THE HIDDEN SHIFT PROBLEM

A PREPRINT

Tarkan Al-Kazily **Ethan Shea** **Conner Hansen**

March 21, 2019

ABSTRACT

The topic of our project is the Hidden Shift Problem, a subset of Hidden Subgroup Problems, which provides another class of problems for which quantum computers are better suited than classical computers. In this paper, we give a brief exposition on the Hidden Shift Problem and how it relates to other topics, including bent boolean functions, Simon's Problem, and the Hidden Subgroup Problem. Alongside this paper, we have created a Kata to walkthrough the implementation of the oracles and algorithms presented in this paper.

1 Introduction

The Hidden Shift Problem presented by Rötteler[1] asks you to determine the value of s, given two bent boolean functions f and g such that $g(x) = f(x + s)$. Through our Kata and this paper, we hope to explain the requisite background needed to understand the Hidden Shift Problem, and how to solve it efficiently with a quantum computer. The most natural quantum algorithm to discuss in conjunction with the Hidden Shift Problem is Simon's Algorithm, along with the more general Hidden Subgroup Problem, that they are both instances of. After presenting two quantum solutions to the Hidden Shift Problem, we discuss how these compare with classical solutions. Finally, we conclude with a breakdown of our Hidden Shift Kata, detailing the overall structure, unit tests, and additional libraries we have included in it.

2 Boolean and Bent Functions

Formally, an n-ary boolean function is a map $\mathbb{Z}_2^n \to \mathbb{Z}_2$. We define a bent boolean function in terms of the Walsh transform. The Walsh transform of a boolean function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ is defined as:

Definition 2.1 (Walsh Transform/Fourier Coefficients)**.** The Walsh transform, $\hat{f} : \mathbb{Z}_2^n \to \mathbb{Z}$, of a boolean function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ is:

$$\hat{f}(a) = \sum_{x \in \mathbb{Z}_2^n} (-1)^{f(x)+a\cdot x} \tag{1}$$

These are (un-normalized) the Fourier Coefficients for f at frequencies a, and satisfy

$$f(x) = \sum_{a \in \mathbb{Z}_2^n} \frac{1}{2^n} \hat{f}(a)(-1)^{a\cdot x} \tag{2}$$

Note that the linear boolean functions on $\mathbb{Z}_2^n$ are precisely the functions $f(x) = a \cdot x,\ a \in \mathbb{Z}_2^n$. Thus, for each bitstring $a \in \mathbb{Z}_2^n$, the Walsh transform measures how many inputs on which f agrees with the linear function defined by a. A bent function is one that is equidistant from all linear functions, in this sense.

Definition 2.2 (Bent Function)**.** A boolean function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ is bent if $|\hat{f}(x)|$ is a constant function over $\mathbb{Z}_2^n$.

A few notes: first, verifying the "bentness" of a boolean function through the definition is an exponential problem, running in $O(2^{2n})$ time for n input bits. Second, for all bent functions $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$, n must be even, and for all

$a \in \mathbb{Z}_2^n$, $|\hat{f}(a)| = 2^{\frac{n}{2}}$. The algorithms discussed in this paper work exclusively on bent functions, and as such we will only consider functions with this property.

With every bent function, we can define another function $\tilde{f}$ called the dual of f.

Definition 2.3 (Dual Function)**.** The dual function $\tilde{f} : \mathbb{Z}_2^n \to \mathbb{Z}_2$ of bent function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ is defined such that:

$$\sqrt{2^{-n}}\hat{f}(w) =: (-1)^{\tilde{f}(w)} \tag{3}$$

That is, it is a boolean function over the same domain mapping to zero any input for which the Walsh transform of f is positive, and to one any input for which it is negative. This function is bent as well, and its dual is f.

We work with two specific classes of bent function in our katas: quadratic and inner product.

