Math. Struct. in Comp. Science (2018), *vol.* 28, *pp.* 1–13. © Cambridge University Press 2016
doi:10.1017/S0960129516000013 First published online 9 February 2016

A quantum algorithm to approximate the linear structures of Boolean functions

HONGWEI LI[†,‡,§,¶] and LI YANG[†,§,‖]

[†]*State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China*
Email: hwli428@sina.com
[‡]*School of Mathematics and Statistics, Henan Institute of Education, Zhengzhou 450046, Henan, China*
[§]*Data Assurance and Communication Security Research Center, Chinese Academy of Sciences, Beijing 100093, China*
Email: yangli@iie.ac.cn
[¶]*University of Chinese Academy of Sciences, Beijing 100049, China*

Received 3 November 2014; revised 19 June 2015

A quantum algorithm to determine approximations of linear structures of Boolean functions is presented and analysed. Similar results have already been published (see Simon's algorithm) but only for some promise versions of the problem, and it has been shown that no exponential quantum speedup can be obtained for the general (no promise) version of the problem. In this paper, no additional promise assumptions are made. The approach presented is based on the method used in the Bernstein–Vazirani algorithm to identify linear Boolean functions and on ideas from Simon's period finding algorithm. A proper combination of these two approaches results here to a polynomial-time approximation to the linear structures set. Specifically, we show how the accuracy of the approximation with high probability changes according to the running time of the algorithm. Moreover, we show that the time required for the linear structure determine problem with high success probability is related to so called relative differential uniformity δ_f of a Boolean function f. Smaller differential uniformity is, shorter time is needed.

1. Introduction

Linear structures of Boolean functions are of importance in cryptography (Connor and Klapper 1994; Dubuc 2001; Feng and Pei 1999; Feng and Xiao 1995; Lai 1995). Given a quantum oracle to a multiple output Boolean function g, under the promise that g is one to one or g has a period (i.e. a linear structure of g), Simon's algorithm (Simon 1997) could efficiently determine the exact case and find the period in the second case. Inspired by Simon's algorithm, Shor (1997) provided a polynomial-time algorithm for factoring integers. Both Simon's and Shor's quantum algorithms exhibit exponential speedups over the best-known classical algorithms. Unfortunately, as shown in Beals *et al.* (2001), the exponential speedup obtained for promise versions of these problems cannot be obtained for the general case.

‖ Corresponding author.

https://doi.org/10.1017/S0960129516000013 Published online by Cambridge University Press

Recently, probabilistic quantum algorithms have been widely used to approximately solve problems (Aharonov *et al.* 2009; Brassard *et al.* 2002; Brassard *et al.* 2011; Drucker and Wolf 2011). In this paper, we present and analyse a new quantum algorithm for approximating the linear structures set of a Boolean function f with no promise at all.

The Deutsch–Jozsa algorithm (Deutsch and Jozsa 1992) and the Bernstein–Vazirani algorithm (Bernstein and Vazirani 1993) can be expressed as the same network (Cleve *et al.* 1998). Suppose that we run a quantum network to compute an n-bit Boolean function, but we do not perform the final projective measurement. The output will then be a superposition of all states $|w\rangle$ ($w \in \{0,1\}^n$), and the amplitude corresponding to each state $|w\rangle$ is its Walsh spectrum value $S_f(w)$. Hillery *et al.* have studied the properties of Boolean functions based on the Bernstein–Vazirani algorithm (Floess *et al.* 2013; Hillery and Anderson 2011). In addition, a link between the Walsh spectrums and the linear structures of Boolean functions has been shown in Dubuc (2001) and Feng and Pei (1999).

The Bernstein–Vazirani algorithm and results in Dubuc (2001) and Feng and Pei (1999) are behind the results in this paper. At first, we generalize a result from them and then we present our algorithm and analyse it using the result obtained by the above generalization.

2. Preliminaries and notations

2.1. *The linearity of Boolean functions*

Let n be a positive integer. $F_2 = \{0, 1\}$ denotes the finite field of characteristic 2, and F_2^n is the vector space over F_2. A mapping from F_2^n to F_2 is called a Boolean function. Let $\mathfrak{B}_n$ denote the set of all Boolean functions of n variables.

Definition 2.1. A vector $a \in F_2^n$ is defined as a linear structure of a function $f \in \mathfrak{B}_n$, if

$$f(x \oplus a) + f(x) = f(a) + f(0), \ \forall x \in F_2^n, \tag{1}$$

where $\oplus$ denotes bitwise exclusive-or, the addition operation in F_2^n.

Let U_f denote the set of all the linear structures of f, and

$$U_f^i = \{a \in F_2^n \mid f(x \oplus a) + f(x) = i, \forall x \in F_2^n\} \ (i = 0, 1). \tag{2}$$

Obviously, $U_f = U_f^0 \bigcup U_f^1$.

