrings with the given pattern.

$$f_P\left(\frac{1}{\sqrt{T}}\sum_{x=0}^{T-1}|x\rangle \otimes |S_x\rangle \otimes |0\rangle^{\otimes m}\right)$$
$$= \frac{1}{\sqrt{T}}\sum_{x=0}^{T-1}|x\rangle \otimes |S_x\rangle \otimes |P \oplus S_x\rangle. \tag{5}$$

Now, measure the third register, i.e. the ancilla qubits. A value of '0' indicates an exact match, with the first register containing the matched index. A second measurement on the first register would output the matched index, or one of the indices in the case of multiple match. A non-zero value in the first measurement indicates a partial match and represents the Hamming distance between the searched pattern and the substring at the index pointed by the first register.

Measurements on a large ensemble of states given by eq. (5) will provide a map of how close the pattern is, with all the substrings.

The parallel run of oracle will improve the running time. The amortized running time will be $O(M)$ because the unit consisting of pre-processing and the oracle can be run in parallel. The number of such parallel units will be $N-M+1$. Any $|x\rangle$ state in the defined range (need not be sequential) can be fed to any of the units.

Let us take an arbitrary example of DNA sequencing to show how the above articulated algorithm would work. There are four types of nucleotides – Adenine (A), Thymine (T), Guanine (G), Cytosine (C). For simplicity, we have omitted Uracil, the 5th one. Suppose

$$\begin{aligned} |A\rangle &= |000\rangle \\ |T\rangle &= |010\rangle \\ |G\rangle &= |100\rangle \\ |C\rangle &= |110\rangle. \end{aligned} \tag{6}$$

We represent the pattern P as a tensor product of a particular combination of the above four bases. For example, a pattern of $AGGCA$ can be represented as $|A\rangle \otimes |G\rangle \otimes |G\rangle \otimes |C\rangle \otimes |A\rangle$.

The oracle operation, for an input string $AATTTGCCCCAGGCACGGGA$ is given by

$$\begin{aligned}
&f_{\{P\}}\left(\frac{1}{\sqrt{16}}\sum_{x=0}^{15}|x\rangle\otimes|S_x\rangle\otimes|0\rangle^{\otimes 5}\right)\\
&\quad=\frac{1}{4}[|0\rangle\otimes|AATTT\rangle\otimes|AGGCA\oplus AATTT\rangle\\
&\qquad+|1\rangle\otimes|ATTTG\rangle\otimes|AGGCA\oplus ATTTG\rangle\\
&\qquad+\cdots+|10\rangle\otimes|AGGCA\rangle\otimes|AGGCA\rangle\\
&\qquad\oplus AGGCA+\cdots+|15\rangle\otimes|CGGGA\rangle\\
&\qquad\otimes|AGGCA\oplus CGGGA\rangle]\\
&\quad=\frac{1}{4}[|0\rangle\otimes|AATTT\rangle\otimes|01111\rangle+|1\rangle\otimes|ATTTG\rangle\\
&\qquad\otimes|01111\rangle+\cdots+|10\rangle\otimes|AGGCA\rangle\\
&\qquad\otimes|00000\rangle+\cdots+|15\rangle\otimes|CGGGA\rangle\otimes|10010\rangle].
\end{aligned} \tag{7}$$

All the 16 units above can be run in parallel, as oracle units are functionally identical and mutually exclusive. A measurement outcome of $|00000\rangle$ on the third register indicates an exact match. This will result in the first register attaining the state $|10\rangle$, which represents the index of the matched substring. Additionally, the outcomes $|00001\rangle$ and $|00011\rangle$ indicate the subsequent longest prefix matches.

Alternatively, the output of eq. (7) can be used as an input to the minimum finding algorithm [8], to obtain the minimum Hamming distance and therefore, the closest pattern match. Here, the advantage of parallel computation is lost. However, the running time is comparable with that of Grover's search.

4. Conclusion

We have shown two methods of pattern matching oracle constructions. The first one, in combination with the Grover's search algorithm, provides a means to perform exact and partial pattern matching. The second oracle is efficient due to its parallel processing nature and provides statistics of how close the search pattern is, with respect to all the possible substrings. Thus, the second oracle construction can have a wide application in pattern searching.

The oracle running time in both the methods will be determined by the respective constructions, which can be studied further. Another interesting area to explore would be the different ways of preparing and preprocessing an input string.

References

[1] H Ramesh and V Vinay, *J. Discrete Algorithms* **1**, 103 (2003)
[2] L Grover, A fast quantum mechanical algorithm for database search, *Proceedings of 28th ACM Symposium on Theory of Computing*, 1996, pp. 212–219
[3] P Mateus and Y Omar, arXiv:quant-ph/0508237 (2005)
[4] S Kapil and R Akhtar, https://doi.org/10.1016/j.procs.2020.03.230.
[5] M Boyer, G Brassard, P Høyer and A Tapp, arXiv:quant-ph/9605034
[6] G Brassard, P Høyer and A Tapp, arXiv:quant-ph/9805082
[7] C R Wie, arXiv:1907.08119
[8] C Dürr and P Høyer, arXiv:quant-ph/9607014