Definition 2.4 (Inner Product Function)**.** For n even, the inner product function on n bits is:

$$f(x_1, \ldots, x_{\frac{n}{2}}, y_1, \ldots, y_{\frac{n}{2}}) = \Sigma_{i=1}^{\frac{n}{2}} x_i y_i \tag{4}$$

The inner product functions are bent, and additionally have the property that they are their own dual.

Definition 2.5 (Quadratic Bent Function)**.** The class of bent quadratic boolean functions $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ can be expressed in the following form:

$$f(x) = xQx^t + Lx^t \tag{5}$$

Where $L \in \mathbb{Z}_2^n$, $Q \in \mathbb{Z}_2^{n \times n}$, Q is upper-triangular with an all-zero diagonal, and the matrix $Q + Q^t$ has full rank.

The dual of a quadratic bent function is another bent quadratic function.

See Rötteler [1] for additional classifications, theorems, justifications, and properties concerning bent functions beyond the scope of this paper.

3 The Hidden Shift Problem

Recall Simon's Problem, in which we are given a quantum oracle for a function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^{n-1}$ that promises that $f(y) = f(x)$ if and only if $y = x + s$ for all $y, x \in \mathbb{Z}_2^n$, with which we are tasked with finding $s \in \mathbb{Z}_2^n$. It's known that the most efficient classical algorithm will make exponential calls to f, but that we can determine s in linear time with a constant probability of success with the quantum circuit in Figure 1.

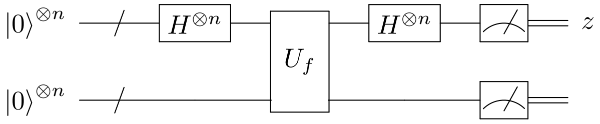

Figure 1: Simon's Algorithm Circuit

Each application of this circuit produces a random measurement z with the property that the inner product $sz^T = 0$, and with $n - 1$ independent outcomes we are able to compute s through Gaussian elimination, which happens with a constant probability of success.

Simon's Problem demonstrates that there are tasks that are impossible for a classical computer to solve efficiently, but are easily computed with a quantum computer. The question is how does this generalize to a larger class of problems, which brings our focus to the Hidden Shift Problem.

Definition 3.1 (Hidden Shift Problem, [1])**.** Let $n \geq 1$ and $\mathcal{O}_f$ be an oracle for two Boolean functions $f, g : \mathbb{Z}_2^n \to Z_2$ such that (i) f and g are bent functions (see Definition 2.2) and (ii) $g(x) = f(x + s)$ for all $x \in \mathbb{Z}_2^n$. We say that $\mathcal{O}_f$ hides an instance of a shifted bent function problem for the bent function f and the hidden shift $s \in \mathbb{Z}_2^n$. If in addition to f and g the oracle provides access to the dual bent function $\tilde{f}$, then we use the notation $\mathcal{O}_{f,\tilde{f}}$ to indicate this potentially more powerful oracle.

Both Simon's Problem and the Hidden Shift Problem are examples of the Hidden Subgroup Problem, which is heavily rooted in Group Theory.

Definition 3.2 (Hidden Subgroup Problem, [2])**.** Let f be a function from a finitely generated group G to a finite set X such that f is constant on the cosets of a subgroup K, and distinct on each coset. Given a quantum black box for performing the unitary transform $U|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$, for $x \in G, y \in X$, and $\oplus$ an appropriately chosen binary operation on X, find a generating set for K.

In the case of Simon's Problem, G is the set of binary strings $\mathbb{Z}_2^n$ with the operation $+$ that is elementwise addition modulo 2, and the function f is a Boolean function that is constant on the cosets determined by the subgroup $\{0, s\}$. In 3.1.1 we will see how to construct an instance of Simon's Problem from the Hidden Shift Problem, and therefore tie it back to the Hidden Subgroup Problem.

3.1 Solutions to the Hidden Shift Problem

First, we will see how a quantum computer can efficiently solve the Hidden Shift Problem by presenting two quantum algorithms initially given by Rötteler [1], and the implementation of which is broken down into two tasks in our kata. We then briefly compare these to classical solutions of the Hidden Shift Problem.