Let

$$V_{f,a}^i = \{x \in F_2^n \mid f(x \oplus a) + f(x) = i\} \ (i = 0, 1), \ \forall a \in F_2^n. \tag{3}$$

Let $|V|$ denote the cardinality of V. Obviously, $0 \leqslant |V_{f,a}^i|/2^n \leqslant 1$. $a \in U_f^i$ if and only if $|V_{f,a}^i|/2^n = 1$. In this paper, we will use $1 - |V_{f,a}^i|/2^n$ to describe the extent to which a vector a approximates a linear structure, and we expect that it will be sufficiently small. These expectations will be formulated using concepts from the following two definitions:

Definition 2.2. A vector $a \in F_2^n$ is called a quasi linear structure of a function $f \in \mathfrak{B}_n$, if

$$1 - \frac{|\{x \in F_2^n \mid f(x \oplus a) + f(x) = i\}|}{2^n} < l(n), \tag{4}$$

where the function $l(n)$ is negligible. More specifically, for any polynomial $p(\cdot)$, there exists an $N > 0$ such that for all integers $n > N$, $l(n) < 1/p(n)$. That is, Equation (1) holds except for a negligible number of x.

https://doi.org/10.1017/S0960129516000013 Published online by Cambridge University Press

Definition 2.3. The relative differential uniformity (differential uniformity for short) of $f \in \mathfrak{B}_n$ is

$$\delta_f = \frac{1}{2^n} \max_{0 \neq a \in F_2^n} \max_{i \in F_2} |\{x \in F_2^n \mid f(x \oplus a) + f(x) = i\}|. \tag{5}$$

δ_f as defined in Equation (5) satisfies $\frac{1}{2} \leqslant \delta_f \leqslant 1$. Moreover, $U_f \neq \{0\}$ if and only if $\delta_f = 1$.

To study linear structures of a Boolean function, we will need also the Walsh spectrum of Boolean functions.

Definition 2.4. The Walsh spectrum of $f \in \mathfrak{B}_n$ is defined as

$$S_f(w) = \frac{1}{2^n} \sum_{x \in F_2^n} (-1)^{f(x)+w \cdot x}, \tag{6}$$

for all $w \in F_2^n$.

The following two theorems demonstrate the links between the Walsh spectrum and linear structures, and this paper has been much inspired by them.

Theorem 2.1 (Dubuc 2001; Feng and Pei 1999). Let $U_f = U_f^0 \bigcup U_f^1$ be the set of linear structures of $f \in \mathfrak{B}_n$. Then, $a \in U_f^i$ $(i = 0, 1)$ if and only if $S_f(w) = 0$ for all $w \in F_2^n$ with $w \cdot a = \bar{i} = i + 1$ (i.e. $w \cdot a \neq i$).

We will generalize this theorem in the following section. For convenience, let

$$N_f^0 = \{w \in F_2^n \mid S_f(w) = 0\}, \qquad N_f^1 = \{w \in F_2^n \mid S_f(w) \neq 0\}, \tag{7}$$

and let

$$B_a^0 = \{w \in F_2^n \mid w \cdot a = 0\}, \qquad B_a^1 = \{w \in F_2^n \mid w \cdot a = 1\}, \ \forall a \in F_2^n. \tag{8}$$

From Theorem 2.1, one can deduce the following theorem:

Theorem 2.2 (Dubuc 2001; Feng and Pei 1999). For $f \in \mathfrak{B}_n$, $i \in \{0, 1\}$,

$$U_f^i = \{a \in F_2^n \mid a \cdot w = i, \forall w \in N_f^1\}. \tag{9}$$

This result follows from those in Dubuc (2001) and for the case $i = 0$, the result is also given in Feng and Pei (1999). From Theorem 2.2, it follows that if we can determine the set N_f^1, we will get U_f^i. Moreover, we will show that repeating the Bernstein–Vazirani algorithm yields a subset of N_f^1.

https://doi.org/10.1017/S0960129516000013 Published online by Cambridge University Press

2.2. *The Bernstein–Vazirani algorithm*

The Bernstein–Vazirani algorithm (Bernstein and Vazirani 1993; Cleve *et al.* 1998; Nielsen and Chuang 2010) enables us to distinguish linear functions. Specifically, suppose

$$f(x) = a \cdot x = \sum_{i=1}^{n} a_i x_i. \tag{10}$$

The algorithm aims to determine a. We now describe the mechanism of the algorithm.

1. Perform the Hadamard transform $H^{(n+1)}$ on the initial state $|\psi_0\rangle = |0\rangle^{\otimes n}|1\rangle$, giving

$$|\psi_1\rangle = \sum_{x \in F_2^n} \frac{|x\rangle}{\sqrt{2^n}} \cdot \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{11}$$

2. Apply the f-controlled-NOT gate on $|\psi_1\rangle$, producing

$$|\psi_2\rangle = \sum_{x \in F_2^n} \frac{(-1)^{f(x)}|x\rangle}{\sqrt{2^n}} \cdot \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{12}$$

3. Apply again n Hadamard gates to the first n qubits, to obtain

$$|\psi_3\rangle = \sum_{y \in F_2^n} \frac{1}{2^n} \sum_{x \in F_2^n} (-1)^{f(x)+y\cdot x}|y\rangle \cdot \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{13}$$

If $f(x) = a \cdot x$,

$$\begin{aligned} |\psi_3\rangle &= \sum_{y \in F_2^n} \chi_a(y)|y\rangle \cdot \frac{|0\rangle - |1\rangle}{\sqrt{2}} \\ &= |a\rangle \cdot \frac{|0\rangle - |1\rangle}{\sqrt{2}}, \end{aligned} \tag{14}$$

where

$$\chi_a(y) = \begin{cases} 1 & \text{if } y = a, \\ 0 & \text{if } y \neq a. \end{cases} \tag{15}$$

If we now measure the first n qubits of $|\psi_3\rangle$ in the computational basis, we find a with probability 1.

If $f(x) \in \mathfrak{B}_n$ is not linear, then one run of the Bernstein–Vazirani algorithm without the final measurement provides

$$|\psi_3\rangle = \sum_{y \in F_2^n} S_f(y)|y\rangle \cdot \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{16}$$

according to Equations (6) and (13). If we afterwards measure the first n qubits of $|\psi_3\rangle$ in the computational basis, we obtain a y with probability $(S_f(y))^2$ (we will write it as $S_f^2(y)$ for convenience). That is, if we repeat the algorithm again and again, we obtain a $y \in N_f^1$ each time. It is thus conceivable that the Bernstein–Vazirani algorithm can be used to investigate linear structures of Boolean functions.

https://doi.org/10.1017/S0960129516000013 Published online by Cambridge University Press

3. Further relations between linear structures and the Walsh spectrum of Boolean functions

Before describing our quantum algorithm, we present at first the following theorem, which plays a pivotal role in using that algorithm to determine linear structures of Boolean functions. Although the qualitative Theorem 2.1 has built a bridge between linear structures and the Walsh spectrum of Boolean functions, the following quantitative theorem is still required to analyse the quantum algorithm.

Theorem 3.1. Let $f \in \mathfrak{B}_n$, then $\forall a \in F_2^n$, $\forall i \in F_2$,

$$\sum_{w \in B_a^i} S_f^2(w) = \frac{|V_{f,a}^i|}{2^n} = \frac{|\{x \in F_2^n \,|\, f(x \oplus a) + f(x) = i\}|}{2^n}. \tag{17}$$

In Kahn *et al.* (1988) and O'Donnell (2008), the authors provided an expression relating the Fourier transform to the influence of a variable on a Boolean function, which is actually a particular case of Theorem 3.1 (if the Hamming weight of a is 1). Theorem 2.1 is also a specific case of Theorem 3.1 (in which both sides of Equation (17) are 1). To prove Theorem 3.1, we need the following lemma, which will be proved in the appendix.

Lemma 3.1 (Feng and Pei 1999).

$$C_f(a) = \sum_{x \in F_2^n} (-1)^{f(x)+f(x \oplus a)} = 2^n \left(\sum_{w \in B_a^0} S_f^2(w) - \sum_{w \in B_a^1} S_f^2(w) \right), \tag{18}$$

where $C_f(a)$ is the correlation function of f, and $-$ is so called subtraction operation of the integer ring.

Proof of Theorem 3.1 First of all,

$$\begin{aligned} C_f(a) &= |\{x \in F_2^n \,|\, f(x \oplus a) + f(x) = 0\}| - |\{x \in F_2^n \,|\, f(x \oplus a) + f(x) = 1\}| \\ &= |V_{f,a}^0| - |V_{f,a}^1|. \end{aligned} \tag{19}$$

From Equations (18) and (19), we have

$$\sum_{w \in B_a^0} S_f^2(w) - \sum_{w \in B_a^1} S_f^2(w) = \frac{|V_{f,a}^0|}{2^n} - \frac{|V_{f,a}^1|}{2^n}. \tag{20}$$

In addition, Parseval's relation[†] gives

$$\sum_{w \in B_a^0} S_f^2(w) + \sum_{w \in B_a^1} S_f^2(w) = \frac{1}{2^n} \sum_{x \in F_2^n} \left[(-1)^{f(x)}\right]^2 = 1. \tag{21}$$

From the definition of $V_{f,a}^i$, it follows

$$|V_{f,a}^0| + |V_{f,a}^1| = 2^n. \tag{22}$$

[†] Parseval's identity can be found in *Walter Rudin: Real and Complex Analysis, Third Edition.* And we will give a proof of Equation (21) in the appendix.

https://doi.org/10.1017/S0960129516000013 Published online by Cambridge University Press

From Equations (21) and (22), we obtain

$$\sum_{w\in B_a^0} S_f^2(w) + \sum_{w\in B_a^1} S_f^2(w) = \frac{|V_{f,a}^0|}{2^n} + \frac{|V_{f,a}^1|}{2^n}. \tag{23}$$