3.1.1 Quantum Solutions

Deterministic Hidden Shift Solution

We can solve the Hidden Shift Problem most efficiently if we are working with an oracle $\mathcal{O}_{f,\tilde{f}}$ with the circuit in Figure 2. Here, $U_{\tilde{f}}$ and U_g are phase flip oracles provided by $\mathcal{O}_{f,\tilde{f}}$ for the functions $\tilde{f}$ and g respectively.

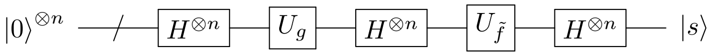

Figure 2: Deterministic Circuit to Solve the Hidden Shift Problem

This algorithm works through the following steps:

i) First, prepare the state $|0\rangle^{\otimes n}$.

ii) Apply the Walsh-Hadamard transform $H^{\otimes n}$ to produce an equal superposition on all states.

iii) Apply U_g. This produces the state $\frac{1}{\sqrt{2^n}}\sum_{x\in\mathbb{Z}_2^n}(-1)^{g(x)}|x\rangle = \frac{1}{\sqrt{2^n}}\sum_{x\in\mathbb{Z}_2^n}(-1)^{f(x+s)}|x\rangle$. Note that since addition in $\mathbb{Z}_2^n$ is a bijection, we can also think of this state as $\frac{1}{\sqrt{2^n}}\sum_{x\in\mathbb{Z}_2^n}(-1)^{f(x)}|x+s\rangle$.

iv) Apply the Walsh-Hadamard transform $H^{\otimes n}$ again. This produces the state

$$\begin{aligned}\frac{1}{\sqrt{2^n}}\sum_{x\in\mathbb{Z}_2^n}(-1)^{f(x)}\left(\frac{1}{\sqrt{2^n}}\sum_{y\in\mathbb{Z}_2^n}(-1)^{(x+s)y^T}|y\rangle\right) &= \sum_{y\in\mathbb{Z}_2^n}(-1)^{sy^T}\left(\frac{1}{2^n}\sum_{x\in\mathbb{Z}_2^n}(-1)^{f(x)}(-1)^{xy^T}\right)|y\rangle \\ &= \sum_{y\in\mathbb{Z}_2^n}(-1)^{sy^T}\hat{f}(y)\,|y\rangle \\ &= \frac{1}{\sqrt{2^n}}\sum_{y\in\mathbb{Z}_2^n}(-1)^{sy^T}(-1)^{\tilde{f}(y)}\,|y\rangle \quad \text{(by 3)}.\end{aligned}$$

v) Apply $U_{\tilde{f}}$. This is just the phase flip oracle that takes $|y\rangle \mapsto (-1)^{\tilde{f}(y)}|y\rangle$, so applying this takes the state to $\frac{1}{\sqrt{2^n}}\sum_{y\in\mathbb{Z}_2^n}(-1)^{sy^T}|y\rangle$.

vi) Apply the Walsh-Hadamard transform $H^{\otimes n}$ to transform the state to $|s\rangle$.

This algorithm allows us to deterministically measure s in only two calls to our oracle $\mathcal{O}_{f,\tilde{f}}$ and a linear number of Hadamard operations.

Hidden Subgroup Based Solution

If we don't have access to an oracle implementing the dual of f, we can reduce the Hidden Shift Problem to Simon's Problem by constructing a periodic function $H(b,x) : \mathbb{Z}_2^{n+1} \to \mathbb{Z}_2^n$ in the following way. First, use f and g to construct new functions

$$F : x \mapsto \sum_{y\in\mathbb{Z}_2^n}(-1)^{f(x+y)}|y\rangle\,,\; G : x \mapsto \sum_{y\in\mathbb{Z}_2^n}(-1)^{g(x+y)}|y\rangle\,.$$

These still satisfy that $G(x) = F(x+s)$, and further are injective functions, since for all $a, b \in \mathbb{Z}_2^n$, if $F(a) = F(b)$, then

$$\begin{aligned}
\sum_{y\in\mathbb{Z}_2^n} (-1)^{f(a+y)} |y\rangle &= \sum_{y\in\mathbb{Z}_2^n} (-1)^{f(b+y)} |y\rangle \\
(-1)^{f(a+y)} &= (-1)^{f(b+y)} \text{ for all } y \in \mathbb{Z}_2^n \\
f(a+y) + f(b+y) &= 0 \text{ for all } y \in \mathbb{Z}_2^n \\
\Delta_{a-b} f(y) &= 0 \text{ for all } y \in \mathbb{Z}_2^n
\end{aligned}$$

A nice property of bent functions is that their derivatives $\Delta_h f(x) = f(x+h) + f(x)$ are balanced boolean functions for all $h \neq 0 \in \mathbb{Z}_2^n$, meaning they take on values 0 and 1 equal number of times. Since the constant 0 function is not balanced, this means that $F(a)$ equaling $F(b)$ (or $G(a) = G(b)$) will imply that a must equal b, and that F and G are injective.

With F and G, define $H(b, x)$ such that $H(0, x) = F(x)$ and $H(1, x) = G(x)$. We can see then that $H(b, x)$ satisfies that $H(b, x) = H(b+1, x+s)$ for all $b \in \mathbb{Z}_2$ and $x \in \mathbb{Z}_2^n$, and that $H(b, x)$ will take on different values otherwise (due to injectivity of F and G). $H(b, x)$ satisfies the requirements for the Hidden Subgroup Problem in the same way as Simon's Problem, with G being the group $\mathbb{Z}_2^{n+1}$ and K being the subgroup generated by $(1, s)$. We can then solve the Hidden Shift Problem in the same way as Simon's Algorithm.

We can implement the quantum circuit for $H(b, x)$ with the circuit in Figure 3, where U_g and U_f are phase flip oracles for g and f.

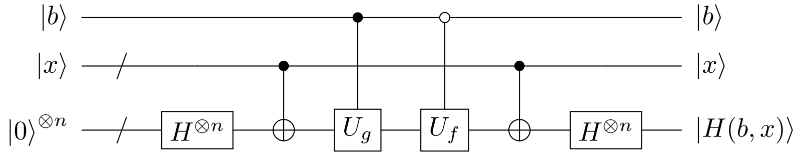

Figure 3: The Hiding Function $H(b, x)$

With a quantum circuit U_H for $H(b, x)$ implemented, we simply need to apply Simon's Algorithm as in Figure 4 to generate elements $a \in \mathbb{Z}_2^{n+1}$ that satisfy $(1, s)a^T = 0$. As in Simon's Algorithm, with $O(n)$ queries to our oracles for f and g through $H(b, x)$ and constant probability, we can produce a system of equations with which to determine s.

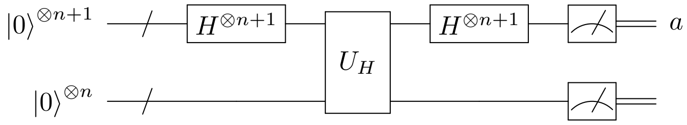

Figure 4: Using Simon's Algorithm to solve the Hidden Shift Problem

3.1.2 Classical Solutions

The naive classical solution to the Hidden Shift Problem for an oracle $\mathcal{O}_f$, much like the classical solution to Simon's Problem, is to sample the functions f and g on every input until determining the period s, requiring $O(2^n)$ queries to the functions. There aren't any significantly better classical algorithms, however. The minimum number of queries turns out to be on the order of $\Theta(\sqrt{2^n})$, just as with Simon's Problem (see [1], Theorem 9), meaning the quantum algorithm gives an exponential speedup.