Combining Equations (20) and (23), we achieve

$$\begin{cases} \sum_{w\in B_a^0} S_f^2(w) = \frac{|V_{f,a}^0|}{2^n}, \\ \sum_{w\in B_a^1} S_f^2(w) = \frac{|V_{f,a}^1|}{2^n}. \end{cases} \tag{24}$$

Equation (24) is essentially the same as Equation (17). □

Theorem 3.1 demonstrates that whenever we run the Bernstein–Vazirani algorithm, the probability of getting w with $w \cdot a = i$ is equal to $\frac{|V_{f,a}^i|}{2^n}$. We will provide our algorithm on the basis of this result in next section.

4. Quantum algorithm for linear structures of Boolean functions

Given a quantum oracle with access to a Boolean function f, how do we decide whether f has non-zero linear structures? We will present a probabilistic quantum algorithm to approximately answer this question. If the differential uniformity δ_f of f is no more than a specific constant independent of n, the algorithm will definitely answer 'no.' If $1 - \frac{1}{e(n)} \leqslant \delta_f \leqslant 1$ ($e(n)$ is an exponential function of n), it will answer 'yes' with high probability and provide some quasi linear structures. These quasi linear structures might be the real ones, but might only be approximate ones.

4.1. *The quantum algorithm*

We now present our algorithm, which begins with the Bernstein–Vazirani algorithm followed by a solution of a system of linear equations. This technique of a solution of equations also appears in Simon's algorithm. The details of the algorithm are as follows.

Algorithm 1

Let $p(n)$ be an arbitrary polynomial of n, and let $\varnothing$ be the empty set.

1. Initialize $H := \varnothing$, $r := p(n)$.
2. For each of r rounds,
   2.1. Run the Bernstein–Vazirani algorithm $n+1$ times to get $n+1$ vectors $w_1, \ldots, w_{n+1} \in N_f^1$.
   2.2. Update $H := H \bigcup \{w_1, \ldots, w_{n+1}\}$.
   2.3. Solve the equations $x \cdot H = 0$ and $x \cdot H = 1$ to get the solution A^0 and A^1, respectively.
   2.4. If $A^0 = \{0\}$ and $A^1 = \varnothing$, then output NO and halt.
3. Report f has quasi linear structures (YES), and output A^0, A^1.

Let us show the correctness of Algorithm 1. Since the set H satisfies $H \subseteq N_f^1$, we get $U_f^i \subseteq A^i$. Hence, when $A^0 = \{0\}$ and $A^1 = \varnothing$, the function f has no non-zero linear

https://doi.org/10.1017/S0960129516000013 Published online by Cambridge University Press

structure. On the other hand, it may not be that $U_f^i \supseteq A^i$, so an element in A^i may not be an exact linear structure, while its quasi-linearity can be demonstrated by Theorem 4.1.

Particularly, we have the following helpful results, which will be proved in the appendix. If $0 \in H$, or there are an even (and positive) number of w_j satisfying $\sum_j w_j \in H$, then $A^1 = \varnothing$, thereby $U_f^1 = \varnothing$. If there are $k = n$ linearly independent vectors in H, then $A^0 = \{0\}$, $|A^1| = 1$ or $|A^1| = 0$, and this leads to $U_f^0 = \{0\}$ and $|U_f^1| \leqslant 1$.

4.2. *Analysis of the above quantum algorithm*

Now, let us consider the following questions. First of all, If we run the algorithm for a function $f \in \mathfrak{B}_n$ at most polynomial times, what will be the result? Naturally, either we can obtain that f has no non-zero linear structure or we cannot. Second, if we cannot, that is, we discover some approximate or exact linear structures through the algorithm, then what property do all of these vectors share? Theorem 4.1 answers this question. Third, what is the smallest running time needed to determine the existence of exact linear structures? Theorem 4.2 answers this question.

Theorem 4.1. Given an oracle access to a Boolean function f with n variables, Algorithm 1 repeats the Bernstein–Vazirani algorithm $m = r \cdot (n+1)$ times, and reports that f has no non-zero linear structure or outputs vectors sets A^0 and A^1, with the following being true for all $a \in A^i\,(i = 0, 1)$:

$$\Pr\left(1 - \frac{|\{x \in F_2^n \mid f(x \oplus a) + f(x) = i\}|}{2^n} < \epsilon\right) > 1 - e^{-2m\epsilon^2}, \tag{25}$$

where ϵ is a number $(0 < \epsilon < 1)$ characterizing the error tolerance, and the left-hand side denotes the probability that $1 - |\{x \in F_2^n \mid f(x \oplus a) + f(x) = i\}|/2^n < \epsilon$.

This demonstrates that for any polynomial function $l(n) > 0$, if $\epsilon = 1/l(n)$, and $r = p(n) = l^2(n)$, the probability in Equation (25) will be greater than $1 - e^{-2n}$. Thus, vectors in the set $A = A^0 \cup A^1$ are quasi linear structures except with a negligible probability. In other words, $\forall\, a \in A^i\,(i = 0, 1)$, (a, i) is a high probability differential.

Proof of Theorem 4.1 For any $a \in A^i\,(i = 0, 1)$, let

$$\frac{|\{x \in F_2^n \mid f(x \oplus a) + f(x) = i\}|}{2^n} = \frac{|V_{f,a}^i|}{2^n} = p, \tag{26}$$

and

$$\frac{|V_{f,a}^{\bar{i}}|}{2^n} = 1 - p = q, \tag{27}$$

(obviously $p, q \in [0, 1]$). Next, we let Y be a random variable,

$$Y = Y(w) = \begin{cases} 0 & w \cdot a = i, \\ 1 & w \cdot a = \bar{i}, \end{cases} \tag{28}$$

where w is the output of the Bernstein–Vazirani algorithm. Then, by Theorem 3.1, the expectation of Y is $\mathrm{E}(Y) = 1 \cdot q = q = 1 - p$. In addition, m runs of the Bernstein–Vazirani

https://doi.org/10.1017/S0960129516000013 Published online by Cambridge University Press

algorithm correspond to m independent identical random variables $Y_1, \ldots, Y_m$. Thus, by Hoeffding's inequality (Hoeffding 1963), we obtain

$$\Pr\left(q - \frac{1}{m}\sum_{j=1}^{m} Y_j \geqslant \epsilon\right) \leqslant e^{-2m\epsilon^2}. \tag{29}$$

Now, that we have obtained $a \in A^i$, $\sum_{j=1}^{m} Y_j$ in Equation (29) must equal 0 (because if there exists some $Y_j = 1$, we cannot obtain $a \in A^i$). Hence,

$$\Pr(q \geqslant \epsilon) \leqslant e^{-2m\epsilon^2}. \tag{30}$$

From Equation (30), it follows

$$\Pr(1 - p < \epsilon) = \Pr(1 - \epsilon < p \leqslant 1) > 1 - e^{-2m\epsilon^2}.$$

□

In other words, $(1 - \frac{1}{m^\lambda}, 1]$ is the confidence interval of p with the confidence level $1 - e^{-2m^{1-2\lambda}}$ (where let $\epsilon = \frac{1}{m^\lambda}$ and $0 < \lambda \leqslant \frac{1}{2}$).

Theorem 4.2. The time required to determine whether a given Boolean function f has non-zero linear structures relies on the differential uniformity δ_f of f. More precisely, if $\delta_f < 1$, running the Bernstein–Vazirani algorithm $O(\frac{n+1}{1-\delta_f})$ times (i.e. $r = O(\frac{1}{1-\delta_f})$ in Algorithm 1) will report that f has no non-zero linear structure.

Proof of Theorem 4.2 If $\exists\, \xi \in F_2^n$, $\mu \in F_2$, such that

$$\begin{aligned}\delta_f &= \frac{1}{2^n}|\{x \in F_2^n \mid f(x \oplus \xi) + f(x) = \mu\}| \\ &= \frac{1}{2^n}\max_{0 \neq a \in F_2^n}\max_{i \in F_2}|\{x \in F_2^n \mid f(x \oplus a) + f(x) = i\}| \\ &= \delta < 1,\end{aligned} \tag{31}$$

then

$$\begin{aligned}&\frac{1}{2^n}|\{x \in F_2^n \mid f(x \oplus \xi) + f(x) = \overline{\mu}\}| \\ &= \frac{1}{2^n}\min_{0 \neq a \in F_2^n}\min_{i \in F_2}|\{x \in F_2^n \mid f(x \oplus a) + f(x) = i\}| \\ &= 1 - \delta > 0.\end{aligned} \tag{32}$$

Therefore, $\forall\, a \in F_2^n$, $a \neq 0$, $\forall\, i \in F_2$,

$$0 < 1 - \delta \leqslant \frac{1}{2^n}|\{x \in F_2^n \mid f(x \oplus a) + f(x) = i\}| \leqslant \delta < 1. \tag{33}$$

Suppose one has repeated the Bernstein–Vazirani algorithm m times, and has obtained H. Then, by Theorem 3.1, the probability that $H \subseteq B_a^i$ is at most

$$\delta^m = (1 - (1 - \delta))^m \leqslant e^{-m(1-\delta)}. \tag{34}$$

https://doi.org/10.1017/S0960129516000013 Published online by Cambridge University Press

This can be made sufficiently small if we choose m of the order $(1-\delta)^{-1}$ (i.e. $m = O(\frac{1}{1-\delta})$). In particular, if

$$m > \frac{c}{1-\delta}, \tag{35}$$

where $c > 1$ is a constant, then

$$\delta^m < \frac{1}{e^c}. \tag{36}$$

Therefore, the probability that $H \nsubseteq B_a^i$ is at least

$$1 - \delta^m > 1 - \frac{1}{e^c}. \tag{37}$$

An alternative statement is $H \cap B_a^i \neq \varnothing$ and $H \cap B_a^{\bar{i}} \neq \varnothing$ with probability close to one.