With access to the dual $\tilde{f}$ of f as well as f and g, an improved classical algorithm can determine the shift in $\Theta(n)$ queries (the proof of which relies on a more complex class of bent functions than presented in this paper, and is given in [1], Theorem 8). We can see that the quantum speedup is no longer exponential, but instead reduces a linear number of queries to a finite number.

It is worth discussing the complexity involved in producing an oracle for $\tilde{f}$, as it can take an exponential number of queries of f just to implement one query for $\tilde{f}$. As such, realizing the oracle $\mathcal{O}_{f,\tilde{f}}$ may not always efficient.

4 The Hidden Shift Kata

4.1 Kata Structure

The kata is broken into three tasks. Task 1 introduces basic bent functions, and has the user implement oracles for the quadratic and inner product bent functions. Additionally, the user also implements a marking to phase oracle converter and an oracle shifter, both of which are necessary to test and implement the Hidden Shift Problem algorithms.

Task 2 first asks the user to implement the Wash-Hadamard transform. This subtask encourages the user to think about the deterministic algorithm in terms of the Wash transform (which defines bent functions) rather than in terms of Hadamard gates. The user is then prepared to implement the full deterministic algorithm given by the circuit in Figure 2 in the second subtask.

Task 3 begins with implementing the hiding function oracle for $H(b, x)$ from oracles for f and g as in the circuit in Figure 3. With the hiding function implemented, the user should see the similarity between this problem and Simon's Algorithm, and be able to implement a single iteration of the full algorithm, just as in the Simon's Algorithm Kata, so that it returns single vector orthogonal to the hidden shift s. The classical part of the algorithm to construct s is implemented as part of the test suite, and will pass when this functionality is implemented.

4.2 Testing

All test cases are performed on multiple inputs of varying sizes to ensure that the solutions are general. For the inner product and quadratic oracle tests, the classical solutions are enumerated and then verified against the user's quantum implementation. To test the user's implementations of the non-measuring subtasks found at the end of task 1 and the start of tasks 2 and 3, the user's solution is compared to the reference solution. To test the actual algorithms for the deterministic and generalized algorithms, we test the algorithms on instances of the Hidden Shift Problem using the inner product reference oracles on varying input sizes using every possible shift s. We validate the deterministic algorithm by comparing the returned array to s directly. For the iterative algorithm, we both test that their result is orthogonal to $(1, s)$, not uniformly 0, and that running the algorithm enough times will eventually allow us to determine s through Gaussian elimination.

4.3 Gaussian Elimination

A small Q# linear algebra library is provided to allow for easy implementation of the Guassian Elimination portion of the full algorithm, both in the test suite and by the user if so desired. This library is able to compute the row reduced echelon form, kernel, and rank of an input matrix in $\mathbb{Z}_2^{m \times n}$. A static test suite is also provided for the library to help verify correctness.

5 Conclusion

Through this paper and our kata, we have introduced the Hidden Shift Problem on bent boolean functions, and the quantum algorithms that demonstrate an exponential speedup over classical solutions. Future work on the kata includes expanding Task 1 to include additional classes of bent functions, and to expand Tasks 2 and 3 to ask the user to implement their own tests for the Hidden Shift Problem. Behind the scenes, we also would like to improve the test suite by using more bent functions with our tests. Future work on the problem itself could involve searching for additional classes of boolean function on which this speedup can be achieved, for example, by determining if looser restrictions on the input functions would suffice. Additionally, new katas exploring other instances of the Hidden Subgroup Problem could help teach how to encode groups into quantum oracles to solve the large class of problems that the Hidden Subgroup Problem presents.

References

[1] Rötteler, Martin, *Quantum Algorithms for Highly Non-Linear Boolean Functions* Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'10), NEC Laboratories, America, pages 448–457, (2010).

[2] Nielson, Michael A. and Chuang, Isaac L. *Quantum Computation and Quantum Information, 10th Ed.* Cambridge University Press; Anniversary edition (January 31, 2011).