We complete now the proof by using a method from Brassard and Høyer (1997). Suppose first that we have obtained k ($0 \leqslant k < n$) linearly independent vectors $w_1, \ldots, w_k$ at some time, and $a \in F_2^n$ is a solution of the equations $w_j \cdot x = 0$ ($j = 1, \ldots, k$). After another expected number of $O(\frac{1}{1-\delta})$ measurements, we will then obtain a w_{k+1} with $w_{k+1} \cdot a = 1$. This w_{k+1} must be linearly independent with $w_1, \ldots, w_k$, because it must satisfy $w \cdot a = 0$ for any linear combination w of $w_1, \ldots, w_k$ ($j = 1, \ldots, k$). Thus, we would obtain n linearly independent vectors $w_1, w_2 \ldots, w_n$ through an expected number of $O(\frac{n}{1-\delta})$ measurements (by mathematical induction). From this fact, we can determine $U_f^0 = \{0\}$. Finally, as for the only possible solution b of the equations $w_j \cdot x = 1$, $j \in \{1, 2, \ldots, n\}$ (b is the only possible element of U_f^1), another expected number of $O(\frac{1}{1-\delta})$ measurements will give a w_{n+1} with $w_{n+1} \cdot b = 0$, showing $b \notin U_f^1$. Therefore, we obtain $U_f^1 = \varnothing$. In summary, we obtain $U_f = \{0\}$ through $m = O(\frac{n+1}{1-\delta_f})$ measurements. □

Generally, if $\delta < 1 - \frac{1}{p(n)}$ ($p(n)$ is a polynomial function of n), by Theorem 4.2, running the Bernstein–Vazirani algorithm $O(\frac{n+1}{1-\delta}) < O((n+1)p(n))$ times (i.e. $r = O(p(n))$ in Algorithm 1) will return a 'no' answer. A special case is when a Boolean function f is a Bent function (i.e. $\Pr(f(x \oplus a) = f(x)) = 1/2$ is true for every $a \neq 0$), in this case $\delta_f = \frac{1}{2}$, for which $O(n)$ runs will be sufficient. If $1 > \delta > 1 - \frac{1}{e(n)}$ (as already mentioned $e(n)$ is an exponential function of n), by Theorems 4.1 and 4.2, at least $O(e(n))$ runs will be required to give the exact result. A special case is $\delta = 1 - \frac{1}{2^{n-1}}$, which requires $O(n2^{n-1})$ runs. If we run the Bernstein–Vazirani algorithm $O(n2^n)$ times and we still obtain quasi linear structures, these must be linear structures except with a negligible probability.

5. Conclusions

A polynomial-time quantum approximate algorithm for deciding whether a function $f \in \mathfrak{B}_n$ has non-zero linear structures has been presented. By contrast, the computational complexity of the best-known classical algorithm for this problem is $O(n2^{n+1})$ (Dubuc 2001).

The mechanism of the quantum algorithm is to repeat the Bernstein–Vazirani algorithm to produce an approximate linear structures set, which includes the linear structures set and gradually tends to it with the increasing of the running time. Meanwhile, the error

https://doi.org/10.1017/S0960129516000013 Published online by Cambridge University Press

range with the error probability was given. Furthermore, the running time of the algorithm to answer the question thoroughly is linked to the differential uniformity of the function.

Finally, it should be noted that we can use a Grover-like operator to amplify the amplitude to obtain a better success probability as in Hillery and Anderson (2011) and Floess *et al.* (2013). However, this is not of our concern. The point is that the quasi linear structures obtained might be of assistance in differential cryptanalysis. In addition, the results elaborated in this paper exemplify the conclusion in Beals *et al.* (2001).

The authors would like to thank anonymous reviewers for their helpful comments and suggestions. This work was supported by the National Natural Science Foundation of China under Grant No.61173157.

Appendix A.

To complete our argument, we provide the following appendix.

Lemma A.1.

$$C_f(a) = \sum_{x \in F_2^n} (-1)^{f(x)+f(x\oplus a)} = 2^n \left(\sum_{w \in B_a^0} S_f^2(w) - \sum_{w \in B_a^1} S_f^2(w) \right),$$

Proof of the Lemma 3.1

$$\begin{aligned}
C_f(a) &= \sum_{x \in F_2^n} (-1)^{f(x)+f(x\oplus a)} \\
&= \sum_{x \in F_2^n} \left(\sum_{w \in F_2^n} S_f(w)(-1)^{w\cdot x} \right) \cdot \left(\sum_{w \in F_2^n} S_f(w)(-1)^{w\cdot (x\oplus a)} \right) \\
&= \sum_{x \in F_2^n} \left(\sum_{w \in B_a^0} S_f(w)(-1)^{w\cdot x} + \sum_{w \in B_a^1} S_f(w)(-1)^{w\cdot x} \right) \cdot \\
&\qquad \left(\sum_{w \in B_a^0} S_f(w)(-1)^{w\cdot x} - \sum_{w \in B_a^1} S_f(w)(-1)^{w\cdot x} \right) \\
&= \sum_{x \in F_2^n} \left[\left(\sum_{w \in B_a^0} S_f(w)(-1)^{w\cdot x} \right)^2 - \left(\sum_{w \in B_a^1} S_f(w)(-1)^{w\cdot x} \right)^2 \right] \\
&= \sum_{x \in F_2^n} \sum_{w \in B_a^0} \sum_{\eta \in B_a^0} S_f(w) S_f(\eta)(-1)^{(w\oplus\eta)\cdot x} - \sum_{x \in F_2^n} \sum_{w \in B_a^1} \sum_{\eta \in B_a^1} S_f(w) S_f(\eta)(-1)^{(w\oplus\eta)\cdot x} \\
&= \sum_{w \in B_a^0} \sum_{\eta \in B_a^0} S_f(w) S_f(\eta) 2^n \chi_w(\eta) - \sum_{w \in B_a^1} \sum_{\eta \in B_a^1} S_f(w) S_f(\eta) 2^n \chi_w(\eta) \\
&= 2^n \left(\sum_{w \in B_a^0} S_f^2(w) - \sum_{w \in B_a^1} S_f^2(w) \right).
\end{aligned}$$

□

https://doi.org/10.1017/S0960129516000013 Published online by Cambridge University Press

Lemma A.2 (Parseval's identity).

$$\sum_{w\in F_2^n} S_f^2(w) = \frac{1}{2^n}\sum_{x\in F_2^n}\left[(-1)^{f(x)}\right]^2,$$

Proof of Lemma A.2

$$\begin{aligned}\sum_{w\in F_2^n} S_f^2(w) &= \sum_{w\in F_2^n}\left(\frac{1}{2^n}\sum_{x\in F_2^n}(-1)^{f(x)+w\cdot x}\right)\cdot\left(\frac{1}{2^n}\sum_{y\in F_2^n}(-1)^{f(y)+w\cdot y}\right)\\ &= \frac{1}{2^n}\sum_{x\in F_2^n}(-1)^{f(x)}\sum_{y\in F_2^n}(-1)^{f(y)}\frac{1}{2^n}\sum_{w\in F_2^n}(-1)^{w\cdot(x\oplus y)}\\ &= \frac{1}{2^n}\sum_{x\in F_2^n}(-1)^{f(x)}\cdot(-1)^{f(x)}\\ &= \frac{1}{2^n}\sum_{x\in F_2^n}1^{f(x)} = 1.\end{aligned}$$

□

From Theorem 2.2, according to the knowledge regarding the solutions to the linear equations in algebra, we have the following conclusions.

Proposition A.1. † If $0\in N_f^1$, i.e. $S_f(0)\neq 0$, then $U_f^1=\varnothing$.

Proof. If $0\in N_f^1$, the equation $0\cdot x=1$ has no solution, so $U_f^1=\varnothing$. □

Proposition A.2. If $\exists\ w_1,w_2\in N_f^1$, and $w_1+w_2\in N_f^1$, then $U_f^1=\varnothing$.

Proof. If $w_1,w_2,w_1+w_2\in N_f^1$, the system of linear equations

$$\begin{cases} w_1\cdot x=1\\ w_2\cdot x=1\\ (w_1\oplus w_2)\cdot x=1\end{cases}$$

has no solution, so $U_f^1=\varnothing$. □

Proposition A.3. If $dimN_f^1=k\,(k\leqslant n)$, then $dimU_f^0=n-k$. And if $U_f^1\neq\varnothing$, then $|U_f^1|=|U_f^0|$, and $dimU_f=n-k+1$.

Proof. If $dimN_f^1=k(k\leqslant n)$, suppose $|N_f^1|=N$, and $N_f^1=\{w_1,w_2,\ldots,w_N\}$. Then, the system of linear equations

$$\begin{cases} w_1\cdot x=0\\ w_2\cdot x=0\\ w_N\cdot x=0\end{cases} \tag{A1}$$

† Proposition A.1 also appears in Feng and Pei 1999.

https://doi.org/10.1017/S0960129516000013 Published online by Cambridge University Press

has a solution space of the dimensions $n-k$. Suppose $A_f^0 = \{a_1^0, \ldots, a_{2^{n-k}}^0\}$ is the solution set of it, then by Theorem 2.2, $U_f^0 = A_f^0$. Suppose b is a special solution of the linear system of equations

$$\begin{cases} w_1 \cdot x = 1 \\ w_2 \cdot x = 1 \\ w_N \cdot x = 1. \end{cases} \tag{A2}$$

Then, $A_f^1 = \{a_1^0 \oplus b, \ldots, a_{2^{n-k}}^0 \oplus b\}$ is the solution set of it. By Theorem 2.2, $U_f^1 = A_f^1$. Therefore, $|U_f^1| = |U_f^0|$, and $dim U_f = n-k+1$. □

Proposition A.4. There might be $U_f^1 \neq \varnothing$ even if $dim N_f^1 = n$.

Proof. For example,

$$f(x_1x_2x_3) = x_1 + x_2 + x_1x_2 + x_2x_3 + x_1x_3, \tag{A3}$$

$$S_f(001) = -\frac{1}{2}, S_f(010) = S_f(100) = S_f(111) = \frac{1}{2};$$

$$S_f(000) = S_f(011) = S_f(101) = S_f(110) = 0. \tag{A4}$$

$$f(x_1x_2x_3 \oplus 111) = x_1 + x_2 + x_1x_2 + x_2x_3 + x_1x_3 + 1 = f(x_1x_2x_3) + 1. \tag{A5}$$

□

References

Aharonov, D., Jones, V. and Landau, Z. (2009). A polynomial quantum algorithm for approximating the Jones polynomial. *Algorithmica* **55** 395–421. A primary version in STOC 2006 427–436.

Beals, R., Buhrman, H., Cleve, R., Mosca, M. and Wolf, R.D. (2001). Quantum lower bound by polynomials. *Journal of the ACM* **48** (4) 778–797 (July). A primary version appeared in FOCS 1998, 352–361.

Bernstein, E. and Vazirani, U. (1993). Quantum complexity theory. In: *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, ACM Press 11–20.

Brassard, G., Dupuis, F., Gambs, S. and Tapp, A. (2011). An optimal quantum algorithm to approximate the mean and its application for approximating the median of a set of points over an arbitrary distance. arXiv: 1106.4267v1 [quant-ph] 21 Jun. 2011.

Brassard, G. and Høyer, P. (1997). An exact quanum polynimial-time algorithm for Simon's problem. In: *Proceedings of the 5th Israeli Symposium on the Theory of Computing Systems* (ISTCS'97), 12–23.

Brassard, G., Høyer, P., Mosca, M. and Tapp, A. (2002). Quantum amplitude amplification and estimation. *Contemporary Mathematics* **305** 53–74. Also at arXiv: quant-ph/0005055v1 15 May 2000.

Cleve, R., Ekert, A., Macchiavello, C. and Mosca, M. (1998). Quantum algorithms revisited. *Proceedings of the Royal Society of London, Series A* **454** 339–354.

Connor, L. and Klapper, A. (1994). Algebraic nonlinearity and its applications to cryptography. *Journal of Cryptology* **7** (4) 213–227.

Deutsch, D. and Jozsa, R. (1992). Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London, Series A* **439** 553–558.

https://doi.org/10.1017/S0960129516000013 Published online by Cambridge University Press

Drucker, A. and Wolf, R.d. (2011). Uniform approximation by (quantum) polynomials. *Quantum Information and Computation* **11** (3) 215–225. Also at Arxiv: 1008.1599 v3 [quant-ph] 14 Mar 2011.

Dubuc, S. (2001). Characterization of linear structures. *Designs, Codes and Cryptography* **22** (1) 33–45.

Feng, D.G. and Pei, D.Y. (1999). *Introduction to Cryptography (in chinese)*, Science Press, Beijing.

Feng, D.G. and Xiao, G.Z. (1995). Character of linear structure of Boolean functions. *Journal of Electronics (in Chinese)* **17** (3) 324–329.

Floess, D., Andersson, E. and Hillery, M. (2013). Quantum algorithms for testing and learning Boolean functions. *Mathematical Structures in Computer Science* **23** (2) 386–398.

Hillery, M. and Anderson, E. (2011). Quantum tests for the linearity and permutation invariance of Boolean functions. *Physical Review A* **84** 062329.

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *American Statistical Association Journal* **58** (301) 13–30 (March).

Kahn, J., Kalai, G. and Linial, N. (1988). The influence of variables on Boolean functions. In: *Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, IEEE Computer Society Press 68–80.

Lai, X.J. (1995). additive and linear structure of cryptographic functions. In: *Fast Software Encryption*, Lecture Notes in Computer Science, volume 1008, Springer-Verlag 75–85.

Nielsen, M.A. and Chuang, I.L. (2010). *Quantum Computation and Quantum Information*, Cambridge University Press.

Shor, P.W. (1997). Polynomial-time algorithm for prime factorization and discrete logarithms on quantum computer. *SIAM Journal on Computing* **26** 1484–1509. A primary version appeared in FOCS 1994, 124–134.

Simon, D.R. (1997). On the Power of Quantum Computation. *SIAM Journal on Computing* **26** 1474–1483.

O'Donnell, R. (2008). Some topics in analysis of Boolean functions. In: *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC'08)*, ACM Press, 569–578.

https://doi.org/10.1017/S0960129516000013 Published online by Cambridge University Press